B
Database Queries

Important Database Tables

A lot of EPiServer database processing deals with about two handfuls of tables. Most aspects of Page Types, Properties and Web Pages are stored in the database. The most important tables in this respect are listed in table B-1.

Table B-1: The most important tables in the database for Page Types, Properties and Web Pages.

Table Name	*Description*
tblPage	Web Pages, a great deal of the contents are used to instantiate the PageData object.
tblPageDefinition	Properties and the Web Pages they are attached to.
tblPageDefinitionType	Property Data Types, both built-in and user-defined.
tblPageType	Defined Page Types.
tblProperty	All Properties for all Web Pages and their values.
tblACL	Access Control List, ACL, for a Web page.
tblSID	EPiServer Security Identifiers for users and groups.
tblSIDGroup	Connector between groups and users (connects entries in tblSID)
tblUser	Registered users on the Web site.

Please note that all queries below assume that the query is executed inside the pertinent database.

The relationships between the tables can be seen in figures B-1 and B-2.

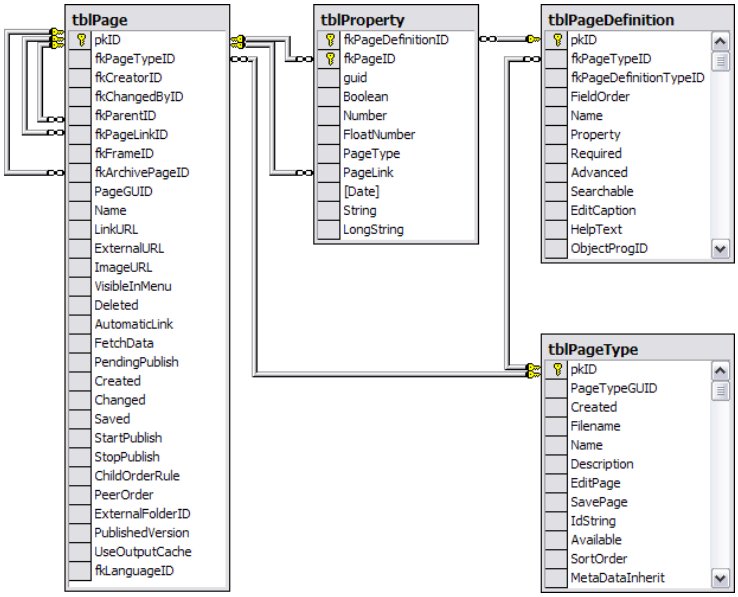

Figure B-1: Relationships between the database tables, tblPage, tblProperty, tblPageDefinition and tblPageType.

In figure B-1, note for example that in tblPage there are two internal relationships. One of them, between columns fkParentID and pkID, is used to realise the page tree hierarchy. Every page in the page tree has a parent except for the page pointed to by EPiServer.Global.EPConfig.RootPage, which has a null entry for fkParentID.

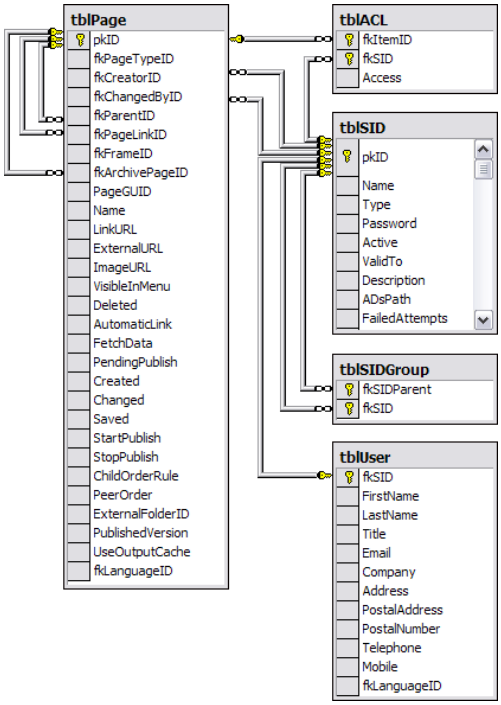

Figure B-2: Relationships between tblPage, tblACL, tblSID, tblSIDGroup and tblUser.

Figure B-2 depicts four different sets of relationships:

- Linking user to EPiServer Security ID, SID, and EPiServer Group
- Linking access permission (Access Control List, ACL) to an EPiServer Security ID
- Linking page creators and changers to EPiServer Security ID

SQL Queries to Retrieve Page Types, Properties and Web Pages

List All Defined Page Types

Example B-1: SQL query to list all defined Page Types.

```
select Name, Filename from tblPageType
order by Name
```

There is more interesting information in the Page Type table, e.g. the column pkID, which is the primary key. You'll find this primary key used as a foreign key in several other tables, mostly under the name fkPageTypeID.

