Autonomous University of Zacatecas

Academic Unit of Electrical Engineering

Software Engineering Academic Program

Group: 5B - Semester: 2022-5º

Practice Number: 11

Reporting Aggregated Data Using the Group Functions

Date: 25/October/2022

Professor:

Aldonso Becerra Sánchez.

Student:

Cristian Omar Alvarado Rodríguez.

Contents

Reporting Aggregated Data Using the Group Functions

October 25th, 2022

1 Introduction

The so-called group functions or group functions are functions that operate on multiple records of a SELECT statement. In this article I will talk about the most common SQL group functions (in a later article I will also talk about the GROUP BY and HAVING clauses, which are closely related to this type of function).

The first thing to know about group functions is that they perform calculations on a group of records, returning a single result, so they allow, for example, obtaining totals.

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

• All group functions ignore column null values, but with the use of the NVL/NVL2/COALESCE functions you can force group functions to include null values.

• Use the DISTINCT column keyword to suppress the count of duplicate values in a column.

• In GROUP BY Syntax is used to divide the rows of a table into groups, group_by_expression specifies columns whose values determine the basis for grouping rows.

- Use HAVING to restrict the groups to display. HAVING is equivalent to WHERE, except that HAVING is for restricting groups, while WHERE is for rows of data. group_condition is the constraint condition applied to groups.

Some Group Functions.

Each of the functions accepts one argument. Below we show the use of some of them:

- AVG: Returns the mean or average value of expr. Skip null values.

- COUNT: Returns the Number of rows, in which expr evaluates to something not null (if * is used, it counts all the rows of the table, including duplicates and those that contain null values).

- MAX: Returns the maximum value of expr; null values are ignored.

- MIN: Returns the minimum value of expr; null values are ignored.

- STDDEV: Returns the Standard Deviation of expr; null values are ignored.

- SUM: Returns the total sum of expr. null values are ignored.

- VARIANCE: Returns the variance or square of the standard deviation of expr. Skip null values.

2 Practice objective

Use SQL SELECT statements for retrieving data from database by means of different contexts using different Oracle functions.

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.

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.

Explanation: C, The DUAL table only has one row and one column, so the COUNT(*) function returns the number of rows in a table or group which in this case returns 1.

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.

Explanation: B, Group functions can operate on multiple rows at once, unlike single-row functions.

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.

Explanation: A, The SUM function ignores null values and adds non-null values. Since nine rows contain the value SALARY = 100, the function returns 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.

Explanation: C, The COUNT (*) function considers all the rows, that is, it counts how many rows there are in the employees table, including those with NULL values. COUNT(SALARY) returns 9 since it only counts the rows that do have a salary (it ignores NULL values).

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.

Explanation: C, The NVL function converts the NULL value to 100 and thereafter the average function adds the SALARY values and gets 1000 and dividing this by the number of records which is 10 gives 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.

Explanation: C, The data is segmented on the function of the SALARY column. This creates two groups: one with SALARY values of 100 and the other with a NULL SALARY value, the average length of the SALARY 100 value is 3 for the rows in the first group. The NULL salary value is first converted to the number 0 by the NVL function, and the average length of SALARY is 1. The SUM function operates on the two groups by adding the values 3 and 1 returning 4.

7. How many records are returned by the following query?
SELECT SUM(SALARY), DEPARTMENT_ID FROM EMPLOYEES GROUP BY DEPARTMENT_ID; Assume there are 11 no null and 1 null unique DEPARTMENT_ID values. All records have a no null SALARY value. (Choose the best answer.)

A) 12

B) 11

C) NULL

D) None of the above.

Explanation: A, There are 12 different DEPARTMENT_ID values, since this is the grouping attribute, 12 groups are created including 1 with a Null DEPARTMENT_ID value and therefore 12 rows are returned because there are twelve different DEPARTMENT_ID groups.

8. What values are returned after executing the following statement?
SELECT JOB_ID, MAX_SALARY FROM JOBS GROUP BY MAX_SALARY;
Assume that the JOBS table has ten records with the same JOB_ID value of DBA and the same MAX_SALARY 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.

Explanation: C, In order for the GROUP BY clause to be used, a group function must appear in the SELECT statement.

INCLUDE OR EXCLUDE GROUPED ROWS USING THE HAVING CLAUSE.

9. How many rows of data are returned after executing the following statement? SELECT DEPT_ID, SUM(NVL(SALARY,100)) FROM EMP GROUP BY DEPT_ID 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 DEPT_ID values of 10 and 20, respectively. (Choose the best answer.)

A) Two rows

B) One row

C) Zero rows

D) None of the above.

