Construcción de Software para la Toma de Decisiones

Laboratorio 20: Consultas en SQL.

Santiago Palacios Menes | A01276169

9 Septiembre de 2024

}

Consulta de un tabla completa

Algebra relacional.
materiales

SQL
select * from materiales

Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0007 segundos.)

```
select * from materiales;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códi

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

←T→			clave	descripcion	precio	impuesto
Editar	Copiar	Borrar	1000	Varilla 3/16	100	10
Editar	Copiar	Borrar	1010	Varilla 4/32	115	11.5
Editar	Copiar	Borrar	1020	Varilla 3/17	130	13
Editar	Copiar	Borrar	1030	Varilla 4/33	145	14.5
Editar	Copiar	Borrar	1040	Varilla 3/18	160	16
Editar	Copiar	Borrar	1050	Varilla 4/34	175	17.5
Editar	Copiar	Borrar	1060	Varilla 3/19	190	19
Editar	Copiar	Borrar	1070	Varilla 4/35	205	20.5
Editar	Copiar	Borrar	1080	Ladrillos rojos	50	5
Editar	Copiar	Borrar	1090	Ladrillos grises	35	3.5
Editar	Copiar	Borrar	1100	Block	30	3
Editar	Copiar	Borrar	1110	Megablock	40	4
Editar	Copiar	Borrar	1120	Sillar rosa	100	10
Editar	Copiar	Borrar	1130	Sillar gris	110	11

Selección

Algebra relacional.
SL{clave=1000}(materiales)

SQL
select * from materiales
where clave=1000

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0007 segundos.)

```
select * from materiales where clave=1000;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

	clave	descripcion	precio	impuesto
Editar Copiar Borrar	1000	Varilla 3/16	100	10

Proyección

Algebra relacional.
PR{clave,rfc,fecha} (entregan)

SQL
select clave,rfc,fecha from entregan

Servidor: 127.0.0.1 » Base de datos: lab » Tabla: entregan

Examinar | Estructura | SQL | Buscar | Insertar | Exportar | Importar

Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0006 segundos.)

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códi

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

←T→			clave	rfc	fecha
Editar	Copiar	Borrar	1000	AAAA800101	2001-12-13
Editar	Copiar	Borrar	1200	EEEE800101	2003-03-15
Editar	Copiar	Borrar	1400	AAAA800101	1999-04-07
Editar	Copiar	Borrar	1010	BBBB800101	1998-07-28
Editar	Copiar	Borrar	1210	FFFF800101	2000-05-21
Editar	Copiar	Borrar	1410	BBBB800101	2000-05-18
Editar	Copiar	Borrar	1020	CCCC800101	2003-12-16
Editar	Copiar	Borrar	1220	GGGG800101	2005-07-03
Editar	Copiar	Borrar	1420	CCCC800101	2001-09-09
Editar	Copiar	Borrar	1030	DDDD800101	1998-01-12
Editar	Copiar	Borrar	1230	HHHH800101	1998-09-12
Editar	Copiar	Borrar	1430	DDDD800101	2005-04-30
Editar	Copiar	Borrar	1040	EEEE800101	1999-12-18
Editar	Copiar	Borrar	1150	HHHH800101	2002-11-14
Editar	Copiar	Borrar	1240	AAAA800101	0000-00-00
Editar	Copiar	Borrar	1050	FFFF800101	2004-02-07

Consola

Reunión Natural

Algebra relacional.

entregan JN materiales

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

Mostrando filas 0 - 24 (87 en total, 0 en la consulta, La consulta tardó 0,0010 segundos.)

```
select * from materiales,entregan where materiales.clave = entregan.clave;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

clave	descripcion	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	BBBB800101	5001	1998-07-28	528
1010	Varilla 4/32	115	11.5	1010	BBBB800101	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 rojos	50	5	1080	AAAA800101	5008	2005-04-03	86
1080	Ladrillos rojos	50	5	1080	AAAA800101	5011	2003-08-28	699
1090	Ladrillos grises	35	3.5	1090	BBBB800101	5009	1997-03-13	73
1090	Ladrillos grises	35	3.5	1090	BBBB800101	5010	1998-11-17	421
[illegible]	[illegible]	30	3	1100	CCCC800101	5009	2000-12-07	466

Consola

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

No, pues estan pidiendo los materiales que esten en entregado

Reunión con criterio específico

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

SQL
select * from entregan,proyectos
where entregan.numero < = proyectos.numero

Mostrando filas 0 - -1 (total de 0, La consulta tardó 0,0020 segundos.)

