

## Laboratorio #5

- Funciones de Agregación
- GROUP BY, ORDER BY y HAVING



## 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, de un conjunto de valores nos permiten obtener valores de medias, máximos, mínimos, etc.

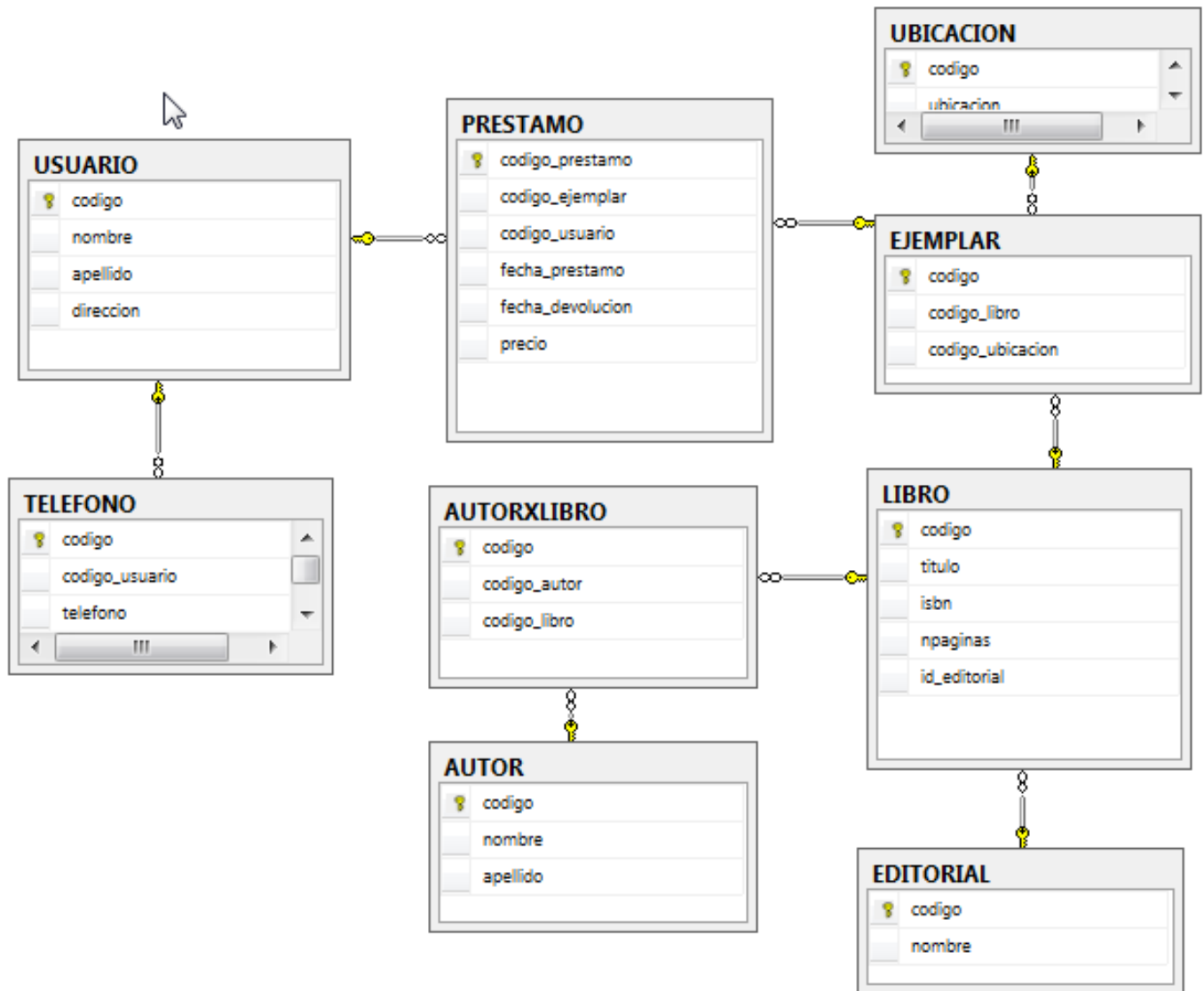
Algunas funciones de agregación que soporta SQL son:

Función de agregación	Resultado
AVG	Devuelve el valor promedio del campo que se especifique en la consulta.
COUNT	Devuelve el número total de filas seleccionadas por la consulta.
MIN	Devuelve el valor mínimo del campo que se especifique en la consulta.
MAX	Devuelve el valor máximo del campo que se especifique en la consulta.
SUM	Suma los valores del campo especificado.

Todas estas funciones sólo se puede utilizar en columnas numéricas y se aplican a una sola columna, que se especificara entre paréntesis, excepto la función COUNT, que se puede aplicar a una columna o conjunto de columnas (indicar un “\*”) numéricas o no numéricas.

