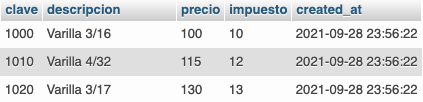
**Consulta de una Tabla Completa**

**Algebra relacional**  
materiales  
  
**SQL**  
SELECT \*

FROM materiales



Renglones: 45

**Selección**

**Algebra relacional**  
SL{clave=1000}(materiales)  
  
**SQL**  
SELECT \*

FROM materiales

WHERE clave=1000

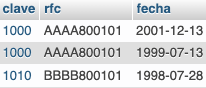


Renglones: 1

**Proyección**

**Algebra relacional**  
PR{clave,rfc,fecha} (entregan)  
  
**SQL**  
SELECT clave, rfc, fecha

FROM entregan



Renglones: 87

**Reunión Natural**

**Algebra relacional**  
entregan JN materiales  
  
**SQL**  
SELECT \*

FROM materiales, entregan

WHERE materiales.clave = entregan.clave



Renglones: 45

**Reunión con criterio específico**

**Algebra relacional**  
entregan JN{entregan.numero <= proyectos.numero} proyectos  
  
**SQL**  
SELECT \* from entregan, proyectos

WHERE entregan.numero <= proyectos.numero



Renglones: 45

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

**Algebra relacional**  
SL{clave=1450}(entregan) UN SL{clave=1300}(entregan)  
  
**SQL**  
(SELECT \* FROM entregan WHERE clave=1450)

union

(SELECT \* FROM entregan WHERE clave=1300)



Renglones: 2

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

**Algebra relacional**  
PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan))  
  
**SQL**

**Nota:** Debido a que en SQL server no tiene definida alguna palabra reservada que nos permita hacer esto de una manera entendible, veremos esta sección en el siguiente laboratorio con el uso de Subconsultas. Un ejemplo de un DBMS que si tiene la implementación de una palabra reservada para esta función es Oracle, en él si se podría generar la consulta con una sintaxis como la siguiente:

(SELECT clave FROM entregan WHERE numero=5001)

intersect

(SELECT clave FROM entregan WHERE numero=5018)

**Con Subconsulta**

SELECT clave

FROM entregan

WHERE numero = 5001 IN (SELECT clave FROM entregan WHERE numero = 5018)

**Diferencia (se ilustra con selección)**

**Algebra relacional**  
entregan - SL{clave=1000}(entregan)  
  
**SQL**  
(SELECT \* FROM entregan)

minus

(SELECT \* FROM entregan WHERE clave=1000)

Nuevamente, "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.

SELECT \*

FROM entregan

WHERE clave != 1000



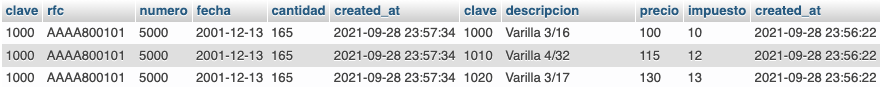
Renglones: 85

**Producto Cartesiano**

**Algebra relacional**

entregan X materiales  
  
**SQL**  
SELECT \*

FROM entregan, materiales



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

Renglones: 3915

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

Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

Recuerda que la fecha puede indicarse como '01-JAN-2000' o '01/01/00'.

Importante: Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa que vas a trabajar con ese formato de fechas.

**SQL**  
SELECT M.descripcion, DATE\_FORMAT(fecha, '%e/%c/%Y') 'Fecha'

-- FORMATO DD/MM/YYYY

FROM entregan E, materiales M

WHERE fecha >= '2000-01-01' AND fecha <= '2000-12-31' AND (M.clave = E.clave)



¿Por qué aparecen varias veces algunas descripciones de material?

Por que el mismo material fue entregado en fechas distintas

Renglones: 12

**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.

¿Qué resultado obtienes en esta ocasión?

Todos los materiales entregados entre esas fechas sin repeticiones  
  
**SQL**  
SELECT DISTINCT M.descripcion

-- FORMATO DD/MM/YYYY

FROM entregan E, materiales M

WHERE fecha >= '2000-01-01' AND fecha <= '2000-12-31' AND (M.clave = E.clave)



Renglones: 10

**Ordenamientos**

Si al final de una sentencia select se agrega la cláusula order by campo [desc] [,campo [desc] ...]

donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.

Obtén los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto, presentando las fechas de la más reciente a la más antigua.  
  
**SQL**  
SELECT P.numero, denominacion, fecha, cantidad

FROM proyectos P, entregan E

WHERE (P.numero = E.numero)

ORDER BY P.numero, fecha DESC



Renglones: 87

**Uso de Expresiones**

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo en una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

+ Suma

- Resta

\* Producto

/ División

Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre" ). Para SQL Server también pueden utilizarse comillas simples.

**Operadores de cadena**

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.  
  
**SQL**  
SELECT \*

FROM materiales

WHERE descripcion LIKE 'Si%'



Renglones: 2

¿Qué resultado obtienes?

Todas las descripciones de materiales que comienzan con *Si*

Explica que hace el símbolo '%'.

Permite que con que coincida la parte indicada te arroje el resultado, sin importar que haya o no más caracteres después.

¿Qué sucede si la consulta fuera : LIKE 'Si' ?

Únicamente arrojaría los resultados cuya descripción sea *Si,* si comienzan de esa manera, pero tienen más letras no aparecen en el resultado.

¿Qué resultado obtienes?



Una tabla vacía

Explica a qué se debe este comportamiento.

Ya que la descripción de ningún material es como tal ‘Si’, regresa una tabla vacía.

Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.

Su sintaxis es : Expresión + Expresión.

Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:

SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;

¿Qué resultado obtienes de ejecutar el siguiente código?

¿Para qué sirve DECLARE?

¿Cuál es la función de @foo?

¿Que realiza el operador SET?

Sin embargo, tenemos otros operadores como [ ] , [^] y \_.

[ ] - Busca coincidencia dentro de un intervalo o conjunto dado. Estos caracteres se pueden utilizar para buscar coincidencias de patrones como sucede con LIKE.

[^] - En contra parte, este operador coincide con cualquier caracter que no se encuentre dentro del intervalo o del conjunto especificado.

\_ - El operador \_ o guion bajo, se utiliza para coincidir con un caracter de una comparación de cadenas.

Ahora explica el comportamiento, función y resultado de cada una de las siguientes consultas:

**SQL**

SELECT rfc

FROM entregan

WHERE rfc LIKE '[A-D]%';

**SQL**  
SELECT rfc

FROM entregan

WHERE rfc LIKE '[^A]%';

**SQL**

SELECT Numero

FROM entregan

WHERE Numero LIKE '\_\_\_6';

Regresa todos los números de materiales entregados que contengan un 6



**Operadores compuestos**

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

\* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

**Operadores Lógicos**

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.  
  
**SQL**  
SELECT \*

FROM materiales

WHERE descripcion LIKE 'Si%'



Renglones: 2