Database Access

Adrian Streitz, Johannes Rank, Borys Levkovskyi

Prerequisites

- Experience with ABAP Workbench and Eclipse
- Basic ABAP programming skills

Agenda

I. **Data dictionary**

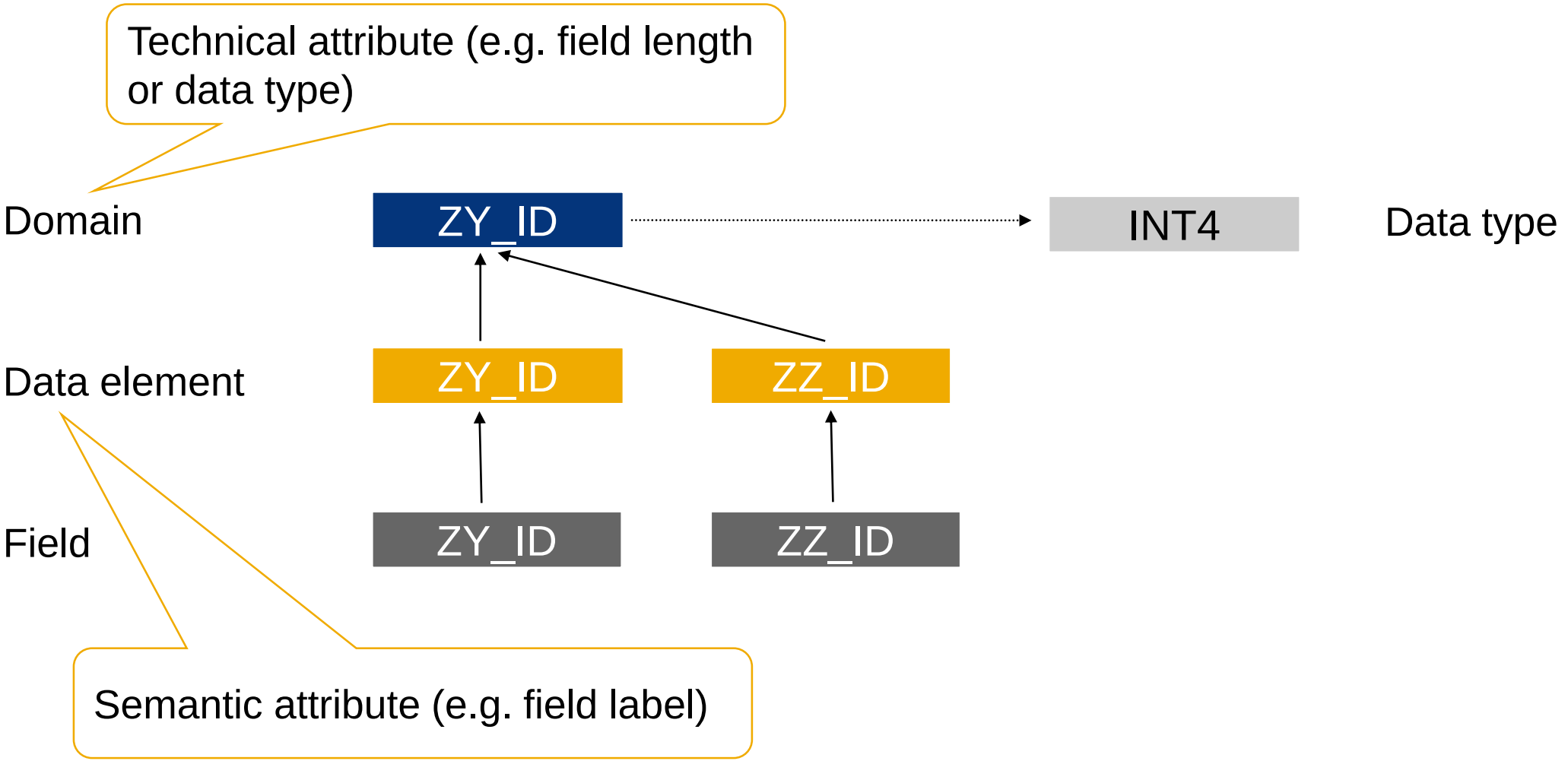
II. **SAP flight example**

III. **SAP OpenSQL**

What is the data dictionary?

