

Practice 6: DML

Jesús Manuel Juárez Pasillas

September 24 2021

1 Introduction

DML statements allow us to perform a wide variety of operations on a database. With this we can make insertions, modifications or deletions depending on how we need it. All these DML declarations contain a great variety of options with which we can solve certain problems faster than just using the declaration as is.

2 Developing

2.1 Activity 1:

Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question.

1. Which of the following commands can be rolled back?

- A. TRUNCATE
- B. DELETE
- C. UPDATE
- D. MERGE
- E. COMMIT
- F. INSERT

As long as a commit is not made right after the chosen statements, the changes will not be made definitively, therefore they can be undone.

2. How can you change the primary key value of a row?

- A. The row must be removed with a DELETE and reentered with an INSERT.
- B. This is only possible if the row is first locked with a SELECT FOR UPDATE.
- C. You cannot change the primary key value.

D. Change it with a simple UPDATE statement.

If the row for which you want to change the value of the primary key is not being used by some other table, it can be changed with an UPDATE.

3. If an UPDATE or DELETE command has a WHERE clause that gives it a scope of several rows, what will happen if there is an error part way through execution?
- A. The command will skip the row that caused the error and continue.
 - B. Whatever work the command had done before hitting the error will be rolled back, but work done already by the transaction will remain.
 - C. The command will stop at the error, and the rows that have been updated or deleted will remain updated or deleted.
 - D. The whole transaction will be rolled back.

If a statement fails in a transaction of several of these, it will only reverse what that statement did and continue doing the others.

4. If a table T1 has four numeric columns, C1, C2, C3, and C4, which of these statements will succeed?
- A. insert into T1 values (1,2,3,null);
 - B. insert into T1 select * from T1;
 - C. insert into T1 values ('1','2','3','4');
 - D. All the statements (A, B, and C) will succeed.
 - E. None of the statements (A, B, or C) will succeed.

All the ways in which the values are inserted are valid so it would not give any error

5. Study the result of this SELECT statement:
SQL> select * from t1;

C1	C2	C3	C4
1	2	3	4
5	6	7	8

If you issue this statement:

insert into t1 (c1,c2) values(select c1,c2 from t1);
why will it fail?

- A. Because the VALUES keyword is not used with a subquery.
- B. Because the subquery is not scalar: it should use MAX or MIN to generate scalar values.

- C. Because the subquery returns multiple rows: it requires a WHERE clause to restrict the number of rows returned to one.
- D. Because values are not provided for all the table's columns: there should be NULLs for C3 and C4.
- E. It will succeed, inserting two rows with NULLs for C3 and C4.

You cannot use a subquery that has more than one record within a values.

6. Consider this statement:
- ```
insert into regions (region_id,region_name)
values ((select max(region_id)+1 from regions), 'Spain');
```
- What will the result be?
- A. The statement will execute without error.
  - B. The statement will fail if the REGIONS table has a third column.
  - C. The statement will not succeed if the value generated for REGION\_ID is not unique, because REGION\_ID is the primary key of the REGIONS table.
  - D. The statement has a syntax error because you cannot use the VALUES keyword with a subquery.

If the regions table only has those two columns, you will not have an error since the subquery returns a single value.

7. You want to insert a row and then update it. What sequence of steps should you follow?
- A. INSERT, COMMIT, SELECT FOR UPDATE, UPDATE, COMMIT
  - B. INSERT, SELECT FOR UPDATE, UPDATE, COMMIT
  - C. INSERT, COMMIT, UPDATE, COMMIT
  - D. INSERT, UPDATE, COMMIT

It is only necessary to make a commit for the two sentences, since if you do one the other will modify the one that was done first, even if the commit has not been done.

8. If you issue this command:
- ```
update employees set salary=salary * 1.3;
```
- what will be the result?
- A. Every row will have SALARY incremented by 30 percent, unless SALARY was NULL.
 - B. There will be an error if any row has its SALARY column NULL.
 - C. The first row in the table will be updated.
 - D. The statement will fail because there is no WHERE clause to restrict the rows affected.

