

```
/* Microsoft SQL Server - Scripting */
/* Database: NEWPMS */
/* Script for updating the PMS 2.0 database to PMS 2001 */
/* Only SQL Server 7.0 and later */

/* Before this script is run
- SQL Server version is 7.0 or later
- SQLServer and SQLAgent is running
- NEWPMS database created
- Options: truncate log on checkpoint + select into/bulk copy
- Must have log device = approx. 15% of datadevice size
*/

set quoted_identifier on
go

/*****
/* Check that DB version is 7.0 or later */
*****/
if exists(select * from master.dbo.sysobjects where id = object_id('master.dbo.sysusages'))
begin
    select 'SQL Version MUST be 7.0 or later for this script - Press Cancel (Red square)'
end
go

if exists(select * from master.dbo.sysobjects where id = object_id('master.dbo.sysusages'))
begin
    waitfor time '23:59:59'
end
go

/*****
/* Check that NEWPMS DB exists */
*****/
if not exists (select * from master..sysdatabases where name = 'NEWPMS')
begin
    select 'Database NEWPMS does not exist. Press Cancel (Red square) to abort this script'
end
go

if not exists (select * from master..sysdatabases where name = 'NEWPMS')
begin
    waitfor time '23:59:59'
end
go

USE NEWPMS
GO

/*****
/* Check that DB is not already version 2001 */
*****/
if exists (select * from sysobjects where id = object_id('dbo.PMSVersion_2001') and sysstat & 0xf = 3)
begin
    select 'Database seems to be PMS 2001 version already'
end
GO

if exists (select * from sysobjects where id = object_id('dbo.PMSVersion_2001') and sysstat & 0xf = 3)
begin
    waitfor time '23:59:59'
end
GO

/*****
/* Check that DB is version 2.0 */
*****/
```

```

/*****
if not exists (select * from sysobjects where id = object_id('dbo.PMSVersion_20') and sysstat & 0xf = 3)
begin
    select 'Database is an earlier version than PMS 2.0'
end
GO

if not exists (select * from sysobjects where id = object_id('dbo.PMSVersion_20') and sysstat & 0xf = 3)
begin
    waitfor time '23:59:59'
end
GO

/*****
/* Starting update
/*****
select 'Updating PMS Database to version 2001'
select 'Press Cancel (Red square) to abort this script'
go

waitfor delay '00:00:15'
go

/*****
/* Adding new country codes
/*****
insert into "PMSCountry" values ( 'Nor', 'Norway', 'NO' )
insert into "PMSCountry" values ( 'Swe', 'Sweden', 'SE' )
insert into "PMSCountry" values ( 'Fin', 'Finland', 'FI' )
insert into "PMSCountry" values ( 'Ice', 'Iceland', 'IS' )
insert into "PMSCountry" values ( 'Deu', 'Germany', 'DE' )
insert into "PMSCountry" values ( 'Irl', 'Ireland', 'IE' )
insert into "PMSCountry" values ( 'Can', 'Canada', 'CA' )
GO

/*****
/* Create dial-log table
/*****
CREATE TABLE "dbo"."PMSDialLog" (
    "StatId" "int" NOT NULL ,
    "Time" "datetime" NOT NULL ,
    "Type" "varchar" (10) NOT NULL ,
    "DialAttempts" "int" NOT NULL ,
    "Status" "varchar" (10) NOT NULL ,
    "Description" "varchar" (255) NOT NULL ,
    "Acknowledged" "tinyint" NULL
)
GO

/*****
/* Create station type table
/*****
CREATE TABLE "dbo"."PMSStationTypes" (
    "StationType" "varchar" (5) NOT NULL ,
