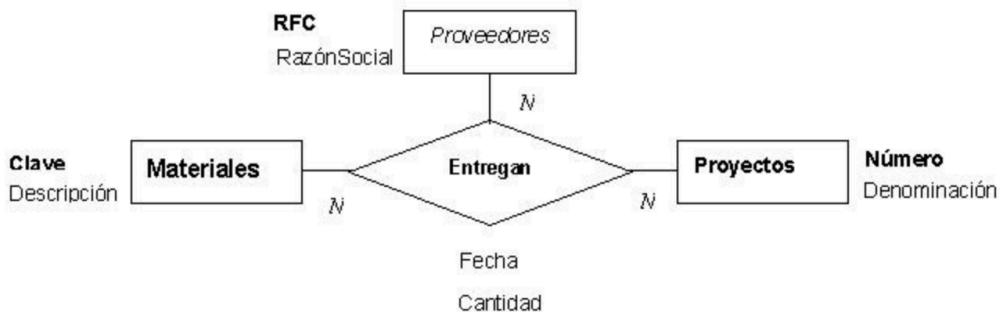


Laboratorio 20: Consultas en SQL

En este laboratorio se va a trabajar con la base de datos rcorse cuyo modelo entidad relación es el siguiente:



Posteriormente el MER se traduce a Modelo Relacional (MR), es importante seguir las reglas de traducción:

Materiales(Clave,Descripción,Costo)
Proveedores(RFC,RazonSocial)
Proyectos(Numero,Denominacion)
Entregan(Clave,RFC,Numero,Fecha,Cantidad)

Para construir la base de datos se utilizará el gestor MySQL Workbench, el comando para crear una base de datos es el siguiente:

```
CREATE DATABASE [nombre de la db]  
USE [nombre de la db]
```

Para crear una tabla se ejecuta el siguiente comando, es necesario incluir los atributos y el tipo de dato de cada uno. Las llaves también se le asignan, tanto primarias como foráneas (en caso de tener)

```
CREATE TABLE [nombre de tabla] (  
    atributo [tipo de dato]  
    atributo2 [tipo de dato]  
    atributo n [tipo de dato]  
    primary key [atributo que sea la llave primaria]  
    foreign key [atributo que sea la llave foránea] references [tabla que referencia]  
    [llave primaria de la llave que referencia]
```

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

) Esto es sólo un ejemplo, hay más reglas a tomar en cuenta.

MySQL Workbench cuenta con una forma de creación de tablas más visual:

The screenshot shows the MySQL Workbench interface for creating a table. The table name is 'entregan' and it is being created in the schema 'rcortese'. The table has five columns: 'clave' (INT, PK), 'rfc' (VARCHAR(15)), 'numero' (INT), 'fecha' (DATE), and 'cantidad' (INT). The 'clave' column is set as the primary key. The 'Column details' panel for 'clave' shows its type as INT, storage as VIRTUAL, and it is marked as a primary key. Other options like Not NULL, Unique, Binary, Unsigned, Auto Increment, and Generated are also visible.

Usando el MR como referencia, se crean las tablas. Una vez creadas, hay que cargarlas con datos, el script para crear las tablas y cargar los datos fue proporcionado. Ahora podemos comenzar a hacer las consultas para el laboratorio 20.

Todas las consultas que pueden plantearse con álgebra relacional, pueden expresarse con SQL. Convenio: para evitar las letras griegas originales del álgebra relacional, en esta lectura se utiliza la siguiente notación:

SL{condición} : selección con el criterio condición.

PR{lista de columnas}: proyección de lista de columnas.

JN: reunión natural (natural join).

JN{condición}: reunión con el criterio condición (teta join).

UN: unión.

IN: intersección.

- : diferencia

X: producto cartesiano.

Al diseñar consultas en SQL es importante considerar los siguientes puntos:

- La lista de columnas de la cláusula SELECT es la lista de la proyección final (más externa)
- La lista de tablas de la cláusula FROM incluye a todas las tablas participantes.
- Las condiciones se expresan en la cláusula WHERE, combinándolas con AND (o con OR según el significado específico).

Abre una sesión de Analizador de Consultas y ejecuta cada una de las sentencias SQL. 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. A continuación se presenta la equivalencia entre los operadores de álgebra relacional y SQL:

Consulta de un tabla completa

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

Algebra relacional.

materiales

SQL

select * from materiales

clave	descripción	precio	impuesto
1000	Varilla 3/16	100	10
1010	Varilla 4/32	115	11.5
1020	Varilla 3/17	130	13
1030	Varilla 4/33	145	14.5
1040	Varilla 3/18	160	16
1050	Varilla 4/34	175	17.5
1060	Varilla 3/19	190	19
1070	Varilla 4/35	205	20.5
1080	Ladrillos rojos	50	5
1090	Ladrillos grises	35	3.5
1100	Block	30	3
1110	Megablock	40	4
1120	Sillar rosa	100	10
1130	Sillar gris	110	11
1140	Cantera blanca	200	20
1150	Cantera gris	1210	121
1160	Cantera rosa	1420	142
1170	Cantera amar...	230	23
1180	Recubrimient...	200	20
1190	Recubrimient...	220	22

Selección

Algebra relacional.

SL{clave=1000}(materiales)

SQL

select * from materiales

where clave=1000

clave	descripción	precio	impuesto
1000	Varilla 3/16	100	10

Proyección

Algebra relacional.

