

ITESM Tecnológico de Monterrey Campus Querétaro-



Tecnológico de Monterrey

Laboratorio 20: Consultas en SQL

-Construcción de software y toma de decisiones (Gpo 501)-

DOCENTE:

Alejandro Fernández Vilchis
Enrique Alfonso Calderón Balderas
Denisse L. Maldonado Flores

ALUMNO

Mónica Monserrat Martínez Vásquez | [A01710965](#)
Imanol Muñiz Ramirez | [A01701713](#)
Juan Antonio Landeros Velázquez | [A00574000](#)
Armando Méndez Castro | [A01277796](#)

Consulta de un tabla completa

Algebra relacional.
materiales

SQL

select * from materiales

	123 Clave ▼	ABC Descripción ▼	123 Precio ▼
1	1,000	Varilla 3/16	100
2	1,010	Varilla 4/32	115
3	1,020	Varilla 3/17	130
4	1,030	Varilla 4/33	145
5	1,040	Varilla 3/18	160
6	1,050	Varilla 4/34	175
7	1,060	Varilla 3/19	190
8	1,070	Varilla 4/35	205
9	1,080	Ladrillos rojos	50
10	1,090	Ladrillos grises	35
11	1,100	Block	30
12	1,110	Megablock	40
13	1,120	Sillar rosa	100
14	1,130	Sillar gris	110
15	1,140	Cantera blanca	200
16	1,150	Cantera gris	1,210
17	1,160	Cantera rosa	1,420
18	1,170	Cantera amarilla	230
19	1,180	Recubrimiento P100'	200
20	1,190	Recubrimiento P1010	220
21	1,200	Recubrimiento P1015	240
22	1,210	Recubrimiento P1025	250
23	1,220	Recubrimiento P1035	280
24	1,230	Cemento	300
25	1,240	A...	300

Selección

Algebra relacional.

$SL\{clave=1000\}(materiales)$

SQL

```
select * from materiales
```

```
where clave=1000
```

	123 Clave ▼	ABC Descripción ▼	123 Precio ▼
1	1,000	Varilla 3/16	100

Proyección

Algebra relacional.

$PR\{clave, rfc, fecha\}(entregan)$

SQL

```
select clave, rfc, fecha from entregan
```

	123 clave	ABC rfc	📅 fecha
1	1,270	DDDD800101	1997-03-09
2	1,090	BBBB800101	1998-03-01
3	1,170	BBBB800101	1998-04-02
4	1,400	AAAA800101	1998-05-06
5	1,090	BBBB800101	1998-06-06
6	1,340	CCCC800101	1998-06-11
7	1,140	GGGG800101	1998-07-02
8	1,420	CCCC800101	1998-07-04
9	1,290	FFFF800101	1998-08-01
10	1,250	BBBB800101	1998-08-05
11	1,000	AAAA800101	1998-08-07
12	1,430	DDDD800101	1998-09-01
13	1,030	DDDD800101	1998-09-04
14	1,120	EEEE800101	1998-12-03
15	1,330	BBBB800101	1998-12-06
16	1,190	DDDD800101	1998-12-09
17	1,330	BBBB800101	1998-12-10
18	1,350	DDDD800101	1999-02-08
19	1,430	DDDD800101	1999-02-09
20	1,070	HHHH800101	1999-03-12
21	1,020	CCCC800101	1999-04-02
22	1,150	HHHH800101	1999-04-03
23	1,050	EEEE800101	1999-04-06

