Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Software Engineering Academic Program

Group: 5B - Semester: 2022-5º

Practice Number: 08

Practice Name: Restricting and Sorting Data

Date: 06/October/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Contents

Restricting and Sorting Data

September 06, 2022

1 Introduction

The DML (Data Modification Language) is one of the fundamental parts of the SQL language. It is formed by the instructions capable of modifying (add, change or delete) the data of the tables.

The set of DML statements that are executed consecutively is called a transaction. The interesting thing about transactions is that we can cancel them, since they form a logical unit of work that until they are accepted, their results will not be final.

In all DML statements, the only data returned by the system is the number of rows that have been modified by executing the statement.

The elements used to manipulate the data are the following:

-SELECT, this statement is used to query the data.
-INSERT, with this instruction we can insert the values in a database.
-UPDATE, used to modify the values of one or more records.
-DELETE is used to remove rows from a table.

Data queries with SQL (DQL):

DQL is short for SQL Data Query Language. The only command that belongs to this language is the versatile SELECT command. This command fundamentally allows:

1- Get data from certain columns of a table (projection).

2- Get records (rows) from a table according to certain criteria (selection).

3- Mix data from different tables (association, join).

4- Perform calculations on the data group data.

So far we have seen how the SELECT statement can be used to retrieve all or a subset of the columns from a table. But this effect affects all rows in the table, unless we specify something else in the **WHERE** clause. This is where we must propose the condition that all the rows must meet to appear in the result of the query. The complexity of the search criteria is practically unlimited, and it is possible to combine various types of operators with column functions, composing more or less complex expressions.

ORDER BY clause: Used to specify the sort order of the response to the query. By default the order is ascending, although a descending order can be specified. Sorting can be set on the content of columns or on expressions with columns.

2 Practice objective

Use SQL SELECT statements for retrieving data from database by means of different contexts where data are filtered using the WHERE clause and ordered using the ORDER clause.

3 Developing

Activity 1: Read all the choices carefully because there might be more than one correct answer. Choose all the correct answers for each question.

Explain the reason for your answer.

LIMIT THE ROWS RETRIEVED BY A QUERY.

1. Which two clauses of the SELECT statement facilitate selection and projection?

R = C) SELECT, WHERE;

Explanation: The SELECT clause makes projection easy by specifying the list of columns to pass from the table, and the WHERE clause makes selection easy by limiting the number of rows to retrieve based on the condition.

2. Choose the query that extracts the LAST_NAME, JOB_ID, and SALARY values from the EMPLOYEES table for records having JOB_ID values of either SA_REP or MK_MAN and having SALARY values in the range of $1000 to $4000. The SELECT and FROM clauses are SELECT LAST_NAME, JOB_ID, SALARY FROM EMPLOYEES:

R = B) WHERE JOB_ID IN ('SA_REP','MK_MAN') AND SALARY BETWEEN 1000 AND 4000;

Explanation: The IN operator effectively checks whether the JOB_ID for a particular row is SA_REP or MK_MAN, while the BETWEEN operator effectively measures whether the SALARY value is within the requested range of $1000 to $4000.

3. Which of the following WHERE clauses contains an error? The SELECT and FROM clauses are SELECT * FROM EMPLOYEES:

R = C) WHERE JOB_ID IN (SA_REP,MK_MAN);

Explanation: The literal characters that the IN operator compares to the JOB_ID column must be enclosed in single quotes. Note that option B does not require quotes around numeric characters, but their presence does not cause the statement to fail.

4. Choose the WHERE clause that extracts the DEPARTMENT_NAME values containing the character literal "er" from the DEPARTMENTS table. The SELECT and FROM clauses are SELECT DEPARTMENT_NAME FROM DEPARTMENTS:

R = B) WHERE DEPARTMENT_NAME LIKE '%er%';

Explanation: The LIKE operator tests the DEPARTMENT_NAME column of each row for values that contain the characters "er". The percentage symbols before and after the character literal indicate that any characters enclosing the "er" literal are permissible.

5.Which two of the following conditions are equivalent to each other?

R = A) WHERE COMMISSION_PCT IS NULL.
D) WHERE NOT(COMMISSION_PCT IS NOT NULL)

Explanation: The IS NULL operator correctly evaluates the COMMISSION_PCT column for NULL values, la opcion D uses the NOT operator to negate the already negative version of the IS NULL operator, IS NOT NULL. Two negatives return a positive, and therefore A and D are equivalent.

6. Which three of the following conditions are equivalent to each other?

R = A, C AND D.

Explanation: Each of these conditions tests for SALARY values in the range of $2000 to $5000.