Since there is no where in the statement where nulls are excluded, this would give an error if one exists.

9. How can you delete the values from one column of every row in a table?
- A. Use the UPDATE command.
 - B. Use the DROP COLUMN command.
 - C. Use the DELETE COLUMN command.
 - D. Use the TRUNCATE COLUMN command.

With an Update you can “delete” the data in a column by changing its value to null.

10. Which of these commands will remove every row in a table?
- A. An UPDATE command, setting every column to NULL and with no WHERE clause.
 - B. A TRUNCATE command.
 - C. A DROP TABLE command.
 - D. A DELETE command with no WHERE clause.

The easiest way to delete all the records from a table is with the delete from table_name statement.

11. User JOHN updates some rows and asks user MICHAEL to log in and check the changes before he commits them. Which of the following statements is true?
- A. JOHN must commit the changes so that MICHAEL can see them, but only JOHN can roll them back.
 - B. MICHAEL will not be able to see the changes.
 - C. JOHN must commit the changes so that MICHAEL can see them and, if necessary, roll them back.
 - D. MICHAEL can see the changes but cannot alter them because JOHN will have locked the rows.

As the commit has not yet been made, the new user will not be able to see the changes until a commit is made.

12. User JOHN updates some rows but does not commit the changes. Use MICHAEL queries the rows that JOHN updated. Which of the following statements is true?
- A. MICHAEL will not be able to see the rows because they will be locked.
 - B. MICHAEL will see the state of the state of the data as it was when JOHN last created a SAVEPOINT.

C. MICHAEL will see the old versions of the rows.

D. MICHAEL will be able to see the new values, but only if he logs in as JOHN.

As the changes have not been committed, users other than John will see the old rows.

13. Which of these commands will terminate a transaction?

A. ROLLBACK TO SAVEPOINT

B. TRUNCATE

C. ROLLBACK

D. COMMIT

E. SAVEPOINT

F. DELETE

These commands terminate transactions by removing or committing the changes that have been made.

2.2 Activity 2:

Insert a dataset for the scheme SALES carried out in practice 4, activity 2. Take into account the following instructions:

- a) Use sequences as possible for primary key values (using pseudo-columns CURRVAL and NEXTVAL).
- b) Insert a representative set of values for each table.
- c) Use SYSDATE as possible.

Sentences:

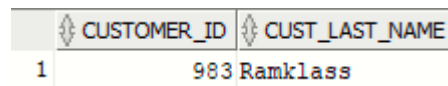
- insert into Channels values(channels_id.nextval, 'channel ' || channels_id.currval);
- insert into Channels values(channels_id.nextval, 'channel ' || channels_id.currval);
- insert into Channels values(channels_id.nextval, 'channel ' || channels_id.currval);
- insert into Channels values(channels_id.nextval, 'channel ' || channels_id.currval);
- insert into Channels values(channels_id.nextval, 'channel ' || channels_id.currval);
- insert into Employees values(emp_id.nextval, 'Juan', 'Fernandez', null, 'address ' || emp_id.currval, sysdate, 'Man', 'Fernando');
- insert into Employees values(emp_id.nextval, 'Lucia', 'martinez', 1, 'address ' || emp_id.currval, sysdate, 'Man', 'Fernando');

