Advanced Data Base
(8trd157)

Lab2 *(project: phase 2 of 4)*

Logical Data Model Design using Oracle 11g

(1 week) Paul Girard, Ph.D.

Specific Objectives of Lab2

- Apply some quantitative constraints
- Test a connection to the database **cndb** using **sqlplus** server with **SSH** *(putty)* on dimensxcn1.uqac.ca
- Translate all your conceptual data models *(global & partials)* to a normalized relational form
- Create and load your tables with SQL*Loader
- Test your tables with Oracle SQL using **sqlplus.** Test each user transaction.
- Create all necessary partial views to support your user transactions

Description

After some analysis, the following constraints have been observed :

- The store has an average of 10,000 parts in stock ; 10 purchasing agents work for the company and each one is responsible for 1,000 parts .
- Purchase orders may be sent by an agent to any of the 200 suppliers. Each purchasing agent prepares an average of 10 purchase orders per day.
- Each supplier may supply an average of 1000 products to this company. Those products may interest more than one purchasing agent.
- An average of 5 different products are purchased on each purchase order.
- The unit price of a part is modified by each purchase order containing a product for that given part. The product unit from the supplier is an *integer [declared as* ***number(4)*** *in Oracle]* and the local unit part is *alphanumeric [****char(15)****in Oracle]*.

Methodology

1. **Test the connection with sqlplus** *(server)* **to the primary database cndb under Oracle 11g with the server dimensxcn1.uqac.ca**

- Use putty with SSH to access dimensxcn1.uqac.ca to test **cndb** using your account ora***** with Oracle 11g.
- The same Unix user name and password has been used to create the Oracle user name and password
- If you change your password under Unix *(dimensxcn1.uqac.ca)* with the command *passwd* do the same with your Oracle password using the command **alter user** under sqlplus
- Do a connection to *cndb* in your account with *sqlplus.*

```
[pgirard@dimensxcn1]$ sqlplus
  SQL*Plus: Release 11.2.0.3.0 Production on Thu Mar 29 15:39:28 2012
  Copyright (c) 1982, 2011, Oracle.  All rights reserved.

  Enter user-name: pgirard
  Enter password:
  Connected to:
  Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
  With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

Optional ➔

- check the memory used by SGA of *cndb*
- display the attributes of the table *dba_tables,*
- display the number of tables owned by the Oracle data dictionary *(dba_tables),*
- display the number of users defined in the database *(dba_users),*
- display the name and the status of *tablespaces* defined in *cndb*
- close the connection with the command *disconnect* and quit with *exit.*

Example of a session under Oracle using the database *cndb (see annex 1)*

```
[pgirard@dimensxcn1]$  sqlplus
Enter user-name: pgirard
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SQL> alter user pgirard identified by new_password ;
SQL> show sga;
SQL> desc dba_tables;
SQL> select count(*) "Number of tables"  from dba_tables;
SQL> select count(*) "Number of users" from dba_users;
SQL> select tablespace_name, status from dba_tablespaces;
SQL> select file_name from dba_data_files;
SQL> disconnect;
SQL> exit

[pgirard@dimensxcn1 $
```

2. Creation of tables and views needed by your logical data model under Oracle 11g

By using an appropriate editor, supporting FTP open and save, like *EditPlus,* create a new file *(ex. crelab2.sql)* to create all tables, views *(see step 4)*, index and constraints of your logical data model. Pay attention to each attribute data format, default and null option. If you judge that some tables should have an index based on a primary key, it is better to wait after the data loading *(more efficient this way).* If the attribute of a table is a foreign key, specify it with a constraint *(references).* If the table has a primary key based on a single or multiple attributes specify it as a column or a table constraint respectively. This file will have to be executed many times, so do not forget to drop each table and view before the creation and <u>in the inverse order</u> to respect the integrity constraints as shown in the following example.

note: *It is much easier to test the creation of tables first, then insert the views, load data into tables* ***and only after*** *include the constraints and restart the creation and loading*

cretab.sql

