

List of all Scripts in alphabetical order I			
Name of the Program	Signature	Directory	.log ?
A1111.SCRIP	Script-41	KSS	
A1111LOAD_UPD.SCRIP	Script-42	KSS	
A1111_COPY.SCRIP	Script-59	KSS	
A1111_SHIPS.SCRIP	Script-43	KSS	Log-43
A1111_SHIPS_UPD.SCRIP	Script-44	KSS	Log-44
AARI.SCRIP	Script-1	OOS	
AARIINDEX.SCRIP	Script-16	KSS	
AARILDGO.SCRIP	Script-140	KSS?	Log-140
AARI_AND_SHIP_TO_CRUISE.SCRIP	Script-78	KSS	Log-78
AARI_COPY.SCRIP	Script-61	OOS	
AARI_CRUISE_VIEW.SCRIP	Script-79	KSS	Log-79
AARI_STAT.SCRIP	Script-3	OOS	
AARI_UPDATE_SHIPS.SCRIP	Script-39	KSS	Log-39
ADD_KUROPATKIN_SHIPS.SCRIP	Script-80	KSS	Log-80
ADD_SHIPS_JARE140491.SCRIP	Script-76	KSS	Log-76
ADD_TOKYO_SHIPS.SCRIP	Script-35	KSS	Log-35
ALTER_TAB_SHIP.SCRIP	Script-81	KSS	
ARGENTINE.SCRIP	Script-103	KSS	Log-103
ARGENTINELOAD_COPY.SCRIP	Script-104	KSS	Log-104
ARGENTINELOAD_UPD.SCRIP	Script-127	KSS	Log-127
AWI.SCRIP	Script-105	KSS	Log-105
AWILOAD_COPY.SCRIP	Script-128	KSS	Log-128
AWILOAD_UPD.SCRIP	Script-129	KSS	Log-129
BIOMASS.SCRIP	Script-49	KSBS	Log-49
BIOMASS_BRANDINI.SCRIP	Script-58	KSBS	Log-58
BIOMASS_BRANDINI_2.SCRIP	Script-94	KSBS	Log-94
BIOMASS_BRANDINI_UPD.SCRIP	Script-95	KSBS	Log-95
BIOMASS_BRANDINI_UPD2.SCRIP	Script-98	KSBS	Log-98
BIOMASS_COPY_ALL.SCRIP	Script-73	KSBS	Log-73
BIOMASS_UPD.SCRIP	Script-75	KSBS	Log-75
BIOSELPOSDAT.SCRIP	Script-60	KSBS	
BSHLOAD.SCRIP	Script-116	KSS	Log-116
BSHLOAD_COPY.SCRIP	Script-118	KSS	Log-118
BSHLOAD_UPD.SCRIP	Script-117	KSS	Log-117
CHANGECOUNTRYOFSHIP.SCRIP	Script-125	KSS	Log-125
CHANGECRUISE050392.SCRIP	Script-122	KSS	Log-122
CHANGESHIPNAMEANDCOUNTRYINCRUISE.SCRIP	Script-107	KSS	Log-107
CHANGESHIPNAMEINCRUISE.SCRIP	Script-82	KSS	Log-82
CHANGE_VALIDATION_FLAGS.SCRIP	Script-130	KSS	Log-130
C_V_F.SCRIP	Script-136	KSS	Log-136
CVF1250001.SCRIP	Script-132	KSS	
CHECKGARBAGESTANDARDDATA.SCRIP	Script-83	KSS	
CHECKGCSD.SCRIP	Script-12	KSS	
COPY_AARI_TO_CRUISE.SCRIP	Script-97	KSS	Log-97
COPY_GORDON_SHIP_TO_SHIP.SCRIP	Script-84	KSS	Log-84
COPY_GORDON_SHIP_TO_SHIP_PROC.SCRIP	Script-85	KSS	Log-85
CREATEBIBLIOTHEK.SCRIP	Script-138	KSS	LOG-138
CRUISE.SCRIP	Script-26	KSS	Log-26
CRUISE_TABLE.SCRIP	Script-29	KSS	
CRUISE_TEST.SCRIP	Script-27	KSS	
CRUISE_TRIGGER.SCRIP	Script-131	KSS	Log-131
CRUISE_UPD070291.SCRIP	Script-30	KSS	Log-30

List of all Scripts in alphabetical order II			
Name of the Program	Signature	Directory	.log ?
DROP_CRUISE_TRIGGER.SCRIP	Script-86	KSS	
DUMPTRANNOLOG.SCRIP	Script-133	KSS	
GARBAGECOLLECT.SCRIP	Script-13	KSS	
GC.SCRIP	Script-11	KSS	
GONELLA.SCRIP	Script-21	KSS	
GONELLA_COPY.SCRIP	Script-23	KSS	
GONELLA_COPY_ALL.SCRIP	Script-24	KSS	
GONELLA_UPD.SCRIP	Script-22	KSS	
GORDON.SCRIP	Script-2	OOS	
GORDON_COPY.SCRIP	Script-4	OOS	
GORDON_CRUISE.SCRIP	Script-6	KSS	
GORDON_INTERPOLATED_DATA.SCRIP	Script-5	OOS	
GORDON_INTER_DATA.SCRIP	Script-68	OOS	
GORDON_STAT.SCRIP	Script-8	OOS	
GSBACKFILL.SCRIP	Script-14	KSS	
GSBACKFILL_UPD070291.SCRIP	Script-32	KSS	
GSB_UPD.SCRIP	Script-77	KSS	
HAINESLOAD.SCRIP	Script-19	KSS	
HAINESLOAD_COPY.SCRIP	Script-25	KSS	
HAINESLOAD_UPD.SCRIP	Script-20	KSS	
HAINES_SHIPS020991.SCRIP	Script-88	KSS	Log-88
INDEX.SCRIP	Script-62	OOS	
INSERTCRUISE081091.SCRIP	Script-101	KSS	Log-101
INSERTCRUISE130392.SCRIP	Script-123	KSS	Log-123
INSERTONESHIP.SCRIP	Script-115	KSS	Log-115
INSERTSHIP.SCRIP	Script-53	KSS	Log-53
INSERTSHIPCRUISE.SCRIP	Script-54	KSS	Log-54
JARELOAD.SCRIP	Script-46	KSS	
JARE_UPD.SCRIP	Script-48	KSS	
KUROPATKIN.SCRIP	Script-17	KSS	
KUROPATKIN_UPD.SCRIP	Script-18	KSS	
MAKESTATION.SCRIP	Script-65	OOS	
MARION_DUFRESNE_CRUISE.SCRIP	Script-134	KSS	Log-134
MAX_MIN_FROM_SHIP.SCRIP	Script-33	KSS	
MUENCHLOAD.SCRIP	Script-70	KSS	
MUENCHLOAD_UPD.SCRIP	Script-72	KSS	Log-72
MUENCH_COPY_ALL.SCRIP	Script-71	KSS	Log-71
NOWLIN.SCRIP	Script-10	KSS	
NOWLIN_SHIPS.SCRIP	Script-89	KSS	Log-89
NOWLIN_UPD.SCRIP	Script-9	KSS	
PROGRAMTABLE.SCRIP	Script-69	KSS	
REBUILD_ARGENTINE.SCRIP	Script-106	KSS	Log-106
RESETNODC200292.SCRIP	Script-121	KSS	Log-121
ROLLBACK_COMMIT.SCRIP	Script-96	KSBS	
SCHLITZER.SCRIP	Script-108	KSS	Log-108
SCHLITZERLOAD_COPY.SCRIP	Script-110	KSS	Log-110
SCHLITZERLOAD_UPD.SCRIP	Script-109	KSS	Log-109
SHIPCRUISECORRECTION260891.SCRIP	Script-99	KSS	Log-99
SHIPCRUISECORRECTION081091.SCRIP	Script-100	KSS	Log-100

List of all Scripts in alphabetical order III			
Name of the Program	Signature	Directory	.log ?
SHIPS_UPD050291.SCRIP	Script-50	KSS	Log-50
SHIP_UPD_041291.SCRIP	Script-111	KSS	Log-111
SHIP_UPD_051291.SCRIP	Script-112	KSS	Log-112
SHIP_UPD_051291_2.SCRIP	Script-113	KSS	Log-113
SO_USER.SCRIP	Script-137	KSS	Log-137
STATION.SCRIP	Script-64	OOS	
STATION_INDEX.SCRIP	Script-66	OOS	
TOKYO_COPY.SCRIP	Script-45	KSS	
TOKYO_FISHERIES_STATION_UPDATE.SCRIP	Script-38	KSS	
TOKYO_FISH_BOTTOM_DEPTH_UPDATE.SCRIP	Script-37	KSS	
TOKYO_TABLE.SCRIP	Script-34	KSS	
TOKYO_UPD.SCRIP	Script-36	KSS	
TRIGGER.SCRIP	Script-15	KSS	
TRIGGER_DROP.SCRIP	Script-7	KSS	
TRIGGER_UPDATE_LOST_STATIONS.SCRIP	Script-87	KSS	
TRUNCATESHIP.SCRIP	Script-40	KSS	Log-40
UNKNOWNCHANGES291091.SCRIP	Script-47	KSS	Log-47
UNKNOWN_SHIP.SCRIP	Script-90	KSS	Log-90
UPDATECRU24.SCRIP	Script-57	KSS	Log-57
UPDATECRU24S.SCRIP	Script-74	KSS	Log-74
UPDATECRUISE.SCRIP	Script-55	KSS	Log-55
UPDATENODC.SCRIP	Script-119	KSS	Log-119
UPDATENODC200292.SCRIP	Script-120	KSS	Log-120
UPDATESHIPCRUISE030292.SCRIP	Script-114	KSS	Log-114
UPDATESHIPCRUISE060492.SCRIP	Script-126	KSS	Log-126
UPDATESHIPCRUISE100691.SCRIP	Script-56	KSS	
UPDATESHIPCRUISE270392.SCRIP	Script-124	KSS	Log-124
UPDATESHIPCRUISE290891.SCRIP	Script-91	KSS	Log-91
UPDATESHIP190193.SCRIP	Script-139	KSS?	
UPDATESTATIONTRG.SCRIP	Script-93	OOS	Log-93
UPDATEUNKNOWNSHIP.SCRIP	Script-102	KSS	Log-102
UPDATE_BACKFILL_PROC.SCRIP	Script-31	KSS	
UPDATE_DEFAULT_NULL.SCRIP	Script-63	OOS	
UPDATE_NULL_GORDON_INTER_DATA.SCRIP	Script-67	OOS	
UPDATE_SHIPS060991.SCRIP	Script-92	KSS	Log-92
UPDATE_SHIP_ID.SCRIP	Script-52	KSS	
UPDATE_SHIP_ID_IN_CRUISE.SCRIP	Script-51	KSS	
UPDATE_SODE.SCRIP	Script-28	KSS	
VALIDATION_FLAG_DEFAULT.SCRIP	Script-135	KSS	Log-135

List of all Scripts in numerical order I			
Name of the Program	Signature	Directory	.log ?
AARI .SCRIPT	Script-1	OOS	
GORDON .SCRIPT	Script-2	OOS	
AARI_STAT .SCRIPT	Script-3	OOS	
GORDON_COPY .SCRIPT	Script-4	OOS	
GORDON_INTERPOLATED_DATA .SCRIPT	Script-5	OOS	
GORDON_CRUISE .SCRIPT	Script-6	KSS	
TRIGGER_DROP .SCRIPT	Script-7	KSS	
GORDON_STAT .SCRIPT	Script-8	OOS	
NOWLIN_UPD .SCRIPT	Script-9	KSS	
NOWLIN .SCRIPT	Script-10	KSS	
GC .SCRIPT	Script-11	KSS	
CHECKGCSD .SCRIPT	Script-12	KSS	
GARBAGECOLLECT .SCRIPT	Script-13	KSS	
GSBACKFILL .SCRIPT	Script-14	KSS	
TRIGGER .SCRIPT	Script-15	KSS	
AARIINDEX .SCRIPT	Script-16	KSS	
KUROPATKIN .SCRIPT	Script-17	KSS	
KURDPATKIN_UPD .SCRIPT	Script-18	KSS	
HAINESLOAD .SCRIPT	Script-19	KSS	
HAINESLOAD_UPDATE .SCRIPT	Script-20	KSS	
GONELLA .SCRIPT	Script-21	KSS	
GONELLA_UPD .SCRIPT	Script-22	KSS	
GONELLA_COPY .SCRIPT	Script-23	KSS	
GONELLA_COPY_ALL .SCRIPT	Script-24	KSS	
HAINESLOAD_COPY .SCRIPT	Script-25	KSS	
CRUISE .SCRIPT	Script-26	KSS	Log-26
CRUISE_TEST .SCRIPT	Script-27	KSS	
UPDATE_SODB .SCRIPT	Script-28	KSS	
CRUISE_TABLE .SCRIPT	Script-29	KSS	
CRUISE_UPD070291 .SCRIPT	Script-30	KSS	Log-30
UPDATE_BACKFILL_PROC .SCRIPT	Script-31	KSS	
GSBACKFILL_UPD070291 .SCRIPT	Script-32	KSS	
MAX_MIN_FROM_SHIP .SCRIPT	Script-33	KSS	
TOKYOTABLE .SCRIPT	Script-34	KSS	
ADD_TOKYO_SHIPS .SCRIPT	Script-35	KSS	Log-35
TOKYO_UPD .SCRIPT	Script-36	KSS	
TOKYO_FISH_BOTTOM_DEPTH_UPDATE .SCRIPT	Script-37	KSS	
TOKYO_FISHERIES_STATION_UPDATE .SCRIPT	Script-38	KSS	
AARI_UPDATE_SHIPS .SCRIPT	Script-39	KSS	Log-39
TRUNCATESHIP .SCRIPT	Script-40	KSS	Log-40
A1111 .SCRIPT	Script-41	KSS	
A1111LOAD_UPD .SCRIPT	Script-42	KSS	
A1111_SHIPS .SCRIPT	Script-43	KSS	Log-43
A1111_SHIPS_UPD .SCRIPT	Script-44	KSS	Log-44
TOKYO_COPY .SCRIPT	Script-45	KSS	
JARELOAD .SCRIPT	Script-46	KSS	
UNKNOWNCHANGES291091 .SCRIPT	Script-47		Log-47
JARE_UPD .SCRIPT	Script-48	KSS	
BIOMASS .SCRIPT	Script-49	KSBS	Log-49
SHIPS_UPD050291 .SCRIPT	Script-50	KSS	Log-50

List of all Scripts in numerical order II			
Name of the Program	Signature	Directory	.log ?
UPDATE_SHIP_ID_IN_CRUISE.SCRIP	Script-51	KSS	
UPDATE_SHIP_ID.SCRIP	Script-52	KSS	
INSERTSHIP.SCRIP	Script-53	KSS	Log-53
INSERTSHIPCRUISE.SCRIP	Script-54	KSS	Log-54
UPDATECRUISE.SCRIP	Script-55	KSS	Log-55
UPDATESHIPCRUISE100691.SCRIP	Script-56	KSS	
UPDATECRU24.SCRIP	Script-57	KSS	Log-57
BIOMASS_BRANDINI.SCRIP	Script-58	KSBS	Log-58
A1111_COPY.SCRIP	Script-59	KSS	
BIOSELPOSDAT.SCRIP	Script-60	KSBS	
AARI_COPY.SCRIP	Script-61	OOS	
INDEX.SCRIP	Script-62	OOS	
UPDATE_DEFAULT_NULL.SCRIP	Script-63	OOS	
STATION.SCRIP	Script-64	OOS	
MAKESTATION.SCRIP	Script-65	OOS	
STATION_INDEX.SCRIP	Script-66	OOS	
UPDATE_NULL_GORDON_INTER_DATA.SCRIP	Script-67	OOS	
GORDON_INTER_DATA.SCRIP	Script-68	OOS	
PROGRAMTABLE.SCRIP	Script-69	KSS	
MUENCHLOAD.SCRIP	Script-70	KSS	
MUENCH_COPY_ALL.SCRIP	Script-71	KSS	Log-71
MUENCHLOAD_UPD.SCRIP	Script-72	KSS	Log-72
BIOMASS_COPY_ALL.SCRIP	Script-73	KSBS	Log-73
UPDATECRU24S.SCRIP	Script-74	KSS	Log-74
BIOMASS_UPD.SCRIP	Script-75	KSBS	Log-75
ADD_SHIPS_JARE140491.SCRIP	Script-76	KSS	Log-76
GSB_UPD.SCRIP	Script-77	KSS	
AARI_AND_SHIP_TO_CRUISE.SCRIP	Script-78	KSS	Log-78
AARI_CRUISE_VIEW.SCRIP	Script-79	KSS	Log-79
ADD_KUROPATKIN_SHIPS.SCRIP	Script-80	KSS	Log-80
ALTER_TAB_SHIP.SCRIP	Script-81	KSS	
CHANGESHIPNAMEINCRUISE.SCRIP	Script-82	KSS	Log-82
CHECKGARBAGESTANDARDDATA.SCRIP	Script-83	KSS	
COPY_GORDON_SHIP_TO_SHIP.SCRIP	Script-84	KSS	Log-84
COPY_GORDON_SHIP_TO_SHIP_PROC.SCRIP	Script-85	KSS	Log-85
DROP_CRUISE_TRIGGER.SCRIP	Script-86	KSS	
TRIGGER_UPDATE_LOST_STATIONS.SCRIP	Script-87	KSS	
HAINES_SHIPS020991.SCRIP	Script-88	KSS	Log-88
NOWLIN_SHIPS.SCRIP	Script-89	KSS	Log-89
UNKNOWN_SHIP.SCRIP	Script-90	KSS	Log-90
UPDATESHIPCRUISE290891.SCRIP	Script-91	KSS	Log-91
UPDATE_SHIPS060991.SCRIP	Script-92	KSS	Log-92
UPDATESTATIONTRG.SCRIP	Script-93	OOS	Log-93
BIOMASS_BRANDINI_2.SCRIP	Script-94	KSBS	Log-94
BIOMASS_BRANDINI_UPD.SCRIP	Script-95	KSBS	Log-95
ROLLBACK_COMMIT.SCRIP	Script-96	KSBS	
COPY_AARI_TO_CRUISE.SCRIP	Script-97	KSS	Log-97
BIOMASS_BRANDINI_UPD2.SCRIP	Script-98	KSBS	Log-98
SHIPCRUISECORRECTION260891.SCRIP	Script-99	KSS	Log-99
SHIPCRUISECORRECTION081091.SCRIP	Script-100	KSS	Log-100

List of all Scripts in numerical order III			
Name of the Program	Signature	Directory	.log ?
INSERTCRUISE081091.SCRIP	Script-101	KSS	Log-101
UPDATEUNKNOWNSHIP.SCRIP	Script-102	KSS	Log-102
ARGENTINE.SCRIP	Script-103	KSS	Log-103
ARGENTINELOAD_COPY.SCRIP	Script-104	KSS	Log-104
AWI.SCRIP	Script-105	KSS	Log-105
REBUILD_ARGENTINE.SCRIP	Script-106	KSS	Log-106
CHANGESHIPNAMEANDCOUNTRYINCRUISE.SCRIP	Script-107	KSS	Log-107
SCHLITZER.SCRIP	Script-108	KSS	Log-108
SCHLITZERLOAD_UPD.SCRIP	Script-109	KSS	Log-109
SCHLITZERLOAD_COPY.SCRIP	Script-110	KSS	Log-110
SHIP_UPD_041291.SCRIP	Script-111	KSS	Log-111
SHIP_UPD_051291.SCRIP	Script-112	KSS	Log-112
SHIP_UPD_051291_2.SCRIP	Script-113	KSS	Log-113
UPDATESHIPCRUISE030292.SCRIP	Script-114	KSS	Log-114
INSERTONESHIP.SCRIP	Script-115	KSS	Log-115
BSHLOAD.SCRIP	Script-116	KSS	Log-116
BSHLOAD_UPD.SCRIP	Script-117	KSS	Log-117
BSHLOAD_COPY.SCRIP	Script-118	KSS	Log-118
UPDATENODC.SCRIP	Script-119	KSS	Log-119
UPDATENODC200292.SCRIP	Script-120	KSS	Log-120
RESETNODC200292.SCRIP	Script-121	KSS	Log-121
CHANGECRUISE050392.SCRIP	Script-122	KSS	Log-122
INSERTCRUISE130392.SCRIP	Script-123	KSS	Log-123
UPDATESHIPCRUISE270392.SCRIP	Script-124	KSS	Log-124
CHANGECOUNTRYOFSHIP.SCRIP	Script-125	KSS	Log-125
UPDATESHIPCRUISE060492.SCRIP	Script-126	KSS	Log-126
ARGENTINELOAD_UPD.SCRIP	Script-127	KSS	Log-127
AWILOAD_COPY.SCRIP	Script-128	KSS	Log-128
AWILOAD_UPD.SCRIP	Script-129	KSS	Log-129
CHANGE_VALIDATION_FLAGS.SCRIP	Script-130	KSS	Log-130
CRUISE_TRIGGER.SCRIP	Script-131	KSS	Log-131
CVF1250001.SCRIP	Script-132	KSS	
DUMPTRANNOLOG.SCRIP	Script-133	KSS	
MARION_DUFRESNE_CRUISE.SCRIP	Script-134	KSS	Log-134
VALIDATION_FLAG_DEFAULT.SCRIP	Script-135	KSS	Log-135
C_V_F.SCRIP	Script-136	KSS	Log-136
SO_USER.SCRIP	Script-137	KSS	Log-137
CREATEBIBLIOTHEK.SCRIP	Script-138	KSS	Log-138
UPDATESHIP190193.SCRIP	Script-139	KSS?	
AARILDGO.SCRIP	Script-140	KSS?	Log-140

```
/*
** FILENAME: AARI.SCRIPT
*/
/* Sybase-Script made in June 1989 by M. Reinke, Stefan Richter, */
/* AWI-Rechnergruppe */
use master
go
/* Erzeugen der Datenbank */

if exists (select * from master.dbo.sysdatabases
           where name = "SouthernOceanDB")
begin
/* drop database SouthernOceanDB */
print ' There exists already a SouthernOceanDB'
return
end

go
print 'Creating the "SouthernOceanDB" database'
go
create database SouthernOceanDB on Userdisk02=300, Userdisk03=100
go

sp_logdevice SouthernOceanDB, Userdisk03
go

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
create table Aari_Station
  (Aari_Station_Id#      int,
   Cruise_Number        int,
   Station_Number       int,
   Date_Time            datetime NULL,
   Longitude            float,
   Latitude             float,
   Bottom_Depth         int,
   Max_Obse_Depth       int,
   Number_Obse          int,
   Marsden_Square#      int,
   Validation_Flag      bit ,
   Update_Flag          bit )

create table Aari_Standard_Data
  (Aari_Standard_Data_Id# int,
   Aari_Station_Id#       int,
   Depth                  int,
   Temperature            float NULL,
   Salinity               float NULL,
   Oxygen                 float NULL,
   Validation_Flag        bit,
   Update_Flag            bit)

go

create table Aari_Station_Update_Backup
  (Aari_Station_Update_Backup_Id# int,
   Aari_Station_Id#          int,
   Cruise_Number             int,
   Station_Number            int,
   Date_Time                 datetime NULL,
   Longitude                  float,
   Latitude                   float,
```



```
Bottom_Depth          int,
Max Obse Depth        int,
Number Obse           int,
Marsden Square#      int,
Validation_Flag       bit,
Update_Flag           bit)
```

go

```
create table Aari_Update_Backup_Data
(Aari_Update_Backup_Data_Id# int,
Aari_Standard_Data_Id# int,
Aari_Station_Id# int,
Depth int,
Temperature float NULL,
Salinity float NULL,
Oxygen float NULL,
Validation_Flag bit,
Update_Flag bit)
```

go

```
create table Aari_Garbadge_Station
(Aari_Garbadge_Station_Id# int,
Cruise_Number int,
Station_Number int,
Date_Time datetime NULL,
Longitude float,
Latitude float,
Bottom_Depth int,
Max Obse Depth int,
Number Obse int,
Marsden_Square# int)
```

go

```
create table Aari_Garbadge_Data
(Aari_Garbadge_Data_Id# int,
Aari_Station_Id# int,
Depth int,
Temperature float NULL,
Salinity float NULL,
Oxygen float NULL)
```

go

/* Definition der Primaerschlüssel in den Tabellen */

```
execute sp_primarykey Aari_Station, Aari_Station_Id#
execute sp_primarykey Aari_Standard_Data, Aari_Standard_Data_Id#
execute sp_primarykey Aari_Station_Update_Backup,
Aari_Station_Update_Backup_Id#
execute sp_primarykey Aari_Update_Backup_Data, Aari_Update_Backup_Data_Id#
execute sp_primarykey Aari_Garbadge_Station, Aari_Garbadge_Station_Id#
execute sp_primarykey Aari_Garbadge_Data, Aari_Garbadge_Data_Id#
go
```

/* Definition der Sekundaerschlüssel in den Tabellen */

```
execute sp_foreignkey Aari_Standard_Data, Aari_Station, Aari_Station_Id#
execute sp_foreignkey Aari_Station_Update_Backup, Aari_Station,
Aari_Station_Id#
execute sp_foreignkey Aari_Update_Backup_Data, Aari_Standard_Data,
Aari_Standard_Data_Id#
execute sp_foreignkey Aari_Update_Backup_Data, Aari_Station, Aari_Station_Id#
execute sp_foreignkey Aari_Garbadge_Data, Aari_Garbadge_Station,
Aari_Station_Id#
go
```

```
/* Erzeugen der Views */
/* Erzeugen der Prozeduren */

/* Eintragen der Nutzer */
execute sp_addgroup AWIDB
execute sp_adduser WEISS,WEISS,AWIDB
execute sp_adduser WESTERWICK,WESTERWICK,AWIDB
execute sp_adduser REINKE,REINKE,AWIDB
go
execute sp_addgroup AWIEDV
execute sp_adduser RICHTER,RICHTER, AWIEDV
go
grant all on Aari_Station to AWIDB
grant all on Aari_Station_Update_Backup to AWIDB
grant all on Aari_Standard_Data to AWIDB
grant all on Aari_Update_Backup_Data to AWIDB
grant all on Aari_Garbadge_Data to AWIDB
grant all on Aari_Garbadge_Station to AWIDB
go
grant all on Aari_Station to AWIEDV
grant all on Aari_Station_Update_Backup to AWIEDV
grant all on Aari_Standard_Data to AWIEDV
grant all on Aari_Update_Backup_Data to AWIEDV
grant all on Aari_Garbadge_Data to AWIEDV
grant all on Aari_Garbadge_Station to AWIEDV
go

/* end of aari.script */
```



```
/*  
** FILENAME: GORDON.SCRIPT  
*/  
  
/* Sybase-Script made in June 1989 by M. Reinke */  
/* AWI-Rechnergruppe 2.Teil Gordon Daten*/  
use SouthernOceanDB  
go  
  
/* Erzeugen der Tabellen */  
if exists (select * from sysobjects where name = "Gordon_Station")  
begin  
drop table Gordon_Station  
end  
go  
  
create table Gordon_Station  
  (Gordon_Station_Id# int,  
   Station_Name int,  
   Date_Time datetime NULL,  
   Longitude float,  
   Latitude float,  
   Bottom_Depth int NULL,  
   Max_Obse_Depth int,  
   Number_Obse int,  
   Marsden_Square# int NULL,  
   Validation_Flag bit ,  
   Update_Flag bit )  
go  
  
if exists (select * from sysobjects where name = "Gordon_Standard_Data")  
begin  
drop table Gordon_Standard_Data  
end  
go  
  
create table Gordon_Standard_Data  
  (Gordon_Standard_Data_Id# int,  
   Gordon_Station_Id# int,  
   Depth int,  
   Temperature float NULL,  
   Salinity float NULL,  
   Oxygen float NULL,  
   Phosphate float NULL,  
   Silicate float NULL,  
   Nitrat float NULL,  
   Validation_Flag bit,  
   Update_Flag bit)  
go  
  
if exists (select * from sysobjects where name = "Gordon_Statistic")  
begin  
drop table Gordon_Statistic  
end  
go  
  
create table Gordon_Statistic  
  (Depth# int,  
   Marsden_Square# int,  
   Temperature_Mean float NULL,  
   Temperature_Var float NULL,  
   Temperature_Number_Observ float NULL,  
   Salinity_Mean float NULL,  
   Salinity_Var float NULL,
```

```
Salinity_Number_Observ float NULL,  
Oxygen_Mean float NULL,  
Oxygen_Var float NULL,  
Oxygen_Number_Observ float NULL,  
Validation_Method_Id# int NULL)
```

```
go
```

```
/* Definition der Primaerschlüssel in den Tabellen */
```

```
execute sp_primarykey Gordon_Station, Gordon_Station_Id#  
execute sp_primarykey Gordon_Standard_Data, Gordon_Standard_Data_Id#  
execute sp_primarykey Gordon_Statistic, Depth#, Marsden_Square#  
go
```

```
/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Gordon_Standard_Data, Gordon_Station,  
Gordon_Station_Id#
```

```
go
```

```
/* Eintragen der Nutzer */
```

```
grant all on Gordon_Station to AWIDB  
grant all on Gordon_Standard_Data to AWIDB  
grant all on Gordon_Statistic to AWIDB  
go  
grant all on Gordon_Station to AWIEDV  
grant all on Gordon_Standard_Data to AWIEDV  
grant all on Gordon_Statistic to AWIEDV
```

```
go
```

```
/* Erzeugen der Indices */
```

```
/* Erzeugen der Regeln */
```

```
/* Erzeugen der Views */
```

```
/* Erzeugen der Prozeduren */
```

```
create proc Gordon_Longitude_Change
```

```
as
```

```
update Gordon_Station  
set Longitude = Longitude - 360  
where Longitude > 180
```

```
go
```

```
create proc Gordon_Station_Marsden as
```

```
update Gordon_Station  
set G.Marsden_Square# = M.Marsden_Square# from  
Marsden_Square M, Gordon_Station G  
where G.Latitude > M.Latitude_Min and  
G.Latitude <= M.Latitude_Max and  
G.Longitude > M.Longitude_Min and  
G.Longitude <= M.Longitude_Max
```

```
go
```

```
/* end of gordon.script */
```



```
/*  
** FILENAME: AARI_STAT.SCRIPT  
*/  
  
/* Sybase-Script made in June 1989 by M. Reinke, Stefan Richter, */  
/* AWI-Rechnergruppe */  
  
/* Sybase-Script made in June 1989 by M. Reinke */  
/* AWI-Rechnergruppe 3.Teil Aari Statistic*/  
use SouthernOceanDB  
go  
  
/* Erzeugen der Tabellen */  
if exists (select * from sysobjects where name = "Aari_Statistic")  
begin  
drop table Aari_Statistic  
end  
go  
  
create table Aari_Statistic  
  (Depth# int,  
   Marsden_Square# int,  
   Temperature_SXX float NULL,  
   Temperature_SX float NULL,  
   Temperature_Number_Observ float NULL,  
   Salinity_SXX float NULL,  
   Salinity_SX float NULL,  
   Salinity_Number_Observ float NULL,  
   Oxygen_SXX float NULL,  
   Oxygen_SX float NULL,  
   Oxygen_Number_Observ float NULL,  
   Validation_Method_Id# int NULL)  
go  
if exists (select * from sysobjects where name = "Marsden_Square")  
begin  
drop table Marsden_Square  
end  
go  
  
create table Marsden_Square  
  (Marsden_Square# int,  
   Latitude_Min float NULL,  
   Latitude_Max float NULL,  
   Longitude_Min float NULL,  
   Longitude_Max float NULL)  
go  
  
/* Definition der Primaerschlüssel in den Tabellen */  
  
execute sp_primarykey Aari_Statistic, Depth#, Marsden_Square#  
execute sp_primarykey Marsden_Square, Marsden_Square#  
go  
  
/* Definition der Sekundaerschlüssel in den Tabellen */  
  
/* Erzeugen der Views */  
  
/* Sybase-Script made in June 1989 by M. Reinke */  
/* AWI-Rechnergruppe 4.Teil Aari Statistic*/  
  
if exists (select * from sysobjects where name = "Aari_Statistic_View")  
begin  
drop view Aari_Statistic_View  
end
```

go

```
create view Aari_Statistic_View
```

```
(Depth#  
Marsden_Square#  
Temperature_Mean  
Temperature_Var  
Temperature_Number_Observ  
Salinity_Mean  
Salinity_Var  
Salinity_Number_Observ  
Oxygen_Mean  
Oxygen_Var  
Oxygen_Number_Observ  
Validation_Method_Id# ) as
```

```
select Depth#, Marsden_Square#,  
Temperature_SX/Temperature_Number_Observ,  
(Temperature_SXX-(Temperature_SX*Temperature_SX/Temperature_Number_Observ))/  
(Temperature_Number_Observ-1),  
Temperature_Number_Observ,  
Salinity_SX/Salinity_Number_Observ,  
(Salinity_SXX-(Salinity_SX*Salinity_SX/Salinity_Number_Observ))/  
(Salinity_Number_Observ-1),  
Salinity_Number_Observ,  
Oxygen_SX/Oxygen_Number_Observ,  
(Oxygen_SXX-(Oxygen_SX*Oxygen_SX/Oxygen_Number_Observ))/  
(Oxygen_Number_Observ-1),  
Oxygen_Number_Observ,  
Validation_Method_Id#  
from Aari_Statistic  
go
```

```
/* Erzeugen der Prozeduren */
```

```
if exists (select * from sysobjects where name = "fill_aari_statistic")  
begin  
drop proc fill_aari_statistic  
end  
go
```

```
create proc fill_aari_statistic as  
insert Aari_Statistic(Depth#,Marsden_Square#,  
Temperature_SXX,Temperature_SX,  
Temperature_Number_Observ,  
Salinity_SXX,Salinity_SX,  
Salinity_Number_Observ,  
Oxygen_SXX,Oxygen_SX,  
Oxygen_Number_Observ)  
select  
Depth,Marsden_Square#,  
sum(Temperature*Temperature),  
sum(Temperature),count(Temperature),  
sum(Salinity*Salinity),  
sum(Salinity),count(Salinity),  
sum(Oxygen*Oxygen),  
sum(Oxygen),count(Oxygen)  
from Aari_Station,Aari_Standard_Data  
where Aari_Station.Aari_Station_Id# = Aari_Standard_Data.Aari_Station_Id#  
group by Depth,Marsden_Square#  
go
```

```
if exists (select * from sysobjects where name = "aari_tso")  
begin  
drop proc aari_tso  
end
```



```

go

create proc aari_tso
as
select Depth#,Marsden_Square#,
Temperature_Mean,Temperature_Var, Temperature_Number_Observ,
Salinity_Mean,Salinity_Var,
Oxygen_Mean,Oxygen_Var
from Aari_Statistic_View
where Temperature_Number_Observ >1
and Depth# = 0
order by Depth#
go

if exists (select * from sysobjects where name = "aari_tsol")
begin
drop proc aari_tsol
end
go

create procedure aari_tsol
( @Marsden_Square int) as
select * from Marsden_Square
where
Marsden_Square#=@Marsden_Square
go

if exists (select * from sysobjects where name = "aari_tso2")
begin
drop proc aari_tso2
end
go

create procedure aari_tso2
( @Marsden_Square int,
  @Depth int)
as
select Temperature, Salinity,Oxygen
from Aari_Station,Aari_Standard_Data
where
Marsden_Square#=@Marsden_Square and
Depth=@Depth and
Aari_Station.Aari_Station_Id#=Aari_Standard_Data.Aari_Station_Id#
and Temperature is not NULL
go

/* Eintragen der Nutzer */
grant all on Aari_Statistic to AWIDB
go
grant all on Aari_Statistic to AWIEDV
go
grant all on Aari_Statistic_View to AWIDB
go
grant all on Aari_Statistic_View to AWIEDV
go
grant all on fill_aari_statistic to AWIDB
go
grant execute on fill_aari_statistic to AWIEDV
go
grant all on aari_tso to AWIDB
go
grant execute on aari_tso to AWIEDV
go
grant all on aari_tsol to AWIDB

