

```
/* Microsoft SQL Server - Scripting */
/* Database: PMS */
/* PMS Version: 2001 - SQL Server 7.0 or later */

set quoted_identifier on
go

/*****
/* Check SQL Server version */
*****/

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

/*****
/* Initial check that database exists */
*****/

if not exists (select * from master..sysdatabases where name = 'PMS')
begin
    select 'Database PMS does not exist - Press Cancel (Red square)'
end
go

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

USE PMS
GO

/*****
/* Extra warning if trying to overwrite an existing database */
*****/

if not exists (select * from sysobjects where id = object_id('dbo.PMSPCAP') and sysstat & 0xf = 3)
begin
    create table "PMSPCAP" ( a integer)
end
go

if exists (select * from "dbo"."PMSPCAP")
begin
    select 'WARNING: Database already contains data - Press Cancel (Red square)'
end
go

if exists (select * from "dbo"."PMSPCAP")
begin
    waitfor delay '00:05:00'
end
go

/*****
/* Creating the database */
*****/
```

```

/*****/

select 'Creating PMS Database - Press Cancel (Red square) to abort'
go
waitfor delay '00:00:10'
go

/*****/

if not exists (select * from master..syslogins where name = 'PMS_OPERATOR')
BEGIN
    declare @logindb varchar(30), @loginlang varchar(30) select @logindb = 'master', @loginlang =
'us_english'
    if @logindb is null or not exists (select * from master..sysdatabases where name = @logindb)
        select @logindb = 'master'
    if @loginlang is null or (not exists (select * from master..syslanguages where name = @loginlang)
and @loginlang <> 'us_english')
        select @loginlang = @@language
    exec sp_addlogin 'PMS_OPERATOR', 'PMS', @logindb, @loginlang
END
GO

if not exists (select * from sysusers where name = 'PMS' and uid > 16383)
    EXEC sp_addgroup 'PMS'
GO

if not exists (select * from sysusers where name = 'PMS_OPERATOR' and uid < 16382)
    EXEC sp_adduser 'PMS_OPERATOR', 'PMS_OPERATOR', 'PMS'
GO

GRANT BACKUP LOG TO [PMS_OPERATOR]
GO

GRANT DUMP TRANSACTION TO [PMS_OPERATOR]
GO

/*****/
/* Deletion of tables */
/*****/

select 'Deleting tables'
go

waitfor delay '00:00:05'
GO

if exists (select * from sysobjects where id = object_id('dbo.PMSStationInfo') and sysstat & 0xf = 3)
    select 'Deleting existing tables - Press Cancel (Red square) to abort'
go
if exists (select * from sysobjects where id = object_id('dbo.PMSStationInfo') and sysstat & 0xf = 3)
    waitfor delay '00:00:20'
go

if exists (select * from sysobjects where id = object_id(N'[dbo].[pms_check_db_free_space]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[pms_check_db_free_space]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[get_station_params]') and OBJECTPROPERTY
(id, N'IsProcedure') = 1)
drop procedure [dbo].[get_station_params]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[vw_PMSAllStatData]') and OBJECTPROPERTY
(id, N'IsView') = 1)
drop view [dbo].[vw_PMSAllStatData]
GO

