Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Academic Program of Software Engineering

Database Systems Laboratory II
Practice 11 - Reporting Aggregated Data
Using the Group Functions

Professor:
Aldonso Becerra Sánchez

Student:
Daniel Alejandro Morales Castillo

1 Introduction

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.

In the theory class we saw the topic of group functions in chapter 5, many of them are very useful and we can use them to create longer queries with more specific data.

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.

DESCRIBE THE GROUP FUNCTIONS

1. What result is returned by the following statement? SELECT COUNT(*) FROM DUAL; (Choose the best answer.)

A. NULL
B. 0
C. 1
D. None of the above

Answer: C

The count function return the total of records counted, the function count(*) used with DUAL return 1 because we are doing ONE query to the DUAL table.

2. Choose one correct statement regarding group functions

A. Group functions may only be used when a GROUP BY clause is present.
B. Group functions can operate on multiple rows at a time.
C. Group functions only operate on a single row at a time.
D. Group functions can execute multiple times within a single group

Answer: B

The group functions are functions that operate multiple records to create result per group, it is a similar description.

IDENTIFY THE AVAILABLE GROUP FUNCTIONS

3. What value is returned after executing the following statement? SELECT SUM(SALARY) FROM EMPLOYEES; Assume there are 10 employee records and each contains a SALARY value of 100, except for 1, which has a null value in the SALARY field. (Choose the best answer.)

A. 900
B. 1000
C. NULL
D. None of the above

Answer: A

The group functions ignores the null values, the function is going to take only the non null values and return 900.

4. Which values are returned after executing the following statement? SELECT COUNT(*), COUNT(SALARY) FROM EMPLOYEES; Assume there are 10 employee records and each contains a SALARY value of 100, except for 1, which has a null value in their SALARY field. (Choose all that apply.)

A. 10 and 10
B. 10 and NULL
C. 10 and 9
D. None of the above

Answer: C

COUNT(*) return a number that represents the number of records, it does not mater if the records have null values, the COUNT(SALARY) will return the count of the non null values, that is why the SELECT statement returns 10 and 9.

5. What value is returned after executing the following statement? SELECT AVG(NVL(SALARY,100)) FROM EMPLOYEES; Assume there are ten employee records and each contains a SALARY value of 100, except for one employee, who has a null value in the SALARY field. (Choose the best answer.

A. NULL
B. 90
C. 100
D. None of the above

Answer: 100
Using the function NVL the null values on the SALARY column will be changed to 100, the average of ten employees with a salary of 100 each of them is equals to 100.

GROUP DATA USING THE GROUP BY CLAUSE

6. What value is returned after executing the following statement? SELECT SUM((AVG(LENGTH(NVL(SALARY,0))))) FROM EMPLOYEES GROUP BY SALARY; Assume there are ten employee records and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. (Choose the best answer.)

A. An error is returned
B. 3
C. 4
D. None of the above

Answer: 3
First the function NVL changed the null value to 0, the average of the lengths of the values of salary is 3, the SUM of 3 is 3.

7. How many records are returned by the following query? SELECT SUM(SALARY), DEPARTMENTID FROM EMPLOYEES GROUP BY DEPARTMENTID; Assume there are 11 nonnull and 1 null unique DEPARTMENTID values. All records have a nonnull SALARY value. (Choose the best answer.)

A. 12
B. 11
C. NULL
D. None of the above

Answer: 12
The groups formed are 12, the null value also represents a group, the query will return 12 records.

8. What values are returned after executing the following statement? SELECT JOBID, MAXSALARY FROM JOBS GROUP BY MAXSALARY; Assume that the JOBS table has ten records with the same JOBID value of DBA and the same MAXSALARY value of 100. (Choose the best answer.)

A. One row of output with the values DBA, 100
B. Ten rows of output with the values DBA, 100
C. An error is returned
D. None of the above

Answer: C
An error is returned because the clause GROUP BY is incomplete, all the columns in the SELECT statement also must be in the clause GROUP BY.

INCLUDE OR EXCLUDE GROUPED ROWS USING THE HAVING CLAUSE

9. How many rows of data are returned after executing the following statement? SELECT DEPTID, SUM(NVL(SALARY,100)) FROM EMP GROUP BY DEPTID HAVING SUM(SALARY) ¿ 400; Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPTID values of 10 and 20, respectively. (Choose the best answer.)

A. Two rows
B. One row
C. Zero rows
D. None of the above

Answer: A

There are two big groups represented by the DEPTS, it will be returned two rows with the SUM of each DEPT.

10. How many rows of data are returned after executing the following statement? SELECT DEPTID, SUM(SALARY) FROM EMP GROUP BY DEPTID HAVING SUM(NVL(SALARY,100)) ¿ 400; Assume the EMP table has ten rows and each contains a SALARY value of 100, except for one, which has a null value in the SALARY field. The first and second five rows have DEPTID values of 10 and 20, respectively. (Choose the best answer.)

A. Two rows
B. One row
C. Zero rows
D. None of the above

Answer: A

The two groups of DEPT will be showed because both comply with the HAVING condition

Activity 2:

Propose an answer to the following issues:

- You would like to retrieve the earliest date from a column that stores DATE information. Can a group function be utilized to retrieve this value?

Yes, the function MIN also works with dates, i can use the function MIN with the column that contain the dates and the earliest date will be returned.

- Summary statistics are required by senior management. This includes details like number of employees, total staff salary cost, lowest salary, and highest salary values. Can such a report be drawn using one query?

Yes, all of these requirements can be solved using group functions, all use group functions and when you use only group functions all can be retrieved by one query. The total of employees can be solved with COUNT, salary cost can be solved with SUM, lowest salary can be solved using MIN, highest salary can be solved using MAX.

- You are asked to list the number of unique jobs performed by employees in the organization. Counting the JOBID records will give you all the jobs. Is it possible to count the unique jobs?

Yes, you can use the reserved word DISTINCT in the COUNT function, something like COUNT(DISTINCT job id), and only the unique values will be counted.

- You wish to print name badges for the staff who work as sales representatives. Can the length of the shortest and longest LASTNAME values be determined for these employees?

Yes, you can use LENGTH in the functions MAX and MIN, they will return you the maximum an d minimum length of the last names, something like MAX(LENGTH(LAST NAME)).

- Is it possible to count the records in each group, first by dividing the employee records by year of employment, then by job, and finally by salary?

Yes, it can be possible, you only need to specify the groups in the clause GROUP BY, starting by the first big group.