PR{clave,rfc,fecha} (entregan)

SQL

select clave, rfc, fecha from entregan

```
21 •  SELECT clave, rfc, fecha FROM entregan;
22
```

100% 1:17

Result Grid Filter Rows: Search Export:

clave	rfc	fecha
1000	AAAA800101	2001-12-13
1200	EEEE800101	2003-03-15
1400	AAAAB800101	1999-04-07
1010	BBBBB800101	1998-07-28
1210	FFFF800101	2000-05-21
1410	BBBBB800101	2000-05-18
1020	CCCC800101	2003-12-16
1220	GGGG800101	2005-07-03
1420	CCCC800101	2001-09-09
1030	DDDD800101	1998-01-12
1230	HHHH800101	1998-09-12
1430	DDDD800101	2005-04-30
1040	EEEE800101	1999-12-18
1150	HHHH800101	2002-11-14
1240	AAAAB800101	2004-06-23
1050	FFFF800101	2004-02-07
1140	GGGG800101	2000-06-30
1250	BBBBB800101	2003-10-12
1300	GGGG800101	2004-02-28

Reunión Natural

Algebra relacional.

entregan JN materiales

SQL

select * from materiales,entregan

where materiales.clave = entregan.clave

```
23 •  SELECT * FROM materiales, entregan
24    WHERE materiales.clave = entregan.clave;
```

100% 1:22

Result Grid Filter Rows: Search Export:

clave	descripc...	precio	impuesto	clave	rfc	numero	fecha	cantidad
1000	Varilla 3/16	100	10	1000	AAAA800101	5000	2001-12-13	165
1000	Varilla 3/16	100	10	1000	AAAA800101	5019	1999-07-13	254
1010	Varilla 4/32	115	11.5	1010	BBBBB800101	5001	1998-07-28	528
1010	Varilla 4/32	115	11.5	1010	BBBBB800101	5018	1997-02-09	523
1020	Varilla 3/17	130	13	1020	CCCC800101	5002	2003-12-16	582
1020	Varilla 3/17	130	13	1020	CCCC800101	5017	2000-03-29	8
1030	Varilla 4/33	145	14.5	1030	DDDD800101	5003	1998-01-12	202
1030	Varilla 4/33	145	14.5	1030	DDDD800101	5016	2005-06-07	295
1040	Varilla 3/18	160	16	1040	EEEE800101	5004	1999-12-18	263
1040	Varilla 3/18	160	16	1040	EEEE800101	5015	1999-05-29	540
1050	Varilla 4/34	175	17.5	1050	FFFF800101	5005	2004-02-07	503
1050	Varilla 4/34	175	17.5	1050	FFFF800101	5014	2000-04-18	623
1060	Varilla 3/19	190	19	1060	GGGG800101	5006	2004-09-29	324
1060	Varilla 3/19	190	19	1060	GGGG800101	5013	2002-07-23	692
1070	Varilla 4/35	205	20.5	1070	HHHH800101	5007	2006-11-21	2
1070	Varilla 4/35	205	20.5	1070	HHHH800101	5012	2004-11-27	503
1080	Ladrillos ...	50	5	1080	AAAAB800101	5008	2005-04-03	86
1080	Ladrillos ...	50	5	1080	AAAAB800101	5011	2003-08-28	699
1090	Ladrillos...	35	3.5	1090	BBBBB800101	5009	1997-03-13	73
1090	Ladrillos...	35	3.5	1090	BBBBB800101	5010	1998-11-17	421

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

No, no aparecerán en la tabla, en el Join natural sólo se muestran resultados que ambas tablas tengan en común, por lo tanto, si hay un material no entregado, no se verá reflejado en la consulta ya que la tabla entregan no lo tendría registrado.

Reunión con criterio específico

Algebra relacional.

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

SQL

select * from entregan,proyectos

where entregan.numero <= proyectos.numero

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

```
26 •   SELECT * FROM entregan,proyectos  
27     WHERE entregan.numero <= proyectos.numero;  
28
```

100% 1:25

Result Grid Filter Rows: Search Export:

clave	rfc	numero	fecha	cantidad	numero	denominacion
1000	AAAAA800101	5000	2001-12-13	165	5000	Vamos Mexico
1200	EEEE800101	5000	2003-03-15	177	5000	Vamos Mexico
1400	AAAAA800101	5000	1999-04-07	382	5000	Vamos Mexico
1000	AAAAA800101	5000	2001-12-13	165	5001	Aztecon
1200	EEEE800101	5000	2003-03-15	177	5001	Aztecon
1400	AAAAA800101	5000	1999-04-07	382	5001	Aztecon
1010	BBBB800101	5001	1998-07-28	528	5001	Aztecon
1210	FFFF800101	5001	2000-05-21	43	5001	Aztecon
1410	BBBB800101	5001	2000-05-18	601	5001	Aztecon
1000	AAAAA800101	5000	2001-12-13	165	5002	CIT Campeche
1200	EEEE800101	5000	2003-03-15	177	5002	CIT Campeche
1400	AAAAA800101	5000	1999-04-07	382	5002	CIT Campeche
1010	BBBB800101	5001	1998-07-28	528	5002	CIT Campeche
1210	FFFF800101	5001	2000-05-21	43	5002	CIT Campeche
1410	BBBB800101	5001	2000-05-18	601	5002	CIT Campeche
1020	CCCC800101	5002	2003-12-16	582	5002	CIT Campeche
1220	GGGG800101	5002	2005-07-03	24	5002	CIT Campeche
1420	CCCC800101	5002	2001-09-09	603	5002	CIT Campeche
1000	AAAAA800101	5000	2001-12-13	165	5003	Mexico sin ti...
1200	EEEE800101	5000	2003-03-15	177	5003	Mexico sin ti...

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)

```
29 •   (SELECT * FROM entregar WHERE clave=1450)
30 UNION
31 (SELECT * FROM entregar WHERE clave=1300)
32
```

100% 33:26

Result Grid					Filter Rows:	Search	Export:	Print
clave	rfc	numero	fecha	cantidad				
1300	GGGG800101	5005	2004-02-28	521				
1300	GGGG800101	5010	2001-02-10	119				

(No hay materiales cuya clave es 1450)

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

Podemos obtener el mismo resultado utilizando el operador binario *OR* , esto devolverá materiales cuya clave sea 1300 o 1450, como no hay materiales con clave 1450, se probará con otra clave:

```
33 •   SELECT * FROM entregan  
34 WHERE clave=1300 OR clave=1450;
```

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan))

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)

En MySQL workbench, podemos utilizar subconsultas para devolver el resultado el cual sería la clave con el número 5001 y que además esa misma clave también exista en la tabla entregan y donde el número 5018:

```
36 •  SELECT clave
37   FROM entregar
38 WHERE numero = 5001
39 AND clave IN (SELECT clave FROM entregar WHERE numero = 5018);

100%  ◊ | 32:34 |
```

Result Grid Filter Rows: Search Export:

clave
1010

Diferencia (se ilustra con selección)

Algebra relacional.

entregan - SL{clave=1000}(entregan)

SQL

(select * from entreqan)

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.