```

```
if exists (select * from sysobjects where id = object_id(N'[dbo].[vw_PMSArgosStatData]') and
OBJECTPROPERTY(id, N'IsView') = 1)
drop view [dbo].[vw_PMSArgosStatData]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSAlarm]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSAlarm]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSBackgroundSpec]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSBackgroundSpec]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSError]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSError]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSEventLog]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSEventLog]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSGeiger]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSGeiger]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSLog]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSLog]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSPcap]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSPcap]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSRain]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSRain]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSRefreshLog]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSRefreshLog]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSServerEvent]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSServerEvent]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSSpecAnalysis]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSSpecAnalysis]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSTempAtNAI]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSTempAtNAI]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSTempInEnclosure]') and OBJECTPROPERTY
```

```

(id, N'IsUserTable') = 1)
drop table [dbo].[PMSTempInEnclosure]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSTempOutdoor]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSTempOutdoor]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSVersion_2001]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSVersion_2001]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSAuxStationData]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSAuxStationData]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSDialLog]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSDialLog]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSDataRecordTypes]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSDataRecordTypes]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSStationInfo]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSStationInfo]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSStationTypes]') and OBJECTPROPERTY
(id, N'IsUserTable') = 1)
drop table [dbo].[PMSStationTypes]
GO

if exists (select * from sysobjects where id = object_id(N'[dbo].[PMSCountry]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)
drop table [dbo].[PMSCountry]
GO

/
*****
/* Creation of tables */
*****
/

select 'Creating tables'
go

CREATE TABLE "dbo"."PMSVersion_2001" (
    "Dummy" varchar (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
```

```
waitfor delay '00:00:05'  
GO
```

```
/  
*****
```

```
CREATE TABLE [dbo].[PMSAlarm] (  
    [StatId] [int] NOT NULL ,  
    [Source] [varchar] (25) NOT NULL ,  
    [Time] [datetime] NOT NULL ,  
    [AlarmId] [int] NOT NULL ,  
    [AlarmMsg] [text] NOT NULL ,  
    [Acknowledged] [tinyint] NULL ,  
    [AcknowledgedBy] [varchar] (25) NULL ,  
    [AcknowledgeTime] [datetime] NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSBackgroundSpec] (  
    [StatId] [int] NOT NULL ,  
    [StartTime] [smalldatetime] NOT NULL ,  
    [EndTime] [smalldatetime] NOT NULL ,  
    [Elapsed] [real] NOT NULL ,  
    [BkgndGamma] [real] NOT NULL ,  
    [BkgndSpecVal] [real] NOT NULL ,  
    [CompressedSpectrum] [image] NOT NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSCountry] (  
    [CtryId] [int] IDENTITY (1, 1) NOT NULL ,  
    [CtryCode] [varchar] (3) NOT NULL ,  
    [CtryName] [varchar] (255) NOT NULL ,  
    [ISO3166Code] [varchar] (2) NOT NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSError] (  
    [StatId] [int] NOT NULL ,  
    [Time] [datetime] NOT NULL ,  
    [Type] [varchar] (10) NOT NULL ,  
    [ErrorMsg] [varchar] (255) NOT NULL ,  
    [RegisteredTime] [datetime] NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSEventLog] (  
    [StatId] [int] NOT NULL ,  
    [ReportTime] [datetime] NOT NULL ,  
    [ReportId] [int] NOT NULL ,  
    [Report] [text] NOT NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSGeiger] (  
    [StatId] [int] NOT NULL ,  
    [IntervalStart] [smalldatetime] NOT NULL ,  