```

```
go
grant execute on aari_tso1 to AWIEDV
go
grant all on aari_tso2 to AWIDB
go
grant execute on aari_tso2 to AWIEDV
go
/* Erzeugen der Regeln */

/* end of aari_stat.script */
```

```
/*  
** FILENAME: GORDON_COPY.SCRIPT  
*/  
  
create procedure Gordon_copy  
(@id# int)  
as  
/* Fill Station and Standard_Data with Gordon Data*/  
/* M. Reinke 28.6.90 */  
begin  
  
insert Station (Station_Id#, Station Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max Obse Depth,  
Number Obse, Marsden Square#, Validation_Flag, Update_Flag)  
select Gordon_Station_Id#, Station Name,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max Obse Depth,  
Number Obse, Marsden Square#, Validation_Flag, Update_Flag  
from Gordon_Station  
where Gordon_Station_Id#=@id#  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
Depth, Temperature, Salinity, Oxygen, Phosphate, Silicate, Nitrat,  
Validation_Flag, Update_Flag)  
select Gordon_Standard_Data_Id#, Gordon_Station_Id#,  
Depth, Temperature, Salinity, Oxygen, Phosphate, Silicate, Nitrat,  
Validation_Flag, Update_Flag  
from Gordon_Standard_Data  
where Gordon_Station_Id#=@id#  
  
end
```



```
/*  
** FILENAME: GORDON_INTERPOLATED_DATA.SCRIPT  
*/
```

```
/*-----Script for the interpolated Gordon Standard Data ---  
**----- M Reinke 10. Sep 1990 -----  
*/
```

```
use SouthernOceanDB  
go
```

```
create table Gordon_Interpolated_Data  
(Gordon_Interpolated_Data_Id# int,  
Gordon_Station_Id# int,  
Depth int,  
Temperature float NULL,  
Salinity float NULL,  
Oxygen float NULL,  
Validation_Flag int,  
Update_Flag int)
```

```
go
```

```
sp_primarykey Gordon_Interpolated_Data, Gordon_Interpolated_Data_Id#  
go  
sp_foreignkey Gordon_Interpolated_Data, Gordon_Station, Gordon_Station_Id#  
go
```

```
/* Sybase-Script made in Oktober 1990 Lutz-Peter Kurdelski */
/* AWI-Rechnergruppe */
/*
** FILE: GORDON_CRUISE.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go

use SouthernOceanDB
go

/* Die Transaction wird gestartet, um ein Rollback zu ermöglichen. */
/* Erzeugen der Tabellen */

create table Gordon_Cruise
  (Gordon_Cruise_Id#      int,          /* internal identification */
   Ship_Code              varchar(10), /* Guretzky's own number */
   Gordon_Cruise_Name    varchar(10) NULL, /* The research vessel */
   Comment                varchar(255) NULL /* The cruise name */
                                /* A comment or name */
  )
go

create table Gordon_Ship
  (Ship_Id#              int,          /* internal identification */
   Ship_Code             varchar (10), /* identifikation of the vessel */
   Ship_Name             varchar (80) NULL, /* name of the vessel */
   Country               varchar (80) NULL /* country of the vessel */
  )
go

/* Hinzufuegen der Spalte Gordon_Cruise_Number zur Tabelle */
/* Gordon_Station */

alter table Gordon_Station
add Gordon_Cruise_Id# int NULL,
   Ship_Code varchar(10) NULL
go

/* Error log table */

create table errorlog
  (Ship_Code varchar(10),
   Headers fil Number int,
   Longitude float,
   Latitude float,
   Date datetime,
   Gordon_Station_Id# int
  )
go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey Gordon_Cruise, Gordon_Cruise_Id#
```

```
execute sp_primarykey Gordon_Ship, Ship_Code#  
go
```

```
/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Gordon_Cruise, Gordon_Ship, Ship_Code#  
execute sp_foreignkey Gordon_Station, Gordon_Cruise, Gordon_Cruise_Id#  
go
```

```
print 'script successfully completed'  
go
```

```
/* eof gordon_cruise.script */
```



```
/*  
** FILE: TRIGGER_DROP.SCRIPT  
*/  
use SouthernOceanDB  
go  
/* Drop triggers for table Station and Standard_Data */  
/* M. Reinke 16.10.90 */  
  
drop trigger DeleteStationTgr  
go  
  
drop trigger DeleteStandard_DataTgr  
go  
  
drop trigger InsertStationTgr  
go  
  
drop trigger InsertStandard_DataTgr  
go  
  
drop trigger UpdateStationTgr  
go  
  
drop trigger UpdateStandard_DataTgr  
go
```

```
/*  
** FILENAME: GORDON_STAT.SCRIPT  
*/  
  
/* Sybase-Script made in November by M. Reinke  
**   AWI-Rechnergruppe  
**  
**   AWI-Rechnergruppe 3.Teil Gordon Statistic  
*/  
use SouthernOceanDB  
go  
  
/* Erzeugen der Tabellen */  
if exists (select * from sysobjects where name = "Gordon_Statistic")  
begin  
drop table Gordon_Statistic  
end  
go  
  
create table Gordon_Statistic  
  (Depth#          int,  
   Marsden_Square# int,  
   Temperature_SXX float NULL,  
   Temperature_SX  float NULL,  
   Temperature_Number_Observ float NULL,  
   Salinity_SXX    float NULL,  
   Salinity_SX     float NULL,  
   Salinity_Number_Observ float NULL,  
   Oxygen_SXX      float NULL,  
   Oxygen_SX       float NULL,  
   Oxygen_Number_Observ float NULL,  
   Validation_Method_Id# int NULL)  
go  
  
/* Definition der Primaerschlüssel in den Tabellen */  
execute sp_primarykey Gordon_Statistic, Depth#, Marsden_Square#  
go  
  
/* Definition der Sekundaerschlüssel in den Tabellen */  
execute sp_foreignkey Gordon_Statistic, Marsden_Square, Marsden_Square#  
go  
/* Erzeugen der Views */  
  
if exists (select * from sysobjects where name = "Gordon_Statistic_View")  
begin  
drop view Gordon_Statistic_View  
end  
go  
  
create view Gordon_Statistic_View  
  (Depth#          ,  
   Marsden_Square# ,  
   Temperature_Mean ,  
   Temperature_Var ,  
   Temperature_Number_Observ ,  
   Salinity_Mean   ,  
   Salinity_Var,   ,  
   Salinity_Number_Observ ,  
   Oxygen_Mean,   ,  
   Oxygen_Var,    ,  
   Oxygen_Number_Observ,  
   Validation_Method_Id# ) as  
select Depth#, Marsden_Square#,  
Temperature_SX/Temperature_Number_Observ,
```

```

(Temperature_SXX-(Temperature_SX*Temperature_SX/Temperature_Number_Observ))/
(Temperature_Number_Observ-1),
convert(int, Temperature_Number_Observ),
Salinity_SX/Salinity_Number_Observ,
(Salinity_SXX-(Salinity_SX*Salinity_SX/Salinity_Number_Observ))/
(Salinity_Number_Observ-1),
convert(int, Salinity_Number_Observ),
Oxygen_SX/Oxygen_Number_Observ,
(Oxygen_SXX-(Oxygen_SX*Oxygen_SX/Oxygen_Number_Observ))/
(Oxygen_Number_Observ-1),
convert(int, Oxygen_Number_Observ),
Validation_Method_Id#
from Gordon_Statistic
go

```

```

/* Erzeugen der Prozeduren */

```

```

if exists (select * from sysobjects where name = "fill_Gordon_statistic")
begin
drop proc fill_Gordon_statistic
end
go

```

```

create proc fill_Gordon_statistic as
insert Gordon_Statistic(Depth#, Marsden_Square#,
Temperature_SXX, Temperature_SX,
Temperature_Number_Observ,
Salinity_SXX, Salinity_SX,
Salinity_Number_Observ,
Oxygen_SXX, Oxygen_SX,
Oxygen_Number_Observ)
select
Depth, Marsden_Square#,
sum(Temperature*Temperature),
sum(Temperature), count(Temperature),
sum(Salinity*Salinity),
sum(Salinity), count(Salinity),
sum(Oxygen*Oxygen),
sum(Oxygen), count(Oxygen)
from Gordon_Station, Gordon_Interpolated_Data
where Gordon_Station.Gordon_Station_Id# = Gordon_Interpolated_Data.Gordon_Station_Id#
and Marsden_Square# is not NULL
group by Depth, Marsden_Square#
go

```

```

/* Eintragen der Nutzer */
grant select on Gordon_Statistic_View to PHYSIK1
grant select on Gordon_Statistic to PHYSIK1
go

```

```

/* Erzeugen der Regeln */

```

```

/* end of gordon_stat.script */

```



```
/* Script um missing Values auf NULL zu setzen */
/* M. Reinke 7.9.89 */
/* L.P. Kurdelski 1990-11-05 */
/*
** FILE: NOWLIN_UPD.SCRIPT
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Nowlin_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Nowlin_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Nowlin_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Nowlin_Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
go
dump tran SouthernOceanDB with truncate_only
go
update Nowlin_Station
set Marsden_Square# = NULL
where Marsden_Square# = 999
go
dump tran SouthernOceanDB with truncate_only
go
/* eof nowlin_upd.script */
```



```

/* Sybase-Script made in June 1989 by M. Reinke, Stefan Richter, */
/* Lutz-Peter Kurdelski */
/* AWI-Rechnergruppe */
/*
** FILE: NOWLIN.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Nowlin_Station")
drop table Nowlin_Station
go

create table Nowlin_Station
  (Nowlin_Station_Id# int,      /* internal identification */
   Cruise_Number      int,      /* Nowlin nc */
   Station_Number     int,      /* Nowlin ns */
   Longitude          float,    /* Nowlin ongitude */
   Latitude           float,    /* Nowlin atitude */
   Date_Time          datetime NULL,
   Bottom_Depth       int,      /* Nowlin nyear nmonth nda nho */
   Max_Obse_Depth     int,      /* Nowlin nde */
   Number_Obse        int,      /* Nowlin mode */
   Marsden_Square#    int NULL /* Nowlin hz */
                                /* Nowlin msq */)

go

if exists (select * from sysobjects where name="Nowlin_Standard_Data")
drop table Nowlin_Standard_Data
go

create table Nowlin_Standard_Data
  (Nowlin_Standard_Data_Id# int, /* internal identification */
   Nowlin_Station_Id#       int, /* internal identification */
   Depth                    int,  /* Nowlin zst */
   Temperature              float NULL, /* Nowlin tst */
   Salinity                 float NULL, /* Nowlin sst */
   Oxygen                   float NULL /* Nowlin oxst */)

go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey Nowlin_Station, Nowlin_Station_Id#
execute sp_primarykey Nowlin_Standard_Data, Nowlin_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */

execute sp_foreignkey Nowlin_Standard_Data, Nowlin_Station,
                    Nowlin_Station_Id#
go

```

```
/* Erzeugen der Views */
/* Erzeugen der Prozeduren */

/* Eintragen der Nutzer */
grant select on Nowlin_Station to PHYSIK1
grant select on Nowlin_Standard_Data to PHYSIK1
go

/* eof nowlin.script */
```



```
/* created by L.-P. Kurdelski, M. Reinke */  
/* AWI Database Group */
```

```
/*  
** FILENAME: GC.SCRIPT  
**  
** DATE: 11-19-1990  
*/
```

```
/* GarbageCollector to remove garbage from *Standard Data */  
/* if Stations were removed and the trigger was not installed. */  
/* I.e. Station were deleted without notifying StandardData. */
```

```
use master  
go
```

```
if not exists ( select * from master.dbo.sysdatabases  
                where name = "SouthernOceanDB" )
```

```
begin  
    print 'SouthernOceanDB does not exist'  
    return  
end  
go
```

```
use SouthernOceanDB  
go
```

```
if exists (select *  
           from sysobjects  
           where name = "garbageCollectStandardData" )
```

```
begin  
    drop procedure garbageCollectStandardData  
end  
go
```

```
create procedure garbageCollectStandardData  
as  
begin  
begin transaction
```

```
select Aari_Station_Id#  
    into gc  
    from Aari_Station  
    where Aari_Station_Id# not in  
        ( select Station.Station_Id# from Station  
          where Station.Station_Id# < 100000 )
```

```
delete Standard_Data  
from Standard_Data, gc  
where Standard_Data.Station_Id# = gc.Aari_Station_Id#  
and not exists ( select Station.Station_Id# from Station  
                 where Station.Station_Id# = Standard_Data.Station_Id# )
```

```
insert DeleteStation  
select Aari_Station_Id#, getdate(), user_id()  
from gc  
where Aari_Station_Id# not in ( select DeleteStation.Station_Id#  
                               from DeleteStation  
                               where DeleteStation.Station_Id# = gc.Aari_Station_Id# )
```

```
if @@error = 0  
    commit transaction  
else  
    rollback transaction  
end  
go
```

/* end of gc.script */

Script - 12

```
/*  
** SYBASE script  
**  
** FILENAME: CHECKGCSD.SCRIPT  
**  
** Creation: 1990-11-19  
** Creator: L.-P. Kurdelski  
*/  
create procedure checkGarbageStandardData  
as  
  
select Station_Id#  
from Standard_Data  
where Station_Id# not in  
    (select Station.Station_Id# from Station)
```

neueste Version
am 12.9.91
abgespeichert
(unter diesem
Namen)

vgl. Script-83

```
/*  
**  
** FILENAME: GARBAGECOLLECT.SCRIPT  
**  
** AUTHOR: L.-P. Kurdelski  
**  
** DATE: 11-19-1990  
**  
*/
```

```
use SouthernOceanDB  
go
```

```
if @@error != 0  
begin  
    print 'Fehler beim Oeffnen der Datenbank.'  
    return  
end
```

```
go
```

```
execute checkGarbageStandardData  
go
```



```
/* Sybase-Script made in November 1990 by Lutz-Peter Kurdelski */
/* AWI-Rechnergruppe */
/*
** FILE: GSBACKFILL.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
```

```
use master
go
```

```
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
```

```
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
```

```
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go
```

```
use SouthernOceanDB
go
```

```
/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Gordon_Station_Backfill")
drop table Gordon_Station_Backfill
go
```

```
create table Gordon_Station_Backfill
(Gordon_Station_Id# int, /* internal identification */
Ship_Code varchar (10),
Longitude float, /* Nowlin longitude */
Latitude float, /* Nowlin attitude */
Date_Time datetime NULL,
/* Nowlin nyear nmonth nda nho */
Day_Of_Year int /* Tagesnummer */)
go
```

```
/* Definition der Primaerschlüssel in den Tabellen */
```

```
execute sp_primarykey Gordon_Station_Backfill, Gordon_Station_Id#
go
```

```
/* eof gsbackfill.script */
```


Script-1.5

```
use SouthernOceanDB
go
/* Insert triggers for table Station and Standard_Data */
/* M. Reinke 10.5.90 */
/*
** FILE: TRIGGER.SCRIPT
*/

/* Changes
**      1990-11-28      InsertStationTrg      L.-P. Kurdelski
**      1990-11-28      InsertStandardDataTrg  L.-P. Kurdelski
*/

create trigger DeleteStationTgr
on Station for delete
as
begin
delete Standard_Data
from Standard_Data,deleted
where deleted.Station_Id#=Standard_Data.Station_Id#
insert into DeleteStation
select deleted.Station_Id#, getdate(), suser_id()
from deleted
print "Records in tables Station and Standard_Data deleted!"
end

go

create trigger DeleteStandard_DataTgr
on Standard_Data for delete
as
begin
delete Standard_Data
from Standard_Data,deleted
where deleted.Station_Id#=Standard_Data.Station_Id#
insert into DeleteData
select deleted.Standard_Data_Id#, getdate(),suser_id()
from deleted
end

go

create trigger InsertStationTgr
on Station for insert
as
begin
insert into InsertStation
select inserted.Station_Id#, getdate(), suser_id()
from inserted
end

go

create trigger InsertStandard_DataTgr
on Standard_Data for insert
as
begin
insert into InsertData
select inserted.Standard_Data_Id#, getdate(),suser_id()
from inserted
end

go

create trigger UpdateStationTgr
on Station for update
```

```

as
begin
if update (Station_Id#)
begin
print " Station_Id is not allowed to be updated!! Update aborted!"
rollback transaction
return
end
if (select count(*) from deleted)=0
begin
print " No rows found for update! Update aborted!"
rollback transaction
return
end
if not update(Validation_Flag)
begin
print "Validation Flag has also to be updated for documentation purposes!! Up
rollback transaction
return
end
if not exists (select * from Validation, inserted
                where Validation_Id#=Validation_Flag)
begin
print "Validation method not found in table Validation!! Update aborted"
rollback transaction
return
end
insert into UpdateStation
select inserted.Station_Id#, inserted.Validation_Flag, getdate(), suser_id()
from inserted
end

go

create trigger UpdateStandard_DataTgr
on Standard_Data for update
as
begin
if update (Standard_Data_Id#)
begin
print " Standard_Data_Id# is not allowed to be updated!! Update aborted"
rollback transaction
return
end
if (select count(*) from deleted)=0
begin
print "No rows selected !! Update aborted"
rollback transaction
return
end
if not exists (select * from Station s, inserted
                where s.Station_Id#=inserted.Station_Id#)
begin
print " No Station with the new Station_Id# exists! Update aborted!"
rollback transaction
return
end
if not update(Validation_Flag)
begin
print "Validation_Flag has also to be updated for documentation purposes!! Up
rollback transaction
return
end
if not exists (select * from Validation v,inserted
                where v.Validation_Id#=inserted.Validation_Flag)
begin

```

```
print " Validation method not found in table Validation! Update aborted!"
rollback transaction
return
end
insert into UpdateData
select deleted.Standard_Data_Id#,
        deleted.Station_Id#,
        inserted.Station_Id#,
        inserted.Validation_Flag,
        getdate(),
        suser_id()
from inserted, deleted
end
go
/* eof trigger.script */
```



```
/*
** Sybase-Script made in November 1990 by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
** FILE AARIINDEX.SCRIPT
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
/* Definition der Primaerschlüssel in den Tabellen */
/* Definition der Sekundaerschlüssel in den Tabellen */
/* Definition der Indizes */

create unique index Aari_Station_Index
on Aari_Station (Aari_Station_Id#)
go

create unique index Aari_Standard_Data_Index
on Aari_Standard_Data (Aari_Standard_Data_Id#)
go

/* Erzeugen der Views */
/* Erzeugen der Prozeduren */

/* Eintragen der Nutzer */
```

```

/* Sybase-Script made in June 1989 by M. Reinke, Stefan Richter, */
/* Lutz-Peter Kurdelski */
/* AWI-Rechnergruppe */
/*
** FILE: KUROPATKIN.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Kuropatkin_Station")
drop table Kuropatkin_Station
go

create table Kuropatkin_Station
(Kuropatkin_Station_Id# int, /* internal identification */
Cruise_Number int, /* Kuropatkin nc */
Station_Number int, /* Kuropatkin ns */
Longitude float, /* Kuropatkin ongitude */
Latitude float, /* Kuropatkin atitude */
Date_Time datetime NULL,
Bottom_Depth int, /* Kuropatkin nyear nmonth nda nho */
Max_Obse_Depth int, /* Kuropatkin nde */
Number_Obse int, /* Kuropatkin mode */
Marsden_Square# int NULL /* Kuropatkin hz */
/* Kuropatkin msq */)
go

if exists (select * from sysobjects where name="Kuropatkin_Standard_Data")
drop table Kuropatkin_Standard_Data
go

create table Kuropatkin_Standard_Data
(Kuropatkin_Standard_Data_Id# int, /* internal identification */
Kuropatkin_Station_Id# int, /* internal identification */
Depth int, /* Kuropatkin zst */
Temperature float NULL, /* Kuropatkin tst */
Salinity float NULL, /* Kuropatkin sst */
Oxygen float NULL /* Kuropatkin oxst */)
go

/* Definition der Primaerschlüssel in den Tabellen */
execute sp_primarykey Kuropatkin_Station, Kuropatkin_Station_Id#
execute sp_primarykey Kuropatkin_Standard_Data, Kuropatkin_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */
execute sp_foreignkey Kuropatkin_Standard_Data, Kuropatkin_Station,
Kuropatkin_Station_Id#
go

```

```
/* Erzeugen der Views */  
/* Erzeugen der Prozeduren */  
/* Eintragen der Nutzer */  
/* eof kuropatkin.script */
```



```
/*  
** FILENAME: KUROPATKIN_UPD.SCRIPT  
*/  
  
/* Script um missing Values auf NULL zu setzen */  
/* M. Reinke 7.9.89 */  
/* L.P. Kurdelski 1990-11-28 */  
use SouthernOceanDB  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Aari_Station  
set Date_Time = NULL  
where Date_Time="Jan 01 1900 00:00:00.0"  
and Aari_Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Aari_Standard_Data  
set Temperature = NULL  
where Temperature < -10.0  
and Aari_Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Aari_Standard_Data  
set Salinity = NULL  
where Salinity < 0  
and Aari_Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Aari_Standard_Data  
set Oxygen = NULL  
where Oxygen <= -9.0  
and Aari_Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Aari_Station  
set Marsden_Square# = NULL  
where Marsden_Square# = 999  
and Aari_Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
/*  
** update Station  
** Standard_Data  
*/  
go  
update Station  
set Date_Time = NULL  
where Date_Time="Jan 01 1900 00:00:00.0"  
and Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Standard_Data  
set Temperature = NULL  
where Temperature < -10.0  
and Station_Id# between 35963 and 36059  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Standard_Data  
set Salinity = NULL
```

```
where Salinity < 0
  and Station_Id# between 35963 and 36059
go
dump tran SouthernOceanDB with truncate_only
go
update Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
  and Station_Id# between 35963 and 36059
go
dump tran SouthernOceanDB with truncate_only
go
update Station
set Marsden_Square# = NULL
where Marsden_Square# = 999
  and Station_Id# between 35963 and 36059
go
dump tran SouthernOceanDB with truncate_only
go
/* end of kuropatkin_upd.script */
```



```

/* AWI-Rechnergruppe */
/* SYBASE Script created by L.-P. Kurdelski
** on 1991-04-19
**
** FILE: HAINESLOAD.SCRIPT
**
*/

```

```

/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */

```

```
use master
```

```
go
```

```
/* Erzeugen der Datenbank */
```

```
/* Die Datenbank muss nicht mehr erzeugt werden. */
```

```
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
```

```
begin
```

```
print 'SouthernOceanDB does not exist !!!'
```

```
return
```

```
end
```

```
use SouthernOceanDB
```

```
go
```

```
/* Erzeugen der Tabellen */
```

```
if exists (select * from sysobjects where name = "Haines_Station")
```

```
drop table Haines_Station
```

```
go
```

```
create table Haines_Station
```

```

(Haines_Station_Id# int, /* internal identification */
Cruise_Number int, /* Haines nc */
Station_Number int, /* Haines ns */
Longitude float, /* Haines ongitude */
Latitude float, /* Haines attitude */
Date_Time datetime NULL,
Bottom_Depth int NULL, /* Haines nyear nmonth nda nho */
Max_Obse_Depth int NULL, /* Haines nde */
Number_Obse int, /* Haines mode */
Marsden_Square# int NULL /* Haines hz */
/* Haines msq */)

```

```
go
```

```
if exists (select * from sysobjects where name = "Haines_Standard_Data")
```

```
drop table Haines_Standard_Data
```

```
go
```

```
create table Haines_Standard_Data
```

```

(Haines_Standard_Data_Id# int, /* internal identification */
Haines_Station_Id# int, /* internal identification */
Depth int, /* Haines zst */
Temperature float NULL, /* Haines tst */
Salinity float NULL, /* Haines sst */
Oxygen float NULL /* Haines oxst */)

```

```
go
```

```
/* Definition der Primaerschlüssel in den Tabellen */
```

```
execute sp_primarykey Haines_Station, Haines_Station_Id#
```

```
execute sp_primarykey Haines_Standard_Data, Haines_Standard_Data_Id#
```

```
go
```

```
/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Haines_Standard_Data, Haines_Station,
```


Haines_Station_Id#

go

/* eof hainesload.script */

```
/*
** SYBASE Script
**
** Created 1990-11-28
**
** Creator: L.P. Kurdelski
**
** FILE: HAINESLOAD_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** Haines_Station
** Haines_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
go
dump tran SouthernOceanDB with truncate_only
go
update Haines_Station
set Marsden_Square# = NULL
where Marsden_Square# = 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```

/* AWI-Rechnergruppe */
/*
** FILE: GONELLA.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Gonella_Station")
drop table Gonella_Station
go

create table Gonella_Station
(Gonella_Station_Id# int, /* internal identification */
Cruise_Number int, /* Gonella nc */
Station_Number int, /* Gonella ns */
Longitude float, /* Gonella ongitude */
Latitude float, /* Gonella atitude */
Date_Time datetime NULL,
/* Gonella nyear nmonth nda nho */
Bottom_Depth int NULL, /* Gonella nde */
Max_Obse_Depth int NULL, /* Gonella mode */
Number_Obse int, /* Gonella hz */
Marsden_Square# int NULL /* Gonella msq */)

go

if exists (select * from sysobjects where name="Gonella_Standard_Data")
drop table Gonella_Standard_Data
go

create table Gonella_Standard_Data
(Gonella_Standard_Data_Id# int, /* internal identification */
Gonella_Station_Id# int, /* internal identification */
Depth int, /* Gonella zst */
Temperature float NULL, /* Gonella tst */
Salinity float NULL, /* Gonella sst */
Oxygen float NULL /* Gonella oxst */)

go

/* Definition der Primaerschlüssel in den Tabellen */
execute sp_primarykey Gonella_Station, Gonella_Station_Id#
execute sp_primarykey Gonella_Standard_Data, Gonella_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */
execute sp_foreignkey Gonella_Standard_Data, Gonella_Station,
Gonella_Station_Id#
go

/* eof gonella.script */

```



```
/*
** SYBASE script
**
** Created 1990-11-28
**
** Creator: L.P. Kurdelski
**
** FILE: GONELLA_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** Gonella_Station
** Gonella_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Station
set Bottom_Depth = NULL
where Bottom_Depth < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Standard_Data
set Oxygen = NULL
where Oxygen < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Gonella_Station
set Marsden_Square# = NULL
where Marsden_Square# = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
/*  
** SYBASE script  
**  
** Created 1990-12-15  
**  
** Creator: L.-P. Kurdelski  
**  
** FILE: GONELLA_COPY.SCRIPT  
*/  
use SouthernOceanDB  
go  
  
if @@error != 0  
begin  
    print 'Fehler beim Oeffnen der Datenbank.'  
    return  
end  
go  
  
execute Gonella_copy_all  
go
```

```
/*  
** SYBASE script  
**  
** Created 1991-07-10  
**  
** Creator: L.-P. Kurdelski  
**  
** FILE: GONELLA_COPY_ALL.SCRIPT  
*/  
create procedure Gonella_copy_all  
as  
/* Fill Station and Standard_Data with Gonella Data*/  
/* i.e. Marion Dufresne */  
/* L.-P. Kurdelski 1990-12-18 */  
begin  
  
insert Station (Station_Id#, Cruise_Number, Station_Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#)  
select Gonella_Station_Id#, Cruise_Number, Station_Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#  
from Gonella_Station  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
Depth, Temperature, Salinity, Oxygen)  
select Gonella_Standard_Data_Id#, Gonella_Station_Id#,  
Depth, Temperature, Salinity, Oxygen  
from Gonella_Standard_Data  
  
end
```


Script-25

```
/*  
** SYBASE Script  
**  
** Created 1990-12-11  
**  
** Creator: L.-P. Kurdelski  
**  
** FILE: HAINESLOAD_COPY.SCRIPT  
*/  
use SouthernOceanDB  
go  
  
if @@error != 0  
begin  
    print 'Fehler beim Oeffnen der Datenbank.'  
    return  
end  
go  
  
execute Haines_copy_all  
go
```

```
/*
** Sybase-Script made in December 1990 by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
** Modified: 1991-09-04 by lpk   update_ship trigger geandert: Beim
**                               update wird das Schiff in jedem Fall
**                               in die Tabelle Ship_Updated eingetragen.
**
** FILE: cruise.script
**
** Die Tabellen Ship und Cruise werden erzeugt.
**
** Es werden Trigger auf delete, insert, update gelegt,
**   so dass nachvollziehbar ist, wer was wann geendert hat.
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank
**   Die Datenbank muss nicht mehr erzeugt werden.
*/
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Cruise")
  drop table Cruise
go

create table Cruise
  (Cruise_Number#      int,      /* internal identification */
   /* Cruise number */
   Ship_Id#            int,      /* Ship_Id from Ship */
  )

go

if exists (select * from sysobjects where name="Cruise_Deleted")
  drop table Cruise_Deleted
go

create table Cruise_Deleted
  (Cruise_Number#      int,      /* internal identification */
   /* Cruise number */
   Ship_Id#            int,      /* Ship_Id from Ship */
   user_id#            int,      /* who deleted data */
   date                datetime /* date of deletion */
  )

go

if exists (select * from sysobjects where name="Cruise_Updated")
  drop table Cruise_Updated
go

create table Cruise_Updated
  (Cruise_Number#      int,      /* internal identification */
   /* Cruise number */
   Ship_Id#            int,      /* Ship_Id from Ship */
```



```

Old_Cruise_Number# int,      /* internal identification */
                        /* Cruise number */
Old_Ship_Id#       int,      /* Ship_Id from Ship */
user_id#          int,      /* who updated data */
date              datetime /* date of update */
)

```

go

```

if exists (select * from sysobjects where name="Ship")
drop table Ship

```

go

```

create table Ship
(Ship_Id#           int,      /* internal identification */
 Ship_Name         varchar(80), /* the name of the ship */
 Country          varchar(80), /* the country */
 Ship_Code        varchar(8) null /* the ship code */
)

```

go

```

if exists (select * from sysobjects where name="Ship_Deleted")
drop table Ship_Deleted

```

go

```

create table Ship_Deleted
(Ship_Id#           int,      /* internal identification */
 Ship_Name         varchar(80), /* the name of the ship */
 Country          varchar(80), /* the country */
 Ship_Code        varchar(8) null, /* the ship code */
 user_id#         int,
 date             datetime
)

```

go

```

if exists (select * from sysobjects where name="Ship_Updated")
drop table Ship_Updated

```

go

```

create table Ship_Updated
(Ship_Id#           int,      /* internal identification */
 Ship_Name         varchar(80), /* the name of the ship */
 Country          varchar(80), /* the country */
 Ship_Code        varchar(8) null, /* the ship code */
 Old_Ship_Name     varchar(80), /* the name of the ship */
 Old_Country      varchar(80), /* the country */
 Old_Ship_Code     varchar(8) null, /* the ship code */
 user_id#         int,
 date             datetime
)

```

go

/* Definition der Primaerschlüssel in den Tabellen */

```

execute sp_primarykey Cruise, Cruise_Number#
execute sp_primarykey Ship, Ship_Id#

```

go

/* Definition der Sekundaerschlüssel in den Tabellen */

```

execute sp_foreignkey Cruise, Ship, Ship_Id#

```

go

/* Generieren der Indizes */

```

create unique index Cruise_Index
on Cruise (Cruise_Number#)