- Data dictionary = global directory for data types
- Assignment of help texts and explanations for data types in different languages
- Entity–relationship model can be shown as a figure automatically
- Most important objects: structure, table, data element and domain

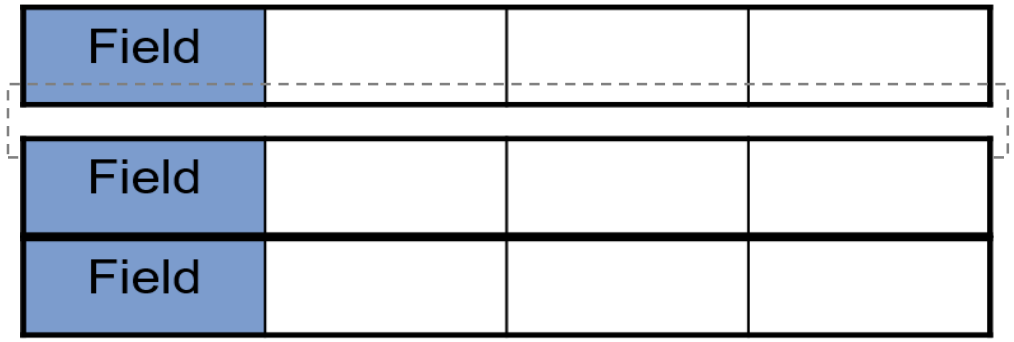
Hierarchy



Hierarchy

Structure

Table

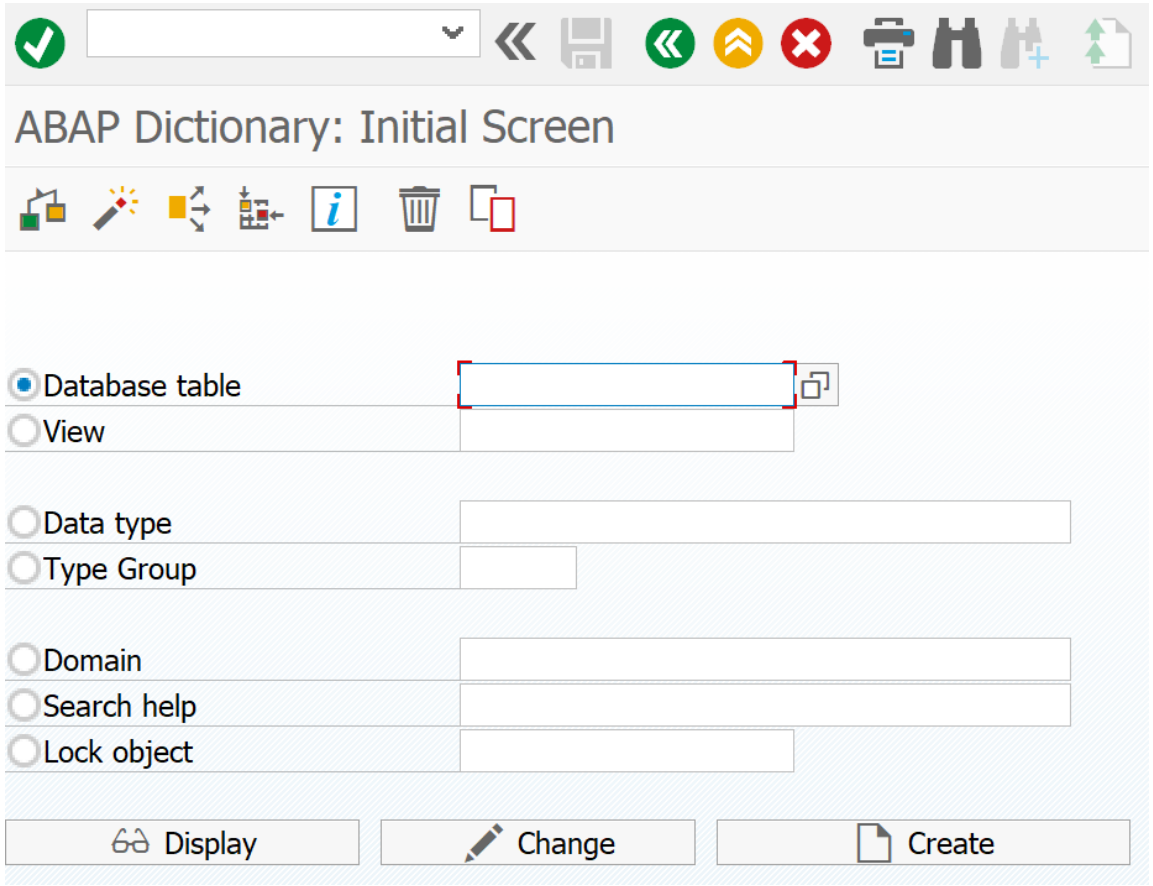


Working with the data dictionary

- Menu path: **Tools • ABAP Workbench • Development • Dictionary**
- Transaction code: SE11
- View, edit, delete, create tables, data types, domains definitions etc.
- Tables and views from the ABAP dictionary represent tables and views from the database
- User interfaces are generated automatically when changing views, tables etc.
- Table may be changed after they are created without losing data

