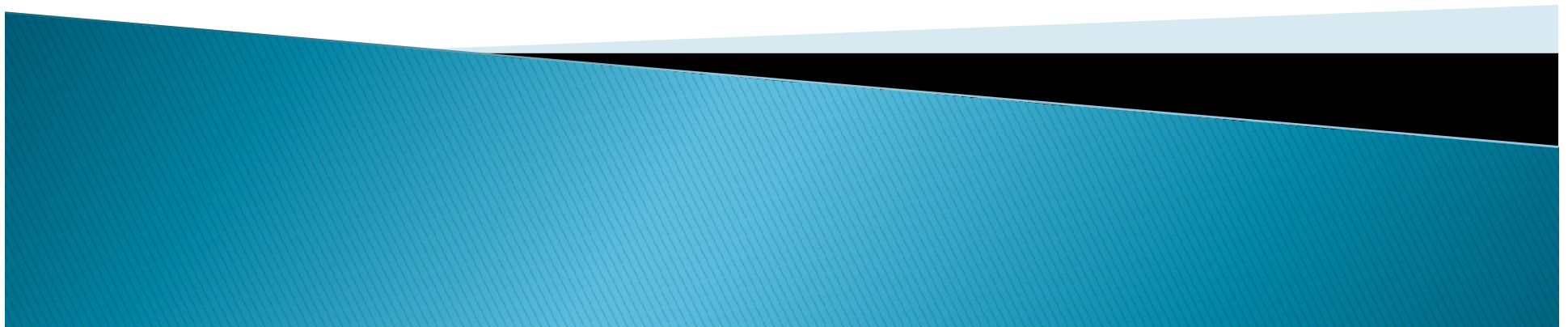


MySQL

Condiciones en Profundidad



Condiciones en profundidad

Creamos en nuestra Base de Datos 'Empresa' una tabla llamada "clientes".

```
CREATE TABLE Clientes(  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  Nombre VARCHAR(50),  
  Edad INT,  
  Email VARCHAR(100)  
);
```

Condiciones en profundidad

Introducimos datos a nuestra tabla “clientes”.

```
INSERT INTO Clientes (Nombre, Edad, Email)
VALUES ('Oscar', 25, 'oscar@gmail.com'),
('Layla', 15, 'layla@gmail.com'),
('Nicolás', 36, 'nicolas@gmail.com'),
('Chanchito', 7, 'chanchi@gmail.com');
```

Condiciones en profundidad

Realizamos una consulta de los datos de la tabla “Clientes”:

```
SELECT * FROM clientes;
```

Realizamos una consulta de los datos de la tabla “Clientes” discriminando por edad, por ejemplo:

```
SELECT * FROM clientes WHERE edad >= 15;
```

Condiciones en profundidad

Pero estas búsquedas con SELECT se pueden especializar mucho:

```
SELECT * FROM clientes WHERE edad > 15 and  
email = 'nicolas@gmail.com';
```

Una búsqueda con OR:

```
SELECT * FROM clientes WHERE edad > 15 or  
email='layla@gmail.com';
```

Condiciones en profundidad

Ahora una búsqueda con negación:

```
SELECT * FROM clientes WHERE email !=  
'layla@gmail.com';
```

Una búsqueda con BETWEEN:

```
SELECT * FROM clientes WHERE edad  
between 15 and 30;
```

Condiciones en profundidad

Ahora una búsqueda por aproximación:

```
SELECT * FROM clientes WHERE email like  
'%gmail.%';
```

En SQL, el carácter "%" se utiliza como comodín en las consultas que involucran la cláusula "LIKE". El carácter "%" representa cualquier secuencia de cero o más caracteres.

Condiciones en profundidad

Ahora una búsqueda ordenada:

```
SELECT * FROM clientes ORDER BY edad asc;
```

En orden descendente:

```
SELECT * FROM clientes ORDER BY edad desc;
```


Condiciones en profundidad

También puedo hacer búsquedas muy concretas:

```
SELECT max(edad) as mayor from clientes;
```

Y ahora el mínimo:

```
SELECT min(edad) as menor from clientes;
```

Condiciones en profundidad

Hasta ahora hemos estado usando el * para el alcance de las búsquedas, pero puedo apuntar solo a columnas:

```
SELECT nombre, edad FROM clientes;
```

Por último, puedo darle un alias a una consulta:

```
SELECT nombre, edad AS años FROM clientes;
```

Ejercicios SQL 04

Añade la tabla “Proveedores”:

```
-- Crear la tabla proveedores  
CREATE TABLE proveedores (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  nombre VARCHAR(100),  
  dirección VARCHAR(100),  
  ciudad VARCHAR(50),  
  codigo_postal VARCHAR(10),  
  país VARCHAR(50)  
);
```

Ejercicios SQL 04

Añade los siguientes datos a la tabla “Proveedores”:

INSERT INTO proveedores

(nombre, dirección, ciudad, codigo_postal, país)

VALUES

('Proveedor A', 'Calle Principal 123', 'Ciudad A', '12345', 'País A'),
('Proveedor B', 'Avenida Central 456', 'Ciudad B', '67890', 'País B'),
('Proveedor C', 'Carrera Norte 789', 'Ciudad C', '54321', 'País C'),
('Proveedor D', 'Avenida Sur 987', 'Ciudad D', '09876', 'País D'),
('Proveedor E', 'Calle Este 654', 'Ciudad E', '24680', 'País E');

Ejercicios SQL 04

- Selecciona todos los proveedores:

```
SELECT * FROM proveedores;
```

- Selecciona el nombre y la ciudad de todos los proveedores:

```
SELECT nombre, ciudad FROM proveedores;
```

- Selecciona los proveedores cuyo país sea "País B":

```
SELECT * FROM proveedores WHERE país = 'País B';
```

Ejercicios SQL 04

- Selecciona los proveedores que tengan el código postal terminado en "5":

```
SELECT * FROM proveedores WHERE codigo_postal LIKE '%5';
```

- Selecciona los proveedores ordenados por nombre en orden descendente:

```
SELECT * FROM proveedores ORDER BY nombre DESC;
```

- Selecciona todos los proveedores cuyo país sea "País A" y la ciudad sea "Ciudad A"..

```
SELECT * FROM proveedores WHERE país = 'País A'  
AND ciudad = 'Ciudad A';
```

Ejercicios SQL 04

- Selecciona todos los proveedores cuya ciudad sea "Ciudad B" o "Ciudad C":

```
SELECT * FROM proveedores WHERE ciudad = 'Ciudad B' OR ciudad = 'Ciudad C';
```

- Selecciona todos los proveedores cuyo país no sea "País D":

```
SELECT * FROM proveedores WHERE NOT pais = 'País D';
```

- Selecciona todos los proveedores cuyo código postal esté comprendido entre '10000' y '20000', ambos inclusive:

```
SELECT * FROM proveedores WHERE codigo_postal BETWEEN '10000' AND '20000';
```

Ejercicios SQL 04

- Selecciona el valor máximo de la columna "edad" de la tabla "clientes" y lo etiqueta como "edad_maxima".

```
SELECT MAX(edad) AS edad_maximo FROM clientes;
```

- Selecciona el valor mínimo de la columna "edad" de la tabla "clientes" y se etiqueta como "edad_minima".

```
SELECT MIN(edad) AS edad_minima FROM clientes;
```


Ejercicios SQL 05

- Creación de la tabla "Productos" en la base de datos "Empresa":

```
CREATE TABLE Productos (  
  ID INT AUTO_INCREMENT PRIMARY KEY,  
  Marca VARCHAR(255),  
  Modelo VARCHAR(255),  
  Stock INT,  
  Precio DECIMAL(10, 2)  
);
```

Ejercicios SQL 05

- Inserción de diez registros en la tabla "Productos":

```
INSERT INTO Productos (Marca, Modelo, Stock, Precio)
VALUES
('Samsung', 'Galaxy S21', 10, 999.99),
('Apple', 'iPhone 12', 5, 1299.99),
('Sony', 'PlayStation 5', 15, 499.99),
('LG', 'OLED TV', 8, 1999.99),
('Nike', 'Air Max 270', 12, 129.99),
('Adidas', 'Ultraboost', 3, 159.99),
('HP', 'Laptop', 20, 899.99),
('Dell', 'Monitor', 7, 299.99),
('Canon', 'EOS R5', 9, 3499.99),
('Logitech', 'Wireless Mouse', 6, 49.99);
```

Ejercicios SQL 05

- Consulta para actualizar el precio de los productos de la marca "Samsung":

```
UPDATE Productos SET Precio = Precio * 0.9 WHERE Marca = 'Samsung';
```

- Consulta para obtener los productos con un stock mayor a 10 unidades y un precio menor a 1000:

```
SELECT * FROM Productos WHERE Stock > 10 AND Precio < 1000;
```

- Consulta para obtener los productos con la marca "Apple" o "Sony":

```
SELECT * FROM Productos WHERE Marca = 'Apple' OR Marca = 'Sony';
```

Ejercicios SQL 05

- Consulta para obtener los productos cuyo modelo contiene la palabra "Max":

```
SELECT * FROM Productos WHERE Modelo LIKE '%Max%';
```

- Consulta para obtener los productos con un precio entre 500 y 1500:

```
SELECT * FROM Productos WHERE Precio BETWEEN 500 AND 1500;
```

- Consulta para obtener los productos ordenados por marca en orden ascendente:

```
SELECT * FROM Productos ORDER BY Marca ASC;
```