    "TypeName" "varchar" (25) NOT NULL ,
    "TypeDescription" "varchar" (150) NOT NULL ,
    "Display" "tinyint" NOT NULL ,
    "Master" "varchar" (5) NULL ,
    "Reference" "varchar" (5) NULL ,
    "Apparatus" "varchar" (10) NULL ,
    "SampleType" "varchar" (10) NULL ,
    "ErrorType" "varchar" (10) NULL ,
    "ErrorValue" "varchar" (10) NULL ,
    "ErrorUnit" "varchar" (10) NULL ,
    "Nuclide" "varchar" (10) NULL ,

```

```

    "MeasureUnit" "varchar" (10) NULL
)
GO

insert into "PMSStationTypes" values ( '1', 'PMS-East Station', 'PMS-East station with Rados GM-tube',
    1, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '2', 'Air sampler station', 'Polish ASS-500 Airsampler',
    1, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '3', 'PMS-East in AirPMS', 'PMS-East station part of AirPMS
combined ASS-500 and PMS-East station',
    0, '5', '1', NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '4', 'ASS-500 in AirPMS', 'ASS-500 part of AirPMS combined ASS-500
and PMS-East station',
    0, '5', '2', NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '5', 'AirPMS Station', 'AirPMS combined station of ASS-500 and PMS-
East station',
    1, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '6', 'Rados station', 'Rados/ALNOR GM probe',
    1, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '7', 'Barsebäck GM probe', 'Barsebäck ring monitoring station',
    1, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( 'D', 'Default', 'Default values for stations',
    1, NULL, NULL, '1', 'A5', 'H', '15', '%', 'T-GAMMA', 'nSv/h')
GO
insert into "PMSStationTypes" values ( '0', 'Danish PMS Station', 'Danish PMS Station with Ionization
chamber',
    1, NULL, NULL, '0', NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( 'T', 'Test station', 'Station used for testing purposes',
    0, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( '0-Air', 'PMS-East in AirPMS', 'PMS-DK station part of AirPMS
combined ASS-500 and PMS-DK station',
    0, 'AirDK', '0', NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( 'ASSDK', 'ASS-500 in AirPMS', 'Danish ASS-500 part of AirPMS
combined ASS-500 and PMS-DK station',
    0, 'AirDK', '2', NULL, NULL, NULL, NULL, NULL, NULL, NULL)
GO
insert into "PMSStationTypes" values ( 'AirDK', 'AirPMS Station', 'Danish AirPMS combined station of
ASS-500 and PMS-DK station',
    1, NULL, NULL, '0', NULL, NULL, NULL, NULL, NULL, NULL)
GO

/*****
/* Create DataRecordTypes table */
*****/

CREATE TABLE "dbo"."PMSDataRecordTypes" (
    "StationType" "varchar" (5) NOT NULL ,
    "RecordType" "varchar" (10) NOT NULL ,
    "Description" "text" NULL ,
    "ApparatusCode" "varchar" (2) NULL
)
GO

insert into "PMSDataRecordTypes" values (
    '1', 'GM',
    'Reading from Rados Geiger-Müller
tube', '1' )
go

```

```
insert into "PMSDataRecordTypes" values (
  '1', 'NAI', 'Spectral readings from NaI
detector', 'P' )
go

insert into "PMSDataRecordTypes" values (
  '1', 'NAI-PARAM', 'Parameters used for the NAI
readings', NULL )
go

insert into "PMSDataRecordTypes" values (