List All Page Template Files, Page Types and Web Pages

Example B-2: SQL Query to list all Page Template Files, Page Types and Web Pages.

```
select distinct tblPageType.FileName 'Page Template File', tblPageType.Name 'Page Type Name',
tblPageType.Description 'Page Type Description', tblPage.Name 'Web Page Name',
tblPage.LinkURL 'Web Page URL'
from tblPageType
inner join tblPage
    on tblPageType.pkID = tblPage.fkPageTypeID
order by tblPage.Name
```

List All Defined Data Types

Example B-3: SQL query to list all Property Data Types.

```
select Name, TypeName, AssemblyName from tblPageDefinitionType
order by TypeName, Name
```

Executing this query against a newly installed EPiServer Web site database would yield a result set much like this:

Table B-2: EPiServer Data Types defined at installation.

Data Type Name	*Full Data Type Name*	*Assembly Name*
Boolean	NULL	NULL
Category	NULL	NULL
Date	NULL	NULL
FloatNumber	NULL	NULL
Form	NULL	NULL
LongString	NULL	NULL
Number	NULL	NULL
PageReference	NULL	NULL
PageType	NULL	NULL
String	NULL	NULL
DocumentUrl	EPiServer.SpecializedProperties.Property-DocumentUrl	EPiServer
Frame	EPiServer.SpecializedProperties.PropertyFrame	EPiServer

Table B-2: EPiServer Data Types defined at installation.

Data Type Name	*Full Data Type Name*	*Assembly Name*
ImageUrl	EPiServer.SpecializedProperties.PropertyImageUrl	EPiServer
Language	EPiServer.SpecializedProperties.PropertyLanguage	EPiServer
Password	EPiServer.SpecializedProperties.PropertyPassword	EPiServer
Selector	EPiServer.SpecializedProperties.PropertySelector	EPiServer
Sid	EPiServer.SpecializedProperties.PropertySid	EPiServer
SortOrder	EPiServer.SpecializedProperties.PropertySortOrder	EPiServer
Url	EPiServer.SpecializedProperties.PropertyUrl	EPiServer
WeekDay	EPiServer.SpecializedProperties.PropertyWeekDay	EPiServer

List All Defined Property Types and Their Data Type

Example B-4: SQL Query to list all defined Property Types and their Data Type.

```
select distinct tblPageDefinition.Name 'Property Name', tblPageDefinitionType.Name 'Data Type',
tblPageDefinition.EditCaption, tblPageDefinition.HelpText
from tblPageDefinition inner join tblPageDefinitionType
    on tblPageDefinition.fkPageDefinitionTypeID=tblPageDefinitionType.pkID
order by tblPageDefinition.Name
```

List All Page Types and Their Properties

Example B-5: SQL query list all defined Page Types and their Properties.

```
select tblPageType.Name 'Page Type Name', tblPageType.Description 'Page Type Description',
tblPageType.FileName 'Page Template File', tblPageDefinition.Name 'Property Name',
tblPageDefinition.EditCaption 'Edit Heading', tblPageDefinition.HelpText 'Help Text'
from tblPageType inner join tblPageDefinition
    on tblPageType.pkID = tblPageDefinition.fkPageTypeID
order by tblPageType.Name, tblPageDefinition.FieldOrder
```

List All Dynamic Properties

Example B-6: SQL query to list all Dynamic Properties.

```
select distinct tblPageDefinition.Name 'Property Name', tblPageDefinitionType.Name 'Data Type',
tblPageDefinition.EditCaption, tblPageDefinition.HelpText
from tblPageDefinition inner join tblPageDefinitionType
    on tblPageDefinition.fkPageDefinitionTypeID=tblPageDefinitionType.pkID
```

```
    where tblPageDefinition.fkPageTypeID is null
order by tblPageDefinition.Name
```