Reunión Natural

Algebra relacional.
entregan JN materiales

SQL
select * from materiales,entregan
where materiales.clave = entregan.clave

	Clave	Descripción	Precio	Clave	RFC	Número	Fecha	Cantidad
1	1,270	Tezontle	80	1,270	DDDD800101	5,008	1997-03-09	546
2	1,090	Ladrillos grises	35	1,090	BBBB800101	5,010	1998-03-01	421
3	1,170	Cantera amarilla	230	1,170	BBBB800101	5,017	1998-04-02	180
4	1,400	Pintura C1011	125	1,400	AAAA800101	5,010	1998-05-06	116
5	1,090	Ladrillos grises	35	1,090	BBBB800101	5,010	1998-06-06	612
6	1,340	Tubería 4.5	250	1,340	CCCC800101	5,016	1998-06-11	674
7	1,140	Cantera blanca	200	1,140	GGGG800101	5,005	1998-07-02	651
8	1,420	Pintura C1012	125	1,420	CCCC800101	5,002	1998-07-04	603
9	1,290	Tubería 3.5	200	1,290	FFFF800101	5,009	1998-08-01	132
10	1,250	Grava	100	1,250	BBBB800101	5,010	1998-08-05	690
11	1,000	Varilla 3/16	100	1,000	AAAA800101	5,000	1998-08-07	165
12	1,430	Pintura B1022	125	1,430	DDDD800101	5,007	1998-09-01	13
13	1,030	Varilla 4/33	145	1,030	DDDD800101	5,016	1998-09-04	139
14	1,120	Sillar rosa	100	1,120	EEEE800101	5,007	1998-12-03	167
15	1,330	Tubería 3.7	240	1,330	BBBB800101	5,017	1998-12-06	558
16	1,190	Recubrimiento P1010	220	1,190	DDDD800101	5,019	1998-12-09	94
17	1,330	Tubería 3.7	240	1,330	BBBB800101	5,013	1998-12-10	554
18	1,350	Tubería 3.8	260	1,350	DDDD800101	5,015	1999-02-08	261
19	1,430	Pintura B1022	125	1,430	DDDD800101	5,003	1999-02-09	576
20	1,070	Varilla 4/35	205	1,070	HHHH800101	5,012	1999-03-12	503
21	1,020	Varilla 3/17	130	1,020	CCCC800101	5,017	1999-04-02	8
22	1,150	Cantera gris	1,210	1,150	HHHH800101	5,015	1999-04-03	458
23	1,050	Varilla 4/34	175	1,050	FFFF800101	5,014	1999-04-06	90
24	1,350	Tubería 3.8	260	1,350	DDDD800101	5,015	1999-05-06	330
25	1,340	Tubería 4.5	250	1,340	FFFF800101	5,004	1999-05-14	42

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

- No, porque el join se hace en el atributo clave y sólo se muestran los que coinciden porque aparecen en las dos tablas, por lo tanto si no se ha entregado, no saldrá su clave en la tabla entregan.

Reunión con criterio específico

Algebra relacional.

entregan JN{entregan.numero <= proyectos.numero} proyectos

SQL

select * from entregan,proyectos

where entregan.numero < = proyectos.numero

	123 Clave	ABC RFC	123 Número	Fecha	123 Cantidad	123 Número	ABC Denominación
1	1,000	AAAA800101	5,000	1998-08-07	165	5,000	Vamos Mexico
2	1,200	EEEE800101	5,000	2000-05-03	177	5,000	Vamos Mexico
3	1,400	AAAA800101	5,000	2002-12-03	382	5,000	Vamos Mexico
4	1,000	AAAA800101	5,000	1998-08-07	165	5,001	Aztecón
5	1,200	EEEE800101	5,000	2000-05-03	177	5,001	Aztecón
6	1,400	AAAA800101	5,000	2002-12-03	382	5,001	Aztecón
7	1,210	FFFF800101	5,001	1999-05-11	43	5,001	Aztecón
8	1,010	BBBB800101	5,001	2000-03-05	528	5,001	Aztecón
9	1,410	BBBB800101	5,001	2000-05-02	601	5,001	Aztecón
10	1,000	AAAA800101	5,000	1998-08-07	165	5,002	CIT Campeche
11	1,200	EEEE800101	5,000	2000-05-03	177	5,002	CIT Campeche
12	1,400	AAAA800101	5,000	2002-12-03	382	5,002	CIT Campeche
13	1,210	FFFF800101	5,001	1999-05-11	43	5,002	CIT Campeche
14	1,010	BBBB800101	5,001	2000-03-05	528	5,002	CIT Campeche
15	1,410	BBBB800101	5,001	2000-05-02	601	5,002	CIT Campeche
16	1,420	CCCC800101	5,002	1998-07-04	603	5,002	CIT Campeche
17	1,220	GGGG800101	5,002	2003-01-02	24	5,002	CIT Campeche
18	1,020	CCCC800101	5,002	2003-05-07	582	5,002	CIT Campeche
19	1,000	AAAA800101	5,000	1998-08-07	165	5,003	Mexico sin ti no estamos c
20	1,200	EEEE800101	5,000	2000-05-03	177	5,003	Mexico sin ti no estamos c
21	1,400	AAAA800101	5,000	2002-12-03	382	5,003	Mexico sin ti no estamos c
22	1,210	FFFF800101	5,001	1999-05-11	43	5,003	Mexico sin ti no estamos c
23	1,010	BBBB800101	5,001	2000-03-05	528	5,003	Mexico sin ti no estamos c
24	1,410	BBBB800101	5,001	2000-05-02	601	5,003	Mexico sin ti no estamos c

Unión (se ilustra junto con selección)

Algebra relacional.