```



```
go
create clustered index Cruise_Ship_Index
on Cruise (Ship_Id#)
go
```

```
create unique index Ship_Index
on Ship (Ship_Id#)
go
```

```
/* Erzeugen der Views */
/* Erzeugen der Prozeduren */
/* Erzeugen von Trigger */
```

```
create trigger Insert_Cruise
on Cruise
for insert
as
begin
    declare @msg varchar(80),
            @sid int

    if not exists (select *
                  from Ship, inserted
                  where Ship.Ship_Id# = inserted.Ship_Id#)
    begin
        select @sid = Ship_Id# from inserted
        select @msg = "INSERT Cruise: Please insert the ship #" +
                    convert(varchar(20),@sid) + " first into Ship"
        print @msg
        rollback transaction
    end
end
go
```

```
create trigger Delete_Cruise
on Cruise
for delete
as
begin
    if exists (select *
              from Station, deleted
              where Station.Cruise_Number = deleted.Cruise_Number#)
    begin
        print "DELETE Cruise: There exists a station corresponding to this C
        rollback transaction
    end
    else
    begin
        insert into Cruise_Deleted
        select deleted.Cruise_Number#,
              deleted.Ship_Id#,
              suser_id(),
              getdate()
        from deleted
    end
end
go
```

```
create trigger Update_Cruise
on Cruise
for update
as
begin
    declare @msg varchar(80),
            @sid int
```

```

if not exists
  (select *
   from Ship, inserted
   where Ship.Ship_Id# = inserted.Ship_Id#)
begin
  select @sid = Ship_Id# from inserted
  select @msg = "UPDATE Cruise: Please insert the ship #" +
               convert(varchar(20),@sid) + " first into Ship"

  print @msg
  rollback transaction
end
else
  begin
    if update(Cruise_Number#)
      begin
        if exists
          (select *
           from Station, deleted
           where Station.Cruise_Number = deleted.Cruise_Number#)
        begin
          select @sid = Cruise_Number from deleted
          select @msg = "UPDATE Cruise: There exists at least one station"
                    + "(" + convert(varchar(20),@sid) + ")" +
                    + " corresponding to this Cruise_Number"

          print @msg
          rollback transaction
        end
      end
    else
      begin
        insert into Cruise_Updated
        select inserted.Cruise_Number#,
               inserted.Ship_Id#,
               deleted.Cruise_Number#,
               deleted.Ship_Id#,
               suser_id(),
               getdate()
        from inserted, deleted
        print "UPDATE Cruise successful"
      end
    end
  end
end
go

```

```

create trigger Delete_Ship
on Ship
for delete
as

```

```

begin
  declare @msg varchar(80),
          @sid int
  if exists (select *
             from Cruise, deleted
             where Cruise.Ship_Id# = deleted.Ship_Id#)
  begin
    select @sid = Ship_Id# from deleted
    select @msg = "DELETE Ship: Please delete the cruises for ship #" +
                 convert(varchar(20),@sid) + " first"

    print @msg
    rollback transaction
  end
end

```

```

end
go

create trigger Update_Ship
on Ship

```

```
for update
as
begin
insert into Ship_Updated
select deleted.Ship_Id#,
inserted.Ship_Name,
inserted.Country,
inserted.Ship_Code,
deleted.Ship_Name,
deleted.Country,
deleted.Ship_Code,
suser_id(),
getdate()
from inserted, deleted
end
go
```

```
/* Eintragen der Nutzer */
grant select, insert, delete on Cruise to PHYSIK1
grant select, insert, delete on Ship to PHYSIK1
go
/* eof cruise.script */
```



```

/* cruise.script      filename: cruise_test
**
** Sybase-Script made in December 1990 by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank
**   Die Datenbank muss nicht mehr erzeugt werden.
**
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Cruise")
drop table Cruise
go

create table Cruise
  (Cruise_Number#      int,      /* internal identification */
   Cruise_number#      int,      /* Cruise number */
   Ship_Id#            int       /* Ship_Id from Ship */)
go

if exists (select * from sysobjects where name="Ship")
drop table Ship
go

create table Ship
  (Ship_Id#            int,      /* internal identification */
   Ship_Name           varchar(80), /* the name of the ship */
   Country             varchar(80) /* the country */)
go

/* Definition der Primaerschlüssel in den Tabellen */
/* Definition der Sekundaerschlüssel in den Tabellen */
/* Generieren der Indizes */
/* Erzeugen der Views */
/* Erzeugen der Prozeduren */
/* Erzeugen von Trigger */
/* Eintragen der Nutzer */

```

wird demnächst im Rechner gelöscht 11.9.91 hi

```
/* Sybase-Script made in December 1990, 1990-12-12 by
** Lutz-Peter Kurdelski
**
** FILENAME: UPDATE_SODB.SCRIPT
**
**     AWI-Rechnergruppe
**
** *****
** * Dieses Script bitte nur ablaufenlassen, wenn die Tabelle *
** * Aari_Cruise noch in der Datenbank enthalten ist.      *
** *****
**
** Umbenennen der Tabelle Aari_Cruise in Aari_Cruise_Bck
** Umbenennen des View Cruise_View in Aari_Cruise
** Entziehen aller Rechte von Aari_Cruise_Bck
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go

use SouthernOceanDB
go

sp_rename Aari_Cruise, Aari_Cruise_Bck
go

sp_rename Cruise_View, Aari_Cruise
go

revoke all on Aari_Cruise_Bck from SOCEAN, PHYSIK1
go
```

```
/* filename : cruise-table */  
/* Sybase-Script made in June 1989 by M. Reinke, Stefan Richter, */  
/* Lutz-Peter Kurdelski */  
/* AWI-Rechnergruppe */  
  
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */  
/* bereits existiert. */  
use master  
go  
/* Erzeugen der Datenbank */  
/* Die Datenbank muss nicht mehr erzeugt werden. */  
if not exists (select * from master.dbo.sysdatabases  
               where name = "SouthernOceanDB")  
begin  
print 'SouthernOceanDB does not exist !!!'  
return  
end  
  
use SouthernOceanDB  
go  
  
grant select on Aari_Cruise to PHYSIK1, SOCEAN  
go
```

wird demnächst im Rechner gelöscht 11.9.91 hi


```
/*
** Sybase-Script made in by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
** Creation-Date: 1991-02-07
**
** Changing of cruise ids in table Cruise which has been
** erroneous in table Aari_Cruise_Bck.
**
** Aendern der Cruise Identifikation in der Tabelle Cruise,
** da diese Daten aus der an dieser Stelle fehlerhaften
** Tabelle Aari_Cruise_Bck entnommen wurden.
**
** FILE: CRUISE_UPD070291.SCRIPT
*/

/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank
** Die Datenbank muss nicht mehr erzeugt werden.
*/
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
/* Definition der Primaerschlüssel in den Tabellen */
/* Definition der Sekundaerschlüssel in den Tabellen */
/* Generieren der Indizes */
/* Erzeugen der Views */
/* Erzeugen der Prozeduren */
/* Erzeugen von Trigger */
/* SQL Statements */

create procedure UpdateShipCruise @cid int,
                                @sn varchar (80) = null,
                                @m int = null
as
print 'enter UpdateShipCruise'
insert Ship
      values(@m,@sn,'unknown',NULL)
update Cruise
      set Ship Id# = @m
      where Cruise Number# = @cid
print 'leave UpdateShipCruise'
go

create table #temp (id# int not null,
                  number int null,
                  name varchar(80) null)

go

begin transaction
go

declare @counter int,
        @max int,
```

```
@m int,  
@number int,  
@s varchar (80),  
@msg varchar(80)
```

```
insert #temp values(1, -25000, 'unknown')  
insert #temp values(2, -25001, 'unknown')  
insert #temp values(3, -25002, 'unknown')  
insert #temp values(4, -25003, 'unknown')  
insert #temp values(5, -25004, 'unknown')  
insert #temp values(6, -25005, 'unknown')  
insert #temp values(7, -25006, 'unknown')  
insert #temp values(8, -25007, 'unknown')  
insert #temp values(9, -25016, 'unknown')  
insert #temp values(10, -25017, 'unknown')  
insert #temp values(11, -25018, 'unknown')  
insert #temp values(12, -25019, 'unknown')  
insert #temp values(13, -25020, 'unknown')  
insert #temp values(14, -25021, 'unknown')  
insert #temp values(15, -25022, 'unknown')  
insert #temp values(16, -25023, 'unknown')  
insert #temp values(17, -25024, 'unknown')
```

```
select @counter = 0  
select @max = (select count(*) from #temp)  
select @m = (select max(Ship_Id#) from Ship where Ship_Id# < 100000)
```

```
while @counter < @max
```

```
begin
```

```
select @counter = @counter + 1
```

```
select @msg = 'loop value = ' + convert(varchar(4),@counter)
```

```
print @msg
```

```
if exists (select * from Cruise
```

```
where Cruise_Number# = (select number from #temp  
where id# = @counter))
```

```
begin
```

```
select @m = @m + 1
```

```
select @msg = ' max ship = ' + convert(varchar(4),@m)
```

```
print @msg
```

```
select @number = (select number from #temp  
where id# = @counter)
```

```
select @msg = ' number = ' + convert(varchar(7),@number)
```

```
print @msg
```

```
select @s = (select name from #temp  
where id# = @counter)
```

```
select @msg = ' name = ' + @s
```

```
print @msg
```

```
exec UpdateShipCruise @cid = @number,  
@sn = @s,  
@m = @m
```

```
end
```

```
end
```

```
select @s =
```

```
(select Ship from Aari_Cruise where Cruise_Number = -25001)
```

```
print @s
```

```
select * from Aari_Cruise
```

```
where Cruise_Number between -25024 and -25016
```

```
or Cruise_Number between -25007 and -25000
```

```
select * from Aari_Cruise where Ship = 'Ob'
```

```
go
```

```
if @@error != 0
```

```
rollback transaction
```

```
else
```

```
commit transaction
```

```
go
drop table #temp
go
drop procedure UpdateShipCruise
go
/* end of cruise_upd070291.script */
```


Script - 31

```
/* SYBASE script by L.-P. Kurdelski
**
** Procedure to set illegal Date_Time values to be NULL.
**
** created 1991-02-12
**
** FILE: update_backfill_proc.script
*/
use SouthernOceanDB
go

create proc Update_Backfill
as
/* created 1991-02-12
** creator L.-P. Kurdelski
**
** Modifying illegal Date_Time values to be NULL
*/

update Gordon_Station_Backfill
set Date_Time = NULL
where Date_Time = 'JAN 1 1900 12:00AM'
go
```

```
/* Sybase-Script made in November 1990 by Lutz-Peter Kurdelski */
/* AWI-Rechnergruppe */
/*
** FILE: GSBACKFILL_UPD070291.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */

use master
go

/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */

if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Gordon_Station_Backfill_BCK")
drop table Gordon_Station_Backfill_BCK
go

if exists (select * from sysobjects where name="Gordon_Station_Backfill")
exec sp_rename Gordon_Station_Backfill, Gordon_Station_Backfill_BCK
go

create table Gordon_Station_Backfill
(Gordon_Station_Id# int, /* internal identification */
Ship_Code varchar (10),
Longitude float, /* Nowlin ongitude */
Latitude float, /* Nowlin atitude */
Date_Time datetime NULL,
/* Nowlin nyear nmonth nda nho */
Day_Of_Year int /* Tagesnummer */)

go

/* Definition der Primaerschluessel in den Tabellen */

execute sp_primarykey Gordon_Station_Backfill, Gordon_Station_Id#
go

/* eof gsbackfill_upd070291.script */
```

```
/**  
** FILE: MAX_MIN_FROM_SHIP.SCRIPT  
**/  
declare @min int,  
        @max int,  
        @msg varchar(80)  
select @min = 0  
select @max = (select max(Ship_Id#)  
              from Ship  
              where Ship_Id# < 100000)  
select @min = (select min(Ship_Id#)  
              from Ship  
              where Ship_Id# between @min and @max)  
select @msg = convert(varchar(10),@min)+' '+convert(varchar(10),@max)  
print @msg
```



```
/* SYBASE script by L.-P. Kurdelski
**
** Creating the table containing the data from
** Tokyo University of Fisheries
**
** created 1991-02-13
**
** FILE: TOKYO_TABLE.SCRIPT
*/

use master
go
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */

/* Standardformat der Stationstabellen ist
** Id#           Identifikator
** Cruise        Nummer der Reise
** Station       Nummer der Station
** Laenge
** Breite
** Zeit
** Tiefe
** bTiefe        beobachtete Tiefe
** n             Anzahl der Messungen
** m             Marsden_Square
*/
if exists (select * from sysobjects
           where name = 'Tokyo_Fisheries_Station')
drop table Tokyo_Fisheries_Station
go

create table Tokyo_Fisheries_Station
  (Station_Id#      int,
   Cruise_Number   int,
   Station_Number  int,
   Longitude        float,
   Latitude         float,
   Date_Time       datetime NULL,
   Bottom_Depth    int NULL,
   Max_Obse_Depth  int,
   Number_Obse     int,
   Marsden_Square# int NULL)
go

/* Standardformat der Tabellen ist
** Id#           Identifikator
** fId#          Join-Feld
** zst           Tiefe
** tst           Temperatur
** sst           Salzgehalt
** oxst          Sauerstoff
*/
if exists (select * from sysobjects
           where name = 'Tokyo_Fisheries_Standard_Data')
drop table Tokyo_Fisheries_Standard_Data
go
```

```
create table Tokyo_Fisheries_Standard_Data
(Standard_Data_Id# int,
 Station_Id#      int,
 Depth           int,
 Temperature     float NULL,
 Salinity        float NULL,
 Oxygen          float NULL)
go

/* Definition der Primaerschluesse in den Tabellen */
execute sp_primarykey Tokyo_Fisheries_Station, Station_Id#
execute sp_primarykey Tokyo_Fisheries_Standard_Data, Standard_Data_Id#
go

/* Definition der Sekundaerschluesel in den Tabellen */
execute sp_foreignkey Tokyo_Fisheries_Standard_Data,
                    Tokyo_Fisheries_Station, Station_Id#
go

/* Zugriffsrechte */
grant select on Tokyo_Fisheries_Station to PHYSIK1
grant select on Tokyo_Fisheries_Standard_Data to PHYSIK1
go

/* eof tokyo_table.script */
```

```
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Cruises to Cruise and Ship
** from Tokyo Fisheries Data
**
** created 1991-02-13
**
** FILE: ADD_TOKYO_SHIPS.SCRIPT
**/

use master
go
if not exists (select * from master.dbo.sysdatabases
              where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

begin transaction
go

declare @shipid int,
        @ship varchar(80),
        @msg varchar(80)

select @ship = 'Umitaka-Maru'
select @shipid = (select Ship_Id#
                 from Ship
                 where Ship_Name like @ship
                 and Ship_Id# < 100000)

insert Cruise values (-26001,@shipid)
insert Cruise values (-26002,@shipid)
insert Cruise values (-26003,@shipid)
insert Cruise values (-26004,@shipid)
insert Cruise values (-26005,@shipid)

select Cruise_Number, Ship from Aari_Cruise where Ship = @ship
go

if @@error != 0
    rollback transaction
else
    commit transaction
go
```



```
/* Script um missing Values auf NULL zu setzen */
/* L.P. Kurdelski 1991-02-14 */
/* Tokyo_Fisheries_Station
**
** FILE: TOKYO_UPD.SCRIPT
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Station
set Bottom_Depth = NULL
where Bottom_Depth < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Standard_Data
set Oxygen = NULL
where Oxygen < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Tokyo_Fisheries_Station
set Marsden_Square# = NULL
where Marsden_Square# = 999
go
dump tran SouthernOceanDB with truncate_only
go

/* eof tokyo_upd.script */
```

```
/*  
** SYBASE Script vom 1991-03-08  
** von L.-P. Kurdelski (AWI)  
**  
** FILE: TOKYO_FISH_BOTTOM_DEPTH_UPDATE.SCRIPT  
**  
** In der Spalte Bottom_Depth der Tabelle Tokyo_Fisheries_Station  
** werden die ungueltigen Werte auf NULL gesetzt.  
** Ein Wert ist ungueltig, wenn er groesser als 8500 ist.  
** Nach Absprache mit V. Guretzky werden zukuenftig ungueltige  
** Werte durch Werte kleiner 0 gekennzeichnet.  
**  
*/  
use SouthernOceanDB  
go  
  
begin transaction  
go  
  
update Tokyo_Fisheries_Station  
set Bottom_Depth = NULL  
where Bottom_Depth > 8500  
go  
  
if @@error != 0  
    rollback transaction  
else  
    commit transaction  
go  
  
/* eof tokyo_fish_bottom_depth_update.script */
```

```
/*  
** SYBASE Script vom 1991-03-08  
** von L.-P. Kurdelski  
**  
** FILENAME: TOKYO_FISHERIES_STATION_UPDATE.SCRIPT  
**  
** Berichtigung einiger fehlerhafter Daten  
*/  
use SouthernOceanDB  
go  
  
begin transaction  
go  
  
update Tokyo_Fisheries_Station  
set Bottom_Depth = 1350  
where Station_Id# = 500002  
  
update Tokyo_Fisheries_Station  
set Latitude = -39.29,  
    Longitude = 115.34  
where Station_Id# = 500114  
  
update Tokyo_Fisheries_Station  
set Date_Time = '3 FEB 1965 23:00'  
where Station_Id# = 500111  
  
go  
  
if @@erro != 0  
    rollback transaction  
else  
    commit transaction  
go
```



```
/*  
** SYBASE Script vom 1991-03-08  
** von L.-P. Kurdelski  
**  
** Modified 1991-08-29 by D. Hinz  
**  
** Update einiger Shiffsdaten in Aari_Cruise  
**  
** FILE: AARI_UPDATE_SHIPS.SCRIPT  
**  
*/
```

```
use SouthernOceanDB  
go
```

```
begin transaction  
go
```

```
exec updateCruise -25000, 'Fuji', 'Japan'  
go
```

```
exec updateCruise -25001, 'Fuji', 'Japan'  
go
```

```
exec updateCruise -25002, 'Fuji', 'Japan'  
go
```

```
exec updateCruise -22012, 'Professor Zubov', 'USSR'  
go
```

```
exec updateCruise -25007, 'Novoukrainka', 'USSR'  
go
```

```
exec updateCruise -22013, 'Prof. Viese', 'USSR'  
go
```

```
if @@error != 0  
    rollback transaction  
else  
    commit transaction  
go
```

```
/* eof aari_update_ships.script */
```

```
/*  
** SYBASE script  
**  
** Creation: 1991-09-03  
**  
** Crfeator: L.-P. Kurdelski  
**  
** FILE: TRUNCATESHIP.SCRIPT  
**  
*/  
use SouthernOceanDB  
go  
  
truncate table Ship  
go  
  
truncate table Cruise  
go  
  
/* end of tuncateship.script */
```

```

/* AWI-Rechnergruppe */

/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
** Created: ??????
**
** Modified: 1991-08-28
**
** Creator: L.-P. Kurdelski
**
** FILE: A1111.SCRIPT
*/
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="A1111_Station")
drop table A1111_Station
go

create table A1111_Station
  (Station_Id# int, /* internal identification */
   Cruise_Number int, /* A1111 nc */
   Station_Number int, /* A1111 ns */
   Longitude float, /* A1111 ongitude */
   Latitude float, /* A1111 atitude */
   Date_Time datetime NULL,
                               /* A1111 nyear nmonth nda nho */
   Bottom_Depth int NULL, /* A1111 nde */
   Max_Obse_Depth int NULL, /* A1111 mode */
   Number_Obse int, /* A1111 hz */
   Marsden_Square# int NULL /* A1111 msq */)
go

if exists (select * from sysobjects where name="A1111_Standard_Data")
drop table A1111_Standard_Data
go

create table A1111_Standard_Data
  (Standard_Data_Id# int, /* internal identification */
   Station_Id# int, /* internal identification */
   Depth int, /* A1111 zst */
   Temperature float NULL, /* A1111 tst */
   Salinity float NULL, /* A1111 sst */
   Oxygen float NULL /* A1111 oxst */)
go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey A1111_Station, Station_Id#
execute sp_primarykey A1111_Standard_Data, Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */

```



```
execute sp_foreignkey A1111_Standard_Data, A1111_Station,  
    Station_Id#  
go  
/* eof a1111.script */
```

```
/* Script um missing Values auf NULL zu setzen
**
** Created: 1990-11-28
**
** Creator: L.-P. Kurdelski
**
** FILE: A1111LOAD_UPD.SCRIPT
**
** A1111_Station
** A1111_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Station
set Bottom_Depth = NULL
where Bottom_Depth < 0 or Bottom_Depth > 8500
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
go
dump tran SouthernOceanDB with truncate_only
go
update A1111_Station
set Marsden_Square# = NULL
where Marsden_Square# = 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
/*
** SYBASE Script by L.-P. Kurdelski
**
** FILE: A1111_SHIPS.SCRIPT
**
** CREATED: 1991-03-27
**
** MODIFIED:
**      1991-08-19      lpk      Procedures insert_into_cruise and
**                               insert_new_ships are expected.
**                               If they are not available they will
**                               be created.
**
**      1991-08-29      lpk      The procedures will not be created.
**
** Insert new ships
** The ships are taken from the A1111 dataset.
*/
if not exists (select * from sysobjects where name like "SouthernOceanDB")
begin
    print "SouthernOceanDB not found"
    return
end
go

use SouthernOceanDB
go

if not exists (select * from sysobjects where name like "insert_into_cruise")
begin
    print "Procedure inser_into_cruise missing."
    return
end
go

if not exists (select * from sysobjects where name like "insert_new_ships")
begin
    print "Procedure insert_new_ships missing."
    return
end
go

if exists (select * from sysobjects where name like "tempships")
begin
    truncate table tempships
end
else
begin
    print "No table tempships"
    return
end
go

declare @zaehler int

select @zaehler = 0
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -21017, "Krusenstern", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -21018, "Krusenstern", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22015, "Prof. Viese", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22016, "Prof. Zubov", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22019, "Pioner Latvii", "USSR")
```



```

select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22020, "Titanit", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22021, "Atlantniro", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22022, "Maltsevo", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22023, "Volniy Veter", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22025, "Pioner Latvii", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22026, "Ocher", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22027, "Patriot", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22030, "Ekliptika", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22031, "Salehard", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22034, "Anchar", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22035, "Novocheboksarsk", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22036, "Afeliy", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22038, "Ekliptika", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22040, "Atlantniro", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22042, "Titanit", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22044, "Monokristall", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22045, "Saulkrasty", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22046, "Andrus Iohann", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22050, "Torok", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22054, "Salehard", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22062, "Tava", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22063, "Anchar", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -22065, "Charoi", "USSR")
go

```

```

begin transaction
go

```

```

exec insert_new_ships
go

```

```

if @@error != 0
begin
print 'transaction aborted'
rollback transaction
end
else
begin
print 'transaction successfull'
commit transaction
end
go

```

```
truncate table tempships  
print "table tempships truncated"  
go
```

```
/* end of A1111_SHIPS.SCRIPT */
```

```
/*
** SYBASE Script by L.-P. Kurdelski
**
** CREATED: 1991-03-27
**
** Modification of the relation between a ship and a cruise.
**
** FILE: A1111_SHIPS_UPD.SCRIPT
*/
use SouthernOceanDB
go

if not exists
    (select * from sysobjects where name like "changeShipNameInCruise")
begin
    print "Procedure changeShipNameInCruise not found."
    return
end
go

exec changeShipNameInCruise "Prof. Vieze", "Prof. Viese", 0, 0
go

begin transaction
$

delete from Ship where Ship_Name like "Prof. Vieze"
go

if @@error != 0
begin
    print 'transaction aborted'
    rollback transaction
end
else
begin
    print 'transaction successfull'
    commit transaction
end
go
```



```
/*
** Sybase script by L.-P. Kurdelski
**
** CREATED: 1991-04-09
**
** FILE: TOKYO_COPY.SCRIPT
**
** Umladen der Tokyo_Fisheries-Daten in die Tabelle Station und StandardData
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name = Tokyo_copy_all)
drop procedure Tokyo_copy_all
go

create procedure Tokyo_copy_all
as
/* Fill Station and Standard Data with Tokyo_Fisheries Data*/
/* L.-P. Kurdelski 1991-04-09 */
begin

insert Station (Station_Id#, Cruise_Number, Station_Number,
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
Number_Obse, Marsden_Square#)
select Station_Id#, Cruise_Number, Station_Number,
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
Number_Obse, Marsden_Square#
from Tokyo_Fisheries_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
Depth, Temperature, Salinity, Oxygen)
select Standard_Data_Id#, Station_Id#, Depth, Temperature, Salinity, Oxygen
from Tokyo_Fisheries_Standard_Data
end
go

begin transaction
go

exec Tokyo_copy_all
go

if @@error != 0
begin
print 'rollback transaction'
rollback transaction
end
else
begin
print 'commit transaction'
commit transaction
end
go

/* eof tokyo_copy.script */
```

```
/* AWI-Rechnergruppe */
/* created by lpk
** 1991-05-17 */
/*
** FILE: JARELOAD.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Jare_Station")
drop table Jare_Station
go

create table Jare_Station
(Jare_Station_Id# int, /* internal identification */
Cruise_Number int, /* Jare CRUNU */
Station_Number int, /* Jare NUMSTAT */
Longitude float, /* Jare A */
Latitude float, /* Jare P */
Date_Time datetime NULL,
/* Jare nyear month nday nhour minut */
Bottom_Depth int NULL, /* Jare ndep */
Max_Obse_Depth int NULL, /* Jare modeph */
Number_Obse int, /* Jare n */
Marsden_Square# int NULL /* Jare msq */)

go

if exists (select * from sysobjects where name="Jare_Standard_Data")
drop table Jare_Standard_Data
go

create table Jare_Standard_Data
(Jare_Standard_Data_Id# int, /* internal identification */
Jare_Station_Id# int, /* internal identification */
Depth int, /* Jare ZZ */
Temperature float NULL, /* Jare TEM */
Salinity float NULL, /* Jare SAL */
Oxygen float NULL, /* Jare OXY */
Phosphate float NULL, /* Jare PO */
Nitrate float NULL, /* Jare N3 */
Silicate float NULL /* Jare SI */)

go

/* Definition der Primaerschlüssel in den Tabellen */
execute sp_primarykey Jare_Station, Jare_Station_Id#
execute sp_primarykey Jare_Standard_Data, Jare_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Jare_Standard_Data, Jare_Station,  
Jare_Station_Id#
```

```
go
```

```
/* eof jareload.script */
```


Script-47

```
/*
** SYBASE script
**
** CREATOR: Lutz-Peter Kurdelski
**
** CREATION: 1991-10-29
**
** FILE: UNKNOWNCHANGES291091.SCRIPT
**
** Changes in the Ship-Cruise Table
** source. oth$daten:[socean.text]unknownchanges.text
**
** Modifications:
** 1991-10-29 "Prof. Viese" should be spelled as "Prof. Vieze"
**           Insert new ship "Investigator", "Australia"
**           "Discovery" is from "U.K."
**           Three new faulty relations corrected fo
**           Cruise_Numbers 9366, 9368, -629.
*/
use SouthernOceanDB
go

declare @smax int

select @smax = max(Ship_Id#) from Ship where Ship_Id# < 100000
select @smax = @smax + 1
insert into Ship (Ship_Id#, Ship_Name, Country)
values (@smax, "Investigator", "Australia")
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Vieze", "USSR", 7660, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Zubov", "USSR", 10186, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Zubov", "USSR", 3494, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Discovery", "U.K.", 55071, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Yelcho", "Chile", 5367, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Edisto", "USA", 3482, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 426, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 634, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 635, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 636, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 637, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 638, 0
go
```

```
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 639, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 643, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Investigator", "Australia", 678, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Gascoyne", "Australia", 410, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Yelcho", "Chile", 5371, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Chipana", "Chile", 5369, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Zubov", "USSR", 14539, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Myslitel", "USSR", 10341, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Vieze", "USSR", 4244, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Melville", "USA", 11669, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Myslitel", "USSR", -1192, 0
go
exec changeShipNameAndCountryInCru
    "Investigator", "India", "Investigator", "Australia", 9366, 0
go
exec changeShipNameAndCountryInCru
    "Investigator", "India", "Investigator", "Australia", 9368, 0
go
exec changeShipNameAndCountryInCru
    "Investigator", "Canada", "Investigator", "Australia", -629, 0
go
```

```
/*
** SYBASE Script
**
** Created 1991-06-05
**
** Creator: L.P. Kurdelski
**
** FILE: JARE_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** Jare_Station
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Station
set Bottom_Depth = NULL
where Bottom_Depth < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Standard_Data
set Oxygen = NULL
where Oxygen < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Jare_Station
set Marsden_Square# = NULL
where Marsden_Square# = 999
go
dump tran SouthernOceanDB with truncate_only
go
```



```
Spec_Vol_Anomaly      float      null,/* Biomass */
                        /* SPEC_VOL ANOMALY */
Delta_D               float      null /* Biomass DELTA_D */)
```

```
go
```

```
/*
```

```
** Definition of primary keys
```

```
*/
```

```
exec sp_primarykey Biomass_Station, Biomass_Station_Id#
```

```
exec sp_primarykey Biomass_Standard_Data, Biomass_Standard_Data_Id#
```

```
go
```

```
/*
```

```
** Definition of foreign keys
```

```
*/
```

```
exec sp_foreignkey Biomass_Standard_Data, Biomass_Station,
                    Biomass_Station_Id#
```

```
go
```

```
/*
```

```
** Enter user
```

```
*/
```

```
grant select on Biomass_Station to PHYSIK1
```

```
grant select on Biomass_Standard_Data to PHYSIK1
```

```
g
```

```
/* end of biomass.script */
```



```
/* SHIPS_UPD050291
**
** Sybase-Script made in by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
** Changes of ship names in table Ship
** - Some ships only defined by their NOCD code now receive
**   the correct name.
** - One name has to be changed
**
** FILE: SHIPS_UPD050291.SCRIPT
**
** Modified 1991-08-29 by D. Hinz
**
*/
```

```
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
```

```
use master
go
```

```
/* Erzeugen der Datenbank
** Die Datenbank muss nicht mehr erzeugt werden.
*/
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
```

```
use SouthernOceanDB
go
```

```
/* Beginn einer Transaction zwecks Fehlerbehebung */
begin transaction
go
```

```
update Ship
  set Ship_Name = 'Goyena'
  where Ship_Name = 'Cayena'
go
```

```
update Ship
  set Ship_Name = 'Billie',
      Ship_Code = 'A5N31'
  where Ship_Name = 'A5N31'
go
```

```
update Ship
  set Ship_Name = 'Eldorado',
      Ship_Code = 'AEL31'
  where Ship_Name = 'AEL31'
go
```

```
update Ship
  set Ship_Name = 'Goyena',
      Ship_Code = 'AGO 8'
  where Ship_Name = 'AGO 8'
go
```

```
update Ship
  set Ship_Name = 'Idaho Standard',
      Ship_Code = 'AIS31'
  where Ship_Name = 'AIS31'
```


go

update Ship

set Ship_Name = 'Mihail Somov',

Ship_Code = 'ASB90'

where Ship_Name = 'ASB90'

go

if @@error != 0

rollback transaction

else

commit transaction

go

/* eof ships_upd050291.script */

```
/*  
** FILE UPDATE_SHIP_ID_IN_CRUISE.SCRIPT  
*/  
create procedure Update_Ship_Id_In_Cruise  
as  
  update Cruise  
  set s.Ship_Id# = g.Ship_Id#  
  from Cruise s, Gordon_Cruise a, Gordon_Ship g, #cruise_ship c  
  where  
    s.Ship_Id# = c.ship  
    and c.cruise = a.Gordon_Cruise_Id#  
    and a.Ship_Code = g.Ship_Code
```

```
/*  
** FILE: UPDATE_SHIP_ID.SCRIPT  
*/  
create procedure Update_Ship_Id  
as  
  update Ship  
  set s.Ship_Id# = g.Ship_Id#  
  from Ship s, Gordon_Cruise a, Gordon_Ship g, #cruise_ship c  
  where  
    s.Ship_Id# = c.ship  
    and c.cruise = a.Gordon_Cruise_Id#  
    and a.Ship_Code = g.Ship_Code
```

letztes Datum der Änderung:

26.6.91

li

Script-53

```
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Ship by presenting the Ship_Name,
** the Country and the Cruise_Number
**
** created 1991-08-28
**
** FILE: INSERTSHIP.SCRIPT
*/
```

```
use master
go
```

```
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
```

```
begin
  print 'SouthernOceanDB does not exist !!!'
  return
end
go
```

```
use SouthernOceanDB
go
```

```
if exists (select * from sysobjects where name like "insertShip")
  drop proc insertShip
```

```
go
```

```
create proc insertShip ( @ship varchar(80),
                        @country varchar(80),
                        @cru int)
```

```
as
/* SYBASE script by L.-P. Kurdelski
```

```
**
** Inserting new Ship by presenting the Ship_Name,
** the Country and the Cruise_Number
**
```

```
** created 1991-08-28
```

```
**
** FILE: INSERTSHIP.SCRIPT
*/
```

```
begin
```

```
  declare @shipid int,
          @msg varchar(80)
```

```
begin transaction
```

```
if exists (select Ship_Id#
           from Ship
           where Ship_Name like @ship)
```

```
begin
```

```
  select @shipid = (select Ship_Id#
                   from Ship
                   where Ship_Name like @ship)
```

```
end
```

```
else
```

```
begin
```

```
  select @shipid = (select max(Ship_Id#)
                   from Ship
                   where Ship_Id# < 100000)
```

```
if @shipid is null
```

```
  select @shipid = 1
```

```
else
```

```
  select @shipid = @shipid + 1
```

```
insert Ship
values (@shipid, @ship, @country, NULL)
end
```

```
insert Cruise values (@cru,@shipid)
```

```
select Cruise_Number, Ship from Aari_Cruise where Ship like @ship
```

```
print @msg
```

```
if @@error != 0
```

```
begin
```

```
rollback transaction
```

```
select @msg = "ERROR: Nothing inserted"
```

```
end
```

```
else
```

```
begin
```

```
commit transaction
```

```
select @msg = "Ship " + @ship + " and Cruise " +
convert(varchar(20),@cru) + " inserted"
```

```
end
```

```
print @msg
```

```
end
```

```
g
/* eof insertship.script */
```

```
/*
** SYBASE Script by L.-P. Kurdelski
**
** FILE: INSERTSHIPCRUISE.SCRIPT
**
** CREATED: 1991-03-27
**
** MODIFIED: 1991-08-19 lpk      Procedures insert_into_cruise and
**                               insert_new_ships are created
**
*/
if not exists (select * from sysobjects where name like "SouthernOceanDB")
begin
    print "SouthernOceanDB not found"
    return
end
go

use SouthernOceanDB
go

if exists (select * from sysobjects where name like "tempships")
    drop table tempships
go

create table tempships ( Id# int,
                        cn int,
                        ship varchar(80),
                        country varchar(80))
go

if exists (select * from sysobjects where name like "insert_into_cruise")
    drop procedure insert_into_cruise
go

create proc insert_into_cruise
    ( @cruisen int, @ship varchar(20), @scountry varchar(20))
as
/* Creation of a stored procedure which inserts a Ship into the
** Cruise table. The parameters are
**   cruisen      Cruise_Number
**   ship         Ship_Name
**   scountry     Country
**
** The procedure looks for the right Ship_Id# of the Ship Name
** found in table Ship. Then the Ship_Id# and the Cruise_Number
** will be inserted into the Cruise table.
**
*/

declare @sid int

select @sid = Ship_Id#
    from Ship
    where Ship_Name like @ship and
           Country like @scountry

if @sid is null
begin
    select @sid = max(Ship_Id#) + 1
        from Ship
        where Ship_Id# < 100000
    insert into Ship values(@sid, @ship, @scountry, null)
end
insert into Cruise values (@cruisen, @sid)
go
```



```
if exists (select * from sysobjects where name like "insert_new_ships")
    drop procedure insert_new_ships
go

create proc insert_new_ships
    as
/*
** insert ship from a table tempships into table Ship and
** the pair Ship_id# and Cruise_Number# into Cruise.
** This procedure requires the procedure insert_into_cruise
*/
    declare @zaehler int,
            @cruisen int,
            @ship varchar(20),
            @scountry varchar(20),
            @m int

    select @zaehler = 1
    select @m = max(Id#) from tempships

    while @zaehler <= @m
    begin
        select @cruisen = cn,
               @ship = ship,
               @scountry = country
        from tempships
        where Id# = @zaehler

        exec insert_into_cruise @cruisen, @ship, @scountry

        select @zaehler = @zaehler+1
    end
go

/* end of insertshipcruise.script */
```

Insert-new-Ships.script
26.6.91

```
create proc insert_new_ships
as
declare @zaehler int,
        @cruisen int,
        @ship varchar(20),
        @scountry varchar(20),
        @m int

select @zaehler = 1
select @m = max(id#) from temptab

while @zaehler <= @m
begin
    select @cruisen = Cruise_Number,
           @ship = Ship,
           @scountry = Country

    from temptab
    where id# = @zaehler

    exec insert_into_cruise @cruisen, @ship, @scountry

    select @zaehler = @zaehler+1
end
```

neu!