- insert into Employees values(emp_id.nextval, 'Pedro', 'Beltran', 1, 'address' || emp_id.currval, sysdate, 'Man', 'Fernando');
- insert into Employees values(emp_id.nextval, 'Maria', 'Juarez', 1, 'address' || emp_id.currval, sysdate, 'Man', 'Fernando');
- insert into Employees values(emp_id.nextval, 'Vanessa', 'Flores', 1, 'address' || emp_id.currval, sysdate, 'Man', 'Fernando');
- insert into Products values(pro_id.nextval, 'Product ' || pro_id.currval, 54.21, 50.50, 'Provider ' || pro_id.currval);
- insert into Products values(pro_id.nextval, 'Product ' || pro_id.currval, 25.25, 22.50, 'Provider ' || pro_id.currval);
- insert into Products values(pro_id.nextval, 'Product ' || pro_id.currval, 12.21, 10.50, 'Provider ' || pro_id.currval);
- insert into Products values(pro_id.nextval, 'Product ' || pro_id.currval, 12.21, 12.00, 'Provider ' || pro_id.currval);
- insert into Products values(pro_id.nextval, 'Product ' || pro_id.currval, 99.99, 90.50, 'Provider ' || pro_id.currval);
- insert into Shops values(sho_id.nextval, 'Shop address ' || sho_id.currval, 1);
- insert into Shops values(sho_id.nextval, 'Shop address ' || sho_id.currval, 2);
- insert into Shops values(sho_id.nextval, 'Shop address ' || sho_id.currval, 3);
- insert into Shops values(sho_id.nextval, 'Shop address ' || sho_id.currval, 4);
- insert into Shops values(sho_id.nextval, 'Shop address ' || sho_id.currval, 5);
- insert into Sales values(sal_id.nextval, 1, 1, 1, 300, 1, sysdate);
- insert into Sales values(sal_id.nextval, 2, 2, 2, 109, 2, sysdate);
- insert into Sales values(sal_id.nextval, 3, 3, 3, 123, 3, sysdate);
- insert into Sales values(sal_id.nextval, 4, 4, 4, 421, 4, sysdate);
- insert into Sales values(sal_id.nextval, 5, 5, 5, 542, 5, sysdate);
- commit;

2.3 Activity 3:

Carry out this exercise in the oe schema. Analyze the results for each sentence.

1. Insert a customer into CUSTOMERS, using a function to generate a unique customer number.
2. Give him a credit limit equal to the average credit limit.
3. Create another customer using the customer just created, but make sure the CUSTOMER_ID is unique.
4. Change the name of the second entered customer.
5. Commit this transaction.
6. Determine the CUSTOMER_IDs of the two new customers and lock the rows.



	CUSTOMER_ID	CUST_LAST_NAME
1	983	Ramklass

Figure 1: Lock the rows.

7. From another session connected to the OE schema, attempt to select the locked rows.
What happened? If the data could be consulted even if they were blocked
8. In this second session connected to the OE schema, attempt to update one of the locked rows. The operation remains until the rows are unlocked.

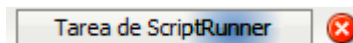
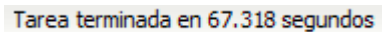


Figure 2: Update one of the locked rows.

9. This command will hang. In the first session, release the locks by issuing a commit. After making the commit the rows are unlocked and the update



A screenshot of a status message that reads 'Tarea terminada en 67.318 segundos'.

Figure 3: Commit.

is done

10. The second session will now complete its update. In the second session, delete the two rows.

```

Error que empieza en la línea: 33 del comando :
truncate table customers
Informe de error -
ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired
00054. 00000 - "resource busy and acquire with NOWAIT specified or timeout expired"
*Cause:      Interested resource is busy.
*Action:     Retry if necessary or increase timeout.

```

Figure 4: Truncate the CUSTOMERS table.

11. In the first session, attempt to truncate the CUSTOMERS table.
12. This will fail because there is a transaction in progress against the table, which will block all DDL commands. In the second session, commit the transaction.
13. The CUSTOMERS table will now be back in the state it was in at the start of the exercise. Confirm this by checking the value of the highest CUSTOMER_ID.

	MAX(CUSTOMER_ID)
1	981

Figure 5: Select the CUSTOMERS table.

2.4 Activity 4:

In this section, use various techniques to insert rows into a table. Follow the next instructions.

1. Connect to the HR schema.
2. Query the REGIONS table, to check what values are already in use for the REGION_ID column.