Podemos usar la cláusula *NOT IN* para obtener las tuplas de la tabla entregan a excepción de los que tengan la clave 1000:

```
41 •   SELECT *
42     FROM entregan
43     WHERE clave NOT IN (SELECT clave FROM entregan WHERE clave = 1000);
```

clave	rfc	numero	fecha	cantidad
1010	BBBBB800101	5001	1998-07-28	528
1010	BBBBB800101	5018	1997-02-09	523
1020	CCCCC800101	5002	2003-12-16	582
1020	CCCCC800101	5017	2000-03-29	8
1030	DDDD800101	5003	1998-01-12	202
1030	DDDD800101	5016	2005-06-07	295
1040	EEE800101	5004	1999-12-18	263
1040	EEE800101	5015	1999-05-29	540
1050	FFFF800101	5005	2004-02-07	503
1050	FFFF800101	5014	2000-04-18	623
1060	GGGG800101	5008	2004-09-29	324
1060	GGGG800101	5013	2002-07-23	692
1070	HHHH800101	5007	2006-11-21	2
1070	HHHH800101	5012	2004-11-27	503
1080	AAAA800101	5008	2005-04-03	86
1080	AAAA800101	5011	2003-08-28	699
1090	BBBBB800101	5009	1997-03-13	73
1090	BBBBB800101	5010	1998-11-17	421
1100	CCCCC800101	5009	2000-12-07	466
1100	CCCCC800101	5010	2001-11-19	699

Producto cartesiano

Algebra relacional.

entregan X materiales

SQL

select * from entregan,materiales

```
45 •   SELECT * FROM entregan,materiales
```

100% ◁ 9:41 |

clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
1000	AAAAB800101	5000	2001-12-13	165	2000	Jabón	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1430	Pintura B1022	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1420	Pintura C1012	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1410	Pintura B1021	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1400	Pintura C1011	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1390	Pintura B1021	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1380	Pintura C1011	725	72.5
1000	AAAAB800101	5000	2001-12-13	165	1370	Pintura B1020	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1360	Pintura C1010	125	12.5
1000	AAAAB800101	5000	2001-12-13	165	1350	Tubería 3.8	260	26
1000	AAAAB800101	5000	2001-12-13	165	1340	Tubería 4.5	250	25
1000	AAAAB800101	5000	2001-12-13	165	1330	Tubería 3.7	240	24
1000	AAAAB800101	5000	2001-12-13	165	1320	Tubería 4.4	230	23
1000	AAAAB800101	5000	2001-12-13	165	1310	Tubería 3.6	220	22
1000	AAAAB800101	5000	2001-12-13	165	1300	Tubería 4.3	210	21
1000	AAAAB800101	5000	2001-12-13	165	1290	Tubería 3.5	200	20
1000	AAAAB800101	5000	2001-12-13	165	1280	Tepetate	34	3.4
1000	AAAAB800101	5000	2001-12-13	165	1270	Tezontle	80	8
1000	AAAAB800101	5000	2001-12-13	165	1260	Gravilla	90	9
1000	AAAAB800101	5000	2001-12-13	165	1250	Grava	100	10

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

Al ser un producto cartesiano, la cantidad de tuplas es el resultado del producto de las tuplas de las tablas entregan y materiales. Usando la cláusula Count(*) en ambas tablas, se puede obtener el número de tuplas de ambas tablas:

SELECT COUNT(*) FROM materiales; Esto devuelve 45

SELECT COUNT(*) FROM entregan; Esto devuelve 87

El producto de 45 y 87 es 3915, si aplicamos count al comando que ejecuta el producto cruz, obtenemos lo siguiente:

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

```
47 •   SELECT COUNT(*) FROM materiales;
48 •   SELECT COUNT(*) FROM entregan;
49 •   SELECT COUNT(*) FROM entregan,materiales;
50
```

100% ◇ | 1:46 |

Result Grid		Filter Rows:	Search	Export:	□
<hr/>					
COUNT(*)					
3915					
<hr/>					

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.

En el caso de esta base de datos, el formato es año-mes-día:

```
51 •   SELECT descripcion
52     FROM entregan e, materiales m
53    WHERE e.clave = m.clave
54   AND fecha BETWEEN '2000-01-01' AND '2000-12-31';
55
```

100% ◇ | 37:49 |

Result Grid		Filter Rows:	Search	Export:	□
<hr/>					
descripcion					
Recubrimiento P1028					
Pintura B1021					
Cantera blanca					
Sillar gris					
Pintura B1022					
Block					
Tuberia 3.6					
Sillar gris					
Varilla 4/34					
Pintura C1010					
Varilla 3/17					
Pintura B1021					
<hr/>					

```
51 •   SELECT descripcion, impuesto
52     FROM entregan e, materiales m
53    WHERE e.clave = m.clave
54   AND fecha BETWEEN '2000-01-01' AND '2000-12-31';
55
```

100% ◇ | 19:47 |

Result Grid		Filter Rows:	Search	Export:	□
<hr/>					
descripcion		impuesto			
Recubrimiento P1028		25			
Pintura B1021		12.5			
Cantera blanca		20			
Sillar gris		11			
Pintura B1022		12.5			
Block		3			
Tuberia 3.6		22			
Sillar gris		11			
Varilla 4/34		17.5			
Pintura C1010		12.5			
Varilla 3/17		13			
Pintura B1021		12.5			
<hr/>					

¿Por qué aparecen varias veces algunas descripciones de material?

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

La razón de que un material aparezca más de una vez es debido a que un material puede ser entregado más de una vez en un periodo de tiempo.

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.

The screenshot shows a MySQL query editor with the following code:

```
51 •   SELECT DISTINCT descripcion
52     FROM entregan e, materiales m
53    WHERE e.clave = m.clave
54   AND fecha BETWEEN '2000-01-01' AND '2000-12-31';
55
```

The results are displayed in a table titled "Result Grid" with one column labeled "descripcion". The data is as follows:

descripcion
Recubrimiento P1028
Pintura B1021
Cantera blanca
Sillar gris
Pintura B1022
Block
Tuberia 3.6
Varilla 4/34
Pintura C1010
Varilla 3/17

¿Qué resultado obtienes en esta ocasión?