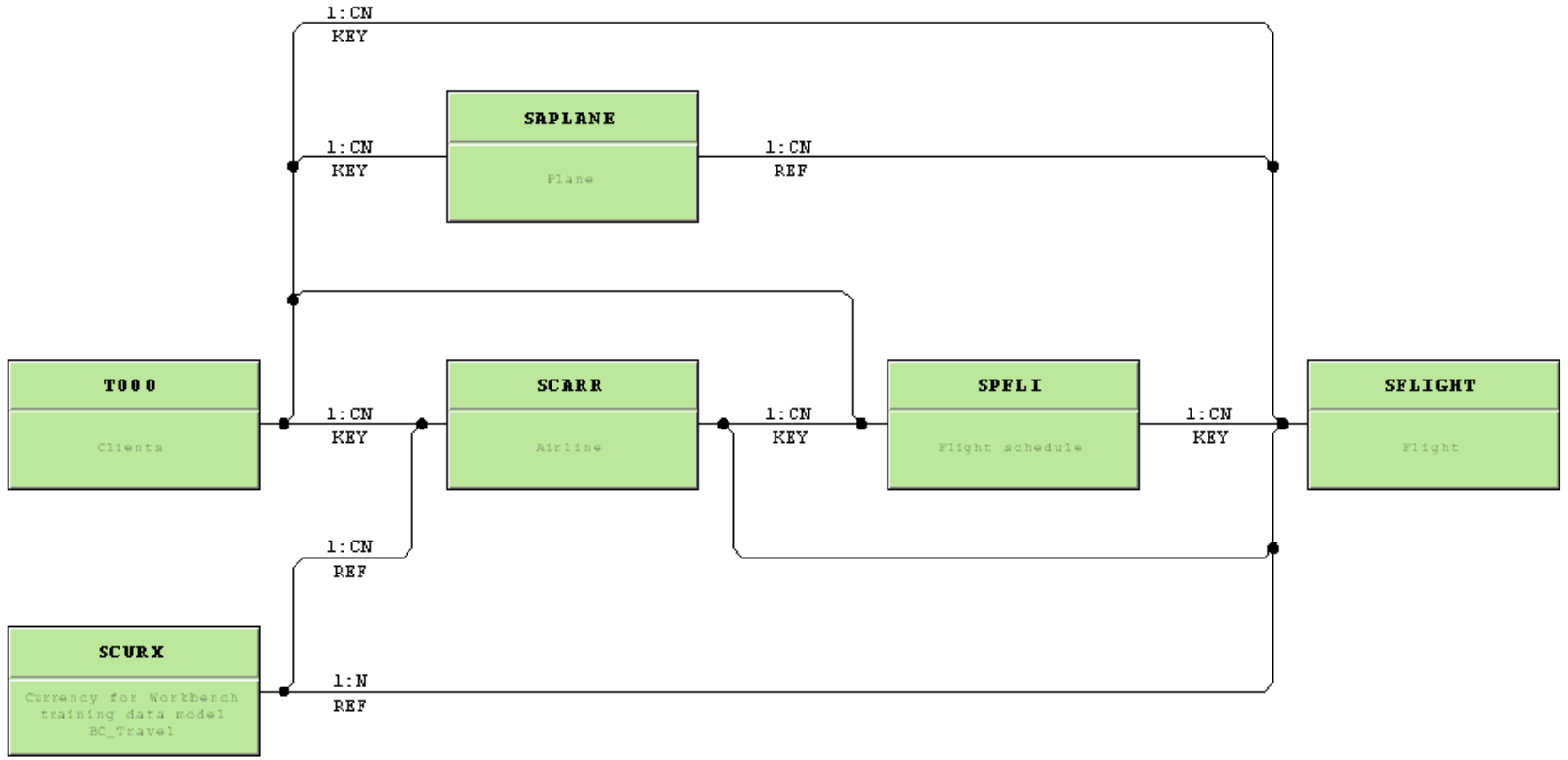
Overview about all objects

- Database table
- View
- Data type
  - Data element
  - Structure
  - Database table
  - Table types
  - Views
  - Class / interface
- Type group
- Domain
- Search help
- Lock object



