

## **Funciones De Agrupación**

**AVG** Devuelve el valor medio

**BIT AND** Devuelve la operación de bits AND para todos los bits de una expresión

**BIT OR** Devuelve la operación de bits OR para todos los bits de una expresión

**BIT XOR** Devuelve la operación de bits XOR para todos los bits de una expresión

**COUNT** Devuelve el número de valores distintos de NULL en las filas recuperadas por una sentencia SELECT

**COUNT DISTINCT** Devuelve el número de valores diferentes, distintos de NULL

**GROUP CONCAT** Devuelve una cadena con la concatenación de los valores de un grupo

**MIN / MAX** Devuelve el valor mínimo O máximo de una expresión

**STD / STDDEV** Devuelve la desviación estándar de una expresión

**SUM** Devuelve la suma de una expresión

**VARIANCE** Devuelve la varianza estándar de una expresión

# Funciones De grupos

## COUNT

### COUNT(expr)

Devuelve un contador con el número de valores distintos de NULL en las filas recuperadas por una sentencia **SELECT**:

```
SELECT COUNT(*) AS Empleados, COUNT(EmComision) AS 'Empleados con Comision';
```

	Empleados	Empleados con...
▶	34	16

**FROM Empleados;COUNT(\*)** es algo diferente en que devuelve un contador con el número de filas recuperadas, contengan o no valores NULL. **COUNT(\*)** está optimizado para regresar mucho más rápido si la sentencia **SELECT** recupera de una tabla, no se piden otras columnas y no existe cláusula **WHERE**. Por ejemplo:

Esta optimización se aplica sólo a tablas **MyISAM** y **ISAM**, ya que se almacena un registro de cuenta exacto para estos tipos de tabla y puede ser accedida muy rápidamente. Para máquinas de almacenamiento transaccionales (**InnoDB**, **BDB**), almacenar una fila de cuenta exacta es más problemático porque pueden ocurrir múltiples transacciones, y cada una puede afectar a la cuenta.

```
SELECT EmCodigoDepartamento, DeNombre, COUNT(*) AS 'Empleados por Departamento'
FROM Empleados INNER JOIN Departamentos ON DeCodigo = EmCodigoDepartamento
GROUP BY EmCodigoDepartamento;
```

EmCodigoDeps...	DeNombre	Empleados por ...
▶ 100	DIRECCION GENERAL	3
110	DIRECCION COMERCIAL	3
111	SECTOR INDUSTRIAL	8
112	SECTOR SERVICIOS	7
120	ORGANIZACION	1
121	PERSONAL	4
122	PROCESO DE DATOS	5
130	FINANZAS	3

Si se usa una función de grupo en una sentencia que contenga la cláusula **GROUP BY**, equivale a agrupar todas las filas.

## COUNT DISTINCT

### COUNT(DISTINCT expr,[expr...])

Devuelve un contador con el número de valores diferentes, distintos de NULL:

```
SELECT COUNT(DISTINCT EmCodigoDepartamento) AS 'Departamentos con Empleados'
```

Departamentos con Empleados	
	8

**FROM Empleados;** En MySQL se puede obtener el número de una combinación de expresiones diferentes que no contengan NULL mediante una lista de expresiones. En SQL-99 se puede hacer una concatenación de todas las expresiones dentro de COUNT(DISTINCT ...).

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

## AVG

AVG(expr)

Devuelve el valor medio de expr:

```
SELECT AVG(EmSalario) AS 'Salario Medio' FROM Empleados
```

Salario Medio
3029.4118

```
SELECT EmCodigoDepartamento, DeNombre, AVG(EmSalario) AS 'Salario Medio'
FROM Empleados INNER JOIN Departamentos ON DeCodigo =
EmCodigoDepartamento
GROUP BY EmCodigoDepartamento;
```

EmCodigoDepa...	DeNombre	Salario Medio
100	DIRECCION GENERAL	5166.6667
110	DIRECCION COMERCIAL	2983.3333
111	SECTOR INDUSTRIAL	2181.2500
112	SECTOR SERVICIOS	2671.4286
120	ORGANIZACION	2700.0000
121	PERSONAL	3100.0000
122	PROCESO DE DATOS	3240.0000
130	FINANZAS	3700.0000

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

## MIN / MAX

MIN(expr)

MAX(expr)

Devuelve el valor mínimo o máximo de expr. **MIN()** y **MAX()** pueden tomar como argumento una cadena, en ese caso devolverán el valor de la cadena mínima o máxima.

```
SELECT DeNombre, MIN(EmSalario) AS 'Menor Salario', MAX(EmSalario) AS
'Mayor Salario'
FROM Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento
```