En esta ocasión, las descripciones no se repitieron

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.

```

56 •   SELECT e.numero, p.denominacion
57     FROM entregan e, proyectos p
58    WHERE e.numero = p.numero
59    AND fecha BETWEEN '2000-01-01' AND '2000-12-31'
60   ORDER BY p.numero;
100% 30:52

```

Result Grid Filter Rows: Search Export:

numero	denominacion
5001	Aztecon
5001	Aztecon
5005	Infonavit Durango
5006	Reconstrucción del templo de Guadalupe
5007	Construcción de plaza Magnolias
5009	Disco Atlantic
5011	Remodelación de aulas del IPP
5013	Reparación de la plaza Sonora
5014	Remodelación de Soriana
5014	Remodelación de Soriana
5017	Reparación de la carretera del sol
5019	Querétaro limpio

```

56 •   SELECT e.numero, p.denominacion, e.cantidad, e.fecha
57     FROM entregan e, proyectos p
58    WHERE e.numero = p.numero
59    AND fecha BETWEEN '2000-01-01' AND '2000-12-31'
60   ORDER BY e.numero DESC;
100% 25:53

```

Result Grid Filter Rows: Search Export:

numero	denominacion	cantidad	fecha
5019	Querétaro limpio	107	2000-11-10
5017	Reparación de la carretera del sol	8	2000-03-29
5014	Remodelación de Soriana	823	2000-04-18
5014	Remodelación de Soriana	265	2000-02-04
5013	Reparación de la plaza Sonora	63	2000-02-09
5011	Remodelación de aulas del IPP	72	2000-09-14
5009	Disco Atlantic	466	2000-12-07
5007	Construcción de plaza Magnolias	13	2000-03-02
5006	Reconstrucción del templo de Guadalupe	562	2000-04-13
5005	Infonavit Durango	583	2000-06-30
5001	Aztecon	43	2000-05-21
5001	Aztecon	601	2000-05-18

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.

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

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 materiales where descripcion LIKE 'Si%'

¿Qué resultado obtienes?

Explica que hace el símbolo '%'.

El operador porcentaje se utiliza en conjunto a *LIKE* para buscar patrones en cadenas de texto, por ende, escribir *LIKE 'Si'* devolverá la descripción de los materiales que empiecen específicamente con la palabra o sílaba "Si".

clave	descripcion	precio	impuesto
1120	Sillar rosa	100	10
1130	Sillar gris	110	11
NULL	NULL	NULL	NULL

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

Si fuera *LIKE 'Si'*, la consulta devolvería la descripción de materiales cuya descripción sea exactamente 'Si'.

¿Qué resultado obtienes?

Al no haber descripciones que sean 'Si', la consulta devuelve una tabla vacía.

clave	descripción	precio	impuesto
NULL	NULL	NULL	NULL

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:

SELECT (descripcion + ', ' + precio) as material FROM materiales;

En MySQL existe la cláusula *CONCAT()* cuya función es la misma:

material
Varilla 3/16, 100
Varilla 4/32, 115
Varilla 3/17, 130
Varilla 4/33, 145
Varilla 3/18, 160
Varilla 4/34, 175
Varilla 3/19, 190
Varilla 4/35, 205
Ladrillos rojos, 50
Ladrillos grises, 35
Block, 30
Megablock, 40

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 contraparte, este operador coincide con cualquier carácter que no se encuentre dentro del intervalo o del conjunto especificado.

_ - El operador '_' o guión bajo, se utiliza para coincidir con un carácter 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]%' ; Esta consulta devuelve los rfc's que empiecen con una letra de la A a la D

SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%' ; Esta consulta devuelve rfc's que no comiencen con una letra A

SELECT Numero FROM Entregan WHERE numero LIKE '__6' ; Esta consulta devolverá rfc's que terminen con un seis

Nota: En mi caso, los corchetes fueron interpretados como una cadena literal, por lo que la consulta resultó vacía, sólo la tercera devolvió lo solicitado.

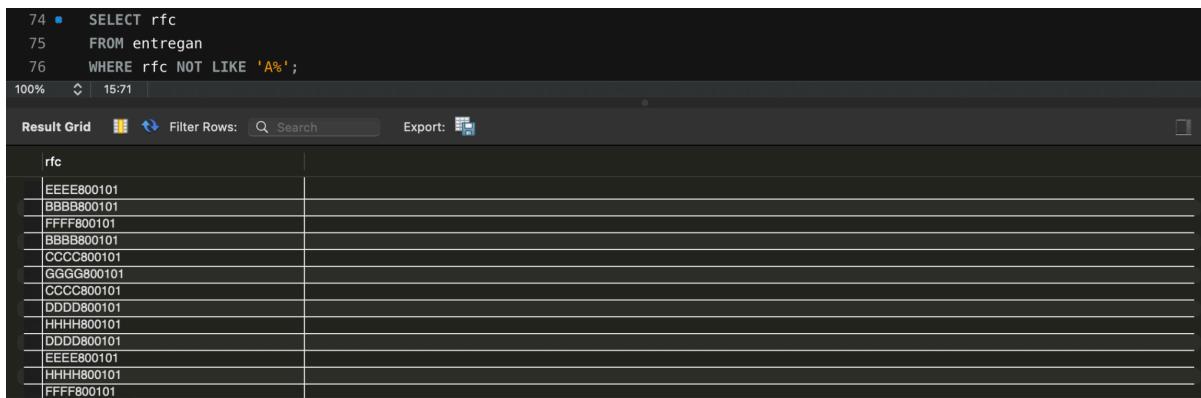
```
68 • SELECT numero FROM entregan WHERE numero LIKE '__6';
69
100% 1:65 | Result Grid Filter Rows: Search Export: □
```

numero
5006
5006
5006
5006
5016
5016
5016
5016
5016

Podemos hacer una consulta equivalente para obtener lo solicitado:

```
70 • SELECT RFC
71   FROM Entregan
72 WHERE RFC LIKE 'A%' OR RFC LIKE 'B%' OR RFC LIKE 'C%' OR RFC LIKE 'D%' ;
100% 54:68 | Result Grid Filter Rows: Search Export: □
```