$SL\{clave=1450\}(entregan) \cup SL\{clave=1300\}(entregan)$

SQL

(select * from entregan where clave=1450)

union

(select * from entregan where clave=1300)

	123 Clave	ABC RFC	123 Número	Fecha	123 Cantidad
1	1,300	GGGG800101	5,005	2002-10-06	521
2	1,300	GGGG800101	5,005	2003-02-02	457
3	1,300	GGGG800101	5,010	2003-08-01	119

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

```
select *
from entregan e
where clave = 1450 or clave = 1300
```

Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

$PR\{clave\}(SL\{numero=5001\}(entregan)) \cap PR\{clave\}(SL\{numero=5018\}(entregan))$

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:

$(select\ clave\ from\ entregan\ where\ numero=5001)$
 $intersect$
 $(select\ clave\ from\ entregan\ where\ numero=5018)$



	clave
1	1,010

Diferencia (se ilustra con selección)

Algebra relacional.

$entregan - SL\{clave=1000\}(entregan)$

SQL

$(select\ * \ from\ entregan)$
 $minus$
 $(select\ * \ from\ entregan\ where\ clave=1000)$

	123 Clave ▼	ABC RFC ▼	123 Número ▼	🕒 Fecha ▼	123 Cantidad ▼
1	1,270	DDDD800101	5,008	1997-03-09	546
2	1,090	BBBB800101	5,010	1998-03-01	421
3	1,170	BBBB800101	5,017	1998-04-02	180
4	1,400	AAAA800101	5,010	1998-05-06	116
5	1,090	BBBB800101	5,010	1998-06-06	612
6	1,340	CCCC800101	5,016	1998-06-11	674
7	1,140	GGGG800101	5,005	1998-07-02	651
8	1,420	CCCC800101	5,002	1998-07-04	603
9	1,290	FFFF800101	5,009	1998-08-01	132
10	1,250	BBBB800101	5,010	1998-08-05	690
11	1,430	DDDD800101	5,007	1998-09-01	13
12	1,030	DDDD800101	5,016	1998-09-04	139
13	1,120	EEEE800101	5,007	1998-12-03	167
14	1,330	BBBB800101	5,017	1998-12-06	558
15	1,190	DDDD800101	5,019	1998-12-09	94
16	1,330	BBBB800101	5,013	1998-12-10	554
17	1,350	DDDD800101	5,015	1999-02-08	261
18	1,430	DDDD800101	5,003	1999-02-09	576
19	1,070	HHHH800101	5,012	1999-03-12	503
20	1,020	CCCC800101	5,017	1999-04-02	8
21	1,150	HHHH800101	5,015	1999-04-03	458
22	1,050	FFFF800101	5,014	1999-04-06	90
23	1,350	DDDD800101	5,015	1999-05-06	330
24	1,210	FFFF800101	5,001	1999-05-11	43

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

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? se multiplica el numero de registros de cada columna, en este caso entregan contiene 121 y materiales contiene 44, por lo tanto

$$121 \times 44 = 5324$$

Construcción de consultas a partir de una especificación

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

```
select m.Descripción
from materiales m
join entregan e on m.Clave = e.Clave
where year(e.Fecha) = 2000
```

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

	DESC Descripción
1	Varilla 4/35
2	Cantera rosa
3	Ladrillos grises
4	Varilla 3/19
5	Pintura C1012
6	Tepetate
7	Varilla 4/32
8	Tubería 3.6
9	Recubrimiento P1010
10	Varilla 3/19
11	Pintura B1021
12	Recubrimiento P1019
13	Varilla 4/33
14	Varilla 3/16
15	Recubrimiento P1019
16	Block
17	Tubería 4.4
18	Pintura C1010
19	Pintura B1020
20	Megablock
21	Varilla 3/18

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.

¿Por qué aparecen varias veces algunas descripciones de material?

