



**Database Management Systems Laboratory
Experiments Phase 3**

Subject: Develop your desktop database management module

**Ekrem Candemir
21327771**

**Berk Kemal Özata
21328653**

Oracle's Data Dictionary

In Oracle databases, "Data Dictionary" term refers to system catalogs. It is the repository of all the meta-data of objects stored in the database. Meta-data is defined as information about information or data about data. The objects stored in the database are tables, views, stored procedures, constraints etc. Data dictionary also includes information about DBMS.

The data dictionary consists of base tables and views. Base tables store information about the database. Users rarely access the base tables because data is stored in a cryptic format. Views are useful information obtained by decoding from base tables.

Data dictionary views are divided into three sets:

- Views with the prefix **DBA_** : Only database administrators can access. It contains all objects in all users' schemas.
- Views with the prefix **ALL_** : It contains all objects that current user can access.
- Views with the prefix **USER_** : It contains objects in the current user's schema.

These views are called static data dictionary views. There is also dynamic views with the prefix **V_\$** that dynamically monitors database activities. They are available only to administrators.

In conclusion, we can use these views to manage the database objects according to our access levels. We can access names of tables and their owners, names of indexes and their related columns, constraints including primary keys, foreign-keys and not-null constraints. Database management systems need these structures like data dictionary to access the data within a database. It can only write or read the data to the database using data dictionary created for that particular database.

In this phase, the views that interested us were **ALL_** and **USER_** views. We have used **USER_** views, because the objects we need to access are belong only to the current user.

Example Usages in Our Application

SELECT table_name FROM user_tables

This query returns all names of the current users' tables.

```
SELECT A.COLUMN_NAME, A.DATA_TYPE, A.NULLABLE, B.CONSTRAINT_TYPE
FROM (
    SELECT usrcols.COLUMN_NAME, usrcols.DATA_TYPE, usrcols.NULLABLE
    FROM user_tab_cols usrcols
    WHERE usrcols.table_name = 'SYSTEMUSER'
    ORDER BY usrcols.column_id
) A
LEFT OUTER JOIN (
    SELECT cols.column_name, cons.constraint_type
    FROM user_constraints cons, user_cons_columns cols
    WHERE cols.table_name = 'SYSTEMUSER'
    AND cons.constraint_name = cols.constraint_name
)
B ON A.COLUMN_NAME = B.COLUMN_NAME;
```

We are experiencing slowness while bringing out the column details, after we added constraint types whether related column is primary and/or foreign key. We get the column details by joining two sub-queries. We think that the performance problem is caused by this.

This query returns information about columns in a table. It joins two sub-query, first one consists of column_name, data_type and nullable values while the second one consists of column_name and constraint_name. We apply left outer join on column names so that columns without constraints are also be returned in this query.

```
SELECT
    src_tbl.TABLE_NAME, cons_col.COLUMN_NAME
  FROM user_constraints src_tbl, user_constraints ref_tbl,
user_cons_columns cons_col
 WHERE ref_tbl.constraint_type = 'R'
    AND src_tbl.constraint_name = ref_tbl.r_constraint_name
    AND cons_col.constraint_name = ref_tbl.r_constraint_name
    AND ref_tbl.table_name = 'SYSTEMUSER';
```

This query finds the table name and column name where foreign keys in a table belong to. user_constraints table is used for accessing the constraint type and user_cons_columns table is used for accessing column names of the reference type constraints. We used this query to retrieve records from where the foreign key belongs to, so that we let users to select only these values using combobox.

Application

In this phase we are expected to develop a desktop application. We should write statistics about this database such as the numbers of the tables, the names of the tables, and the number of records in the tables. At the same time, we are expected to use DML phrases like insert, update, delete in the application. The application will consist of 3 windows in general.

The first window is the login screen(Figure 1). In this application we developed, the login screen consists of 3 text fields. The first is database url, the second is username, and the third is password. If at least one information the user has written is wrong, the program would prompt the user to re-enter.

