

Autonomous University of Zacatecas

ACADEMIC UNIT OF ELECTRICAL ENGINEERING

ACADEMIC PROGRAM OF SOFTWARE ENGINEERING



DATABASE SYSTEMS LABORATORY II
PRACTICE 13 -USING SUBQUERIES

PROFESSOR:
ALDONSO BECERRA SÁNCHEZ

STUDENT:
DANIEL ALEJANDRO MORALES CASTILLO

1 Introduction

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

In the theory class we saw the corresponding to chapter 7, in which we saw the subqueries and the utility they have to get more complex records

The use of subqueries is a technique that allows you to use the result of a `SELECT` table in another `SELECT` query. It allows solving complex queries by using previous results obtained through another query.

2 Development

Activity 1

Write the section that describes the work developed in the following activities. 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.

DEFINE SUBQUERIES

1. Consider this generic description of a SELECT statement: SELECT selectlist FROM table WHERE condition GROUP BY expression1 HAVING expression2 ORDER BY expression3 ; Where could subqueries be used? (Choose all correct answers.)

- A. selectlist
- B. expression2
- C. condition
- D. expression1
- E. table
- F. expression3

Answer: C, E

The sub queries returns rows, these row can be used to be compared in a condition, this means we can use sub queries in the WHERE and HAVING clauses, because we retrieve data to be compared and restrict the data.

2. A query can have a subquery embedded within it. Under what circumstances could there be more than one subquery? (Choose the best answer.)

- A. Subqueries can be embedded within each other with no practical limitations on depth.
- B. It is possible to embed a single-row subquery inside a multiple-row subquery, but not the other way around.
- C. The outer query can have multiple inner queries, but they must not be embedded within each other.
- D. The outer query can include an inner query. It is not possible to have another query within the inner query.

Answer: D

There is no limit embedding, a query can have an inner sub query and that sub query can have an inner sub query, the limitation is in the logic of your

needs, you must use the suitable operator depending of the kind of query

3. Consider this statement: select employeeid, lastname from employees where salary < (select avg(salary) from employees); When will the subquery be executed? (Choose the best answer.)

- A. It will be executed once for every row in the EMPLOYEES table.
- B. It will be executed after the outer query.
- C. It will be executed concurrently with the outer query.
- D. It will be executed before the outer query

Answer: A

It works like the functions, the deepest query will be executed first because the results of the subquery are needed to compare and restrict the rows of the outer query

4. Consider this statement: select o.employeeid, o.lastname from employees o where o.salary < (select avg(i.salary) from employees i where i.departmentid = o.departmentid); When will the subquery be executed? (Choose the best answer.)

- A. It will be executed once for every row in the EMPLOYEES table.
- B. It will be executed after the outer query.
- C. It will be executed concurrently with the outer query.
- D. It will be executed before the outer query

Answer: D

The reason is because the outer query retrieve all the records, the sub query compares with the department id of the outer query, that means the comparisons will be executed once for each record of the table.

**DESCRIBE THE TYPES OF PROBLEMS THAT THE SUB-
QUERIES CAN SOLVE**

5. Consider the following statement: select lastname from employees join departments on employees.departmentid = departments.departmentid where departmentname='Executive'; and this statement: select lastname from employees where departmentid in (select departmentid from departments where departmentname='Executive'); What can be said about the two statements? (Choose two correct answers.)

- A. Both statements will always run successfully, even if there are two departments with DEPARTMENTNAME 'Executive.'
- B. The two statements could generate different results.
- C. The first statement will always run successfully; the second statement will error if there are two departments with DEPARTMENTNAME 'Executive.'
- D. The two statements should generate the same result.

Answer: A,D

Both queries will return the same result, it won't return an error if there are two departments with the same name because we are using the department id to differentiate the departments.

LIST THE TYPES OF SUBQUERIES

6. What are the distinguishing characteristics of a scalar subquery? (Choose two correct answers.)

- A. A scalar subquery returns one row.
- B. A scalar subquery cannot be used as a correlated subquery.
- C. A scalar subquery cannot be used in the SELECT LIST of the parent query.
- D. A scalar subquery returns one column.

Answer: A, D

A scalar subquery returns exactly only one value, this value is used by the outer query.