  '1', 'RAIN', 'Reading from Rain
gauge', NULL )
go

insert into "PMSDataRecordTypes" values (
  '1', 'TEMP-ENCL', 'Temperature in enclosure (inside grey
cabinet)', NULL )
go

insert into "PMSDataRecordTypes" values (
  '1', 'TEMP-NAI', 'Temperature at the NaI detector (inside white bucket in detector
cabin)', NULL )
go

insert into "PMSDataRecordTypes" values (
  '1', 'TEMP-OUT', 'Temperature outdoor (measured inside white detector
cabin)', NULL )
go

insert into "PMSDataRecordTypes" values (
  '6', 'GM', 'Reading from Rados Geiger-Müller
tube', '1' )
go

insert into "PMSDataRecordTypes" values (
  '7', 'GM', 'Reading from Geiger-Müller
tube', '1' )
go

insert into "PMSDataRecordTypes" values (
  '0', 'GM', 'Reading from ionization-
chamber', '0' )
go

insert into "PMSDataRecordTypes" values (
  '0', 'NAI', 'Spectral readings from NaI
detector', 'P' )
go

insert into "PMSDataRecordTypes" values (
  '0', 'NAI-PARAM', 'Parameters used for the NAI
readings', NULL )
go

insert into "PMSDataRecordTypes" values (
  '0', 'RAIN', 'Reading from Rain
gauge', NULL )
go

insert into "PMSDataRecordTypes" values (
  '0', 'TEMP-NAI', 'Temperature at the NaI detector (inside white bucket in detector
cabin)', NULL )
go

insert into "PMSDataRecordTypes" values (
```

```

        '0', 'TEMP-OUT',          'Temperature outdoor (measured inside white detector
cabin)',                        NULL )
go

/*****
/* Create AuxStationInfo table */
*****/
CREATE TABLE "dbo"."PMSAuxStationData" (
    "StatId" "int" NOT NULL ,
    "State" "int" NOT NULL DEFAULT (1) ,
    "NetworkCopy" "tinyint" NOT NULL DEFAULT (0),
    "FetchTenMinData" "tinyint" NOT NULL DEFAULT (0) ,
    "StationCallOnAlarm" "tinyint" NOT NULL DEFAULT (0) ,
    "UseDialoutPrefix" "tinyint" NOT NULL DEFAULT (0) ,
    "LocalData" "tinyint" NOT NULL DEFAULT (0) ,
    "MaxGeiger" "smallint" NOT NULL DEFAULT (200) ,
    "MaxRadon_d_eq" "smallint" NOT NULL DEFAULT (200) ,
    "MaxNorm_Back" "smallint" NOT NULL DEFAULT (200) ,
    "MaxRadon_d_dis" "smallint" NOT NULL DEFAULT (200) ,
    "MaxRemain" "smallint" NOT NULL DEFAULT (20) ,
    "MaxSleep" "smallint" NOT NULL DEFAULT (24) ,
    "CallInterval" "smallint" NOT NULL DEFAULT (240) ,
    "Phone" "varchar" (255) NULL ,
    "StationDialoutPrefix" "varchar" (10) NULL ,
    "NextCall" "smalldatetime" NULL ,
    "LastConnect" "smalldatetime" NULL ,
    "LastDataTransfer" "smalldatetime" NULL ,
    "DialupRetryCount" "smallint" NOT NULL DEFAULT (0),
    "GeigerCurlevel" "float" NULL ,
    "RainCurLevel" "float" NULL ,
    "AlarmTime" "smalldatetime" NULL
)
GO

insert into PMSAuxStationData
select StatId,
State,
NetworkCopy,
FetchTenMinData,
StationCallOnAlarm,
UseDialoutPrefix,
0,
MaxGeiger,
MaxRadon_d_eq,
MaxNorm_Back,
MaxRadon_d_dis,
MaxRemain,
MaxSleep,
CallInterval,
Phone,
StationDialoutPrefix,
NextCall,
LastConnect,
LastDataTransfer,
DialupRetryCount,
GeigerCurlevel,
RainCurLevel,
AlarmTime
from pmsstationinfo
GO

/*****
/* Drop not-needed fields from StationInfo table */
*****/
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'