#54 ersetzt durch Insertshipcruise.script

11.9.91 hi

Script-54

Insert into Cruise Script

26.6.91

```
/*
** SYBASE Script by L.-P. Kurdelski
**
** CREATED:
**
** Creatio of a stored procedure which inserts a Ship into the
** Cruise table. The parameters are
**     cruise_number
**     ship_name
**     country
**
** The procedure looks for the right Ship_ID# of the Ship_Name
** found in table Ship. Then the Ship_Id# and the Cruise_Number will
** be inserted into the Cruise table.
**
** FILE: INSERT_INTO_CRUISE.SCRIPT
**
*/
create proc insert_into_cruise ( @cruisen int, @ship varchar(20),
                                @scountry varchar(20))
as
    declare @sid int
    select @sid = Ship_Id# from Ship
        where Ship_Name like @ship and Country like @scountry

    if @sid is null
        begin
            select @sid = max(Ship_Id#) + 1 from Ship
            insert into Ship
                values(@sid, @ship, @scountry, null)
        end
    insert into Cruise
        values (@cruisen, @sid)
```

neu!

#53 ersetzt durch Insertship.script 11.9.91 li

Script 53

28.5.91

```
/* Sybase script to modify the Gordon_Cruise table.
** created by lpk, AWI 1991-05-28
**
** column Ship_Code is removed and replaced by column Ship_Id#
** This is done for simpliarity and more consintency within the
** definitions of the other table in the database.
**
** FILE: GORDON_CRUISE_DELETE_SHIP_CODE.SCRIPT
*/

/* The database must contain the database SouthernOceanDB
*/

if exists select * from sysobjects where name like "Gordon_Cruise_LPK"
drop table Gordon_Cruise_LPK
go

create table Gordon_Cruise_LPK (Gordon_Cruise_Id# int,
                                Ship_Id# int,
                                Gordon_Cruise_Name varchar(10 null,
                                Comment varchar(255) null)

go

begin transaction
go

insert Gordon_Cruise_LPK (Gordon_Cruise_Id#,
                          Ship_Id#,
                          Gordon_Cruise_Name,
                          Comment)

select Gordon_Cruise_Id#,
       Ship_Id#,
       Gordon_Cruise_Name,
       Comment
from Gordon_Cruise, Gordon_Ship
where Gordon_Cruise.Ship_Code like Gordon_Ship.Ship_Code
go

if @@error != 0
begin
rollback transaction
print "transaction rollback done"
end
else
begin
commit transaction
print "transaction successfull"
end
go
```

liberal
ersatzlos streichen, da nie gelaufen
und im Rechner schon gelöscht

```
/*
** SYBASE script
**
** FILENAME: UPDATECRUISE.SCRIPT
**
** Created: 1991-08-28
**
** Creator: L.-P. Kurdelski
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "updateCruise")
    drop proc updateCruise
go

create procedure updateCruise
    @cru int,
    @ship varchar(80),
    @country varchar(80)
as
/*
** A relation between a ship and a cruise is updated
** i.e. the ship id of the given ship is the replacement
** for the ship id in the table due to the cruise number.
*/
declare @s int

begin
    select
        @s = Ship_Id#
    from
        Ship
    where
        Ship_Name like @ship and
        Country like @country

    update Cruise
    set Ship_Id# = @s
    where Cruise_Number# = @cru
end
go
```

Script-56

```
/* Sybase script
** 1991-06-11 by Jens Begemann and L.-P. Kurdelski
**
** Update of Ship and Cruise in SouthernOceanDB
**
** Data from V. Guretzky
**
** FILE UpdateShipCruise100691.script
**
*/
```

```
/* assume SouthernOceanDB is available */
use SouthernOceanDB
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
select @sid = Ship_Id# from Ship where Ship_Name like "Ob"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Kokar"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt
go
```

```
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
select @sid = Ship_Id# from Ship where Ship_Name like "Eltanin"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Lesnoi"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -667
go
```

```
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go
```



```
begin transaction
go

declare @sid int,
        @sidalt int
select @sid = Ship_Id# from Ship where Ship Name like "Discovery II"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Walther Herwig"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1123
go
```

```
if @@error != 0
begin
rollback transaction
print "rollback"
end
else
begin
commit transaction
print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Cap. Canepa"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Dolphin"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number#=-1809
go
```

```
if @@error != 0
begin
rollback transaction
print "rollback"
end
else
begin
commit transaction
print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Investigator"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number#=643
go
if @@error != 0
```

```
begin
  rollback transaction
  print "rollback"
end
else
begin
  commit transaction
  print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Gascoyne"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number#=424
```

```
if @@error != 0
begin
  rollback transaction
  print "rollback"
end
else
begin
  commit transaction
  print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Melville"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and (Cruise_Number#=12031 or Cruise_Number#=-21814)
```

```
go
if @@error != 0
begin
  rollback transaction
  print "rollback"
end
else
begin
  commit transaction
  print "commit"
end
go
```

```
begin transaction
go
```

```

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Atlantis II"

select @sidalt = Ship_Id# from Ship where Ship_Name like "Atlantis"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number#=-180
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Discovery II"

select @sidalt = Ship_Id# from Ship where Ship_Name like "Schwabenland"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -12
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Salehard"

select @sidalt = Ship_Id# from Ship where Ship_Name like "Kvant"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1117
go
if @@error != 0
begin
    rollback transaction

```



```

    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Gen. San-Martin"

select @sidalt = Ship_Id# from Ship
        where Ship_Name like "unknown" and
               Ship_Id# in (select Ship_Id# from Cruise
                           where Cruise_Number# = -1929)

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1929
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Tui"

select @sidalt = Ship_Id# from Ship where Ship_Name like "Kiwi"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -21649
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

```

```
declare @sid int,  
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Edisto"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = 3482  
go  
if @@error != 0  
begin  
    rollback transaction  
    print "rollback"  
end  
else  
begin  
    commit transaction  
    print "commit"  
end  
go
```

```
begin transaction  
go
```

```
declare @sid int,  
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Anton Bruun"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Burton Island"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = -816  
go  
if @@error != 0  
begin  
    rollback transaction  
    print "rollback"  
end  
else  
begin  
    commit transaction  
    print "commit"  
end  
go
```

```
begin transaction  
go
```

```
declare @sid int,  
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Natal"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Africana II"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = 5406  
go  
if @@error != 0  
begin  
    rollback transaction
```

```

    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Umitaka-Maru"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Poceidon"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1934
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Eltanin"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Mitory-Maru"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1677
go
if @@error != 0
begin
    rollback transaction
    print "rollback"
end
else
begin
    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

```



```
select @sid = Ship_Id# from Ship where Ship_Name like "Atka"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Atlantis II"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -180
go
if @@error != 0
begin
rollback transaction
print "rollback"
end
else
begin
commit transaction
print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Hudson"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 6468
go
if @@error != 0
begin
rollback transaction
print "rollback"
end
else
begin
commit transaction
print "commit"
end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Discovery"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Discovery II"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -21137
go
if @@error != 0
begin
rollback transaction
print "rollback"
end
```

```
else
  begin
    commit transaction
    print "commit"
  end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Carnegie"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 1393
```

```
go
if @@error != 0
  begin
    rollback transaction
    print "rollback"
  end
else
  begin
    commit transaction
    print "commit"
  end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Burton Island"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Glacier"
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 619
```

```
go
if @@error != 0
  begin
    rollback transaction
    print "rollback"
  end
else
  begin
    commit transaction
    print "commit"
  end
go
```

```
begin transaction
go
```

```
declare @sid int,
        @sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Norvegia"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Adm. Moucheg"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = -97
```

```
go  
if @@error != 0  
begin  
rollback transaction  
print "rollback"  
end
```

```
else  
begin  
commit transaction  
print "commit"  
end  
go
```

```
begin transaction  
go
```

```
declare @sid int,  
@sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Diamantina"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Olonets"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = 430
```

```
go  
if @@error != 0  
begin  
rollback transaction  
print "rollback"  
end
```

```
else  
begin  
commit transaction  
print "commit"  
end
```

```
go  
begin transaction  
go
```

```
declare @sid int,  
@sidalt int
```

```
select @sid = Ship_Id# from Ship where Ship_Name like "Vityaz"
```

```
select @sidalt = Ship_Id# from Ship where Ship_Name like "Evrica"
```

```
update Cruise  
set Ship_Id# = @sid  
where Ship_Id# = @sidalt and Cruise_Number# = 42
```

```
go  
if @@error != 0  
begin  
rollback transaction  
print "rollback"  
end
```

```
else  
begin
```



```

    commit transaction
    print "commit"
end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Hakuho_Maru"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Prof. Zubov"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 5547
go
if @@error != 0
    begin
        rollback transaction
        print "rollback"
    end
else
    begin
        commit transaction
        print "commit"
    end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Slava"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Ushakov"

update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 9768
go
if @@error != 0
    begin
        rollback transaction
        print "rollback"
    end
else
    begin
        commit transaction
        print "commit"
    end
go

begin transaction
go

declare @sid int,
        @sidalt int

select @sid = Ship_Id# from Ship where Ship_Name like "Discovery"
select @sidalt = Ship_Id# from Ship where Ship_Name like "Billie"

```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = 54007
go
if @@error != 0
  begin
    rollback transaction
    print "rollback"
  end
else
  begin
    commit transaction
    print "commit"
  end
end
go
/* end of UpdateShipCruise100691.script */
```

```
/*  
** Author: Jens Begemann  
  
** Creation Date: 24.6.1990  
** Modification Date: 25.6.1990  
**  
** File: updatecru24.script  
**  
** Description: Changes in table Cruise  
**              Ship_Id# of selected Cruise_Number# is changed according to  
**              Ship_Id# of the table Ship where Ship_Name and Country  
**              are unknown  
*/
```

```
begin transaction  
go
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 202
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 266
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 410
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 424
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 426
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 634
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 635
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 636
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 637
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 638
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 639
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 643
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 678
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 891
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 895
```

```
update Cruise  
set Ship_Id# = 1000000 where Cruise_Number# = 896
```

```
update Cruise
```


set Ship_Id# = 1000000 where Cruise_Number# = 910
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 1078
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 1393
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2102
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2157
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2162
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2326
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2968
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 2970
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3020
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3070
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3359
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3482
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3494
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3927
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3958
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3995
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3996
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 3997
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4031
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4063
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4231
update Cruise

set Ship_Id# = 1000000 where Cruise_Number# = 4244

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4375

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4395

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4564

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4576

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4597

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4813

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4835

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 4859

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5027

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5223

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5367

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5369

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5371

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5373

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5455

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5547

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5552

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5581

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5596

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 5610

update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 6424

update Cruise

set Ship_Id# = 1000000 where Cruise_Number# = 6468
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 6525
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 7660
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 8683
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 8755
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 9230
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 9443
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 9689
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10109
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10167
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10186
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10279
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10282
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10291
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10341
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10350
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10422
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 10522
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 11319
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 12031
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 12220
update Cruise
set Ship_Id# = 1000000 where Cruise_Number# = 12864
update Cruise


```
set Ship_Id# = 1000000 where Cruise_Number# = 13968
```

```
update Cruise
```

```
set Ship_Id# = 1000000 where Cruise_Number# = 14539
```

```
update Cruise
```

```
set Ship_Id# = 1000000 where Cruise_Number# = 15000
```

```
go
```

```
if @@error != 0
```

```
begin
```

```
rollback transaction
```

```
print "rollback"
```

```
end
```

```
else
```

```
begin
```

```
commit transaction
```

```
print "commit"
```

```
end
```

```
go
```

```
/* eof updatecru24.script */
```

```

/*
**++
** SYBASE script by D. Hinz, L.-P. Kurdelski
**
** FILENAME: BIOMASS_BRANDINI.SCRIPT
**
** Creation date: 1991-07-22
**
** Abstract: Inserting chlorophyll values for BIOMASS
**           from a source given by F. Brandini.
**           Preparing the insertion procedure.
**--
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "insertChlorophyllA")
  drop proc insertChlorophyllA
go

create proc insertChlorophyllA
    (@sn      varchar(8),      /*Station Number*/
     @d       int,             /*Depth*/
     @c_a     float            /*Chlorophyll_A*/
    )

as

begin

declare

    /*Tabelle Biomass_Station*/

    @bsid    int,              /*Biomass_Station_Id#*/
    @bsidd   int,              /*Biomass_Station_Id#*/
    @cnu     int,              /*Cruise Number*/
    @lat     float,            /*Latitude*/
    @long    float,            /*Longitude*/
    @d_t     datetime,         /*Date Time*/
    @m_s     int,              /*Marsden Square#*/
    @cna     varchar(8),      /*Cruise_Name*/

    /*Tabelle Biomass_Standard_Data*/

    @bsdid   int,              /*Biomass_Standard_Data_Id#*/
    @dd      int,              /*Depth*/
    @pd      int,              /*PrDepth*/
    @ot      char(1),          /*ObsType*/
    @dt      char(1),          /*DepthType*/
    @temp    float,            /*Temperature*/
    @sal     float,            /*Salinity*/
    @ox      float,            /*Oxygen*/
    @ip      float,            /*InPhos*/
    @ni      float,            /*Nitrite*/
    @na      float,            /*Nitrate*/
    @am      float,            /*Ammonium*/
    @si      float,            /*Silicate*/
    @ph      float,            /*PH*/
    @chl     float,            /*Chloride*/
    @c_aa    float,            /*Chlorophyll_A*/
    @st      float,            /*Sigma T*/
    @sto     float,            /*Sgma T Obs*/
    @s_v     float,            /*Sound Velocity*/
    @d_h     float,            /*Dynamic_Height*/

```

```

@sth float, /*Sigma_Theta*/
@p_t float, /*Pot_Temp*/
@sca float, /*Spec_Col Anomaly*/
@d_d float /*Delta_D*/

```

```

select @bsid = Biomass_Station_Id#,
@cnu = Cruise_Number,
@sn = Station_Number,
@lat = Latitude,
@long = Longitude,
@dt = Date_Time,
@m_s = Marsden_Square#,
@cna = Cruise_Name
from Biomass_Station
where Cruise_Name like "BES%"
and Station_Number like @sn
and Biomass_Station_Id# < 880000

```

```

select @bsdid = Biomass_Standard_Data_Id#,
@bsidd = Biomass_Station_Id#,
@dd = Depth, /*wird ersetzt!!!*/
@pd = PrDepth,
@ot = ObsType,
@dt = DepthType,
@temp = Temperature,
@sal = Salinity,
@ox = Oxygen,
@ip = InPhos,
@ni = Nitrite,
@na = Nitrate,
@am = Ammonium,
@si = Silicate,
@ph = PH,
@chl = Chloride,
@c_aa = Chlorophyll_A, /*wird ersetzt!!!*/
@st = Sigma_T,
@sto = Sigma_T_Obs,
@s_v = Sound_Velocity,
@d_h = Dynamic_Height,
@sth = Sigma_Theta,
@p_t = Pot_Temp,
@sca = Spec_Col_Anomaly,
@d_d = Delta_D
from Biomass_Standard_Data
where Biomass_Station_Id# = @bsid
and Depth = @d
and Biomass_Standard_Data_Id# < 8800000

```

```

select @bsid = @bsid + 80000
select @bsdid = @bsdid + 800000

```

```

/*
**Die Stationen werden dupliziert und, mit einer neuen Kennung versehen,
**als neue Station eingefuegt. Dies geschieht aus Gruenden der
**Vollstaendigkeit. Diese Stationen koennen zu einem spaeteren Zeitpunkt
**wieder geloescht werden, wenn gekklaert ist, dass die Stationsdaten so
**belassen werden koennen.
*/

```

```

if not exists
(select *
from Biomass_Station
where Biomass_Station_Id# = @bsid)
begin
insert Biomass_Station
(Biomass_Station_Id#,
Cruise_Number,

```



```
Station Number,  
Latitude,  
Longitude,  
Date Time,  
Marsden Square#,  
Cruise Name)
```

```
values (@bsid, @cnu, @sn, @lat, @long, @d_t, @m_s, @cna)
```

```
end
```

```
/*
```

```
**Die Daten werden dupliziert und, mit einer neuen Kennung versehen, als  
**neuer Datensatz in die Tabelle eingefuegt. Dies geschieht, da die  
**urspruenglichen Stationsdaten KEINE Chlorophyll A-Werte enthielten,  
**aber die neuen Werte als zu den alten Station gehoerig betrachtet werden.  
**zu einem spaeteren Zeitpunkt koennen diese Stationsdaten wieder geloescht  
**werden, wenn die BIOMASS-Daten in der nun gueltigen Form gehalten  
**werden sollen.
```

```
*/
```

```
insert Biomass_Standard_Data  
(Biomass_Standard_Data_Id#,  
Biomass_Station_Id#,  
Depth,  
PrDepth,  
ObsType,  
DepthType,  
Temperature,  
Salinity,  
Oxygen,  
InPhos,  
Nitrite,  
Nitrate,  
Ammonium,  
Silicate,  
PH,  
Chloride,  
Chlorophyll_A,  
Sigma_T,  
Sigma_T_Obs,  
Sound_Velocity,  
Dynamic_Height,  
Sigma_Theta,  
Pot_Temp,  
Spec_Col_Anomaly,  
Delta_D)  
values (@bsdid, @bsid, @d, @pd, @ot, @dt, @temp, @sal, @ox, @ip,  
@ni, @na, @am, @si, @ph, @chl, @c_a, @st, @sto, @s_v, @d_h,  
@sth, @p_t, @sca, @d_d)
```

```
end
```

```
go
```

```
/* end of biomass_brandini.script */
```

```
/*
** Sybase script by L.-P. Kurdelski
**
** CREATED: 1991-03-22
**
** Modification:
**
** 1991-08-29   lpk       proc will first be dropped then recreated
**                  and started.
**
** FILE: A1111_COPY.SCRIPT
**
** Umladen der A1111-Daten in die Tabelle Station und StandardData
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "A1111_copy_all")
    drop proc A1111_copy_all
go

create procedure A1111_copy_all
as
/* Fill Station and Standard Data with A1111 Data*/
/* L.-P. Kurdelski 1991-03-27 */
begin

    insert Station (Station_Id#, Cruise Number, Station Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#)
    select Station_Id#, Cruise Number, Station Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#
    from A1111_Station

    insert Standard_Data (Standard_Data_Id#, Station_Id#,
        Depth, Temperature, Salinity, Oxygen)
    select Standard_Data_Id#, Station_Id#, Depth, Temperature, Salinity, Oxygen
    from A1111_Standard_Data
end
go

begin transaction
go

exec A1111_copy_all
go

if @@error != 0
begin
    print 'rollback transaction'
    rollback transaction
end
else
begin
    print 'commit transaction'
    commit transaction
end
go

/*
** end of A1111_copy.script
*/
```



```
/* created by J.Begemann, L.P. Kurdelski */
/*
/* Bioselposdat.script */
/*
/* creat.Date: 7.8.1991 */
/*
/* Procedure to compare the data of position and time for each */
/* Biomass_Station record with the data of position and time */
/* in the Station-Table */
/* if there is coincidence between these data within these conditions */
/* the Station_Number of this concerned record is written out */
```

```
create proc bioselposdat
```

```
as begin
```

```
declare @mins int,
        @maxs int,
        @lat float,
        @lon float,
        @dat datetime,
        @i int,
        @msg varchar(80)
```

```
select @mins = (select min(Biomass_Station_Id#)
                from Biomass_Station)
```

```
select @maxs = (select max(Biomass_Station_Id#)
                from Biomass_Station)
```

```
select @i = @mins
```

```
while @i <= @maxs
begin
```

```
    select @lat = Latitude, @lon = Longitude, @dat = DateTime
    from Biomass_Station
    where Biomass_Station_Id# = @i
```

```
    if exists (select * from Station
              where Latitude between @lat - 0.02 and @lat + 0.02
              and
              Longitude between @lon - 0.02 and @lon + 0.02
              and
              Datetime = @dat
```

```
        begin
```

```
            select @msg = "Station_Number" + convert(varchar(10),@i)
```

```
        end
```

```
    print @msg
```

```
    select @i = @i+1
```

```
end
```



```
/*  
** Sybase-Script made 28.06.90 by M. Reinke  
** modified 05.08.91  
**  
** FILENAME: AARI_COPY.SCRIPT  
**/  
*/
```

```
create procedure Aari_copy  
@id# int  
as  
begin  
insert Station (Station_Id#, Cruise_Number, Station_Number, Date_Time, Longitu  
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag)  
select * from Aari Station where  
Aari_Station_Id#=@id#  
insert Standard_Data (Standard_Data_Id#, Station_Id#, Depth, Temperature, Salinit  
select * from Aari Standard_Data where  
Aari_Station_Id#=@id#  
end
```

```
/*  
** FILENAME: INDEX.SCRIPT  
**  
**Sybase-Script made 07.12.89 by M. Reinke  
*/  
  
use SouthernOceanDB  
go  
dump tran SouthernOceanDB with no_log  
go  
  
if exists (select * from sysindexes where name = "Aari_geo_index")  
begin  
drop index Aari_Station.Aari_geo_index  
end  
go  
  
create clustered index Aari_geo_index on  
Aari_Station (Latitude,Longitude)  
go  
dump tran SouthernOceanDB with no_log  
go  
  
if exists (select * from sysindexes where name = "Aari_marsden_index")  
begin  
drop index Aari_Station.Aari_marsden_index  
end  
go  
  
create index Aari_marsden_index on  
Aari_Station (Marsden_Square#)  
go  
dump tran SouthernOceanDB with no_log  
go  
  
if exists (select * from sysindexes where name = "Aari_data_index")  
begin  
drop index Aari_Standard_Data.Aari_data_index  
end  
go  
  
create clustered index Aari_data_index on  
Aari_Standard_Data (Aari_Station_Id#,Depth)  
go  
dump tran SouthernOceanDB with no_log  
go  
  
if exists (select * from sysindexes where name = "Aari_temperature_index")  
begin  
drop index Aari_Standard_Data.Aari_temperature_index  
end  
go  
  
create index Aari_temperature_index on  
Aari_Standard_Data (Temperature)  
go  
dump tran SouthernOceanDB with no_log  
go
```

```
/*  
** FILENAME: UPDATE_DEFAULT_NULL.SCRIPT  
*/  
  
/* Script um missing Values auf NULL zu setzen */  
/* M. Reinke 7.9.89 */
```

```
use SouthernOceanDB  
go  
dump tran SouthernOceanDB with no_log  
go  
update Aari_Station  
set Date_Time = NULL  
where Date_Time="Jan 01 1900 00:00:00.0"  
go  
dump tran SouthernOceanDB with no_log  
go  
update Aari_Standard_Data  
set Temperature = NULL  
where Temperature = -9999  
go  
dump tran SouthernOceanDB with no_log  
go  
update Aari_Standard_Data  
set Salinity = NULL  
where Salinity = -9999  
go  
dump tran SouthernOceanDB with no_log  
go  
update Aari_Standard_Data  
set Oxygen = NULL  
where Oxygen = -9999  
and Depth <= 100  
go  
dump tran SouthernOceanDB with no_log  
go  
go  
update Aari_Standard_Data  
set Oxygen = NULL  
where Oxygen = -9999  
and Depth > 100  
go  
dump tran SouthernOceanDB with no_log  
go  
update Gordon_Station  
set Date_Time = NULL  
where Date_Time="Jan 01 1900 00:00:00.0"  
go  
dump tran SouthernOceanDB with no_log  
go  
update Gordon_Station  
set Bottom_Depth = NULL  
where Bottom_Depth = 0  
go  
dump tran SouthernOceanDB with no_log  
go  
update Gordon_Station  
set Marsden_Square# = NULL  
where Marsden_Square# = 0  
go  
dump tran SouthernOceanDB with no_log  
go  
update Gordon_Standard_Data  
set Temperature = NULL  
where Temperature = -9999  
go
```



```
dump tran SouthernOceanDB with no_log
go
update Gordon_Standard_Data
set Salinity = NULL
where Salinity = -9999
go
dump tran SouthernOceanDB with no_log
go
update Gordon_Standard_Data
set Oxygen = NULL
where Oxygen = -9999
go
dump tran SouthernOceanDB with no_log
go
update Gordon_Standard_Data
set Phosphate = NULL
where Phosphate = -9999
go
dump tran SouthernOceanDB with no_log
go
update Gordon_Standard_Data
set Silicate = NULL
where Silicate = -9999
go
dump tran SouthernOceanDB with no_log
go
update Gordon_Standard_Data
set Nitrat = NULL
where Nitrat = -9999
go
dump tran SouthernOceanDB with no_log
go
/* end of update_default_null.script */
```

```
/*
** FILENAME: STATION.SCRIPT
*/

/* Sybase-Script made in June 1989 by M. Reinke */
/* AWI-Rechnergruppe 2.Teil Gordon Daten*/
use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name = "Station")
begin
drop table Station
end
go

create table Station
(Station_Id# int,
Cruise_Number int NULL,
Station_Number int NULL,
Date_Time datetime NULL,
Longitude float,
Latitude float,
Bottom_Depth int NULL,
Max_Obse_Depth int NULL,
Number_Obse int,
Marsden_Square# int NULL,
Validation_Flag int NULL,
Update_Flag int NULL)
go

if exists (select * from sysobjects where name = "Standard_Data")
begin
drop table Standard_Data
end
go

create table Standard_Data
(Standard_Data_Id# int,
Station_Id# int,
Depth int,
Temperature float NULL,
Salinity float NULL,
Oxygen float NULL,
Phosphate float NULL,
Silicate float NULL,
Nitrat float NULL,
Validation_Flag int NULL,
Update_Flag int NULL)
go

if exists (select * from sysobjects where name = "Statistic")
begin
drop table Statistic
end
go

create table Statistic
(Depth# int,
Marsden_Square# int,
Temperature_Mean float NULL,
Temperature_Var float NULL,
Temperature_Number_Observ float NULL,
Salinity_Mean float NULL,
```



```
Salinity_Var          float NULL,  
Salinity_Number_Observ float NULL,  
Oxygen_Mean          float NULL,  
Oxygen_Var           float NULL,  
Oxygen_Number_Observ float NULL,  
Validation_Method_Id# int NULL)
```

```
go
```

```
/* Definition der Primaerschlüssel in den Tabellen */
```

```
execute sp_primarykey Station, Station_Id#  
execute sp_primarykey Standard_Data, Standard_Data_Id#  
execute sp_primarykey Statistic, Depth#, Marsden_Square#  
go
```

```
/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Standard_Data, Station,  
                        Station_Id#
```

```
go
```

```
/* Eintragen der Nutzer */
```

```
grant all on Station to AWIDB  
grant all on Standard_Data to AWIDB  
grant all on Statistic to AWIDB  
go  
grant all on Station to AWIEDV  
grant all on Standard_Data to AWIEDV  
grant all on Statistic to AWIEDV  
go  
grant select on Station to PHYSIK1  
grant select on Standard_Data to PHYSIK1  
grant select on Statistic to PHYSIK1  
go
```

```
/* Sybase-Script made in May 1990 by M. Reinke */
```

```
/* AWI-Rechnergruppe 2.Teil Station Daten*/
```

```
use SouthernOceanDB
```

```
go
```

```
if exists (select * from sysobjects where name="DeleteStation")
```

```
begin
```

```
drop table DeleteStation
```

```
end
```

```
go
```

```
create table DeleteStation  
  (Station_Id# int,  
   DeleteDate datetime,  
   UserId# int)
```

```
go
```

```
if exists (select * from sysobjects where name="UpdateStation")
```

```
begin
```

```
drop table UpdateStation
```

```
end
```

```
go
```

```
create table UpdateStation  
  (Station_Id# int,  
   Validation_Flag int,  
   UpdateDate datetime,  
   UserId# int)
```

```
go
```

```
if exists (select * from sysobjects where name="DeleteData")
```

```
begin
```



```

drop table DeleteData
end
go
create table DeleteData
(Standard_Data_Id# int,
DeleteDate datetime,
UserId# int)
go
if exists (select * from sysobjects where name="UpdateData")
begin
drop table UpdateData
end
go
create table UpdateData
(Standard_Data_Id# int,
Old_Station_Id# int,
New_Station_Id# int,
Validation_Flag int,
UpdateDate datetime,
UserId# int)
go
if exists (select * from sysobjects where name="Validation")
begin
drop table Validation
end
go
create table Validation
(Validation_Id# int,
Method_Name varchar(80),
Short_Remarks varchar(254) NULL,
Program_Name varchar(80) NULL,
Date_of_Program datetime NULL,
Version_of_Program int NULL)
go
Insert Validation (Validation_Id#, Method_Name, Short_Remarks)
values( 0, "No Method", "Field not trapped by any validation program")
go
if exists (select * from sysobjects where name="InsertStation")
begin
drop table InsertStation
end
go
create table InsertStation
(Station_Id# int,
InsertDate datetime,
UserId# int)
go
if exists (select * from sysobjects where name="InsertData")
begin
drop table InsertData
end
go
create table InsertData
(Standard_Data_Id# int,
InsertDate datetime,
UserId# int)
go

```

```
/* Definition der Primaerschlüssel in den Tabellen */
```

```
execute sp_primarykey DeleteStation, Station_Id#
execute sp_primarykey UpdateStation, Station_Id#,UpdateDate
execute sp_primarykey DeleteData, Standard_Data_Id#
execute sp_primarykey UpdateData, Standard_Data_Id#,UpdateDate
execute sp_primarykey Validation, Validation_Id#
execute sp_primarykey InsertStation, Station_Id#
execute sp_primarykey InsertData, Standard_Data_Id#
go
```

```
/* Eintragen der Nutzer */
```

```
grant select on UpdateStation to SOCEAN
grant select on DeleteStation to SOCEAN
grant select on UpdateData to SOCEAN
grant select on DeleteData to SOCEAN
grant select on InsertStation to SOCEAN
grant select on InsertData to SOCEAN
```

```
go
```

```
/* Erzeugen der Prozeduren */
```

```
if exists (select * from sysobjects where name = "makeStation")
begin
drop procedure makeStation
end
go
```

```
create procedure makeStation
as
```

```
/* Fill Station and Standard_Data with all Aari and Gordon Data */
```

```
/* M. Reinke 31.5.90 */
```

```
begin
insert Station (Station_Id#, Cruise_Number, Station_Number,
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag)
select * from Aari_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
Depth, Temperature,Salinity, Oxygen, Validation_Flag,Update_Flag)
select * from Aari_Standard_Data
```

```
insert Station (Station_Id#, Station_Number,
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag)
select Gordon_Station_Id#, Station_Name,
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag
from Gordon_Station
```

```
insert Standard_Data (Standard_Data_Id#, Station_Id#,
Depth, Temperature,Salinity, Oxygen, Phosphate,Silicate,Nitrat,
Validation_Flag,Update_Flag)
select Gordon_Standard_Data_Id#, Gordon_Station_Id#,
Depth, Temperature,Salinity, Oxygen, Phosphate,Silicate,Nitrat,
Validation_Flag,Update_Flag
from Gordon_Standard_Data
```