Explanation: B, Two groups are created based on their common DEPT_ID values. The group with DEPT_ID values of ten consists of five rows with SALARY values of 100 in each row. Therefore, the SUM(SALARY) function returns 500 for this group and satisfies the HAVING SUM(SALARY) ¿ 400 clause. The group with DEPT_ID values of 20 has four rows with SALARY values of 100 and one row with a NULL SALARY . SUM(SALARY) only returns 400 and this group does not satisfy the HAVING clause

10. How many rows of data are returned after executing the following statement? SELECT DEPT_ID, SUM(SALARY) FROM EMP GROUP BY DEPT_ID 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 DEPT_ID values of 10 and 20, respectively. (Choose the best answer.)

A) Two rows

B) One row

C) Zero rows

D) None of the above.

Explanation: A, Two groups are created based on their common DEPT_ID values. The group with DEPT_ID values of 10 consists of five rows with SALARY values of 100 in each of them. Therefore the SUM(NVL(SALARY,100)) function returns 500 for this group and it satisfies the HAVING SUM(SALARY) ¿ 400 clause. The group with DEPT_ID values of 20 has four rows with SALARY values of 100 and one row with a null SALARY. SUM(NVL(SALARY,100)) returns 500 and this group satisfies the HAVING clause. Therefore two rows are returned.

Activity 2: Propose an answer to the following issues:

a) 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?

R = Yes, this is possible thanks to the MIN function since this function also works with dates.

b) 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?

R = If possible, since with the help of the group functions it would be done in a simple way, the SUM, MAX and MIN functions would be used.

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

R = If it is possible since it could be used in the following way COUNT(DISTINCT JOB_ID).

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

R = Yes this is possible since the MAX and MIN group functions and the LENGTH function could be used since this last function helps us to obtain the length of a string and the MAX and MIN functions help us to obtain the longest string and the string (last name) with the shortest length respectively.

e) 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?

R = Yes, this is possible since the GROUP BY clause allows you to create subgroups of a group. And to perform the count, the COUNT group function would be used.

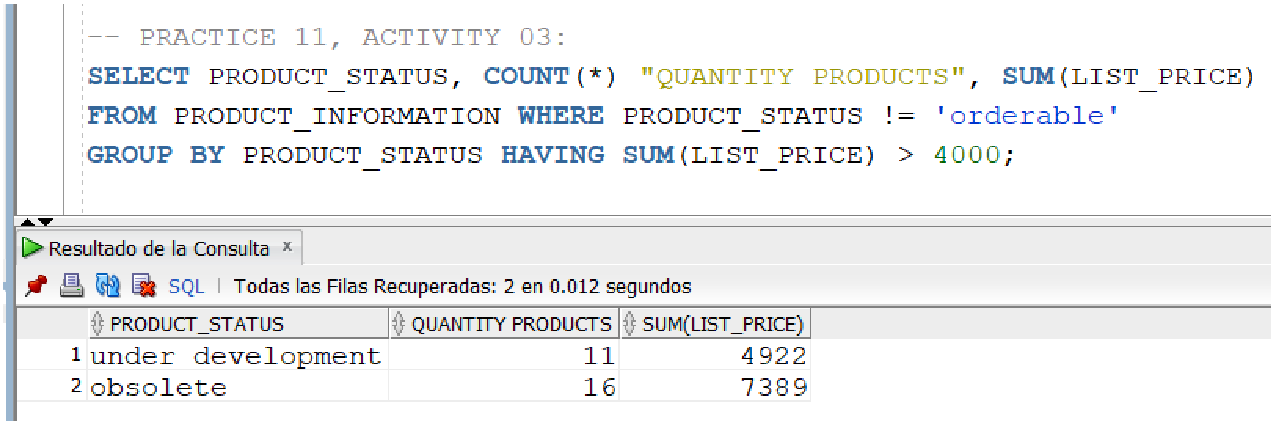
f) Is there a limit to the number of groups within groups that can be formed?

R = There is no limit as "N" groups or subgroups can be created.

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 non-orderable products by their PRODUCT_STATUS and shows the number of products in each group and the sum of the LIST_PRICE of the products per group. Further, only the group-level rows, where the sum of the LIST_PRICE is greater than 4000, must be displayed. A product is non-orderable if the PRODUCT_STATUS value is not equal to the string 'orderable', **see figure 1**.

Note: Capture an image for each statement output.



```
-- PRACTICE 11, ACTIVITY 03:
SELECT PRODUCT_STATUS, COUNT(*) "QUANTITY PRODUCTS", SUM(LIST_PRICE)
FROM PRODUCT_INFORMATION WHERE PRODUCT_STATUS != 'orderable'
GROUP BY PRODUCT_STATUS HAVING SUM(LIST_PRICE) > 4000;
```