List All Web Pages with Their Properties and Current Values

Example B-7: SQL query to list all Web Pages with their Properties and current values.

```
select tblPage.LinkURL 'Web Page URL', tblPage.Name 'Web Page Name', tblPageDefinition.Name
'Property Name', 'Property Value' =
    case
        when tblProperty.Number is null and tblProperty.FloatNumber is null and
            tblProperty.PageType is null and tblProperty.PageLink is null and tblProperty.Date is null and
            tblProperty.String is null and tblProperty.LongString is null
            then 'Boolean: ' + cast( tblProperty.Boolean as varchar( 40 ) )
        when tblProperty.Number is not null then 'Number: '+ cast( tblProperty.Number as varchar( 40 ) )
        when tblProperty.FloatNumber is not null then 'FloatNumber: '+ cast( tblProperty.FloatNumber as
            varchar( 40 ) )
        when tblProperty.PageType is not null then 'PageType: '+ cast( tblProperty.PageTypeas
            varchar( 40 ) )
        when tblProperty.PageLink is not null then 'PageLink: '+ cast( tblProperty.PageLinkas
            varchar( 40 ) )
        when tblProperty.Date is not null then 'Date: '+ cast( tblProperty.Dateas varchar( 40 ) )
        when tblProperty.String is not null then 'String: '+ cast( tblProperty.Stringas varchar( 40 ) )
        when tblProperty.LongString is not null then 'LongString: '+ cast( tblProperty.LongStringas
            varchar( 40 ) )
        else cast( 'Error Determining Value!' as varchar( 40 ) )
    end
from tblProperty inner join tblPage
    on tblProperty.fkPageID = tblPage.pkID inner join tblPageDefinition
    on tblProperty.fkPageDefinitionID = tblPageDefinition.pkID
order by tblPage.LinkURL, tblPage.Name, tblPageDefinition.Name
```

SQL Query to List All User Tables and Their Columns In a SQL Server Database

Example B-8: SQL query to list all user tables and their columns in a Microsoft SQL Server database.

```
select sysobjects.Name, syscolumns.Name
    from sysobjects inner join syscolumns on sysobjects.id = syscolumns.id
where sysobjects.xtype = 'U'
order by sysobjects.Name, syscolumns.colorder
```

SQL Server Procedure to Display the Web Page Hierarchy

SQL Server Books Online presents a stored procedure to display hierarchies in stored data (see 'Expanding Hierarchies'). We found a more elegant, recursive, stored procedure to solve the same problem on the Internet (see http://vyaskn .tripod.com/hierarchies_in_sql_server_databases.htm).

This is what it looks like adapted for tblPage:

Example B-9: SQL commands to produce and execute a recursive stored procedure to display the Web Page hierarchy in table tblPage.

```
IF EXISTS (SELECT name FROM sysobjects
        WHERE name = 'ShowHierarchy' AND type = 'P')
    DROP PROCEDURE ShowHierarchy
go

CREATE PROC dbo.ShowHierarchy ( @Root int ) AS BEGIN
    SET NOCOUNT ON
    DECLARE @PageID int, @PageName varchar(30)

    SET @PageName = (SELECT Name FROM dbo.tblPage WHERE pkID = @Root)
    PRINT REPLICATE( '-', @@NESTLEVEL * 4) + @PageName

    SET @PageID = (SELECT MIN( pkID ) FROM dbo.tblPage WHERE fkParentID = @Root)

    WHILE @PageID IS NOT NULL
    BEGIN
        EXEC dbo.ShowHierarchy @PageID
        SET @PageID = (SELECT MIN( pkID ) FROM dbo.tblPage
            WHERE fkParentID = @Root AND pkID > @PageID)
    END
END
go

ShowHierarchy 1
go
```

Assuming that the smallest pkID value is 1, the commands in example B-9 would produce the complete hierarchy for all pages in tblPage.

To see only the Web pages which are marked as VisibleInMenu (value set to 1), alter the above procedure to look like this:

Example B-10: Recursive stored procedure to list the hierarchy of all visible Web pages.

```
CREATE PROC dbo.ShowHierarchy ( @Root int ) AS BEGIN
    SET NOCOUNT ON
    DECLARE @PageID int, @PageName varchar(30)

    SET @PageName = ( SELECT Name FROM dbo.tblPage WHERE pkID = @Root )
    PRINT REPLICATE('-', @@NESTLEVEL * 4) + @PageName

    SET @PageID = (SELECT MIN( pkID ) FROM dbo.tblPage WHERE fkParentID = @Root
        and VisibleInMenu = 1 )

    WHILE @PageID IS NOT NULL
    BEGIN
```

```
            EXEC dbo.ShowHierarchy @PageID
            SET @PageID = ( SELECT MIN( pkID ) FROM dbo.tblPage WHERE fkParentID = @Root
            and VisibleInMenu = 1 AND pkID > @PageID )
        END
    END
```