```

```
and name like '%state%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%netwo%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxge%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%geige%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%fetch%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%_____stati_____%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%_____stati_____%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%dialu%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%calli%'
exec (@ex)
go

declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxsl%'
exec (@ex)
go
```

```
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxno%'
exec (@ex)
go
```

```
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxra%'
exec (@ex)
go
```

```
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxra%'
exec (@ex)
go
```

```
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%usedi%'
exec (@ex)
go
```

```
declare @ex varchar(100)
select @ex = 'alter table pmsstationinfo drop constraint '+name from sysobjects
where xtype = 'D'
and name like '%maxre%'
exec (@ex)
go
```

```
alter table PMSStationInfo
drop column State,
NetworkCopy,
FetchTenMinData,
StationCallOnAlarm,
UseDialoutPrefix,
MaxGeiger,
MaxRadon_d_eq,
MaxNorm_Back,
MaxRadon_d_dis,
MaxRemain,
MaxSleep,
CallInterval,
Phone,
StationDialoutPrefix,
NextCall,
LastConnect,
LastDataTransfer,
DialupRetryCount,
GeigerCurlevel,
RainCurLevel,
AlarmTime,
GeigerAlarmlevel
GO
```

```
/* Altering tables */
sp_rename 'PMSTempAtPCAP', 'PMSTempAtNAI'
go
```

```
alter table PMSStationInfo
```

```
add EurostatNutsCode varchar(10) NULL
go
```

```
ALTER TABLE PMSAlarm
  ALTER COLUMN Source
  varchar (25) NOT NULL
go
```

```
ALTER TABLE PMSError
  ADD RegisteredTime
  datetime NULL
go
```

```
/* Create views */
```

```
create view vw_PMSArgosStatData as
select si.StatId, StationName, HostName, HorzPos, VertPos, Enabled,
       State, MaxGeiger, GeigerCurLevel, RainCurLevel
from PMSStationInfo si, PMSAuxStationData asd
Where si.StatId = asd.StatId
      and horzpos <> 0.0
      and vertpos <> 0.0
GO
```

```
create view vw_PMSAllStatData as
select si.HostName,
       si.StationName,
       si.HorzPos,
       si.VertPos,
       si.Country,
       si.EurdepStationID,
       si.EurostatNutsCode,
       si.StationType,
       si.Enabled,
       si.PrimaryPMSServer,
       si.LastUpdate,
       si.Comment,
       asd.*
from PMSStationInfo si, PMSAuxStationData asd
Where si.StatId = asd.StatId
GO
```

```
GRANT SELECT , INSERT , DELETE , UPDATE ON [dbo].[vw_PMSAllStatData] TO [PMS]
GO
```

```
GRANT SELECT , INSERT , DELETE , UPDATE ON [dbo].[vw_PMSAllStatData] TO [PMS_OPERATOR]
GO
```

```
GRANT SELECT , INSERT , DELETE , UPDATE ON [dbo].[vw_PMSArgosStatData] TO [PMS]
GO
```

```
GRANT SELECT , INSERT , DELETE , UPDATE ON [dbo].[vw_PMSArgosStatData] TO [PMS_OPERATOR]
GO
```

```
/* Stored procedures */
```

```
CREATE PROCEDURE get_station_params
@type varchar(5), -- Input param; station type.
