



ESCUELA SUPERIOR DE CÓMPUTO
INSTITUTO POLITÉCNICO NACIONAL



CABALLERO HUESCA CARLOS EDUARDO

Reporte de Practica



Practica 9

Columns and Scalar Functions
BASES DE DATOS
2CM5

Hernández Rubio Erika

I. INTRODUCCIÓN

Antecedentes Teóricos:

En el lenguaje SQL estándar existen básicamente **5 tipos de funciones**: **aritméticas**, de **cadenas** de caracteres, de **fechas**, de **conversión**, y **otras funciones** diversas.

Funciones aritméticas

- ▶ **ABS(n)**: Devuelve el valor absoluto de "n".
- ▶ **ROUND(m, n)**: Redondea el número "m" con el número de decimales indicado en "n", si no se indica "n" asume cero decimales.
- ▶ **SQRT(n)**: Devuelve la raíz cuadrada del parámetro que se le pase.
- ▶ **POWER(m, n)**: Devuelve la potencia de "m" elevada el exponente "n".

Funciones de agregación

Las funciones de agregación en SQL nos permiten efectuar operaciones sobre un conjunto de resultados, pero devolviendo un único valor agregado para todos ellos. Es decir, nos permiten obtener medias, máximos, etc... sobre un conjunto de valores.

Las funciones de agregación básicas que soportan todos los gestores de datos son las siguientes:

- ▶ **COUNT**: devuelve el número total de filas seleccionadas por la consulta.
- ▶ **MIN**: devuelve el valor mínimo del campo que especifiquemos.
- ▶ **MAX**: devuelve el valor máximo del campo que especifiquemos.
- ▶ **SUM**: suma los valores del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.
- ▶ **AVG**: devuelve el valor promedio del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.

El utilizar la cláusula **GROUP BY** no garantiza que los datos se devuelvan ordenados. Suele ser una práctica recomendable incluir una cláusula **ORDER BY** por las mismas columnas que utilicemos en **GROUP BY**, especificando el orden que nos interese. Por ejemplo, en el caso anterior

Existe una cláusula especial, parecida a la **WHERE** que ya conocemos que nos permite especificar las condiciones de filtro para los diferentes grupos de filas que devuelven estas consultas agregadas. Esta cláusula es **HAVING**.

HAVING es muy similar a la cláusula **WHERE**, pero en vez de afectar a las filas de la tabla, afecta a los grupos obtenidos.

Problem 1

For all departments, display department number and the sum of all salaries for that department. Name the derived column SUM_SALARY.

```
mysql> SELECT WORKDEPT, SUM(SALARY) AS SUM_SALARY FROM EMPLOYEE GROUP BY WORKDEPT;
```

WORKDEPT	SUM_SALARY
A00	354250
B01	94250
C01	308890
D11	646620
D21	358680
E01	80175
E11	317140
E21	282520

```
8 rows in set (0.04 sec)
```

Problem 2

For all departments display the department number and the number of employees. Name the derived column EMP_COUNT.

```
mysql> SELECT WORKDEPT, COUNT(*) AS EMP_COUNT FROM EMPLOYEE GROUP BY WORKDEPT;
```

WORKDEPT	EMP_COUNT
A00	5
B01	1
C01	4
D11	11
D21	7
E01	1
E11	7
E21	6

```
8 rows in set (0.00 sec)
```

Problem 3

Display those departments which have more than 3 employees, in ascending order.

```
mysql> SELECT WORKDEPT, COUNT(*) AS EMP_COUNT FROM EMPLOYEE GROUP BY WORKDEPT  
-> HAVING COUNT(*) > 3 ORDER BY count(*) asc;
```

WORKDEPT	EMP_COUNT
C01	4
A00	5
E21	6
D21	7
E11	7
D11	11

```
6 rows in set (0.00 sec)  
  
mysql> _
```

Problem 4

For all departments with at least one designer, display the number of designers and the department number. Name the derived column DESIGNER.

```
mysql> SELECT WORKDEPT, COUNT(*) AS DESIGNER FROM EMPLOYEE WHERE JOB LIKE 'DESIGNER';
+-----+-----+
| WORKDEPT | DESIGNER |
+-----+-----+
| D11      | 10       |
+-----+-----+
1 row in set (0.00 sec)
```

Problem 5

Show the average salary for men and the average salary for women for each department. Display the work department, the sex, the average salary, average bonus, average commission, and the number of people in each group. Include only those groups that have two or more people. Show only two decimal places in the averages.

```
mysql> SELECT WORKDEPT,SEX,ROUND(AVG(SALARY),2)AS AVG_SALARY,ROUND(AVG(BONUS),2) AS AVG_BONUS,
-> ROUND(AVG(COMM),2) AS AVG_COMM,COUNT(SEX) AS COUNT_
-> FROM EMPLOYEE GROUP BY WORKDEPT HAVING COUNT(*) >= 2;
+-----+-----+-----+-----+-----+-----+
| WORKDEPT | SEX | AVG_SALARY | AVG_BONUS | AVG_COMM | COUNT_ |
+-----+-----+-----+-----+-----+-----+
| A00      | F   | 70850.00   | 820.00    | 3368.00   | 5       |
| C01      | F   | 77222.50   | 625.00    | 2378.00   | 4       |
| D11      | M   | 58783.64   | 500.00    | 2011.45   | 11      |
| D21      | F   | 51240.00   | 500.00    | 2053.29   | 7       |
| E11      | F   | 45305.71   | 414.29    | 1681.57   | 7       |
| E21      | M   | 47086.67   | 483.33    | 1927.00   | 6       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Problem 6

Display the average bonus and average commission for all departments with an average bonus greater than \$500 and an average commission greater than \$2,000. Display all averages with two digits to the right of the decimal point.

```
mysql> SELECT WORKDEPT,ROUND(AVG(BONUS),2) AS AVG_BONUS, ROUND(AVG(COMM),2) AS AVG_COMM
-> FROM EMPLOYEE GROUP BY WORKDEPT;
+-----+-----+-----+
| WORKDEPT | AVG_BONUS | AVG_COMM |
+-----+-----+-----+
| A00      | 820.00    | 3368.00   |
| B01      | 800.00    | 3300.00   |
| C01      | 625.00    | 2378.00   |
| D11      | 500.00    | 2011.45   |
| D21      | 500.00    | 2053.29   |
| E01      | 800.00    | 3214.00   |
| E11      | 414.29    | 1681.57   |
| E21      | 483.33    | 1927.00   |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> _
```

CONCLUSIONES

- I. La cláusula GROUP BY unida a un SELECT permite agrupar filas según las columnas que se indiquen como parámetros, y se suele utilizar en conjunto con las funciones de agrupación, para obtener datos resumidos y agrupados por las columnas que se necesiten.
- II. Cuando utilizamos la cláusula GROUP BY, los únicos campos que podemos incluir en el SELECT sin que estén dentro de una función de agregación, son los que vayan especificados en el GROUP BY.
- III. La cláusula GROUP BY se puede utilizar con más de un campo al mismo tiempo. Si indicamos más de un campo como parámetro nos devolverá la información agrupada por los registros que tengan el mismo valor en los campos indicados.