    [IntervalLength] [smallint] NOT NULL ,  
    [AverageGeiger] [float] NOT NULL  
)  
GO
```

```
CREATE TABLE [dbo].[PMSLog] (
    [Appl] [varchar] (5) NOT NULL ,
    [StatId] [int] NOT NULL ,
    [EventTime] [datetime] NOT NULL ,
    [EventText] [varchar] (255) NOT NULL ,
    [FileName] [varchar] (255) NULL
)
GO

CREATE TABLE [dbo].[PMSPcap] (
    [StatId] [int] NOT NULL ,
    [IntervalStart] [smalldatetime] NOT NULL ,
    [IntervalLength] [smallint] NOT NULL ,
    [CompressedSpectrum] [image] NOT NULL ,
    [Real] [float] NULL ,
    [Live] [float] NULL ,
    [Peak] [float] NULL ,
    [Gain] [float] NULL ,
    [Volt] [float] NULL
)
GO

CREATE TABLE [dbo].[PMSRain] (
    [StatId] [int] NOT NULL ,
    [IntervalStart] [smalldatetime] NOT NULL ,
    [IntervalLength] [smallint] NOT NULL ,
    [AverageRain] [float] NOT NULL
)
GO

CREATE TABLE [dbo].[PMSRefreshLog] (
    [LastLogWrite] [smalldatetime] NOT NULL
)
GO

CREATE TABLE [dbo].[PMSServerEvent] (
    [EventTime] [datetime] NOT NULL ,
    [EventID] [varchar] (10) NOT NULL ,
    [EventType] [varchar] (5) NOT NULL ,
    [EventObject] [varchar] (5) NOT NULL ,
    [EventText] [varchar] (255) NOT NULL ,
    [OperatorNotify] [tinyint] NULL
)
GO

CREATE TABLE [dbo].[PMSSpecAnalysis] (
    [StatId] [int] NOT NULL ,
    [IntervalStart] [smalldatetime] NOT NULL ,
    [IntervalLength] [smallint] NOT NULL ,
    [Radon_daughter_eq] [real] NOT NULL ,
    [Radon_daughter_diseq] [real] NOT NULL ,
    [Normal_background] [real] NOT NULL ,
    [Rest] [real] NOT NULL ,
    [AnalysisCode] [smallint] NOT NULL
)
GO

CREATE TABLE [dbo].[PMSTempAtNAI] (
    [StatId] [int] NOT NULL ,
    [IntervalStart] [smalldatetime] NOT NULL ,
    [IntervalLength] [smallint] NOT NULL ,
    [AverageTemp] [float] NOT NULL
)
GO

CREATE TABLE [dbo].[PMSTempInEnclosure] (
```

```
[StatId] [int] NOT NULL ,
[IntervalStart] [smalldatetime] NOT NULL ,
[IntervalLength] [smallint] NOT NULL ,
[AverageTemp] [float] NOT NULL
)
GO

CREATE TABLE [dbo].[PMSTempOutdoor] (
    [StatId] [int] NOT NULL ,
    [IntervalStart] [smalldatetime] NOT NULL ,
    [IntervalLength] [smallint] NOT NULL ,
    [AverageTemp] [float] NOT NULL
)
GO

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 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

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

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) ,

    /* Maximum fields */
    "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) ,
```

```

/* Phone fields */
"CallInterval" "smallint" NOT NULL DEFAULT (240) ,
"Phone" "varchar" (255) NULL ,
"StationDialoutPrefix" "varchar" (10) NULL ,
"NextCall" "smalldatetime" NULL ,

/* Date/time fields */
"LastConnect" "smalldatetime" NULL ,
"LastDataTransfer" "smalldatetime" NULL ,
"DialupRetryCount" "smallint" NOT NULL DEFAULT (0),
"GeigerCurlevel" "float" NULL ,
"RainCurLevel" "float" NULL ,
"AlarmTime" "smalldatetime" NULL
)
GO

CREATE TABLE "dbo"."PMSStationInfo" (
/* ID and name fields */
"StatId" "int" NOT NULL IDENTITY , /* Unique machine generated ID */
"HostName" "varchar(255)" NOT NULL, /* The Win-NT name of the computer */
"StationName" "varchar (255)" NOT NULL , /* Some user-defined name */

/* position, country, id, type */
"HorzPos" "float" NOT NULL DEFAULT (0), /* Longitude */
"VertPos" "float" NOT NULL DEFAULT (0), /* Latitude */
"StationType" "varchar(5)" NOT NULL DEFAULT ('1'), /* At the moment this field is used for
the EurDep Apparatus type code; see EurDep spec. app. D1 */
/* This code is '1' for PMS-East
stations (the default) and '0' (letter 'oh', not zero) for dk-stations */
"Country" "varchar(3)" NOT NULL, /* ID for Country */
"EurdepStationID" "varchar(6)" NULL, /* Eurdep ID : CCxxxx where CC is country
code and xxxx is a four digit sequence number */
"EurostatNutsCode" "varchar(10)" NULL, /* Possibility of specifying the EUROSTAT
NUTS code, if different from the default code */
/* CCxxxx where CC is country
code and xxxx is a n-digit region code */

/* Various flags */
"Enabled" "tinyint" NOT NULL DEFAULT (1), /* Is the station active? */
"PrimaryPMSServer" "tinyint" NOT NULL DEFAULT (0), /* Is the server with this database the
primary PMS server for this station */

/* Date/time fields */
"LastUpdate" "smalldatetime" NOT NULL , /* When was this record last written */

/* The comment field. May contain anything. */
"Comment" "text" NULL /* For comment on station status if
defunct or any specials e.g. on phone connection */
)
GO

/
*****
/* Creation of primary keys and constraints */
/*****

select 'Creating primary keys'
go

waitfor delay '00:00:05'
GO

ALTER TABLE [dbo].[PMSAlarm] WITH NOCHECK ADD
CONSTRAINT [pkAlarm] PRIMARY KEY NONCLUSTERED
(
[StatId],