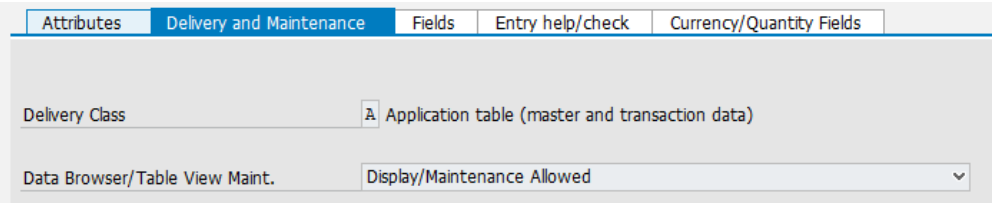
Visualization

- Foreign key relationships can be visualized in data dictionary
- Table SFLIGHT



Data browser

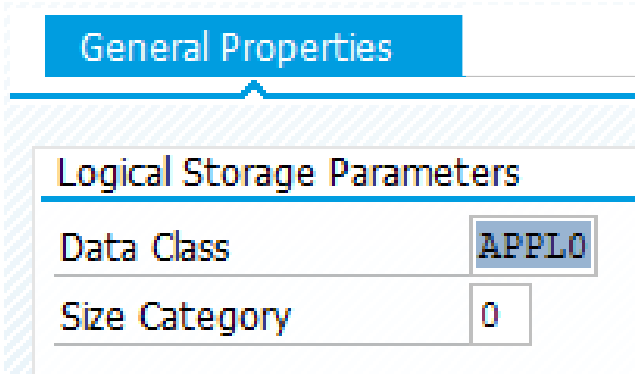
- Menu path: **Tools • ABAP Workbench • Overview • Data Browser**
- Transaction code: SE16
- **View table content**
- Add new entries to tables when adding is permitted for the table
  - It is possible to deny adding entries to a table. To verify if it is allowed please look at the delivery and maintenance settings in the data dictionary.



- Browse the entry help table content
- Copy table content to transport request
- Download table content

Technical settings: Data Class

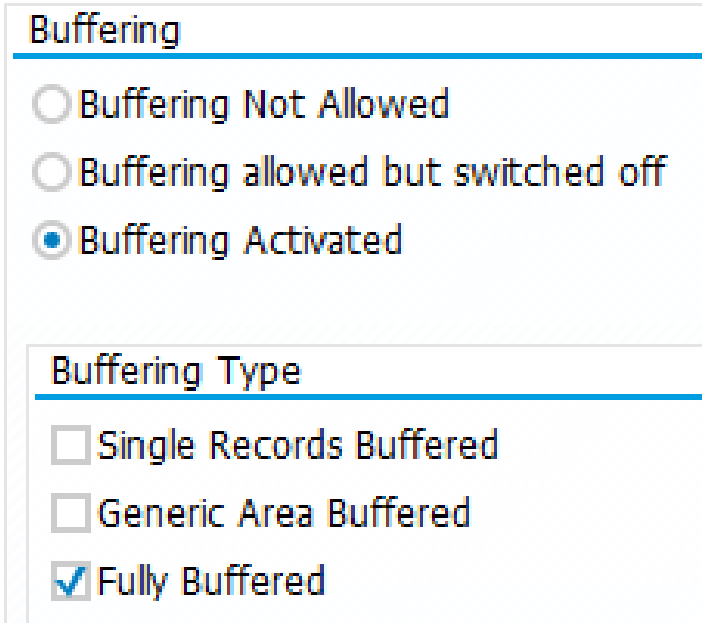
- SE11 -> Technical Settings



- **APPL0** *(master data):* data that is rarely changed, but frequently read
- **APPL1** *(transaction data):* data that is frequently changed
- **APPL2** *(organizational data):* data that is defined when the system is installed and rarely changed

Technical settings: Buffering

- SE11 -> Technical Settings



- **Buffering:** make data available in memory, so request for the same data can be processed faster
- **Full buffering:** Load all records of the table in the buffer, when one record is accessed
- **Generic buffering:** When a record of the table is accessed, all the records that have this record in the generic key fields are loaded in the buffer
- **Single-record buffering:** Only the records of a table that are really accessed are loaded into the buffer

Short introduction to the SAP flight example

- Created and maintained by SAP to demonstrate database operations
- Contains exercise data for airline, flight connection number, flight date, airfare etc.
- Report for data generation: `SAPBC_DATA_GENERATOR` (use Transaction SE38)
- SAP trainings, examples from books build refer to the flight example