SORT THE ROWS RETRIEVED BY A QUERY.

7. Choose one false statement about the ORDER BY clause.

R = C) The ORDER BY clause specifies one or more terms by which the retrieved rows are sorted. These terms can only be column names.

Explanation: The terms specified in an ORDER BY clause can include column names, positional sorting, numeric values, and expressions.

8. The following query retrieves the LAST_NAME, SALARY, and COMMISSION_PCT values for employees whose LAST_NAME begins with the letter R. Based on the following query, choose the ORDER BY clause that first sorts the results by the COMMISSION_PCT column, listing highest commission earners first, and then sorts the results in ascending order by the SALARY column. Any records with NULL COMMISSION_PCT must appear last:
SELECT LAST_NAME, SALARY, COMMISSION_PCT FROM EMPLOYEES WHERE LAST_NAME LIKE 'R%';

R = C. ORDER BY 3 DESC NULLS LAST, 2 ASC;

Explanation: Positional sorting is performed, and the third term in the SELECT list, COMMISSION_PCT, is sorted first in descending order, and any NULL COMMISSION_PCT values are listed last. The second term in the SELECT list, SALARY, is sorted next in ascending order.

AMPERSAND SUBSTITUTION.

9. The DEFINE command explicitly declares a session-persistent substitution variable with a specific value. How is this variable referenced in an SQL statement? Consider an expression that calculates tax on an employee's SALARY based on the current tax rate. For the following session-persistent substitution variable, which statement correctly references the TAX_RATE variable?
DEFINE TAX_RATE = 0.14.

R = B). SELECT SALARY * &TAX_RATE TAX FROM EMPLOYEES;

Explanation: A session-persistent substitution variable may be referenced using an ampersand symbol from within any SQL statement executed in that session.

10. When using ampersand substitution variables in the following query, how many times will you be prompted to input a value for the variable called JOB the first time this query is executed?
SELECT FIRST_NAME, '&JOB' FROM EMPLOYEES WHERE JOB_ID LIKE '%'||JOB||'%' AND '&&JOB' BETWEEN 'A' AND 'Z';

R = D) 3.

Explanation: TThe first time this statement is executed, two single ampersand substitution variables are encountered before the third double ampersand substitution variable. If the first reference on line one of the query contained a double ampersand substitution, you would only be prompted to input a value once.

Activity 2: Propose an answer to the following issues:

a) The SELECT list of a query contains a single column. Is it possible to sort the results retrieved by this query by another column?

R = Yes, although the query result will only show the data from one column, the data will not be ordered by the column specified in the ORDER BY clause.

b) Ampersand substitution variables support reusability of repetitively executed SQL statements. If a substituted value is to be used multiple times at different parts of the same statement, is it possible to be prompted to submit a substitution value just once and for that value to automatically be substituted during subsequent references to the same variable?

R = This is TRUE since this will avoid having to request the same value many times to be used in the statement.

c) You have been tasked to retrieve the LAST_NAME and DEPARTMENT_ID values for all rows in the EMPLOYEES table. The output must be sorted by the nullable DEPARTMENT_ID column, and all rows with NULL DEPARTMENT_ID values must be listed last. Is it possible to provide the results as requested?

R = If this is possible, the sentence that should be used is the following:
SELECT LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT_ID IS NOT NULL ORDER BY DEPARTMENT_ID;

d) You have a complex query with multiple conditions. Is there a restriction on the number of conditions you can specify in the WHERE clause? Is there a limit to the number of comparison operators you can use in a single query?

R = There is no limit to the where clause.

e) You have been tasked to locate rows in the EMPLOYEES table where the SALARY values contain the numbers 8 and 0 adjacent to each other. The SALARY column has a NUMBER data type. Is it possible to use the LIKE comparison operator with numeric data?

R = If it is possible to use LIKE since this operator does not have any problem between searching for numbers or strings, the sentence of this section would be as follows:
SELECT * FROM EMPLOYEES WHERE SALARY LIKE '%80%';

f) By restricting the rows returned from the JOBS table to those which contain the value SA_REP in the JOB_ID column, is a projection, selection or join performed?

R = It is a selection since the data to be consulted is restricted by a condition.

Activity 3: Connect to the OE schema and complete the following tasks.

A customer requires a hard disk drive and a graphics card for her personal computer. She is willing to spend between $500 and $800 on the disk drive but is unsure about the cost of a graphics card. Her only requirement is that the resolution supported by the graphics card should be either 1024×768 or 1280×1024. As the sales representative, you have been tasked to write one query that searches the PRODUCT_INFORMATION table where the PRODUCT_NAME value begins with HD (hard disk) or GP (graphics processor) and their list prices. Remember the hard disk list prices must be between $500 and $800 and the graphics processors need to support either 1024×768 or 1280×1024. Sort the results in descending LIST_PRICE order.