7. Which comparison operator can be used with multiple-row subqueries? (Choose the best answer.)

- A. ALL
- B. ANY
- C. IN
- D. NOT IN
- E. All the above can be used.

Answer: E

All the operators shown in the answer are used to compare multiple results retrieved by a sub query, ALL and ANY need an comparing operator.

WRITE SINGLE-ROW AND MULTIPLE-ROW SUBQUERIES

8. Consider this statement: select lastname, (select count(*) from departments) from employees where salary = (select salary from employees); What is wrong with it? (Choose the best answer.)

- A. The statement will run but is extremely inefficient because of the need to run the second subquery once for every row in EMPLOYEES.
- B. The statement will fail if the second query returns more than one row.
- C. The statement will fail because the subquery in the SELECT list references a table that is not listed in the FROM clause.
- D. Nothing is wrong—the statement should run without error.

Answer: C

The WHERE clause are using a single-row operator (=) and the sub query will returns more than one row, you need to use the correct operator, the operator depends of the kind of sub query used.

9. Which of the following statements are equivalent? (Choose two answers.)

- A. select employeeid from employees where salary < all (select salary from employees where departmentid=10);
- B. select employeeid from employees e join departments d on e.departmentid= d.departmentid where e.salary < (select min(salary) from employees) and d.departmentid=10;
- C. select employeeid from employees where salary not <= any (select salary from employees where departmentid=10);
- D. select employeeid from employees where salary < (select min(salary) from employees where departmentid=10);

Answer: A, D

Both queries returns the same result, both will returns the employee id of the employees with less salary then all the salaries of the employees of department 10.

10. Consider this statement, which is intended to prompt for an employee's name and then find all employees who have the same job as the first employee: select lastname,employeeid from employees where jobid = (select jobid from employees where lastname = 'Name'); What would happen if a value were given for Name that did not match with any row in EMPLOYEES? (Choose the best answer.)

- A. The statement would return every row in the table.
- B. The statement would fail with an error.
- C. The statement would return all rows where JOBID is NULL.
- D. The statement would return no rows.

Answer: C

The name do not match, the sub query return no rows, there is no row to compare an the syntax are good, the query will success but nothing is retrieved.

Activity 2:

- How can you best design subqueries such that they will not fail with “ORA01427: single-row subquery returns more than one row” errors?

The easy way is using multiple row operators, these operators works for one or more results, but the correct way is to design the sub query and observe and choose what kind of operator must be used, single-row operators are used by single-row queries, multiple-row operators are used by multiple-row queries.

- Star Transformation. An extension of the use of subqueries as an alternative to a join is to enable the star transformation often needed in data warehouse applications. Consider a large table recording sales. Each sale is marked as being of a particular product to a particular buyer through a particular channel. These attributes are identified by codes, used as foreign keys to dimension tables with rows that describe each product, buyer, and channel. To identify all sales of books to buyers in Germany through Internet orders, one could run a query like this:

```
select ... from sales s, products p, buyers b, channels c where
s.prodcode=p.prodcode and s.buycode=b.buycode and s.chancode=c.chancode
and p.product='Books' and b.country='Germany' and c.channel='Internet';
```

This query uses the WHERE clause to join the tables and then to filter the results. The following is an alternative query that will yield the same result:

```
select ... from sales where prodcode in (select prodcode from products where product='Books') and buycode in (select buycode from buyers where country='Germany') and chancode in (select chancode from channels where channel='Internet');
```

 The rewrite of the first statement to the second is the star transformation. Apart from being an inherently more elegant structure (most SQL developers with any sense of aesthetics will agree with that), there are technical reasons why the database may be able to execute it more efficiently than the original query. Also, star queries are easier to maintain; it is very simple to add more dimensions to the query or to replace the single literals ('Books,' 'Germany,' and 'Internet') with lists of values.

NOTE (on the JOB): There is an instance initialization parameter, `STARTRANSFORMATIONENABLED`, which (if set to true) will permit the Oracle query optimizer to re-write code into star queries. Sometimes there is a choice between using a subquery or using some other technique: the star transformation is a case in point. Which is better?

I think the second one are the best option, it is less code, it is simplest to understand i think, it has a better structure and it is more easy to modify the query, maybe the option depend of the likes

of the developer but we need to keep it simple.