- Porque se pueden haber vendido los mismos productos durante el año

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.

	asc Descripción ▼
1	Varilla 4/35
2	Cantera rosa
3	Ladrillos grises
4	Varilla 3/19
5	Pintura C1012
6	Tepetate
7	Varilla 4/32
8	Tubería 3.6
9	Recubrimiento P1010
10	Pintura B1021
11	Recubrimiento P1019
12	Varilla 4/33
13	Varilla 3/16
14	Block
15	Tubería 4.4
16	Pintura C1010
17	Pintura B1020
18	Megablock
19	Varilla 3/18
20	Varilla 4/34
21	Tubería 3.7
22	Arena

¿Qué resultado obtienes en esta ocasión?

- Devuelve una tabla sin repetir varias veces el material

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.

```
select p.número, p.denominación, e.fecha, e.cantidad
from proyectos p
join entregan e on p.Número = e.Número
order by p.Número and Fecha
```

	123 número	ABC denominación	📅 fecha	123 cantidad
1	5,008	Televisa en acción	1997-03-09	546
2	5,017	Reparación de la carretera c	1998-04-02	180
3	5,010	Construcción de Hospital In	1998-06-06	612
4	5,005	Infonavit Durango	1998-07-02	651
5	5,009	Disco Atlantic	1998-08-01	132
6	5,000	Vamos Mexico	1998-08-07	165
7	5,016	Ampliación de la carretera a	1998-09-04	139
8	5,017	Reparación de la carretera c	1998-12-06	558
9	5,013	Reparación de la plaza Son	1998-12-10	554
10	5,003	Mexico sin ti no estamos co	1999-02-09	576
11	5,017	Reparación de la carretera c	1999-04-02	8
12	5,014	Remodelación de Soriana	1999-04-06	90
13	5,001	Aztecón	1999-05-11	43
14	5,006	Reconstrucción del templo c	1999-07-01	336
15	5,012	Restauración de instalacion	1999-08-03	115
16	5,003	Mexico sin ti no estamos co	1999-09-02	202
17	5,019	Queretaro limpio	1999-09-06	244
18	5,016	Ampliación de la carretera a	1999-10-02	11
19	5,006	Reconstrucción del templo c	1999-10-05	460
20	5,018	Tu cambio por la educacion	1999-11-12	53
21	5,004	Educando en Coahuila	1999-12-11	263
22	5,016	Ampliación de la carretera a	2000-01-06	162
23	5,013	Reparación de la plaza Son	2000-02-01	692
24	5,007	Construcción de plaza Magr	2000-03-02	331
25	5,019	Queretaro limpio	2000-03-08	463
26	5,006	Reconstrucción del templo c	2000-04-05	224

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.

SELECT * FROM productos where Descripcion LIKE 'Si%'

¿Qué resultado obtienes?

	123 Clave ▼	ABC Descripción ▼	123 Precio ▼
1	1,120	Sillar rosa	100
2	1,130	Sillar gris	110

Explica que hace el símbolo '%'.
- El % después de "Si" indica que cualquier secuencia de caracteres (incluso una secuencia vacía) puede seguir después de "Si".

¿Qué sucede si la consulta fuera : LIKE 'Si' ?
- No da ningún resultado porque sin el % significa que busca materiales que tengan descripcion 'Si'

¿Qué resultado obtienes?
- Vacío

Explica a qué se debe este comportamiento.
- No hay materiales con descripción 'Si'

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;

```
DECLARE @foo varchar(40);
```

```
DECLARE @bar varchar(40);
```

```
SET @foo = '¿Que resultado?';
```

```
SET @bar = ' ¿???'
```

```
SET @foo += ' obtienes?';
```

```
PRINT @foo + @bar;
```

¿Qué resultado obtienes de ejecutar el siguiente código?

¿Para qué sirve DECLARE?

- indica que se pueden especificar y asignar valores a varias variables

¿Cuál es la función de @foo?

- ¿Qué resultado?

¿Que realiza el operador SET?

- Asignar

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:

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

- no regresa nada porque no corresponden a la expresión regular

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
```

- no regresa nada porque no corresponden a la expresión regular

```
SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
```

- numeros que tengan el 6 como cuarta cifra

Operadores compuestos.

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

Operadores Lógicos.

Los operadores lógicos comprueban la verdad de una condición, al igual que los operadores de comparación, devuelven un tipo de dato booleano (True, false o unknown).

ALL Es un operador que compara un valor numérico con un conjunto de valores representados por un subquery. La condición es verdadera cuando todo el conjunto cumple la condición.

ANY o SOME Es un operador que compara un valor numérico con un conjunto de valores. La condición es verdadera cuando al menos un dato del conjunto cumple la condición.

La sintaxis para ambos es: valor_numerico {operador de comparación} subquery

BETWEEN Es un operador para especificar intervalos. Una aplicación muy común de dicho operador son intervalos de fechas.