```
SELECT * FROM entregan, proyectos WHERE entregan.numero <= proyectos.numero;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

clave	rfc	numero	fecha	cantidad	numero	denominacion
1000	AAAA800101	5000	2001-12-13	165	5000	Vamos Mexico
1200	EEEE800101	5000	2003-03-15	177	5000	Vamos Mexico
1400	AAAA800101	5000	1999-04-07	382	5000	Vamos Mexico
1000	AAAA800101	5000	2001-12-13	165	5001	Aztecon
1200	EEEE800101	5000	2003-03-15	177	5001	Aztecon
1400	AAAA800101	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	AAAA800101	5000	2001-12-13	165	5002	CIT Campeche
1200	EEEE800101	5000	2003-03-15	177	5002	CIT Campeche
1400	AAAA800101	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	AAAA800101	5000	2001-12-13	165	5003	Mexico sin ti no estamos completos
1200	EEEE800101	5000	2003-03-15	177	5003	Mexico sin ti no estamos completos
1400	AAAA800101	5000	1999-04-07	382	5003	Mexico sin ti no estamos completos

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)

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0008 segundos.)

```
(select * from entregan where clave=1450) union (select * from entregan where clave=1300);
```

[Editar en línea] [Editar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave:

Opciones extra

clave	rfc	numero	fecha	cantidad
1300	GGGG800101	5005	2004-02-28	521
1300	GGGG800101	5010	2001-02-10	119

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

SELECT *
FROM entregan
WHERE clave IN (1450, 1300);

da lo mismo pues quieres la clave de 1450 o 1300

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0006 segundos.)

```
SELECT * FROM entregan WHERE clave IN (1450, 1300);
```

Perfilando [Editar en línea] [Editar] [Explicar

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Orde

Opciones extra

				clave	rfc	numero	fecha	cantidad
Editar	Copiar	Borrar		1300	GGGG800101	5005	2004-02-28	521
Editar	Copiar	Borrar		1300	GGGG800101	5010	2001-02-10	119

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)
```

Adaptado a PhpMyadmin

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

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0014 segundos.)

```
SELECT clave FROM entregan WHERE numero = 5001 AND clave IN (SELECT clave FROM entregan WHERE numero
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

←T→	clave
Editar Copiar Borrar	1010

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 NOT IN (SELECT clave FROM entregan WHERE clave = 1000);

NOT IN sustituye a Minus

```
SELECT * FROM entregan WHERE clave NOT IN (SELECT clave FROM entregan WHERE clave = 1000);
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

←T→	clave	rfc	numero	fecha	cantidad
Editar Copiar Borrar	1010	BBBB800101	5001	1998-07-28	528
Editar Copiar Borrar	1010	BBBB800101	5018	1997-02-09	523
Editar Copiar Borrar	1020	CCCC800101	5002	2003-12-16	582
Editar Copiar Borrar	1020	CCCC800101	5017	2000-03-29	8
Editar Copiar Borrar	1030	DDDD800101	5003	1998-01-12	202
Editar Copiar Borrar	1030	DDDD800101	5016	2005-06-07	295
Editar Copiar Borrar	1040	EEEE800101	5004	1999-12-18	263
Editar Copiar Borrar	1040	EEEE800101	5015	1999-05-29	540
Editar Copiar Borrar	1050	FFFF800101	5005	2004-02-07	503
Editar Copiar Borrar	1050	FFFF800101	5014	2000-04-18	623
Editar Copiar Borrar	1060	GGGG800101	5006	2004-09-29	324
Editar Copiar Borrar	1060	GGGG800101	5013	2002-07-23	692
Editar Copiar Borrar	1070	HHHH800101	5007	2006-11-21	2
Editar Copiar Borrar	1070	HHHH800101	5012	2004-11-27	503
Editar Copiar Borrar	1080	AAAA800101	5008	2005-04-03	86
Editar Copiar Borrar	1080	AAAA800101	5011	2003-08-28	699
Editar Copiar Borrar	1090	BBBB800101	5009	1997-03-13	73

Producto cartesiano

Algebra relacional.
entregan X materiales

SQL

select * from entregan,materiales

Mostrando filas 0 - 24 (87 en total, 0 en la consulta, La consulta tardó 0,0009 segundos.)

