

Contenidos teóricos

Sitio: Agencia de Aprendizaje a lo largo de la Vida

Curso: CAC Big Data MODULO I 2C

Libro: Contenidos teóricos

Imprimido por: Eduardo Moreno

Día: lunes, 21 de agosto de 2023, 20:54

Tabla de contenidos

1. DML -Parte 2
2. Alias en SQL
3. Columnas calculadas

1. DML -Parte 2

SQL - DML(II)

4. Búsquedas en la base de datos

Obtener los datos de las columnas c1 y c2 de una tabla

```
SELECT c1, c2 FROM tabla;
```

Obtener todas las filas y columnas de una tabla

```
SELECT * FROM tabla;
```

Obtener datos y filtrar filas con una condición

```
SELECT c1, c2 FROM tabla WHERE condición [AND/OR condición2];
```

Obtener filas distintas de una tabla (que no se repitan los datos)

```
SELECT DISTINCT c1 FROM tabla WHERE condición [AND/OR condición2];
```

Ordenar los resultados de forma ascendente o descendente

```
SELECT c1, c2 FROM tabla ORDER BY c1 ASC [DESC];
```

Obtener los primeros 10 registros de una tabla con la columna c1 ordenada en forma descendente

```
SELECT c1, c2 FROM tabla ORDER BY c1 DESC LIMIT 10;
```

Ejemplo de búsquedas en múltiples tablas. Complementar con el apunte de JOIN.

```
SELECT c1, c2 FROM tabla1 INNER JOIN tabla2 ON condición;
```

```
SELECT c1, c2 FROM tabla1 LEFT JOIN tabla2 ON condición;
```

```
SELECT c1, c2 FROM tabla1 RIGHT JOIN tabla2 ON condición;
```

Obtener todas las columnas de los registros cuya columna *variable1* tenga un valor entre *nro1* y *nro2*

```
SELECT * FROM tabla WHERE variable1 BETWEEN nro1 AND nro2;
```

Obtener todas las columnas de de los registros cuya columna *variable1* contenga la cadena *una_cadena*

```
SELECT * FROM tabla WHERE variable1 [NOT] LIKE una_cadena;
```

Obtener todas las columnas de de los registros cuya columna *variable1* contenga una cadena que contiene la letra 'm'

```
SELECT * FROM tabla WHERE variable1 LIKE '%m%';
```

Obtener todas las columnas de de los registros cuya columna *variable1* contenga una cadena que termina con 's'

```
SELECT * FROM tabla WHERE variable1 LIKE '%s';
```

Obtener todas las columnas de de los registros cuya columna *variable1* contenga una cadena que comienza con 'A'

```
SELECT * FROM tabla WHERE variable1 LIKE 'A%';
```

2. Alias en SQL

SQL - Alias

Los alias se utilizan en SQL para asignarle un nombre temporal a una columna o a una tabla.

Sirven para abreviar los nombres en sentencias muy extensas y para que el código sea más legible.

El alias se elimina en cuanto termina la consulta.

Para crear un alias, utilizamos la palabra clave AS.

Sintaxis:

```
SELECT nombre_columna AS alias_col
```

```
FROM table_name
```

```
SELECT nombre_columna
```

```
FROM table_name AS alias_tabla
```

Ejemplo:

TABLA_CLIENTES			TABLA_PEDIDOS				
cliente_nro	nombre	apellido	nro_orden	empleado_nro	cliente_nro	proveedor	precio
1	Sally	Thompson	1	1	42	Harrison	\$235,00
2	Sally	Henderson	2	4	1	Ford	\$234,00
3	Harry	Henderson	3	1	68	Jason	\$415,00
4	Sandra	Wellington	4	2	112	Statham	\$350,00
			5	3	42	Dwayne	\$235,00
			6	2	112	Johnson	\$350,00
			7	2	42	Diesel	\$235,00

```
SELECT p.nro_orden, p.proveedor, c.nombre, c.apellido
```

```
FROM tabla_clientes AS c, tabla_pedidos AS p
```

```
WHERE c.nombre = 'Sally'
```

```
AND c.cliente_nro =p.cliente_nro
```

Sin alias:

```
SELECT tabla_pedidos.nro_orden, tabla_pedidos.proveedor, tabla_clientes.nombre, tabla_clientes.apellido
```

```
FROM tabla_clientes, tabla_pedidos
```

```
WHERE tabla_clientes.nombre = 'Sally'
```

```
AND tabla_clientes.cliente_nro =tabla_pedidos .cliente_nro
```

3. Columnas calculadas

Funciones en SQL

1. COUNT()

Devuelve el número de filas que coincide con un criterio.

Sintaxis:

```
SELECT COUNT(nombre_columna)
```

```
FROM nombre_tabla
```

```
WHERE condición;
```

2. AVG()

Devuelve el valor promedio de una columna con valores numéricos que cumplan la condición.

Sintaxis:

```
SELECT AVG(nombre_columna)
```

```
FROM nombre_tabla
```

```
WHERE condición;
```

3. SUM()

Devuelve el valor de la suma de una columna, siempre y cuando cumplan con la condición.

Sintaxis:

```
SELECT SUM(nombre_columna)
```

```
FROM nombre_tabla
```

```
WHERE condición;
```

Ejemplo:

TABLA_CLIENTES			TABLA_PEDIDOS				
cliente_nro	nombre	apellido	nro_orden	empleado_nro	cliente_nro	proveedor	precio
1	Sally	Thompson	1	1	42	Harrison	\$235,00
2	Sally	Henderson	2	4	1	Ford	\$234,00
3	Harry	Henderson	3	1	68	Jason	\$415,00
4	Sandra	Wellington	4	2	112	Statham	\$350,00
			5	3	42	Dwayne	\$235,00
			6	2	112	Johnson	\$350,00
			7	2	42	Diesel	\$235,00

```
SELECT COUNT(nombre)
```

```
FROM tabla_clientes
```

```
WHERE nombre = 'Sally';
```

```
SELECT AVG(precio)
```

```
FROM tabla_pedidos
```

```
WHERE cliente_nro = 112;
```

```
SELECT SUM(precio)
```

```
FROM tabla_pedidos
```

```
WHERE empleado_nro = 2;
```

4. Otras funciones y operaciones

MIN() / MAX() devuelven el menor y mayor valor de una columna.

Además, pueden “agregarse” provisoriamente columnas calculadas a partir de las existentes.

Si quisiera, por ejemplo, agregar una columna a la tabla de pedidos, donde me mostrara el IVA y el precio final:

```
SELECT precio, precio*0.21, precio + (precio*0.21)
```

```
FROM tabla_pedidos
```

El resultado de la búsqueda me devolverá una tabla de tres columnas: precio, iva y precio con iva.

La estructura de la tabla no se ve afectada. Como los alias, estas columnas se extinguen al terminar la consulta.

Podemos ponerle nombre a estas “nuevas columnas”, utilizando alias. En el ejemplo anterior:

```
SELECT precio, precio*0.21 AS IVA, precio + (precio*0.21)
```

```
AS Total FROM tabla_pedidos;
```