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa

Figure 6: Check values.

3. Insert a row into the REGIONS table, providing the values in line.
4. Insert a row into the REGIONS table, providing the values as substitution variables.

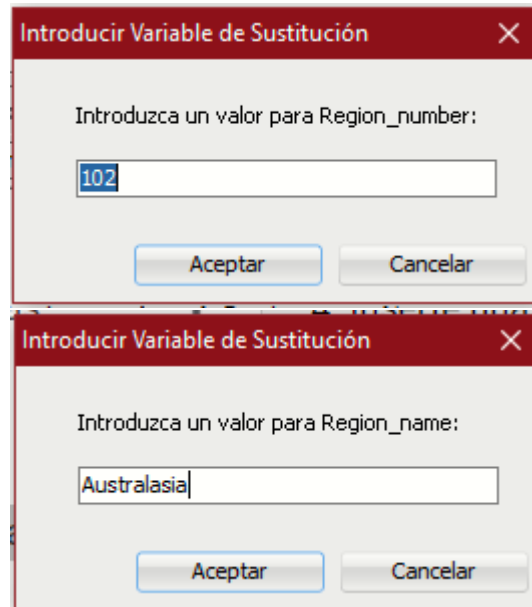


Figure 7: Providing the values as substitution variables.

5. Insert a row into the REGIONS table, calculating the REGION_ID to be one higher than the current high value. This will need a scalar subquery.
6. Confirm the insertion of the rows.

	REGION_ID	REGION_NAME
1	101	Great Britain
2	102	Australasia
3	103	Oceania
4	1	Europe
5	2	Americas
6	3	Asia
7	4	Middle East and Africa

Figure 8: Confirm the insertion of the rows.

7. Commit the insertions.

2.5 Activity 5:

In this section, use various techniques to update rows in a table. Follow the next instructions.

1. Connect to the HR schema.
2. Update a single row, identified by primary key.

```
1 fila actualizadas.
```

Figure 9: Update a single row.

3. Update a set of rows, using a nonequality predicate.

```
3 filas actualizadas.
```

Figure 10: Update a set of rows.

4. Update a set of rows, using subqueries to select the rows and to provide values.

```
3 filas actualizadas.
```

Figure 11: Update a set of rows (subqueries).

5. Confirm the state of the rows.

REGION_ID	REGION_NAME
1	204 Iberia
2	205 Iberia
3	206 Iberia
4	1 Europe
5	2 Americas
6	3 Asia
7	4 Middle East and Africa

Figure 12: Confirm the state of the rows.

2.6 Activity 6:

In this section, use various techniques to delete rows in a table. Follow the next instructions.

1. Connect to the HR schema using SQL Developer.
2. Remove one row, using the equality predicate on the primary key.

1 fila eliminado

Figure 13: Remove one row.

3. Attempt to remove every row in the table by omitting a WHERE clause.

```
Error que empieza en la línea: 33 del comando :
delete from regions
Informe de error -
ORA-02292: integrity constraint (HR.COUNTR_REG_FK) violated - child record found
```

Figure 14: Attempt to remove every row.

4. Remove rows with the row selection based on a subquery.

2 filas eliminado

Figure 15: Remove rows with the row selection based on a subquery.

5. Confirm that the REGIONS table now contains just the original four rows.

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Figure 16: Confirm that the REGIONS table.

6. Commit the deletions.

2.7 Activity 7:

The MERGE command is often ignored, because it does nothing that cannot be done with INSERT, UPDATE, and DELETE. It is, however, very powerful, in that with one pass through the data it can carry out all three operations. This can improve performance dramatically. Here is a simple example, do not forget to include and analyze the results.

1. Create a new table employees2.
2. Describe the new table.

Nombre	¿Nulo?	Tipo
EMPLOYEE_ID		NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

Figure 17: Describe the new table.

