

Adabas SQL Server

The Adabas SQL catalog contains all the necessary information on Adabas tables and views. This information can be documented in Predict, and from this documentation a table or view can be created with a Predict generation, incorporation or administration function.

This section covers the following topics:

- General Information
 - Documenting Adabas SQL Server in Predict
 - Naming Conventions for Adabas SQL Server
 - Generating, Incorporating and Comparing Adabas SQL Objects
 - Administrating Adabas SQL Server Objects
 - XRef Data for Adabas SQL Server Objects
-

General Information

Predict supports the following SQL statements:

- CREATE TABLE DESCRIPTION
- CREATE CLUSTER DESCRIPTION
- CREATE VIEW
- DROP TABLE DESCRIPTION
- DROP CLUSTER DESCRIPTION
- DROP VIEW

Note:

The statements CREATE TABLE DESCRIPTION and CREATE CLUSTER DESCRIPTION are supported instead of CREATE TABLE and CREATE CLUSTER to pass existing data structures to the Adabas SQL Server. Also, Predict descriptions already take account of a variety of Adabas-specific features.

Prerequisites

Parts of the ADVANCED Interactive Facilities of Adabas SQL Server must be installed within Natural. For detailed information see the requirements table in the respective part of the **Predict Installation documentation**.

Documenting Adabas SQL Server in Predict

The following Adabas SQL Server objects can be documented in Predict:

- Adabas tables
- Adabas Views
- Indexes
- Unique elements
- Primary and foreign keys
- Referential constraints

Documenting Adabas Tables

There are two methods of documenting Adabas tables:

With Files of Type A - with Adabas SQL usage set to Y

If an Adabas file corresponds **exactly** to a base table in Adabas SQL Server, it can be documented as a file of type A (SQL).

The Adabas file must not contain groups structures or multiple value fields. Rotated fields are not supported with this method.

This method is retained for reasons of compatibility with earlier Predict versions.

With Files of Type AT

Tables can also be documented with files of type AT (Adabas cluster table). Files of this type can be understood as userviews to an Adabas file.

Files of type AT have the following additional attribute:

Table level	<p>0 Only "flat" structures are permitted (no MU or PE fields).</p> <p>1 For defining multiple fields and periodic groups.</p> <p>2 For defining multiple fields within a periodic group.</p>
-------------	--

There are two methods of documenting periodic groups and multiple value fields in AT files:

- If the occurrences of PE/MU fields are **fixed**, you can use rotated fields in the AT file.
- If the occurrences of PE/MU fields are **variable**, use subtables (AT files at level 1 or level 2).

Documenting Adabas Views

Adabas views can be documented with files of type B.

Subselect Clauses and Expressions in Field Definitions

The documentation of views is supported by an additional editor to specify the part of the subselect clause starting from the first FROM clause.

The selection clause of the subselect is documented by the specification of the field list of the view. The specified list of tables/views in the first FROM clause of the subselect is generated by Predict and will be updated if a field from an additional table/view is added to the view. Correlation names can be added to the tables and views in the list (using functions of the editor). The remaining part of the subselect clause is left unchanged.

The expression used to define Adabas SQL fields can contain complex expressions. Fields that are defined not only by a single column name but use either a constant or a more complex expression are called derived fields. A special editor is provided for the specification of the expression of derived fields.

In the field expression and in the subselect clause, comment lines (lines starting with * or **) and remarks within a line (starting with /*) are allowed.

Documenting Adabas SQL Databases/ Tablespaces

Adabas SQL database and tablespace definitions need not be documented with separate Predict objects:

- Adabas SQL databases need not be documented with separate Predict objects, because the information to which Adabas database an Adabas SQL tablespace belongs is documented with the file-database link.
- The properties of Adabas SQL tablespaces are documented as attributes of Adabas file objects.

The physical implementation of Adabas tables in Adabas can be performed by Predict (with Generate Adabas file or Generate ADACMP/ADAFDU).

The description needed by Adabas SQL to address the Adabas files can be generated with the function Generate Adabas table description.

Documenting Adabas SQL Columns

Adabas SQL columns are documented as field objects in Predict.

Adabas SQL Server has fields with data type SEQNO. These fields are documented in Predict with fields of type QN. This data type is used for documenting occurrences of MU or PE fields:

- SEQNO(0) corresponds to the ISN of the underlying Adabas C table
- SEQNO(1) corresponds to the index of a multiple-value or periodic field.
- SEQNO(2) corresponds to a multiple-value field within a periodic group.

For fields of this type, the column Occ represents an individual occurrence of a PE or MU field, and not the maximum number of occurrences.