Note: Capture an image for each statement output.

The following **figure number 1** shows the statement made for activity three.

```
-- PRACTICE 08, ACTIVITY 03.
SELECT PRODUCT_NAME, LIST_PRICE, PRODUCT_DESCRIPTION FROM PRODUCT_INFORMATION
WHERE (PRODUCT_NAME LIKE '%HD%' OR PRODUCT_NAME LIKE '%GP%') AND
((LIST_PRICE BETWEEN 500 AND 800) AND PRODUCT_NAME LIKE '%HD%')
OR (PRODUCT_NAME LIKE '%GP 1024x768%' or PRODUCT_NAME LIKE '%GP 1280x1024%')
ORDER BY LIST_PRICE DESC;
```

Resultado de la Co... SQL | Todas las Filas Recuperadas: 8 en 0.271 segundos

	PRODUCT_NAME	LIST_PRICE	PRODUCT_DESCRIPTION
1	HD 18.2GB @10000 /E	800	External hard drive disk - 18.2 GB, rated up to 10,000 RP
2	HD 12GB @7200 /SE	775	12GB capacity hard disk drive (external) SCSI, 7200 RPM.
3	HD 12GB /S	633	12GB hard disk drive for Standard Mount. Supra9 hot plugg
4	HD 12GB /R	612	12GB Removable hard disk drive. With compatibility across
5	HD 12GB /N	567	12GB hard disk drive for Narrow Mount. Supra9 hot pluggab
6	HD 12GB /I	543	12GB capacity harddisk drive (internal). Supra drives eli
7	GP 1280x1024	98	Graphics Processor, resolution 1280 X 1024 pixels. High e
8	GP 1024x768	78	Graphics Processor, resolution 1024 X 768 pixels. Outstan

Figure 1: *SELECT statement, data retrieval.*

Activity 4: This exercise must be performed using HR schema.

• Retrieve a list of DEPARTMENT_NAME values that end with the three letters "ing" from the DEPARTMENTS table, see **figure 2**:

```
-- PRACTICE 08, ACTIVITY 04:
SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_NAME LIKE '%ing';
```

Resultado de la Consulta SQL | Todas las Filas Recuperadas: 7 en 0.002 segundos

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	20	Marketing	201	1800
2	30	Purchasing	114	1700
3	50	Shipping	121	1500
4	110	Accounting	205	1700
5	170	Manufacturing	(null)	1700
6	190	Contracting	(null)	1700
7	260	Recruiting	(null)	1700