Activity 3:

This exercise must be performed using HR schema.

- a) Write a query that uses subqueries in the column projection list. The query will report on the current (date of today) numbers of departments and staff:



The screenshot shows an Oracle SQL Developer window titled 'hr_2021'. The 'Hoja de Trabajo' (Worksheet) tab is active, displaying the following SQL query:

```
SELECT SYSDATE TODAY, (SELECT COUNT(*) FROM DEPARTMENTS) DEPARTMENT_COUNT, (SELECT COUNT(*) FROM EMPLOYEES ) EMPLOYEES_COUNT FROM DUAL;
```

The 'Resultado de la Consulta' (Query Result) tab shows the results of the query. The status bar indicates 'Todas las Filas Recuperadas: 1 en 0.01 segundos' (All rows retrieved: 1 in 0.01 seconds). The results are displayed in a table with the following columns and data:

	TODAY	DEPARTMENT_COUNT	EMPLOYEES_COUNT
1	11/11/21	27	107

b) Write a query to identify all the employees who are managers. This will require using a subquery in the WHERE clause to select all the employees whose EMPLOYEEID appears as a MANAGERID:

```
--b
SELECT LAST_NAME AS WHO_ARE MANAGERS FROM EMPLOYEES
WHERE EMPLOYEE_ID IN (SELECT MANAGER_ID FROM EMPLOYEES);
```

Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 18 en 0.004 segundos	
WHO_ARE MANAGERS	
1	Cambrault
2	De Haan
3	Errazuriz
4	Fripp
5	Greenberg
6	Hartstein
7	Higgins
8	Hunold
9	Kaufling
10	King
11	Kochhar
12	Mourgos
13	Partners
14	Raphaely
15	Russell
16	Vollman
17	Weiss
18	Zlotkey

c) Write a query to identify the highest salary paid in each country.
This will require using a subquery in the FROM clause:

```
--C
SELECT "MAX(SALARY)" , COUNTRY_ID
FROM (SELECT MAX(E.SALARY) "MAX(SALARY)", C.COUNTRY_ID FROM EMPLOYEES E JOIN DEPARTMENTS D
ON (E.DEPARTMENT_ID = D.DEPARTMENT_ID) JOIN LOCATIONS LOC ON (D.LOCATION_ID = LOC.LOCATION_ID)
JOIN COUNTRIES C ON (LOC.COUNTRY_ID = C.COUNTRY_ID) GROUP BY C.COUNTRY_ID);
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 4 en 0.014 segundos

	MAX(SALARY)	COUNTRY_ID
1	24000	US
2	13000	CA
3	10000	DE
4	14000	UK

d) Write a query that will identify all employees who work in departments located in the United Kingdom. This will require three levels of nested subqueries in the WHERE clause:

```
--d
SELECT LAST_NAME FROM EMPLOYEES WHERE DEPARTMENT_ID IN (SELECT D.DEPARTMENT_ID FROM DEPARTMENTS D JOIN LOCATIONS LOC ON (D.LOCATION_ID = LOC.LOCATION_ID)
WHERE LOC.LOCATION_ID IN (SELECT LOC.LOCATION_ID FROM LOCATIONS LOC JOIN COUNTRIES C ON (C.COUNTRY_ID = LOC.COUNTRY_ID) AND C.COUNTRY_ID='UK'));
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 35 en 0.015 segundos

LAST_NAME
1 Mavris
2 Zlotkey
3 Vishney
4 Tuvault
5 Tucker
6 Taylor
7 Sully
8 Smith
9 Smith
10 Sewall
11 Russell
12 Partners
13 Ozer
14 Olsen
15 McEwen
16 Marvins
17 Livingston
18 Lee
19 Kumar
20 King
21 Johnson
22 Hutton
23 Hall
24 Greene
25 Fox

e) Write a query to identify all the employees who earn more than the average and who work in any of the IT departments. This will require two subqueries in the WHERE clause, not nested:

<pre>SELECT LAST_NAME FROM EMPLOYEES WHERE DEPARTMENT_ID IN(SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE DEPARTMENT_NAME='IT') AND SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES);</pre>	
Resultado de la Consulta x	
Todas las Filas Recuperadas: 1 en 0.003 segundos	
LAST_NAME	
1 Hunold	

f) Write a query to determine who earns more than Mr. Tobias. Write a query to determine who earns more than Mr. Taylor. Write the sentence to be useful no matter the number of rows returned by the subquery in the WHERE clause (use \exists operator). There can be several solutions (show two):

The screenshot shows a SQL query execution environment. At the top, a query is entered in a text area:

```
--f  
SELECT LAST_NAME FROM EMPLOYEES WHERE SALARY > (SELECT SALARY FROM EMPLOYEES WHERE LOWER(LAST_NAME) = LOWER ('%LAST_NAME%')) ORDER BY LAST_NAME;
```

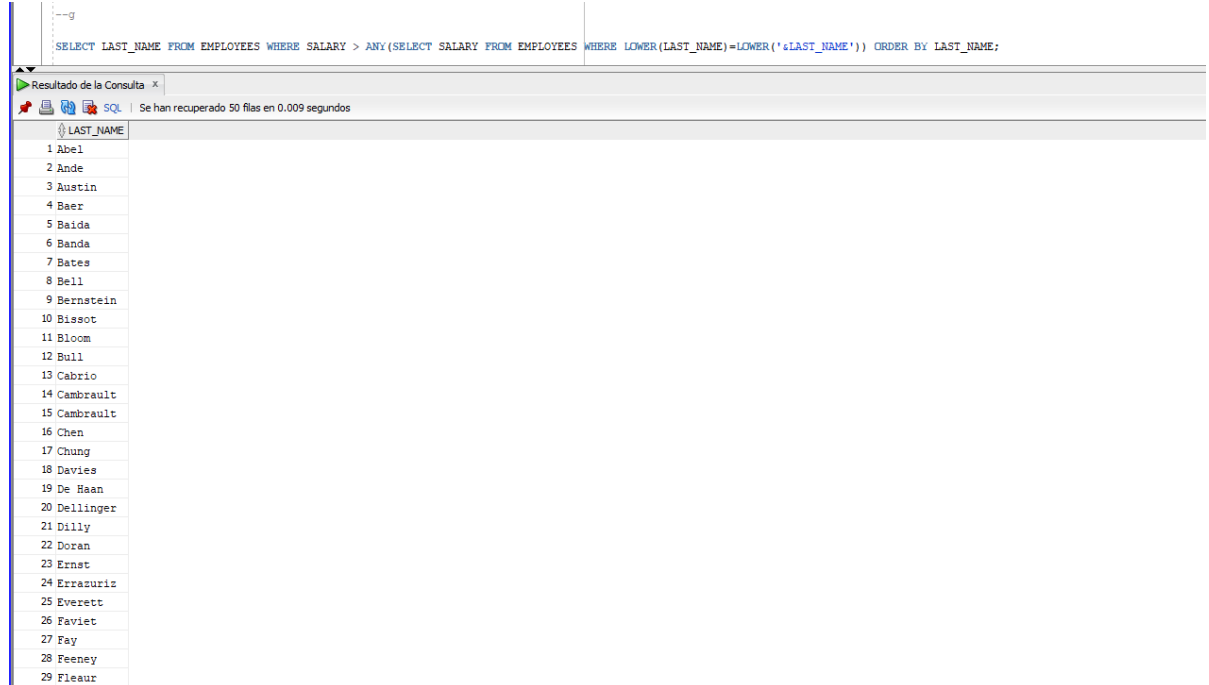
Below the query, a status bar indicates "Se han recuperado 50 filas en 0.008 segundos". A table of results is displayed with the column "LAST_NAME". The results are as follows:

	LAST_NAME
1	Abel
2	Ande
3	Austin
4	Baer
5	Baida
6	Banda
7	Bates
8	Bell
9	Bernstein
10	Bissot
11	Bloom
12	Bull
13	Cabrio
14	Cambrault
15	Cambrault
16	Chen
17	Chung
18	Davies
19	De Haan
20	Dellinger
21	Dilly
22	Doran
23	Ernst
24	Errazuriz
25	Everett

On the right side, a dialog box titled "Introducir Variable de Sustitución" is open. It prompts the user to "Introduzca un valor para LAST_NAME:" and has a text input field containing "TOBIAS". There are "Aceptar" and "Cancelar" buttons at the bottom of the dialog.