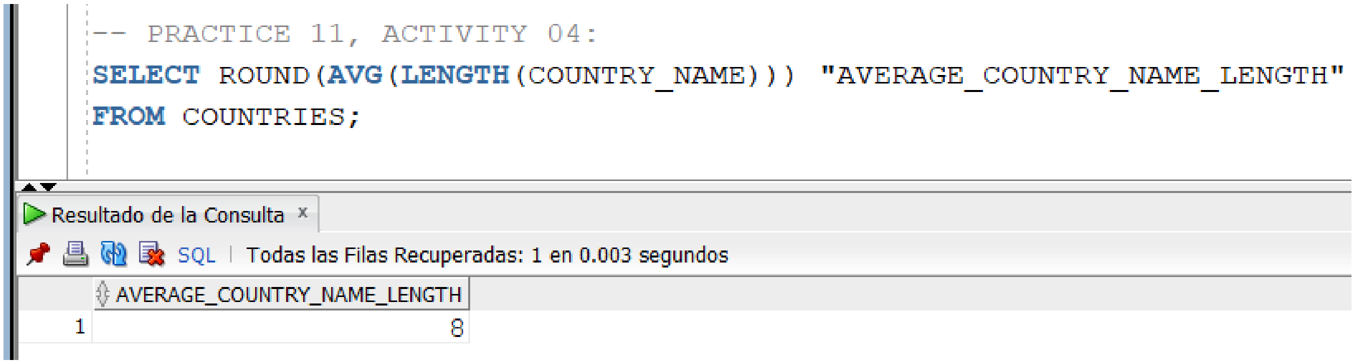
Todas las Filas Recuperadas: 2 en 0.012 segundos

	PRODUCT_STATUS	QUANTITY PRODUCTS	SUM(LIST_PRICE)
1	under development	11	4922
2	obsolete	16	7389

Figure 1: *Select statement and group functions.*

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

- The COUNTRIES table stores a list of COUNTRY_NAME 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., **see figure 2**.



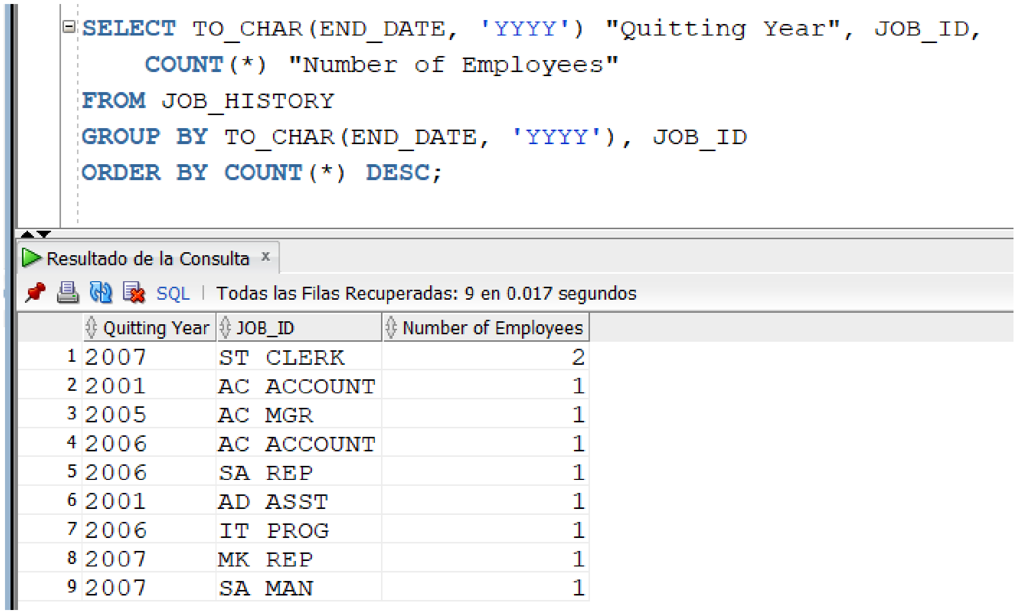
```
-- PRACTICE 11, ACTIVITY 04:
SELECT ROUND(AVG(LENGTH(COUNTRY_NAME))) "AVERAGE_COUNTRY_NAME_LENGTH"
FROM COUNTRIES;
```

Todas las Filas Recuperadas: 1 en 0.003 segundos

	AVERAGE_COUNTRY_NAME_LENGTH
1	8

Figure 2: *Select statement and group functions.*

- 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 JOB_ID, and the number of employees who left a particular job in that year, **see figure 3**.