```

```
        [Time],
        [AlarmId]
    )
GO

ALTER TABLE [dbo].[PMSBackgroundSpec] WITH NOCHECK ADD
    CONSTRAINT [pkBackgroundSpec] PRIMARY KEY NONCLUSTERED
    (
        [StatId]
    )
GO

ALTER TABLE [dbo].[PMSCountry] WITH NOCHECK ADD
    CONSTRAINT [pkCountry] PRIMARY KEY NONCLUSTERED
    (
        [CtryId]
    ),
    CONSTRAINT [uCountry1] UNIQUE NONCLUSTERED
    (
        [CtryCode]
    ),
    CONSTRAINT [uCountry2] UNIQUE NONCLUSTERED
    (
        [CtryName]
    ),
    CONSTRAINT [uCountry3] UNIQUE NONCLUSTERED
    (
        [IS03166Code]
    )
GO

ALTER TABLE [dbo].[PMSError] WITH NOCHECK ADD
    CONSTRAINT [pkError] PRIMARY KEY NONCLUSTERED
    (
        [StatId],
        [Time],
        [Type]
    )
GO

ALTER TABLE [dbo].[PMSEventLog] WITH NOCHECK ADD
    CONSTRAINT [pkEventLog] PRIMARY KEY NONCLUSTERED
    (
        [StatId],
        [ReportTime],
        [ReportId]
    )
GO

ALTER TABLE [dbo].[PMSGeiger] WITH NOCHECK ADD
    CONSTRAINT [pkGeiger] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO

ALTER TABLE [dbo].[PMSLog] WITH NOCHECK ADD
    CONSTRAINT [pkLog] PRIMARY KEY NONCLUSTERED
    (
        [EventTime],
        [Appl],
        [StatId],
        [EventText]
    )
GO
```

```
ALTER TABLE [dbo].[PMSPcap] WITH NOCHECK ADD
    CONSTRAINT [pkPcap] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO
```

```
ALTER TABLE [dbo].[PMSRain] WITH NOCHECK ADD
    CONSTRAINT [pkRain] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO
```

```
ALTER TABLE [dbo].[PMSServerEvent] WITH NOCHECK ADD
    CONSTRAINT [DF_PMSServer_Opera_2AD55B43] DEFAULT (0) FOR [OperatorNotify],
    CONSTRAINT [pkServerEvent] PRIMARY KEY NONCLUSTERED
    (
        [EventTime],
        [EventID]
    )
GO
```

```
ALTER TABLE [dbo].[PMSSpecAnalysis] WITH NOCHECK ADD
    CONSTRAINT [pkSpecAnalysis] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO
```

```
ALTER TABLE [dbo].[PMSStationInfo] WITH NOCHECK ADD
    CONSTRAINT [pkStationInfo] PRIMARY KEY NONCLUSTERED
    (
        [StatId]
    ),
    CONSTRAINT [uStationInfo1] UNIQUE NONCLUSTERED
    (
        [StationName]
    ),
    CONSTRAINT [uStationInfo2] UNIQUE NONCLUSTERED
    (
        [HostName]
    )
GO
```

```
ALTER TABLE [dbo].[PMSTempAtNAI] WITH NOCHECK ADD
    CONSTRAINT [pkTempAtPCAP] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO
```

```
ALTER TABLE [dbo].[PMSTempInEnclosure] WITH NOCHECK ADD
    CONSTRAINT [pkTempInEnclosure] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
```

```
)
GO

ALTER TABLE [dbo].[PMSTempOutdoor] WITH NOCHECK ADD
    CONSTRAINT [pkTempOutdoor] PRIMARY KEY NONCLUSTERED
    (
        [IntervalStart],
        [StatId],
        [IntervalLength]
    )
GO

ALTER TABLE [dbo].[PMSAuxStationData] WITH NOCHECK ADD
    CONSTRAINT [pkAuxStationData] PRIMARY KEY NONCLUSTERED
    (
        [StatId]
    )
GO

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

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

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

CREATE INDEX [ixPMSAlarm] ON [dbo].[PMSAlarm]([StatId], [Acknowledged], [Time])
GO

CREATE INDEX [ixLogRead] ON [dbo].[PMSLog]([StatId])
GO

CREATE INDEX [ixLogRead2] ON [dbo].[PMSLog]([Appl], [EventTime])
GO

/*****
/* granting permissions */
*****/

select 'Granting permissions'
go

waitfor delay '00:00:05'
GO

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

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

GO

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

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

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

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

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

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

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

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

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

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

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

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

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

GRANT REFERENCES , **SELECT** , **INSERT** , **DELETE** , **UPDATE** **ON** [dbo].[PMS pcap] **TO** [PMS_OPERATOR]
GO

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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]
GO

/*****
/* Creation of constraints */
*****/

select 'Creating foreign keys'
go

waitfor delay '00:00:05'
GO

ALTER TABLE [dbo].[PMSAlarm] ADD
    CONSTRAINT [fkSidAlarm] FOREIGN KEY
    (
        [StatId]
    ) REFERENCES [dbo].[PMSStationInfo] (
        [StatId]
```