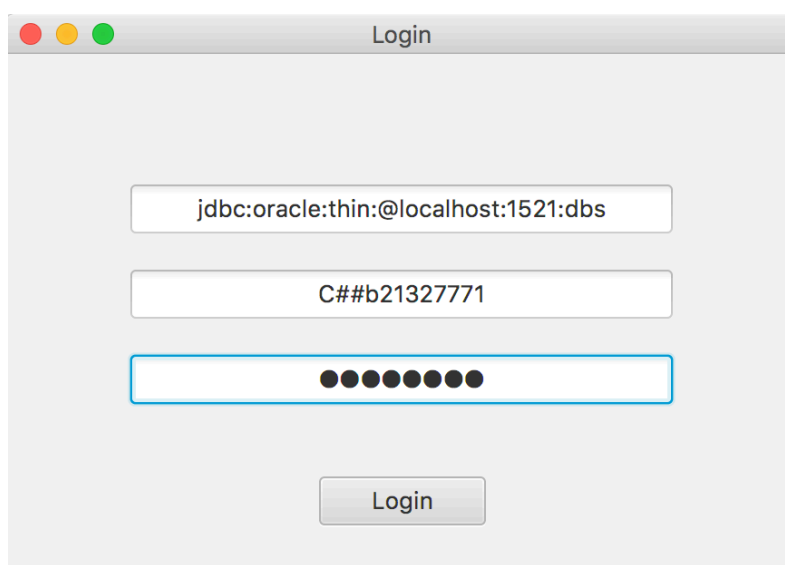


Figure 1

Number of Tables		22
Table Name	Number of Rows	
AUTHOR	12	
BOOKAUTHOR	10	
PUBLISHER	17	
CATEGORYTBL	9	
CATEGORYINHERITANCE	2	
FILETBL	25	
BOOKCATEGORY	16	
EBOOK	17	
AUDIOBOOK	6	
LIBRARYTBL	20	
SHELF	11	
SHEFFILE	10	
ROLETBL	1	
PERMISSIONTBL	2	
ROLEPERMISSION	0	
USERTBL	14	
SYSTEMUSER	2	
CONTACT	0	
COMMERCIALUSER	10	
PAYMENTDETAIL	10	
BOOK	10	

Figure 2

[illegible][illegible]

Figure 3

In the second window named Data Records, the information of the records of the selected table in the left column is displayed. There are also 3 buttons you can use in this window to use DML phrases (Figure 4). A separate pop-up window opens when you click each button.

[illegible]

Figure 4

Number of Tables			User Tables						
22			<div>Column Details</div> <div>Data Records</div>						
Table Name	Number of Rows		Column Name	Column Type	Nullable	Is Primary	Is Foreign		
AUTHOR	12		CATEGORYID	NUMBER	N	True	False		
BOOKAUTHOR	10		CATEGORY_NAME	VARCHAR2	N	False	False		
PUBLISHER	17		INSERT_USER	NUMBER	N	False	True		
CATEGORYTBL	9		LAST_UPDATE_USER	NUMBER	N	False	True		
CATEGORYINHERITANCE	2		INSERT_DATE	DATE	Y	False	False		
FILETBL	25		LAST_UPDATE_DATE	DATE	Y	False	False		
BOOKCATEGORY	16								
EBOOK	17								
AUDIOBOOK	6								
LIBRARYTBL	20								
SHELF	11								
SHELFFILE	10								
ROLETBL	1								
PERMISSIONTBL	2								
ROLEPERMISSION	0								
USERTBL	14								
SYSTEMUSER	2								
CONTACT	0								
COMMERCIALUSER	10								
PAYMENTDETAIL	10								
BOOK	10								

Figure 5 - Full view of the main screen

When the Insert button is clicked, a pop-up window opens to add a new record to that table (Figure 6). This window contains the column names of the table and the text fields needed to add a new record. Insert_user and last_update_user fields have a ComboBox (Figure 6.1). DatePicker exists in the date fields for convenience (Figure 6.2). The user clicks Insert Row. After pressing this button, if the operation is successful, a pop-up window opens and says that “Insert Operation Successful!” (Figure 9), otherwise it says “Insert Operation Failed!”. (Figure 9.1). If the user closes this pop-up, the user sees the updated version of the table.