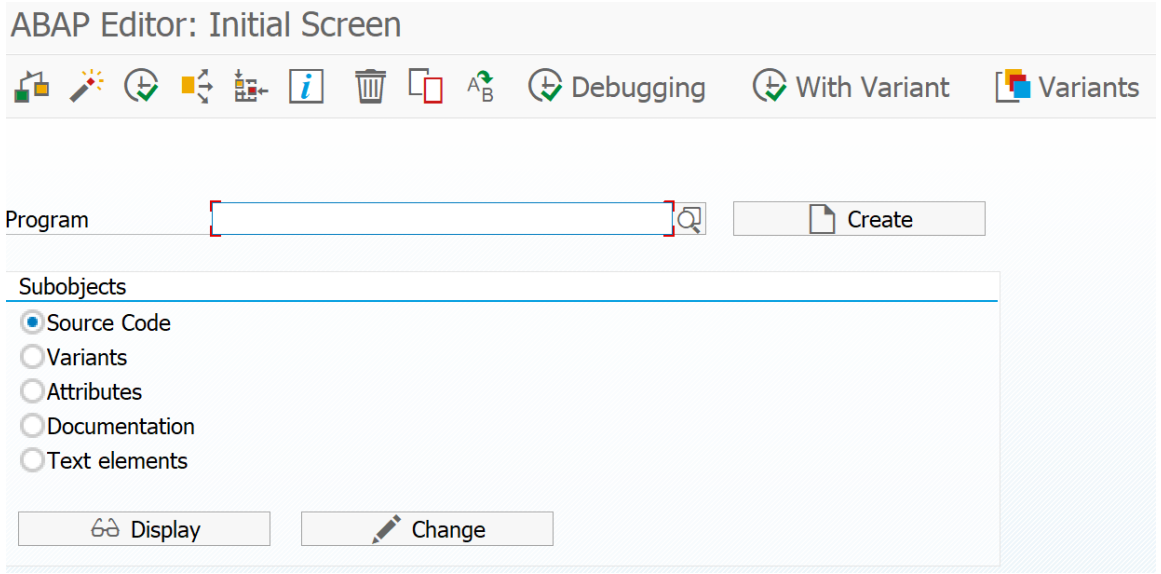
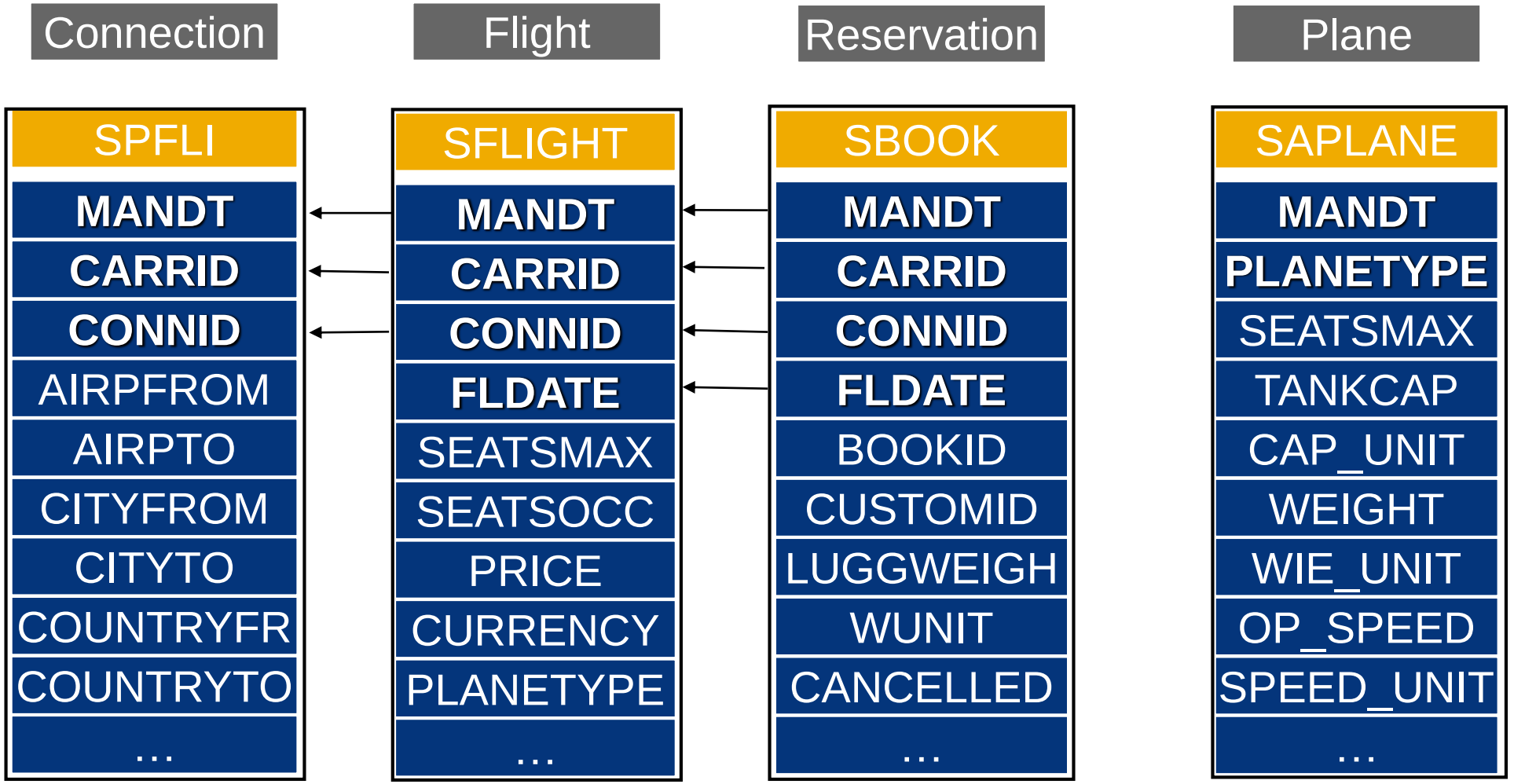