Figure 2: *SELECT statement.*

• The JOBS table contains descriptions of different types of jobs an employee in the organization may occupy. It contains the JOB_ID, JOB_TITLE, MIN_SALARY, and MAX_SALARY columns. You are required to write a query that extracts the JOB_TITLE, MIN_SALARY, and MAX_SALARY columns, as well as an expression called VARIANCE, which is the difference between the MAX_SALARY and MIN_SALARY values, for each row. The results must include only JOB_TITLE values that contain either the word "President" or "Manager." Sort the list in descending order based on the VARIANCE expression. If more than one row has the same VARIANCE value, then, in addition, sort these rows by JOB_TITLE in reverse alphabetic order, see **figure 3**:

```
SELECT JOB_TITLE, MAX_SALARY, MIN_SALARY,
       MAX_SALARY - MIN_SALARY "VARIANCE"
FROM JOBS WHERE JOB_TITLE LIKE '%President%' OR JOB_TITLE LIKE '%Manager%'
ORDER BY MAX_SALARY - MIN_SALARY DESC, JOB_TITLE DESC;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 8 en 0.004 segundos

	JOB_TITLE	MAX_SALARY	MIN_SALARY	VARIANCE
1	President	40000	20080	19920
2	Administration Vice President	30000	15000	15000
3	Sales Manager	20080	10000	10080
4	Finance Manager	16000	8200	7800
5	Accounting Manager	16000	8200	7800
6	Purchasing Manager	15000	8000	7000
7	Marketing Manager	15000	9000	6000
8	Stock Manager	8500	5500	3000

Figure 3: *SELECT statement.*

• A common calculation performed by the Human Resources department relates to the calculation of taxes levied upon an employee. Although, this is done for all employees, there are always a few staff members who dispute the tax deducted from their income. The tax deducted per employee is calculated by obtaining the annual salary for the employee and multiplying this by the current tax rate, which may vary from year to year. You are required to write a reusable query using the current tax rate and the EMPLOYEE_ID number as inputs and return the EMPLOYEE_ID, FIRST_NAME, SALARY, ANNUAL SALARY (SALARY * 12), TAX_RATE, and TAX (TAX_RATE * ANNUAL SALARY) information, see **figure 4, 5 and 6**:

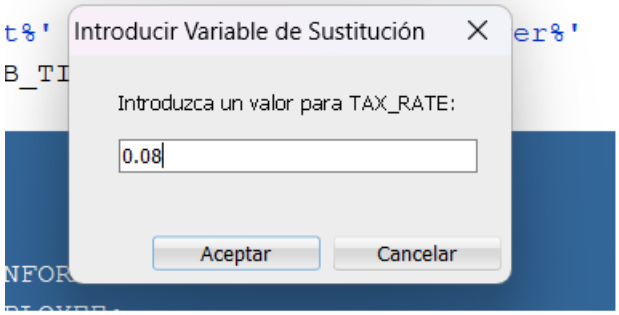


Figure 4: *Request for a substitution variable.*

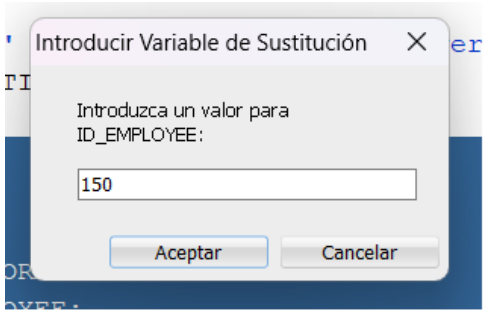
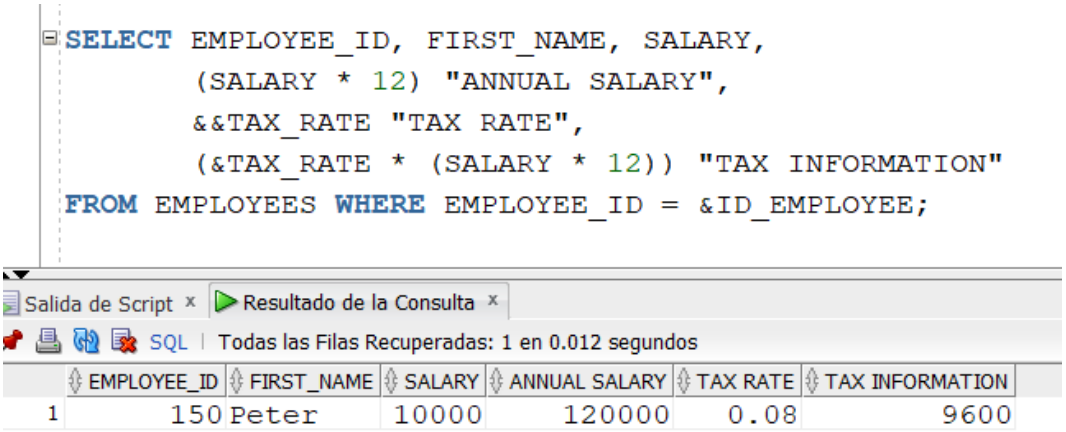


Figure 5: *Request for a substitution variable.*



```
SELECT EMPLOYEE_ID, FIRST_NAME, SALARY,
       (SALARY * 12) "ANNUAL SALARY",
       &&TAX_RATE "TAX RATE",
       (&TAX_RATE * (SALARY * 12)) "TAX INFORMATION"
FROM EMPLOYEES WHERE EMPLOYEE_ID = &ID_EMPLOYEE;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0.012 segundos

	EMPLOYEE_ID	FIRST_NAME	SALARY	ANNUAL SALARY	TAX RATE	TAX INFORMATION
1	150	Peter	10000	120000	0.08	9600

Figure 6: *SELECT statement.*

Activity 5: In this practice, you build more reports, including statements that use the WHERE clause and the ORDER BY clause. You make the SQL statements more reusable and generic by including the ampersand substitution.

The HR department needs your assistance in creating some queries.