Insert Row

BOOKID

BOOKNAME

BOOK_SUMMARY

COVER_IMAGE

INSERT_USER

LAST_UPDATE_USER

INSERT_DATE

LAST_UPDATE_DATE

Insert Row

Figure 6

INSERT_USER

LAST_UPDATE_USER

1

2

Figure 6.1

BOOKNAME

BOOK_SUMMARY

COVER_IMAGE

INSERT_USER

LAST_UPDATE_USER

INSERT_DATE

LAST_UPDATE_DATE

< April > < 2017 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Figure 6.2

When the Update button is clicked, a pop-up window opens with the informations on the selected line(Figure 7). The user updates what he/she wants to update from these lines and clicks the Update Row button. After pressing this button, if the operation is successful, a pop-up window opens and says that “Update Operation Successful!”(Figure 9), otherwise it says “Update Operation Failed!”.(Figure 9.1). If the user closes this pop-up, the user sees the updated version of the table.

The 'Update Row' dialog box displays the following information:

Field	Value
BOOKID	1
BOOKNAME	The Two Towers
BOOK_SUMMARY	The Fellowship was scattered.
COVER_IMAGE	twotowers.jpg
INSERT_USER	2
LAST_UPDATE_USER	2
INSERT_DATE	2017-04-09 12:51:11.0
LAST_UPDATE_DATE	2017-04-09 12:51:11.0

Update Row

Figure 7

The last button is the Delete button. When this button is clicked, a pop-up window opens with informations on the line to be deleted(Figure 8). The only thing a user needs to do to delete a record is to press the Delete Row button. if the operation is successful, a pop-up window opens and says that “Delete Operation Successful!”(Figure 9), otherwise it says “Delete Operation Failed!”(Figure 9.1). If we close this pop-up, we see the updated version of the table.

The 'Delete Row' dialog box displays the following information:

Field	Value
BOOKID	1
BOOKNAME	The Two Towers
BOOK_SUMMARY	The Fellowship was scattered.
COVER_IMAGE	twotowers.jpg
INSERT_USER	2
LAST_UPDATE_USER	2
INSERT_DATE	2017-04-09 12:51:11.0
LAST_UPDATE_DATE	2017-04-09 12:51:11.0

Delete Row

Figure 8

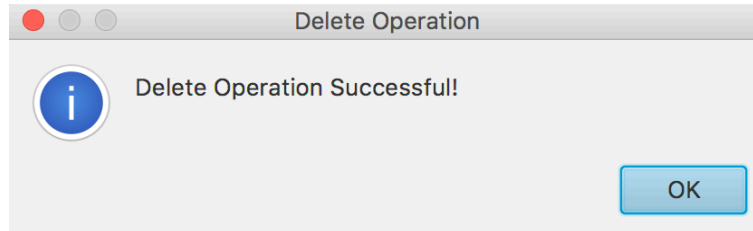


Figure 9 - Successful Operation Example

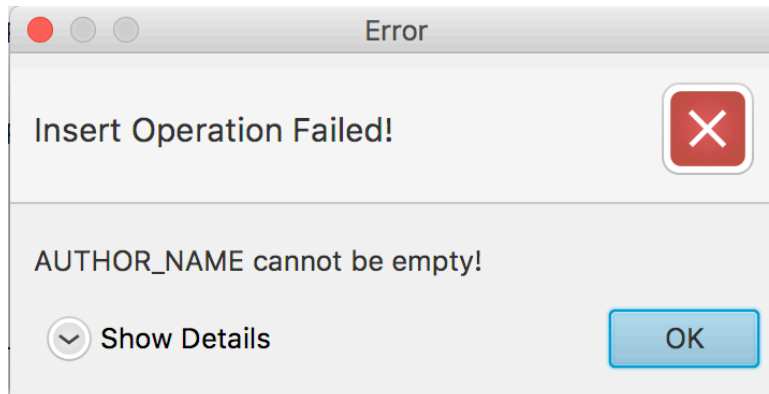


Figure 9.1 - Unsuccessful Operation Example

References

- 1) <https://docs.oracle.com/>
- 2) <https://www.techopedia.com/definition/27752/data-dictionary>
- 3) <http://w2.syronex.com/jmr/edu/db/oracle-data-dictionary/>
- 4) <https://www.techonthenet.com/>