```
end
```

```
go
```

/* end of station.script */


```
/*  
** FILENAME: MAKESTATION.SCRIPT  
*/  
  
create procedure makeStation  
as  
/* Fill Station and Standard_Data with all Aari and Gordon Data */  
/* M. Reinke 31.5.90 */  
begin  
  
insert Station (Station_Id#, Cruise_Number, Station_Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag)  
select * from Aari_Station  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
Depth, Temperature, Salinity, Oxygen, Validation_Flag, Update_Flag)  
select * from Aari_Standard_Data  
  
insert Station (Station_Id#, Station_Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag)  
select Gordon_Station_Id#, Station_Name,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#, Validation_Flag, Update_Flag  
from Gordon_Station  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
Depth, Temperature, Salinity, Oxygen, Phosphate, Silicate, Nitrat,  
Validation_Flag, Update_Flag)  
select Gordon_Standard_Data_Id#, Gordon_Station_Id#,  
Depth, Temperature, Salinity, Oxygen, Phosphate, Silicate, Nitrat,  
Validation_Flag, Update_Flag  
from Gordon_Standard_Data  
  
end
```

*Hinweis: genau identisch mit dem Ende
von STATION.SCRIPT (Script-64)*

```
/*  
** FILENAME: STATION_INDEX.SCRIPT  
*/  
  
/* Sybase-Script made in June 1989 by M. Reinke */  
/* AWI-Rechnergruppe 2.Teil Gordon Daten*/  
use SouthernOceanDB  
go  
  
/* Erzeugen der Tabellen */  
if not exists (select * from sysobjects where name = "Station")  
begin  
return  
end  
go  
  
create clustered index station_idx on Station  
(Latitude,Longitude, Date_Time)  
go  
  
create unique index station_id#_idx on Station  
(Station_Id#)  
go  
  
create clustered index standard_data_idx on Standard_Data  
(Station_Id#, Depth)  
go  
  
create unique index standard_data_id#_idx on Standard_Data  
(Standard_Data_Id#)  
go
```

```
/*  
** FILENAME: UPDATE_NULL_GORDON_INTER_DATA.SCRIPT  
**  
** Sybase-Script made 28.11.90 by M. Reinke  
*/
```

```
use SouthernOceanDB  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Gordon_Interpolated_Data  
set Temperature = NULL  
where Temperature = -99.9  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Gordon_Interpolated_Data  
set Salinity = NULL  
where Salinity = -99  
go  
dump tran SouthernOceanDB with truncate_only  
go  
update Gordon_Interpolated_Data  
set Oxygen = NULL  
where Oxygen = -99  
go  
dump tran SouthernOceanDB with truncate_only  
go
```



```
/*  
** FILENAME: GORDON_INTER_DATA.SCRIPT  
*/
```

```
/* Sybase-Script made in Sep 1990 by M. Reinke */  
/* AWI-Rechnergruppe 3.Teil Gordon Daten*/  
use SouthernOceanDB  
go
```

```
/* Erzeugen der Indices fuer Gordon_Interpolated_Data*/  
if not exists (select * from sysobjects where name = "Gordon_Interpolated_Data")  
begin  
return  
end  
go
```

```
create clustered index gordon_interpol_data_idx on Gordon_Interpolated_Data  
(Gordon_Station_Id#, Depth)  
go
```

```
create unique index gordon_interplo_data_id#_idx on Gordon_Interpolated_Data  
(Gordon_Interpolated_Data_Id#)  
go
```

```
/* SYBASE script by Jens Begemann
**
** Name: programtable.script
**
** Crea.Date: 01.08.91
**
** Creating the table 'Programs', which contains all programs
** (Fortran, C and scripts), that are have been used on the SouthernOceanDB,
** with names and whole programtext
*/

use master
go
if not exists (select * from master.dbo.sysdatabases
              where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist!!!'
return
end

use SouthernOceanDB
go

/* Creating of the table */
if exists (select * from sysobjects
          where name = 'Programs')
  drop table Programs
go

create table Programs
  (Program_Id#      int,
   Program_Name    varchar(64),
   Creator         varchar(40),
   Date            datetime,
   Description     varchar(255),
   Program_Text    text,
   Type            varchar(8),
   Data            varchar(10),
   Function        varchar(10),
   Curr_Id#       int)

/* Definition des Primaerschluessels in der Tabelle */
execute sp_primarykey Programs, Program_Id#
go
```

```

/* AWI-Rechnergruppe
** created by lpk
** 1991-08-09
**
** FILE_NAME: MUENCHLOAD.SCRIPT
**
* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
* bereits existiert.
*/
use master
go
/*
** Erzeugen der Datenbank
**
** Die Datenbank muss nicht mehr erzeugt werden.
*/
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Muench_Station")
drop table Muench_Station
go

create table Muench_Station
(Muench_Station_Id# int, /* internal identification */
Cruise_Number int, /* Muench CRUNU */
Station_Number int, /* Muench NUMSTAT */
Longitude float, /* Muench A */
Latitude float, /* Muench P */
Date_Time datetime NULL,
Bottom_Depth int NULL, /* Muench nyear month nday nhour minut */
Max_Obse_Depth int NULL, /* Muench ndep */
Number_Obse int, /* Muench modeph */
Marsden_Square# int NULL /* Muench nlev */
/* Muench msq */)
go

if exists (select * from sysobjects where name="Muench_Standard_Data")
drop table Muench_Standard_Data
go

create table Muench_Standard_Data
(Muench_Standard_Data_Id# int, /* internal identification */
Muench_Station_Id# int, /* internal identification */
Depth int, /* Muench ZZ */
Temperature float NULL, /* Muench TEM */
Salinity float NULL /* Muench SAL */)
go

/* Definition der Primaerschlüssel in den Tabellen */
execute sp_primarykey Muench_Station, Muench_Station_Id#
execute sp_primarykey Muench_Standard_Data, Muench_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */

```



```
execute sp_foreignkey Muench_Standard_Data, Muench_Station,  
Muench_Station_Id#
```

```
go
```

```
/* Erzeugen der Views */
```

```
/* Erzeugen der Prozeduren */
```

```
/* Eintragen der Nutzer */
```

```
/* end of muenchload.script */
```

Script-71

```
/*  
** File: Muench_copy_all.script  
*/  
use SouthernOceanDB  
go  
create procedure Muench_copy_all  
as  
/* Fill Station and Standard Data with Muench Data*/  
/* L.-P. Kurdelski 1991-08-12 */  
begin  
  
insert Station (Station_Id#, Cruise Number, Station Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#)  
select Muench_Station_Id#, Cruise Number, Station Number,  
Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,  
Number_Obse, Marsden_Square#  
from Muench_Station  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
Depth, Temperature, Salinity)  
select Muench_Standard_Data_Id#, Muench_Station_Id#,  
Depth, Temperature, Salinity  
from Muench_Standard_Data  
  
end  
go
```

Script-72

```
/* Script um missing Values auf NULL zu setzen
** L.P. Kurdelski 1991-08-28
**
** File: MUENCHLOAD_UPD.SCRIPT
**
** Muench_Station
** Muench_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Muench_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Muench_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update Muench_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Muench_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Muench_Station
set Marsden_Square# = NULL
where Marsden_Square# = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```



```
/*  
** File: Biomass_copy_all.script  
**  
** Created: 08-12-91 by L.-P. Kurdelski  
*/  
use SouthernOceanDB  
go  
  
if exists (select * from sysobjects where name like "Biomass_copy_all")  
    drop procedure Biomass_Copy_all  
go  
  
create procedure Biomass_copy_all  
as  
/* Fill Station and Standard Data with Biomass Data*/  
/* L.-P. Kurdelski 1991-08-12 */  
begin  
  
insert Station (Station_Id#, Cruise_Number,  
    Date_Time, Longitude, Latitude, Marsden_Square#)  
    select Biomass_Station_Id#, Cruise_Number,  
        Date_Time, Longitude, Latitude, Marsden_Square#  
from Biomass_Station  
  
insert Standard_Data (Standard_Data_Id#, Station_Id#,  
    Depth, Temperature, Salinity, Oxygen)  
select Biomass_Standard_Data_Id#, Biomass_Station_Id#,  
    Depth, Temperature, Salinity, Oxygen  
from Biomass_Standard_Data  
  
end  
go
```

Script-74

```
/*  
** created by Jens Begemann  
  
** Date: 24.6.91  
** modified : 25.6.91  
**  
** File: updatecru24s  
**  
** Description: Changes of table Cruise:  
**              selected Cruise_Numbers get new Ship_Id#  
**              changes of ships  
*/
```

```
begin transaction  
go
```

```
declare @sid int
```

```
select @sid = Ship_Id# from Ship  
       where Ship_Name like "Eвриka"
```

```
update Cruise  
set Ship_Id# = @sid where Cruise_Number# = 42  
go
```

```
if @@error != 0  
begin  
    rollback transaction  
    print "rollback"  
end  
else  
begin  
    commit transaction  
    print "commit"  
end  
go
```

```
begin transaction  
go
```

```
declare @sid int
```

```
select @sid = Ship_Id# from Ship  
       where Ship_Name like "Babchisarai"
```

```
update Cruise  
set Ship_Id# = @sid where Cruise_Number# = 2501  
go
```

```
if @@error != 0  
begin  
    rollback transaction  
    print "rollback"  
end  
else  
begin  
    commit transaction  
    print "commit"  
end  
go
```

```
begin transaction  
go
```

```
declare @sid int
```

```
select @sid = Ship_Id# from Ship  
       where Ship_Name like "Africana II"
```

```
update Cruise  
set Ship_Id# = @sid where Cruise_Number# = 5406  
go
```

```
if @@error != 0  
begin  
    rollback transaction  
    print "rollback"  
end  
else  
begin  
    commit transaction  
    print "commit"  
end  
go
```

```
/* eof updatecru24s.script */
```



```
/* Sybase script
**
** Filename : Biomass_UPD.script
**
** Author: L.-P. Kurdelski
**
** Creation date : 1991-07-09
**
** Modification History:
**
**      1991-08-01      lpk      Chlorophyll_A for values less than
**                          or equal to 0.0 set to NULL
**                          Chlorophyll A values for cruise
**                          ACS1 divided by 10
**                          (Hint by F. Brandini)
**      1991-07-15      lpk      illegal values for Temperature,
**                          Salinity etc. i.e. values which
**                          are not possible in nature in this cases
**
** Update of illegal values in the database SouthernOceanDB
** tables Biomass_Station and Biomass_Standard_Data. Illegal
** values are stated by the number -9999.0. This is updated
** to NULL.
**
*
use SouthernOceanDB
go

update Biomass_Station
set Marsden_Square# = NULL
where Marsden_Square# = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Station
set Date_Time = NULL
where Date_Time = "Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Temperature = NULL
where Temperature = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Temperature = NULL
where Temperature < -9
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Salinity = NULL
where Salinity = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Salinity = NULL
where Salinity < 0
```

```
go
dump tran SouthernOceanDB with truncate_only
go

/*
** Hint from Mark Thorley, Biomass Data Centre
**
** If there are Salinity values below 32 and above 0
** and it seems that they are divided by factor 10
** multiply them by 10 and it is ok.
*/
update Biomass_Standard_Data
set Salinity = Salinity * 10
where Salinity < 30
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Oxygen = NULL
where Oxygen = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Oxygen = NULL
where Oxygen < 0
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Nitrite = NULL
where Nitrite = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Nitrite = NULL
where Nitrite < 0
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Nitrate = NULL
where Nitrate = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Nitrate = NULL
where Nitrate < 0
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Ammonium = NULL
where Ammonium = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```



```
update Biomass_Standard_Data
set Ammonium = NULL
where Ammonium < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Silicate = NULL
where Silicate = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Silicate = NULL
where Silicate < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Ph = NULL
where Ph = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Ph = NULL
where Ph < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Chloride = NULL
where Chloride < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Chlorophyll_A = NULL
where Chlorophyll_A = -9999
   or Chlorophyll_A = 0.0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Chlorophyll_A = Chlorophyll_A / 10.0
where Biomass_Station_Id# in
   (select Biomass_Station_Id#
    from Biomass_Station
    where Cruise_Name like "ACS1")
go
```

```
update Biomass_Standard_Data
set Chlorophyll_A = NULL
where Chlorophyll_A < 0
go
dump tran SouthernOceanDB with truncate_only
go
```



```
update Biomass_Standard_Data
set Sigma_T = NULL
where Sigma_T = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sigma_T = NULL
where Sigma_T < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sigma_T_Obs = NULL
where Sigma_T_Obs = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sigma_T_Obs = NULL
where Sigma_T_Obs < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sound_Velocity = NULL
where Sound_Velocity = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sound_Velocity = NULL
where Sound_Velocity < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Dynamic_Height = NULL
where Dynamic_Height = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sigma_Theta = NULL
where Sigma_Theta = -9999
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Sigma_Theta = NULL
where Sigma_Theta < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Biomass_Standard_Data
set Pot_Temp = NULL
where Pot_Temp = -9999
```

```
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Pot_Temp = NULL
where Pot_Temp < -9 /* !!!!! see log from 1991-07-15 */
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Spec_Vol_Anomaly = NULL
where Spec_Vol_Anomaly = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set Delta_D = NULL
where Delta_D = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set InPhos = NULL
where InPhos = -9999
go
dump tran SouthernOceanDB with truncate_only
go

update Biomass_Standard_Data
set InPhos = NULL
where InPhos < 0
go
dump tran SouthernOceanDB with truncate_only
go

/* end of biomass_upd.script */
```

```
/*
** SYBASE script
**
** FILE: add_ships_jare140491.script
**
** CREATION: 1991-08-19 lpk
**
** MODIFIED:
**
** 1991-08-20 lpk table tempships is only truncated
**
** Two ships and eight cruises should be inserted into ship and cruise
** tables. First truncate table temptab then insert the ships into
** temptab. Last use procedure insert_new_ships to insert ships into
** cruise table.
*/
if not exists (select * from sysobjects where name like "SouthernOceanDB")
begin
    print "SouthernOceanDB not found"
    return
end
go

use SouthernOceanDB
go

if not exists (select * from sysobjects where name like "insert_into_cruise")
begin
    print "procedure insert_into_cruise expected"
    return
end
go

if not exists (select * from sysobjects where name like "insert_new_ships")
begin
    print "procedure insert_new_ships expected"
    return
end
go

if not exists (select * from sysobjects where name like "temptab")
begin
    print "Table tempships does not exist."
    return
end
go

/*
** clean up tempships
*/
truncate table tempships
go

begin transaction
go

/*
** filling tempships with new ships
*/

declare @zaehler int

select @zaehler = 1
insert tempships values(@zaehler, -25026, "Fuji", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25027, "Fuji", "Japan")
```



```
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25028, "Shirase", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25029, "Shirase", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25030, "Shirase", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25031, "Shirase", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25032, "Shirase", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, -25033, "Shirase", "Japan")
go
```

```
exec insert_new_ships
go
```

```
if @@error != 0
begin
print "transaction aborted"
rollback transaction
end
else
begin
print "transaction successfull"
commit transaction
end
go
```

```
truncate table tempships
go
/* eof add_ships_jare140491.script */
```

```
/*
** SYBASE script
**
** Creator: L.-P. Kurdelski
**
** Creation: 1991-09-02
**
** FILE: GSB_UPD.SCRIPT
**
** Modifying the date values in the table Gordon_Station_Backfill
** for some Station according to the test done by V. Gurätzky on
** 1990-11-22.
** One station with missing Longitude is also changed to the
** correct value.
**
** THIS SCRIPT IS ONLY PRESENTED FOR DOCUMENTATION PURPOSES.
** DO NOT RUN THIS SCRIPT IF YOU DO NOT REALLY KNOW WHAT YOU ARE DOING.
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "Gordon_Station_Backfill")
begin
print "Table Gordon_Station_Backfill not found."
return
end
go

update Station
set Latitude = -67.241997
where Station_Id# = 5731
go

update Gordon_Station_Backfill
set Date_Time = "Jan 12 1963 12:00AM"
where Gordon_Station_Id# = 1411
go

update Gordon_Station_Backfill
set Date_Time = "Jan 22 1963 12:00AM"
where Gordon_Station_Id# = 1464
go

update Gordon_Station_Backfill
set Date_Time = "Feb 14 1975 12:00AM"
where Gordon_Station_Id# = 4092
go
```

```
/*
**      Dieses Script laedt die Cruise_Number aus der Tabelle
**      Aari_Cruise_BCK sowie die entsprechende Ship_Id# aus
**      der Tabelle Ship in die Tabelle Cruise.
**
**      FILE: aari_and_ship_to_cruise.script
**
**      creation date: 1991-08-21
**
**      author: D. Hinz
**      but it really works
**/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "aari_and_ship_to_cruise")
    drop proc aari_and_ship_to_cruise
go

create proc aari_and_ship_to_cruise
as
    insert into Cruise
    select Ship_Id#, Cruise_Number
    from Ship, Aari_Cruise
    where Ship.Ship_Name like Aari_Cruise.Ship
go
```



```
/*
** SYBASE script
**
** Creation: 1991-09-09
**
** Creator: L.-P. Kurdelski
**
** FILE: AARI_CRUISE_VIEW.SCRIPT
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "Aari_Cruise2")
    drop view Aari_Cruise2
go

create view Aari_Cruise2 (Cruise_Number, Ship, Country)
as
    select c.Cruise_Number#, s.Ship_Name, s.Country
    from Ship s, Cruise c
    where s.Ship_Id# = c.Ship_Id#
go

/* grant or revoke access rights
*/
grant select on Aari_Cruise2 to SOCEAN
go
```

```
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Cruises to Cruise and Ship
** from Kuropatkin Data
**
** created 1991-08-28
**
** FILE: ADD_KUROPATKIN_SHIPS.SCRIPT
**/
```

```
use master
go
if not exists (select * from master.dbo.sysdatabases
              where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
```

```
use SouthernOceanDB
go
```

```
exec insertShip 'N.Kuropatkin', 'USSR', -23011
go
```

```
91-06-26
1990-12-12
/* Sybase-Script made in December 1990, 1990-12-12 by
** Lutz-Peter Kurdelski
** AWI-Rechnergruppe
**
** Erweiterung der Tabelle Ship um die Spalte
** Ship_Code
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
**
** FILE: ALTER_TAB_SHIP.SCRIPT
*/
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go

u SouthernOceanDB
go

alter table Ship
  add Ship_Code varchar(8) null
go
```



```
/*
** SYBASE script
**
** Created: 1991-08-29
**
** Creator: L.-P. Kurdelski
**
** FILE: changeShipNameInCruise.script
**
** Change the ship ids in table Cruise due to the change
** of name from old ship to new ship.
**
** An old ship name, a new ship name and two cruise number are
** used as parameters. If the second number is 0 then only the
** first number is used (just as you expect). Otherwise both
** numbers are used in a junction.
**
** If both cruise numbers are 0 it is ok.
** if the first cruise number is 0 and the second is not 0 this
** is an error.
**
** Summary
**      cru1      cru2      comment
**      0          0        change all ship_id for the old ship
**      !=0        0        ... only for cruise cru1
**      !=0        !=0      ... only for cruise cru1 and cru2
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "changeShipNameInCruise")
drop proc changeShipNameInCruise
go

create proc changeShipNameInCruise
    @oldShip varchar(40),
    @newShip varchar(40),
    @cru1 int,
    @cru2 int

as
begin
/*
* Change the ship ids in table Cruise due to the change
** of name from old ship to new ship.
**
** An old ship name, a new ship name and two cruise number are
** used as parameters. If the second number is 0 then only the
** first number is used (just as you expect). Otherwise both
** numbers are used in a junction.
**
** If both cruise numbers are 0 it is ok.
** if the first cruise number is 0 and the second is not 0 this
** is an error.
**
** Summary
**      cru1      cru2      comment
**      0          0        change all ship_id for the old ship
**      !=0        0        ... only for cruise cru1
**      !=0        !=0      ... only for cruise cru1 and cru2
**      0          !=0      illegal
**
*/
declare @sid int,
        @sidalt int,
        @msg varchar(80)
```

```

begin transaction

select @sidalt = Ship_Id#
from Ship
where Ship_Name like @oldShip

select @sid = Ship_Id#
from Ship
where Ship_Name like @newShip

if @sid is null
begin
select @sid = Ship_Id#
from Ship
where Ship_Name like "unknown"

select @msg = "Unknown ship inserted for " + @newShip
print @msg
end

if @cru2 = 0
begin
if @cru1 = 0
begin
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt
end
else
begin
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and
Cruise_Number# = @cru1
end
end
else
begin
if @cru1 = 0
begin
print "ERROR: cru1 = 0 and cru2 != 0"
rollback transaction
return
end
else
begin
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and
(Cruise_Number# = @cru1 or Cruise_Number# = @cru2)
end
end

if @@error != 0
begin
rollback transaction
select @msg = "ERROR occured during update of " + @oldShip
end
else
begin
commit transaction
select @msg = "Update of " + @oldShip + " to " + @newShip +
" successful."
end
print @msg

```

```
end  
go  
/* end of changeShipNameInCruise.script */
```


Script-03

```
/*  
** FILE checkGarbageStandardData.script  
*/  
create procedure checkGarbageStandardData  
as
```

```
select Station_Id#  
from Standard_Data  
where Station_Id# not in  
  (select Station.Station_Id# from Station)
```

29.8.91

vaf. Script-12

Script-84

```
/*  
** SYBASE script  
**  
** Created: 1991-09-03  
**  
** Creator: L.-P. Kurdelski  
**  
** FILE: copy_gordon_ship_to_ship.script  
**  
*/  
use SouthernOceanDB  
go  
  
exec copy_gordon_ship_to_ship  
go  
  
exec copy_gordon_cruise_to_cruise  
go
```

Script-85

```
/*
** SYBASE script
**
** Created: 1991-08-28
**
** Creator: L.-P. Kurdelski
**
** FILE: copy_gordon_ship_to_ship_proc.script
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "copy_gordon_ship_to_ship")
drop proc copy_gordon_ship_to_ship
go

if exists (select * from sysobjects where name like
"copy_gordon_cruise_to_cruise")
drop proc copy_gordon_cruise_to_cruise
go

create procedure copy_gordon_ship_to_ship
as
/* Copy the Gordon_Ships and relations to the table Ship */
begin
insert into Ship (Ship Id#, Ship_Name, Country, Ship_Code)
select s.Ship_Id#, s.Ship_Name, s.Country, s.Ship_Code
from Gordon_Ship s
end
go

create procedure copy_gordon_cruise_to_cruise
as
/* Copy the Gordon_Ships and relations to the table Ship */
begin
insert into Cruise (Ship Id#, Cruise Number#)
select s.Ship_Id#, c.Gordon_Cruise_Id#
from Gordon_Ship s, Gordon_Cruise c
where s.Ship_Code like c.Ship_Code
end
go

/* eof copy_gordon_ship_to_ship_proc.script */
```


Script - 86

```
/* Sybase-Script made in December 1990 by Lutz-Peter Kurdelski  
**   AWI-Rechnergruppe  
**  
** FILE: DROP_CRUISE_TRIGGER.SCRIPT  
*/
```

30.8.91

```
use master  
go  
/* Erzeugen der Datenbank  
**   Die Datenbank muss nicht mehr erzeugt werden.  
*/  
if not exists (select * from master.dbo.sysdatabases  
               where name = "SouthernOceanDB")  
begin  
print 'SouthernOceanDB does not exist !!!'  
return  
end  
  
use SouthernOceanDB  
go  
  
drop trigger Insert_Cruise on Cruise_Table  
go  
  
drop trigger Delete_Ship on Ship_Table  
go
```

Script-87

```
/*  
** FILENAME: TRIGGER_UPDATE_LOST_STATIONS.SCRIPT
```

```
**  
** DATE: 11-16-90
```

12.9.91

```
*/  
/* created by L.-P. Kurdelski, M. Reinke */  
/* AWI Database Group */
```

```
/* GarbageCollector to remove garbage from *Standard Data */  
/* if Stations were removed and the trigger was not installed. */  
/* I.e. Station were deleted without notifying StandardData. */
```

```
use master  
go
```

```
if not exists ( select * from master.dbo.sysdatabases  
                where name = "SouthernOceanDB")
```

```
begin  
    print 'SouthernOceanDB does not exist'  
    return  
end  
go
```

```
use SouthernOceanDB  
go
```

```
if exists (select *  
           from sysobjects  
           where name = "garbageCollectStandardData")
```

```
begin  
    drop procedure garbageCollectStandardData  
end  
go
```

```
create procedure garbageCollectStandardData as  
begin  
    begin transaction
```

```
create table #temp ( Station_Id int )  
go
```

```
select Station_Id#  
into #temp  
from Aari_Station  
where Station_Id# not in  
    ( select Station.Station_Id# from Station  
      where Station.Station_Id# < 100000 )
```

```
delete Standard_Data  
from Standard_Data, #temp  
where Standard_Data.Station_Id# = #temp.Station_Id#  
and not exists ( select Station.Station_Id# from Station  
                 where Station.Station_Id# = Standard_Data.Station_Id# )
```

```
insert DeleteStation  
select Station_Id#, getdate(), user_id()  
from #temp  
where Station_Id# not in ( select DeleteStation.Station_Id#  
                           from DeleteStation  
                           where DeleteStation.Station_Id# = #temp.Station_Id# )
```

```
if @@error = 0  
    commit transaction  
else  
    rollback transaction  
end
```

go

/* end of trigger_update_lost_stations.script */


```
/*
** SYBASE script
**
** Creator: L.-P. Kurdelski
**
** Creation: 1991-09-02
**
** FILE: HAINES_SHIPS020991.SCRIPT
**
** Adding ships and cruises due to Haines data to the
** tables Ship and Cruise. The data are read from DBTEXT1.DAT
** from V. Guretzky. Some checks marked that the cruise
** numbers must be modified by adding 1099 to the cruise number.
*/
use SouthernOceanDB
go

if not exists (select * from sysobjects where name like "Ship")
begin
    print "Table Ship not found"
    return
end
go

if not exists (select * from sysobjects where name like "Cruise")
begin
    print "Table Cruise not found"
    return
end
go

/*
** clean up tempships
*/
truncate table tempships
go

begin transaction
go

declare @zaehler int

select @zaehler = 0
/*
** 1      54001      Westwind      USA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55100, "Westwind", "USA")
/*
** 2      54002      Melville      USA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55101, "Melville", "USA")
/*
** 3      54003      Melville      USA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55102, "Melville", "USA")
/*
** 4      54004      Atlantis II   USA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55103, "Atlantis II", "USA")
/*
** 5      54005      Thomas Washington USA
*/
```

```
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55104, "Thomas Washington", "USA")
/*
** 6      54006      Meiring Naude      RSA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55105, "Meiring Naude", "RSA")
/*
** 7      54007      Discovery II      UK
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55106, "Discovery II", "U.K.")
/*
** 8      54008      Knorr      USA
*/
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 55107, "Knorr", "USA")
go

exec insert_new_ships
go

if @@error != 0
begin
print "transaction aborted"
rollback transaction
end
else
begin
print "Ships inserted"
commit transaction
end
go

truncate table tempships
go
/* eof of Haines_Ships020991.script
```



```
/*
** SYBASE Script by L.-P. Kurdelski
**
** FILE: NOWLIN_SHIPS.SCRIPT
**
** CREATED: 1991-08-30
**
** MODIFIED:
**
** Insert new ships
** The ships are taken from the Nowlin dataset.
*/
if not exists (select * from sysobjects where name like "SouthernOceanDB")
begin
    print "SouthernOceanDB not found"
    return
end
go

use SouthernOceanDB
go

if not exists (select * from sysobjects where name like "insert_into_cruise")
begin
    print "Procedure inser_into_cruise missing."
    return
end
go

if not exists (select * from sysobjects where name like "insert_new_ships")
begin
    print "Procedure insert_new_ships missing."
    return
end
go

if exists (select * from sysobjects where name like "tempships")
begin
    truncate table tempships
end
else
begin
    print "No table tempships"
    return
end
go

declare @zaehler int

select @zaehler = 0
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54000, "Atlantis II", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54001, "Knorr", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54002, "AGO 8", "Argentina")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54003, "Atka", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54004, "Discovery II", "U.K.")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54005, "AHZ49", "Japan")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54006, "Melville", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54007, "A5N31", "USA")
```



```
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54008, "Hudson", "Canada")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54009, "Islas Orcadas", "Argentina")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54010, "Eltanin", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54011, "Thomas Washington", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54012, "Glacier", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54013, "Discovery II", "U.K.")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54014, "AEL31", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54015, "R.D.Conrad", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54016, "AVZ90", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54017, "ASB90", "USSR")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54018, "AIS31", "USA")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54019, "AAQ06", "FRG")
select @zaehler = @zaehler + 1
insert tempships values(@zaehler, 54020, "AMT06", "FRG")
go
```

```
begin transaction
go
```

```
exec insert_new_ships
go
```

```
if @@error != 0
begin
    print 'transaction aborted'
    rollback transaction
end
else
begin
    print 'transaction successfull'
    commit transaction
end
```

```
9
truncate table tempships
print "table tempships truncated"
go
```

```
/* end of NOWLIN_SHIPS.SCRIPT */
```

Script - 90

```
/*
** SYBASE script
**
** FILE: unknown_ship.script
**
** CREATION: 1991-08-19 lpk
**
** Insert the "unknown" ship into the ship table
*/
if not exists (select * from sysobjects where name like "SouthernOceanDB")
begin
    print "SouthernOceanDB not found"
    return
end
go

use SouthernOceanDB
go

insert Ship
values (1000000,"unknown","unknown",NULL)
go
```


Script-91

```
/*
** Sybase script
**
** Written 1991-06-11 by Jens Begemann and L.-P. Kurdelski
**
** Modified 1991-08-29 by D. Hinz and L.-P. Kurdelski
** ( uebersichtlicher!)
**
**      old version: FILE UpdateShipCruise100691.script
**      new version: FILE UpdateShipCruise290891.script
**
** Update of Ship and Cruise in SouthernOceanDB
**
** Data from V. Guretzky
**
*/

/* assume SouthernOceanDB is available */
use SouthernOceanDB
go

exec changeShipNameInCruise "Prof. Zubov", "Investigator", 643, 0
go

exec changeShipNameInCruise "Prof. Zubov", "Gascoyne", 424, 0
go

exec changeShipNameInCruise "Prof. Zubov", "Melville", 12031, -21814
go

exec changeShipNameInCruise "Prof. Zubov", "Edisto", 3482, 0
go

exec changeShipNameInCruise "Prof. Zubov", "Hudson", 6468, 0
go

exec changeShipNameInCruise "Prof. Zubov", "Carnegie", 1393, 0
go

exec changeShipNameInCruise "Prof. Zubov", "Hakuho_Maru", 5547, 0
go

exec changeShipNameInCruise "Kokar", "Ob", 0, 0
go

exec changeShipNameInCruise "Lesnoi", "Eltanin", -667, 0
go

exec changeShipNameInCruise "Walter Herwig", "Discovery II", -1123, 0
go

exec changeShipNameInCruise "Dolphin", "Cap. Canepa", -1809, 0
go

exec changeShipNameInCruise "Atlantis", "Atlantis II", -180, 0
go

exec changeShipNameInCruise "Schwabenland", "Discovery II", -12, 0
go

exec changeShipNameInCruise "Kvant", "Salehard", -1117, 0
go

declare @sid int,
        @sidalt int
select @sid = Ship_Id# from Ship where Ship_Name like "Gen. San-Martin"
```



```
select @sidalt = Ship_Id# from Ship
      where Ship_Name like "unknown" and
            Ship_Id# in (select Ship_Id# from Cruise
                        where Cruise_Number# = -1929)
```

```
update Cruise
set Ship_Id# = @sid
where Ship_Id# = @sidalt and Cruise_Number# = -1929
go
```

```
exec changeShipNameInCruise "Kiwi", "Tui", -21649, 0
go
```

```
exec changeShipNameInCruise "Burton Island", "Anton Bruun", -816, 0
go
```

```
exec changeShipNameInCruise "Africana II", "Natal", 5406, 0
go
```

```
exec changeShipNameInCruise "Poceidon", "Umitaka-Maru", -1934, 0
go
```

```
exec changeShipNameInCruise "Mitory-Maru", "Eltanin", -1677, 0
go
```

```
exec changeShipNameInCruise "Atlantis II", "Atka", -180, 0
go
```

```
exec changeShipNameInCruise "Discovery II", "Discovery", -21137, 0
go
```

```
exec changeShipNameInCruise "Glacier", "Burton Island", 619, 0
go
```

```
exec changeShipNameInCruise "Adm. Moucheg", "Norvegia", -97, 0
go
```

```
exec changeShipNameInCruise "Olonets", "Diamantina", 430, 0
go
```

```
exec changeShipNameInCruise "Evrice", "Vityaz", 42, 0
go
```

```
exec changeShipNameInCruise "Ushakov", "Slava", 9768, 0
go
```

```
exec changeShipNameInCruise "Billie", "Discovery", 54007, 0
go
```

```
/* eof UpdateShipCruise290891.script */
```

```
/*
** SYBASE script
**
** Creation: 1991-09-06
**
** Creator: L.-P. Kurdelski
**
** FILE: UPDATE_SHIPS060991.SCRIPT
**
** Spelling corrections
*/
use SouthernOceanDB
go

exec changeShipNameInCruise "AHZ49", "Fuji-Maru", 54005, 0
go

update Ship
set Ship Code = "AHZ49"
where Ship Name like "Fuji-Maru"
and Country like "Japan"
go

delete Ship
where Ship Name like "AHZ49"
and Country like "Japan"
go

exec changeShipNameInCruise "Anton", "Anton Bruun", -21047, 0
go

delete Ship
where Ship Name like "Anton"
and Country like "USA"
go

exec changeShipNameInCruise "Cascoyne", "Gascoyne", 3014, 0
go

delete Ship
where Ship Name like "Cascoyne"
and Country like "Australia"
go

update Ship
set Ship Name = "Charoit"
where Ship Name like "Charoi"
and Country like "USSR"
go

exec changeShipNameInCruise "Fuji", "Fuji-Maru", -25027, 0
go
exec changeShipNameInCruise "Fuji", "Fuji-Maru", -25026, 0
go
exec changeShipNameInCruise "Fuji", "Fuji-Maru", -25002, 0
go
exec changeShipNameInCruise "Fuji", "Fuji-Maru", -25001, 0
go
exec changeShipNameInCruise "Fuji", "Fuji-Maru", -25000, 0
go
delete Ship
where Ship Name like "Fuji"
and Country like "Japan"
go

update Ship
```

```
set Ship Name = "Prof. Vieze"  
where Ship_Name like "Prof. Vieze"  
go
```

```
update Ship  
set Ship Name = "Ranquel"  
where Ship_Name like "Rambuel"  
and Country like "Argentina"  
go
```