RFC
AAAA800101
AAAAB00101
BBBB800101
BBBBB800101
CCCC800101
CCC800101
DDDD800101
DDDDB800101
AAAAB00101
BBBB800101
CCCC800101
DDDD800101
DDDDB800101



The screenshot shows a MySQL query results grid. The query is:

```
74 • SELECT rfc
75   FROM entregan
76  WHERE rfc NOT LIKE 'A%';
100% 15:71
```

The results grid has a header row labeled "rfc". Below it, there are 15 rows of data, each containing a different 6-digit string starting with E, B, F, C, G, D, H, or F.

rfc
EEEE800101
BBBB800101
FFFF800101
BBBB800101
CCCC800101
GGGG800101
CCCC800101
DDDD800101
HHHH800101
DDDD800101
EEEE800101
HHHH800101
FFFF800101

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,Numer0,Fecha,Cantidad
FROM Entregan
WHERE Numer0 Between 5000 and 5010;
```

¿Cómo filtrarías rangos de fechas?

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

```
78 •   SELECT clave,rfc,numero,fecha,cantidad  
79     FROM entregar  
80    WHERE numero Between 5000 and 5010;  
100% 25:76 |
```

Result Grid Edit: Export/Import:

clave	rfc	numero	fecha	cantidad	
1000	AAAAA800101	5000	2001-12-13	165	
1010	BBBBB800101	5001	1998-07-28	528	
1020	CCCCC800101	5002	2003-12-16	582	
1030	DDDD800101	5003	1998-01-12	202	
1040	EEE800101	5004	1999-12-18	263	
1050	FFFF800101	5005	2004-02-07	503	
1060	GGGG800101	5006	2004-09-29	324	
1070	HHHH800101	5007	2006-11-21	2	
1080	AAA800101	5008	2005-04-03	86	
1090	BBBBB800101	5009	1997-03-13	73	
1090	BBBBB800101	5010	1998-11-17	421	
1100	CCCCC800101	5009	2000-12-07	466	
1100	CCCCC800101	5010	2001-11-19	699	

Para filtrar los rangos de fechas, podemos aplicar una lógica similar a la usada con número, en este caso lo haremos para se sea en un periodo de un año:

```
82 •   SELECT clave,rfc,numero,fecha,cantidad  
83     FROM entregar  
84    WHERE fecha Between '2001-01-01' and '2001-12-31';  
100% 36:80 |
```

Result Grid Edit: Export/Import:

clave	rfc	numero	fecha	cantidad	
1000	AAAAA800101	5000	2001-12-13	165	
1100	CCCCC800101	5010	2001-11-19	699	
1180	CCCCC800101	5018	2001-10-24	407	
1260	CCCCC800101	5006	2001-04-09	460	
1260	CCCCC800101	5009	2001-07-28	631	
1300	GGGG800101	5010	2001-02-10	119	
1330	BBBBB800101	5013	2001-03-13	554	
1420	CCCCC800101	5002	2001-09-09	603	
HULL	HULL	HULL	HULL	HULL	

Esta consulta devuelve el resultado en el año de 2001 (importante tener en cuenta el formato de fecha).

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

SELECT RFC,Cantidad, Fecha,Numero

FROM [Entregar]

WHERE [Numero] Between 5000 and 5010 AND

Exists (SELECT [RFC]

FROM [Proveedores]

WHERE RazonSocial LIKE 'La%' and [Entregar].[RFC] = [Proveedores].[RFC])

```
86 •   SELECT rfc,cantidad, fecha,numero  
87     FROM entregar  
88    WHERE numero Between 5000 and 5010 AND  
89      Exists ( SELECT rfc  
90        FROM proveedores  
91       WHERE razonsocial LIKE 'La%' and entregar.rfc = proveedores.rfc);  
100% 51:84 |
```

Result Grid Edit: Export:

rfc	cantidad	fecha	numero	
AAAAA800101	165	2001-12-13	5000	
AAAAA800101	86	2005-04-03	5008	
AAAAA800101	152	2004-06-23	5004	
AAAAA800101	382	1999-04-07	5000	
AAAAA800101	116	2005-04-21	5010	
CCCCC800101	582	2003-12-16	5002	
CCCCC800101	466	2000-12-07	5009	
CCCCC800101	699	2001-11-19	5010	
CCCCC800101	460	2001-04-09	5006	
CCCCC800101	631	2001-07-28	5009	
CCCCC800101	603	2001-09-09	5002	
CCCCC800101	278	1999-05-05	5008	

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

¿Qué hace la consulta?

Devuelve rfc, cantidad, fecha y numero de la tabla entregan cuyo numero esta entre 5000 y 5010.

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

Exists verifica si al menos una fila de la columna razonsocial de la tabla proveedores comienza con 'La' y el rfc coincide en ambas tablas.

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

NOTA: Se utiliza dentro del WHERE pero debe contener un parámetro. 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

```
93 •  SELECT rfc,cantidad, fecha,numero
94   FROM entregan
95   WHERE numero Between 5000 and 5010
96   AND rfc IN ( SELECT rfc
97     FROM proveedores
98     WHERE razonsocial LIKE 'La%');

100% 1:92 |
```

Result Grid Filter Rows: Search Export:

rfc	cantidad	fecha	numero
AAAAA800101	165	2005-12-13	5000
AAAAA800101	86	2005-04-03	5008
AAAAA800101	152	2004-06-23	5004
AAAAA800101	382	1999-04-07	5000
AAAAA800101	116	2005-04-21	5010
CCCC800101	582	2003-12-16	5002
CCCC800101	466	2000-12-07	5009
CCCC800101	699	2001-11-19	5010
CCCC800101	460	2001-04-09	5006
CCCC800101	631	2001-07-28	5009
CCCC800101	603	2001-09-09	5002
CCCC800101	278	1999-05-05	5008

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.