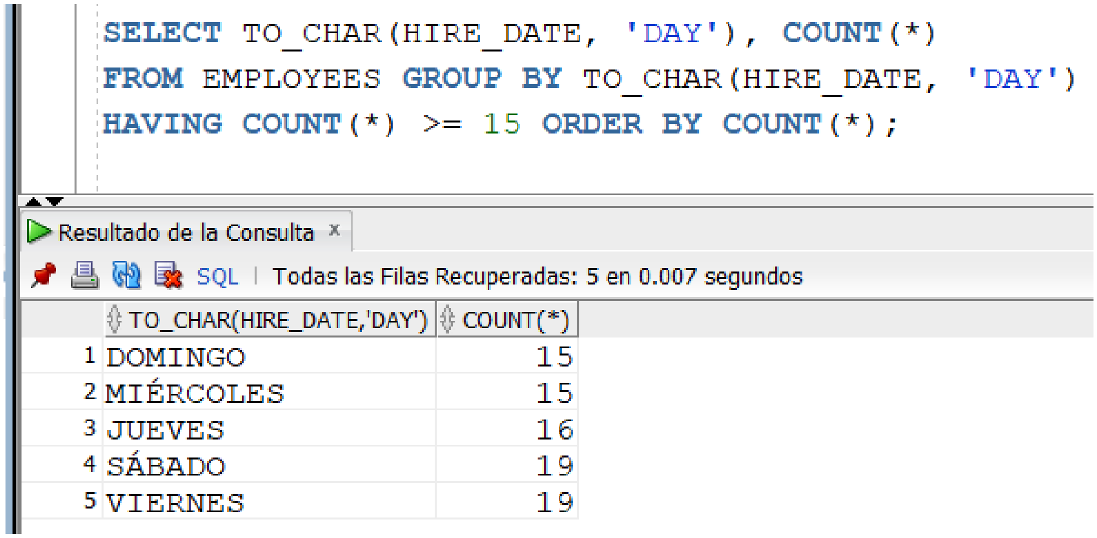
```
SELECT TO_CHAR(END_DATE, 'YYYY') "Quitting Year", JOB_ID,
    COUNT(*) "Number of Employees"
FROM JOB_HISTORY
GROUP BY TO_CHAR(END_DATE, 'YYYY'), JOB_ID
ORDER BY COUNT(*) DESC;
```

Todas las Filas Recuperadas: 9 en 0.017 segundos

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

Figure 3: *Select statement and group functions.*

- 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, **see figure 4**.



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

Todas las Filas Recuperadas: 5 en 0.007 segundos

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

Figure 4: *Select statement and group functions.*

Activity 5: 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

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

True group functions work on a set of data (groups) and for each group returns a result.

2. Group functions include nulls in calculations. True/False

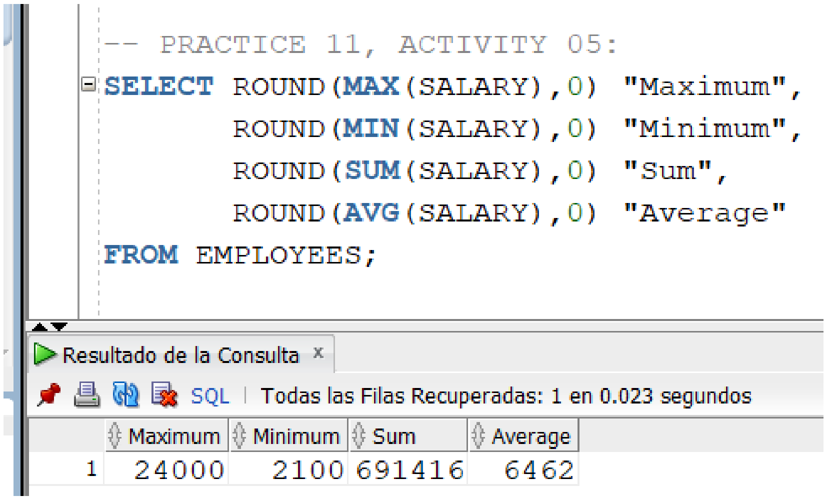
False, Group functions ignore null values.

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

True, the WHERE if clause can restrict the result of the groups with a condition.

The HR department needs the following reports:

4. CFind 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 lab_11_04.sql. Run the query., **see figure 5**.



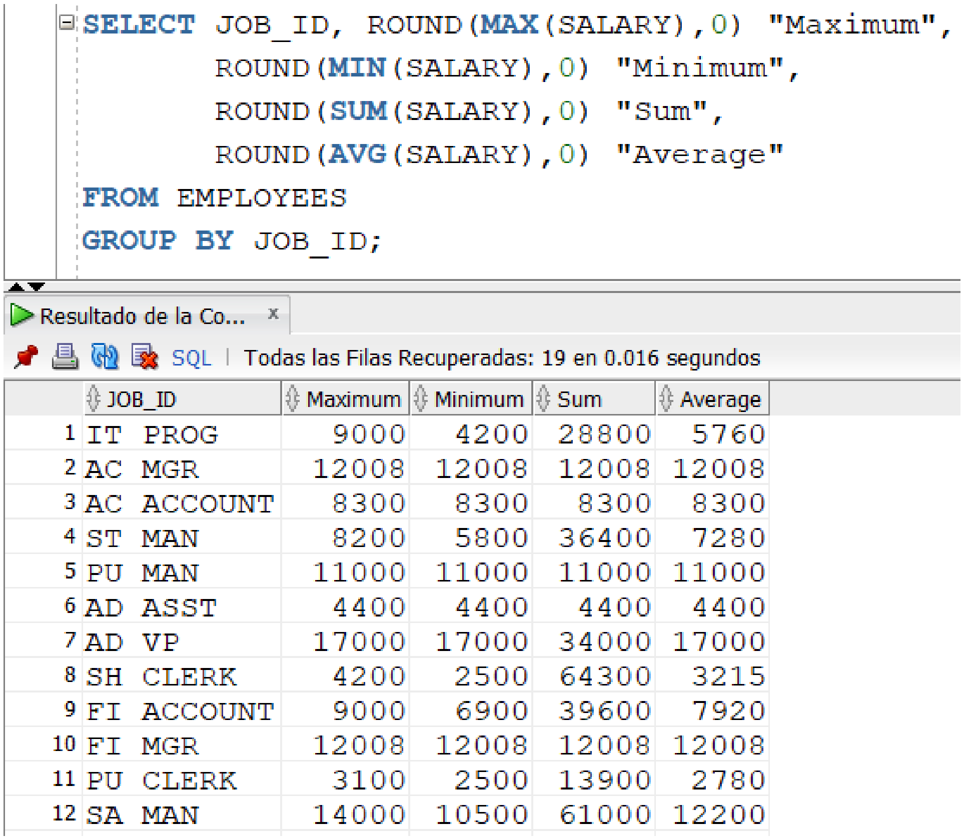
```
-- PRACTICE 11, ACTIVITY 05:
SELECT ROUND(MAX(SALARY),0) "Maximum",
       ROUND(MIN(SALARY),0) "Minimum",
       ROUND(SUM(SALARY),0) "Sum",
       ROUND(AVG(SALARY),0) "Average"
FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 1 en 0.023 segundos

```
    Maximum  Minimum  Sum     Average
1   24000       2100  691416     6462
```

Figure 5: *Select statement and group functions.*

5. Modify the query in lab_11_04.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab_11_04.sql as lab_11_05.sql. Run the statement in lab_11_05.sql., **see figure 6**.



```
SELECT JOB_ID, ROUND(MAX(SALARY),0) "Maximum",
       ROUND(MIN(SALARY),0) "Minimum",
       ROUND(SUM(SALARY),0) "Sum",
       ROUND(AVG(SALARY),0) "Average"
FROM EMPLOYEES
GROUP BY JOB_ID;
```

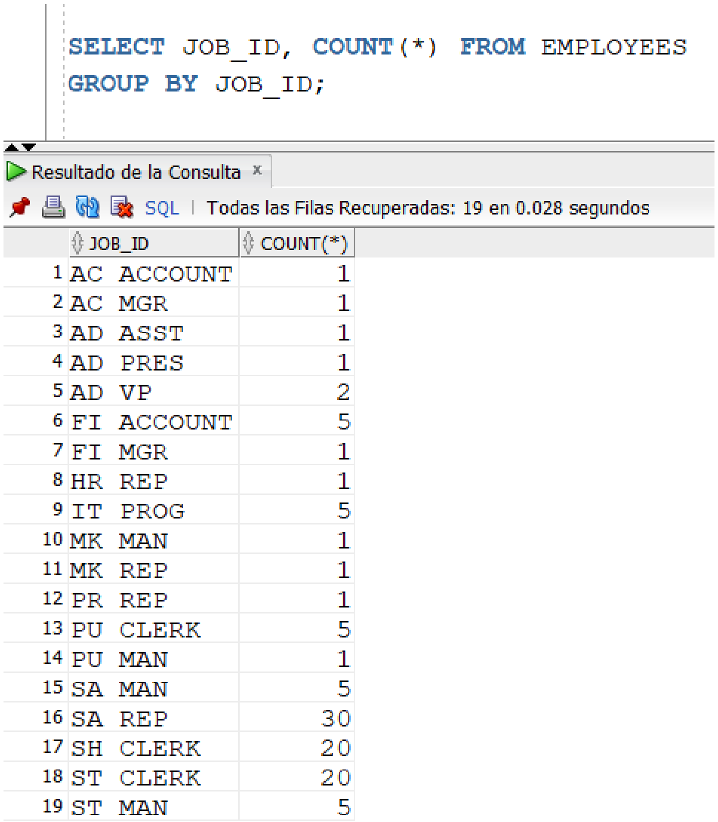
Todas las Filas Recuperadas: 19 en 0.016 segundos

	JOB_ID	Maximum	Minimum	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

Figure 6: *Select statement and group functions.*

6. Write a query to display the number of people with the same job, **see figure 7**.

Generalize the query so that the user in the HR department is prompted for a job title. Save the script to a file named lab_11_06.sql. Run the query. Enter IT_PROG when prompted, **see figure 8**.

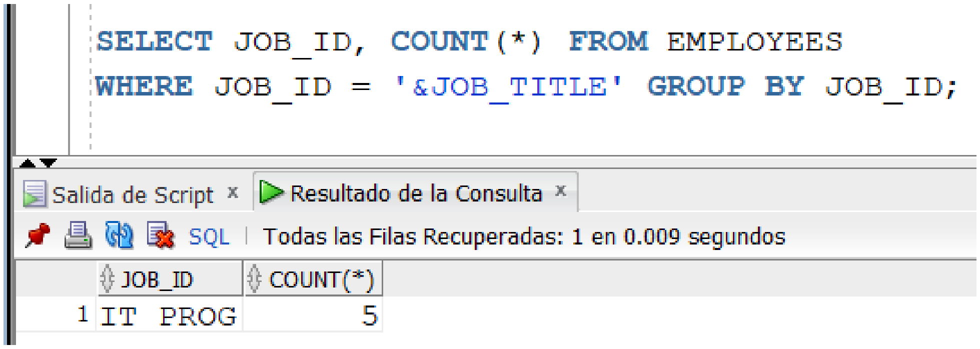