```
SELECT Clave,RFC,Numero,Fecha,Cantidad
FROM Entregan
WHERE Numero Between 5000 and 5010;
```

¿Cómo filtrarías rangos de fechas?

WHERE, BETWEEN

EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.

```
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] )
```

¿Qué hace la consulta?

- Se seleccionan las entregas (de la tabla [Entregan]) cuyo Numero está entre 5000 y 5010 y que fueron realizadas por proveedores cuya RazonSocial comienza con "La".

¿Qué función tiene el paréntesis () después de EXISTS?

- abrir una subconsulta

IN Especifica si un valor dado tiene coincidencias con algún valor de una subconsulta.

NOTA: Se utiliza dentro del WHERE pero debe contener un parametro. Ejemplo: Where proyecto.id IN Lista_de_Proyectos_Subquery

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN

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

NOT Simplemente niega la entrada de un valor booleano.

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

```
SELECT RFC, Cantidad, Fecha, Numero
FROM [Entregan]
WHERE [Numero] BETWEEN 5000 AND 5010
AND [RFC] NOT IN (
    SELECT [RFC]
    FROM [Proveedores]
    WHERE RazonSocial LIKE 'La%'
)
```

El Operador TOP, es un operador que recorre la entrada, un query, y sólo devuelve el primer número o porcentaje específico de filas basado en un criterio de ordenación si es posible.

¿Qué hace la siguiente sentencia? Explica por qué.

- Realiza una selección de las dos primeras filas de la tabla

```
SELECT TOP 2 * FROM Proyectos
```

¿Qué sucede con la siguiente consulta? Explica por qué.

- Lo mismo que la anterior pero utilizando Numero como una variable declarada previamente.

```
SELECT TOP Numero FROM Proyectos
```

Modificando la estructura de un tabla existente.

Agrega a la tabla materiales la columna PorcentajeImpuesto con la instrucción:

```
ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);
```

A fin de que los materiales tengan un impuesto, les asignaremos impuestos ficticios basados en sus claves con la instrucción:

```
UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;
```

esto es, a cada material se le asignará un impuesto igual al doble de su clave dividida entre diez.

Revisa la tabla de materiales para que compruebes lo que hicimos anteriormente.

¿Qué consulta usarías para obtener el importe de las entregas es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega y el precio del material y el impuesto asignado?

```
SELECT
    e.cantidad * m.precio * (1 + m.PorcentajeImpuesto / 100) AS ImporteTotal
FROM
    entregas e
JOIN
    materiales m ON e.clave_material = m.clave;
```

Creación de vistas

La sentencia:

Create view nombrevista (nombrecolumna1 , nombrecolumna2 ,..., nombrecolumna3)
as select...

Permite definir una vista. Una vista puede pensarse como una consulta etiquetada con un nombre, ya que en realidad al referirnos a una vista el DBMS realmente ejecuta la consulta asociada a ella, pero por la cerradura del álgebra relacional, una consulta puede ser vista como una nueva relación o tabla, por lo que es perfectamente válido emitir la sentencia:

```
select * from nombrevista
```

¡Como si nombrevista fuera una tabla!

Comprueba lo anterior, creando vistas para cinco de las consultas que planteaste anteriormente en la práctica . Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

La parte (nombrecolumna1,nombrecolumna2,.de la sentencia create view puede ser omitida si no hay ambigüedad en los nombres de las columnas de la sentencia select asociada.

Importante: Las vistas no pueden incluir la cláusula order by.

A continuación se te dan muchos enunciados de los cuales deberás generar su correspondiente consulta.

En el reporte incluye la sentencia, una muestra de la salida (dos o tres renglones) y el número de renglones que SQL Server reporta al final de la consulta.

Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos".

Los materiales (clave y descripción) que han sido proporcionados por el proveedor "Acme tools".

El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales.

El Total entregado por cada material en el año 2000.

La Clave del material más vendido durante el 2001. (se recomienda usar una vista intermedia para su solución)

Productos que contienen el patrón 'ub' en su nombre.

Denominación y suma del total a pagar para todos los proyectos.

Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Solo usando vistas).

Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Sin usar vistas, utiliza not in, in o exists).

Costo de los materiales y los Materiales que son entregados al proyecto Televisa en acción cuyos proveedores también suministran materiales al proyecto Educando en Coahuila.

Reto: Usa solo el operador NOT IN en la consulta anterior (No es parte de la entrega).

Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos.