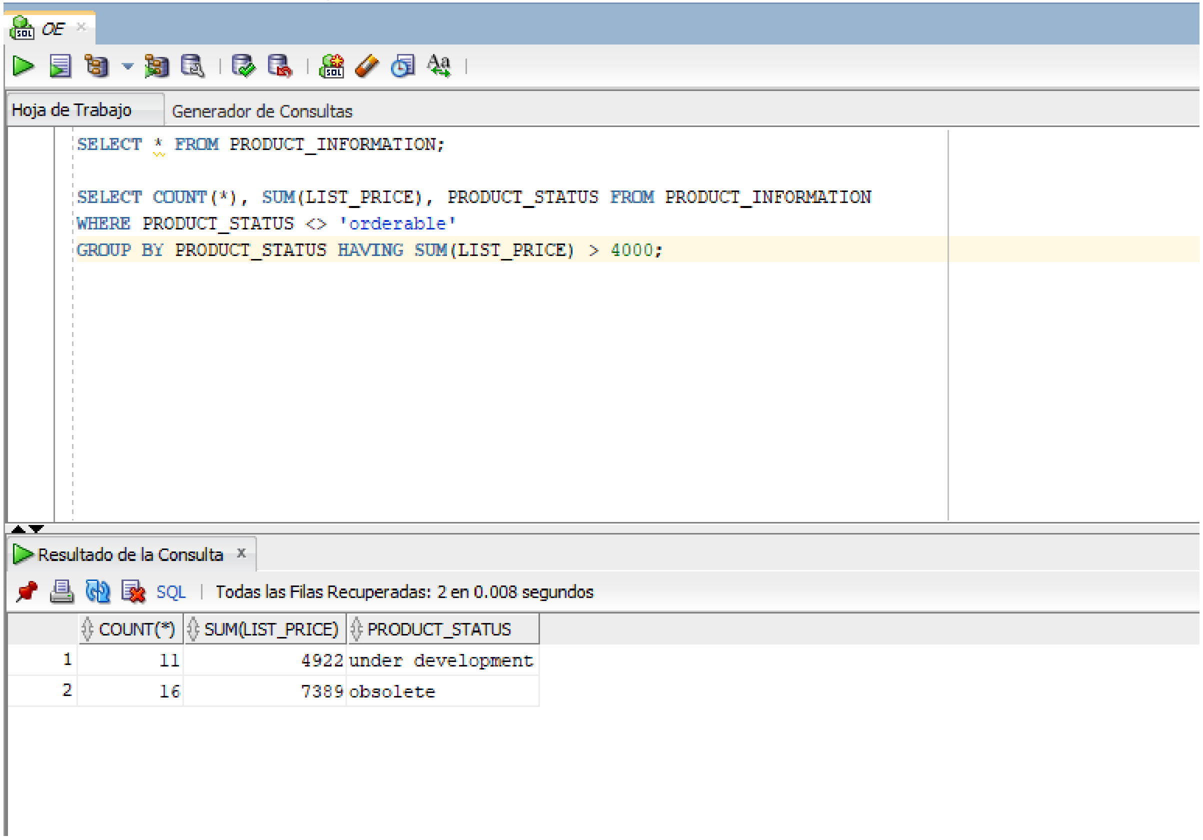
• Is there a limit to the number of groups within groups that can be formed?

No, but meanwhile we use more groups the results will be smaller, it can depends of the columns of the table and the distribution of the data.

Activity 3:

Connect to the OE schema and complete the following tasks. Using SQL Developer, connect to the OE schema and complete the following tasks. The PRODUCT INFORMATION table lists items that are orderable and others that are planned, obsolete, or under development. You are required to prepare a report that groups the nonorderable products by their PRODUCT STATUS and shows the number of products in each group and the sum of the LISTPRICE of the products per group. Further, only the grouplevel rows, where the sum of the LISTPRICE is greater than 4000, must be displayed. A product is nonorderable if the PRODUCTSTATUS value is not equal to the string ‘orderable’.

NOTE: Capture an image for each statement output.



```
SELECT * FROM PRODUCT_INFORMATION;

SELECT COUNT(*), SUM(LIST_PRICE), PRODUCT_STATUS FROM PRODUCT_INFORMATION
WHERE PRODUCT_STATUS <> 'orderable'
GROUP BY PRODUCT_STATUS HAVING SUM(LIST_PRICE) > 4000;
```

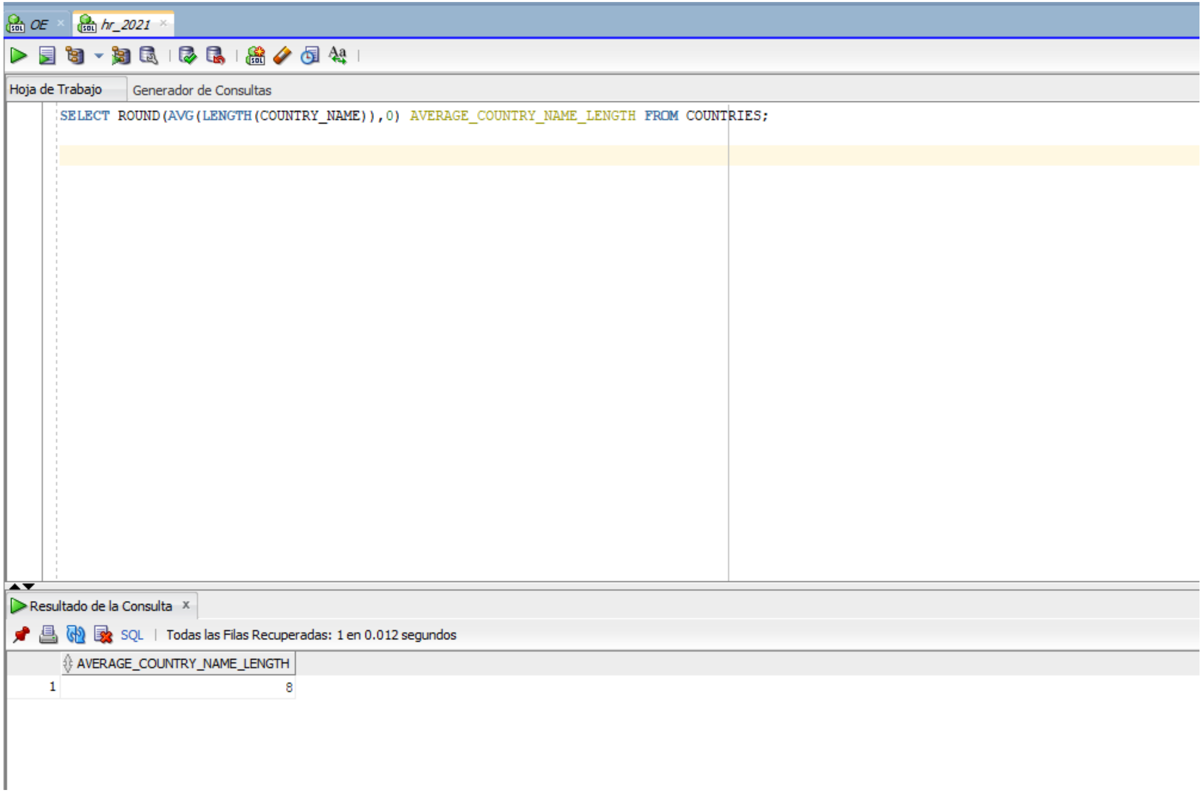
Todas las Filas Recuperadas: 2 en 0.008 segundos

	COUNT(*)	SUM(LIST_PRICE)	PRODUCT_STATUS
1	11	4922	under development
2	16	7389	obsolete

Activity 4:

This exercise must be performed using HR schema.

- The COUNTRIES table stores a list of COUNTRYNAME values. You are required to calculate the average length of all the country names. Any fractional components must be rounded to the nearest whole number.