```
exec changeShipNameInCruise "Ranonel", "Ranquel", 116, 0  
go
```

```
delete Ship  
where Ship_Name like "Ranonel"  
and Country like "Argentina"  
go
```

```
update Ship  
set Ship Name = "Yamana"  
where Ship_Name like "Yamaha"  
and Country like "Argentina"  
go
```

```
/ end of UPDATE_SHIPS060991.SCRIPT */
```



```
/*  
** FILENAME: UPDATESTATIONTRG.SCRIP  
*/  
  
use SouthernOceanDB  
go  
/*  
** Insert triggers for table Station and Standard Data.  
** Special solution for updating the validation flags.  
**  
** created: 1991-09-05  
** creator: l.-P. kurdelski  
*/  
drop trigger UpdateStationTgr  
go  
  
drop trigger UpdateStandard_DataTgr  
go  
  
create trigger UpdateStationTgr  
on Station for update  
as  
begin  
if update (Station_Id#)  
begin  
print " Station_Id is not allowed to be updated!! Update aborted!"  
rollback transaction  
return  
end  
if (select count(*) from deleted)=0  
begin  
print " No rows found for update! Update aborted!"  
rollback transaction  
return  
end  
if not update(Validation_Flag)  
begin  
print "Validation Flag has also to be updated for documentation purposes!! Up  
rollback transaction  
return  
end  
insert into UpdateStation  
select inserted.Station_Id#, inserted.Validation_Flag, getdate(), suser_id()  
from inserted  
e  
  
go  
  
create trigger UpdateStandard_DataTgr  
on Standard_Data for update  
as  
begin  
if update (Standard_Data_Id#)  
begin  
print " Standard_Data_Id# is not allowed to be updated!! Update aborted"  
rollback transaction  
return  
end  
if (select count(*) from deleted)=0  
begin  
print "No rows selected !! Update aborted"  
rollback transaction  
return  
end  
if not exists (select * from Station s, inserted  
where s.Station_Id#=inserted.Station_Id#)
```

```
begin
  print " No Station with the new Station_Id# exists! Update aborted!"
  rollback transaction
  return
end
if not update(Validation_Flag)
begin
  print "Validation_Flag has also to be updated for documentation purposes!! Up
  rollback transaction
  return
end
insert into UpdateData
select deleted.Standard_Data_Id#,
       deleted.Station_Id#,
       inserted.Station_Id#,
       inserted.Validation_Flag,
       getdate(),
       suser_id()
from inserted, deleted
end
go
/* end of updatestationtrg.script */
```



```
/*  
** biomass_brandini_2.script  
**
```

```
** Zum Einfuegen von Chlorophyll_A-Werten in die  
** Biomass_Standard_Data der SouthernOceanDB  
**
```

```
** Autoren: D. Hinz, J. Begemann  
** Datum: 22-Juli-1991  
*/
```

→ 5.8. letzte Änderung
& Lauf

```
use SouthernOceanDB  
go
```

```
exec insertChlorophyllA "4458",0,0.17  
go
```

```
exec insertChlorophyllA "4458",23,0.22  
go
```

```
exec insertChlorophyllA "4458",50,0.16  
go
```

```
exec insertChlorophyllA "4458",100,0.26  
go
```

```
exec insertChlorophyllA "4458",150,0.21  
go
```

```
exec insertChlorophyllA "4459",0,0.13  
go
```

```
exec insertChlorophyllA "4459",20,0.16  
go
```

```
exec insertChlorophyllA "4459",50,0.15  
go
```

```
exec insertChlorophyllA "4460",0,0.06  
go
```

```
exec insertChlorophyllA "4460",25,0.05  
go
```

```
exec insertChlorophyllA "4460",50,0.07  
go
```

```
exec insertChlorophyllA "4460",100,0.26  
go
```

```
exec insertChlorophyllA "4460",150,0.18  
go
```

```
exec insertChlorophyllA "4461",0,0.14  
go
```

```
exec insertChlorophyllA "4461",20,0.18  
go
```

```
exec insertChlorophyllA "4461",30,0.25  
go
```

```
exec insertChlorophyllA "4461",47,0.19  
go
```

```
exec insertChlorophyllA "4461",75,0.19  
go
```


exec insertChlorophyllA "4461",100,0.12
go

exec insertChlorophyllA "4461",150,0.10
go

exec insertChlorophyllA "4462",0,0.13
go

exec insertChlorophyllA "4462",50,0.15
go

exec insertChlorophyllA "4462",100,0.08
go

exec insertChlorophyllA "4462",150,0.04
go

exec insertChlorophyllA "4463",30,0.14
go

exec insertChlorophyllA "4463",39,0.21
g

exec insertChlorophyllA "4463",75,0.29
go

exec insertChlorophyllA "4463",100,0.13
go

exec insertChlorophyllA "4463",150,0.09
go

exec insertChlorophyllA "4464",0,0.49
go

exec insertChlorophyllA "4464",10,0.45
go

exec insertChlorophyllA "4464",50,0.25
go

exec insertChlorophyllA "4464",100,0.16
g

exec insertChlorophyllA "4465",0,0.45
go

exec insertChlorophyllA "4465",15,0.50
go

exec insertChlorophyllA "4465",30,0.45
go

exec insertChlorophyllA "4465",50,0.34
go

exec insertChlorophyllA "4465",100,0.20
go

exec insertChlorophyllA "4465",150,0.10
go

exec insertChlorophyllA "4466",0,0.75
go

exec insertChlorophyllA "4466",25,0.46
go

exec insertChlorophyllA "4466",50,0.17
go

exec insertChlorophyllA "4466",75,0.28
go

exec insertChlorophyllA "4466",100,0.26
go

exec insertChlorophyllA "4466",150,0.24
go

exec insertChlorophyllA "4467",0,0.24
go

exec insertChlorophyllA "4467",25,0.24
go

exec insertChlorophyllA "4467",50,0.22
go

exec insertChlorophyllA "4467",75,0.25
go

exec insertChlorophyllA "4467",100,0.21
go

exec insertChlorophyllA "4467",125,0.06
go

exec insertChlorophyllA "4467",150,0.12
go

exec insertChlorophyllA "4468",0,0.50
go

exec insertChlorophyllA "4468",25,0.25
go

exec insertChlorophyllA "4468",50,0.29
go

exec insertChlorophyllA "4468",75,0.36
go

exec insertChlorophyllA "4468",100,0.24
go

exec insertChlorophyllA "4468",125,0.14
go

exec insertChlorophyllA "4468",150,0.38
go

exec insertChlorophyllA "4469",0,0.35
go

exec insertChlorophyllA "4469",30,0.29
go

exec insertChlorophyllA "4469",50,0.15
go


```
exec insertChlorophyllA "4469",100,0.35
go
exec insertChlorophyllA "4470",0,0.02
go
exec insertChlorophyllA "4470",30,0.01
go
exec insertChlorophyllA "4470",75,0.02
go
exec insertChlorophyllA "4470",100,0.02
go
exec insertChlorophyllA "4470",150,0.02
go
exec insertChlorophyllA "4471",0,0.01
go
exec insertChlorophyllA "4471",25,0.02
go
exec insertChlorophyllA "4471",50,0.02
go
exec insertChlorophyllA "4471",75,0.01
go
exec insertChlorophyllA "4471",100,0.07
go
exec insertChlorophyllA "4471",125,0.00
go
exec insertChlorophyllA "4471",150,0.00
go
exec insertChlorophyllA "4472",0,0.02
go
exec insertChlorophyllA "4472",10,0.01
go
exec insertChlorophyllA "4472",25,0.01
go
exec insertChlorophyllA "4472",30,0.02
go
exec insertChlorophyllA "4472",50,0.10
go
exec insertChlorophyllA "4472",100,0.07
go
exec insertChlorophyllA "4472",150,0.01
go
exec insertChlorophyllA "4473",0,0.02
go
exec insertChlorophyllA "4473",20,0.01
go
```


exec insertChlorophyllA "4473",24,0.03
go

exec insertChlorophyllA "4473",30,0.03
go

exec insertChlorophyllA "4473",50,0.05
go

exec insertChlorophyllA "4473",100,0.03
go

exec insertChlorophyllA "4473",150,0.00
go

exec insertChlorophyllA "4474",0,0.10
go

exec insertChlorophyllA "4474",28,0.13
go

exec insertChlorophyllA "4474",50,0.17
go

exec insertChlorophyllA "4474",100,0.32
go

exec insertChlorophyllA "4474",150,0.31
go

exec insertChlorophyllA "4475",0,0.20
go

exec insertChlorophyllA "4475",23,0.91
go

exec insertChlorophyllA "4475",50,0.13
go

exec insertChlorophyllA "4475",100,0.32
go

exec insertChlorophyllA "4475",150,0.06
go

exec insertChlorophyllA "4476",0,0.16
go

exec insertChlorophyllA "4476",10,0.29
go

exec insertChlorophyllA "4476",20,0.26
go

exec insertChlorophyllA "4476",30,0.31
go

exec insertChlorophyllA "4476",50,0.21
go

exec insertChlorophyllA "4476",100,0.29
go

exec insertChlorophyllA "4476",150,0.26
go

exec insertChlorophyllA "4477",0,1.27
go

exec insertChlorophyllA "4477",25,0.42
go

exec insertChlorophyllA "4477",50,0.20
go

exec insertChlorophyllA "4477",100,0.20
go

exec insertChlorophyllA "4477",150,0.63
go

exec insertChlorophyllA "4478",0,0.56
go

exec insertChlorophyllA "4478",20,0.31
go

exec insertChlorophyllA "4478",25,0.39
go

exec insertChlorophyllA "4478",50,0.19
go

exec insertChlorophyllA "4478",100,0.15
go

exec insertChlorophyllA "4480",0,0.45
go

exec insertChlorophyllA "4480",21,0.54
go

exec insertChlorophyllA "4480",50,0.29
go

exec insertChlorophyllA "4480",100,0.14
go

exec insertChlorophyllA "4480",150,0.32
go

exec insertChlorophyllA "4481",0,1.00
go

exec insertChlorophyllA "4481",25,0.96
go

exec insertChlorophyllA "4481",50,0.66
go

exec insertChlorophyllA "4481",100,0.18
go

exec insertChlorophyllA "4481",150,0.08
go

exec insertChlorophyllA "4482",0,1.09
go

exec insertChlorophyllA "4482",25,0.97
go

exec insertChlorophyllA "4482",50,0.77
go

exec insertChlorophyllA "4482",75,0.47
go

exec insertChlorophyllA "4482",100,0.31
go

exec insertChlorophyllA "4483",0,1.38
go

exec insertChlorophyllA "4483",10,1.07
go

exec insertChlorophyllA "4483",20,1.69
go

exec insertChlorophyllA "4483",50,0.70
go

exec insertChlorophyllA "4483",100,0.33
go

exec insertChlorophyllA "4483",150,0.21
go

exec insertChlorophyllA "4484",0,0.88
go

exec insertChlorophyllA "4484",30,1.02
go

exec insertChlorophyllA "4484",50,0.33
go

exec insertChlorophyllA "4484",100,0.19
go

exec insertChlorophyllA "4484",150,0.14
go

exec insertChlorophyllA "4485",0,0.82
go

exec insertChlorophyllA "4485",10,0.94
go

exec insertChlorophyllA "4485",25,0.19
go

exec insertChlorophyllA "4485",30,0.14
go

exec insertChlorophyllA "4485",50,0.15
go

exec insertChlorophyllA "4485",100,0.05
go

exec insertChlorophyllA "4485",150,0.09
go

exec insertChlorophyllA "4486",0,0.53
go

exec insertChlorophyllA "4486",10,0.45
go

exec insertChlorophyllA "4486",25,0.38
go

exec insertChlorophyllA "4486",50,0.28
go

exec insertChlorophyllA "4486",75,0.30
go

exec insertChlorophyllA "4487",0,0.24
go

exec insertChlorophyllA "4487",10,0.22
go

exec insertChlorophyllA "4487",20,0.17
go

exec insertChlorophyllA "4487",30,0.17
go

exec insertChlorophyllA "4487",50,0.26
go

exec insertChlorophyllA "4487",75,0.24
go

exec insertChlorophyllA "4488",0,0.16
go

exec insertChlorophyllA "4488",25,0.09
go

exec insertChlorophyllA "4488",50,0.10
go

exec insertChlorophyllA "4488",100,0.07
go

exec insertChlorophyllA "4488",150,0.05
go

exec insertChlorophyllA "4489",0,0.08
go

exec insertChlorophyllA "4489",10,0.05
go

exec insertChlorophyllA "4489",25,0.03
go

exec insertChlorophyllA "4489",30,0.07
go

exec insertChlorophyllA "4489",50,0.07
go

exec insertChlorophyllA "4489",100,0.10
go

exec insertChlorophyllA "4489",150,0.07
go

exec insertChlorophyllA "4490",0,0.78
go

exec insertChlorophyllA "4490",25,0.70
go

exec insertChlorophyllA "4490",50,0.37
go

exec insertChlorophyllA "4490",100,0.28
go

exec insertChlorophyllA "4490",150,0.15
go

exec insertChlorophyllA "4491",0,0.14
go

exec insertChlorophyllA "4491",25,0.17
go

exec insertChlorophyllA "4491",50,0.13
go

exec insertChlorophyllA "4491",100,0.14
go

exec insertChlorophyllA "4491",150,0.09
go

exec insertChlorophyllA "4492",0,0.22
go

exec insertChlorophyllA "4492",20,0.23
go

exec insertChlorophyllA "4492",30,0.19
go

exec insertChlorophyllA "4492",50,0.20
go

exec insertChlorophyllA "4492",100,0.08
go

exec insertChlorophyllA "4492",150,0.08
go

exec insertChlorophyllA "4493",0,0.32
go

exec insertChlorophyllA "4493",25,0.36
go

exec insertChlorophyllA "4493",50,0.16
go

exec insertChlorophyllA "4493",100,0.32
go

exec insertChlorophyllA "4493",150,0.06
go

exec insertChlorophyllA "4494",0,0.62
go

exec insertChlorophyllA "4494",10,0.43
go

exec insertChlorophyllA "4494",25,0.38
go

exec insertChlorophyllA "4494",50,0.33
go

exec insertChlorophyllA "4494",75,0.70
go

exec insertChlorophyllA "4494",100,0.28
go

exec insertChlorophyllA "4494",150,0.30
go

exec insertChlorophyllA "4495",0,0.30
go

exec insertChlorophyllA "4495",50,0.33
go

exec insertChlorophyllA "4495",100,0.23
go

exec insertChlorophyllA "4495",150,0.24
go

exec insertChlorophyllA "4496",0,0.31
go

exec insertChlorophyllA "4496",20,0.15
go

exec insertChlorophyllA "4496",30,0.21
go

exec insertChlorophyllA "4496",100,0.38
go

exec insertChlorophyllA "4496",150,0.24
go

exec insertChlorophyllA "4497",25,0.21
go

exec insertChlorophyllA "4497",50,0.15
go

exec insertChlorophyllA "4497",100,0.19
go

exec insertChlorophyllA "4497",150,0,19
go

exec insertChlorophyllA "4498",0,0.18
go

exec insertChlorophyllA "4498",25,0.19
go

exec insertChlorophyllA "4498",50,0.24
go

exec insertChlorophyllA "4498",100,0.19
go

exec insertChlorophyllA "4498",150,0.20
go

exec insertChlorophyllA "4499",0,0.27
go

exec insertChlorophyllA "4499",24,0.32
go

exec insertChlorophyllA "4499",50,0.35
go

exec insertChlorophyllA "4499",100,0.48
go

exec insertChlorophyllA "4499",150,0.35
go

exec insertChlorophyllA "4500",0,0.37
go

exec insertChlorophyllA "4500",50,0.26
go

exec insertChlorophyllA "4500",100,0.24
go

exec insertChlorophyllA "4500",150,0.26
go

exec insertChlorophyllA "4501",0,0.34
go

exec insertChlorophyllA "4501",25,0.35
go

exec insertChlorophyllA "4501",50,0.21
go

exec insertChlorophyllA "4501",100,0.12
g

exec insertChlorophyllA "4501",150,0.17
go

exec insertChlorophyllA "4502",0,0.15
go

exec insertChlorophyllA "4502",25,0.17
go

exec insertChlorophyllA "4502",50,0.18
go

exec insertChlorophyllA "4502",100,0.12
go

exec insertChlorophyllA "4502",150,0.12
go

exec insertChlorophyllA "4739",0,0.23
go

exec insertChlorophyllA "4739",24,0.68
go

exec insertChlorophyllA "4739",50,0.59
go

exec insertChlorophyllA "4739",75,0.26
go

exec insertChlorophyllA "4739",100,0.80
go

exec insertChlorophyllA "4739",150,0.05
go

exec insertChlorophyllA "4740",0,0.30
go

exec insertChlorophyllA "4740",10,0.28
go

exec insertChlorophyllA "4740",20,0.26
go

exec insertChlorophyllA "4740",25,0.36
go

exec insertChlorophyllA "4740",50,0.32
go

exec insertChlorophyllA "4740",100,0.02
go

exec insertChlorophyllA "4740",150,0.01
go

exec insertChlorophyllA "4741",0,0.42
go

exec insertChlorophyllA "4741",22,0.42
go

exec insertChlorophyllA "4741",50,0.38
go

exec insertChlorophyllA "4741",75,0.25
go

exec insertChlorophyllA "4741",100,0.26
go

exec insertChlorophyllA "4741",150,0.17
go

exec insertChlorophyllA "4742",0,0.55
go

exec insertChlorophyllA "4742",10,0.55
go

exec insertChlorophyllA "4742",25,0.38
go

exec insertChlorophyllA "4742",50,0.25
go


```
exec insertChlorophyllA "4742",75,0.13
go
exec insertChlorophyllA "4742",100,0.23
go
exec insertChlorophyllA "4742",150,0.09
go
exec insertChlorophyllA "4743",0,0.34
go
exec insertChlorophyllA "4743",25,0.34
go
exec insertChlorophyllA "4743",50,0.25
go
exec insertChlorophyllA "4743",75,0.25
go
exec insertChlorophyllA "4743",100,0.25
go
exec insertChlorophyllA "4743",150,0.11
go
exec insertChlorophyllA "4744",0,0.44
go
exec insertChlorophyllA "4744",25,0.38
go
exec insertChlorophyllA "4744",50,0.25
go
exec insertChlorophyllA "4744",75,0.17
go
exec insertChlorophyllA "4744",100,0.17
go
exec insertChlorophyllA "4744",150,0.17
go
exec insertChlorophyllA "4745",0,0.63
go
exec insertChlorophyllA "4745",10,0.59
go
exec insertChlorophyllA "4745",30,0.55
go
exec insertChlorophyllA "4745",50,0.46
go
exec insertChlorophyllA "4745",75,0.55
go
exec insertChlorophyllA "4745",100,0.38
go
exec insertChlorophyllA "4745",150,0.21
go
```



```
exec insertChlorophyllA "4746",0,0.30
go
exec insertChlorophyllA "4746",25,0.32
go
exec insertChlorophyllA "4746",50,0.30
go
exec insertChlorophyllA "4746",75,0.21
go
exec insertChlorophyllA "4746",100,0.10
go
exec insertChlorophyllA "4746",150,0.06
go
exec insertChlorophyllA "4747",0,0.46
go
exec insertChlorophyllA "4747",10,0.55
go
exec insertChlorophyllA "4747",25,0.55
go
exec insertChlorophyllA "4747",50,0.55
go
exec insertChlorophyllA "4747",75,0.23
go
exec insertChlorophyllA "4747",100,0.13
go
exec insertChlorophyllA "4747",150,0.03
go
exec insertChlorophyllA "4748",0,0.19
go
exec insertChlorophyllA "4748",25,0.23
go
exec insertChlorophyllA "4748",50,0.34
go
exec insertChlorophyllA "4748",75,0.25
go
exec insertChlorophyllA "4748",100,0.23
go
exec insertChlorophyllA "4748",150,0.19
go
exec insertChlorophyllA "4749",0,0.10
go
exec insertChlorophyllA "4749",10,0.09
go
exec insertChlorophyllA "4749",23,0.07
go
```


exec insertChlorophyllA "4749",50,0.18
go

exec insertChlorophyllA "4749",75,0.15
go

exec insertChlorophyllA "4749",100,0.14
go

exec insertChlorophyllA "4749",150,0.01
go

exec insertChlorophyllA "4750",0,0.18
go

exec insertChlorophyllA "4750",10,0.18
go

exec insertChlorophyllA "4750",30,0.17
go

exec insertChlorophyllA "4750",50,0.17
go

exec insertChlorophyllA "4750",75,0.13
go

exec insertChlorophyllA "4750",100,0.05
go

exec insertChlorophyllA "4750",150,0.04
go

exec insertChlorophyllA "4751",0,0.13
go

exec insertChlorophyllA "4751",10,0.13
go

exec insertChlorophyllA "4751",23,0.12
go

exec insertChlorophyllA "4751",50,0.15
go

exec insertChlorophyllA "4751",75,0.14
go

exec insertChlorophyllA "4751",100,0.06
go

exec insertChlorophyllA "4751",150,0.01
go

exec insertChlorophyllA "4752",0,0.38
go

exec insertChlorophyllA "4752",10,0.27
go

exec insertChlorophyllA "4752",30,0.23
go

exec insertChlorophyllA "4752",50,0.21
go

exec insertChlorophyllA "4752",75,0.13
go

exec insertChlorophyllA "4752",100,0.05
go

exec insertChlorophyllA "4752",150,0.08
go

exec insertChlorophyllA "4753",0,0.75
go

exec insertChlorophyllA "4753",10,1.00
go

exec insertChlorophyllA "4753",30,2.01
go

exec insertChlorophyllA "4753",50,1.51
go

exec insertChlorophyllA "4753",75,1.51
go

exec insertChlorophyllA "4753",100,0.34
go

exec insertChlorophyllA "4753",150,0.25
go

exec insertChlorophyllA "4754",0,1.33
go

exec insertChlorophyllA "4754",10,0.82
go

exec insertChlorophyllA "4754",24,0.38
go

exec insertChlorophyllA "4754",50,0.69
go

exec insertChlorophyllA "4754",75,0.38
go

exec insertChlorophyllA "4754",100,0.17
go

exec insertChlorophyllA "4754",150,0.08
go

exec insertChlorophyllA "4755",0,0.63
go

exec insertChlorophyllA "4755",10,0.63
go

exec insertChlorophyllA "4755",25,0.50
go

exec insertChlorophyllA "4755",50,0.38
go

exec insertChlorophyllA "4755",75,0.25
go

exec insertChlorophyllA "4755",100,0.09
go

exec insertChlorophyllA "4755",150,0.07
go

exec insertChlorophyllA "4756",0,0.30
go

exec insertChlorophyllA "4756",10,0.30
go

exec insertChlorophyllA "4756",25,0.34
go

exec insertChlorophyllA "4756",50,0.44
go

exec insertChlorophyllA "4757",0,6.28
go

exec insertChlorophyllA "4757",10,5.02
go

exec insertChlorophyllA "4757",24,5.65
go

exec insertChlorophyllA "4757",50,5.02
go

exec insertChlorophyllA "4757",75,2.54
go

exec insertChlorophyllA "4757",95,1.07
go

exec insertChlorophyllA "4758",0,1.69
go

exec insertChlorophyllA "4758",10,1.48
go

exec insertChlorophyllA "4758",30,0.94
go

exec insertChlorophyllA "4758",50,0.75
go

exec insertChlorophyllA "4758",75,1.90
go

exec insertChlorophyllA "4758",100,0.12
go

exec insertChlorophyllA "4758",150,0.09
go

exec insertChlorophyllA "4759",0,0.82
go

exec insertChlorophyllA "4759",10,0.38
go

exec insertChlorophyllA "4759",30,0.63
go

exec insertChlorophyllA "4759",50,0.63
go

exec insertChlorophyllA "4759",75,0.75
go

exec insertChlorophyllA "4759",100,0.23
go

exec insertChlorophyllA "4759",150,0.13
go

exec insertChlorophyllA "4760",0,1.26
go

exec insertChlorophyllA "4760",10,1.90
go

exec insertChlorophyllA "4760",30,2.33
go

exec insertChlorophyllA "4760",50,0.56
go

exec insertChlorophyllA "4760",75,0.36
go

exec insertChlorophyllA "4760",100,0.34
go

exec insertChlorophyllA "4760",150,0.17
go

exec insertChlorophyllA "4761",0,1.19
go

exec insertChlorophyllA "4761",10,0.88
go

exec insertChlorophyllA "4761",30,1.50
go

exec insertChlorophyllA "4761",50,1.51
go

exec insertChlorophyllA "4761",75,0.88
go

exec insertChlorophyllA "4761",100,0.56
go

exec insertChlorophyllA "4761",150,0.19
go

exec insertChlorophyllA "4762",0,37.68
go

exec insertChlorophyllA "4762",10,31.40
go

exec insertChlorophyllA "4762",25,31.40
go

exec insertChlorophyllA "4762",50,10.05
go

exec insertChlorophyllA "4762",75,1.63
go

exec insertChlorophyllA "4762",100,1.26
go

exec insertChlorophyllA "4762",150,0.42
go

exec insertChlorophyllA "4763",0,0.69
go

exec insertChlorophyllA "4763",10,1.51
go

exec insertChlorophyllA "4763",20,1.13
go

exec insertChlorophyllA "4763",47,0.88
go

exec insertChlorophyllA "4763",75,0.50
go

exec insertChlorophyllA "4763",100,0.36
go

exec insertChlorophyllA "4763",150,0.09
go

exec insertChlorophyllA "4764",0,0.63
go

exec insertChlorophyllA "4764",10,0.27
go

exec insertChlorophyllA "4764",25,0.40
go

exec insertChlorophyllA "4764",50,0.30
go

exec insertChlorophyllA "4764",75,0.27
go

exec insertChlorophyllA "4764",100,0.34
go

exec insertChlorophyllA "4765",0,9.40
go

exec insertChlorophyllA "4765",10,3.38
go

exec insertChlorophyllA "4765",24,3.81
go

exec insertChlorophyllA "4765",50,4.23
go

exec insertChlorophyllA "4765",75,11.30
go

exec insertChlorophyllA "4765",100,1.00
go

exec insertChlorophyllA "4765",150,0.76
go

exec insertChlorophyllA "4766",0,26.32
go

exec insertChlorophyllA "4766",10,28.26
go

exec insertChlorophyllA "4766",25,24.32
go

exec insertChlorophyllA "4766",50,6.77
go

exec insertChlorophyllA "4766",75,2.26
go

exec insertChlorophyllA "4766",100,1.63
go

exec insertChlorophyllA "4766",150,2.01
go

exec insertChlorophyllA "4767",0,0.88
go

exec insertChlorophyllA "4767",10,0.88
go

exec insertChlorophyllA "4767",23,1.19
go

exec insertChlorophyllA "4767",50,0.38
go

exec insertChlorophyllA "4767",75,0.27
go

exec insertChlorophyllA "4767",100,0.23
go

exec insertChlorophyllA "4767",150,0.11
go

exec insertChlorophyllA "4768",0,1.19
go

exec insertChlorophyllA "4768",10,0.94
go

exec insertChlorophyllA "4768",25,0.82
go

exec insertChlorophyllA "4768",50,0.75
go

exec insertChlorophyllA "4768",75,15.13
go

exec insertChlorophyllA "4768",96,0.42
go

exec insertChlorophyllA "4770",0,2.33
go

exec insertChlorophyllA "4770",10,1.50
go

exec insertChlorophyllA "4770",25,0.69
go

exec insertChlorophyllA "4770",50,0.19
go

exec insertChlorophyllA "4770",75,0.13
go

exec insertChlorophyllA "4770",100,0.13
go

exec insertChlorophyllA "4770",150,0.09
go

exec insertChlorophyllA "4771",0,0.17
go

exec insertChlorophyllA "4771",10,0.15
go

exec insertChlorophyllA "4771",24,0.11
go

exec insertChlorophyllA "4771",50,0.22
go

exec insertChlorophyllA "4771",75,0.14
go

exec insertChlorophyllA "4771",100,0.02
go

exec insertChlorophyllA "4771",150,0.01
go

exec insertChlorophyllA "4772",0,0.34
go

exec insertChlorophyllA "4772",10,0.30
go

exec insertChlorophyllA "4772",25,0.14
go

exec insertChlorophyllA "4772",50,0.17
go

exec insertChlorophyllA "4772",75,0.15
go

exec insertChlorophyllA "4772",100,0.11
go

exec insertChlorophyllA "4772",150,0.07
go

exec insertChlorophyllA "4773",0,0.07
go

exec insertChlorophyllA "4773",10,0.07
go

exec insertChlorophyllA "4773",22,0.07
go

exec insertChlorophyllA "4773",50,0.10
go

exec insertChlorophyllA "4773",75,0.16
go

exec insertChlorophyllA "4773",100,0.10
go

exec insertChlorophyllA "4773",150,0.09
go

exec insertChlorophyllA "4774",0,0.02
go

exec insertChlorophyllA "4774",10,0.07
go

exec insertChlorophyllA "4774",24,0.07
go

exec insertChlorophyllA "4774",50,0.03
go

exec insertChlorophyllA "4774",75,0.07
go

exec insertChlorophyllA "4774",100,0.11
go

exec insertChlorophyllA "4774",150,0.09
go

exec insertChlorophyllA "4775",0,0.15
go

exec insertChlorophyllA "4775",10,0.17
go

exec insertChlorophyllA "4775",25,0.15
go

exec insertChlorophyllA "4775",48,0.21
go

exec insertChlorophyllA "4775",100,0.09
go

exec insertChlorophyllA "4775",150,0.01
go

exec insertChlorophyllA "4776",0,0.10
go

exec insertChlorophyllA "4776",10,0.12
go

exec insertChlorophyllA "4776",23,0.07
go

exec insertChlorophyllA "4776",50,0.13
go

exec insertChlorophyllA "4776",75,0.15
go

exec insertChlorophyllA "4776",100,0.13
go

exec insertChlorophyllA "4776",150,0.02
go

exec insertChlorophyllA "4777",0,0.10
go

exec insertChlorophyllA "4777",10,0.11
go

exec insertChlorophyllA "4777",25,0.08
go

exec insertChlorophyllA "4777",50,0.06
go

exec insertChlorophyllA "4777",75,0.12
go

exec insertChlorophyllA "4777",100,0.06
go

exec insertChlorophyllA "4777",150,0.01
go

exec insertChlorophyllA "4778",0,0.23
go

exec insertChlorophyllA "4778",10,0.16
go

exec insertChlorophyllA "4778",24,0.15
go

exec insertChlorophyllA "4778",50,0.17
go

exec insertChlorophyllA "4778",75,0.27
go

exec insertChlorophyllA "4778",100,0.11
go

exec insertChlorophyllA "4778",150,0.02
go

exec insertChlorophyllA "4779",0,0.10
go

exec insertChlorophyllA "4779",10,0.08
go

exec insertChlorophyllA "4779",25,0.07
go

exec insertChlorophyllA "4779",50,0.11
go

exec insertChlorophyllA "4779",75,0.15
go

exec insertChlorophyllA "4779",100,0.07
go

exec insertChlorophyllA "4779",150,0.01
go

exec insertChlorophyllA "4780",0,0.19
go

exec insertChlorophyllA "4780",10,0.17
go

exec insertChlorophyllA "4780",25,0.21
go

exec insertChlorophyllA "4780",50,0.17
go

exec insertChlorophyllA "4780",75,0.09
go

exec insertChlorophyllA "4780",100,0.05
go

exec insertChlorophyllA "4780",150,0.01
go

exec insertChlorophyllA "4781",0,0.21
go

exec insertChlorophyllA "4781",10,0.21
go

exec insertChlorophyllA "4781",24,0.27
go

exec insertChlorophyllA "4781",50,0.23
go

exec insertChlorophyllA "4781",75,0.17
go

exec insertChlorophyllA "4781",100,0.09
go

exec insertChlorophyllA "4781",150,0.05
go

exec insertChlorophyllA "4782",0,0.21
go

exec insertChlorophyllA "4782",10,0.27
go

exec insertChlorophyllA "4782",24,0.30
go

exec insertChlorophyllA "4782",50,0.21
go

exec insertChlorophyllA "4782",75,0.19
go

exec insertChlorophyllA "4782",100,0.15
go

exec insertChlorophyllA "4782",150,0.09
go

exec insertChlorophyllA "4783",0,1.00
go

exec insertChlorophyllA "4783",10,0.75
go

exec insertChlorophyllA "4783",20,0.44
go

exec insertChlorophyllA "4783",30,0.63
go

exec insertChlorophyllA "4783",50,0.44
go

exec insertChlorophyllA "4783",75,0.21
go

exec insertChlorophyllA "4783",100,0.30
go

exec insertChlorophyllA "4783",150,0.13
go

exec insertChlorophyllA "4784",0,0.15
go

exec insertChlorophyllA "4784",10,0.15
go

exec insertChlorophyllA "4784",23,0.13
go

exec insertChlorophyllA "4784",50,0.25
go

exec insertChlorophyllA "4784",75,0.10
go

exec insertChlorophyllA "4784",100,0.07
go

exec insertChlorophyllA "4784",150,0.03
go

exec insertChlorophyllA "4785",0,0.30
go

exec insertChlorophyllA "4785",10,0.88
go

exec insertChlorophyllA "4785",25,1.00
go

exec insertChlorophyllA "4785",50,0.44
go

exec insertChlorophyllA "4785",75,0.27
go

exec insertChlorophyllA "4785",100,0.15
go