```
select * from entregan,materiales;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
1000	AAAA800101	5000	2001-12-13	165	2000	Jabón	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1430	Pintura B1022	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1420	Pintura C1012	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1410	Pintura B1021	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1400	Pintura C1011	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1390	Pintura B1021	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1380	Pintura C1011	725	72.5
1000	AAAA800101	5000	2001-12-13	165	1370	Pintura B1020	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1360	Pintura C1010	125	12.5
1000	AAAA800101	5000	2001-12-13	165	1350	Tubería 3.8	260	26
1000	AAAA800101	5000	2001-12-13	165	1340	Tubería 4.5	250	25
1000	AAAA800101	5000	2001-12-13	165	1330	Tubería 3.7	240	24
1000	AAAA800101	5000	2001-12-13	165	1320	Tubería 4.4	230	23
1000	AAAA800101	5000	2001-12-13	165	1310	Tubería 3.6	220	22
1000	AAAA800101	5000	2001-12-13	165	1300	Tubería 4.3	210	21
1000	AAAA800101	5000	2001-12-13	165	1290	Tubería 3.5	200	20
1000	AAAA800101	5000	2001-12-13	165	1280	Tepetate	34	3.4
1000	AAAA800101	5000	2001-12-13	165	1270	Tezontle	80	8
1000	AAAA800101	5000	2001-12-13	165	1260	Gravilla	90	9
Consola	AA800101	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 entrega y de materiales?