```
drop table employee;
drop table department;
create table department
          (name_dept       char(10)         primary key,
          no_dept          number(2)        unique,
          tel_dept         char(8)          default '545-5011',
          budget           number(7)        default 20000);
create table employee
          (emp_number      number(5)        primary key,
          name_emp         char(15)         not null,
          addr             char(25)         default 'Saguenay',
          commission       number(10),
          salary           number(5)        default 0,
          dept_name_       char(10)         references departement(name_dept));
create index emp_num_idx on employee(emp_number);
```

- An execution example of *cretab.sql* with **sqlplus** will show these results
  (note: sqlplus assumes a default extension of .sql to a file)

```
SQL>  @cretab
Table dropped.
Table dropped.
Table created.
Table created.
Index created
SQL>
```

3. Use the server approach with putty(SSH) to create and load all base tables with SQL*Loader in your user schema *ora*****

- From my web site, use **module 5 example** ***(****p.9, p.19-36)* to create your tables, views, constraints & index and **module 6** ***(****p.57-60)* to load your tables with *SQL*Loader method 2.*

- Proceed one table at a time; for example, the table part *(already created)* will need 2 files : part.ctl and part.dat. Then use the command **sqlldr** to test it. After execution the file part.bad will be created if there is something wrong and the file part.log gives the result of *sqlldr*. Verify each table data with a **select * from *table****;*

- With the same method, create approximately 3 purchasing agents, 6 parts under the responsability of different purchasing agents; at least one part will have at least 2 components and one component will be used by at least 2 parts. Create also 2 suppliers with some products having their product unit > 2 and a different name than the name of part. Create 2 purchase orders, each one created by a different purchasing agent. You will then have twice the number of files **.ctl** and **.dat** than the number of tables. If you have a character type data, use the delimeter apostrophe. (example 'motor 425')

- When each loaded table has been verified by a select, execute again crelab2.sql and load them once with **loadlab2** having all sqlldr commands *(one line per table).* **Annex 2** gives an example of the sequence. Under Unix, make **loadlab2** executable with the command **chmod**
  [pgirard@dimensxcn1]$ **chmod 700 loadlab2** and execute with

```
[pgirard@dimensxcn1]$ ./loadlab2
```

example of *loadlab2* for the tables *part* and *component*

```
sqlldr userid=ora*****/password control=part log=part
sqlldr userid=ora*****/ password control=component log=component
```

example of an execution of *loadlab2*

```
dimensxcn1:ora11001>chmod 700 loadlab2
dimensxcn1:ora11001>./ loadlab2
```

- Create also a file **initab.sql** to initialize all your tables with **sqlplus** without dropping them.

example of initab for tables *part* and *component*

```
delete from part;
delete from component;
```

execution of *initab* with sqlplus

```
SQL>   @initab
```

- Connect to the database with *sqlplus*, display all your tables and test some user transactions to validate your data model.

example of *explosion.sql* displaying the components of part_id 1001

```
SQL> @explosion

Part ID of Component                Part name
---------------------------         ---------------
1003                                'piles AA'
1005                                'alternateur 1'
```

4. Creation of partial views for each user type

- Create all necessary partial views for each one of the 5 different user types for security and confidentiality reasons. Each user type must only access the necessary attributes of each table. If a user must access all attributes of a table, he will use the complete table, if this is not the case, create a view with an appropriate name to be recognized *(base table – user type).* The command **create view** is used to do so. *(see module 5)*

  For example this command creates the partial view **part_sks** for the store keeper supervisor.