```
exec insertChlorophyllA "4785",150,0.44
go
exec insertChlorophyllA "4786",0,0.50
go
exec insertChlorophyllA "4786",10,0.21
go
exec insertChlorophyllA "4786",25,0.17
go
exec insertChlorophyllA "4786",50,0.27
go
exec insertChlorophyllA "4786",75,0.17
go
exec insertChlorophyllA "4786",100,0.07
go
exec insertChlorophyllA "4786",150,0.03
gr
/* end of biomass_brandini_2.script */
```

```

/*
***++
** SYBASE script by D. Hinz, L.-P. Kurdelski
**
** FILENAME: BIOMASS_BRANDINI_UPD.SCRIPT
**
** Creation date: 1991-07-22
**
** Abstract: Inserting chlorophyll values for BIOMASS
**           from a source given by F. Brandini.
**           Preparing the insertion procedure.
**
** Modification history
**
**           1991-08-05      lpk      no insertion of duplicated stations
**                               no insertion of duplicated data
**                               update of standard_data
**__
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "insertChlorophyllA")
drop proc insertChlorophyllA

create proc insertChlorophyllA
    (@sn      varchar(8),      /*Station Number*/
     @d       int,             /*Depth*/
     @c_a     float            /*Chlorophyll_A*/
    )
as
begin
    declare
        @bsid  int,
        @bsdid int,
        @bsdidd int

    /*
    **Die Daten werden dupliziert und, mit einer neuen Kennung versehen, als
    **neuer datensatz in die Tabelle eingefuegt. Dies geschieht, da die
    **urspruenglichen Stationsdaten KEINE Chlorophyll A-Werte enthielten,
    **aber die neuen Werte als zu den alten Station gehoerig betrachtet werden.
    **zu einem spaeteren Zeitpunkt koennen diese Stationsdaten wieder geloescht
    **werden, wenn die BIOMASS-Daten in der nun gueltigen Form gehalten
    **werden sollen.
    */
    /*
    ** Waehle die Stations Id aus, die zu der gegebenen Stations_Number
    ** gehoert.
    */
begin transaction

select @bsid = Biomass_Station_Id#
from Biomass_Station
where Station_Number like @sn
and Cruise_Name like "BES%"
/*
** Wenn zu dieser Station und zur gegebenen Tiefe ein Datensatz existiert,
** dann wird dieser Datensatz aktualisiert.
*/
if exists
    (select Biomass_Standard_Data_Id#

```



```

    from Biomass_Standard_Data
    where Depth = @d
        and Biomass_Station_Id# = @bsid
        and Biomass_Standard_Data_Id# < 8800000)
begin
    select @bsdid = Biomass_Standard_Data_Id#
    from Biomass_Standard_Data
    where Depth = @d
        and Biomass_Station_Id# = @bsid
        and Biomass_Standard_Data_Id# < 8800000

    update Biomass_Standard_Data
    set Chlorophyll_A = @c_a
    where Biomass_Standard_Data_Id# = @bsdid

    print "data update successful"
end
else
begin
/*
** Ist dieser Datensatz nicht vorhanden, dann wird der Datensatz mit
** der groessten Tiefe gesucht, die kleiner ist als die gegebenen Tiefe.
*/
    select @bsdid = Biomass_Standard_Data_Id#
    from Biomass_Standard_Data
    where Depth = (select max(Depth) from Biomass_Standard_Data
                    where Biomass_Station_Id# = @bsid
                    and Depth < @d)
        and Biomass_Station_Id# = @bsid
        and Biomass_Standard_Data_Id# < 8800000
/*
** Da bereits ein Datensatz in dieser Form existieren kann,
** muss auch bei den neu eingefuehrten Datensatzen geprueft
** werden, ob die Tiefe hineinpasst.
*/
/*
** Wenn bereits ein Datensatz eingefuegt wurde,
** dann nehme dessen Nummer und addiere 1 hinzu,
** sonst nehme die urspruengliche Id und addiere
** 800000 hinzu.
** Das verfahren funktioniert, da die Tiefen in
** aufsteigender Folge angeboten werden.
*/
    if exists
        (select Biomass_Standard_Data_Id#
         from Biomass_Standard_Data
         where Depth = (select max(Depth) from Biomass_Standard_Data
                         where Biomass_Station_Id# = @bsid
                         and Depth < @d)
             and Biomass_Station_Id# = @bsid
             and Biomass_Standard_Data_Id# > 8800000)
    begin
        select @bsdidd = Biomass_Standard_Data_Id#
        from Biomass_Standard_Data
        where Depth = (select max(Depth) from Biomass_Standard_Data
                        where Biomass_Station_Id# = @bsid
                        and Depth < @d)
            and Biomass_Station_Id# = @bsid
            and Biomass_Standard_Data_Id# > 8800000

        select @bsdid = @bsdidd + 1
    end
    else
    begin
        select @bsdid = @bsdid + 800000
    end
end

```

```
/*
** Dann fuege den neuen Datensatz hinzu.
*/
insert Biomass_Standard_Data
      (Biomass_Standard_Data_Id#,
       Biomass_Station_Id#,
       Depth,
       Chlorophyll_A,
       PrDepth,
       DepthType,
       ObsType)
values(@bsdid,@bsid,@d,@c_a,@d,"-","-")

print "data insert successfull"
end
end

if @@error = 0
begin
print "transaction committed"
commit transaction
end
else
begin
print "transaction rolled back"
rollback transaction
end
go

/* end of biomass_brandini_upd.script */
```


Script-96

```
/*
** rollback_commit.script
**
** Autor: L.-P. Kurdelski
** datum: 1991-08-05
*/
use SouthernOceanDB
go

create procedure rollbackcommit
as
begin
  if @@error != 0
  begin
    rollback transaction
    print "transaction rolled back"
  end
else
  begin
    commit transaction
    print "transaction committed"
  end
end
end
g
```

```
/*  
** biomass_brandini_upd2.script  
**
```

```
** Zum Einfuegen von Chlorophyll A-Werten in die  
** Biomass_Standard_Data der SouthernOceanDB  
**
```

```
** Autoren: D. Hinz, J. Begemann  
** Datum: 22-Juli-1991  
*/
```

```
use SouthernOceanDB  
go  
exec insertChlorophyllA "4465",15,0.50  
go  
exec insertChlorophyllA "4467",25,0.24  
go  
exec insertChlorophyllA "4474",28,0.13  
go  
/* end of biomass_brandini_upd2.script */
```

~~Script-98~~

Script-98


```
/*
** SYASE Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: ShipCruiseCorrection260891.script
**
*/
use SouthernOceanDB
go

begin transaction
go

exec updateCruise 9890, "unknown", "unknown"
exec updateCruise 9768, "Ushakov", "USSR"
exec updateCruise 11669, "unknown", "unknown"
exec updateCruise 10589, "Prof. Zubov", "USSR"
exec updateCruise -22011, "Prof. Zubov", "USSR"
exec updateCruise 890, "unknown", "unknown"
exec updateCruise 619, "Glacier", "USA"
exec updateCruise 891, "unknown", "unknown"
exec updateCruise -667, "Lesnoi", "USSR"
exec updateCruise 4231, "unknown", "unknown"
exec updateCruise 10097, "unknown", "unknown"
exec updateCruise 608, "unknown", "unknown"
go

if @@error != 0
begin
print "transaction rollback"
rollback transaction
end
else
begin
print "transaction commit"
commit transaction
end
go
```

```
/*
** SYASE Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: ShipCruiseCorrection081091.script
**
*/
use SouthernOceanDB
go

begin transaction
go

exec updateCruise -22013, "Prof. Vieze", "USSR"
go

if @@error != 0
begin
print "transaction rollback"
rollback transaction
end
else
begin
print "transaction commit"
commit transaction
end
go
```


Script-101

```
/*  
** SYBASE Script  
**  
** CREATED: 1991-10-08  
**  
** CREATOR: L.-P. Kurdelski  
**  
** FILE: insertCruise081091.script  
**  
*/  
use SouthernOceanDB  
go  
  
exec insert_into_cruise 55071, "unknown", "unknown"  
go
```

Script-102

```
/*
** SYBASE Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: updateunknownship.script
**
*/
use SouthernOceanDB
go

begin transaction
go

update Cruise
set Ship_Id# = 1000000
where Ship_Id# in (select Ship_Id#
                   from Ship
                   where Ship_Name like "unknown"
                      and Country like "unknown"
                      and Ship_Id# < 1000000)

go

if @@error != 0
begin
    rollback transaction
    print "No update"
end
else
begin
    commit transaction
    print "Unknown ships updated"
end
go
```


Script - 103

```
/*
** Sybase Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: ARGENTINE.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Argentine_Station")
drop table Argentine_Station
go

create table Argentine_Station
(Argentine_Station_Id# int, /* internal identification */
Cruise_Number int, /* Argentine nc */
Station_Number int, /* Argentine ns */
Longitude float, /* Argentine ongitude */
Latitude float, /* Argentine atitude */
Date_Time datetime NULL, /* Argentine nyear nmonth nda nho */
Bottom_Depth int NULL, /* Argentine nde */
Max_Obse_Depth int NULL, /* Argentine mode */
Number_Obse int, /* Argentine hz */
Marsden_Square# int NULL /* Argentine msq */)
go

if exists (select * from sysobjects where name="Argentine_Standard_Data")
drop table Argentine_Standard_Data
go

create table Argentine_Standard_Data
(Argentine_Standard_Data_Id# int, /* internal identification */
Argentine_Station_Id# int, /* internal identification */
Depth int, /* Argentine zst */
Temperature float NULL, /* Argentine tst */
Salinity float NULL, /* Argentine sst */
Oxygen float NULL /* Argentine oxst */)
go

/* Definition der Primaerschlüssel in den Tabellen */
execute sp_primarykey Argentine_Station, Argentine_Station_Id#
execute sp_primarykey Argentine_Standard_Data, Argentine_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */
```

```
execute sp_foreignkey Argentine_Standard_Data, Argentine_Station,  
        Argentine_Station_Id#  
go  
  
/* Erzeugen der Views */  
/* Erzeugen der Prozeduren */  
  
/* Eintragen der Nutzer */  
grant select on Argentine_Station to PHYSIK1  
go  
  
grant select on Argentine_Standard_Data to PHYSIK1  
go  
  
/* eof Argentine.script */
```



```
/*
** SYBASE Script
**
** Created 1991-10-29
**
** Creator: L.-P. Kurdelski
**
** FILE: ARGENTINELOAD_COPY.SCRIPT
*/
use SouthernOceanDB
go

if @@error != 0
begin
    print 'Fehler beim Oeffnen der Datenbank.'
    return
end
go

if exists (select * from sysobjects where name like "Argentine_copy_all")
    drop procedure Argentine_copy_all
go

create procedure Argentine_copy_all
as
/* Fill Station and Standard Data with Argentine Data*/
/* L.-P. Kurdelski 1991-10-28 */
begin

insert Station (Station_Id#, Cruise_Number, Station_Number,
    Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
    Number_Obse, Marsden_Square#)
    select Argentine_Station_Id#, Cruise_Number, Station_Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#
from Argentine_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
    Depth, Temperature, Salinity, Oxygen)
select Argentine_Standard_Data_Id#, Argentine_Station_Id#,
    Depth, Temperature, Salinity, Oxygen
from Argentine_Standard_Data

end
go

execute Argentine_copy_all
go
```

```

/*
** Sybase Script
**
** CREATED: 1991-11-11
**
** CREATOR: L.-P. Kurdelski
**
** FILE: AWI.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="AWI_Station")
drop table AWI_Station
go

create table AWI_Station
(AWI_Station_Id# int,          /* internal identification */
Cruise_Number    int,        /* AWI nc */
Station_Number   int,        /* AWI ns */
Longitude        float,      /* AWI ongitude */
Latitude         float,      /* AWI atitude */
Date_Time        datetime NULL,
/* AWI nyear nmonth nda nho */
Bottom_Depth     int NULL, /* AWI nde */
Max_Obse_Depth  int NULL, /* AWI mode */
Number_Obse     int,        /* AWI nz */
Number_Standard_Levels int, /* AWI nsl */
Marsden_Square# int NULL /* AWI msq */)

go

if exists (select * from sysobjects where name="AWI_Standard_Data")
drop table AWI_Standard_Data
go

create table AWI_Standard_Data
(AWI_Standard_Data_Id# int,          /* internal identification */
AWI_Station_Id#      int,          /* internal identification */
Depth                int,          /* AWI zst */
Temperature          float NULL,   /* AWI tst */
Salinity             float NULL /* AWI sst */)

go

/* Definition der Primaerschluessel in den Tabellen */

execute sp_primarykey AWI_Station, AWI_Station_Id#
execute sp_primarykey AWI_Standard_Data, AWI_Standard_Data_Id#
go

/* Definition der Sekundaerschluessel in den Tabellen */

```



```
execute sp_foreignkey AWI_Standard_Data, AWI_Station,  
    AWI_Station_Id#  
go  
  
/* Erzeugen der Views */  
/* Erzeugen der Prozeduren */  
  
/* Eintragen der Nutzer */  
grant select on AWI_Station to PHYSIK1  
go  
  
grant select on AWI_Standard_Data to PHYSIK1  
go  
  
/* eof AWI.script */
```

```
/*
** SYBASE script
**
** CREATOR: L.-P. Kurdelski
**
** CREATIONDATE: 1991-11-06
**
** FILE: REBUILD_ARGENTINE.SCRIPT
**
** Deleting Argentine data from Station and Standard_Data
** and preparing the tables Argentine_Station and
** Argentine_Standard_Data for reload.
*/
use SouthernOceanDB
go

declare @maxs int,
        @mins int

select @maxs=max(Station_Id#),
       @mins=min(Station_Id#)
from Argentine_Station

delete
from Station
where Station_Id# between @mins and @maxs

print "Station deleted (Data delete via trigger)"
go

truncate table Argentine_Station
go

truncate table Argentine_Standard_Data
go

print "DONE"
go
```


Script - 107

```
/*
** SYBASE script
**
** Created: 1991-08-29      4.12.91
**
** Creator: L.-P. Kurdelski
**
** FILE: changeShipNameAndCountryInCruise.script
**
** Change the ship ids in table Cruise due to the change
** of name from old ship to new ship.
**
** An old ship name, a new ship name and two cruise number are
** used as parameters. If the second number is 0 then only the
** first number is used (just as you expect). Otherwise both
** numbers are used in a junction.
**
** If both cruise numbers are 0 it is ok.
** if the first cruise number is 0 and the second is not 0 this
** is an error.
**
** Summary
**      cru1      cru2      comment
**      0          0        change all ship_id for the old ship
**      !=0        0        ... only for cruise cru1
**      !=0        !=0      ... only for cruise cru1 and cru2
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects
           where name like "changeShipNameAndCountryInCru")
  drop proc changeShipNameAndCountryInCru
go

create proc changeShipNameAndCountryInCru
  @oldShip varchar(40),
  @oldCountry varchar(40),
  @newShip varchar(40),
  @newCountry varchar(40),
  @cru1 int,
  @cru2 int

as
  begin
/*
** Change the ship ids in table Cruise due to the change
** of name from old ship to new ship.
**
** An old ship name, an old country name, a new ship name,
** a new country name, and two cruise number are
** used as parameters. If the second number is 0 then only the
** first number is used (just as you expect). Otherwise both
** numbers are used in a junction.
**
** If both cruise numbers are 0 it is ok.
** if the first cruise number is 0 and the second is not 0 this
** is an error.
**
** Summary
**      cru1      cru2      comment
**      0          0        change all ship_id for the old ship
**      !=0        0        ... only for cruise cru1
**      !=0        !=0      ... only for cruise cru1 and cru2
**      0          !=0      illegal
**
*/
```

```

*/
declare @sid int,
        @sidalt int,
        @msg varchar(80)

begin transaction

select @sidalt = Ship_Id#
from Ship
where Ship_Name like @oldShip
and Country like @oldCountry

select @sid = Ship_Id#
from Ship
where Ship_Name like @newShip
and Country like @newCountry

if @sidalt is null
begin
    select @msg = "Old ship '"+@oldShip+" '"+@oldCountry+"'" not found"
    print @msg
    return
end

if @sid is null
begin
    select @sid = max(Ship_Id#)
    from Ship
    where Ship_Id# < 100000

    insert Ship values (@sid,@newShip,@newCountry,NULL)
end

if @cru2 = 0
begin
    if @cru1 = 0
    begin
        update Cruise
        set Ship_Id# = @sid
        where Ship_Id# = @sidalt
    end
    else
    begin
        update Cruise
        set Ship_Id# = @sid
        where Ship_Id# = @sidalt and
            Cruise_Number# = @cru1
    end
end
else
begin
    if @cru1 = 0
    begin
        print "ERROR: cru1 = 0 and cru2 != 0"
        rollback transaction
        return
    end
    else
    begin
        update Cruise
        set Ship_Id# = @sid
        where Ship_Id# = @sidalt and
            (Cruise_Number# = @cru1 or Cruise_Number# = @cru2)
    end
end
end

```



```
if @@error != 0
  begin
    rollback transaction
    select @msg = "ERROR occured during update of " + @oldShip
  end
else
  begin
    commit transaction
    select @msg = "Update of " + @oldShip + " to " + @newShip +
      " successful."
  end
  print @msg
end
go
/* end of changeShipNameInCruise.script */
```

Script - 108

```
/*
** Sybase Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: SCHLITZER.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="Schlitzer_Station")
drop table Schlitzer_Station
go

create table Schlitzer_Station
(Schlitzer_Station_Id# int,          /* internal identification */
Cruise_Number         int,          /* Schlitzer nc */
Station_Number        int,          /* Schlitzer ns */
Longitude             float,        /* Schlitzer ongitude */
Latitude              float,        /* Schlitzer atitude */
Date_Time             datetime NULL, /* Schlitzer nyear nmonth nda nho */
Bottom_Depth          int NULL,     /* Schlitzer nde */
Max_Obse_Depth        int NULL,     /* Schlitzer mode */
Number_Obse           int,          /* Schlitzer hz */
Marsden_Square#       int NULL     /* Schlitzer msq */)
go

if exists (select * from sysobjects where name="Schlitzer_Standard_Data")
drop table Schlitzer_Standard_Data
go

create table Schlitzer_Standard_Data
(Schlitzer_Standard_Data_Id# int,    /* internal identification */
Schlitzer_Station_Id#         int,    /* internal identification */
Depth                         int,     /* Schlitzer zst */
Temperature                   float NULL, /* Schlitzer tst */
Salinity                      float NULL, /* Schlitzer sst */
Oxygen                        float NULL, /* Schlitzer oxst */
Phosphate                     float NULL,
Silicate                      float NULL,
Nitrate                       float NULL )
go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey Schlitzer_Station, Schlitzer_Station_Id#
execute sp_primarykey Schlitzer_Standard_Data, Schlitzer_Standard_Data_Id#
go
```

```
/* Definition der Sekundaerschlüssel in den Tabellen */  
execute sp_foreignkey Schlitzer_Standard_Data, Schlitzer_Station,  
    Schlitzer_Station_Id#  
go  
/* Erzeugen der Views */  
/* Erzeugen der Prozeduren */  
/* Eintragen der Nutzer */  
grant select on Schlitzer_Station to PHYSIK1  
go  
grant select on Schlitzer_Standard_Data to PHYSIK1  
go  
/* eof Schlitzer.script */
```


Script - 109

```
/*
** SYBASE Script
**
** Created: 1991-10-09
**
** Creator: L.P. Kurdelski
**
** FILE: SchlitzerLOAD_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** Schlitzer_Station
** Schlitzer_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Phosphate = NULL
where Phosphate < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Silicate = NULL
where Silicate < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Schlitzer_Standard_Data
set Nitrate= NULL
where Nitrate< 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update Schlitzer_Station
set Marsden_Square# = NULL
where Marsden_Square# <= 0
go
dump tran SouthernOceanDB with truncate_only
go
```

Script - 110

```
/*
** SYBASE Script
**
** Created 1991-10-29
**
** Creator: L.-P. Kurdelski
**
** FILE: SchlitzerLOAD_COPY.SCRIPT
*/
use SouthernOceanDB
go

if @@error != 0
begin
    print 'Fehler beim Oeffnen der Datenbank.'
    return
end
go

if exists (select * from sysobjects where name like "Schlitzer_copy_all")
    drop procedure Schlitzer_copy_all
go

create procedure Schlitzer_copy_all
as
/* Fill Station and Standard_Data with Schlitzer Data*/
/* L.-P. Kurdelski 1991-10-28 */
begin

insert Station (Station_Id#, Cruise_Number, Station_Number,
    Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
    Number_Obse, Marsden_Square#)
    select Schlitzer_Station_Id#, Cruise_Number, Station_Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#
from Schlitzer_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
    Depth, Temperature, Salinity, Oxygen)
select Schlitzer_Standard_Data_Id#, Schlitzer_Station_Id#,
    Depth, Temperature, Salinity, Oxygen
from Schlitzer_Standard_Data

end
go

execute Schlitzer_copy_all
go
```


Script - 111

```
/*
** SYBASE script
**
** CREATOR: L.-P. Kurdelski
**
** CREATIONDATE: 1991-12-04
**
** FILE: SHIP_UPD_04121991.SCRIPT
**
** Updating ships
*/
use SouthernOceanDB
go

/*
** Change the names of two ships
*/
if exists (select * from Ship where Ship_Name like 'Dege')
    begin
        update Ship
        set Ship_Name = 'Degei'
        where Ship_Name like 'Dege'
    end
else
    print "Ship 'Dege' does not exist."
go

if exists (select * from Ship where Ship_Name like 'Diagnita')
    begin
        update Ship
        set Ship_Name = 'Diaguaita'
        where Ship_Name like 'Diagnita'
    end
else
    print "Ship 'Diagnita' does not exist."
go

/*
** Change relation between Ship and Cruise including the country.
*/
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Th. Thompson", "USA", 12220, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Degei", "Australia", 2157, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Saga", "Australia", 1078, 0
go

exec changeShipNameAndCountryInCru
    "Boa", "Argentina", "Cap. Canepa", "Argentina", 5052, 0
go

exec changeShipNameAndCountryInCru
    "Cap. Canepa", "Argentina", "Gen. San-Martin", "Argentina", -1809, 0
go

exec changeShipNameAndCountryInCru
    "Boa", "Argentina", "Cap. Canepa", "Argentina", 5054, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Cap. Canepa", "Argentina", 6525, 0
```

```
go
/*
** Inserting the ships into a temporary table "tempships"
*/
/*
** Clearing the table.
*/
truncate table tempships
go
/*
** Inserting
*/
insert tempships (Id#, cn, ship, country)
      values (1, 58001, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (2, 58002, "Bahia Blanca", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (3, 58003, "Madryn", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (4, 58803, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (5, 58804, "Yamana", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (6, 58805, "Republica", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (7, 58805, "Ranquel", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (8, 58807, "Diaguita", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (9, 58004, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (10, 58005, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (11, 58006, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (12, 58008, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (13, 58009, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (14, 58010, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (15, 58011, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (16, 58012, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (17, 58013, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (18, 58014, "Cap. Canepa", "Argentina")
```

```
go
insert tempships (Id#, cn, ship, country)
      values (19, 58015, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (20, 58016, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (21, 58017, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (22, 58018, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (23, 58019, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (24, 58020, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (25, 58021, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (26, 58022, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (27, 58024, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (28, 58026, "Gen. Zapiola", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (29, 58027, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (30, 58028, "Comodoro Augusto Lasserre", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (31, 58029, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (32, 58030, "Gen. Zapiola", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (33, 58031, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (34, 58032, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (35, 58033, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (36, 58034, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (37, 58035, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (38, 58036, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (39, 58037, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (40, 58038, "Gen. San-Martin", "Argentina")
```



```
go
insert tempships (Id#, cn, ship, country)
      values (41, 58039, "Gen. Zapiola", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (42, 58040, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (43, 58041, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (44, 58042, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (45, 58044, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (46, 58045, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (47, 58046, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (48, 58047, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (49, 58048, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (50, 58049, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (51, 58050, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (52, 58051, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (53, 58052, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (54, 58053, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (55, 58054, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (56, 58055, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (57, 58056, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (58, 58057, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (59, 58058, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (60, 58059, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (61, 58060, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (62, 58061, "Cap. Canepa", "Argentina")
```

```
go
insert tempships (Id#, cn, ship, country)
      values (63, 58063, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (64, 58064, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (65, 58065, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (66, 58066, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (67, 58067, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (68, 58068, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (69, 58069, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (70, 58070, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (71, 58071, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (72, 58073, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (73, 58074, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (74, 58075, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (75, 58076, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (76, 58078, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (77, 58079, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (78, 58080, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (79, 58081, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (80, 58084, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (81, 58085, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (82, 58089, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (83, 58100, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (84, 58101, "San Juan", "Argentina")
```

```
go
insert tempships (Id#, cn, ship, country)
      values (85, 58808, "San Luis", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (86, 58809, "Alferez Mackinlay", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (87, 58102, "Gen. Zapiola", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (88, 58103, "Goyena", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (89, 58104, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (90, 58105, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (91, 58106, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (92, 58107, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (93, 58108, "San Juan", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (94, 58109, "Islas Orcadas", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (95, 58300, "Walther Herwig", "FRG")
go
insert tempships (Id#, cn, ship, country)
      values (96, 58301, "Walther Herwig", "FRG")
go
insert tempships (Id#, cn, ship, country)
      values (97, 58302, "Walther Herwig", "FRG")
go
insert tempships (Id#, cn, ship, country)
      values (98, 58306, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (99, 58307, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (100, 58308, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (101, 58309, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (102, 58310, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (103, 58311, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (104, 58312, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (105, 58313, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (106, 58314, "unknown", "Argentina")
```



```
go
insert tempships (Id#, cn, ship, country)
      values (107, 58315, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (108, 58316, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (109, 58317, "Gen. San-Martin", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (110, 58318, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (111, 58319, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (112, 58320, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (113, 58321, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (114, 58322, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (115, 58323, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (116, 58324, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (117, 58325, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (118, 58326, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (119, 58327, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (120, 58328, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (121, 58329, "Doctor Holmberg", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (122, 58330, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (123, 58331, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (124, 58332, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (125, 58333, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (126, 58334, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (127, 58335, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (128, 58336, "Cap. Canepa", "Argentina")
```

```
go
insert tempships (Id#, cn, ship, country)
      values (129, 58337, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (130, 58338, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (131, 58339, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (132, 58340, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (133, 58341, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (134, 58342, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (135, 58343, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (136, 58344, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (137, 58345, "Cap. Canepa", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (138, 58346, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (139, 58347, "unknown", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (140, 58432, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (141, 58433, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (142, 58435, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (143, 58438, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (144, 58440, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (145, 58441, "Puerto Deseado", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (146, 58600, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (147, 58601, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (148, 58602, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (149, 58603, "Austral", "Argentina")
go
insert tempships (Id#, cn, ship, country)
      values (150, 58604, "unknown", "Argentina")
```

```
go
insert tempships (Id#, cn, ship, country)
    values (151, 58605, "unknown", "Argentina")
go

/*
** Inserting ships into the Ship and Cruise tables
*/
begin transaction
go

exec insert_new_ships
go

if @@error != 0
    begin
        rollback transaction
        print "No ship inserted"
    end
else
    begin
        commit transaction
    end
go

print "DONE"
go
/* end of ship_upd_04121991.script */
```


Script -112

```
/*
** SYBASE script
**
** CREATOR: L.-P. Kurdelski
**
** CREATIONDATE: 1991-12-05
**
** FILE: SHIP_UPD_05121991.SCRIPT
**
** Updating ships
*/
use SouthernOceanDB
go
/*
** Updating the ship cruise relation
*/
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Fuji-Maru", "Japan", -25000, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Fuji-Maru", "Japan", -25001, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Fuji-Maru", "Japan", -25002, 0
go

print "DONE"
go
/* end of ship_upd_05121991.script */
```

Script - 113

```
/*
** SYBASE script
**
** CREATOR: L.-P. Kurdelski
**
** CREATIONDATE: 1991-12-05
**
** FILE: SHIP_UPD_05121991_2.SCRIPT
**
** Updating ships
*/
use SouthernOceanDB
go
/*
** Updating the ship cruise relation
*/
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Chatyr-Dag", "USSR", 10350, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Gizhiga", "USSR", 10097, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Prof. Zubov", "USSR", -22012, 0
go
exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Evrica", "USSR", 4231, 0
go
print "DONE"
go
/* end of ship_upd_05121991_2.script */
```

Script - 114

```
/*
** Sybase script
**
** Written 1992-02-03 by L.-P. Kurdelski
**
** FILE: updateshipcruise030292.script
**
** Update of Ship and Cruise in SouthernOceanDB
**
** Data from V. Guretzky
**
*/

/* assume SouthernOceanDB is available */
use SouthernOceanDB
go

exec insertOneShip "Spectrum", "USSR"
go

exec insertOneShip "Quantum", "USSR"
go

exec changeShipNameAndCountryInCruise
    "unknown", "unknown", "Spectrum", "USSR", 11319, 0
go

exec changeShipNameAndCountryInCruise
    "unknown", "unknown", "Quantum", "USSR", 9890, 0
go

exec changeShipNameAndCountryInCruise
    "unknown", "unknown", "Skif", "USSR", 9230, 0
go

/* eof UpdateShipCruise030292.script */
```


script-115

```
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Ship by presenting the Ship_Name,
** the Country
**
** created 1991-02-03
**
** FILE: INSERTONESHIP.SCRIPT
**/

use master
go

if not exists (select * from master.dbo.sysdatabases
              where name = "SouthernOceanDB")
begin
    print 'SouthernOceanDB does not exist !!!'
    return
end
go

use SouthernOceanDB
go

if exists (select * from sysobjects where name like "insertOneShip")
    drop proc insertOneShip
go

create proc insertOneShip ( @ship varchar(80),
                          @country varchar(80))

as
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Ship by presenting the Ship_Name, the Country
**
** created 1991-02-03
**
** FILE: INSERTONESHIP.SCRIPT
**/
begin
    declare @shipid int,
            @msg varchar(80)

    select @msg = "Ship " + @ship + ", Country " + @country

    if exists (select Ship_Id#
              from Ship
              where Ship_Name like @ship
              and Country like @country)
begin
    select @msg = @msg + " already exists"
end
else
begin
    select @shipid = (select max(Ship_Id#)
                    from Ship
                    where Ship_Id# < 100000)

    if @shipid is null
        select @shipid = 1
    else
        select @shipid = @shipid + 1

    insert Ship
    values (@shipid, @ship, @country, NULL)
```

```
        select @msg = @msg + " inserted"
      end
    print @msg
  end
go
/* eof insertOneShip.script */
```

Script - 116

```
/*
** Sybase Script
**
** CREATED: 1992-02-04
**
** CREATOR: L.-P. Kurdelski
** LOAD
** FILE: BSH.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="BSH_Station")
drop table BSH_Station
go

create table BSH_Station
  (BSH_Station_Id# int,      /* internal identification */
   Cruise_Number   int,     /* BSH nc */
   Station_Number  int,     /* BSH ns */
   Longitude       float,   /* BSH ongitude */
   Latitude        float,   /* BSH atitude */
   Date_Time       datetime NULL,
                               /* BSH nyear nmonth nda nho */
   Bottom_Depth    int NULL, /* BSH nde */
   Max_Obse_Depth  int NULL, /* BSH mode */
   Number_Obse     int,     /* BSH nz */
   Marsden_Square# int NULL /* BSH msq */)

go

if exists (select * from sysobjects where name="BSH_Standard_Data")
drop table BSH_Standard_Data
go

create table BSH_Standard_Data
  (BSH_Standard_Data_Id# int,      /* internal identification */
   BSH_Station_Id#      int,      /* internal identification */
   Depth                int,      /* BSH zst */
   Temperature          float NULL, /* BSH tst */
   Salinity             float NULL, /* BSH sst */
   Oxygen               float NULL, /* BSH Oxygen */
   Phosphate            float NULL, /* BSH Phosphate */
   Silicate             float NULL, /* BSH Silicate */
   Nitrate              float NULL /* BSH Nitrate */)

go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey BSH_Station, BSH_Station_Id#
execute sp_primarykey BSH_Standard_Data, BSH_Standard_Data_Id#
go
```



```
/* Definition der Sekundaerschlüssel in den Tabellen */  
execute sp_foreignkey BSH_Standard_Data, BSH_Station,  
    BSH_Station_Id#  
go  
/* Erzeugen der Views */  
/* Erzeugen der Prozeduren */  
/* Eintragen der Nutzer */  
grant select on BSH_Station to PHYSIK1  
go  
grant select on BSH_Standard_Data to PHYSIK1  
go  
/* eof BSH.script */
```

script-117

```
/*
** SYBASE Script
**
** Created: 1992-02-04
**
** Creator: L.P. Kurdelski
**
** FILE: BSHLOAD_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** BSH_Station
** BSH_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Oxygen = NULL
where Oxygen < 0
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Phosphate = NULL
where Phosphate < 0
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Silicate= NULL
where Silicate < 0
go
dump tran SouthernOceanDB with truncate_only
go
update BSH_Standard_Data
set Nitrate = NULL
where Nitrate < 0
go
dump tran SouthernOceanDB with truncate_only
go
```

```
update BSH_Station
set Marsden_Square# = NULL
where Marsden_Square# <= 0
go
dump tran SouthernOceanDB with truncate_only
go
```


Script - 118