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

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

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

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

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

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

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

```
ALTER TABLE [dbo].[PMSDataRecordTypes] ADD
```

```
CONSTRAINT [FK_PMSDataRecordTypes_PMSStationTypes] FOREIGN KEY
(
    [StationType]
) REFERENCES [dbo].[PMSStationTypes] (
    [StationType]
)
GO
```

```
ALTER TABLE [dbo].[PMSStationInfo] ADD
CONSTRAINT [fkCountry] FOREIGN KEY
(
    [Country]
) REFERENCES [dbo].[PMSCountry] (
    [CtryCode]
),
CONSTRAINT [fkStationType] FOREIGN KEY
(
    [StationType]
) REFERENCES [dbo].[PMSStationTypes] (
    [StationType]
)
GO
```

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

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

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

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

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

```
SET QUOTED_IDENTIFIER ON SET ANSI_NULLS ON
GO
```

```
/*
*****
*/ Creation of objects */
/*****
```

```
select 'Creating other objects'
go
```

```
waitfor delay '00:00:05'
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,
@sampletype 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,
        @sampletype = 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,
               @sampletype = 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: Stored Procedure dbo.pms_check_db_free_space    Script Date: 26.02.2001 11:26:55 *****/
create procedure "dbo"."pms_check_db_free_space"
as
begin
    declare @dbsize      dec(15,5)
    declare @percentused dec(15,5)

    select @dbsize = sum(size)
           from sysfiles
           where status & 0x40 = 0

    select @percentused =
           (select sum(reserved)
            from sysindexes
            where indid in (0, 1, 255)
           ) / @dbsize * 100

    if @percentused > 95
        begin
            insert into pmsserverevent values
            ( getdate(), 'ALARMDBFULL', 'ALARM', 'DB', 'Database more than 98% full', 0 )
        end
    else
        if @percentused > 85
            begin
                insert into pmsserverevent values
                ( getdate(), 'WARNDBFULL', 'WARN', 'DB', 'Database more than 90% full', 0 )
            end
end
GO

SET QUOTED_IDENTIFIER OFF    SET ANSI_NULLS ON
GO

GRANT EXECUTE ON [dbo].[pms_check_db_free_space] TO [PMS]
GO

GRANT EXECUTE ON [dbo].[pms_check_db_free_space] TO [PMS_OPERATOR]
GO