```
create view part_sks as
    select qte_mag, unit_cost from part;
```

- Add those commands to the file **crelab2.sql** and do not forget to drop them at the beginning with the command **drop view ******

- Keep your documentation up to date by adding a new section called Logical Data Model under Oracle 11g Environment. This section will contain the following pages :

  - one page for your global relational logical data model for Oracle *(list your tables line by line);* do not forget to underline the primary key.

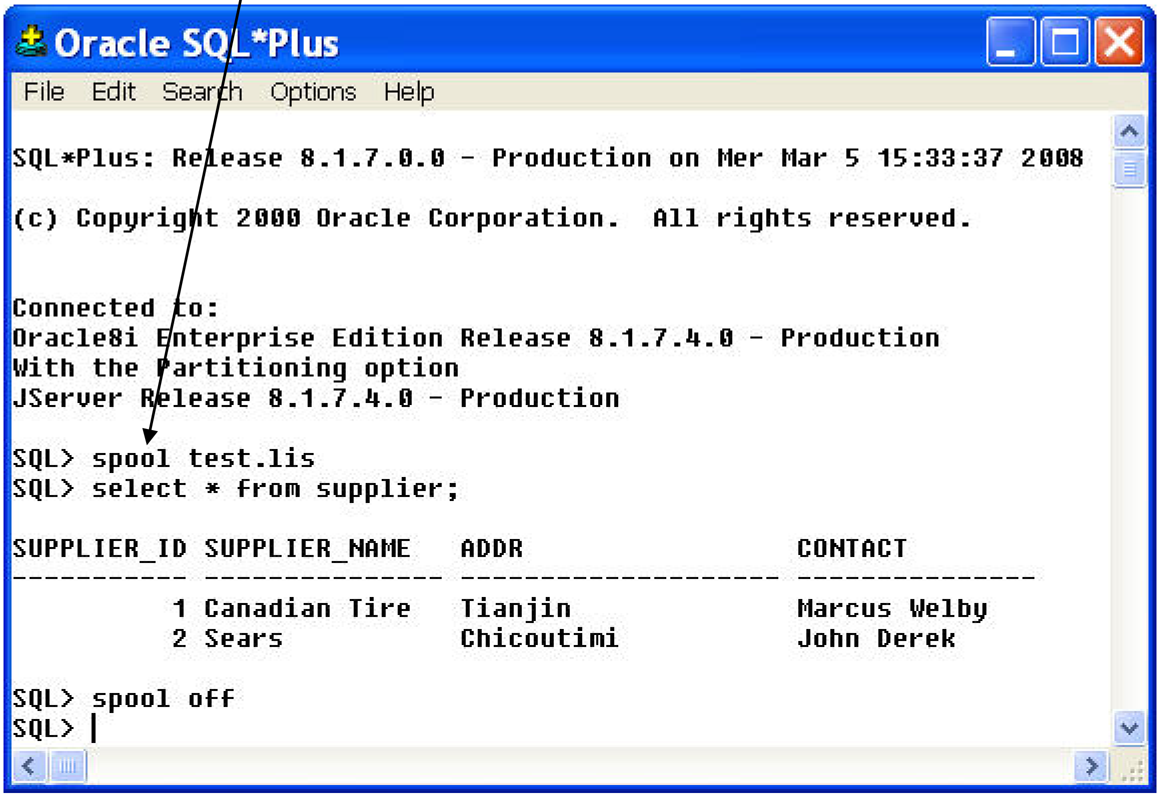
- one page for the relational global data model shown on a graphical form with Visio showing the commons attributes between tables. Do not forget that there is in NO RELATION in a relational data model. Use the connector line *(already downloaded from the web site)* with Visio to connect all corresponding attributes.
- one page to display the content of the command file **cretab2.sql** showing the creation of tables and views and an example of its execution with sqlplus server *(or sqlplus client using the spool).*
- all necessary pages showing all **.ctl** et **.dat** files for all table
- one page to display the contents of **loadlab2** et **initab.sql** and an example of its execution with sqlplus server *(or sqlplus client using the spool).*
- and one page for each partial data model using a graphical form showing all views and tables for each one of the 5 user types.

note: The command "spool" of **sqlplus** records a session in a file and gives the possibility to copy and paste the contents in the documentation.

SQL> spool *filenamer*	to record a session
SQL> spool off	to stop the recording

Example

(report1.lis can be accessed after the SQLPLUS session)



```
SQL*Plus: Release 8.1.7.0.0 - Production on Mer Mar 5 15:33:37 2008

(c) Copyright 2000 Oracle Corporation.  All rights reserved.