```
/*
** SYBASE Script
**
** Created 1992-02-04
**
** Creator: L.-P. Kurdelski
**
** FILE: BSHLOAD_COPY.SCRIPT
*/
use SouthernOceanDB
go

if @@error != 0
begin
    print 'Fehler beim Oeffnen der Datenbank.'
    return
end
go

if exists (select * from sysobjects where name like "BSH_copy_all")
    drop procedure BSH_copy_all
go

create procedure BSH_copy_all
as
/* Fill Station and Standard_Data with BSH Data*/
/* L.-P. Kurdelski 1992-02-04*/
begin

insert Station (Station_Id#, Cruise_Number, Station_Number,
    Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
    Number_Obse, Marsden_Square#)
    select BSH_Station_Id#, Cruise_Number, Station_Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#
from BSH_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
    Depth, Temperature, Salinity)
select BSH_Standard_Data_Id#, BSH_Station_Id#,
    Depth, Temperature, Salinity
from BSH_Standard_Data

end
go

execute BSH_copy_all
go
```

Script - 119

```
/* Sybase-Script made in February, 20, 1992 by
** Lutz-Peter Kurdelski
**
** FILENAME: UPDATENODC.SCRIPT
**
**   AWI-Rechnergruppe
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end
go

use SouthernOceanDB
go

create procedure updateNODC (@ship varchar(80),
                           @country varchar(80),
                           @NODC varchar(80))
as
/*
** Update of an existing or non existing NODC code.
** This code is the Ship_Code in table Ship and view
** Aari_Cruise2
**/
declare @msg varchar(80)

if exists (select * from Ship
          where Ship_Name like @ship and
                Country like @country)
begin
    update Ship
    set Ship_Code = @NODC
    where Ship_Name like @ship and
          Country like @country
end
else
begin
    select @msg = "Ship "+@ship+", "+@country+" not found."
    print @msg
end
go
/* end of updateNODC.script */
```


Script - 120

```

/*
** SYBASE script
**
** CREATOR: Lutz-Peter Kurdelski
**
** CREATION: 1991-10-29
**
** FILE: updateNODC200292.SCRIPT
**
** Changes in the Ship, "Cruise Table
** source. oth$daten:[socean.dhi]bshcru3.dat
**
*/
use SouthernOceanDB
go

update Ship
set Ship_Name = "S.F. Baird"
where Ship_Name like "S.E. Baird" and
      Country LIKE "USA"
go

create table #newships (id int,
                        nodc varchar(5),
                        cruise int,
                        ship varchar(80),
                        country varchar(80))

go

begin transaction
go

insert #newships values ( 1, "GA09", 70133, "Gascoyne", "Australia")
insert #newships values ( 2, "GA09", 70059, "Gascoyne", "Australia")
insert #newships values ( 3, "GA26", 70014, "Galathea", "Denmark")
insert #newships values ( 4, "CN31", 70002, "Carnegie", "USA")
insert #newships values ( 5, "GL31", 70591, "Glacier", "USA")
insert #newships values ( 6, "BI31", 70592, "Burton Island", "USA")
insert #newships values ( 7, "SI31", 70613, "Staten Island", "USA")
insert #newships values ( 8, "SB31", 70802, "S.F. Baird", "USA")
insert #newships values ( 9, "BI31", 71214, "Burton Island", "USA")
insert #newships values (10, "AR31", 72046, "Argo", "USA")
insert #newships values (11, "ET31", 72701, "Eltanin", "USA")
insert #newships values (12, "OC31", 72994, "Oceanographer", "USA")
insert #newships values (13, "CO35", 78348, "Coriolis", "France")
insert #newships values (14, "CO35", 78351, "Coriolis", "France")
insert #newships values (15, "SY49", 70709, "Albatross", "Japan")
insert #newships values (16, "YI49", 70865, "Koyo-Maru", "Japan")
insert #newships values (17, "K649", 71003, "Kaiyo-Maru", "Japan")
insert #newships values (18, "CY49", 71481, "Koyo-Maru", "Japan")
insert #newships values (19, "TI61", 70004, "Tui", "New Zealand")
insert #newships values (20, "TU61", 70005, "Taranui", "New Zealand")
insert #newships values (21, "TU61", 70010, "Taranui", "New Zealand")
insert #newships values (22, "TI61", 70012, "Tui", "New Zealand")
insert #newships values (23, "TU61", 70032, "Taranui", "New Zealand")
insert #newships values (24, "TU61", 70033, "Taranui", "New Zealand")
insert #newships values (25, "TU61", 70041, "Taranui", "New Zealand")
insert #newships values (26, "VT61", 70044, "Viti", "New Zealand")
insert #newships values (27, "TI61", 70951, "Tui", "New Zealand")
insert #newships values (28, "TI61", 70952, "Tui", "New Zealand")
insert #newships values (29, "PD90", 70247, "Prof. Deryugin", "USSR")
insert #newships values (30, "AM90", 70303, "Acad. Korolev", "USSR")
insert #newships values (31, "VK90", 70345, "Vojeikov", "USSR")
insert #newships values (32, "VK90", 70434, "Vojeikov", "USSR")
insert #newships values (33, "AB90", 70499, "Alba", "USSR")
insert #newships values (34, "VI90", 70862, "Vityaz", "USSR")

```



```
insert #newships values (35, "PY90", 77090, "Priboy", "USSR")
go

if @@error != 0
  rollback transaction
else
  commit transaction
go

declare @i int,
        @m int,
        @cruise int,
        @nodc varchar(5),
        @ship varchar(80),
        @country varchar(80)

select @m = max(id) from #newships
select @i = 1

while @i <= @m
begin
  select @nodc = nodc,
         @ship = ship,
         @country = country,
         @cruise = cruise
  from #newships
  where id = @i

  exec updateNODC @ship, @country, @nodc

  exec insert_into_cruise @cruise, @ship, @country

  select @i = @i + 1
end
go

drop table #newships
go
/* end of updateNODC200292.script */
```

script - 121

```
/*
** SYBASE script
**
** CREATOR: Lutz-Peter Kurdelski
**
** CREATION: 1992-03-01
**
** FILE: resetNODC200292.SCRIPT
**
** RESET
** changes in the Ship, "Cruise Table
** source. oth$daten:[socean.dhi]bshcru3.dat
**
*/
use SouthernOceanDB
go

create table #newships (id int,
                        nodc varchar(5),
                        cruise int,
                        ship varchar(80),
                        country varchar(80))

go

begin transaction
go

insert #newships values ( 1, "GA09", 70133, "Gascoyne", "Australia")
insert #newships values ( 2, "GA09", 70059, "Gascoyne", "Australia")
insert #newships values ( 3, "GA26", 70014, "Galathea", "Denmark")
insert #newships values ( 4, "CN31", 70002, "Carnegie", "USA")
insert #newships values ( 5, "GL31", 70591, "Glacier", "USA")
insert #newships values ( 6, "BI31", 70592, "Burton Island", "USA")
insert #newships values ( 7, "SI31", 70613, "Staten Island", "USA")
insert #newships values ( 8, "SB31", 70802, "S.F. Baird", "USA")
insert #newships values ( 9, "BI31", 71214, "Burton Island", "USA")
insert #newships values (10, "AR31", 72046, "Argo", "USA")
insert #newships values (11, "ET31", 72701, "Eltanin", "USA")
insert #newships values (12, "OC31", 72994, "Oceanographer", "USA")
insert #newships values (13, "CO35", 78348, "Coriolis", "France")
insert #newships values (14, "CO35", 78351, "Coriolis", "France")
insert #newships values (15, "SY49", 70709, "Albatross", "Japan")
insert #newships values (16, "CY49", 70865, "Koyo-Maru", "Japan")
insert #newships values (17, "K649", 71003, "Kaiyo-Maru", "Japan")
insert #newships values (18, "CY49", 71481, "Koyo-Maru", "Japan")
insert #newships values (19, "TI61", 70004, "Tui", "New Zealand")
insert #newships values (20, "TU61", 70005, "Taranui", "New Zealand")
insert #newships values (21, "TU61", 70010, "Taranui", "New Zealand")
insert #newships values (22, "TI61", 70012, "Tui", "New Zealand")
insert #newships values (23, "TU61", 70032, "Taranui", "New Zealand")
insert #newships values (24, "TU61", 70033, "Taranui", "New Zealand")
insert #newships values (25, "TU61", 70041, "Taranui", "New Zealand")
insert #newships values (26, "VT61", 70044, "Viti", "New Zealand")
insert #newships values (27, "TI61", 70951, "Tui", "New Zealand")
insert #newships values (28, "TI61", 70952, "Tui", "New Zealand")
insert #newships values (29, "PD90", 70247, "Prof. Deryugin", "USSR")
insert #newships values (30, "AM90", 70303, "Acad. Korolev", "USSR")
insert #newships values (31, "VK90", 70345, "Vojeikov", "USSR")
insert #newships values (32, "VK90", 70434, "Vojeikov", "USSR")
insert #newships values (33, "AB90", 70499, "Alba", "USSR")
insert #newships values (34, "VI90", 70862, "Vityaz", "USSR")
insert #newships values (35, "PY90", 77090, "Priboy", "USSR")

go

if @@error != 0
    rollback transaction
```

```
else
  commit transaction
go

declare @i int,
        @m int,
        @cruise int,
        @nodc varchar(5),
        @ship varchar(80),
        @country varchar(80)

select @m = max(id) from #newships
select @i = 1

while @i <= @m
begin
  select @nodc = nodc,
         @ship = ship,
         @country = country,
         @cruise = cruise
  from #newships
  where id = @i

  delete from Cruise where Cruise_Number# = @cruise

  select @i = @i + 1
end
go

drop table #newships
go
/* end of resetNODC200292.script */
```


script - 122

```

/*
** SYBASE script
**
** AUTHOR: Lutz-Peter Kurdelski
**
** DATE: 1992-03-05
**
** FILE: ChangeCruise050392.script
**
** SOURCE: oth$daten:[socean.text]kurdell8.note
*/

```

```

use SouthernOceanDB
go

drop trigger Update_Cruise
go

```

```

create table #tempships (id# int,
                        cruise int,
                        oldShip varchar(80),
                        newShip varchar(80),
                        oldCountry varchar(80),
                        newCountry varchar(80))

```

```

go
/*
**      Cruise  Old_Ship  New_Ship      Old_Country  New_Country
**
*/
insert #tempships values( 1, 10522, "unknown", "Foton",          "unknown", "US
SR")
insert #tempships values( 2,  4576, "unknown", "Priliv",          "unknown", "US
SR")
insert #tempships values( 3,  2970, "unknown", "Priboy NPS",        "unknown", "US
SR")/*      ***** */
insert #tempships values( 4,  9689, "unknown", "Salehard",          "unknown", "US
SR")
insert #tempships values( 5, 11319, "unknown", "Spektrum",          "unknown", "US
SR")
insert #tempships values( 6,   896, "unknown", "Prof. Zubov",        "unknown", "US
SR")
insert #tempships values( 7,  6468, "unknown", "Hudson",             "unknown", "Ca
nada")
insert #tempships values( 8,  2968, "unknown", "Okean",              "unknown", "US
SR")/*      ***** */
insert #tempships values( 9,  2501, "unknown", "Argus",              "unknown", "US
SR")
insert #tempships values(10,  4813, "unknown", "Evrica",             "unknown", "US
SR")
insert #tempships values(11,  4375, "unknown", "Lesnoi",           "unknown", "US
SR")
insert #tempships values(12,  4395, "unknown", "Prognoz",        "unknown", "US
SR")
insert #tempships values(13,  4597, "unknown", "Salehard",          "unknown", "US
SR")
insert #tempships values(14,  4564, "unknown", "Acad. Kurchatov",    "unknown", "US
SR")
insert #tempships values(15,  3070, "unknown", "Vema",             "unknown", "US
A")
insert #tempships values(16, 13968, "unknown", "Yelcho",           "unknown", "Ch
ile")
insert #tempships values(17,   202, "unknown", "Prognoz",        "unknown", "US
SR")
insert #tempships values(18,   266, "unknown", "Belogorsk",      "unknown", "US
SR")
insert #tempships values(19,  4859, "unknown", "Walther Herwig",    "unknown", "FR
G")
insert #tempships values(20,  3997, "unknown", "Dana",             "unknown", "De
nmark")
insert #tempships values(21,   891, "unknown", "Davydov",          "unknown", "US

```

```

SR")
insert #tempships values(22, 890, "unknown", "Bellinsgausen", "unknown", "US
SR")
insert #tempships values(23, 9890, "unknown", "Quantum", "unknown", "US
SR")
go

select * from #tempships
go

declare @os varchar(80),
        @oc varchar(80),
        @ns varchar(80),
        @nc varchar(80),
        @cr int,
        @id int,
        @ma int,
        @msg varchar(80)

select @id = 1
select @ma = max(id#) from #tempships

while @id <= @ma
begin
    ( select @os = oldShip,
        @oc = oldCountry,
        @ns = newShip,
        @nc = newCountry,
        @cr = cruise
    from #tempships
    where id# = @id

    select @msg = @os+" "+@oc+" - "+@ns+" "+@nc
    print @msg

    exec changeShipNameAndCountryInCru @os, @oc, @ns, @nc, @cr, 0

    select @id = @id + 1
end
go

drop table #tempships
go

```

```
/*
** SYBASE script
**
** AUTHOR: L.-P. Kurdelski
**
** DATE: 1992-03-13
**
** FILE: insertcruise130392.script
**
** SOURCE: OTH$DATEN:[SOCEAN.TEXT]KURDEL19.NOTE
**
*/
use SouthernOceanDB
go

exec insert_into_cruise 60001,"unknown", "unknown"
exec insert_into_cruise 60002, "unknown", "unknown"
exec insert_into_cruise 60003, "unknown", "unknown"
exec insert_into_cruise 60004, "unknown", "unknown"
exec insert_into_cruise 58806, "unknown", "Argentina"
exec insert_into_cruise 999, "Lena", "USSR"
go

/* end of insertcruise130392.script */
```


Script-124

```
/*
** Sybase script
**
** Written 1992-03-27 by L.-P. Kurdelski
**
** Update of Ship and Cruise in SouthernOceanDB
**
** Data from V. Guretzky, file kurdel21.note
**
** File: UpdateShipCruise270392.script
**
*/

/* assume SouthernOceanDB is available */
use SouthernOceanDB
go

exec changeCountryOfShip "Argo", "Canada", "USA", -987
go

exec changeCountryOfShip "Staten Island", "Australia", "USA", -927
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Melville", "USA", 60001, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Hakuho-Maru", "Japan", 60002, 0
go

exec changeShipNameAndCountryInCru
    "unknown", "unknown", "Hudson", "Canada", 60003, 0
go

declare @maxid int

select @maxid = max(Ship_Id#) from Ship
where Ship_Id# < 100000

select @maxid = @maxid + 1

insert Ship
values (@maxid, "T.G.Thompson", "USA", NULL)

exec changeShipNameAndCountryInCru
    "J.G.Thompson", "USA", "T.G.Thompson", "USA", 12219, 0

exec changeShipNameAndCountryInCru
    "Th. Thompson", "USA", "T.G.Thompson", "USA", 12220, 0
go
/* eof UpdateShipCruise030292.script */
```

```
/*
** SYBASE script
**
** AUTHOR: Lutz-Peter Kurdelski
**
** DATE: 1992-03-27
**
** WHY: changing a country of a ship
**
** FILE: changeCountryOfShip.script
*/
use SouthernOceanDB
go

if exists ( select * from sysobjects where name like "changeCountryOfShip")
    drop procedure changeCountryOfShip
go

create proc changeCountryOfShip
    @ship varchar(40),
    @oldCountry varchar(40),
    @newCountry varchar(40),
    @cru int

as
begin
/*
** Change the ships country
*/
    declare @sid int,
            @msg varchar(80)

    begin transaction

    select @sid = Ship_Id#
    from Cruise
    where Cruise_Number# = @cru

    if @sid is not NULL
    begin
        update Ship
        set Country = @newCountry
        where Ship_Id# = @sid

        select @msg = "Done."

        commit transaction
    end
    else
    begin
        select @msg = "Ship '"+@ship+"' not found"

        rollback transaction
    end

    print @msg
end
go
/*end of changeCountryOfShip*/
```

```
/*
** Sybase script
**
** Written 1992-04-06 by L.-P. Kurdelski
**
** Update of Ship and Cruise in SouthernOceanDB
**
** Data from V. Guretzky, file kurdel22.note
**
** File: UpdateShipCruise060492.script
**
*/

/* assume SouthernOceanDB is available */
use SouthernOceanDB
go

exec changeShipNameAndCountryInCru
"unknown", "Denmark", "Dana", "Denmark", 55018, 0
go

exec changeShipNameAndCountryInCru
"Explorer", "Scotland", "Explorer", "U.K. Denmark", -163, 0
go
by hand
```

```
exec insertShip "Polarstern", "FRG", 59001
go
exec insertShip "Polarstern", "FRG", 59002
go
exec insertShip "Polarstern", "FRG", 59003
go
exec insertShip "Polarstern", "FRG", 59004
go
exec insertShip "Polarstern", "FRG", 59005
go
exec insertShip "Polarstern", "FRG", 59006
go
/* end of updateshipcruise060492.script */
```

U.K.
by hand
U.K.

Script - 127

```
/*
** SYBASE Script
**
** Created: 1991-10-09
**
** Creator: L.P. Kurdelski
**
** FILE: ARGENTINELOAD_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** Argentine_Station
** Argentine_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Standard_Data
set Oxygen = NULL
where Oxygen <= -9.0
go
dump tran SouthernOceanDB with truncate_only
go
update Argentine_Station
set Marsden_Square# = NULL
where Marsden_Square# <= 0
go
dump tran SouthernOceanDB with truncate_only
go
```

Script - 128

```
/*
** SYBASE Script
**
** Created 1991-11-11
**
** Creator: L.-P. Kurdelski
**
** FILE: AWILOAD_COPY.SCRIPT
*/
use SouthernOceanDB
go

if @@error != 0
begin
    print 'Fehler beim Oeffnen der Datenbank.'
    return
end
go

if exists (select * from sysobjects where name like "AWI_copy_all")
    drop procedure AWI_copy_all
go

create procedure AWI_copy_all
as
/* Fill Station and Standard_Data with AWI Data*/
/* L.-P. Kurdelski 1991-11-11*/
begin

insert Station (Station_Id#, Cruise_Number, Station_Number,
    Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
    Number_Obse, Marsden_Square#)
    select AWI_Station_Id#, Cruise_Number, Station_Number,
        Date_Time, Longitude, Latitude, Bottom_Depth, Max_Obse_Depth,
        Number_Obse, Marsden_Square#
from AWI_Station

insert Standard_Data (Standard_Data_Id#, Station_Id#,
    Depth, Temperature, Salinity)
select AWI_Standard_Data_Id#, AWI_Station_Id#,
    Depth, Temperature, Salinity
from AWI_Standard_Data

end
go

execute AWI_copy_all
go
```

Script - 129

```
/*
** SYBASE Script
**
** Created: 1991-11-11
**
** Creator: L.P. Kurdelski
**
** FILE: AWILOAD_UPD.SCRIPT
**
** Script um missing Values auf NULL zu setzen
**
** AWI_Station
** AWI_Standard_Data
*/
use SouthernOceanDB
go
dump tran SouthernOceanDB with truncate_only
go
update AWI_Station
set Date_Time = NULL
where Date_Time="Jan 01 1900 00:00:00.0"
go
dump tran SouthernOceanDB with truncate_only
go
update AWI_Station
set Bottom_Depth = NULL
where Bottom_Depth = -9999
go
dump tran SouthernOceanDB with truncate_only
go
update AWI_Standard_Data
set Temperature = NULL
where Temperature < -10.0
go
dump tran SouthernOceanDB with truncate_only
go
update AWI_Standard_Data
set Salinity = NULL
where Salinity < 0
go
dump tran SouthernOceanDB with truncate_only
go
update AWI_Station
set Marsden_Square# = NULL
where Marsden_Square# <= 0
go
dump tran SouthernOceanDB with truncate_only
go
```


Script - 130

```
/*
** SYBASE script
**
** CREATED: 1991-10-09
**
** CREATOR: L.-P. Kurdelski
**
** FILE: CHANGE_VALIDATION_FLAGS.SCRIPT
**
** ABSTRACT:
**     The validation flags will be set to a default of 0 instead of NULL.
**     This is a correction to the existing data. After this correction
**     is done there will be a default bound to the column of every table
**     containing a validation flag that will be set the validation flag
**     to zero.
**
*/
use SouthernOceanDB
go

/* update Station
** set Validation_Flag = 0
** where Validation_Flag is null
** go
**
** dump tran SouthernOceanDB with no_log
** go
*/

if exists (select * from sysobjects where name like "updateValidationFlags")
    drop procedure updateValidationFlags
go

create procedure updateValidationFlags (@start int, @end int)
as
begin
    update Standard_Data
    set Validation_Flag = 0
    where Validation_Flag is null
    and Standard_Data_Id# between @start and @end

    if @@error != 0
        print "no update"
    else
        print "updated"
end
go

/* exec updateValidationFlags 1, 400000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 400001, 500000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 500001, 750000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 750001, 1000000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 1000001, 1250000
** go
```

```
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 1250001, 2000000
** go
*/
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 3000001, 3500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 3500001, 4000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 4000001, 4500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 4500001, 5000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 5000001, 5500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 5500001, 6000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 6000001, 6500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 6500001, 7000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7000001, 8000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7000001, 7500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7500001, 9000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 9000001, 9500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 9500001, 10000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 10000001, 15000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 15000001, 20000000
go
dump tran SouthernOceanDB with no_log
```

```
go
exec updateValidationFlags 20000001, 25000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 25000001, 30000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 30000001, 35000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 35000001, 40000000
go
print "DONE"
go
```


Script -131

```
/*
** Sybase-Script made in December 1990 by Lutz-Peter Kurdelski
**   AWI-Rechnergruppe
**
** FILE: cruise_trigger.script
**
** Es werden Trigger auf delete, insert, update gelegt,
** so dass nachvollziehbar ist, wer was wann geendert hat.
**
** Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB
** bereits existiert.
*/
use master
go
/* Erzeugen der Datenbank
** Die Datenbank muss nicht mehr erzeugt werden.
*/
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen von Trigger */
if exists ( select * from sysobjects where name like "Insert_Cruise" )
  drop trigger Insert_Cruise
go

create trigger Insert_Cruise
on Cruise
for insert
as
  begin
    declare @msg varchar(80),
            @sid int

    if not exists (select *
                  from Ship, inserted
                  where Ship.Ship_Id# = inserted.Ship_Id#)
    begin
      select @sid = Ship_Id# from inserted
      select @msg = "INSERT Cruise: Please insert the ship #" +
                  convert(varchar(20),@sid) + " first into Ship"
      print @msg
      rollback transaction
    end
  end
go

if exists ( select * from sysobjects where name like "Delete_Cruise" )
  drop trigger Delete_Cruise
go

create trigger Delete_Cruise
on Cruise
for delete
as
  begin
    if exists (select *
              from Station, deleted
              where Station.Cruise_Number = deleted.Cruise_Number#)
    begin
```

```

ruise_Numberprint "DELETE Cruise: There exists a station corresponding to this C♦
    rollback transaction
end
else
begin
insert into Cruise_Deleted
select deleted.Cruise_Number#,
       deleted.Ship_Id#,
       suser_id(),
       getdate()
from deleted
end
end
go

if exists ( select * from sysobjects where name like "Update_Cruise" )
drop trigger Update_Cruise
go

create trigger Update_Cruise
on Cruise
for update
as
begin
declare @msg varchar(80),
        @sid int

if not exists
(select *
 from Ship, inserted
 where Ship.Ship_Id# = inserted.Ship_Id#)
begin
select @sid = Ship_Id# from inserted
select @msg = "UPDATE Cruise: Please insert the ship #" +
             convert(varchar(20),@sid) + " first into Ship"
print @msg
rollback transaction
end
else
begin
if update(Cruise_Number#)
begin
if exists
(select *
 from Station, deleted
 where Station.Cruise_Number = deleted.Cruise_Number#)
begin
select @sid = Cruise_Number# from deleted
select @msg = "UPDATE Cruise: There exists at least one station"
             + "(" + convert(varchar(20),@sid) + ")" +
             + " corresponding to this Cruise_Number"
print @msg
rollback transaction
end
else
begin
insert into Cruise_Updated
select inserted.Cruise_Number#,
       inserted.Ship_Id#,
       deleted.Cruise_Number#,
       deleted.Ship_Id#,
       suser_id(),
       getdate()
from inserted, deleted
print "UPDATE Cruise successful"
end
end
end

```

```

        end
    end
end
go

if exists ( select * from sysobjects where name like "Delete_Ship" )
    drop trigger Delete_Ship
go

create trigger Delete_Ship
on Ship
for delete
as
begin
    declare @msg varchar(80),
            @sid int
    if exists (select *
              from Cruise, deleted
              where Cruise.Ship_Id# = deleted.Ship_Id#)
    begin
        select @sid = Ship_Id# from deleted
        select @msg = "DELETE Ship: Please delete the cruises for ship #" +
                    convert(varchar(20),@sid) + " first"
        print @msg
        rollback transaction
    end
end
go

if exists ( select * from sysobjects where name like "Update_Ship" )
    drop trigger Update_Ship
go

create trigger Update_Ship
on Ship
for update
as
begin
    insert into Ship_Updated
    select deleted.Ship_Id#,
           inserted.Ship_Name,
           inserted.Country,
           inserted.Ship_Code,
           deleted.Ship_Name,
           deleted.Country,
           deleted.Ship_Code,
           suser_id(),
           getdate()
    from inserted, deleted
end
go
/* eof cruise_trigger.script */

```



```
/*
** SYBASE script
**
** CREATED: 1991-10-09
**
** CREATOR: L.-P. Kurdelski
**
** FILE: CVF1250001.SCRIPT
**
** ABSTRACT:
**     The validation flags will be set to a default of 0 instead of NULL.
**     This is a correction to the existing data. After this correction
**     is done there will be a default bound to the column of every table
**     containing a validation flag that will be set the validation flag
**     to zero.
**
*/
use SouthernOceanDB
go

/* update Station
** set Validation_Flag = 0
** where Validation_Flag is null
** go
**
** dump tran SouthernOceanDB with no_log
** go
*/

if exists (select * from sysobjects where name like "updateValidationFlags")
    drop procedure updateValidationFlags
go

create procedure updateValidationFlags (@start int, @end int)
as
begin
    update Standard_Data
    set Validation_Flag = 0
    where Validation_Flag is null
    and Standard_Data_Id# between @start and @end

    if @@error != 0
        print "no update"
    else
        print "updated"
end
go

/* exec updateValidationFlags 1, 400000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 400001, 500000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 500001, 750000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 750001, 1000000
** go
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 1000001, 1250000
** go
```

```
** dump tran SouthernOceanDB with no_log
** go
** exec updateValidationFlags 1250001, 2000000
** go
*/
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 3000001, 3500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 3500001, 4000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 4000001, 4500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 4500001, 5000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 5000001, 5500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 5500001, 6000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 6000001, 6500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 6500001, 7000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7000001, 8000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7000001, 7500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 7500001, 9000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 9000001, 9500000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 9500001, 10000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 10000001, 15000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 15000001, 20000000
go
dump tran SouthernOceanDB with no_log
```

```
go
exec updateValidationFlags 20000001, 25000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 25000001, 30000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 30000001, 35000000
go
dump tran SouthernOceanDB with no_log
go
exec updateValidationFlags 35000001, 40000000
go
print "DONE"
go
```


Script-133

```
/*
** SYBASE script
**
** CREATION: 1991-10-10
**
** CREATOR: L.-P. Kurdelski
**
** FILE: DUMPTRANNOLOG.SCRIPT
**
** ABSTRACT:
**
**     Dumping with NO_LOG to clear transaction log
**     Dumping database to save data
**
*/
dump transaction SouthernOceanDB with no_log
go

dump database SouthernOceanDB to SouthernOceanDBDump
go
```

Script - 134

```
/* SYBASE script by L.-P. Kurdelski
**
** Inserting new Cruises to Cruise and Ship
** from Marion Dufresne Data
**
** created 1991-09-13
**
** FILE: MARION_DUFRESNE_CRUISE.SCRIPT
*/

use master
go
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

exec insertShip 'Marion Dufresne', 'France', 57001
go
exec insertShip 'Marion Dufresne', 'France', 57002
go
exec insertShip 'Marion Dufresne', 'France', 57003
go
exec insertShip 'Marion Dufresne', 'France', 57004
go
exec insertShip 'Marion Dufresne', 'France', 57005
go
exec insertShip 'Marion Dufresne', 'France', 57006
go
exec insertShip 'Marion Dufresne', 'France', 57007
go

/* end of Marion_Dufresne_Cruise.script */
```

Script-135

```
/*
** Sybase Script
**
** CREATED: 1991-10-08
**
** CREATOR: L.-P. Kurdelski
**
** FILE: VALIDATION_FLAG_DEFAULT.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

create default Validation_Flag_Default
as 0
go

exec sp_bindefault Validation_Flag_Default, "Station.Validation_Flag"
go

exec sp_bindefault Validation_Flag_Default, "Standard_Data.Validation_Flag"
go

/* eof Argentine.script */
```


Script - 136

```
/*
** SYBASE script
**
** CREATED: 1991-10-23
**
** CREATOR: L.-P. Kurdelski
**
** FILE: c_v_f.script
**
** ABSTRACT: see change_validation_flags.script
**
*/
use SouthernOceanDB
go

if exists (select * from sysobjects where name like "updateValidationFlags")
    drop procedure updateValidationFlags
go

create procedure updateValidationFlags (@start int, @end int)
as
    begin
        update Standard_Data
        set Validation_Flag = 0
        where Validation_Flag is null
            and Standard_Data_Id# between @start and @end

        if @@error != 0
            print "no update"
        else
            print "updated"
    end
go

exec updateValidationFlags 1250001, 2000000
go
dump tran SouthernOceanDB with truncate_only
go
```

Script - 137

```
/*
** Sybase Script
**
** CREATED: 1992-06-23
**
** CREATOR: L.-P. Kurdelski
**
** FILE: SO_USER.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

exec sp_addlogin "so_user", "ocean", "SouthernOceanDB"
go

use SouthernOceanDB
go

/* Eintragen der Nutzer */
exec sp_adduser so_user
go

grant select on Station to so_user
go

grant select on Standard_Data to so_user
go

/* eof so_user.script */
```

```
/*
** SYBASE script
**
** CREATOR: Lutz-Peter Kurdelski
**
** DATE: 1992-0625
**
** FILE: createBibliothek.script
**
*/
use master
go

if exists (select * from sysobjects where name like "BibliotheksDB")
    drop BibliotheksDB
go

create database BibliotheksDB
on Userdisk01 = 50, Userdisk02 = 50, Userdisk03 = 50
log on Translog01 = 10
go

if exists (select * from syslogins where name like "sbresny")
    exec sp_droplogin sbresny
go

if exists (select * from syslogins where name like "laetsch")
    exec sp_droplogin laetsch
go

if exists (select * from syslogins where name like "schuedde")
    exec sp_droplogin schuedde
go

if exists (select * from syslogins where name like "baumert")
    exec sp_droplogin baumert
go

if exists (select * from syslogins where name like "gomez")
    exec sp_droplogin gomez
go

exec sp_addlogin "gomez", "michael", BibliotheksDB
exec sp_addlogin "sbresny", "regine", BibliotheksDB
exec sp_addlogin "laetsch", "brigitte", BibliotheksDB
exec sp_addlogin "baumert", "susanne", BibliotheksDB
exec sp_addlogin "schuedde", "schuedde", BibliotheksDB
go

exec sp_changedbowner gomez
go
/* enf of createBibliothek.script*
```



```
/* SYBASE script by L.-P. Kurdelski
**
** Updating a Ship name
**
** created 1993-01-19
**
** FILE: UPDATESHIP190193.SCRIPT
**/

use master
go

if not exists (select * from master.dbo.sysdatabases
              where name = "SouthernOceanDB")
  begin
    print 'SouthernOceanDB does not exist !!!'
    return
  end
go

use SouthernOceanDB
go

update Ship
set Ship_Name = "unknown"
where Ship_Id# = 299
go
/* eof updateShip190193.script */
```

script - 160

```
/* Sybase-Script made in Dec 1993 by M. Reinke */
/* AWI-Rechnergruppe */
/*
** FILE: AariLdgo.SCRIPT
*/
/* Dieses Script geht davon aus, dass die Datenbank SouthernOceanDB */
/* bereits existiert. */
use master
go
/* Erzeugen der Datenbank */
/* Die Datenbank muss nicht mehr erzeugt werden. */
if not exists (select * from master.dbo.sysdatabases
               where name = "SouthernOceanDB")
begin
print 'SouthernOceanDB does not exist !!!'
return
end

use SouthernOceanDB
go

/* Erzeugen der Tabellen */
if exists (select * from sysobjects where name="AariLdgo_Station")
drop table AariLdgo_Station

create table AariLdgo_Station
(AariLdgo_Station_Id# int,          /* internal identification */
Cruise_Number      int,          /* AariLdgo nc */
Station_Number     int,          /* AariLdgo ns */
Longitude          float,        /* AariLdgo ongitude */
Latitude           float,        /* AariLdgo atitude */
Date_Time          datetime NULL,
                               /* AariLdgo nyear nmonth nda nho */
Bottom_Depth       int,          /* AariLdgo nde */
Max_Obse_Depth     int,          /* AariLdgo mode */
Number_Obse        int,          /* AariLdgo hz */
Marsden_Square#    int NULL /* AariLdgo msq */)

go

if exists (select * from sysobjects where name="AariLdgo_Standard_Data")
drop table AariLdgo_Standard_Data
go

create table AariLdgo_Standard_Data
(AariLdgo_Standard_Data_Id# int,    /* internal identification */
AariLdgo_Station_Id# int,          /* internal identification */
Depth int,                          /* AariLdgo zst */
Temperature float NULL,             /* AariLdgo tst */
Salinity float NULL,                /* AariLdgo sst */
Oxygen float NULL                    /* AariLdgo oxst */)

go

/* Definition der Primaerschlüssel in den Tabellen */

execute sp_primarykey AariLdgo_Station, AariLdgo_Station_Id#
execute sp_primarykey AariLdgo_Standard_Data, AariLdgo_Standard_Data_Id#
go

/* Definition der Sekundaerschlüssel in den Tabellen */

execute sp_foreignkey AariLdgo_Standard_Data, AariLdgo_Station,
AariLdgo_Station_Id#
go

/* Erzeugen der Views */
/* Erzeugen der Prozeduren */

/* Eintragen der Nutzer */
grant select on AariLdgo_Station to PHYSIK1
grant select on AariLdgo_Standard_Data to PHYSIK1
```

go

/* eof AariLdgo.script */