```
SELECT JOB_ID, COUNT(*) FROM EMPLOYEES
GROUP BY JOB_ID;
```

Todas las Filas Recuperadas: 19 en 0.028 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

Figure 7: *Select statement and group functions.*



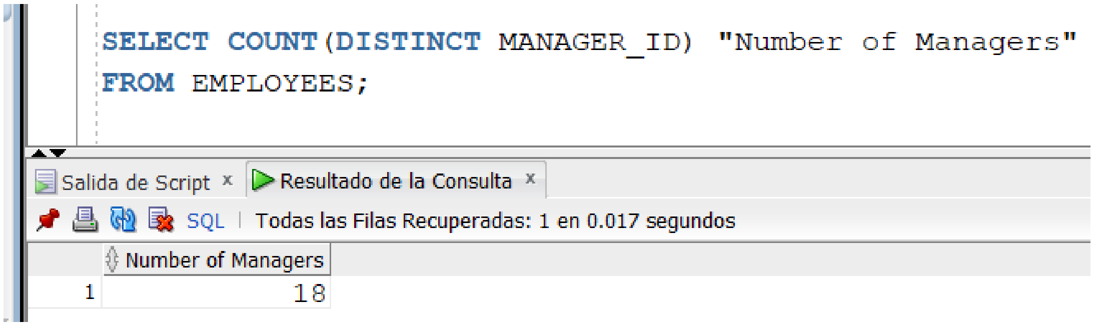
```
SELECT JOB_ID, COUNT(*) FROM EMPLOYEES
WHERE JOB_ID = '&JOB_TITLE' GROUP BY JOB_ID;
```

Todas las Filas Recuperadas: 1 en 0.009 segundos

	JOB_ID	COUNT(*)
1	IT PROG	5

Figure 8: *Select statement and group functions.*

7. Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers, **see figure 9**.



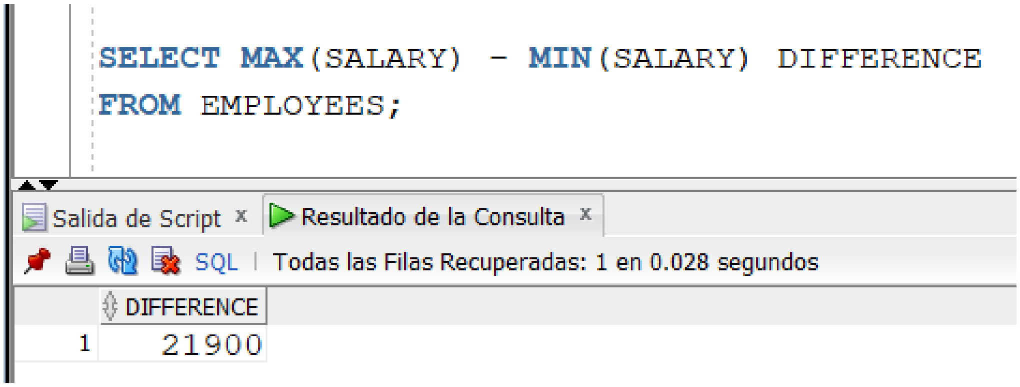
```
SELECT COUNT(DISTINCT MANAGER_ID) "Number of Managers"
FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 1 en 0.017 segundos

	Number of Managers
1	18

Figure 9: *Select statement and group functions.*

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE, **see figure 10**.



