GENESIS NAVAS/PPT.1421071/FICHA:3002085

Taller Práctico: Consultas SQL usando JOIN

Objetivo

Practicar el uso de sentencias INNER JOIN, LEFT JOIN, RIGHT JOIN y FULL JOIN para combinar información de múltiples tablas.

Estructura de la Base de Datos

Tabla: clientes

id_cliente	nombre_cliente	ciudad
1	Juan Gómez	Bogotá
2	Ana Torres	Medellín
3	Pedro Ríos	Cali

Tabla: pedidos

id_pedido	id_cliente	fecha
1	1	2024-06-01
2	2	2024-06-02
3	1	2024-06-03

Tabla: productos

id_producto	nombre_producto	precio
1	Laptop	3000
2	Mouse	50
3	Teclado	80

Tabla: detalle_pedido

id_detalle	id_pedido	id_producto	cantidad
1	1	1	1
2	1	2	2
3	2	3	1

INNER JOIN básico

```
SELECT c.nombre AS cliente, p.fecha FROM clientes c INNER JOIN pedidos p ON c.id_cliente = p.id_cliente;
```

Muestra el nombre del cliente y la fecha de sus pedidos.

JOIN con múltiples tablas

```
SELECT c.nombre AS cliente, pr.nombre AS producto, dp.cantidad FROM clientes c INNER JOIN pedidos p ON c.id_cliente = p.id_cliente INNER JOIN detalle_pedido dp ON p.id_pedido = dp.id_pedido INNER JOIN productos pr ON dp.id_producto = pr.id_producto;
```

Lista el nombre del cliente, nombre del producto y la cantidad comprada.

LEFT JOIN

```
SELECT c.nombre AS cliente, MAX(p.fecha) AS ultima_fecha_pedido FROM clientes c LEFT JOIN pedidos p ON c.id_cliente = p.id_cliente GROUP BY c.id_cliente, c.nombre;
```

Muestra todos los clientes y, si tienen pedidos, la fecha del último pedido.

RIGHT JOIN

```
SELECT pr.nombre AS producto, dp.cantidad FROM detalle_pedido dp RIGHT JOIN productos pr ON dp.id_producto = pr.id_producto;
```

Muestra todos los productos que han sido pedidos, incluyendo los que no tengan pedido aún.

FULL OUTER JOIN

```
SELECT c.nombre AS cliente, p.fecha FROM clientes c LEFT JOIN pedidos p ON c.id_cliente = p.id_cliente UNION SELECT c.nombre AS cliente, p.fecha FROM clientes c RIGHT JOIN pedidos p ON c.id_cliente = p.id_cliente;
```

Muestra todos los clientes y todos los pedidos, coincidan o no.

JOIN con condiciones

```
SELECT DISTINCT c.nombre FROM clientes c JOIN pedidos p ON c.id_cliente = p.id_cliente JOIN detalle_pedido dp ON p.id_pedido = dp.id_pedido GROUP BY c.nombre, p.id_pedido HAVING COUNT(dp.id_producto) > 1;
```

Muestra los clientes que han comprado más de un producto en un pedido.

JOIN con funciones agregadas

```
SELECT c.nombre AS cliente, SUM(dp.cantidad * pr.precio) AS total_gastado FROM clientes c JOIN pedidos p ON c.id_cliente = p.id_cliente JOIN detalle_pedido dp ON p.id_pedido = dp.id_pedido JOIN productos pr ON dp.id_producto = pr.id_producto GROUP BY c.nombre;
```

Muestra el total gastado por cada cliente.

Clientes sin pedidos

```
SELECT c.nombre FROM clientes c LEFT JOIN pedidos p ON c.id_cliente = p.id_cliente WHERE p.id_pedido IS NULL;
```

Lista los nombres de clientes que aún no han hecho ningún pedido.

Productos no vendidos

```
SELECT pr.nombre FROM productos pr LEFT JOIN detalle_pedido dp ON pr.id_producto = dp.id_producto WHERE dp.id_detalle IS NULL;
```

Muestra los productos que no han sido incluidos en ningún pedido.

JOIN con ORDER BY

```
SELECT p.id_pedido, c.nombre AS cliente, p.fecha FROM pedidos p JOIN clientes c ON p.id_cliente = c.id_cliente ORDER BY p.fecha DESC;
```

Lista los pedidos con nombre del cliente, fecha y ordenados por fecha descendente.