NOTA: Las funciones anteriores son las básicas en SQL, pero cada sistema gestor de bases de datos relacionales ofrece su propio conjunto, más amplio, con otras funciones de agregación particulares.

Para los ejemplos de esta práctica consideremos la siguiente base de datos:



## MIN () y MAX ()

Las funciones MIN () y MAX () devuelven el valor más pequeño y el valor más grande, respectivamente, **de la columna seleccionada**.

Sintaxis básica:

```
SELECT MIN(column_name)
FROM table_name
[WHERE condition];
```

```
SELECT MAX(column_name)
FROM table_name
[WHERE condition];
```

### Ejemplo:

Seleccionar el número más alto de páginas que tienen los libros.

```
SELECT MAX(npaginas) as 'Numero mayor' FROM libro;
```

Para el número menor de páginas solo se cambia la función MAX por MIN.

## SUM () y AVG ()

La función AVG () devuelve el valor promedio y la función SUM () devuelve la suma total de una columna numérica.

Sintaxis:

```
SELECT SUM(column_name)
      FROM table_name
      [WHERE condition];
```

```
SELECT AVG(column_name)
      FROM table_name
      [WHERE condition];
```

### Ejemplo:

Calcular la suma total de lo recaudado en los préstamos.

```
SELECT SUM(precio) as 'Suma de dinero' FROM PRESTAMO;
```

Para calcular el promedio se cambia la función SUM por AVG.

## COUNT ()

La función COUNT (column\_name) devuelve el número de valores de la columna especificada.

Sintaxis:

```
SELECT COUNT(column_name) FROM table_name [WHERE condition];
```

Para contar todos los registros de una tabla se usa:

```
SELECT COUNT(*) FROM table_name;
```

### Ejemplo:

Contar cuantos usuarios hay en la base de datos:

```
SELECT COUNT(*) AS 'Total de usuarios' FROM USUARIO;
```

Si se necesita de una columna los registros que no son repetidos se usa **COUNT (DISTINCT column\_name)** que devuelve el número de valores distintos de la columna especificada.

Su sintaxis es la siguiente:

```
SELECT COUNT(DISTINCT column_name) FROM table_name;
```

### Ejemplo:

Contar los usuarios que han hecho préstamos

```
SELECT COUNT(DISTINCT u.nombre) AS 'Usuarios Activos'
FROM USUARIO u INNER JOIN PRESTAMO p
ON p.codigo_usuario = u.codigo;
```

Nótese que si un usuario ha prestado más de un libro solo lo cuenta una vez.

## GROUP BY

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.

La instrucción GROUP BY se usa a menudo con funciones agregadas (COUNT, MAX, MIN, SUM, AVG) para agrupar el conjunto de resultados por una o más columnas.

Sintaxis:

```
SELECT column_name
FROM table_name
WHERE condition
GROUP BY column_name;
```

### Ejemplo:

Contar cuantos telefonos tiene cada usuario

```
SELECT u.nombre + ' ' + u.apellido as Usuario, COUNT(t.telefono) as 'Total telefonos'
FROM TELEFONO t INNER JOIN USUARIO u ON u.codigo=t.codigo_usuario
GROUP BY u.nombre + ' ' + u.apellido;
```

## ORDER BY

La instrucción ORDER BY se utiliza para ordenar el conjunto de resultados en orden ascendente o descendente.

ORDER BY ordena los registros en orden ascendente de forma predeterminada, para ordenar los registros en orden descendente, se usa la palabra clave DESC.

Sintaxis:

```
SELECT column1, column2, ...
      FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

Ejemplo:

Contar cuantos telefonos tiene cada usuario y ordenarlos alfabeticamente descendente.

```
SELECT u.nombre + ' ' + u.apellido as Usuario, COUNT(t.telefono) as 'Total telefonos'
FROM TELEFONO t INNER JOIN USUARIO u ON u.codigo=t.codigo_usuario
GROUP BY u.nombre + ' ' + u.apellido
ORDER BY 'Usuario' DESC;
```

## HAVING

La instrucción HAVING es equivalente a la instrucción WHERE, pero funciona con condiciones booleanas sobre columnas calculadas con funciones de agregación.

Sintaxis:

```
SELECT column_name
      FROM table_name
GROUP BY column_name
      HAVING condition
ORDER BY column_name;
```

Ejemplo:

Contar cuantos telefonos tiene cada usuario que tenga en su nombre completo al menos una letra “o” y ordenarlos alfabeticamente descendente.

```
SELECT u.nombre + ' ' + u.apellido as Usuario, COUNT(t.telefono) as 'Total telefonos'
FROM TELEFONO t INNER JOIN USUARIO u ON u.codigo=t.codigo_usuario
GROUP BY u.nombre + ' ' + u.apellido
HAVING u.nombre + ' ' + u.apellido LIKE '%o%'
ORDER BY 'Usuario' DESC
```