@dispname varchar(25) OUTPUT,
@description varchar(150) OUTPUT,
@display bit OUTPUT,
@apparatus varchar(10) OUTPUT,
@samplotype varchar(10) OUTPUT,
@errortype varchar(10) OUTPUT,
@errorvalue varchar(10) OUTPUT,
```



```

@errorunit varchar(10) OUTPUT,
@nuclide varchar(10) OUTPUT,
@measureunit varchar(10) OUTPUT
AS
BEGIN
DECLARE
    @cnt int
SELECT @dispname = NULL,
        @description = NULL,
        @display = NULL,
        @apparatus = NULL,
        @samplotype = NULL,
        @errortype = NULL,
        @errorvalue = NULL,
        @errorunit = NULL,
        @nuclide = NULL,
        @measureunit = NULL
SELECT @cnt = count(*) FROM PMSStationTypes
WHERE StationType = @type
IF @cnt = 0
    SELECT @dispname = 'ERR-01',
           @description = 'Stationtype does not exist in table'
ELSE
BEGIN
    SELECT @cnt = count(*) FROM PMSStationTypes
    WHERE Master = @type
    IF @cnt > 0
        SELECT @dispname = 'ERR-02',
               @description = 'Stationtype has children'
    ELSE
        BEGIN
            SELECT @dispname = TypeName,
                   @description = TypeDescription,
                   @display = Display,
                   @apparatus = Apparatus,
                   @samplotype = SampleType,
                   @errortype = ErrorType,
                   @errorvalue = ErrorValue,
                   @errorunit = ErrorUnit,
                   @nuclide = Nuclide,
                   @measureunit = MeasureUnit
            FROM PMSStationTypes
            WHERE stationtype = @type
            IF @apparatus IS NULL
                SELECT @apparatus = apparatus FROM PMSStationTypes WHERE StationType = 'D'
            IF @samplotype IS NULL
                SELECT @samplotype = samplotype FROM PMSStationTypes WHERE StationType = 'D'
            IF @errortype IS NULL
                SELECT @errortype = errortype FROM PMSStationTypes WHERE StationType = 'D'
            IF @errorvalue IS NULL
                SELECT @errorvalue = errorvalue FROM PMSStationTypes WHERE StationType = 'D'
            IF @errorunit IS NULL
                SELECT @errorunit = errorunit FROM PMSStationTypes WHERE StationType = 'D'
            IF @nuclide IS NULL
                SELECT @nuclide = nuclide FROM PMSStationTypes WHERE StationType = 'D'
            IF @measureunit IS NULL
                SELECT @measureunit = measureunit FROM PMSStationTypes WHERE StationType = 'D'
        END
    END
END
GO

/***** Object: Trigger dbo.tr_CurGeiger    Script Date: 26.02.2001 11:26:55 *****/

DROP TRIGGER "tr_CurGeiger"
GO

```

```

CREATE TRIGGER "tr_CurGeiger" ON "dbo"."PMSGeiger"
FOR INSERT
AS
begin
    DECLARE @alarmtime smalldatetime
    DECLARE @statid integer
    DECLARE @maxgeiger integer
    DECLARE @newtime smalldatetime
    DECLARE @geiger float

    /* Retrieve the inserted values from the pseudo table 'inserted' */
    select
        @statid = statid,
        @newtime = intervalstart,
        @geiger = AverageGeiger
    from inserted

    /* Get last time from StationInfo */
    select @alarmtime = AlarmTime,
        @maxgeiger = MaxGeiger
    from PMSAuxStationData where Statid = @statid

    /* If newer or same time then update */
    if (@newtime >= @alarmtime or @alarmtime is null)
    begin
        update PMSAuxStationData
        set     geigercurlevel = @geiger,
              alarmtime      = @newtime
        where  statid = @statid
    end
end

GO

SET QUOTED_IDENTIFIER OFF      SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON      SET ANSI_NULLS ON
GO

/***** Object: Trigger dbo.tr_CurRain    Script Date: 26.02.2001 11:26:55 *****/

DROP TRIGGER "tr_CurRain"
GO

CREATE TRIGGER "tr_CurRain" ON "dbo"."PMSRain"
FOR INSERT
AS
begin
    DECLARE @alarmtime smalldatetime
