grupoesfera | INNOVACION Y DESARROLLO

Introducción al lenguaje de consulta estructurado (SQL)

Structured Query Language

Agenda

- Sentencias avanzadas en SQL.
  - Funciones de agregación
  - Funciones propietarias
  - Uso del ORDER BY

Resumen

- Sintaxis

```
SELECT campo1, campo2, … , campoN
FROM tabla1, tabla2, … , tablaN
WHERE condicion1, condicion2, … , condicionN
```

Nota: Los campos pueden ser calculados

- Operadores Where
  - *BETWEEN*, LIKE, IN

Resumen (cont.)

- Consultas en más de una tabla:

```
SELECT campo1, campo2, …, campoN
FROM Tabla1

UNION [ALL]

SELECT campo4, campo5, …, campoM
FROM Tabla2
```

Resumen (cont.)

- Consultas en más de una tabla:

```
SELECT [nombreTabla].campo1 as nombreCampo1,
       [nombreTabla].campo2 as nombreCampo2, … ,
       [nombreTabla].campoN as nombreCampoN
FROM tabla1 as nombreTabla1,
     tabla2 as nombreTabla2, … ,
     tablaN as nombreTablaN
WHERE condicion1, condicion2, … , condicionN
```

Nota: Es MUY importante agregar las condiciones que relacionan las tablas intervinientes.

Resumen (cont.)

- Consultas en más de una tabla:

```
SELECT tabla1.campo1, tabla2.campo2, … , campoN
FROM tabla1
     [INNER|LEFT|RIGHT] JOIN tabla2
     ON tabla1.campo1 = tabla2.campo2
WHERE condicion1, condicion2, … , condicionN
```

Funciones de agregación

- Son funciones propias del motor de base de datos que nos permiten realizar cálculos con los datos de las tablas.
- Procesan un conjunto de valores y retornan un solo valor.
- Las funciones de agregación se usan frecuentemente con la cláusula GROUP BY.
- Funciones de agregación:
  - **AVG**() - Promedio.
  - **COUNT**() - Número de registros.
  - **MAX**() - Máximo valor.
  - **MIN**() - Mínimo valor.
  - **SUM**() - Suma.

Funciones de agregación (cont.)

- Sintaxis

```
SELECT función(campo1)
FROM tabla
WHERE condicion1, condicion2, … , condicionN
```

- Por ejemplo, para contar cuantas facturas hay en la tabla de facturas, se podría realizar la siguiente consulta

```
SELECT count(*)
FROM Factura
```

DISTINCT

- En una tabla, alguna columna pude tener valores repetidos. Si se quieren obtener los distintos valores para una columna de una tabla, se debe utilizar la palabra DISTINCT.

- Sintaxis

```
SELECT DISTINCT campo1, campo2, … , campoN
FROM tabla
WHERE condicion1, condicion2, … , condicionN
```

- *Si se utiliza más de una columna, devuelve las todos los registros cuya combinacion sea única.*

DISTINCT (cont.)

- La cláusula DISTINCT se puede utilizar en conjunto con algunas funciones de agregación.

- Por ejemplo, la siguiente sentencia es válida.

```
SELECT SUM(DISTINCT montoTotal)
FROM Factura
```

Funciones propietarias

- Existen otro tipo de funciones que nos ofrecen los motores de base de datos para utilizar en las consultas.
- Estas funciones dependen del motor de base de datos y puden variar entre las distintas implementaciones.
- En su gran mayoría, las funciones básicas (manejo de cadenas de caracteres y fechas) están presente en todos los motores de base de datos, pero las la sintaxis puede variar de uno a otro.
- Existen funciones en T-SQL para obtener información de la base de datos, correr estadísticas y reportes, administrar la seguridad de la base de datos.

Funciones propietarias (cont.)

- Estos son algunos grupos de funciones:
  - Funciones de Fecha: Realizan operaciones en campos de tipo fecha y devuelven cadenas de caracteres números u otras fechas.
  - Funciones Matemáticas: Realizan operaciones en los campos y devuelven un valor numérico.
  - Funciones de Cadenas de Caracteres: Realizan operaciones en una cadena de caracteres y devuelven una cadena de caracteres o un valor numérico.

Funciones propietarias (cont.)