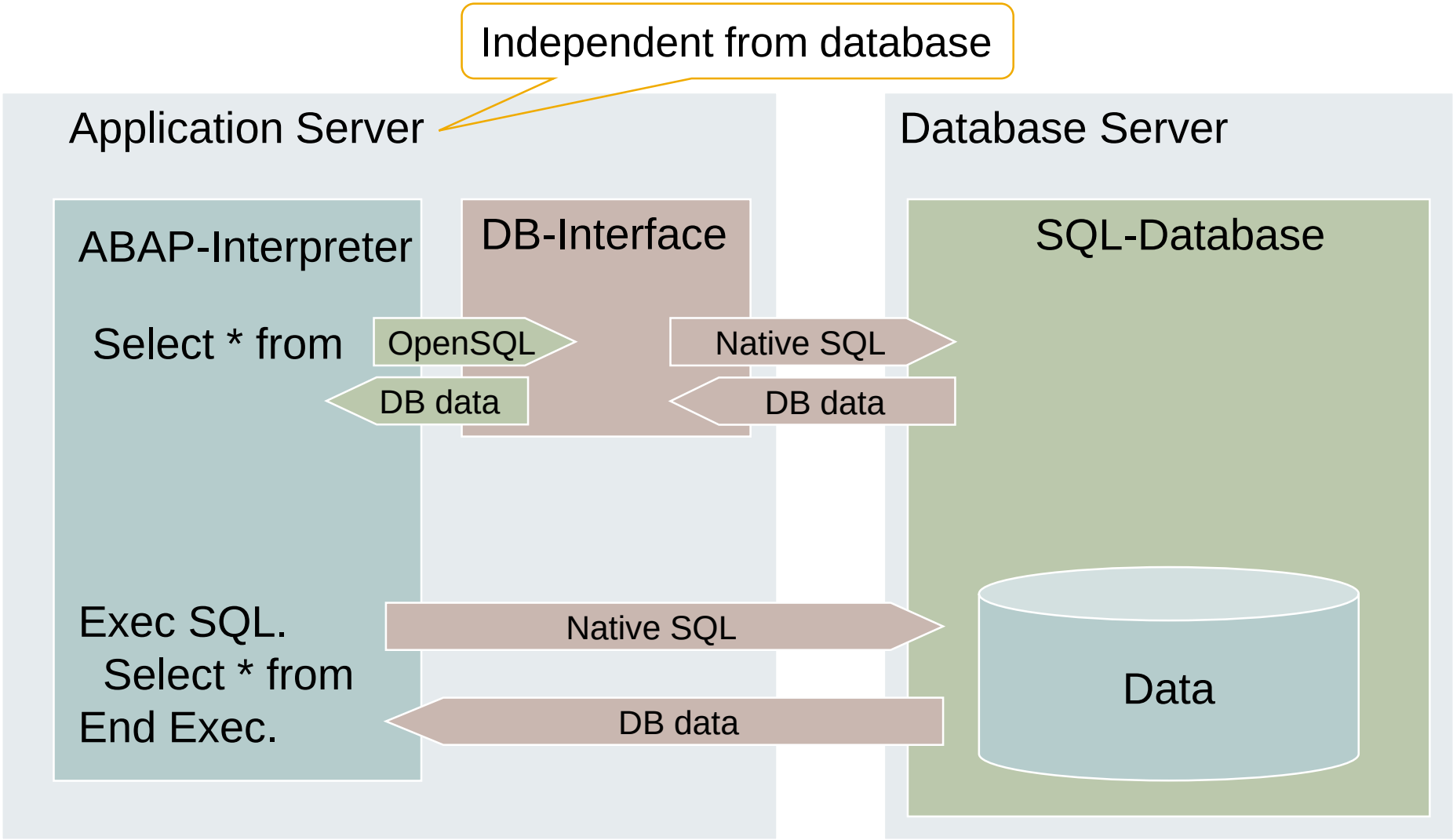
Table structure



Now you know about the data dictionary and the SAP flight example.