    DECLARE @statid integer
    DECLARE @newtime smalldatetime
    DECLARE @rain float

    /* Retrieve the inserted values from the pseudo table 'inserted' */
    select
        @statid = statid,
        @newtime = intervalstart,
        @rain = AverageRain
    from inserted

    /* Get last time from StationInfo */
    select @alarmtime = AlarmTime from PMSAuxStationData where Statid = @statid

    /* If newer or same time then update */
    if (@newtime >= @alarmtime or @alarmtime is null)
    begin

```

```
        update PMSAuxStationData
        set     raincurlevel = @rain,
              alarmtime    = @newtime
        where  statid = @statid
    end
end
GO
```

```
/* Create constraints (where necessary) */
/*****
```

```
ALTER TABLE [dbo].[PMSDialLog] WITH NOCHECK ADD
    CONSTRAINT [pkDialLog] PRIMARY KEY CLUSTERED
    (
        [StatId],
        [Time],
        [Type]
    )
```

```
ALTER TABLE [dbo].[PMSAuxStationData] WITH NOCHECK ADD
    CONSTRAINT [PK_PMSAuxStationData] PRIMARY KEY CLUSTERED
    (
        [StatId]
    )
GO
```

```
ALTER TABLE [dbo].[PMSStationTypes] WITH NOCHECK ADD
    CONSTRAINT [PK_PMSStationTypes] PRIMARY KEY CLUSTERED
    (
        [StationType]
    )
GO
```

```
ALTER TABLE [dbo].[PMSDataRecordTypes] WITH NOCHECK ADD
    CONSTRAINT [PK_PMSDataRecordTypes] PRIMARY KEY CLUSTERED
    (
        [StationType],
        [RecordType]
    )
GO
```

```
ALTER TABLE [dbo].[PMSDialLog] WITH NOCHECK ADD
    CONSTRAINT [fkSidLog] FOREIGN KEY
    (
        [StatId]
    ) REFERENCES [PMSStationInfo] (
        [StatId]
    )
GO
```

```
ALTER TABLE [dbo].[PMSAuxStationData] ADD
    CONSTRAINT [FK_PMSAuxStationData_PMSStationInfo] FOREIGN KEY
    (
        [StatId]
    ) REFERENCES [dbo].[PMSStationInfo] (
        [StatId]
    )
GO
```

```
ALTER TABLE [dbo].[PMSStationInfo] ADD
    CONSTRAINT [FK_PMSStationInfo_PMSStationTypes] FOREIGN KEY
    (
        [StationType]
```

```

    ) REFERENCES [dbo].[PMSStationTypes] (
        [StationType]
    )
GO

ALTER TABLE [dbo].[PMSDataRecordTypes] ADD
    CONSTRAINT [FK_PMSDataRecordTypes_PMSStationTypes] FOREIGN KEY
    (
        [StationType]
    ) REFERENCES [dbo].[PMSStationTypes] (
        [StationType]
    )
GO

/*****
/* Granting permissions */
*****/

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSAuxStationData] TO [PMS]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSAuxStationData] TO [PMS_OPERATOR]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSDataRecordTypes] TO [PMS]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSDataRecordTypes] TO
[PMS_OPERATOR]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSDialLog] TO [PMS]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSDialLog] TO [PMS_OPERATOR]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSStationTypes] TO [PMS]
GO

GRANT REFERENCES , SELECT , INSERT , DELETE , UPDATE ON [dbo].[PMSStationTypes] TO [PMS_OPERATOR]

/*****
/* Updating version info tables */
*****/

if exists (select * from sysobjects where id = object_id('dbo.PMSVersion_20') and sysstat & 0xf = 3)
begin
    drop table "dbo"."PMSVersion_20"
end
GO

CREATE TABLE "dbo"."PMSVersion_2001" (
    "Dummy" char (1) NULL
)
GO

if not exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[DBVersion]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
CREATE TABLE [dbo].[DBVersion] (
    [Application] [varchar] (50) NOT NULL ,
    [Version] [varchar] (50) NOT NULL ,
    [TimeModified] [datetime] NOT NULL ,
    CONSTRAINT [PK_DBVERSION] PRIMARY KEY CLUSTERED
    (
        [Application]

```

```
)
)
GO

insert into "dbo"."DBVersion" values ( 'PMSDB', '3.0.0.4', getdate() )
GO

/*****
/* Update finished */
*****/
select 'Update of PMS Database from 2.0 to 2001 finished'
go
```