--f2	
SELECT LAST_NAME FROM EMPLOYEES WHERE EMPLOYEE_ID IN (SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE SALARY > (SELECT SALARY FROM EMPLOYEES WHERE LOWER(LAST_NAME) = LOWER ('%LAST_NAME'))) ORDER BY LAST_NAME;	
Resultado de la Consulta x	
Se han recuperado 50 filas en 0.004 segundos	
LAST_NAME	
1 Abel	
2 Ande	
3 Austin	
4 Baer	
5 Baida	
6 Banda	
7 Bates	
8 Bell	
9 Bernstein	
10 Bissot	
11 Bloom	
12 Bull	
13 Cabrio	
14 Cambrault	
15 Cambrault	
16 Chen	
17 Chung	
18 Davies	
19 De Haan	
20 Dellinger	
21 Dilly	
22 Doran	
23 Ernst	
24 Errazuriz	
25 Everett	

g) Later exercise included a query that attempted to find all employees whose salary is higher than that of a nominated employee. There are other queries that will run successfully; construct two other solutions, one using the ANY comparison operator, the other using the MIN aggregation function. Now that you have several solutions, do they all give the same result? All these “solutions” are in fact just ways of avoiding error. They do not necessarily give the result the user wants, and they may not be consistent. What change needs to be made to give a consistent, unambiguous, result?



The screenshot shows an Oracle SQL Developer window with a query in the top pane and its results in the bottom pane. The query is:

```
--g
SELECT LAST_NAME FROM EMPLOYEES WHERE SALARY > ANY(SELECT SALARY FROM EMPLOYEES WHERE LOWER(LAST_NAME)=LOWER('&LAST_NAME')) ORDER BY LAST_NAME;
```

The results pane shows a list of 29 employees, ordered by last name. The status bar indicates that 50 rows were retrieved in 0.009 seconds.

LAST_NAME
1 Abel
2 Ande
3 Austin
4 Baer
5 Baida
6 Banda
7 Bates
8 Bell
9 Bernstein
10 Bissot
11 Bloom
12 Bull
13 Cabrio
14 Cambrault
15 Cambrault
16 Chen
17 Chung
18 Davies
19 De Haan
20 Dellinger
21 Dilly
22 Doran
23 Ernst
24 Errazuriz
25 Everett
26 Faviet
27 Fay
28 Feeney
29 Fleaur

SELECT LAST_NAME FROM EMPLOYEES WHERE SALARY > (SELECT MIN(SALARY) FROM EMPLOYEES WHERE LOWER(LAST_NAME)=LOWER('&LAST_NAME')) ORDER BY LAST_NAME;	
Resultado de la Consulta x	
SQL Se han recuperado 50 filas en 0.004 segundos	
LAST_NAME	
1 Abel	
2 Ande	
3 Austin	
4 Baer	
5 Baida	
6 Banda	
7 Bates	
8 Bell	
9 Bernstein	
10 Bissot	
11 Bloom	
12 Bull	
13 Cabrio	
14 Cambrault	
15 Cambrault	
16 Chen	
17 Chung	
18 Davies	
19 De Haan	
20 Dellinger	
21 Dilly	
22 Doran	
23 Ernst	
24 Errazuriz	
25 Everett	
26 Faviet	
27 Fay	
28 Feeney	
...	

h) Design a query that will prompt for a department name (no matter if the input is lower or upper case) and list the last name of every employee in that department, use a subquery in the WHERE clause: For instance, if the input is a department name with the string “Pu...”:

```
--h
SELECT DEPARTMENT_NAME FROM DEPARTMENTS;
SELECT E.LAST_NAME, D.DEPARTMENT_NAME FROM EMPLOYEES E JOIN DEPARTMENTS D ON(D.DEPARTMENT_ID = E.DEPARTMENT_ID)
WHERE D.DEPARTMENT_ID IN(SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE LOWER(DEPARTMENT_NAME) LIKE LOWER ('%&DEP_NAME%'));
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 7 en 0.002 segundos

	LAST_NAME	DEPARTMENT_NAME
1	Raphaely	Purchasing
2	Khoo	Purchasing
3	Baida	Purchasing
4	Tobias	Purchasing
5	Himuro	Purchasing
6	Colmenares	Purchasing
7	Baer	Public Relations

Activity 4:

You will write complex queries using nested SELECT statements. For practice questions, you may want to create the inner query first. Make sure that it runs and produces the data that you anticipate before you code the outer query. 1. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