3. Insert a new row into table employees2.
4. Update the salary of the inserted row.
5. Execute the merge operation.

```
107 filas fusionadas.
```

Figure 18: Execute the merge operation.

6. View the final data in employees2.

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	(null)	21/09/05	AD_VP	17000	(null)	(null)	(null)
3	102	Lex	De Haan	LDEHAAN	(null)	13/01/01	AD_VP	17000	(null)	(null)	(null)
4	103	Alexander	Hunold	AHUNOLD	(null)	03/01/06	IT_PROG	9000	(null)	(null)	(null)
5	104	Bruce	Ernst	BERNST	(null)	21/05/07	IT_PROG	6000	(null)	(null)	(null)
6	105	David	Austin	DAUSTIN	(null)	25/06/05	IT_PROG	4800	(null)	(null)	(null)
7	106	Valli	Pataballa	VPATABAL	(null)	05/02/06	IT_PROG	4800	(null)	(null)	(null)
8	107	Diana	Lorentz	DLORENTZ	(null)	07/02/07	IT_PROG	4200	(null)	(null)	(null)
9	108	Nancy	Greenberg	NGREENBE	(null)	17/08/02	FI_MGR	12008	(null)	(null)	(null)
10	109	Daniel	Faviet	DFAVIET	(null)	16/08/02	FI_ACCOUNT	9000	(null)	(null)	(null)
11	110	John	Chen	JCHEN	(null)	28/09/05	FI_ACCOUNT	8200	(null)	(null)	(null)
12	111	Ismael	Sciarra	ISCIARRA	(null)	30/09/05	FI_ACCOUNT	7700	(null)	(null)	(null)
13	112	Jose Manuel	Urman	JMURMAN	(null)	07/03/06	FI_ACCOUNT	7800	(null)	(null)	(null)
14	113	Luis	Popp	LPOPP	(null)	07/12/07	FI_ACCOUNT	6900	(null)	(null)	(null)
15	114	Den	Raphaely	DRAPHEAL	(null)	07/12/02	FU_MAN	11000	(null)	(null)	(null)
16	115	Alexander	Khoo	AKHOO	(null)	18/05/03	FU_CLERK	3100	(null)	(null)	(null)
17	116	Shelli	Baida	SBAIDA	(null)	24/12/05	FU_CLERK	2900	(null)	(null)	(null)
18	117	Sigal	Tobias	STOBIAS	(null)	24/07/05	FU_CLERK	2800	(null)	(null)	(null)
19	118	Guy	Himuro	GHIMURO	(null)	15/11/06	FU_CLERK	2600	(null)	(null)	(null)
20	119	Karen	Colmenares	KCOLMENA	(null)	10/08/07	FU_CLERK	2500	(null)	(null)	(null)

Figure 19: View the final data in employees2.

2.8 Activity 8:

Propose a solution to the following scenarios:

- a) Transactions, like constraints, are business rules: a technique whereby the database can enforce rules developed by business analysts. If the “logical unit of work” is huge, such as an accounting suite period rollover, should this actually be implemented as one transaction?
 - a) No, it is better to separate a huge transaction into smaller ones and then put the transactions together into a transaction that will not be so huge and will only have the important information.
- b) Being able to do DML operations, look at the result, then roll back and try them again can be very useful. But is it really a good idea?
 - a) Depending on the database in which the operations are being carried out, since if it is done in a database in operation and an error is made when making the transaction, such as confirming the changes, it will be very difficult to go back, instead if only done in a personal scheme is very good to continue learning.

2.9 Activity 9:

Only the output images will be included for each transaction. In this exercise, demonstrate the use of transaction control statements and transaction isolation. Connect to the HR schema with two sessions concurrently. These can be two SQL Developer sessions.

The following table lists steps to follow in each session.

1. Statements 1:

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa

Figure 20: Statements 1 (Session 1 and 2).

2. Statements 3:

	REGION_ID	REGION_NAME
1	100	UK
2	1	Europe
3	2	Americas
4	3	Asia
5	4	Middle East and Africa

Figure 21: Statements 3 (Session 1).

