SQL Access to the Migrated Files

This chapter covers the following topics:

- Introduction
- How to Access the Migrated Data
- Improving the Access to Migrated Data
- Restrictions for SQL Applications

Introduction

The Adabas SQL Gateway provides business users real-time access to Adabas data, no matter where that data resides, using standard SQL-based desktop tools.

With Adabas SQL Gateway, mission-critical Adabas data can be accessed quickly and easily by any ODBC, JDBC, OLE DB or .NET standard SQL application. IT and business users also can access Adabas with SQL-based reporting tools, such as Business Objects, Crystal Reports and MS Office. With Adabas SQL Gateway, IT can deliver business users the information they need, straight to their desktops.

After the DL/I databases have been converted to Adabas files, these files can be accessed by SQL applications with the Adabas SQL Gateway. The advantages for the Adabas data offered by the Adabas SQL Gateway can be extended to DL/I data this way.

This section explains:

- How to access the migrated data from SQL applications.
- How to improve the access.
- What restrictions apply for the SQL applications.

The installation and operation of the Adabas SQL Gateway is described in the Adabas SQL Gateway manual.

How to Access the Migrated Data

Import the Adabas definitions to the Adabas SQL Gateway data dictionary with the CONNX Data Dictionary Manager. You can either use the FDT import or if you have already generated DDMs for Natural, you can use the SYSTRANS import function.

As described in the section Using ADL Files with Natural/Adabas in this documentation, ADL mirrors the hierarchical structure defined in the DL/I database by using physical pointers and foreign keys. An SQL table should reflect all the fields belonging to a DL/I segment plus the foreign key fields. SQL tables should not contain the physical pointer fields nor other ADL internal fields (Z0 – Z8, secondary index descriptor).

Once the data definitions are imported, you can read the data with SQL applications.

Improving the Access to Migrated Data

Depending on the data you want to read with the SQL application, you should consider defining new descriptors and superdescriptors in the Adabas file. You may define as descriptor:

- Sequence fields of dependent segments (the root sequence field is already a descriptor).
- Foreign key fields.

You may define as superdescriptors:

- The concatenated key fields, i.e. all foreign key fields of the parents of a segment. This superdescriptor may additionally include the sequence field of the segment itself if available.
- Fields building up a secondary index.

You may define other fields as descriptors or superdescriptors even if there is no counterpart in the DL/I structure. But ...

- Do not build superdescriptors from fields of different segments. It will not contain any data.
- Do not allocate descriptors/superdescriptors which you do not use. It is an overhead in performance.

Restrictions for SQL Applications

SQL applications must not update the migrated data. If an SQL application would insert dependent segment data, DL/I applications would not “see” this data. This is because there is no "Consistency" for SQL which updates the ADL internal physical pointer fields which are required for the DL/I applications.

Moreover if an SQL application updates the data, it can destroy the hierarchy. For example if it deletes a root segment data, the dependent data is still in the database but cannot longer be accessed by DL/I applications. Once again, no "Consistency" prevents the SQL application from destroying the referential integrity.

Note that these restrictions apply only as long as DL/I applications access the data. As soon as all DL/I applications have been replaced, you can modify the data with SQL applications without restrictions.