```
93 •  SELECT rfc,cantidad, fecha,numero
94   FROM entregan
95   WHERE numero Between 5000 and 5010
96   AND rfc NOT IN ( SELECT rfc
97     FROM proveedores
98     WHERE razonsocial LIKE 'La%');

100% 1:92 |
```

Result Grid Filter Rows: Search Export:

rfc	cantidad	fecha	numero
BBBB800101	528	1998-07-28	5001
DDDD800101	202	1998-01-12	5003
EEEE800101	263	1999-12-18	5004
FFFF800101	503	2004-02-07	5005
GGGG800101	324	2004-09-29	5006
HHHH800101	2	2006-11-21	5007
BBBB800101	73	1997-03-13	5009
BBBB800101	421	1998-11-17	5010
DDDD800101	337	2005-06-03	5008
EEEE800101	692	2003-11-21	5007
FFFF800101	562	2000-04-13	5006
GGGG800101	583	2000-06-30	5005
HHHH800101	453	2002-11-14	5004
EEEE800101	177	2003-03-15	5000
FFFF800101	43	2000-05-21	5001
GGGG800101	24	2005-07-03	5002
HHHH800101	530	1998-09-12	5003
BBBB800101	71	2003-10-12	5005
RRRR800101	691	1999-01-25	5010

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

```
100 •  SELECT rfc, cantidad, fecha, numero
101   FROM entregan
102  WHERE numero BETWEEN 5000 AND 5010
103  ⊕ AND cantidad > ALL (
104    SELECT cantidad
105     FROM entregan
106    WHERE cantidad <= 50
107  );
```

100% 31:98 |

Result Grid Filter Rows: Search Export:

rfc	cantidad	fecha	numero
BBBB800101	528	1998-07-28	5001
CCCC800101	582	2003-12-16	5002
DDDD800101	202	1998-01-12	5003
EEFF800101	263	1999-12-18	5004
FFFF800101	503	2004-02-07	5005
GGGG800101	324	2004-09-29	5006
AAAA800101	86	2005-04-03	5007
BBBB800101	73	1997-03-13	5009
BBBB800101	421	1998-11-17	5010
CCCC800101	466	2000-12-07	5009
CCCC800101	699	2001-11-19	5010
DDDD800101	337	2005-06-03	5008
EEEE800101	692	2003-11-21	5007
FFFF800101	562	2000-04-13	5006
GGGG800101	583	2000-06-30	5005
HHHH800101	453	2002-11-14	5004

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.

SELECT TOP 2 * FROM Proyectos. Esto devuelve las primeras dos filas de la tabla proyecto ya que se especifica que deben ser las primeras 2. En MySQL Workbench, podemos utilizar la cláusula *LIMIT* para lograr el mismo resultado.

```
109 •  SELECT * FROM proyectos
110  LIMIT 2;
```

100% 3:107 |

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

numero	denominacion
5000	Vamos Mexico
5001	Aztecon
HULL	HULL

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

SELECT TOP Numero FROM Proyectos. En MySQL Workbench, el equivalente sería lo siguiente:

SELECT numero FROM proyectos

LIMIT numero;

Esta consulta tal cual no es válida porque numero en límite es interpretado como variable y no está declarado. En este contexto, número sería la cantidad para que aparezcan las primeras número filas de la tabla proveedores.

```
114 •   SELECT numero FROM proyectos
115     LIMIT 10;
100%  ◁  34:112 |
```

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

numero
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
NULL

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.

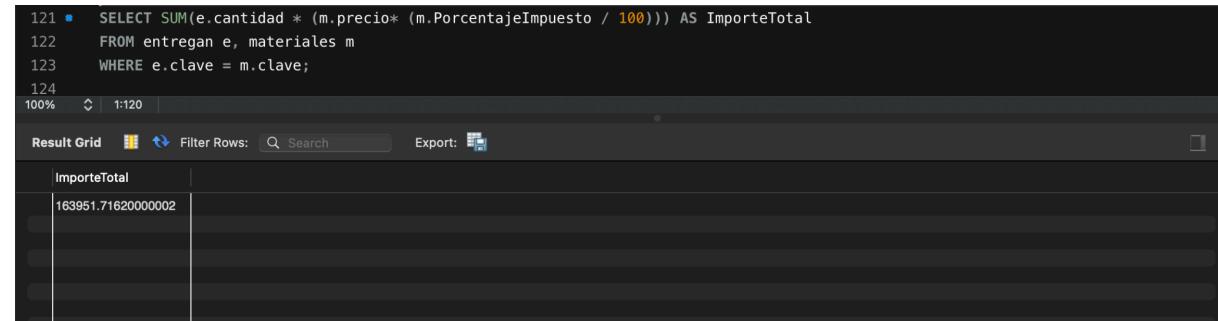
```
117 •   ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);
118
119 •   UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;
120
100%  ◁  1:116 |
```

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

clave	descripcion	precio	impuesto	Porcentajeimpues...
1020	Varilla 3/17	130	13	2.04
1030	Varilla 4/33	145	14.5	2.06
1040	Varilla 3/18	160	16	2.08
1050	Varilla 4/34	175	17.5	2.10
1060	Varilla 3/19	190	19	2.12
1070	Varilla 4/35	205	20.5	2.14
1080	Ladrillos rojos	50	5	2.16
1090	Ladrillos grises	35	3.5	2.18
1100	Block	30	3	2.20
1110	Megablock	40	4	2.22
1120	Sillar rosa	100	10	2.24
1130	Sillar gris	110	11	2.26
1140	Cantera blanca	200	20	2.28
1150	Cantera gris	1210	121	2.30
1160	Cantera rosa	1420	142	2.32
1170	Cantera amar...	230	23	2.34
1180	Recubrimient...	200	20	2.36
1190	Recubrimient...	220	22	2.38
1200	Recubrimient...	240	24	2.40
1210	Recubrimient...	250	25	2.42
1220	Recubrimient...	280	28	2.44
1230	Cemento	200	20	2.46

¿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?

Sería a través de una operación del producto del precio de un material por el porcentaje de impuesto (El porcentaje está expresado en decimal, así que hay que dividir entre 100). Posteriormente multiplicarlo por la cantidad, por último hacer una sumatoria con la cláusula *SUM()*. Es necesario hacer un Join natural entre materiales y entregan:



The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window with the following SQL query:

```
121 •  SELECT SUM(e.cantidad * (m.precio* (m.PorcentajeImpuesto / 100))) AS ImporteTotal  
122   FROM entregan e, materiales m  
123  WHERE e.clave = m.clave;  
124  
100% ◇ | 1:120 |
```

Below the code editor is a "Result Grid" window. It has a header row with a single column labeled "ImporteTotal". The data row contains the value "163951.71620000002".

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

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.

1. Join natural de las tablas materiales y entregan