To consolidate your knowledge, you can do task 1 and 2 of the Database Access exercise.

OpenSQL vs. Native SQL



Instructions (examples)

- Select:

SELECT	*	FROM	table	INTO	structure	WHERE	condition
SELECT SINGLE	FIELD		view		internal table		

- Aggregation: SELECT MAX | SELECT MIN | SELECT COUNT

- Joins:
  - table1 INNER JOIN table2
  - table1 LEFT OUTER JOIN table2

Procedure for database access

General:

- Be specific to decrease the load on the database
- Access to database takes 10,000 time longer than access to buffers
- Avoid table scans, use indexes

Procedure:

- Be specific to decrease the load on the database

1. Read database data into internal tables

- Internal tables are tables for holding data during runtime

2. Change data per row

- Use workareas to hold one row of the table and change data in the workarea

3. Write changes back to database

Types of database changes

Modify instruction:

```
MODIFY <dbtable> [CLIENT SPECIFIED] FROM <workarea>.

MODIFY <dbtable> [CLIENT SPECIFIED] FROM TABLE <internaltable>
```

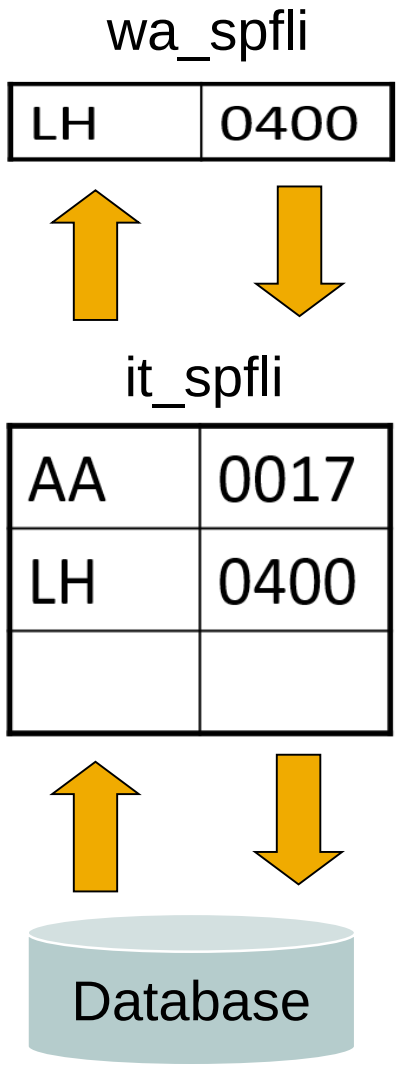
a) Modify = Update – the command updates the database table if the database table already contains dataset

b) Modify = Insert – the command inserts new datasets into the database table

Delete instruction:

```
DELETE FROM <dbtable> WHERE [SQL statement].
```

Procedure for database access



Workarea
(one row)

- Use the workarea to modify one data set

Internal table
(selected content)

- Use the internal table to store the selected database content temporarily
- Internal tables are deleted after program is finished

Database table
(whole content)

- Use the database to read/write data

Indexes

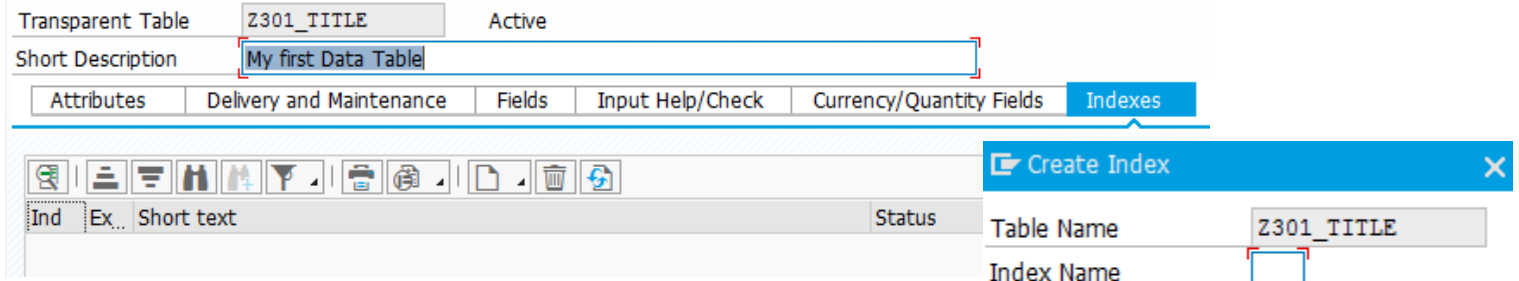
General:

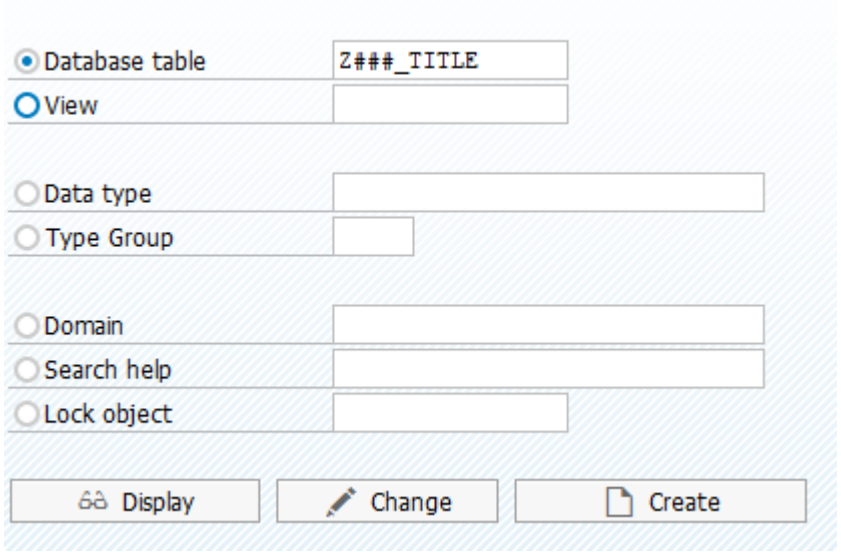
- Indexes (secondary keys) can further improve the efficiency of search operations
- Index: database object that can be defined for an individual column or a series of columns in a database table
- Consists of one or more inversion lists (set of non-overlapping numeric ranges)
- Indexes provide access to the table data via non-key columns
- Necessary if the table is frequently used without taking advantage of the primary key

Create index (secondary key)

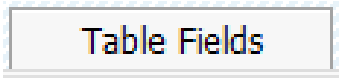
Step-by-Step:

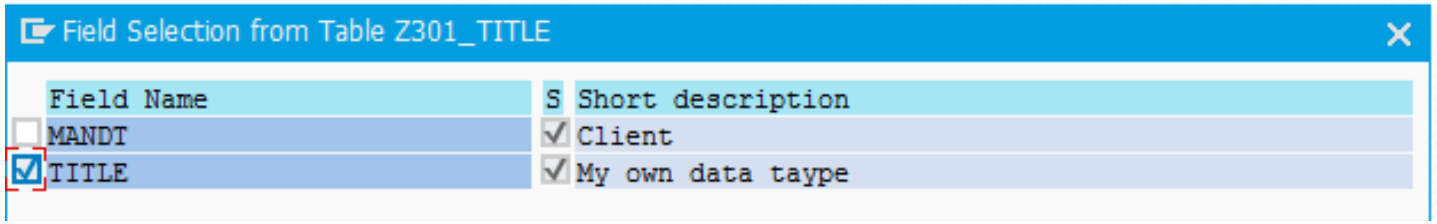
- Go to SE11, enter required Database, press change
- Switch to the tab indexes, click create -> create index





- Choose an index name
- Select the fields the index should comprise





Now you know about SAP OpenSQL.

To consolidate your knowledge, you can do task 3 and 4 of the Database Access exercise.

Check your knowledge

Check your knowledge

- You can use the data dictionary to add entries to a table manually.

  ☐ True ☐ False

- Explain a reason not to use Native-SQL in favor of OpenSQL (think about an example scenario where Native-SQL is problematic).

- Why is the database data read into internal tables

  ☐ To save memory ☐ To accelerate the application ☐ To avoid high disk usage

Solution

Check your knowledge

- You can use the data dictionary to add entries to a table manually.

  ☐ True ☒ False

- Explain a reason not to use Native-SQL in favor of OpenSQL (think about an example scenario where Native-SQL is problematic).

  See slide 13 - OpenSQL vs. Native SQL

- Why is the database data read into internal tables

  ☐ To save memory ☒ To accelerate the application ☐ To avoid high disk usage

© 2018 SAP SE or an SAP affiliate company. All rights reserved.