Corporación Universitaria Empresarial Alexander von Humboldt
VIGILADA MINEDUCACIÓN

SELECT Estudiante.Nombre AS NombreEstudiante, Curso.Nombre AS Curso, LEN(Estudiante.Nombre) + LEN(Curso.Nombre) AS LongitudTotal FROM Estudiante JOIN Inscripcion ON Estudiante.ID = Inscripcion.EstudianteID JOIN Curso ON Inscripcion.CursoID = Curso.ID;

En este ejemplo, he utilizado la función LEN para calcular la longitud de las cadenas de texto de los nombres de estudiantes y cursos, y luego he sumado esas longitudes. Este cálculo se almacena en la columna LongitudTotal. Puedes ajustar esta lógica según el tipo de operaciones matemáticas que desees realizar en tu ejercicio práctico.

SELECT Cliente.Nombre AS NombreCliente, Producto.Nombre AS NombreProducto, Pedido.Cantidad, Producto.PrecioUnitario, Pedido.Cantidad * Producto.PrecioUnitario AS PrecioTotal FROM Cliente JOIN Pedido ON Cliente.ID = Pedido.ClienteID JOIN Producto ON Pedido.ProductoID = Producto.ID WHERE Pedido.FechaEntrega < '2023-03-01';

En esta consulta, he agregado la columna `PrecioTotal`, que se calcula multiplicando la cantidad de cada pedido por el precio unitario del producto. Esta operación matemática te dará el precio total para cada producto en cada pedido. Puedes ajustar esta lógica según tus necesidades específicas o agregar otras operaciones que desees enseñar en tu clase.

SELECT Cliente.Nombre AS NombreCliente, SUM(Pedido.Cantidad) AS TotalProductosPedidos, SUM(Pedido.CantidadEntregada) AS TotalProductosEntregados FROM Cliente JOIN Pedido ON Cliente.ID = Pedido.ClienteID JOIN Producto ON Pedido.ProductoID = Producto.ID WHERE Pedido.FechaEntrega < '2023-03-01' GROUP BY Cliente.Nombre;

En esta consulta, se utiliza SUM para sumar la cantidad total de productos pedidos (Pedido.Cantidad) y la cantidad total de productos entregados (Pedido.CantidadEntregada) para cada cliente. La cláusula GROUP BY agrupa los resultados por el nombre del cliente.

SELECT Cliente.Nombre AS NombreCliente, Producto.Nombre AS NombreProducto, Pedido.Cantidad, Producto.PrecioUnitario, ROUND(Pedido.Cantidad * Producto.PrecioUnitario, 2) AS PrecioTotalRedondeado FROM Cliente JOIN Pedido ON Cliente.ID = Pedido.ClienteID JOIN Producto ON Pedido.ProductoID = Producto.ID WHERE Pedido.FechaEntrega < '2023-03-01';

En esta consulta, se utiliza ROUND para redondear el resultado de la multiplicación de la cantidad de productos por el precio unitario (Pedido.Cantidad * Producto.PrecioUnitario) a dos decimales. La función ROUND toma dos argumentos: la expresión que deseas redondear y el número de decimales al que deseas redondear.

SELECT Cliente.Nombre AS NombreCliente, Producto.Nombre AS NombreProducto, CEIL(SUM(Pedido.Cantidad)) AS TotalProductosPedidosRedondeado FROM Cliente JOIN Pedido ON Cliente.ID = Pedido.ClienteID JOIN Producto ON

Pedido.ProductoID = Producto.ID WHEREPedido.FechaEntrega < '2023-03-01' GROUP BY Cliente.Nombre, Producto.Nombre;

En esta consulta, SUM(Pedido.Cantidad) suma la cantidad total de productos pedidos para cada cliente y producto, y CEIL redondea este valor hacia arriba a un número entero.

SELECT Empleado.Nombre, Departamento.Nombre AS Departamento, Empleado.Salario, Empleado.Salario * 12 AS SalarioAnual FROM Empleado JOIN Departamento ON Empleado.DepartamentoID = Departamento.ID WHERE Empleado.Salario > 50000;

En esta consulta, he agregado la columna SalarioAnual, que calcula el salario anual multiplicando el salario mensual (Empleado.Salario) por 12. Puedes ajustar esta lógica según tus necesidades específicas.

Si deseas utilizar la función AVG para calcular el salario promedio de los empleados que ganan más de $50,000, aquí tienes una versión modificada de la consulta:

SELECT Departamento.Nombre AS Departamento, AVG(Empleado.Salario) AS SalarioPromedio FROM Empleado JOIN Departamento ON Empleado.DepartamentoID = Departamento.ID WHERE Empleado.Salario > 1 GROUP BY Departamento.Nombre;

En esta consulta, he utilizado la función AVG(Empleado.Salario) para calcular el salario promedio de los empleados que ganan más de $50,000 agrupados por departamento. La cláusula GROUP BY se utiliza para agrupar los resultados por el nombre del departamento.