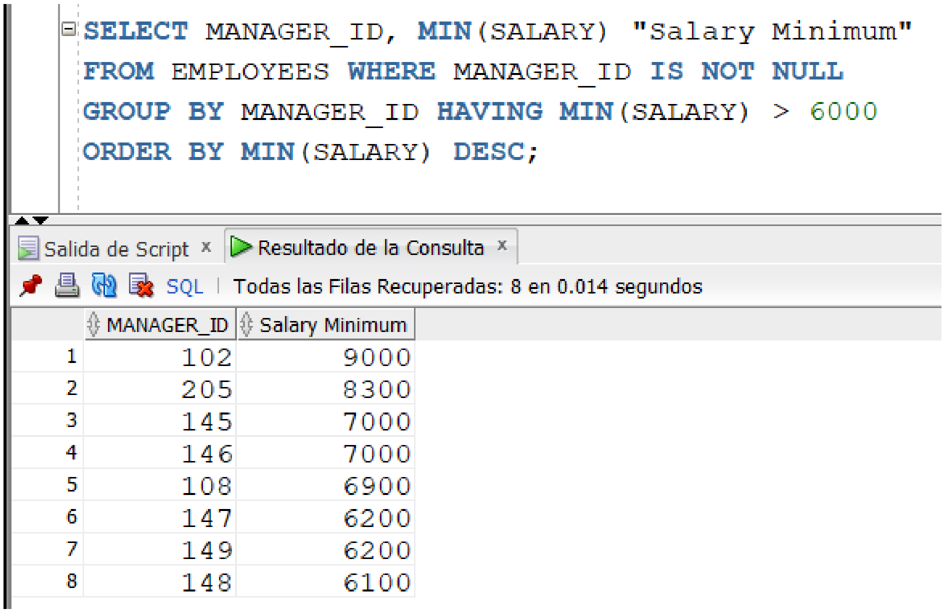
```
SELECT MAX(SALARY) - MIN(SALARY) DIFFERENCE
FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 1 en 0.028 segundos

	DIFFERENCE
1	21900

Figure 10: *Select statement and group functions.*

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, **see figure 11**.



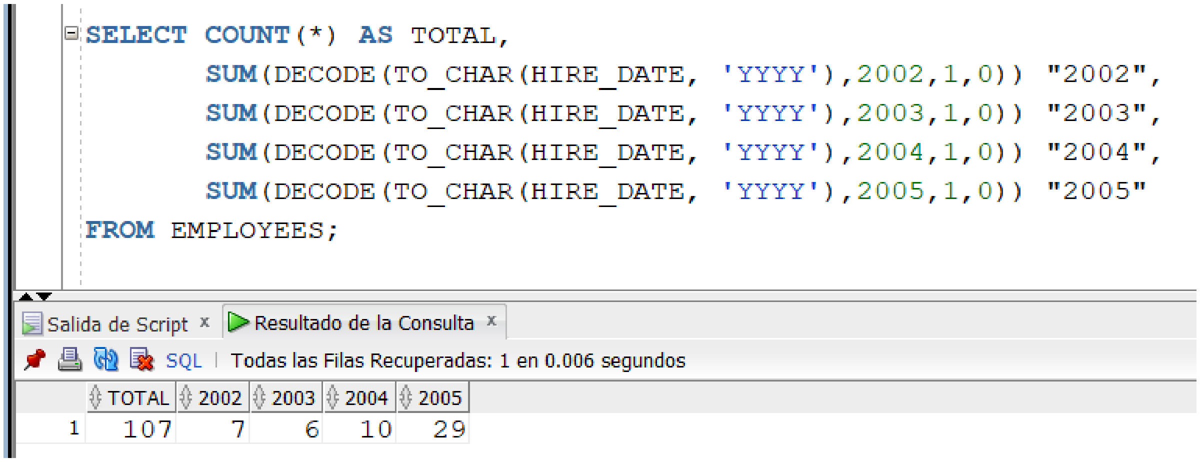
```
SELECT MANAGER_ID, MIN(SALARY) "Salary Minimum"
FROM EMPLOYEES WHERE MANAGER_ID IS NOT NULL
GROUP BY MANAGER_ID HAVING MIN(SALARY) > 6000
ORDER BY MIN(SALARY) DESC;
```

Todas las Filas Recuperadas: 8 en 0.014 segundos

	MANAGER_ID	Salary Minimum
1	102	9000
2	205	8300
3	145	7000
4	146	7000
5	108	6900
6	147	6200
7	149	6200
8	148	6100

Figure 11: *Select statement and group functions.*

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, **see figure 12**. Because the new versions of oracle do not have workers hired in the years 1995, 1996, 1997 and 1998, I decided to change the years to 2002, 2003, 2004 and 2005.



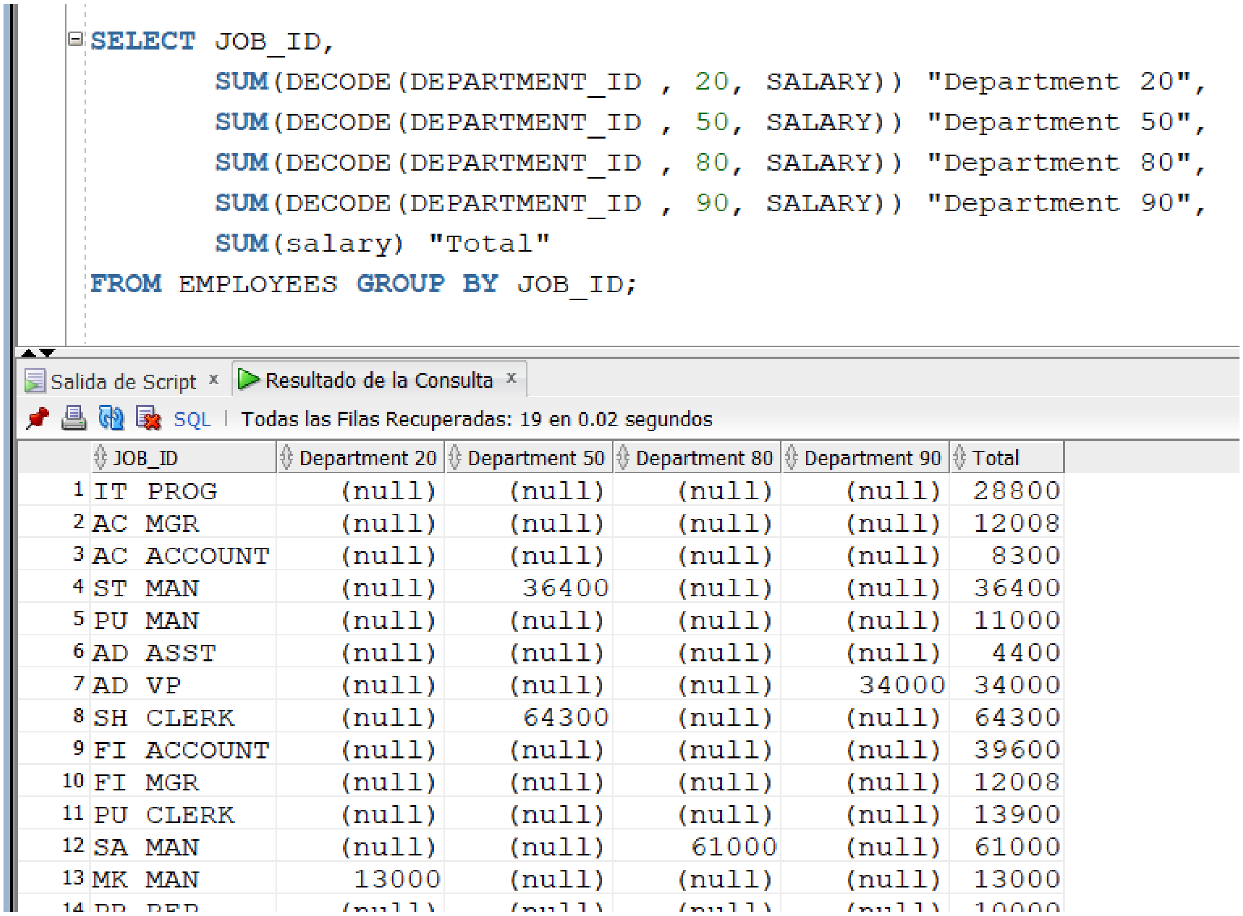
```
SELECT COUNT(*) AS TOTAL,
        SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),2002,1,0)) "2002",
        SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),2003,1,0)) "2003",
        SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),2004,1,0)) "2004",
        SUM(DECODE(TO_CHAR(HIRE_DATE, 'YYYY'),2005,1,0)) "2005"
FROM EMPLOYEES;
```

Todas las Filas Recuperadas: 1 en 0.006 segundos

	TOTAL	2002	2003	2004	2005
1	107	7	6	10	29

Figure 12: *Select statement and group functions.*

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, **see figure 13**.