Connected to:
Oracle8i Enterprise Edition Release 8.1.7.4.0 - Production
With the Partitioning option
JServer Release 8.1.7.4.0 - Production

SQL> spool test.lis
SQL> select * from supplier;

SUPPLIER_ID SUPPLIER_NAME   ADDR                  CONTACT
----------- --------------- --------------------- ---------------
          1 Canadian Tire   Tianjin               Marcus Welby
          2 Sears           Chicoutimi            John Derek

SQL> spool off
SQL> |
```

==> The complete documentation *(pdf or html)* for Oracle 11g is on my web site

Annex 1
Connection to a database to check some Oracle system tables

```
SQL> show sga

Total System Global Area  239198368 bytes
Fixed Size                    73888 bytes
Variable Size              81657856 bytes
Database Buffers          157286400 bytes
Redo Buffers                 180224 bytes

SQL> desc dba_tables;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 TABLE_NAME                                NOT NULL VARCHAR2(30)
 TABLESPACE_NAME                                    VARCHAR2(30)
 CLUSTER_NAME                                       VARCHAR2(30)
 IOT_NAME                                           VARCHAR2(30)
 PCT_FREE                                           NUMBER
 PCT_USED                                           NUMBER
 INI_TRANS                                          NUMBER
 MAX_TRANS                                          NUMBER
 INITIAL_EXTENT                                     NUMBER
 NEXT_EXTENT                                        NUMBER
 MIN_EXTENTS                                        NUMBER
 MAX_EXTENTS                                        NUMBER
 PCT_INCREASE                                       NUMBER
 FREELISTS                                          NUMBER
 FREELIST_GROUPS                                    NUMBER
 LOGGING                                            VARCHAR2(3)
 BACKED_UP                                          VARCHAR2(1)
 NUM_ROWS                                           NUMBER
 BLOCKS                                             NUMBER
 EMPTY_BLOCKS                                       NUMBER
 AVG_SPACE                                          NUMBER
 CHAIN_CNT                                          NUMBER
 AVG_ROW_LEN                                        NUMBER
 AVG_SPACE_FREELIST_BLOCKS                          NUMBER
 NUM_FREELIST_BLOCKS                                NUMBER
 DEGREE                                             VARCHAR2(10)
 INSTANCES                                          VARCHAR2(10)
 CACHE                                              VARCHAR2(5)
 TABLE_LOCK                                         VARCHAR2(8)
 SAMPLE_SIZE                                        NUMBER
 LAST_ANALYZED                                      DATE
 PARTITIONED                                        VARCHAR2(3)
 IOT_TYPE                                           VARCHAR2(12)
 TEMPORARY                                          VARCHAR2(1)
 SECONDARY                                          VARCHAR2(1)
 NESTED                                             VARCHAR2(3)
 BUFFER_POOL                                        VARCHAR2(7)
 ROW_MOVEMENT                                       VARCHAR2(8)
 GLOBAL_STATS                                       VARCHAR2(3)
 USER_STATS                                         VARCHAR2(3)
 DURATION                                           VARCHAR2(15)
 SKIP_CORRUPT                                       VARCHAR2(8)
 MONITORING                                         VARCHAR2(3)
 CLUSTER_OWNER                                      VARCHAR2(30)
```

```
SQL> select count(*) "Number of tables"  from dba_tables;

Number of tables
----------------
            1157

SQL>  select count(*) "Number of users" from dba_users;

Number of users
---------------
            182

SQL> select tablespace_name, status from dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
TEMPORARY_DATA                 ONLINE
ROLLBACK_DATA                  ONLINE
USER_DATA                      ONLINE
INDEX_DATA                     ONLINE
TOOLS_DATA                     ONLINE
TSDEV_DEMO                     ONLINE

7 rows selected.

SQL>  select file_name from dba_data_files;

FILE_NAME
--------------------------------------------------------------------------------
/disk/disk1/oracle/oradata/cndb/sys1cndb.ora
/disk/disk1/oracle/oradata/cndb/tmp1cndb.ora
/disk/disk1/oracle/oradata/cndb/rbs1cndb.ora
/disk/disk1/oracle/oradata/cndb/usr1cndb.ora
/disk/disk1/oracle/oradata/cndb/ixd1cndb.ora
/disk/disk1/oracle/oradata/cndb/tool1cndb.ora
/disk/disk1/oracle/OraHome1/dbs/tsdevd.dat

7 rows selected.