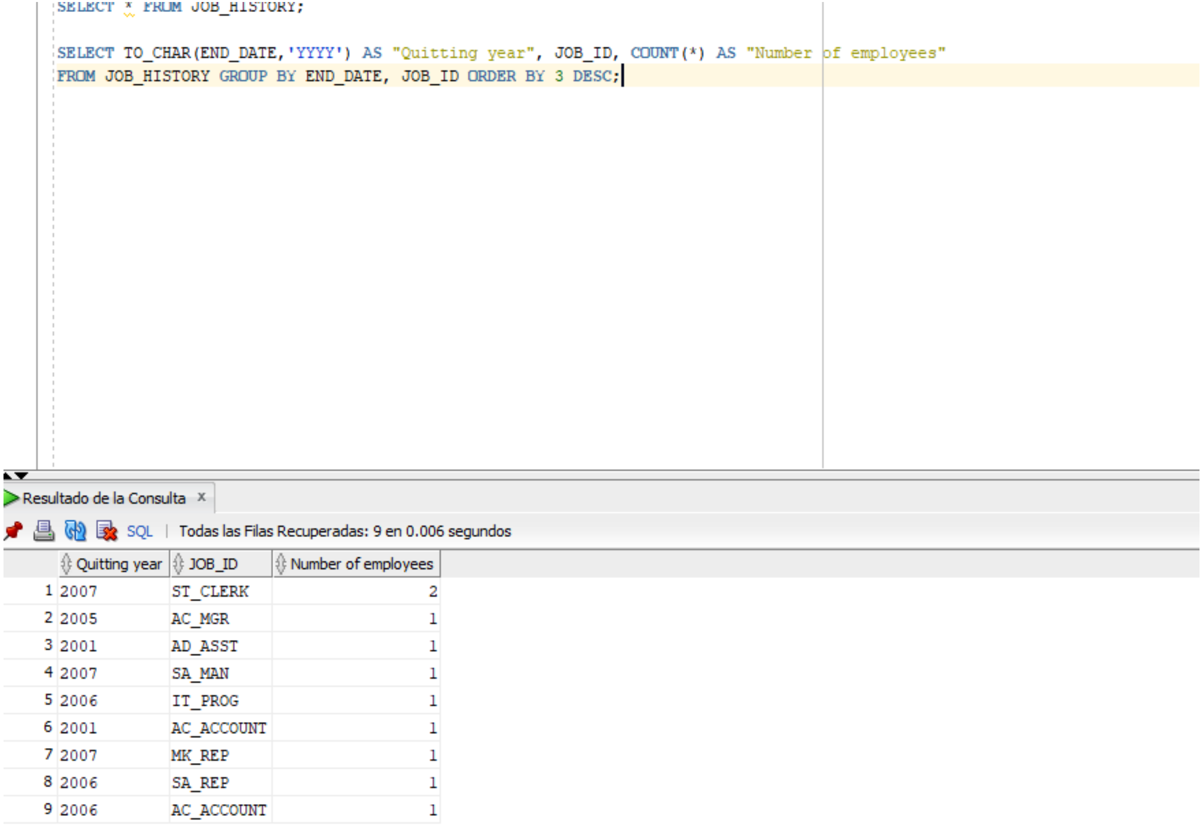


```
SELECT ROUND(AVG(LENGTH(COUNTRY_NAME)),0) AVERAGE_COUNTRY_NAME_LENGTH FROM COUNTRIES;
```

Todas las Filas Recuperadas: 1 en 0.012 segundos

	AVERAGE_COUNTRY_NAME_LENGTH
1	8

- **Analysis of staff turnover is a common reporting requirement. You are required to create a report containing the number of employees who left their jobs, grouped by the year in which they left. The jobs they performed are also required. The results must be sorted in descending order based on the number of employees in each group. The report must list the year, the JOBID, and the number of employees who left a particular job in that year.**



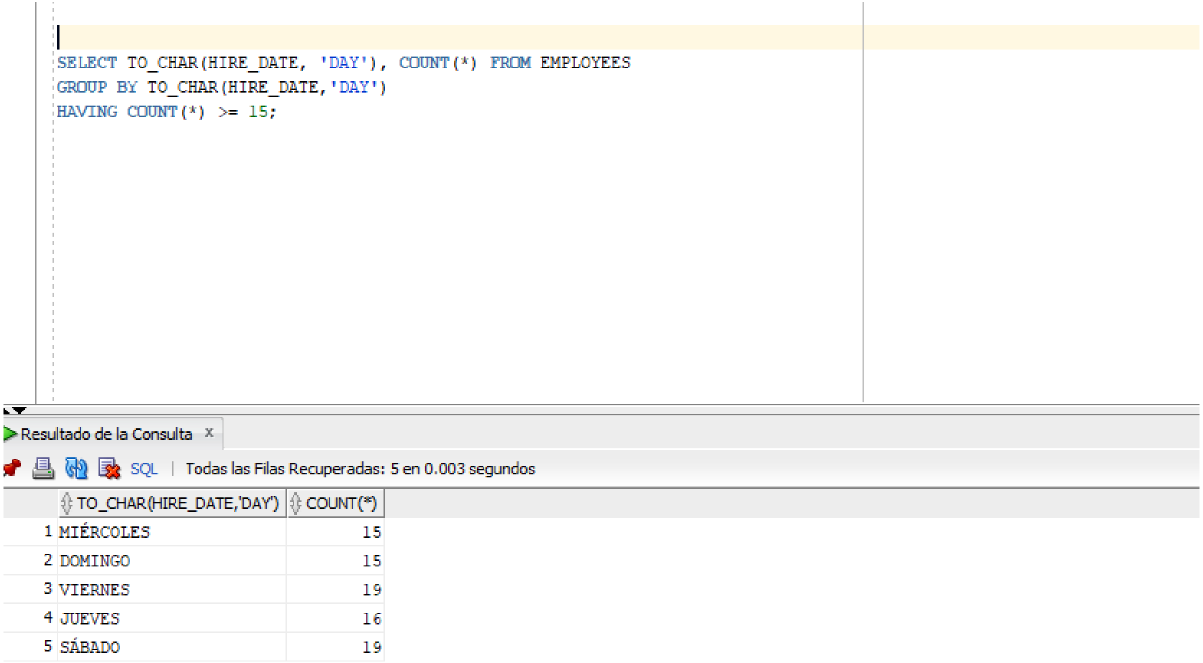
```
SELECT * FROM JOB_HISTORY;

SELECT TO_CHAR(END_DATE,'YYYY') AS "Quitting year", JOB_ID, COUNT(*) AS "Number of employees"
FROM JOB_HISTORY GROUP BY END_DATE, JOB_ID ORDER BY 3 DESC;
```

Todas las Filas Recuperadas: 9 en 0.006 segundos

	Quitting year	JOB_ID	Number of employees
1	2007	ST_CLERK	2
2	2005	AC_MGR	1
3	2001	AD_ASST	1
4	2007	SA_MAN	1
5	2006	IT_PROG	1
6	2001	AC_ACCOUNT	1
7	2007	MK_REP	1
8	2006	SA_REP	1
9	2006	AC_ACCOUNT	1

- The company is planning a recruitment drive and wants to identify the days of the week on which 15 or more staff members were hired. Your report must list the days and the number of employees hired on each of them



```
SELECT TO_CHAR(HIRE_DATE, 'DAY'), COUNT(*) FROM EMPLOYEES
GROUP BY TO_CHAR(HIRE_DATE,'DAY')
HAVING COUNT(*) >= 15;
```

Todas las Filas Recuperadas: 5 en 0.003 segundos

	TO_CHAR(HIRE_DATE,'DAY')	COUNT(*)
1	MIÉRCOLES	15
2	DOMINGO	15
3	VIERNES	19
4	JUEVES	16
5	SÁBADO	19

Activity 5:

Determine the validity of the following three statements. Circle either True or False and explain the reason. At the end of this practice, you should be familiar with using group functions and selecting groups of data.