1. Because of budget issues, the HR department needs a report that displays the last name and salary of employees who earn more than $12,000. Save your SQL statement as a file named lab_8_01.sql. Run your query, see **figure 7**:

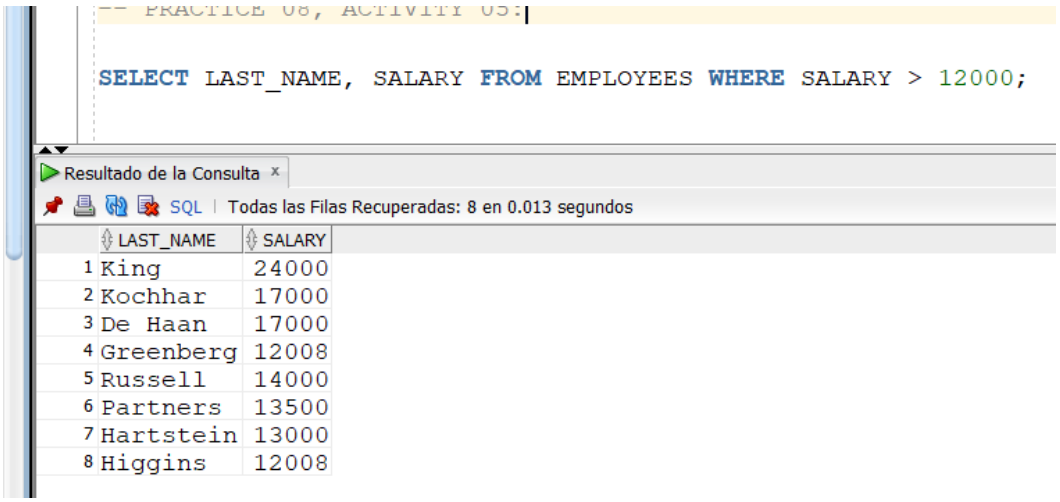


Figure 7: *SELECT statement.*

2. Open a new SQL Worksheet. Create a report that displays the last name and department number for employee number 176. Run the query, see **figure 8**:

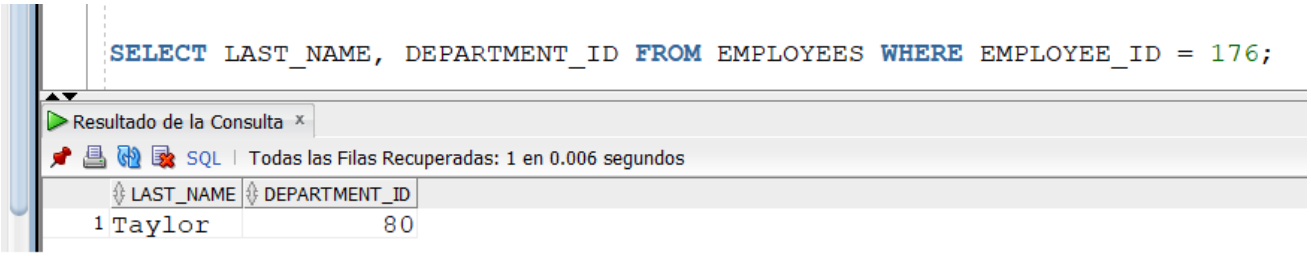


Figure 8: *SELECT statement.*

3. The HR department needs to find high-salary and low-salary employees. Modify lab_8_01.sql to display the last name and salary for any employee whose salary is not in the range of $5,000 to $12,000. Save your SQL statement as lab_8_03.sql, see **figure 9**:

```
SELECT LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY NOT BETWEEN 5000 AND 12000;
```

Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0.008 segundos

	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Austin	4800
5	Pataballa	4800
6	Lorentz	4200
7	Greenberg	12008
8	Khoo	3100
9	Baida	2900
10	Tobias	2800
11	Himuro	2600
12	Colmenares	2500
13	Nayer	3200
14	Mikkilineni	2700

Figure 9: *SELECT statement.*

4. Create a report to display the last name, job ID, and hire date for employees with the last names of Matos and Taylor. Order the query in ascending order by the hire date, see **figure 10**:

```
SELECT LAST_NAME, JOB_ID, HIRE_DATE FROM EMPLOYEES
WHERE LAST_NAME IN ('Matos', 'Taylor') ORDER BY HIRE_DATE ASC;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 3 en 0.004 segundos

	LAST_NAME	JOB_ID	HIRE_DATE
1	Taylor	SH_CLERK	24/01/06
2	Matos	ST_CLERK	15/03/06
3	Taylor	SA_REP	24/03/06

Figure 10: *SELECT statement.*

5. Display the last name and department ID of all employees in departments 20 or 50 in ascending alphabetical order by name, see **figure 11**:

```
SELECT LAST_NAME, DEPARTMENT_ID FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (20, 50) ORDER BY LAST_NAME ASC;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 47 en 0.004 segundos