```
SELECT * FROM materiales, entregan  
WHERE materiales.clave = entregan.clave;
```

```

131 • CREATE VIEW VistaJoin AS
132   SELECT m.clave, m.descripcion, m.precio, e.rfc, e.numero, e.fecha, e.cantidad
133   FROM materiales m, entregan e
134   WHERE m.clave = e.clave;
135
136 • SELECT * FROM VistaJoin;
137
100% ◇ | 27:129 |

```

Result Grid Filter Rows: Search Export:

clave	descripcion	precio	rfc	numero	fecha	cantidad
1000	Varilla 3/16	100	AAAAA800101	5000	2001-12-13	165
1000	Varilla 3/16	100	AAAAA800101	5019	1999-07-13	254
1010	Varilla 4/32	115	BBBBB800101	5001	1998-07-28	528
1010	Varilla 4/32	115	BBBBB800101	5018	1997-02-09	523
1020	Varilla 3/17	130	CCCCC800101	5002	2003-12-16	582
1020	Varilla 3/17	130	CCCCC800101	5017	2000-03-29	8
1030	Varilla 4/33	145	DDDD800101	5003	1988-01-12	202
1030	Varilla 4/33	145	DDDD800101	5016	2005-06-07	295
1040	Varilla 3/18	160	EEEE800101	5004	1999-12-18	263
1040	Varilla 3/18	160	EEEE800101	5015	1999-05-29	540
1050	Varilla 4/34	175	FFFF800101	5005	2004-02-07	503
1050	Varilla 4/34	175	FFFF800101	5014	2000-04-18	623
1060	Varilla 3/19	190	GGGG800101	5006	2004-09-29	324
1060	Varilla 3/19	190	GGGG800101	5013	2002-07-23	692
1070	Varilla 4/35	205	HHHH800101	5007	2006-11-21	2
1070	Varilla 4/35	205	HHHH800101	5012	2004-11-27	503

2. Los materiales entregados cuyas claves sean 1300 o 1400

SELECT * FROM entregan

WHERE clave=1300 OR clave=1400;

```

138 • CREATE VIEW VMateriales1300_1400 AS
139   SELECT *
140   FROM entregan
141   WHERE clave = 1300 OR clave = 1400;
142
143 • SELECT * FROM VMateriales1300_1400;
144
145
100% ◇ | 1:137 |

```

Result Grid Filter Rows: Search Export:

clave	rfc	numero	fecha	cantidad
1300	GGGG800101	5005	2004-02-28	521
1300	GGGG800101	5010	2001-02-10	119
1400	AAAA800101	5000	1999-04-07	382
1400	AAAA800101	5010	2005-04-21	116

3. Descripciones de los materiales entregados en el año 2000 (que las descripciones no se repitan):

SELECT DISTINCT descripcion

FROM entregan e, materiales m

WHERE e.clave = m.clave

AND fecha BETWEEN '2000-01-01' AND '2000-12-31';

```
145 • CREATE VIEW Descripciones2000 AS
146     SELECT DISTINCT m.descripcion
147     FROM entregan e
148     JOIN materiales m ON e.clave = m.clave
149     WHERE fecha BETWEEN '2000-01-01' AND '2000-12-31';
150
151 • SELECT * FROM Descripciones2000
152
```

100% 1:144 |

Result Grid Filter Rows: Search Export:

descripcion
Recubrimiento P1028
Pintura B1021
Cantera blanca
Siller grs
Pintura B1022
Block
Tubería 3.6
Varilla 4/34
Pintura C1010
Varilla 3/17

4. rfc que empiezan con una letra de la A a la D:

```
SELECT rfc
FROM entregan
WHERE rfc LIKE 'A%' OR RFC LIKE 'B%' OR RFC LIKE 'C%' OR RFC LIKE
'D%';
```

```
153 • CREATE VIEW VistaRFC_AD AS
154     SELECT rfc
155     FROM entregan
156     WHERE rfc LIKE 'A%' OR rfc LIKE 'B%' OR rfc LIKE 'C%' OR rfc LIKE 'D%';
157
158 • SELECT * FROM VistaRFC_AD;
159
160
```

100% 1:152 |

Result Grid Filter Rows: Search Export:

rfc
AAAAA800101
BBBBB800101
BBBBB800101
BBBBB800101
BBBBB800101

5. rfc, cantidad, fecha y numero de la tabla entregan cuyo numero está entre 5000 y 5010, además, cuya razón social comienza con “La” y coincide en las tablas entregan y proveedores:

```
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%');
```

Ian Julián Estrada Castro

A01352823

TC2005B - Construcción de software y toma de decisiones

```
160 • CREATE VIEW VistaEntregas5000_5010 AS
161     SELECT rfc, cantidad, fecha, numero
162     FROM entregan
163     WHERE numero BETWEEN 5000 AND 5010
164     AND rfc NOT IN (SELECT rfc FROM proveedores WHERE razonsocial LIKE 'La%');
165
166 • SELECT * FROM VistaEntregas5000_5010;
167
100% 1:159 |
```

Result Grid Filter Rows: Search Export:

rfc	cantidad	fecha	numero
BBBB800101	528	1998-07-28	5001
DDDD800101	202	1998-01-12	5003
EEEE800101	263	1999-12-18	5004
FFFF800101	503	2004-02-07	5005
GGGG800101	324	2004-09-29	5006
HHHH800101	2	2006-11-21	5007
BBBB800101	73	1997-03-13	5009
BBBB800101	421	1998-11-17	5010
DDDD800101	337	2005-06-03	5008
EEEE800101	692	2003-11-21	5007
FFFF800101	562	2000-04-13	5006
GGGG800101	583	2000-06-30	5005
HHHH800101	453	2002-11-14	5004
EEEE800101	177	2003-03-16	5000
FFFF800101	43	2000-05-21	5001
GGGG800101	24	2005-07-03	5002

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.

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

```
168 • SELECT m.clave, m.descripcion
169     FROM materiales m, entregan e, proyectos p
170     WHERE m.clave = e.clave
171     AND e.numero = p.numero
172     AND p.denominacion = 'Méjico sin ti no estamos completos';
173
174
100% 27:166 |
```

Result Grid Filter Rows: Search Export:

clave	descripcion
1030	Varilla 4/33
1230	Cemento
1430	Pintura B1022

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