```
SELECT LAST_NAME, HIRE_DATE FROM EMPLOYEES WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '&NAME') AND LAST_NAME <> '&NAME';
```

Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 33 en 0.003 segundos	
LAST_NAME	HIRE_DATE
1 Russell	01/10/04
2 Partners	05/01/05
3 Errazuriz	10/03/05
4 Cambrault	15/10/07
5 Tucker	30/01/05
6 Bernstein	24/03/05
7 Hall	20/08/05
8 Olsen	30/03/06
9 Cambrault	09/12/06
10 Tuvault	23/11/07
11 King	30/01/04
12 Sully	04/03/04
13 McEwen	01/08/04
14 Smith	10/03/05
15 Doran	15/12/05
16 Sewall	03/11/06
17 Vishney	11/11/05
18 Greene	19/03/07
19 Marvins	24/01/08
20 Lee	23/02/08
21 Ande	24/03/08
22 Banda	21/04/08
23 Ozer	11/03/05
24 Bloom	23/03/06
25 Fox	24/01/06
26 Smith	23/02/07
27 Bates	24/03/07
28 Kumar	21/04/08
29 Abel	11/05/04

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

`SELECT EMPLOYEE_ID, LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES) ORDER BY SALARY;`

Resultado de la Consulta x

Se han recuperado 50 filas en 0.004 segundos

	EMPLOYEE_ID	LAST_NAME	SALARY
1	203	Mavris	6500
2	123	Vollman	6500
3	165	Lee	6800
4	113	Popp	6900
5	155	Tuvault	7000
6	161	Sewall	7000
7	178	Grant	7000
8	164	Marvins	7200
9	172	Bates	7300
10	171	Smith	7400
11	154	Cambrault	7500
12	160	Doran	7500
13	111	Sciarra	7700
14	112	Urman	7800
15	122	Kaufling	7900
16	120	Weiss	8000
17	159	Smith	8000

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains the letter “u.” Save your SQL statement as lab1303.sql. Run your query.

```
SELECT EMPLOYEE_ID, LAST_NAME
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '%u%');
```

Salida de Script x	Resultado de la Consulta x
SQL Se han recuperado 50 filas en 0.004 segundos	
EMPLOYEE_ID	LAST_NAME
1	107 Lorentz
2	106 Pataballa
3	105 Austin
4	104 Ernst
5	103 Hunold
6	199 Grant
7	198 OConnell
8	197 Feeney
9	196 Walsh
10	195 Jones
11	194 McCain
12	193 Everett
13	192 Bell
14	191 Perkins
15	190 Gates
16	189 Dilly
17	188 Chung
18	187 Smith

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700

Modify the query so that the user is prompted for a location ID. Save this to a file named lab1304.sql.

SQL Query Editor showing the modification of the query to prompt for a location ID.

```
SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID
FROM EMPLOYEES WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE LOCATION_ID =1700);
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 18 en 0.005 segundos

	LAST_NAME	DEPARTMENT_ID	JOB_ID
1	King	90	AD_PRES
2	Kochhar	90	AD_VP
3	De Haan	90	AD_VP
4	Greenberg	100	FI_MGR
5	Faviet	100	FI_ACCOUNT
6	Chen	100	FI_ACCOUNT
7	Sciarra	100	FI_ACCOUNT
8	Urman	100	FI_ACCOUNT
9	Popp	100	FI_ACCOUNT
10	Raphaely	30	PU_MAN
11	Khoo	30	PU_CLERK
12	Baida	30	PU_CLERK
13	Tobias	30	PU_CLERK
14	Himuro	30	PU_CLERK
15	Colmenares	30	PU_CLERK
16	Whalen	10	AD_ASST
17	Higgins	110	AC_MGR