	LAST_NAME	DEPARTMENT_ID
1	Atkinson	50
2	Bell	50
3	Bissot	50
4	Bull	50
5	Cabrio	50
6	Chung	50
7	Davies	50
8	Dellinger	50
9	Dilly	50
10	Everett	50
11	Fay	20
12	Feeney	50
13	Fleaur	50
14	Fripp	50
15	Gates	50
16	Gee	50

Figure 11: *SELECT statement.*

6. Modify lab_13_03.sql to display the last name and salary of employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Resave lab_13_03.sql as lab_13_06.sql, see **figure 12**:

```
SELECT LAST_NAME, SALARY AS "Monthly Salary" FROM EMPLOYEES
WHERE (SALARY BETWEEN 5000 AND 12000) AND (DEPARTMENT_ID IN (20, 50));
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 6 en 0 segundos

	LAST_NAME	Monthly Salary
1	Weiss	8000
2	Fripp	8200
3	Kaufling	7900
4	Vollman	6500
5	Mourgos	5800
6	Fay	6000

Figure 12: *SELECT statement.*

7. The HR department needs a report that displays the last name and hire date for all employees who were hired in 1994, see **figure 13**:

Note: In the HR schema there are no employees hired in the year 1994, so you decide to change the year to 2006. This may be due to the version of Oracle DataBase.

```
SELECT LAST_NAME, HIRE_DATE FROM EMPLOYEES
WHERE HIRE_DATE >= '01-ENE-06' AND HIRE_DATE <= '31-DEC-06';
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 24 en 0 segundos

	LAST_NAME	HIRE_DATE
1	Hunold	03/01/06
2	Pataballa	05/02/06
3	Urman	07/03/06
4	Himuro	15/11/06
5	Mikkilineni	28/09/06
6	Rogers	26/08/06
7	Seo	12/02/06
8	Patel	06/04/06
9	Matos	15/03/06
10	Vargas	09/07/06

Figure 13: *SELECT statement.*

8. Create a report to display the last name and job title of all employees who do not have manager, see **figure 14**:

```
SELECT LAST_NAME, JOB_ID FROM EMPLOYEES
WHERE MANAGER_ID IS NULL;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0.003 segundos

	LAST_NAME	JOB_ID
1	King	AD_PRES

Figure 14: *SELECT statement.*

9. Create a report to display the last name, salary, and commission of all employees who earn commissions. Sort data in descending order of salary and commissions. Use the column's numeric position in the ORDER BY clause, see **figure 15**:

```
SELECT LAST_NAME, SALARY, COMMISSION_PCT FROM EMPLOYEES
WHERE COMMISSION_PCT IS NOT NULL ORDER BY 2 DESC, 3 DESC;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 35 en 0.005 segundos

	LAST_NAME	SALARY	COMMISSION_PCT
1	Russell	14000	0.4
2	Partners	13500	0.3
3	Errazuriz	12000	0.3
4	Ozer	11500	0.25
5	Cambrault	11000	0.3
6	Abel	11000	0.3
7	Vishney	10500	0.25
8	Zlotkey	10500	0.2
9	King	10000	0.35
10	Tucker	10000	0.3
11	Bloom	10000	0.2
12	Fox	9600	0.2

Figure 15: *SELECT statement.*

10. Members of the HR department want to have more flexibility with the queries that you are writing. They would like a report that displays the last name and salary of employees who earn more than an amount that the user specifies after a prompt. Save this query to a file named lab_8_10.sql. If you enter 12000 when prompted, the report displays the following results, see **figure 16 and 17**.