Determine the validity of the following three statements. Circle either True or False and explain the reason.

1. Group functions work across many rows to produce one result per group. [✓ True]/False

Answer: Unlike single-row functions, group functions operate on sets of rows to give one result per group. These games can cover the whole table or the table divided into groups.

2. Group functions include nulls in calculations.

[✓ True]/False

Asnwer: I think that all group functions have some restrictions, but if you can include nulls

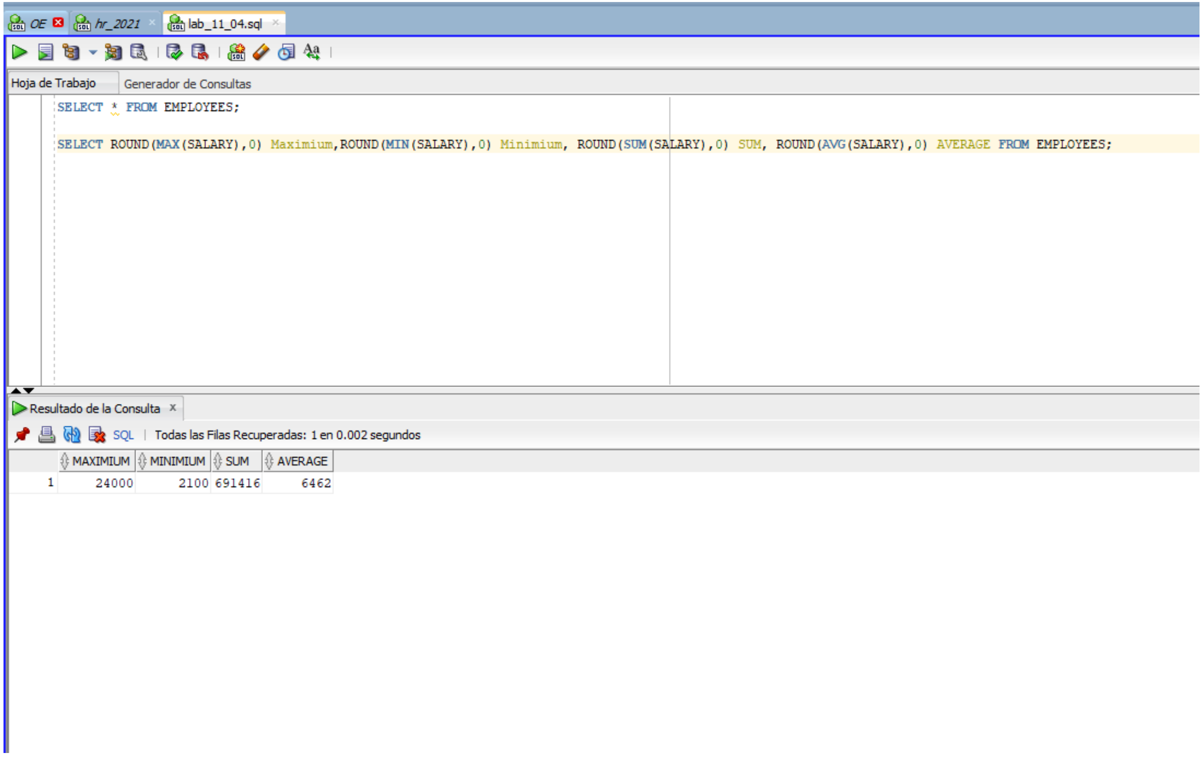
3. The WHERE clause restricts rows before inclusion in a group calculation.

[✓ True]/False

Answer: we can use HAVING to restrict the groups to show. HAVING is equivalent to WHERE, with the difference that HAVING is for restricting groups, whereas WHERE restricts rows of data.

The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Save your SQL statement as lab1104.sql. Run the query.



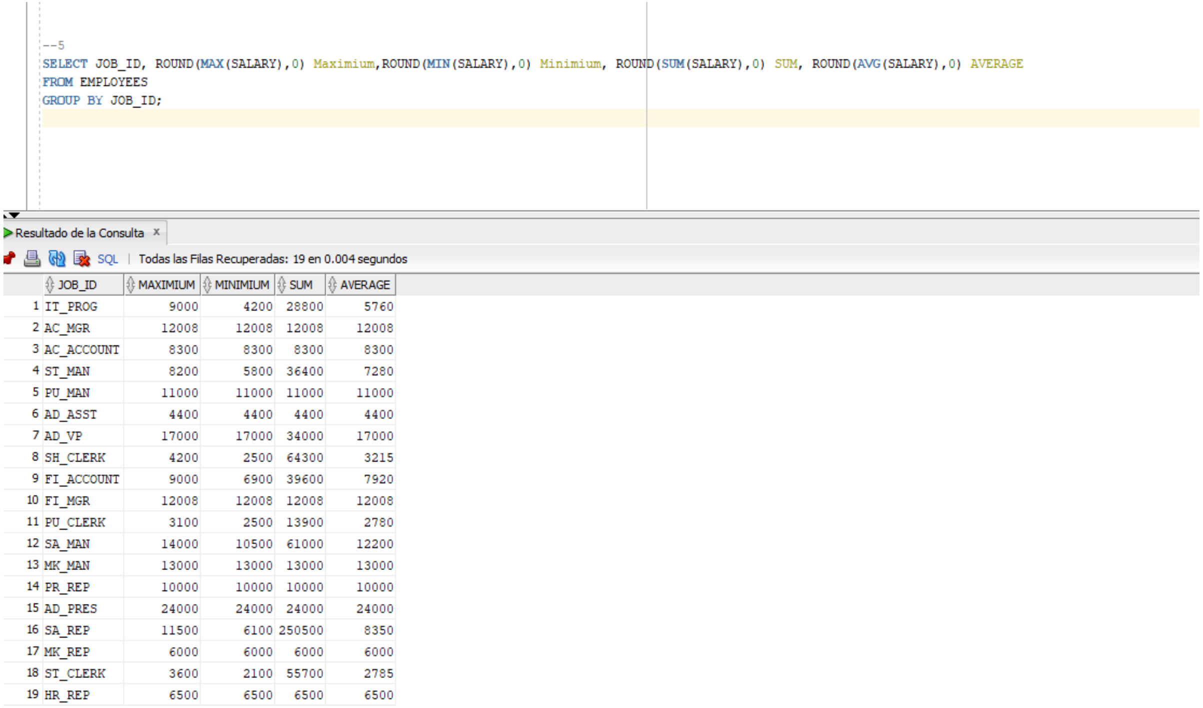
```
SELECT * FROM EMPLOYEES;

SELECT ROUND(MAX(SALARY),0) Maximium,ROUND(MIN(SALARY),0) Minimium, ROUND(SUM(SALARY),0) SUM, ROUND(AVG(SALARY),0) AVERAGE FROM EMPLOYEES;
```

SQL | Todas las Filas Recuperadas: 1 en 0.002 segundos

	MAXIMIUM	MINIMIUM	SUM	AVERAGE
1	24000	2100	691416	6462

5. Modify the query in lab1104.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab1104.sql as lab1105.sql. Run the statement in lab1105.sql.