el numero de la tupla es de el producto de las filas en cada tabla

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.

```
SELECT m.descripcion
FROM entregan e
JOIN materiales m ON e.clave = m.clave
WHERE YEAR(e.fecha) = 2000;
```

```
SELECT m.descripcion FROM entregan e JOIN materiales m ON e.clave = m.clave WHERE YEAR(e.fecha) = 200
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear códig

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

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

¿Por qué aparecen varias veces algunas descripciones de material?

no hay un distinct para eliminar duplicados

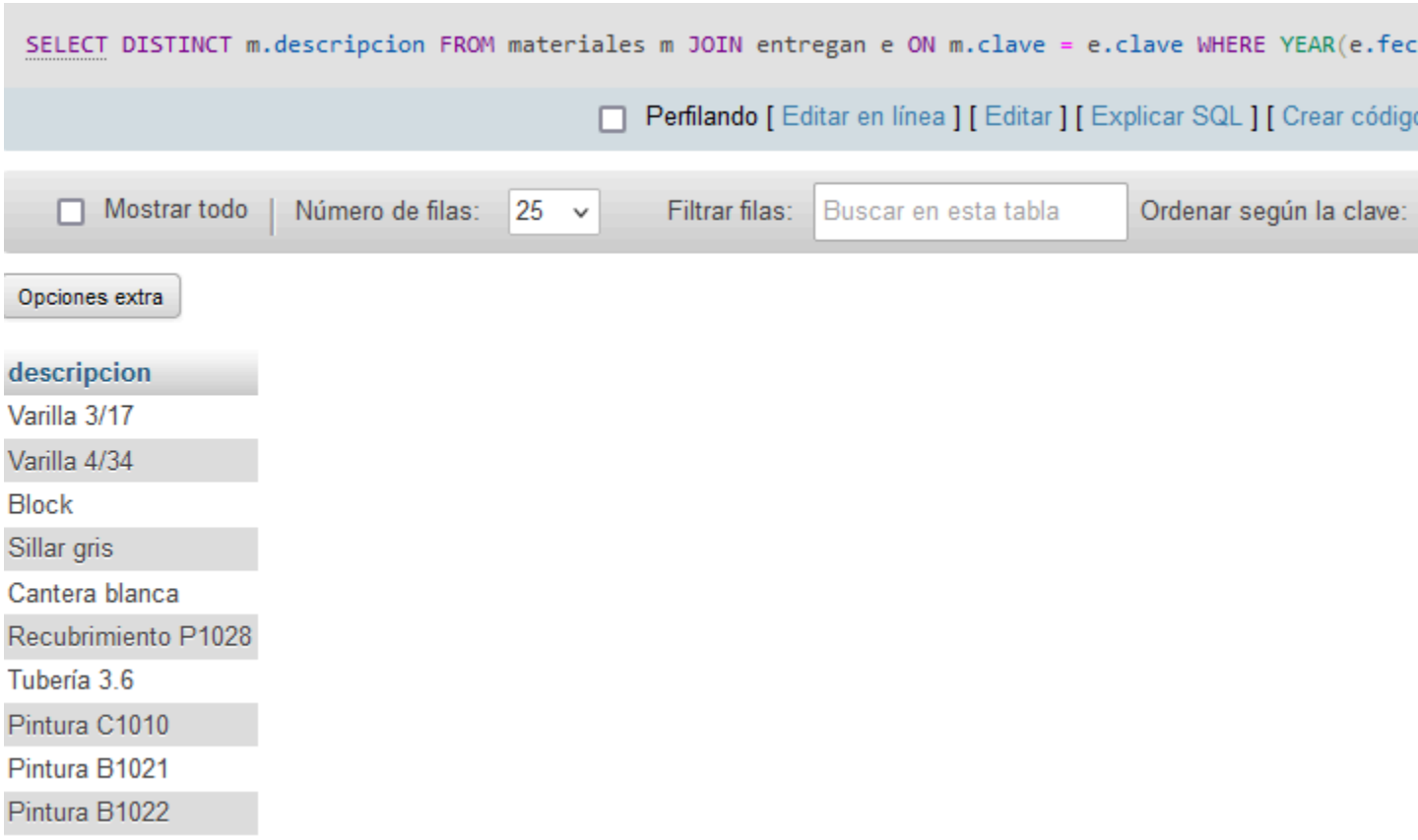
Uso del calificador distinct

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.

SET dateformat dmy;

SELECT DISTINCT m.descripcion
FROM materiales m
JOIN entregan e ON m.clave = e.clave
WHERE YEAR(e.fecha) = 2000;



¿Qué resultado obtienes en esta ocasión?

El mismo que el anterior pero ahora ya no hay duplicados

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.numero AS numero_proyecto, p.denominacion, e.fecha, e.cantidad
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
ORDER BY p.numero, e.fecha DESC;
```

Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0022 segundos.) [fecha: **2003-03-15... - 2005-06-06...**]

```
SELECT p.numero AS numero_proyecto, p.denominacion, e.fecha, e.cantidad FROM proyectos p JOIN entreg
p.numero = e.numero ORDER BY p.numero, e.fecha DESC;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

numero_proyecto	denominacion	fecha	cantidad
5000	Vamos Mexico	2003-03-15	177
5000	Vamos Mexico	2001-12-13	165
5000	Vamos Mexico	1999-04-07	382
5001	Aztecon	2000-05-21	43
5001	Aztecon	2000-05-18	601
5001	Aztecon	1998-07-28	528
5002	CIT Campeche	2005-07-03	24
5002	CIT Campeche	2003-12-16	582
5002	CIT Campeche	2001-09-09	603
5003	Mexico sin ti no estamos completos	2005-04-30	576
5003	Mexico sin ti no estamos completos	1998-09-12	530
5003	Mexico sin ti no estamos completos	1998-01-12	202
5004	Educando en Coahuila	2002-11-14	453
5004	Educando en Coahuila	1999-12-18	263

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 materiales where Descripcion LIKE 'Si%'

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0004 segundos.)