- Funciones de Fecha
  - **DAY**(*fecha*)/**MONTH**(*fecha*)/**YEAR**(*fecha*): Devuelve cada una de las partes de una fecha.
  - **GETDATE**(): Devuelve la fecha actual del servidor donde está corriendo el motor de base de datos. "**GETDATE**" Se usa en MSSQL, en MySQL existen las siguientes funciones:

    CURDATE(): Devuelve la fecha actual (no incluye la hora)

    CURTIME(): Devuelve la hora actual (no incluye la fecha)

    NOW(): Devuelve la fecha actual y la hora

Funciones propietarias (cont.)

- Funciones de Fecha
  - **DATEADD**(*unidad*, *numero*, *fecha*): Devuelve una fecha en la que sumó el número de unidades a la fecha original. La unidad puede ser “year”, “month”, “day”. “**DATEADD**” Se usa en MSSQL, en MySQL se usa “**DATE_ADD**”, ejemplos:
  - `SELECT DATE_ADD('1998-01-02', INTERVAL 6 DAY);`
    - RESULTADO: 1998-01-08
  - `SELECT DATE_ADD('1998-01-02', INTERVAL 3 MONTH);`
    - RESULTADO: 1998-04-02

Funciones propietarias (cont.)

- Funciones Matemáticas
  - **ABS**(*valor*): Devuelve el valor absoluto del argumento.
  - **ROUND**(*valor*, *decimales*): Devuelve el número *valor* redondeado a tantos decimales como se desee.
  - **FLOOR**(*valor*): Devuelve el entero más grande que sea menor o igual que el valor especificado.
  - **POWER**(*valor*, *potencia*): Devuelve el *valor* elevado a la *potencia*.

Funciones propietarias (cont.)

- Funciones de Cadenas de Caracteres
  - **CONCAT**(*cadena1*, *cadena2*, …, *cadenaN*): Devuelve una cadena de caracteres con la unión de todas las cadenas especificadas.
  - **LEFT**(*cadena*, *n*): Devuelve los primeros 'n' caracteres de la *cadena*.
  - **LEN**(*cadena*): Devuelve la cantidad de caracteres que existen en la *cadena*. En MySQL el equivalente es "**LENGTH**"
  - **LOWER**(*cadena*): Devuelve la cadena de caracteres convertida a minúsculas.

TOP

- MS SQL Server permite obtener los primeron N registros al momento de realizar una colsulta.
- Esto es útil para utilizar en tablas con gran volumen de datos.
- Sintaxis

```
SELECT TOP cantidad
FROM tabla
WHERE condicion
```

- En MySQL se usa el LIMIT

```
SELECT campo1,campo2,...,campoN
FROM tabla
LIMIT cantidad
```

ORDER BY

- Se utiliza esta cláusula para ordenar los resultados de una consulta.
- Se puede ordenar por una o varias columnas.
- Por defecto, al ordenar un resultado el orden se realiza en forma Ascendente (ASC).
- Si se desea ordernar en forma descendiente, se debe utilizar la palabra DESC.
- En una base de datos, no puede determinarse el orden que tendrán los resultados a menos que se especifique la cláusula ORDER BY.

ORDER BY (cont.)

- Sintaxis

```
SELECT campo1, campo2, … , campoN
FROM tabla1, tabla2, … , tablaN
WHERE condicion1, condicion2, … , condicionN
ORDER BY campo1 ASC, campo2 ASC

SELECT campo1, campo2, … , campoN
FROM tabla1, tabla2, … , tablaN
WHERE condicion1, condicion2, … , condicionN
ORDER BY campo1 DESC, campo2 DESC
```

Ejercicios

- E23: Obtener todos los nombres de los productos y los empleados en un listado determinando cuales son productos y cuales son empleados, listando primero los productos y luego los empleados.
- E24: Obtener la cantidad de Facturas existentes en la base de datos.
- E25: Obtener la monto total de Facturas que se realizaron en el último año.
- E26: Obtener el promedio gastado por el cliente “Ana”.
- E27: Obtener el máximo valor de una factura emitida por la sucursal “Olivos”.

Ejercicios (cont.)

- E28: Obtener los nombres de los distintos clientes.
- E29: Obtener la cantidad de categorías distintas que existen para los clientes.
- E30: Obtener solo los 5 primeros clientes.
- E31: Obtener los clientes que cumplan años en febrero y que hayan tenido una factura en la sucursal “Remedios de Escalada”.
- E32: Obtener los primeros 6 caracteres de las sucursales seguidas por … (tres puntos).

FIN