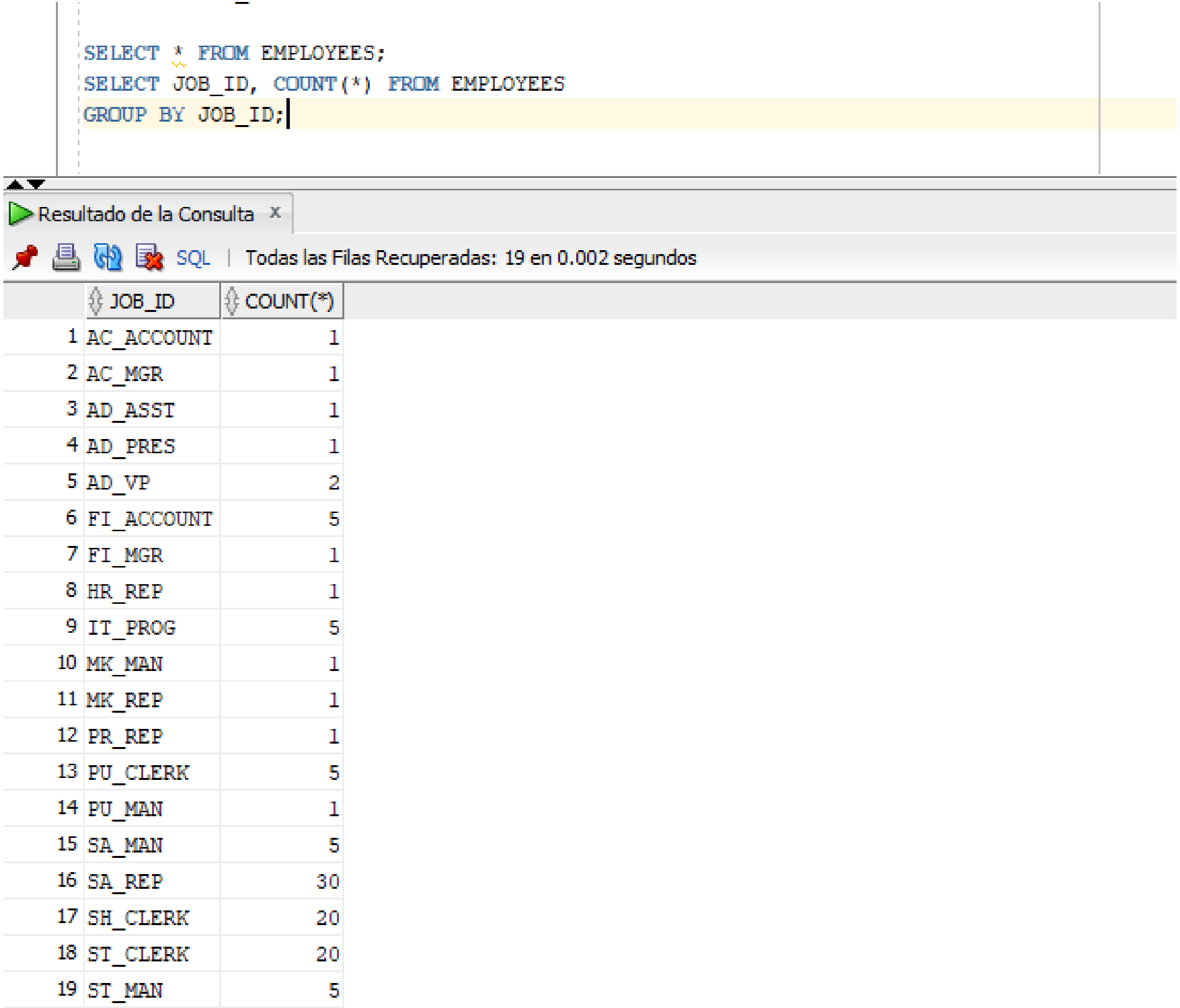
```
--5
SELECT JOB_ID, ROUND(MAX(SALARY),0) Maximium,ROUND(MIN(SALARY),0) Minimium, ROUND(SUM(SALARY),0) SUM, ROUND(AVG(SALARY),0) AVERAGE
FROM EMPLOYEES
GROUP BY JOB_ID;
```

Todas las Filas Recuperadas: 19 en 0.004 segundos

	JOB_ID	MAXIMIUM	MINIMIUM	SUM	AVERAGE
1	IT_PROG	9000	4200	28800	5760
2	AC_MGR	12008	12008	12008	12008
3	AC_ACCOUNT	8300	8300	8300	8300
4	ST_MAN	8200	5800	36400	7280
5	PU_MAN	11000	11000	11000	11000
6	AD_ASST	4400	4400	4400	4400
7	AD_VP	17000	17000	34000	17000
8	SH_CLERK	4200	2500	64300	3215
9	FI_ACCOUNT	9000	6900	39600	7920
10	FI_MGR	12008	12008	12008	12008
11	PU_CLERK	3100	2500	13900	2780
12	SA_MAN	14000	10500	61000	12200
13	MK_MAN	13000	13000	13000	13000
14	PR_REP	10000	10000	10000	10000
15	AD_PRES	24000	24000	24000	24000
16	SA_REP	11500	6100	250500	8350
17	MK_REP	6000	6000	6000	6000
18	ST_CLERK	3600	2100	55700	2785
19	HR_REP	6500	6500	6500	6500

6. Write a query to display the number of people with the same job

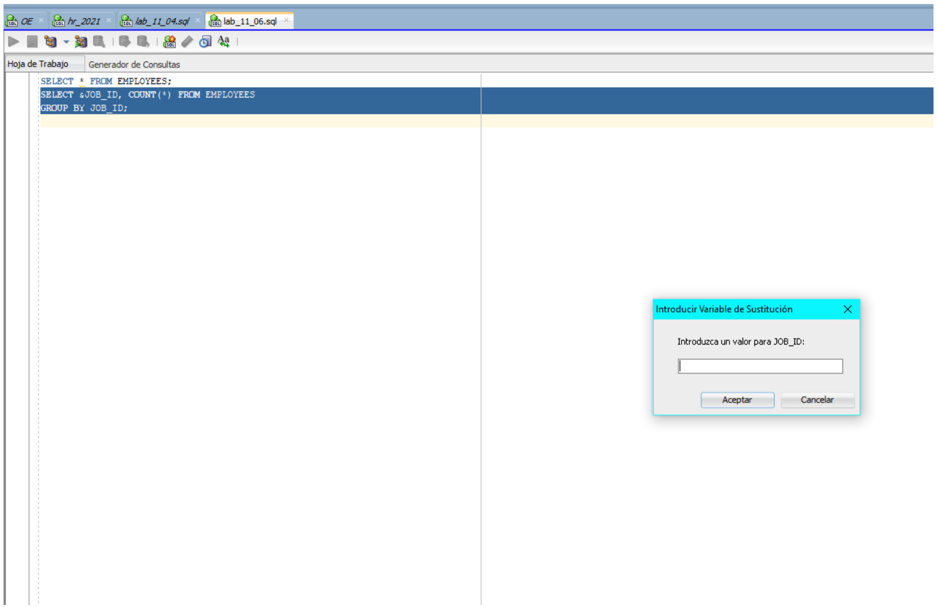
Generalize the query so that the user in the HR department is prompted for a job title. Save the script to a file named lab1106.sql. Run the query. Enter ITPROG when prompted.



```
SELECT * FROM EMPLOYEES;
SELECT JOB_ID, COUNT(*) FROM EMPLOYEES
GROUP BY JOB_ID;
```