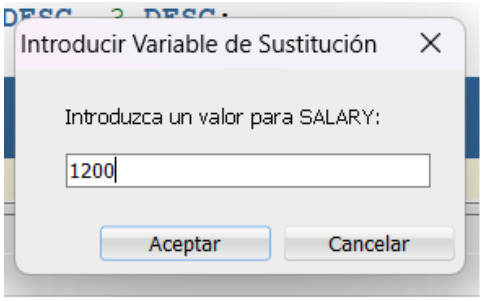
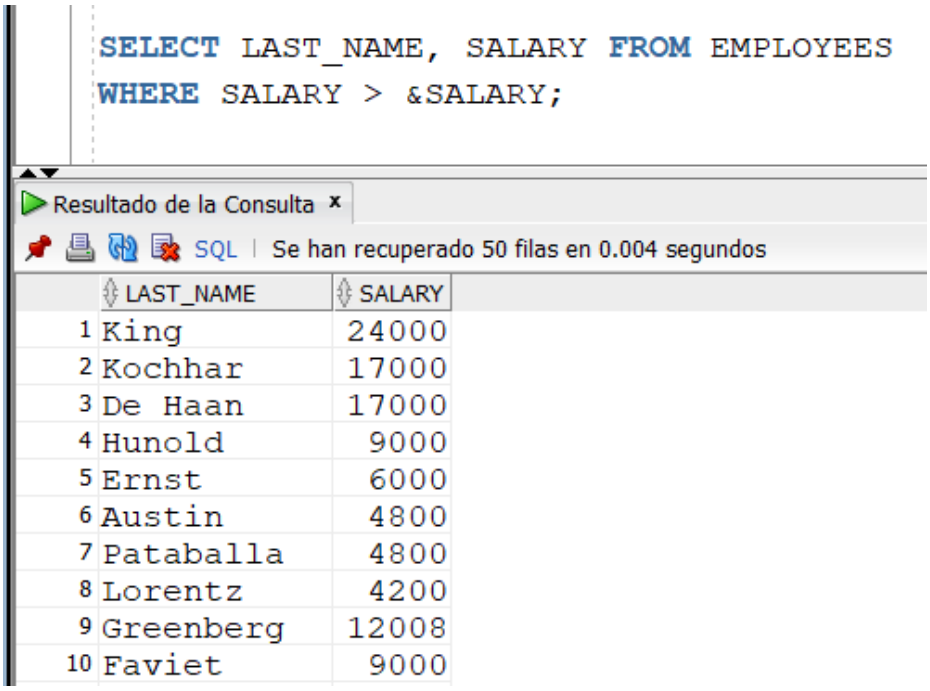


Figure 16: *Request for a substitution variable.*



	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Hunold	9000
5	Ernst	6000
6	Austin	4800
7	Pataballa	4800
8	Lorentz	4200
9	Greenberg	12008
10	Faviet	9000

Figure 17: *SELECT statement.*

11. The HR department wants to run reports based on a manager. Create a query that prompts the user for a manager ID and generates the employee ID, last name, salary, and department for that manager's employees. The HR department wants the ability to sort the report on a selected column, see **figure 18, 19 and 20**.

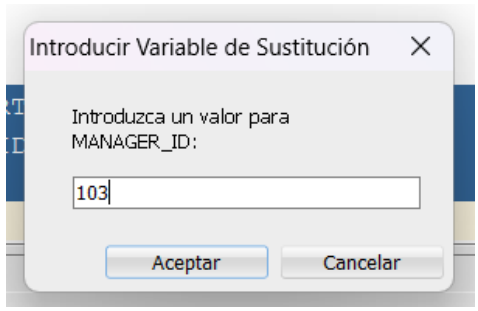


Figure 18: *Request for a substitution variable.*

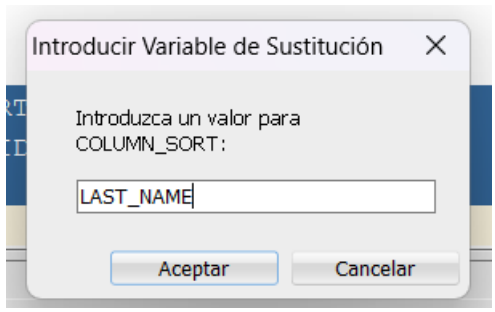
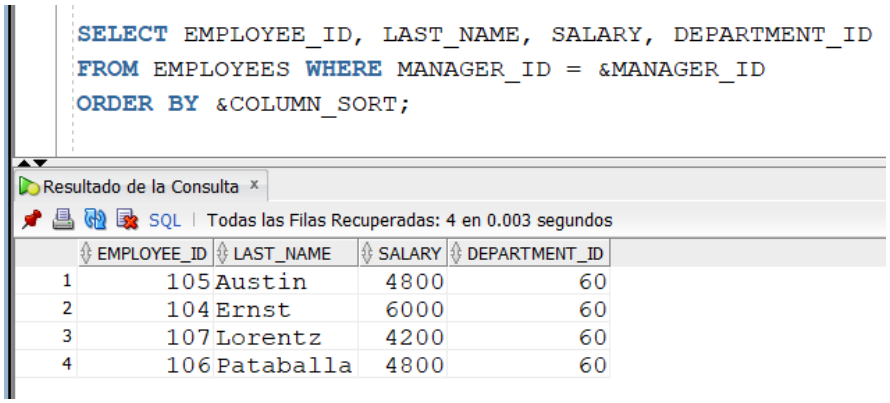


Figure 19: *Request for a substitution variable.*



```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEES WHERE MANAGER_ID = &MANAGER_ID
ORDER BY &COLUMN_SORT;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 4 en 0.003 segundos

	EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
1	105	Austin	4800	60
2	104	Ernst	6000	60
3	107	Lorentz	4200	60
4	106	Pataballa	4800	60

Figure 20: *SELECT statement.*

12. Display all employee last names in which the third letter of the name is "a", see **figure 21**.