DeNombre	Menor Salario	Mayor Salario
DIRECCION COMERCIAL	2000	4800
DIRECCION GENERAL	3800	7200
FINANZAS	2900	4200
ORGANIZACION	2700	2700
PERSONAL	1900	4400
PROCESO DE DATOS	1750	4500
SECTOR INDUSTRIAL	1000	3100
SECTOR SERVICIOS	1800	3800

GROUP BY DeNombre; Con **MIN()**, **MAX()** y otras funciones, MySQL normalmente compara las columnas *ENUM* y *SET* por sus valores de cadena, en lugar de pos sus posiciones relativas dentro del conjunto.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

```
SELECT MIN(EmSalario) AS 'Menor Salario', MAX(EmSalario) AS 'Mayor Salario'
FROM Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento;
```

Menor Salario	Mayor Salario
1000	7200

## STD / STDDEV

STD(expr)

STDDEV(expr)

Devuelve la desviación estándar de la expresión (la raíz cuadrada de VARIANCE()). Esta es una extensión para SQL-99. El formato de **STDDEV()** de esta función se proporciona para compatibilidad con **Oracle**.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

```
SELECT  STD(EmSalario), STDDEV(EmSalario)
FROM    Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento;
```

STD(EmSalario)	STDDEV(EmSalario)
1217.7678	1217.7678

```
SELECT  DeNombre, STD(EmSalario), STDDEV(EmSalario)
FROM    Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento
GROUP BY DeNombre;
```

DeNombre	STD(EmSalario)	STDDEV(EmSalario)
DIRECCION COMERCIAL	1286.0361	1286.0361
DIRECCION GENERAL	1465.9089	1465.9089
FINANZAS	571.5476	571.5476
ORGANIZACION	0.0000	0.0000
PERSONAL	886.0023	886.0023
PROCESO DE DATOS	1102.4518	1102.4518
SECTOR INDUSTRIAL	599.9674	599.9674
SECTOR SERVICIOS	743.9552	743.9552

## SUM

### SUM(expr)

Devuelve la suma de la expresión expr. Si el conjunto de resultados no tiene filas, devuelve NULL.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

```
SELECT  DeNombre, SUM(EmSalario) 'Total Salarios', SUM(EmComision)
'Total Comisiones'
FROM    Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento
GROUP BY DeNombre;
```

DeNombre	Total Salarios	Total Comisiones
DIRECCION COMERCIAL	8950	500
DIRECCION GENERAL	15500	NULL
FINANZAS	11100	800
ORGANIZACION	2700	800
PERSONAL	12400	NULL
PROCESO DE DATOS	16200	NULL
SECTOR INDUSTRIAL	17450	7300
SECTOR SERVICIOS	18700	5900

## VARIANCE

### VARIANCE(expr)

Devuelve la varianza estándar de la expresión expr (considerando la filas como la población completa, no como una muestra; de modo que usa el número de filas como denominador). Esto es una extensión a SQL-99 (disponible sólo en versión 4.1 o superior).

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

```
SELECT      DeNombre,      VARIANCE(EmSalario)      'Varianza Salarios',  
VARIANCE(EmComision) 'VARIANZA Comisiones'  
FROM Departamentos INNER JOIN Empleados ON DeCodigo =  
EmCodigoDepartamento  
GROUP BY DeNombre;
```

DeNombre	Varianza Salarios	VARIANZA Comisiones
DIRECCION COMERCIAL	1653888.8889	0.0000
DIRECCION GENERAL	2148888.8889	NULL
FINANZAS	326666.6667	0.0000
ORGANIZACION	0.0000	0.0000
PERSONAL	785000.0000	NULL
PROCESO DE DATOS	1215400.0000	NULL
SECTOR INDUSTRIAL	359960.9375	5306.1224
SECTOR SERVICIOS	553469.3878	11388.8889

## GROUP CONCAT

### GROUP\_CONCAT(expr)

Sintaxis completa:

```
GROUP_CONCAT([DISTINCT] expr [,expr ...]  
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC]  
[,col ...]]  
[SEPARATOR str_val])
```

Devuelve una cadena con la concatenación de los valores del grupo:

```
SELECT DeNombre, GROUP_CONCAT(EmNombre)  
FROM Departamentos INNER JOIN Empleados ON DeCodigo =  
EmCodigoDepartamento  
GROUP BY DeNombre;
```