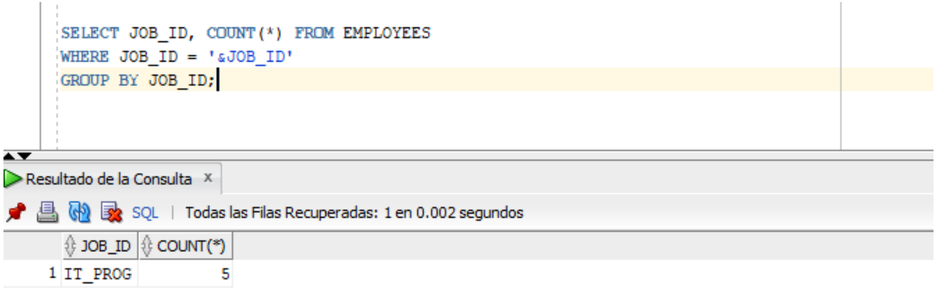
Todas las Filas Recuperadas: 19 en 0.002 segundos

	JOB_ID	COUNT(*)
1	AC_ACCOUNT	1
2	AC_MGR	1
3	AD_ASST	1
4	AD_PRES	1
5	AD_VP	2
6	FI_ACCOUNT	5
7	FI_MGR	1
8	HR_REP	1
9	IT_PROG	5
10	MK_MAN	1
11	MK_REP	1
12	PR_REP	1
13	PU_CLERK	5
14	PU_MAN	1
15	SA_MAN	5
16	SA_REP	30
17	SH_CLERK	20
18	ST_CLERK	20
19	ST_MAN	5



```
SELECT * FROM EMPLOYEES;
SELECT &JOB_ID, COUNT(*) FROM EMPLOYEES
GROUP BY JOB_ID;
```

Introduzca un valor para JOB_ID:

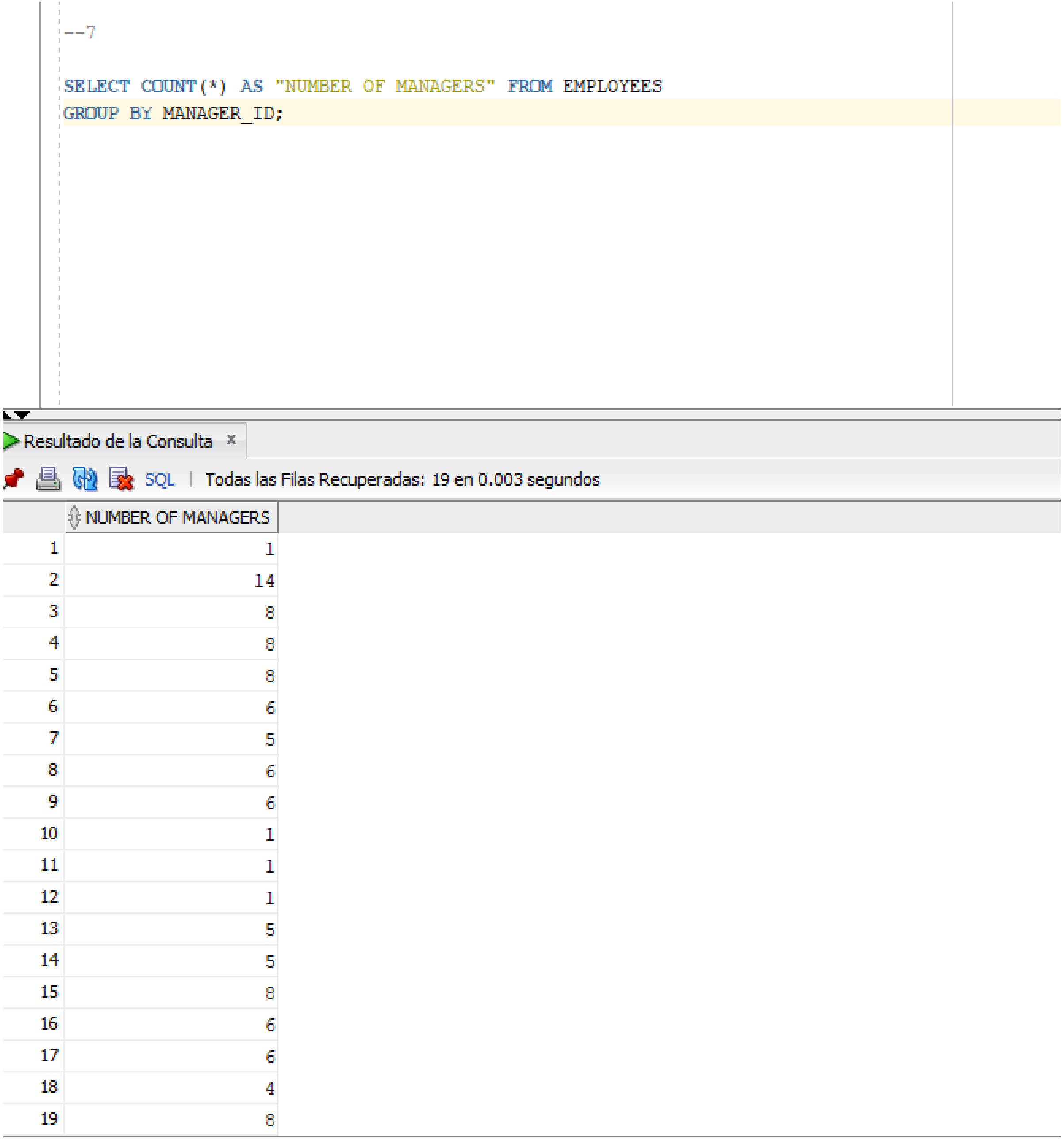


```
SELECT JOB_ID, COUNT(*) FROM EMPLOYEES
WHERE JOB_ID = '&JOB_ID'
GROUP BY JOB_ID;
```

Todas las Filas Recuperadas: 1 en 0.002 segundos

	JOB_ID	COUNT(*)
1	IT_PROG	5

7. Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGERID column to determine the number of managers.