These fields can be given a name in a table description and can be selected. However, they are only necessary if you want to perform a search operation using individual occurrences. If an AT file contains more than one MU fields or fields from more than one periodic group, it is only possible to address the same occurrence by means of a SEQNO(1) or SEQNO(2) field.

Documenting Indexes

- The following attributes of indexes can be specified:
 - index name,
 - definition of index
- If the index consists of only one column, the field documenting the column is marked as a descriptor with descriptor type D.
- If the index consists of multiple columns, it is documented as a field with field type SP (superfield) and descriptor type D.

Documenting Unique Elements

Unique elements are documented as Fields with descriptor type D, unique option U and suppression option R.

Documenting Primary and Foreign Keys

Primary Keys

A primary key always includes an index with descriptor option P.

Foreign Keys

Foreign keys are documented as follows:

- If the foreign key consists of only one column, the field documenting the foreign key is marked as a descriptor with descriptor type F or E.
- If the foreign key consists of multiple columns, it is documented as a field with field type SP (superfield) and descriptor type F or E. The descriptor type F means the field is a foreign key and an index. E means the field is a foreign key without an index.

Documenting Referential Constraints

Referential constraints are documented as file relations of type R (referential constraint). The relation is established between a unique element and a foreign key. Primary and foreign key must belong to different subtables. The subtables themselves must belong to the same Adabas file.

Naming Conventions for Adabas SQL Server

Adabas SQL naming conventions have to be followed when creating or maintaining Predict objects for Adabas SQL. The following rules apply:

- Valid identifiers are from 2 to 32 characters long, must start with an alpha character (A - Z) and may be followed by either an alpha, a digit or underscore. Identifiers must obey these rules if Predict maintenance and generation functions are to be applied.
- The identifier of a table or view must be given in qualified form: the schema identifier (maximum length 32 characters), a delimiter and the table/view name (maximum length 32 characters). A hyphen is used as a delimiter (not a period as in SQL). An example: SYSSAG-SYSCOLUMNS. Hyphens in names are treated as follows:
 - When a table/view is generated from a Predict table/view object the hyphen will be transformed into a period (.).
 - Because hyphens are used as delimiters, only one hyphen can occur in the SQL identifier. Column names must not contain a hyphen.
 - The hyphen can be used as a minus sign or negative sign in the field expression or the subselect clause and must then be preceded by a blank.

Correlation Names

Correlation names can be defined in the subselect clause of a view. If a correlation name is defined for a table/view in the subselect clause, all references (in field expressions as well as in the field editor of the view) to columns of the table/view must be qualified with the correlation name. If no correlation name is defined for a table/view in the subselect clause, all references to columns of the table/view must be fully qualified with creator-tablename-columnname (for example: SYSSAG-SYSCOLUMNS-COLNAME).

Index Names

Index names must be fully qualified: schema, delimiter, index name.

Generating, Incorporating and Comparing Adabas SQL Objects

Prerequisites

The following Predict functions are subject to SQL security mechanisms:

- Function Generate Adabas Table/View and administration functions Purge and Refresh:
When the catalog is accessed for the first time, the user ID DBA is used for read access to the catalog.
- Generate Adabas Table/View:
If a description is generated into a schema which is not owned by the catalog user, a window appears in which you can enter the ID and password of the schema owner. In batch mode, use the command SET SCHEMA_OWNER.
- Incorporation and Comparison functions:
User must have SELECT privilege for the schema definition_schema which is delivered with Adabas SQL Server.

Generate Table Description, Cluster Description

With the Predict function Generate Adabas Table/View, a CREATE TABLE DESCRIPTION statement or a CREATE CLUSTER DESCRIPTION statement is generated from a file of type A (with SQL usage set to Y) or from a file of type AT. See table below:

Constellation	Generated Command	Note
One Adabas file corresponding to one SQL table	GENERATE TABLE DESCRIPTION	
One AT file corresponding to one SQL table		
Multiple AT files corresponding to multiple SQL tables	GENERATE CLUSTER DESCRIPTION	Specify only one file in the cluster. All files in the cluster will be used for generation.

The statements add descriptions of multiple tables in an Adabas SQL catalog **without** creating an Adabas file (the standard SQL statements CREATE TABLE/CLUSTER generate both an Adabas file and a description in an Adabas SQL catalog).

This description contains the following:

- a list of fields in the file
- details of descriptors and superdescriptors, unique constraints, primary and foreign keys
- database name and file number. This information is used to access the Adabas file.
The database name is taken from the catalog. If no database name exists, a CREATE DATABASE statement is generated. If the ID of the Predict database object complies with SQL naming conventions, this name is taken. If not, the database name DB_DBnr is generated.