```
CREATE VIEW MaterialesProyectoMexico AS
SELECT m.clave, m.descripcion
FROM materiales m
JOIN entregan e ON m.clave = e.clave
JOIN proyectos p ON e.numero = p.numero
WHERE p.denominacion = 'México sin ti no estamos completos';
CREATE VIEW MaterialesProveedorAcme AS
SELECT m.clave, m.descripcion
FROM materiales m
JOIN proveedores pr ON m.clave = pr.rfc
WHERE pr.razonsocial = 'Acme tools';
CREATE VIEW RFCProveedores2000 AS
SELECT pr.rfc
FROM proveedores pr
JOIN entregan e ON pr.rfc = e.rfc
WHERE YEAR(e.fecha) = 2000
GROUP BY pr.rfc
HAVING AVG(e.cantidad) >= 300;
CREATE VIEW TotalEntregado2000 AS
SELECT e.clave, SUM(e.cantidad) AS TotalEntregado
FROM entregan e
WHERE YEAR(e.fecha) = 2000
GROUP BY e.clave;
CREATE VIEW VentasPorMaterial2001 AS
SELECT e.clave, SUM(e.cantidad) AS TotalVendido
FROM entregan e
WHERE YEAR(e.fecha) = 2001
GROUP BY e.clave;
CREATE VIEW MaterialMasVendido2001 AS
SELECT TOP 1 clave
FROM VentasPorMaterial2001
ORDER BY TotalVendido DESC;
CREATE VIEW ProductosConUb AS
SELECT clave, descripcion
FROM materiales
WHERE descripcion LIKE '%ub%';
CREATE VIEW TotalPagarProyectos AS
SELECT p.denominacion, SUM(e.cantidad * m.precio * (1 + m.impuesto / 100))
AS TotalPagar
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
```

```

JOIN materiales m ON e.clave = m.clave
GROUP BY p.denominacion;
CREATE VIEW ProveedoresTelevisa AS
SELECT p.denominacion, pr.rfc, pr.razonsocial
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
JOIN proveedores pr ON e.rfc = pr.rfc
WHERE p.denominacion = 'Televisa en acción';
CREATE VIEW ProveedoresSoloTelevisa AS
SELECT denominacion, rfc, razonsocial
FROM ProveedoresTelevisa
WHERE rfc NOT IN (
    SELECT pr.rfc
    FROM proyectos p
    JOIN entregan e ON p.numero = e.numero
    JOIN proveedores pr ON e.rfc = pr.rfc
    WHERE p.denominacion = 'Educando en Coahuila'
);
SELECT p.denominacion, pr.rfc, pr.razonsocial
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
JOIN proveedores pr ON e.rfc = pr.rfc
WHERE p.denominacion = 'Televisa en acción'
AND pr.rfc NOT IN (
    SELECT pr.rfc
    FROM proyectos p
    JOIN entregan e ON p.numero = e.numero
    JOIN proveedores pr ON e.rfc = pr.rfc
    WHERE p.denominacion = 'Educando en Coahuila'
);
CREATE VIEW MaterialesCostoTelevisa AS
SELECT m.clave, m.descripcion, m.precio, e.cantidad, (e.cantidad * m.precio
* (1 + m.impuesto / 100)) AS Costo
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
JOIN materiales m ON e.clave = m.clave
JOIN proveedores pr ON e.rfc = pr.rfc
WHERE p.denominacion = 'Televisa en acción'
AND pr.rfc IN (
    SELECT pr.rfc
    FROM proyectos p
    JOIN entregan e ON p.numero = e.numero
    JOIN proveedores pr ON e.rfc = pr.rfc
    WHERE p.denominacion = 'Educando en Coahuila'
);
CREATE VIEW CostoTotalMateriales AS
SELECT m.descripcion, COUNT(e.numero) AS VecesEntregado, SUM(e.cantidad *
m.precio * (1 + m.impuesto / 100)) AS CostoTotal
FROM materiales m
JOIN entregan e ON m.clave = e.clave
GROUP BY m.descripcion;

```

	ABC denominacion ▼	ABC rfc ▼	ABC razonsocial ▼
1	Televisa en acción	CCCC800101	La Ferre
2	Televisa en acción	DDDD800101	Cecoferre
3	Televisa en acción	DDDD800101	Cecoferre