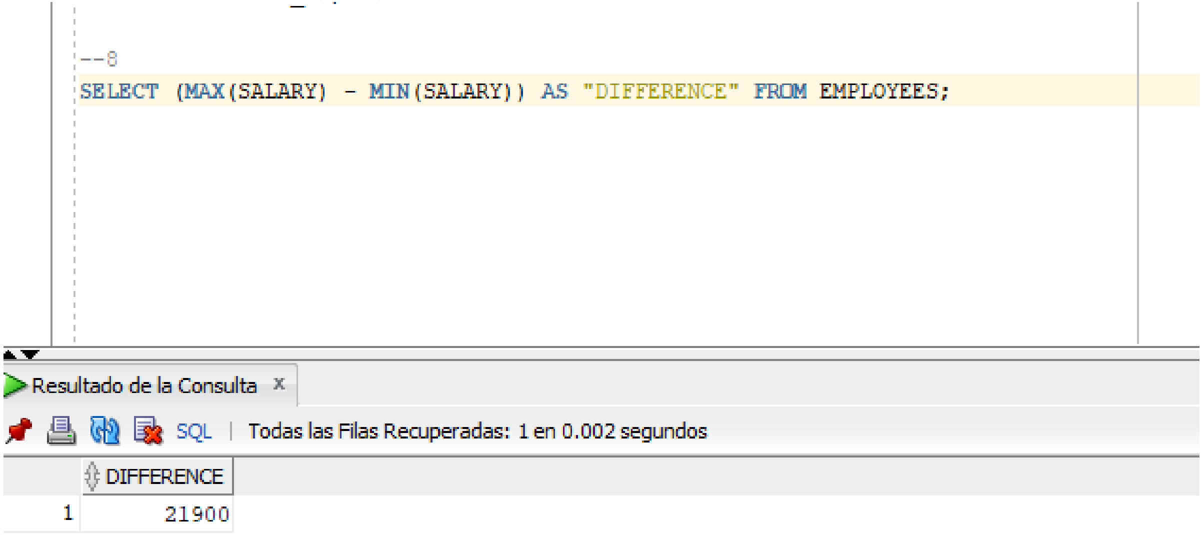
```
--7

SELECT COUNT(*) AS "NUMBER OF MANAGERS" FROM EMPLOYEES
GROUP BY MANAGER_ID;
```

Todas las Filas Recuperadas: 19 en 0.003 segundos

	NUMBER OF MANAGERS
1	1
2	14
3	8
4	8
5	8
6	6
7	5
8	6
9	6
10	1
11	1
12	1
13	5
14	5
15	8
16	6
17	6
18	4
19	8

8. **Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.**

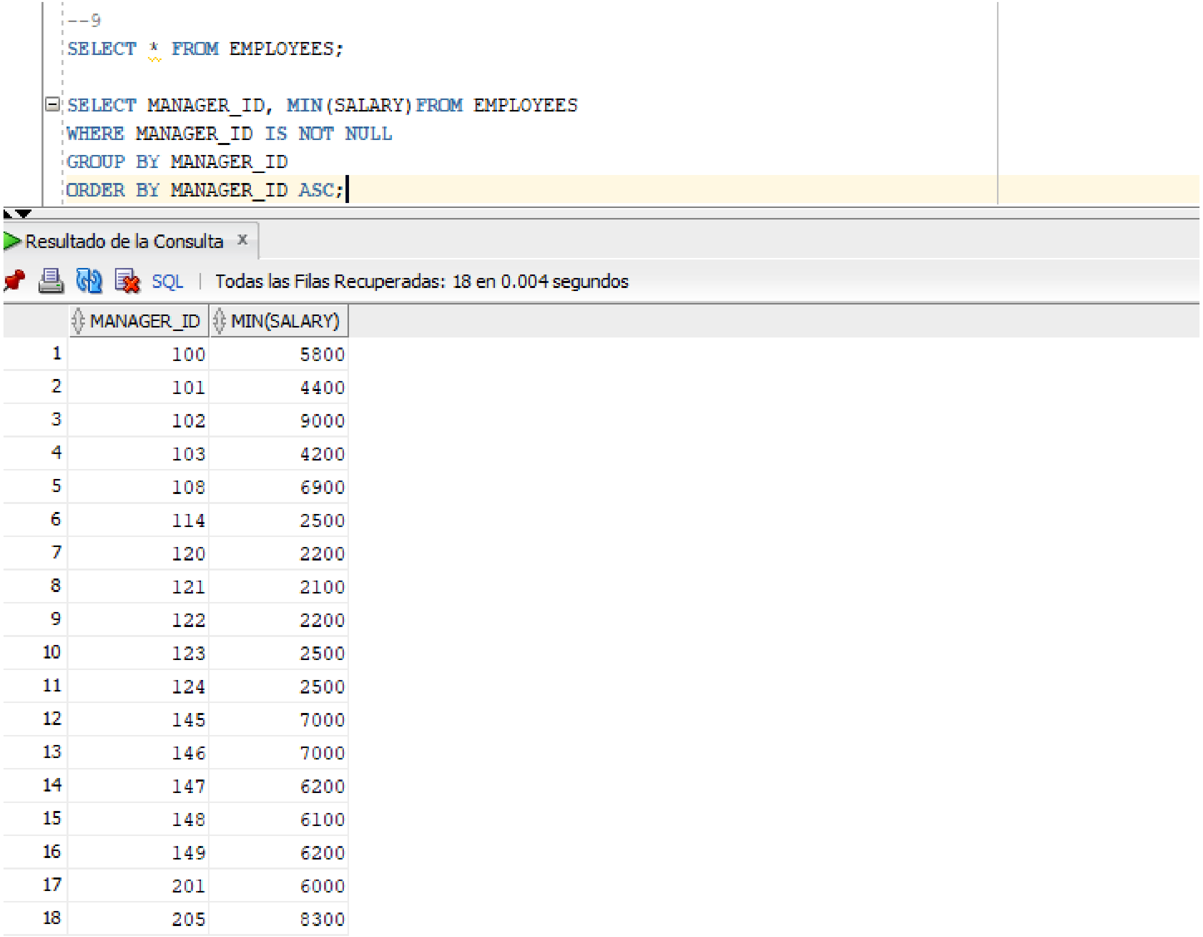


```
--8
SELECT (MAX(SALARY) - MIN(SALARY)) AS "DIFFERENCE" FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 1 en 0.002 segundos

```
     DIFFERENCE
1         21900
```

9. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is 6,000 or less. Sort the output in descending order of salary.



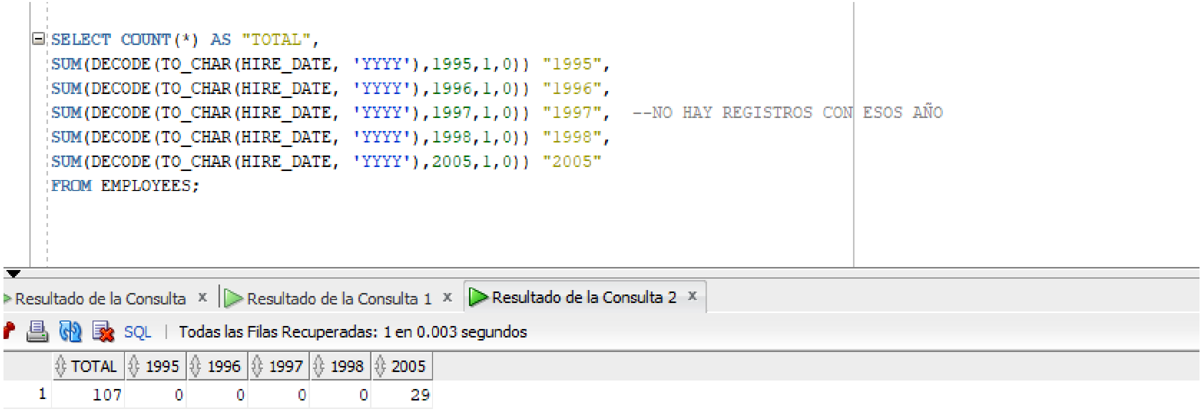
```
--9
SELECT * FROM EMPLOYEES;