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa
5	101	UK

Figure 22: Statements 3 (Session 2).

3. Statements 5:

	REGION_ID	REGION_NAME
1	100	UK
2	1	Europe
3	2	Americas
4	3	Asia
5	4	Middle East and Africa

Figure 23: Statements 5 (Session 1).

	REGION_ID	REGION_NAME
1	100	UK
2	1	Europe
3	2	Americas
4	3	Asia
5	4	Middle East and Africa
6	101	UK

Figure 24: Statements 5 (Session 2).

4. Statements 7:

	REGION_ID	REGION_NAME
1	100	UK
2	1	Europe
3	2	Americas
4	3	Asia
5	4	Middle East and Africa

Figure 25: Statements 7 (Session 1 and 2).

5. Statements 8:

```
delete from regions where region_id = 101
Error en la línea de comandos : 7 Columna : 13
Informe de error -
Error SQL: ORA-00942: table or view does not exist
00942. 00000 - "table or view does not exist"
*Cause:      |
*Action:
```

Figure 26: Statements 8 (Session 2).

6. Statements 9:

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa

Figure 27: Statements 9 (Session 1).

	REGION_ID	REGION_NAME
1	100	UK
2	1	Europe
3	2	Americas
4	3	Asia
5	4	Middle East and Africa

Figure 28: Statements 9 (Session 2).

7. Statements 11:

REGION_ID	REGION_NAME
1	Europe
2	Americas
3	Asia
4	Middle East and Africa

Figure 29: Statements 11 (Session 1 and 2).

2.10 Activity 10:

The HR department wants you to create SQL statements to insert, update, and delete employee data. As a prototype, you use the MY_EMPLOYEE table before giving the statements to the HR department.

"Plugins" folder is located inside the "Plugins.zip" file.

1. Create the DDL sentence to build the MY_EMPLOYEE table used in this activity (see item 2). Save this sentences in lab_06.01.sql.
The file (lab_06.01.sql) is located inside the attached folder called "Plugins".
2. Describe the structure of the MY_EMPLOYEE table to identify the column names.

Nombre	Nulo?	Tipo
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

Figure 30: Describe the structure of the MY_EMPLOYEE.

3. Create an INSERT statement to add the first row of data to the MY_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause. Do not enter all rows yet.
 - INSERT INTO MY_EMPLOYEE VALUES(1,'Patel','Ralph','rpatel',895);
4. Populate the MY_EMPLOYEE table with the second row of the sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

- INSERT INTO MY_EMPLOYEE(ID, LAST_NAME, FIRST_NAME, USERID, SALARY) VALUES(2, 'Dancs', 'Betty', 'bdancs', 860);

5. Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

Figure 31: Confirm your addition to the table.

6. Write an INSERT statement in a dynamic reusable script file to load the remaining rows into the MY_EMPLOYEE table. The script should prompt for all the columns (ID, LAST_NAME, FIRST_NAME, USERID, and SALARY). Save this script to a lab_06_06.sql file.

The file (lab_06_06.sql) is located inside the attached folder called "Plugins".

7. Populate the table with the next two rows of the sample data listed in step 3 by running the INSERT statement in the script that you created.

```
Antiguo:INSERT INTO MY_EMPLOYEE VALUES(&ID,&LAST_NAME,&FIRST_NAME,&USERID,&SALARY)
Nuevo:INSERT INTO MY_EMPLOYEE VALUES(3,'Biri','Ben','bbiri',1100)

1 fila insertadas.

Antiguo:INSERT INTO MY_EMPLOYEE VALUES(&ID,&LAST_NAME,&FIRST_NAME,&USERID,&SALARY)
Nuevo:INSERT INTO MY_EMPLOYEE VALUES(4,'Newman','Chad','cnewman',750)

1 fila insertadas.
```

Figure 32: INSERT statement with script.