The available options are described under Adabas Table/View in the section **Generation** in the **External Objects in Predict documentation**.

If the database/file number is a logical file number or the number of an anchor file (with a file of type Expanded), the table description is appended with 'MODIFICATION NOT ALLOWED'. This has the result that ALTER statements for this table are rejected. Only DML (Data Manipulation Language) statements can be executed for tables marked in this manner; DDL (Data Definition Language) statements are not possible except DROP TABLE DESCRIPTION.

The generated SQL statements can be saved in a protocol.

Generate View

With the Predict function Generate Adabas Table/View, a CREATE VIEW statement is generated from a file of type B.

The available options are described under Adabas Table/View in the section **Generation** in the **External Objects in Predict documentation**.

Incorporate Table

With the Predict function Incorporate Adabas Table/View, a file of type A (with SQL usage set to Y) or a file of type AT is documented in Predict from the entry in the Adabas SQL catalog. The constellation in the catalog determines which file type is incorporated in Predict:

Constellation	Incorporated Object
External object is contained in the catalog as a table and created with CREATE TABLE / CREATE TABLE DESCRIPTION or Cluster with one table and no rotating fields	File of type A (with SQL usage) set to Y.
Master Adabas file exists in Predict (determined using DBnr/Fnr).	File of type AT.

The database number is interpreted as a physical Adabas database and a link is created from this database to the Adabas file. The values of the Adabas attributes can either be taken from the tablespace attributes in the catalog, or default values are used which can be adapted to the 'real world' with the function Compare Adabas File.

Incorporate View

With the Predict function Incorporate Adabas Table/View, a file of type B is documented in Predict from the entry in the Adabas SQL catalog.

Compare Adabas Table/View

Files of type A (SQL), type AT and type B are compared with information contained in the description of the table/view in an Adabas SQL catalog. See the section Comparison in the **External Objects in Predict documentation** for more details.

Administrating Adabas SQL Server Objects

The following administration functions are valid for Adabas table descriptions. Enter function code L and object code FI in any Predict main menu or the command ADMINISTRATE FILE. File type is A (SQL), AT or B and external object code is EQ.

For more information see the section Files in the section **Administration of External Objects** in the **External Objects in Predict documentation**.

Disconnect Implementation

Deletes the generation protocol and the generation pointer from the Predict File object to the Adabas table description, but the table description is left intact.

With files of type AT, all files used for generation are disconnected in a single operation.

Display Implementation

Displays documentation data, generation options and generated table description of specified file ID(s).

Rename Implementation

Moves the generation protocol to another member and/or library.

Purge Implementation

Deletes Adabas table descriptions and all dependent views.

A DROP TABLE DESCRIPTION, DROP CLUSTER DESCRIPTION or DROP VIEW statement will delete the definition from the Adabas SQL catalog and any statements referencing this table / view are marked as invalid.

When a table / view is dropped, all dependent views are dropped too.

Note:

The function is equivalent to the function Generate with Replace table/view set to Y.

Select Implementation

Selects Adabas table/view for further processing.

XRef Data for Adabas SQL Server Objects

Programs using embedded SQL must be precompiled with the Adabas SQL precompiler before the host language compiler is executed.

During precompilation, an option is available to create XRef data. The creation of XRef data is controlled by options specified in the Adabas SQL parameters. See the Adabas SQL documentation for a detailed description of these options.

As described in the section Third Generation Languages in this documentation, XRef data is always assigned to members contained in logical libraries. The library containing the member can be specified explicitly or - if no library is specified - the member is assigned to a default library depending on the host language. See table below.

Host Language	Default-Library
C	*SYSCCC*
COBOL	*SYSCOB*
PL/I	*SYSPLI*
FORTRAN	*SYSFOR*

The following XRef data is stored for programs processed by the Adabas SQL precompiler:

- directory information (user ID, terminal ID, date and time of precompilation)
- each Copy/Include Code member used in the program
- each table name used in certain clauses of DML statements, together with the usage:

Statement	Clause	Usage
SELECT	FROM	Read
INSERT	INTO	Store
UPDATE		Update
DELETE	FROM	Delete

- each column name used in certain clauses of DML statements, together with the usage:

Statement	Clause	Usage
SELECT		Read
SELECT	WHERE	Search
INSERT	INTO	Store
UPDATE	<row amendment expression>	Update
UPDATE	WHERE	Search
DELETE	WHERE	Search