SELECT MANAGER_ID, MIN(SALARY)FROM EMPLOYEES
WHERE MANAGER_ID IS NOT NULL
GROUP BY MANAGER_ID
ORDER BY MANAGER_ID ASC;
```

Todas las Filas Recuperadas: 18 en 0.004 segundos

	MANAGER_ID	MIN(SALARY)
1	100	5800
2	101	4400
3	102	9000
4	103	4200
5	108	6900
6	114	2500
7	120	2200
8	121	2100
9	122	2200
10	123	2500
11	124	2500
12	145	7000
13	146	7000
14	147	6200
15	148	6100
16	149	6200
17	201	6000
18	205	8300

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.



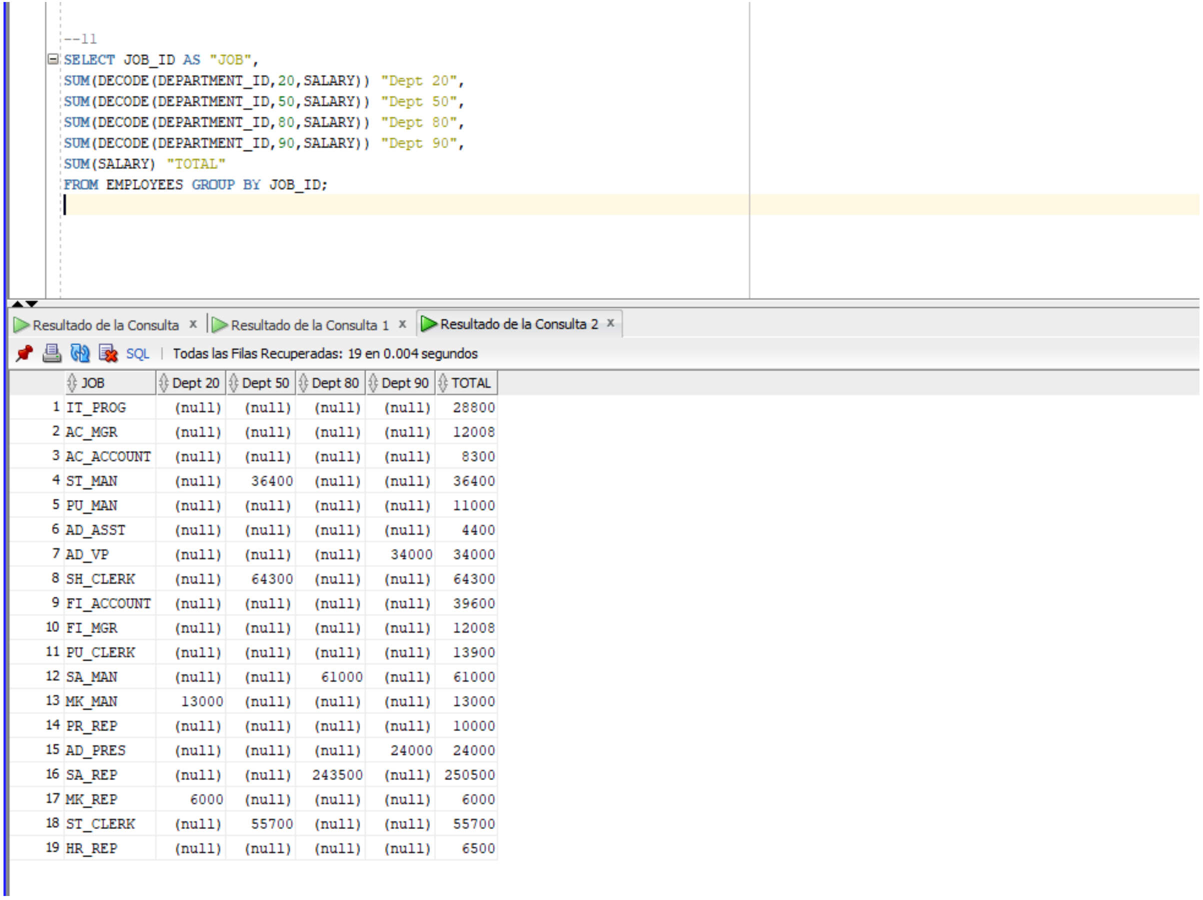
```
SELECT COUNT(*) AS "TOTAL",
SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1995,1,0)) "1995",
SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1996,1,0)) "1996",
SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1997,1,0)) "1997",  --NO HAY REGISTROS CON ESOS AÑO
SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),1998,1,0)) "1998",
SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),2005,1,0)) "2005"
FROM EMPLOYEES;
```

Resultado de la Consulta ×
Resultado de la Consulta 1 ×
Resultado de la Consulta 2 ×
SQL

Todas las Filas Recuperadas: 1 en 0.003 segundos

	TOTAL	1995	1996	1997	1998	2005
1	107	0	0	0	0	29

11. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.



```
--11
SELECT JOB_ID AS "JOB",
SUM(DECODE(DEPARTMENT_ID,20,SALARY)) "Dept 20",
SUM(DECODE(DEPARTMENT_ID,50,SALARY)) "Dept 50",
SUM(DECODE(DEPARTMENT_ID,80,SALARY)) "Dept 80",
SUM(DECODE(DEPARTMENT_ID,90,SALARY)) "Dept 90",
SUM(SALARY) "TOTAL"
FROM EMPLOYEES GROUP BY JOB_ID;
```

Resultado de la Consulta ×
Resultado de la Consulta 1 ×
Resultado de la Consulta 2 ×
SQL

Todas las Filas Recuperadas: 19 en 0.004 segundos

	JOB	Dept 20	Dept 50	Dept 80	Dept 90	TOTAL
1	IT_PROG	(null)	(null)	(null)	(null)	28800
2	AC_MGR	(null)	(null)	(null)	(null)	12008
3	AC_ACCOUNT	(null)	(null)	(null)	(null)	8300
4	ST_MAN	(null)	36400	(null)	(null)	36400
5	PU_MAN	(null)	(null)	(null)	(null)	11000
6	AD_ASST	(null)	(null)	(null)	(null)	4400
7	AD_VP	(null)	(null)	(null)	34000	34000
8	SH_CLERK	(null)	64300	(null)	(null)	64300
9	FI_ACCOUNT	(null)	(null)	(null)	(null)	39600
10	FI_MGR	(null)	(null)	(null)	(null)	12008
11	PU_CLERK	(null)	(null)	(null)	(null)	13900
12	SA_MAN	(null)	(null)	61000	(null)	61000
13	MK_MAN	13000	(null)	(null)	(null)	13000
14	PR_REP	(null)	(null)	(null)	(null)	10000
15	AD_PRES	(null)	(null)	(null)	24000	24000
16	SA_REP	(null)	(null)	243500	(null)	250500
17	MK_REP	6000	(null)	(null)	(null)	6000
18	ST_CLERK	(null)	55700	(null)	(null)	55700
19	HR_REP	(null)	(null)	(null)	(null)	6500

3 PRE-EVALUATION

Practices pre-Assessment for Database Systems Laboratory II Pre-Assessment PRACTICE 11 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 the content of the current chapter of the theory class to improve and learn more, i liked from this practice the use of GROUP BY clause because now i understand it better.