```
SELECT * FROM materiales WHERE Descripcion LIKE 'Si%' LIMIT 0, 25;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

Opciones extra

←T→			clave	descripcion	precio	impuesto
Editar	Copiar	Borrar	1120	Sillar rosa	100	10
Editar	Copiar	Borrar	1130	Sillar gris	110	11

¿Qué resultado obtienes?
tengo 2 materiales que inicia con “SI”
Explica que hace el símbolo '%'.
hace que puedas escribir lo que sea después o antes del %
¿Qué sucede si la consulta fuera : LIKE 'Si' ?
No saldria nada pues no hay materiales que se llamen SI
¿Qué resultado obtienes?
nada
Explica a qué se debe este comportamiento.
debido a que le estas pidiendo materiales que se llamen Si en lugar de Productos que se llamen Sí% (% siendo lo que sea despues)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0004 segundos.)

```
SELECT * FROM materiales WHERE Descripcion LIKE 'Si';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

clave	descripcion	precio	impuesto

Operaciones sobre los resultados de la consulta

Crear vista

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:

```
SET @foo = '¿Qué resultado';
SET @bar = ' ¿¿¿???';

SET @foo = CONCAT(@foo, ' obtienes?');
SELECT CONCAT(@foo, @bar) AS Resultado;
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

```
SET @foo = '¿Qué resultado';
```

[Editar en línea] [Editar] [

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

```
SET @bar = ' ¿¿¿???';
```

[Editar en línea] [Editar] [

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

```
SET @foo = CONCAT(@foo, ' obtienes?');
```

[Editar en línea] [Editar] [

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edi
disponibles.

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0002 segundos.)

```
SELECT CONCAT(@foo, @bar) AS Resultado;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

Resultado

¿Qué resultado obtienes? ¿¿¿???

¿Qué resultado obtienes de ejecutar el siguiente código?
el texto que FLO guardo
¿Para qué sirve DECLARE?
Declara variables
¿Cuál es la función de @foo?
Almacena y Manipula caracteres
¿Que realiza el operador SET?
Asigna un valor auna variable

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]%';
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0008 segundos.)

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

RFC

Operaciones sobre los resultados de la consulta

Crear vista

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0003 segundos.)

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

RFC

Operaciones sobre los resultados de la consulta

Crear vista

Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0002 segundos.)

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0002 segundos.)

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave:

Opciones extra

←T→			Numero
Editar	Copiar	Borrar	5006
Editar	Copiar	Borrar	5006
Editar	Copiar	Borrar	5006
Editar	Copiar	Borrar	5006
Editar	Copiar	Borrar	5016
Editar	Copiar	Borrar	5016
Editar	Copiar	Borrar	5016
Editar	Copiar	Borrar	5016
Editar	Copiar	Borrar	5016

RFC LIKE selecciona valores en el RFC que comienza entre A y D, después quita los valores que inicen con la letra A y selecciona el numero que tiene 6 caracteres

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;
```

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

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

				Clave	RFC	Numero	Fecha	Cantidad
Editar	Copiar	Borrar		1000	AAAA800101	5000	2001-12-13	165
Editar	Copiar	Borrar		1200	EEEE800101	5000	2003-03-15	177
Editar	Copiar	Borrar		1400	AAAA800101	5000	1999-04-07	382
Editar	Copiar	Borrar		1010	BBBB800101	5001	1998-07-28	528
Editar	Copiar	Borrar		1210	FFFF800101	5001	2000-05-21	43
Editar	Copiar	Borrar		1410	BBBB800101	5001	2000-05-18	601
Editar	Copiar	Borrar		1020	CCCC800101	5002	2003-12-16	582
Editar	Copiar	Borrar		1220	GGGG800101	5002	2005-07-03	24
Editar	Copiar	Borrar		1420	CCCC800101	5002	2001-09-09	603
Editar	Copiar	Borrar		1030	DDDD800101	5003	1998-01-12	202
Editar	Copiar	Borrar		1230	HHHH800101	5003	1998-09-12	530
Editar	Copiar	Borrar		1430	DDDD800101	5003	2005-04-30	576
Editar	Copiar	Borrar		1040	EEEE800101	5004	1999-12-18	263
Editar	Copiar	Borrar		1150	HHHH800101	5004	2002-11-14	453
Editar	Copiar	Borrar		1240	AAAA800101	5004	0000-00-00	152
Editar	Copiar	Borrar		1050	FFFF800101	5005	2004-02-07	503
Editar	Copiar	Borrar		1140	GGGG800101	5005	2000-06-30	583
Consola	itar	Copiar	Borrar	1250	BBBB800101	5005	2003-10-12	71

