

Table of Contents

Behind a well-designed database application, there is always a well-designed data model. This model ideally fulfills the third NF, or at least a reasonable medium between second and third NF. In the standard case, the user interface visualizes the data model and allows the collection of master- and transaction data.

However, the database offers many advantages that we can apply to the user interface without wasting much time. One of these advantages is the “default value” of columns in tables.

Frequently, so-called flag columns exist, which have a default value configured in the database. Examples here are the Y/N (Yes/No) columns. The default value can be N (No).

During the creation of a dataset, the developer would have to consider these default values in the user interface and preset them for the user. This is a repetitive and error-prone activity. Besides, the developer would surely rather spend time on challenging problems than on boring tasks!

This is where JVx comes in handy. The framework recognizes the default values of columns automatically and assumes the defined values during the creation of new datasets directly in the user interface. It should be noted, however, that only constant default values are considered since function calls are always database-dependent and could contain logic.

Example

Our application contains user administration for the creation and administration of application users. The underlying database table was created as follows (Oracle Syntax):

[create.sql](#)

```
CREATE TABLE USERS
(
  ID                NUMBER(16) NOT NULL,
  USERNAME          VARCHAR2(200) NOT NULL,
  PASSWORD          VARCHAR2(200),
  CHANGE_PASSWORD  CHAR(1) DEFAULT 'N' NOT NULL,
  ACTIVE            CHAR(1) DEFAULT 'Y' NOT NULL,
  VALID_FROM       DATE,
  VALID_TO         DATE,
  CREATED_BY       VARCHAR2(200) NOT NULL,
  CREATED_ON       DATE DEFAULT sysdate NOT NULL,
  CHANGED_BY      VARCHAR2(200),
  CHANGED_ON      DATE,
  TITLE            VARCHAR2(64),
  FIRST_NAME       VARCHAR2(200),
  LAST_NAME        VARCHAR2(200),
  EMAIL            VARCHAR2(200),
  PHONE            VARCHAR2(200),
  MOBILE           VARCHAR2(200)
);

-- Create/Recreate primary, unique and foreign key constraints
ALTER TABLE USERS
  ADD CONSTRAINT USER_PK PRIMARY KEY (ID);
```

```
ALTER TABLE USERS
  ADD CONSTRAINT USER_UK UNIQUE (USERNAME);
```

We now need a server object for access to the database or table:

```
public DBStorage getUsers() throws Exception
{
    DBStorage users = (DBStorage)get("users");
    if (users == null)
    {
        users = new DBStorage();
        users.setDBAccess(getDBAccess());
        users.setWritebackTable("USERS");
        users.setDefaultSort(new SortDefinition("USERNAME"));
        users.open();

        put("users", users);
    }
    return users;
}
```

At the client, we now create an object for server access:

```
rdbUsers.setDataSource(dataSource);
rdbUsers.setName("users");
rdbUsers.open();
rdbUsers.getRowDefinition().setTableColumnNames(
    new String[] {"USERNAME", "ACTIVE",
"CHANGE_PASSWORD"});
```

If we now create a new user, the fields "ACTIVE" and "CHANGE_PASSWORD" are automatically filled with the default values from the database. The field "ACTIVE" gets the value "Y" and "CHANGE_PASSWORD" gets the value "N".

The following methods can be used to ignore the default values:

```
//per instance
users.setDefaultValue(false)

//for all instances (static)
DBStorage.setDefaultDefaultValue(false)
```

If no default values were defined in the database, they can also be set via the API:

```
users.open();

//sets the default value to "X"
users.getMetaData().getColumnMetaData("ACTIVE").setDefaultValue("X");
```

If default values are not used, we can achieve the default setting via the use of client side events, such as:

```
rdbUsers.eventAfterInserting().addListener(this, "afterInserting");  
...  
public void afterInserting(DataBookEvent pEvent) throws Exception  
{  
    pEvent.getChangedDataBook().setValue(COLUMN, VALUE);  
}
```

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

http://doc.sibvisions.com/jvx/server/storage/dbdefault_values



Last update: **2020/06/29 13:31**