SQL Query Editor showing the modified query with a prompt for the location ID.

```
SELECT LAST_NAME, DEPARTMENT_ID, JOB_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE LOCATION_ID = &location);
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 18 en 0.003 segundos

	LAST_NAME	DEPARTMENT_ID	JOB_ID
1	King	90	AD_PRES
2	Kochhar	90	AD_VP
3	De Haan	90	AD_VP
4	Greenberg	100	FI_MGR
5	Faviet	100	FI_ACCOUNT
6	Chen	100	FI_ACCOUNT
7	Sciarra	100	FI_ACCOUNT
8	Urman	100	FI_ACCOUNT
9	Popp	100	FI_ACCOUNT
10	Raphaely	30	PU_MAN
11	Khoo	30	PU_CLERK
12	Baida	30	PU_CLERK
13	Tobias	30	PU_CLERK
14	Himuro	30	PU_CLERK
15	Colmenares	30	PU_CLERK
16	Whalen	10	AD_ASST
17	Higgins	110	AC_MGR

5. Create a report for HR that displays the last name and salary of every employee who reports to King.

```
SELECT LAST_NAME, SALARY
FROM EMPLOYEES
WHERE MANAGER_ID IN (SELECT EMPLOYEE_ID FROM EMPLOYEES WHERE LAST_NAME='King');
```

Salida de Script x		Resultado de la Consulta x	
		SQL Todas las Filas Recuperadas: 14 en 0.002 segundos	
LAST_NAME	SALARY		
1 Kochhar	17000		
2 De Haan	17000		
3 Raphaely	11000		
4 Weiss	8000		
5 Fripp	8200		
6 Kaufling	7900		
7 Vollman	6500		
8 Mourgos	5800		
9 Russell	14000		
10 Partners	13500		
11 Errazuriz	12000		
12 Cambrault	11000		
13 Zlotkey	10500		
14 Hartstein	13000		

6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

```
SELECT DEPARTMENT_ID, LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM DEPARTMENTS WHERE DEPARTMENT_NAME = 'Executive');
```

Salida de Script x Resultado de la Consulta x
SQL | Todas las Filas Recuperadas: 3 en 0.002 segundos

DEPARTMENT_ID	LAST_NAME	JOB_ID
1	King	AD_PRES
2	Kochhar	AD_VP
3	De Haan	AD_VP

7. Modify the query in lab1303.sql to display the employee number, last name, and salary of all employees who earn more than the average salary, and who work in a department with any employee whose last name contains a “u.” Resave lab1303.sql as lab1307.sql. Run the statement in lab1307.sql.

```
SELECT EMPLOYEE_ID, LAST_NAME, SALARY FROM EMPLOYEES
WHERE SALARY > (SELECT AVG(SALARY) FROM EMPLOYEES) AND DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME LIKE '%u%');
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 36 en 0.004 segundos

EMPLOYEE_ID	LAST_NAME	SALARY
1	103 Hunold	9000
2	123 Vollman	6500
3	122 Kaufling	7900
4	121 Fripp	8200
5	120 Weiss	8000
6	177 Livingston	8400
7	176 Taylor	8600
8	175 Hutton	8800
9	174 Abel	11000
10	172 Bates	7300
11	171 Smith	7400
12	170 Fox	9600
13	169 Bloom	10000
14	168 Ozer	11500
15	165 Lee	6800
16	164 Marvins	7200
17	163 Greene	9500

3 PRE-EVALUATION

Practices pre-Assessment for Database Systems Laboratory II Pre-Assessment PRACTICE 13 carried out by student

**1 COMPLIES WITH THE REQUESTED FUNCTIONALITY
YES**

**4 HAS THE CORRECT INDENTATION
YES**

**6 HAS AN EASY WAY TO ACCESS THE PROVIDED FILES
YES**

**7 HAS A REPORT WITH IDC FORMAT
YES**

**8 REPORT INFORMATION IS FREE OF SPELLING ERRORS
YES**

**9 DELIVERED IN TIME AND FORM
YES**

**10 IS FULLY COMPLETED (SPECIFY THE PERCENTAGE
COMPLETED)
YES,100 percent**

4 Conclusion

This was a very complete practice, always it is important to review and practice

I found the practice interesting, I think I could learn more since in the examples that come in the oracle document they are very few and very simple or at least they only give you the idea, in this practice I was able to practice more with the subqueries and improve on it.