¿Cómo filtrarías rangos de fechas?

usando Bewtewwn o los signos >= o <=

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])

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0012 segundos.)

```
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 );
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	RFC	Cantidad	Fecha	Numero
Editar Copiar Borrar	AAAA800101	165	2001-12-13	5000
Editar Copiar Borrar	AAAA800101	86	2005-04-03	5008
Editar Copiar Borrar	AAAA800101	152	0000-00-00	5004
Editar Copiar Borrar	AAAA800101	382	1999-04-07	5000
Editar Copiar Borrar	AAAA800101	116	2005-04-21	5010
Editar Copiar Borrar	CCCC800101	582	2003-12-16	5002
Editar Copiar Borrar	CCCC800101	466	2000-12-07	5009
Editar Copiar Borrar	CCCC800101	699	2001-11-19	5010
Editar Copiar Borrar	CCCC800101	460	2001-04-09	5006
Editar Copiar Borrar	CCCC800101	631	2001-07-28	5009
Editar Copiar Borrar	CCCC800101	603	2001-09-09	5002
Editar Copiar Borrar	CCCC800101	278	1999-05-05	5008

¿Qué hace la consulta?

da la cantidad, RFC, Número y Fecha donde el número sea 5000 a 5010 y la razón social que inicien con La.

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

El paréntesis despues de EXIST significa una subconsuklta de una condicion)

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

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.

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 IN

```
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%'
);
```

ANY

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

TOP

```
SELECT TOP 5 RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 5000 AND 5010
ORDER BY Fecha DESC;
```

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

¿Qué hace la siguiente sentencia? Explica por qué.

Consulta

SELECT TOP 2 * FROM Proyectos

SELECT * FROM Proyectos LIMIT 2;

Usa Top 2 para limitar el resultado a 2 filas seleccionados

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

SELECT TOP Numero FROM Proyectos
SELECT Numero FROM Proyectos LIMIT 2;

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0004 segundos.)

```
SELECT Numero FROM Proyectos LIMIT 2;
```

Perfilando [

Opciones extra

←T→			Numero
Editar	Copiar	Borrar	5000
Editar	Copiar	Borrar	5001

Usa Top 2 para limitar el resultado a 2 filas seleccionados y solo mostrando numeros en lugar de todos los datos

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);

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0457 segundos.)

```
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.

45 filas afectadas. (La consulta tardó 0,0116 segundos.)

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

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.Numero, e.Cantidad, m.Precio, m.PorcentajeImpuesto,
        (e.Cantidad * m.Precio * (1 + m.PorcentajeImpuesto / 100)) AS ImporteTotal
FROM Entregan e
JOIN Materiales m ON e.ClaveMaterial = 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.

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".

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 = 'Mexico sin ti no estamos completos';

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0009 segundos.)

```
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 = 'Mexico sin ti no estamos completos';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ningun

Opciones extra

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

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

No hay ACME Tools

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0008 segundos.)

```
SELECT m.clave, m.descripcion FROM materiales m JOIN entregan e ON m.clave = e.clave JOIN proveedores p
ON e.rfc = p.rfc WHERE p.razonSocial = 'Acme tools';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

clave	descripcion

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

SELECT e.rfc

FROM entregan e
WHERE YEAR(e.fecha) = 2000
GROUP BY e.rfc
HAVING AVG(e.cantidad) >= 300;

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0011 segundos.)

```
SELECT e.rfc FROM entregan e WHERE YEAR(e.fecha) = 2000 GROUP BY e.rfc HAVING AVG(e.cantidad) >= 300;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ningun

Opciones extra

←T→			rfc
Editar	Copiar	Borrar	BBBB800101
Editar	Copiar	Borrar	FFFF800101
Editar	Copiar	Borrar	GGGG800101

El Total entregado por cada material en el año 2000.

SELECT m.clave, SUM(e.cantidad) AS total_entregado FROM materiales m JOIN entregan e ON m.clave = e.clave WHERE YEAR(e.fecha) = 2000 GROUP BY m.clave;

Mostrando filas 0 - 10 (total de 11, La consulta tardó 0,0007 segundos.)