Supongamos que deseas contar el número de empleados en cada departamento que ganan más de $50,000. Aquí tienes una consulta utilizando la función COUNT:

SELECT Departamento.Nombre AS Departamento, COUNT(*) AS CantidadEmpleados FROM Empleado JOIN Departamento ON Empleado.DepartamentoID = Departamento.ID WHERE Empleado.Salario > 50000 GROUP BY Departamento.Nombre;

En esta consulta, he utilizado COUNT(*) para contar el número de empleados en cada departamento que cumplen con la condición de tener un salario mayor a $50,000. La cláusula GROUP BY agrupa los resultados por el nombre del departamento.

Supongamos que deseas encontrar el salario máximo de cada departamento entre los empleados que ganan más de $50,000. Aquí tienes una consulta utilizando la función MAX:

SELECT Departamento.Nombre AS Departamento, MAX(Empleado.Salario) AS SalarioMaximo FROM Empleado JOIN Departamento ON Empleado.DepartamentoID = Departamento.ID WHERE Empleado.Salario > 50000 GROUP BY Departamento.Nombre;

En esta consulta, he utilizado MAX(Empleado.Salario) para encontrar el salario máximo entre los empleados de cada departamento que cumplen con la condición de tener un salario mayor a $50,000. La cláusula G ROUP BY agrupa los resultados por el nombre del departamento.

LIKE SQL

La siguiente declaración SQL selecciona a todos los clientes con un país que comience por «Es»:

```
SELECT * FROM Clientes WHERE Ciudad LIKE '%Es%';
```

La siguiente declaración SQL selecciona a todos los clientes con una Ciudad que acabe por «drid», para buscar Madrid, por ejemplo:

```
SELECT * FROM Clientes WHERE Ciudad LIKE '_drid';
```

La siguiente declaración SQL selecciona a todos los clientes con una Ciudad que comience con «m», «n» s «v»:

```
SELECT * FROM Clientes WHERE Ciudad LIKE '[mnsv]%';
```

La siguiente declaración SQL selecciona a todos los clientes con una Ciudad que comience con «a», «b» o «c»:

```
SELECT * FROM Clientes WHERE Ciudad LIKE '[a-c]%';
```

Sintaxis para utilizar SQL IN

```
SELECT nombre_columna(s) FROM nombre_tabla WHERE nombre_columna IN (valor1, valor2,
...);
```

La siguiente instrucción SQL selecciona todos los clientes que se encuentran en **“Mexico”, “Colombia”** y **“Argentina”**: :

```
SELECT nombre_columna(s) FROM nombre_tabla WHERE nombre_columna IN ('Mexico',
'Colombia', 'Argentina');
```

La siguiente instrucción SQL selecciona todos los clientes que **NO** se encuentran en **“Mexico”, “Colombia”** o **“Argentina”**:

```
SELECT * FROM Clientes WHERE Pais NOT IN ('Mexico', 'Colombia', 'Argentina');
```

La siguiente declaración SQL selecciona todos los clientes que son de los mismos países que los proveedores:

```
SELECT * FROM Clientes WHERE Pais IN (SELECT Pais FROM Proveedores);
```

SQL BETWEEN: Cómo usarlo en la condición

El operador BETWEEN en SQL selecciona valores dentro de un rango dado. Los valores pueden ser números, texto o fechas. El operador BETWEEN es inclusivo: se incluyen los valores de inicio y finalización, esto quiere decir que se seleccionan solamente los valores que están entre los valores que establezcamos en el operador between.

```
SELECT nombre_columna(s) FROM nombre_tabla WHERE nombre_columna BETWEEN valor1 AND valor2;
```

Ejemplo de BETWEEN en una condición

A continuación, mostramos como realizar una consulta SQL utilizando el operador BETWEEN dentro de la condición WHERE.

```
SELECT * FROM Productos WHERE Precio BETWEEN 5 AND 15;
```

Ejemplo NOT BETWEEN

Para mostrar los productos fuera del rango del ejemplo anterior, puedes utilizar el operador NOT BETWEEN, para seleccionar los valores que **NO** estén en ese rango:

```
SELECT * FROM Productos WHERE Precio NOT BETWEEN 5 AND 15;
```

Ejemplo BETWEEN con IN

A continuación se muestra una consulta **SQL**, combinando el operador BETWEEN y el operador IN dentro de la cláusula WHERE.

```
SELECT * FROM Productos WHERE (Precio BETWEEN 5 AND 15) AND NOT CategoriaID IN
(1,2,3);
```

Entregable:

Realice consultas aplicadas la base de datos proyectoscue.sql dónde aplique los operadores marcados con color **naranja**

Pegue un pantallazo y suba el archivo en pdf en la actividad Operadores SQL