```

```
SET QUOTED_IDENTIFIER ON SET ANSI_NULLS ON
GO
```

```

/*****
/* Triggers */
*****/

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

```

```

CREATE TRIGGER "tr_UpdRefreshLog" ON "dbo"."PMSLog"
FOR INSERT
AS
BEGIN
    DECLARE @eventtime smalldatetime
    SELECT @eventtime = i.EventTime FROM Inserted i
    UPDATE PMSRefreshLog Set LastLogWrite = @eventtime
END

```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON SET ANSI_NULLS ON
GO
```

```

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

```

```

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 *****/

```

```

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 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
```

```

/
*****
/* Creation data                                     */
/*****

```

```
select 'Creating data'
go
```

```
waitfor delay '00:00:05'
GO
```

```
insert into "PMSCountry" values ( 'Den', 'Denmark', 'DK' )
insert into "PMSCountry" values ( 'Pol', 'Poland', 'PL' )
insert into "PMSCountry" values ( 'Est', 'Estonia', 'EE' )
insert into "PMSCountry" values ( 'Lat', 'Latvia', 'LV' )
insert into "PMSCountry" values ( 'Lit', 'Lithuania', 'LT' )
insert into "PMSCountry" values ( 'Rus', 'Russia', 'RU' )
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
```

```
insert into "PMSRefreshLog" values ( getdate() )
GO
```

```
insert into "PMSLog" values ( 'PMSMN',0,getdate(),'PMS 2001 system installed',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 )
GO
```

```
insert into "PMSStationTypes" values ( '2', 'Air sampler station', 'Polish ASS-500 Airsampler',
1, 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)
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)
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)
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

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

/
*****
/* Creation data */
/*****

select 'Creating SQLAgent jobs'
go

waitfor delay '00:00:05'
GO

SET QUOTED_IDENTIFIER OFF SET ANSI_NULLS ON
GO

if exists (select * from msdb.dbo.systasks where name = 'pms_check_db_free_space_task')
exec msdb.dbo.sp_droptask 'pms_check_db_free_space_task'
GO

exec msdb.dbo.sp_addtask
  'pms_check_db_free_space_task', /* task name */
  'TSQL', /* Task type; Run task in Transact-SQL */
  NULL, /* Server */
  'PMS_OPERATOR', /* User */
  'PMS', /* Database */
  1, /* Enabled */
  4, /* Run every x days daily */
  1, /* x = 1 day */
  8, /* Run every x hours */
  6, /* x = 6 hours */
  1, /* Irrelevant */
  4, /* Que? */
  0, /* Start date; 0 means from now on */
  0, /* End date; 0 mens never */
  0, /* Start time on day; 0 means 00:00 */
  0, /* End date; 0 mens 23:59 */
  0, /* Irrelevant */
  0, /* Irrelevant */
  0, /* Irrelevant; thread priority */
  NULL, /* Irrelevant; e-mails handled by PMSAlarmView */
  3, /* The number of retry-attempts */
  1, /* Wait x minutes before retrying; x = 1 */

```

```
'pms_check_db_free_space', /* The command to execute */
2      /* Relevant only if e-mail operator defined */
GO

waitfor delay '00:00:05'
GO

select 'Creation of PMS Database finished'
go
```