```
SELECT LAST_NAME FROM EMPLOYEES WHERE LAST_NAME LIKE '__a%';
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 3 en 0.005 segundos

	LAST_NAME
1	Grant
2	Grant
3	Whalen

Figure 21: *SELECT statement.*

13. Display the last names of all employees who have both an "a" and an "e" in their last name, see **figure 22**.

```
SELECT LAST_NAME FROM EMPLOYEES
WHERE LAST_NAME LIKE '%a%' AND LAST_NAME LIKE '%e%';
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 17 en 0.006 segundos

	LAST_NAME
1	Baer
2	Bates
3	Colmenares
4	Davies
5	De Haan
6	Faviet
7	Fleaur
8	Gates
9	Hartstein
10	Markle
11	Nayer
12	Partners
13	Patel
14	Philtanker
15	Raphaely
16	Sewall
17	Whalen

Figure 22: *SELECT statement.*

14. Display the last name, job, and salary for all employees whose jobs are either those of a sales representative or of a stock clerk, and whose salaries are not equal to $2,500, $3,500, or $7,000, see **figure 23**.

```
SELECT LAST_NAME, JOB_ID, SALARY FROM EMPLOYEES
WHERE (JOB_ID = 'ST_CLERK' OR JOB_ID = 'SA_REP')
AND SALARY NOT IN (2500, 3500, 7000);
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 43 en 0.01 segundos

	LAST_NAME	JOB_ID	SALARY
1	Nayer	ST_CLERK	3200
2	Mikkilineni	ST_CLERK	2700
3	Landry	ST_CLERK	2400
4	Markle	ST_CLERK	2200
5	Bissot	ST_CLERK	3300
6	Atkinson	ST_CLERK	2800
7	Olson	ST_CLERK	2100
8	Mallin	ST_CLERK	3300
9	Rogers	ST_CLERK	2900
10	Gee	ST_CLERK	2400
11	Philtanker	ST_CLERK	2200
12	Ladwig	ST_CLERK	3600
13	Stiles	ST_CLERK	3200
14	Seo	ST_CLERK	2700
15	Davies	ST_CLERK	3100
16	Matos	ST_CLERK	2600
17	Tucker	SA_REP	10000
18	Bernstein	SA_REP	9500
19	Hall	SA_REP	9000
20	Olsen	SA_REP	8000

Figure 23: *SELECT statement.*

15. Modify lab_8_06.sql to display the last name, salary, and commission for all employees whose commission is 20%. Resave lab_8_06.sql as lab_8_15.sql. Rerun the statement in lab_8_15.sql, see **figure 24**.

```
SELECT LAST_NAME AS "Employee",
       SALARY AS "Monthly Salary",
       COMMISSION_PCT
FROM EMPLOYEES WHERE COMMISSION_PCT = 0.20;
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 7 en 0.006 segundos

	Employee	Monthly Salary	COMMISSION_PCT
1	Zlotkey	10500	0.2
2	Olsen	8000	0.2
3	Cambrault	7500	0.2
4	Bloom	10000	0.2
5	Fox	9600	0.2
6	Taylor	8600	0.2
7	Livingston	8400	0.2

Figure 24: *SELECT statement.*

4 Pre-assessment

In this section you will find the Pre-assessment.

Criteria to be evaluate	Does it comply?	(%)
COMPLIES WITH THE REQUESTED FUNCTIONALITY	YES	
HAS THE CORRECT INDENTATION	YES	
HAS AN EASY WAY TO ACCESS THE PROVIDED FILES	YES	
HAS A REPORT WITH IDC FORMAT	YES	
REPORT INFORMATION IS FREE OF SPELLING ERRORS	YES	
DELIVERED IN TIME AND FORM	YES	
IS FULLY COMPLETED (SPECIFY THE PERCENTAGE COMPLETED)	YES	100%

5 Conclusion

The Oracle DML statements are transcendental in the handling of SQL statements at the level of both administrator and database programmer, since they allow the data manipulation of database schemes regardless of the platform used to generate it. This kind of statements can provide you data treatment mechanisms during daily programmer's days.

SQL language allows the realization of projection and selection of data to satisfy the needs of reports that may be required for a programmer, developer or end user.

This practice number 8 helped me to practice the uses of the SELECT statement for data retrieval. Finally, an important thing to mention is that the SQL language allows the projection and selection of data to satisfy the needs of reports that may be necessary for a programmer, developer or end user.