```
SELECT JOB_ID,
        SUM(DECODE(DEPARTMENT_ID , 20, SALARY)) "Department 20",
        SUM(DECODE(DEPARTMENT_ID , 50, SALARY)) "Department 50",
        SUM(DECODE(DEPARTMENT_ID , 80, SALARY)) "Department 80",
        SUM(DECODE(DEPARTMENT_ID , 90, SALARY)) "Department 90",
        SUM(salary) "Total"
FROM EMPLOYEES GROUP BY JOB_ID;
```

Todas las Filas Recuperadas: 19 en 0.02 segundos

	JOB_ID	Department 20	Department 50	Department 80	Department 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

Figure 13: *Select statement and group functions.*

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 projection of the data within a relational database is stored in the table in the form of rows and columns. Projections are the first items identified during query execution. They are the selected columns within a table for which a query has been designed. Projections are mentioned in the first part of the SQL query, that is, the SELECT statement. After identifying the projections within the query frame, the next step would be to identify the rows that are relevant to the query. Filters are mentioned within the WHERE clause of the query and will identify the rows to be included in the results, the latter is called a selection.

This practice number 11 helped me practice the uses of the SELECT statement for data retrieval and projection. Finally, something important to mention is that the SQL language allows the projection and selection of data to meet the reporting needs that a programmer, developer or end user may need.