```
174 • SELECT m.clave, m.descripcion, p.razonsocial
175     FROM materiales m, entregan e, proveedores p
176     WHERE m.clave = e.clave AND e.rfc = p.rfc AND p.razonsocial = 'Acme tools';
177
100% 5:170 |
```

Result Grid Filter Rows: Search Export:

clave	descripcion	razonsocial

No hay ningún proveedor cuya denominación sea Acme Tools.

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

```
178 •   SELECT e.rfc
179     FROM entregan e
180    WHERE fecha BETWEEN '2000-01-01' AND '2000-12-31'
181   AND e.cantidad >= 300
182   GROUP BY e.rfc;
183
100%  ◊ | 1:177 |
```

Result Grid	
	rfc
	BBBB800101
	CCCC800101
	FFFF800101
	GGGG800101

4. El Total entregado por cada material en el año 2000.

```
184 •   SELECT e.clave, SUM(e.cantidad * m.precio) AS TotalEntregado, m.descripcion
185     FROM entregan e, materiales m
186    WHERE e.clave = m.clave AND fecha BETWEEN '2000-01-01' AND '2000-12-31'
187   GROUP BY e.clave;
188
100%  ◊ | 16:182 |
```

Result Grid		
clave	TotalEntrega...	descripcion
1020	1040	Varilla 3/17
1050	109025	Varilla 4/34
1100	13980	Block
1130	68750	Sillar gris
1140	116600	Cantera blanca
1210	10750	Recubrimiento P1028
1310	15840	Tubería 3.6
1360	33125	Pintura C1010
1390	13375	Pintura B1021
1410	75125	Pintura B1021
1430	1625	Pintura B1022

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

```
190 •   SELECT e.clave, m.descripcion
191     FROM entregan e, materiales m
192    WHERE e.clave = m.clave
193    AND e.fecha BETWEEN '2001-01-01' AND '2001-12-31'
194   GROUP BY e.clave
195   ORDER BY SUM(e.cantidad) DESC
196   LIMIT 1;
197
100%  ◊ | 1:188 |
```

Result Grid	
clave	descripc...
1260	Gravilla

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

```
198 •   SELECT clave, descripcion
199     FROM materiales
200    WHERE descripcion LIKE '%ub%';
201
100%  ◊ | 30:195 |
```

Result Grid	
clave	descripcion
1180	Recubrimiento P1001
1190	Recubrimiento P1010
1200	Recubrimiento P1019
1210	Recubrimiento P1028
1220	Recubrimiento P1037
1290	Tubería 3.5
1300	Tubería 4.3
1310	Tubería 3.6
1320	Tubería 4.4
1330	Tubería 3.7
1340	Tubería 4.5
1350	Tubería 3.8
NULL	NULL

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

```

202 •   SELECT p.denominacion, SUM(m.precio * e.cantidad) AS Total
203     FROM proyectos p, entregan e, materiales m
204    WHERE p.numero = e.numero AND e.clave = m.clave
205    GROUP BY p.denominacion;

```

100% ▾ 31:200 |

Result Grid Filter Rows: Search Export:

denominacion	Total
Vamos Mexico	106730
Aztecon	146595
CIT Campeche	157755
Mexico sin ti no estamos completos	260290

8. 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).

```

207 •   CREATE VIEW Televisa AS
208     SELECT DISTINCT rfc
209     FROM entregan
210    Ⓛ WHERE numero = (
211        SELECT numero
212        FROM proyectos
213       WHERE denominacion = 'Televisa en accion');
214
215 •   CREATE VIEW Educando_Coahuila AS
216     SELECT DISTINCT rfc
217     FROM entregan
218    Ⓛ WHERE numero = (
219        SELECT numero
220        FROM proyectos
221       WHERE denominacion = 'Educando en Coahuila');
222

```

100% ▾ 21:216 |

Result Grid Filter Rows: Search Export:

rfc	razonsocial
DDDD800101	Cecoferre
CCCC800101	La Ferre

9. 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).

```

232 •   SELECT DISTINCT pr.rfc, pr.razonsocial
233     FROM proveedores pr
234    Ⓛ WHERE pr.rfc IN (
235        SELECT DISTINCT e.rfc
236        FROM entregan e
237    Ⓛ WHERE e.numero = (
238        SELECT numero
239        FROM proyectos
240       WHERE denominacion = 'Televisa en accion'
241      )
242    ) AND pr.rfc NOT IN (
243        SELECT DISTINCT e.rfc
244        FROM entregan e
245    Ⓛ WHERE e.numero = (
246        SELECT numero
247        FROM proyectos
248       WHERE denominacion = 'Educando en Coahuila'
249      )
250  );
251

```

rfc	razonsocial
CCCC800101	La Ferre
DDDD800101	Cecoferre
HULL	HULL

10. 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.

```
252 •  SELECT m.clave, m.descripcion, m.precio
253   FROM materiales m, entregan e1, entregan e2, proyectos p1, proyectos p2
254  WHERE m.clave = e1.clave
255    AND m.clave = e2.clave
256    AND p1.numero = e1.numero
257    AND p2.numero = e2.numero
258    AND p1.denominacion = 'Televisa en acción'
259    AND p2.denominacion = 'Educando en Coahuila';
260
```

100% ◁ 11:245 |

Result Grid Filter Rows: Search Export:

clave	descripcion	precio

En SQL es posible crear alias de una misma tabla para instanciarla más de una vez, en este caso se hace por que se requiere saber de los proyectos 'Educando en Coahuila' y 'Televisa en acción'.

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

```
261 •  SELECT m.descripcion AS nombreMaterial, COUNT(e.clave) AS entregado, SUM(e.cantidad * m.precio) AS Total
262   FROM materiales m, entregan e
263  WHERE m.clave = e.clave
264  GROUP BY m.descripcion;
265
```

100% ◁ 44:258 |

Result Grid Filter Rows: Search Export:

nombreMaterial	entregado	Total
Varilla 3/16	2	41900
Varilla 4/32	2	120865
Varilla 3/17	2	76700
Varilla 4/33	2	72065