SQL> disconnect;
Disconnected from Oracle11g Enterprise Edition Release 8.1.7.4.0 - Production
With the Partitioning option
JServer Release 8.1.7.4.0 - Production

SQL> exit

dimensxcn1:pgirard>
```

Annex 2
Creation of tables, views and index and loading of tables

```
SQL> @crelab2

View dropped.
View dropped.
View dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table dropped.
Table created.
Table created.
View created.
View created.
Table created.
Table created.
Table created.
Table created.
Table created.
Table created.
View created.
Table created.
Table created.
Table created.
Index created.

SQL> host ./loadlab2

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 2
Commit point reached - logical record count 3

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 6

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 4

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 1
Commit point reached - logical record count 2

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 11
```

```
SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 6

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 10
Commit point reached - logical record count 11

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:17 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 2

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:17 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 1
Commit point reached - logical record count 2

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:17 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 1
Commit point reached - logical record count 2

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:17 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.
Commit point reached - logical record count 3
Commit point reached - logical record count 4
```

```
SQL> host more part.log

SQL*Loader: Release 8.1.7.4.0 - Production on Wed Mar 5 13:33:16 2008
 (c) Copyright 2000 Oracle Corporation.  All rights reserved.

Control File:   part.ctl
Data File:      part.dat
  Bad File:     part.bad
  Discard File:  none specified

 (Allow all discards)

Number to load: ALL
Number to skip: 0
Errors allowed: 50
Bind array:     64 rows, maximum of 65536 bytes
Continuation:    none specified
Path used:      Conventional

Table PART, loaded from every logical record.
Insert option in effect for this table: INSERT

   Column Name                  Position   Len  Term Encl Datatype
------------------------------ --------- ----- ---- ---- -------------------
PART_ID                             FIRST     *   ,       CHARACTER
PART_NAME                            NEXT     *   ,       CHARACTER
STOCK_QTY                            NEXT     *   ,       CHARACTER
ORDER_QTY                            NEXT     *   ,       CHARACTER
```

```
MIN_QTY                                NEXT     *   ,       CHARACTER
UNIT                                   NEXT     *   ,       CHARACTER
UNIT_PRICE                             NEXT     *   ,       CHARACTER

Table PART:
  6 Rows successfully loaded.
  0 Rows not loaded due to data errors.
  0 Rows not loaded because all WHEN clauses were failed.
  0 Rows not loaded because all fields were null.

Space allocated for bind array:                  65016 bytes(36 rows)
Space allocated for memory besides bind array:        0 bytes

Total logical records skipped:          0
Total logical records read:             6
Total logical records rejected:         0
Total logical records discarded:        0

Run began on Wed Mar 05 13:33:16 2008
Run ended on Wed Mar 05 13:33:16 2008

Elapsed time was:     00:00:00.08
CPU time was:         00:00:00.02

SQL>
```

Annex 3
Some Unix Commands

- **passwd** modifies your password. You should use it immediately after your first login

- **ls** displays the files in the current directory

- **ls- ls** displays the detailed files in the current directory

- **more *file*** and **cat** *file* displays the content of the file *file*

- **lpr *file*** prints the file *file*

- **pwd** will give you the current path

- **rm *file*** deletes the file le fichier *file* in your current directory

- **mkdir** (**rmdir**) creates (deletes) a directory

- **cd *new_dir*** change the default directory

- **cd ..**   goes back to one directory

- **chmod** *** ***file*** modifies the access rights *(read, write, execute)* to a file or a directory for the owner, the group and other

ex. chmod 700	gives RWE to the owner
chmod 740	gives all rights to the owner, a read right to the group
chmod 755	gives all rights to the owner, and a read/execute to all people including the group

- **logout** (or **exit**) ends your session.

note ⟶ use lowercase characters for these commands