DeNombre	GROUP_CONCAT(EmNombre)
DIRECCION COMERCIAL	PEREZ, MARCOS,MORAN, CARMEN,CAMPOS, ROMULO
DIRECCION GENERAL	GALVEZ, PILAR,ALBA, ADRIANA,LOPEZ, ANTONIO
FINANZAS	FLOR, DOROTEA,GARCIA, AUGUSTO,FIERRO, CLAUDIA
ORGANIZACION	GIL, GLORIA
PERSONAL	PONS, CESAR,PEREZ, JULIO,VEIGA, JULIANA,RUIZ, FABIOLA
PROCESO DE DATOS	POLO, OTILIA,SANZ, CORNELIO,CAMPS, AURELIO,MUÑOZ, AZUCENA,MORA, VALERIANA
SECTOR INDUSTRIAL	AGUIRRE, AUREO,SANZ, LAVINIA,LARA, DORINDA,LARA, LUCRECIA,DURAN, LIVIA,PINO, DIANA,VAZQUEZ, HONORIA,SANTOS, SANCHO
SECTOR SERVICIOS	LASA, MARIO,TEROL, LUCIANO,GARCIA, OCTAVIO,DIEZ, AMELIA,MARTIN, MICAELA,PEREZ, SABINA,TORRES, HORACIO

O:

```
SELECT DeNombre, GROUP_CONCAT(EmNombre ORDER BY EmNombre DESC
SEPARATOR " ")
FROM Departamentos INNER JOIN Empleados ON DeCodigo =
EmCodigoDepartamento
GROUP BY DeNombre;
```

DeNombre	GROUP_CONCAT(EmNombre ORDER BY EmNombre DESC SEPARATOR " ")
DIRECCION COMERCIAL	PEREZ, MARCOS MORAN, CARMEN CAMPOS, ROMULO
DIRECCION GENERAL	LOPEZ, ANTONIO GALVEZ, PILAR ALBA, ADRIANA
FINANZAS	GARCIA, AUGUSTO FLOR, DOROTEA FIERRO, CLAUDIA
ORGANIZACION	GIL, GLORIA
PERSONAL	VEIGA, JULIANA RUIZ, FABIOLA PONS, CESAR PEREZ, JULIO
PROCESO DE DATOS	SANZ, CORNELIO POLO, OTILIA MUÑOZ, AZUCENA MORA, VALERIANA CAMPS, AURELIO
SECTOR INDUSTRIAL	VAZQUEZ, HONORIA SANZ, LAVINIA SANTOS, SANCHI PINO, DIANA LARA, LUCRECIA LARA, DORINDA DURAN, LIVIA AGUIRRE, AUREO
SECTOR SERVICIOS	TORRES, HORACIO TEROL, LUCIANO PEREZ, SABINA MARTIN, MICAELA LASA, MARIO GARCIA, OCTAVIO DIEZ, AMELIA

En MySQL se pueden obtener los valores concatenados de combinaciones de expresiones. Se pueden eliminar valores duplicados usando DISTINCT. Si se desea ordenar los valores del resultado se puede usar la cláusula ORDER BY. Para ordenar en orden inverso, añadir la palabra clave DESC (descendente) al nombre de la columna por la que se está ordenando en la cláusula ORDER BY. Por defecto, el orden es ascendente; que se puede especificar explícitamente usando la palabra clave ASC. SEPARATOR es el valor de cadena que se insertará entre los valores del resultado. Por defecto es una coma (","). Se puede eliminar el separador por completo especificando SEPARATOR "". También se puede limitar la longitud máxima con la variable group\_concat\_max\_len en la configuración. La sintaxis para hacerlo durante la ejecución de MySQL:

```
SET [SESSION | GLOBAL] group_concat_max_len = unsigned_integer;
```

Si se ha asignado una longitud máxima, el resultado se truncará a esa longitud. La función **GROUP\_CONCAT()** es una implementación mejorada de la función básica **LIST()** soportada por **Sybase SQL Anywhere**. **GROUP\_CONCAT()** mantiene compatibilidad con la extremadamente limitada funcionalidad de **LIST()**, si sólo se especifica una columna y ninguna otra opción. **LIST()** tiene un modo de orden por defecto.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

## BIT AND

### BIT\_AND(expr)

Devuelve la operación de bits AND para todos los bits de expr. El cálculo se realiza con precisión de 64 bits (BIGINT). Desde MySQL 4.0.17, esta función devuelve 18446744073709551615 si no existen filas que coincidan. (Es el valor BIGINT sin signo con todos los bits a 1.) Antes de 4.0.17, la función devolvía -1 en ese caso.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

## **BIT\_OR**

**BIT\_OR(expr)**

Devuelve la operación de bits OR para todos los bits de expr. El cálculo se realiza con precisión de 64 bits (BIGINT). Esta función devuelve 0 si no existen filas que coincidan.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.

## **BIT\_XOR**

**BIT\_XOR(expr)**

Devuelve la operación de bits XOR para todos los bits de expr. El cálculo se realiza con precisión de 64 bits (BIGINT). Esta función devuelve 0 si no existen filas que coincidan. Esta función está disponible desde MySQL 4.1.1.

Si se usa una función de grupo en una sentencia que contenga la cláusula GROUP BY, equivale a agrupar todas las filas.