**Consulta de un tabla completa**

**SELECT \* from materiales**

Imagen que contiene Tabla

Descripción generada automáticamente

**Selección**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Proyección**

Imagen que contiene Calendario

Descripción generada automáticamente

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?

El numero de tuplas incrementa porque estamos mesclando 2 tablas

**Construcción de consultas a partir de una especificación**

Tabla

Descripción generada automáticamente

Si algún material no ha se ha entregado ¿Aparecería en el resultado de esta consulta?

No, ya que no habrá match entre clave de materiales y clave de entrega

**Reunión con criterio específico**

Tabla

Descripción generada automáticamente

**Unión (se ilustra junto con selección)**

Interfaz de usuario gráfica

Descripción generada automáticamente

¿Cuál sería una consulta que obtuviera el mismo resultado sin usar el operador Unión? Compruébalo.

Select \* From entregan

Where clave IN (1450, 1300);

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Intersección (se ilustra junto con selección y proyección)**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Producto cartesiano**

**Tabla

Descripción generada automáticamente**

**¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?**

El número de tuplas es el producto cartesiano: el número de tuplas de entregan multiplicado por el número de tuplas de materiales.

**Construcción de consultas a partir de una especificación.**

SELECT m.descripcion

FROM entregan e

JOIN materiales m ON e.material\_id = m.id

WHERE TO\_CHAR(e.fecha\_entrega, 'YYYY') = '2000';

**¿Por qué aparecen varias veces algunas descripciones de material?**

Las descripciones de material pueden aparecer varias veces en el resultado por varias razones comunes:

1. Múltiples entregas del mismo material: Si el mismo material fue entregado en varias ocasiones en el año 2000, habrá múltiples filas en la tabla entregan que corresponden al mismo material. Cada fila de entregan se une a su correspondiente descripción en materiales, lo que resulta en descripciones repetidas.
2. Producto cartesiano: Si la relación entre las tablas no está definida correctamente, o si la clave de unión no es única (por ejemplo, si material\_id no es una clave única en la tabla entregan), puede generarse un producto cartesiano parcial que repita descripciones.
3. Falta de uso de DISTINCT: Si no deseas mostrar descripciones repetidas, puedes utilizar DISTINCT para eliminar duplicados.

1. Uso del calificador distinct

En el resultado anterior, observamos que una misma descripción de material aparece varias veces.  
Agrega la palabra DISTINCT inmediatamente después de la palabra SELECT a la consulta que planteaste antes.

Consulta modificada:

SELECT DISTINCT m.descripcion  
FROM entregan e  
JOIN materiales m ON e.material\_id = m.id  
WHERE TO\_CHAR(e.fecha\_entrega, 'YYYY') = '2000';

Resultado obtenido: El uso de DISTINCT elimina las descripciones duplicadas, mostrando cada descripción de material entregado en el año 2000 una sola vez.

2. Ordenamientos

Consulta para obtener los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto (descendente):

SELECT proyecto\_numero, denominacion, fecha\_entrega, cantidad\_entrega  
FROM entregan  
ORDER BY proyecto\_numero DESC, fecha\_entrega DESC;

3. Uso de expresiones

Ejemplo de uso de expresiones aritméticas en una consulta SQL:

SELECT cantidad \* precio\_unitario AS total\_precio  
FROM materiales;

4. Operadores de cadena

Consulta utilizando el operador LIKE y comodines (%):

SELECT \* FROM productos WHERE descripcion LIKE 'Si%';  
Resultado: Se devuelven todos los productos cuya descripción comienza con 'Si'.

El símbolo '%' se utiliza como comodín para representar cualquier número de caracteres.

Si la consulta fuera 'LIKE Si', se devolverían solo los productos cuya descripción es exactamente 'Si', sin más caracteres.

5. Operadores compuestos

Ejemplo de uso de operadores compuestos en SQL:

UPDATE materiales SET cantidad += 10 WHERE material\_id = 1;

6. Operadores lógicos

Ejemplo de uso de operadores lógicos ALL y ANY:

SELECT \* FROM proyectos WHERE proyecto\_id > ALL (SELECT proyecto\_id FROM entregas);

7. BETWEEN

Ejemplo de uso de BETWEEN para filtrar por fechas:

SELECT Clave, RFC, Numero, Fecha, Cantidad  
FROM Entregan  
WHERE Fecha BETWEEN '2000-01-01' AND '2000-12-31';

8. EXISTS

Consulta utilizando EXISTS para verificar la existencia de proveedores:

SELECT RFC, Cantidad, Fecha, Numero  
FROM Entregan  
WHERE Numero BETWEEN 5000 AND 5010  
AND EXISTS (SELECT RFC FROM Proveedores WHERE RazonSocial LIKE 'La%' AND Entregan.RFC = Proveedores.RFC);

9. Operador IN

Consulta equivalente a la anterior utilizando IN en lugar de EXISTS:

SELECT RFC, Cantidad, Fecha, Numero  
FROM Entregan  
WHERE Numero BETWEEN 5000 AND 5010  
AND RFC IN (SELECT RFC FROM Proveedores WHERE RazonSocial LIKE 'La%');