

MySQL

Docs_

Relacionar más de 2 tablas



Relacionar más de dos tablas con MySQL

Ejemplo 1

Supongamos que tenemos las siguientes tres tablas:

Tabla "clientes":

id_cliente (clave primaria)
nombre
direccion

Tabla "ordenes":

id_orden (clave primaria)
fecha
id_cliente (clave foránea que hace referencia a la tabla "clientes")

Tabla "productos":

id_producto (clave primaria)
nombre
precio

Tabla "ordenes_productos":

id_orden (clave foránea que hace referencia a la tabla "ordenes")
id_producto (clave foránea que hace referencia a la tabla "productos")
cantidad

Para obtener una lista de todas las órdenes de un cliente en particular, incluyendo los detalles de cada producto comprado en cada orden (con la cantidad correspondiente), podemos usar la siguiente consulta SQL:

```
SELECT c.nombre AS nombre_cliente, o.id_orden, o.fecha, p.nombre AS  
nombre_producto, p.precio, op.cantidad  
FROM clientes c  
JOIN ordenes o ON c.id_cliente = o.id_cliente  
JOIN ordenes_productos op ON o.id_orden = op.id_orden  
JOIN productos p ON op.id_producto = p.id_producto  
WHERE c.id_cliente = 123;
```

Relacionar más de dos tablas con MySQL

En esta consulta, estamos haciendo cuatro uniones (joins) para unir las cuatro tablas. La primera unión une la tabla "clientes" con la tabla "ordenes" usando el campo "id_cliente" como clave foránea. La segunda unión une la tabla "ordenes" con la tabla intermedia "ordenes_productos" que relaciona cada orden con los productos que se compraron en ella. La tercera unión une la tabla "ordenes_productos" con la tabla "productos" usando el campo "id_producto" como clave foránea. La cuarta unión se realiza para obtener el nombre del cliente y del producto, en lugar de solo el ID.

La consulta también incluye un filtro (WHERE) para limitar los resultados a un cliente en particular (en este caso, el cliente con el ID 123).

Cada registro en el resultado incluirá el nombre del cliente, el ID de la orden, la fecha de la orden, el nombre del producto, el precio del producto y la cantidad de productos comprados en esa orden.

Relacionar más de dos tablas con MySQL

Ejemplo 2

Tabla "clientes":

id_cliente (clave primaria)
nombre
email
telefono

Tabla "pedidos":

id_pedido (clave primaria)
fecha
id_cliente (clave foránea que hace referencia a la tabla "clientes")

Tabla "detalle_pedido":

id_detalle (clave primaria)
id_pedido (clave foránea que hace referencia a la tabla "pedidos")
id_producto (clave foránea que hace referencia a la tabla "productos")
cantidad
precio_unitario

Tabla "productos":

id_producto (clave primaria)
nombre
precio

Para obtener una lista de todos los pedidos de un cliente en particular, incluyendo el detalle de cada producto comprado en cada pedido (con el total correspondiente), podemos usar la siguiente consulta SQL:

```
SELECT c.nombre AS nombre_cliente, p.fecha, prod.nombre AS nombre_producto,  
dp.cantidad, prod.precio, dp.cantidad * prod.precio AS total  
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 prod ON dp.id_producto = prod.id_producto  
WHERE c.id_cliente = 123;
```

Relacionar más de dos tablas con MySQL

En esta consulta, estamos haciendo cuatro uniones (joins) para unir las cuatro tablas. La primera unión une la tabla "clientes" con la tabla "pedidos" usando el campo "id_cliente" como clave foránea. La segunda unión une la tabla "pedidos" con la tabla "detalle_pedido" que relaciona cada pedido con los productos que se compraron en él. La tercera unión une la tabla "detalle_pedido" con la tabla "productos" usando el campo "id_producto" como clave foránea. La cuarta unión se realiza para obtener el nombre del cliente y del producto, en lugar de solo el ID.

La consulta también incluye un filtro (WHERE) para limitar los resultados a un cliente en particular (en este caso, el cliente con el ID 123).

Cada registro en el resultado incluirá el nombre del cliente, la fecha del pedido, el nombre del producto, la cantidad de productos comprados en ese pedido, el precio unitario del producto y el total correspondiente a la cantidad multiplicada por el precio unitario.

Relacionar más de dos tablas con MySQL

Ejemplo 3

Supongamos que tenemos las siguientes tres tablas:

Tabla "personas":

id_persona (clave primaria)
nombre
apellido
fecha_nacimiento
direccion
telefono
correo_electronico

Tabla "experiencia_laboral":

id_experiencia (clave primaria)
empresa
puesto
fecha_inicio
fecha_fin
id_persona (clave foránea que hace referencia a la tabla "personas")

Tabla "educacion":

id_educacion (clave primaria)
institucion
titulo
fecha_inicio
fecha_fin
id_persona (clave foránea que hace referencia a la tabla "personas")

Para obtener una lista de la experiencia laboral y la educación de una persona en particular, podemos usar la siguiente consulta SQL:

```
SELECT p.nombre, p.apellido, p.fecha_nacimiento, el.empresa, el.puesto,  
el.fecha_inicio AS inicio_experiencia, el.fecha_fin AS fin_experiencia,  
ed.institucion, ed.titulo, ed.fecha_inicio AS inicio_educacion, ed.fecha_fin  
AS fin_educacion  
FROM personas p  
LEFT JOIN experiencia_laboral el ON p.id_persona = el.id_persona  
LEFT JOIN educacion ed ON p.id_persona = ed.id_persona  
WHERE p.id_persona = 123;
```

Relacionar más de dos tablas con MySQL

En esta consulta, estamos haciendo dos uniones izquierdas (left joins) para unir las tres tablas. La primera unión izquierda une la tabla "personas" con la tabla "experiencia_laboral" usando el campo "id_persona" como clave foránea. La segunda unión izquierda une la tabla "personas" con la tabla "educacion" usando el campo "id_persona" como clave foránea.

La consulta también incluye un filtro (WHERE) para limitar los resultados a una persona en particular (en este caso, la persona con el ID 123).

Cada registro en el resultado incluirá el nombre, apellido y fecha de nacimiento de la persona, junto con la información de la experiencia laboral (empresa, puesto, fecha de inicio y fecha de fin) y la educación (institución, título, fecha de inicio y fecha de fin). Si la persona no tiene experiencia laboral o educación registrada, los campos correspondientes aparecerán vacíos.