8. Confirm your additions to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

Figure 33: Select 8.

9. Make the data additions permanent.

- COMMIT;
- Change the last name of employee 3 to Drexler.
 - UPDATE MY_EMPLOYEE SET LAST_NAME = 'Drexler' WHERE ID = 3;
 - Change the salary to \$1,000 for all employees who have a salary less than \$900.
 - UPDATE MY_EMPLOYEE SET SALARY = 1000 WHERE SALARY < 900;
 - Verify your changes to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Figure 34: Select 12.

- Delete Betty Dancs from the MY_EMPLOYEE table.
 - DELETE MY_EMPLOYEE WHERE LAST_NAME = 'Dancs' AND FIRST_NAME = 'Betty';
- Confirm your changes to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Figure 35: Select 14.

- Commit all pending changes.
 - COMMIT;

16. Populate the table with the last row of the sample data listed in step 3 by using the statements in the script that you created in step 6. Run the statements in the script.

```
Antiguo:INSERT INTO MY_EMPLOYEE VALUES(&ID,'&LAST_NAME','&FIRST_NAME','&USERID',&SALARY)
Nuevo:INSERT INTO MY_EMPLOYEE VALUES(5,'Ropeburn','Audrey','aropebur',1550)

1 fila insertadas.
```

Figure 36: Run the statements in the script.

17. Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

Figure 37: Select 17.

18. Mark an intermediate point in the processing of the transaction.

- SAVEPOINT S1;

19. Delete all the rows from the MY_EMPLOYEE table.

- DELETE MY_EMPLOYEE;

20. Confirm that the table is empty.

ID	LAST_NAME	FIRST_NA...	USERID	SALARY
----	-----------	-------------	--------	--------

Figure 38: Select 20.

21. Discard the most recent DELETE operation without discarding the earlier INSERT operation.

- ROLLBACK TO SAVEPOINT S1;

22. Confirm that the new row is still intact.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
5	Ropeburn	Audrey	aropebur	1550
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Figure 39: Select 22.

23. Make the data addition permanent.

- COMMIT;

24. Modify the lab_06_06.sql script such that the USERID is generated automatically by concatenating the first letter of the first name and the first seven characters of the last name. The generated USERID must be in lowercase. Hence, the script should not prompt for the USERID. Save this script to a file named lab_06_24.sql.

The file (lab_06_24.sql) is located inside the attached folder called "Plugins".

25. Run the script, lab_06_24.sql to insert the following record.

```
Nuevo:DECLARE
  L_NAME VARCHAR2(25);
  F_NAME VARCHAR2(25);
BEGIN
  L_NAME := 'Anthony';
  F_NAME := 'Mark';
  INSERT INTO MY_EMPLOYEE VALUES(6,L_NAME,F_NAME,
    LOWER(SUBSTR(F_NAME,1,1)||SUBSTR(L_NAME,1,7)),1230);
END;
Procedimiento PL/SQL terminado correctamente.
```

Figure 40: Run the script.

26. Confirm that the new row was added with correct USERID.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
5	Ropeburn	Audrey	aropebur	1550
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
6	Anthony	Mark	manthony	1230

Figure 41: Select 26.

3 Pre-Assessment:

- Practices pre-Assessment for Database Systems Laboratory II

Practice	Pre-Assessment
COMPLIES WITH THE REQUESTED FUNCTIONALITY	X
HAS THE CORRECT INDENTATION	X
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	X
HAS A REPORT WITH IDC FORMAT	X
REPORT INFORMATION IS FREE OF SPELLING ERRORS	X
DELIVERED IN TIME AND FORM	
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	100

Table 1: Pre-Assessment.

4 Conclusion:

DML statements allow us to develop a wide variety of options with which we can manipulate the data in a database. These allow us to make the sentences that are used more reusable, they also allow us to manipulate the way in which one or more elements are inserted, modified or eliminated depending on what we want to achieve with said sentence.