```
SELECT m.clave, SUM(e.cantidad) AS total_entregado FROM materiales m JOIN entregan e ON m.clave = e.clave
WHERE YEAR(e.fecha) = 2000 GROUP BY m.clave;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

clave	total_entregado
1020	8
1050	623
1100	466
1130	625
1140	583
1210	43
1310	72
1360	265
1390	107
1410	601
1430	13

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

```
DROP VIEW IF EXISTS MaterialVentas2001;

CREATE VIEW MaterialVentas2001 AS
SELECT m.clave, SUM(e.cantidad) AS total_vendido
FROM materiales m
JOIN entregan e ON m.clave = e.clave
WHERE YEAR(e.fecha) = 2001
GROUP BY m.clave;

SELECT * FROM MaterialVentas2001;
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0044 segundos.)

```
CREATE VIEW MaterialVentas2001 AS SELECT m.clave, SUM(e.cantidad) AS total_vendido FROM materiales m JOIN
entregan e ON m.clave = e.clave WHERE YEAR(e.fecha) = 2001 GROUP BY m.clave;
```

[Editar en línea] [Editar] [Crear código PHP]

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 6 (total de 7, La consulta tardó 0,0016 segundos.)

```
SELECT * FROM MaterialVentas2001;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

clave	total_vendido
1000	165
1100	699
1180	407
1260	1091
1300	119
1330	554
1420	603

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

```
SELECT clave, descripcion
FROM materiales
WHERE descripcion LIKE '%ub%';
```

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0005 segundos.)

```
SELECT clave, descripcion FROM materiales WHERE descripcion LIKE '%ub%';
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ningun

Opciones extra

←T→			clave	descripcion
Editar	Copiar	Borrar	1180	Recubrimiento P1001
Editar	Copiar	Borrar	1190	Recubrimiento P1010
Editar	Copiar	Borrar	1200	Recubrimiento P1019
Editar	Copiar	Borrar	1210	Recubrimiento P1028
Editar	Copiar	Borrar	1220	Recubrimiento P1037
Editar	Copiar	Borrar	1290	Tubería 3.5
Editar	Copiar	Borrar	1300	Tubería 4.3
Editar	Copiar	Borrar	1310	Tubería 3.6
Editar	Copiar	Borrar	1320	Tubería 4.4
Editar	Copiar	Borrar	1330	Tubería 3.7
Editar	Copiar	Borrar	1340	Tubería 4.5
Editar	Copiar	Borrar	1350	Tubería 3.8

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

```
SELECT p.denominacion, SUM(e.cantidad * m.precio) AS total_pagar
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
JOIN materiales m ON e.clave = m.clave
GROUP BY p.denominacion
LIMIT 0, 25;
```

Mostrando filas 0 - 19 (total de 20, La consulta tardó 0,0012 segundos.)

```
SELECT p.denominacion, SUM(e.cantidad * m.precio) AS total_pagar FROM proyectos p JOIN entregan e ON
p.numero = e.numero JOIN materiales m ON e.clave = m.clave GROUP BY p.denominacion LIMIT 0, 25;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Opciones extra

denominacion	total_pagar
Vamos Mexico	106730
Aztecon	146595
CIT Campeche	157755
Mexico sin ti no estamos completos	260290
Educando en Coahuila	620610
Infonavit Durango	321135
Reconstrucción del templo de Guadalupe	220580
Construcción de plaza Magnolias	122969
Televisa en acción	99848
Disco Atlantic	158100
Construcción de Hospital Infantil	144295
Remodelación de aulas del IPP	225835
Restauración de instalaciones del CEA	846380
Reparación de la plaza Sonora	527485
Remodelación de Soriana	284450
CIT Yucatan	851700
Ampliación de la carretera a la huasteca	565135
Reparación de la carretera del sol	137060
Tu cambio por la educación	467675
Queretaro limpio	449715

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).

```
CREATE VIEW ProveedoresTelevisa AS
SELECT DISTINCT pr.rfc, pr.razonsocial
FROM entregan e
JOIN proveedores pr ON e.rfc = pr.rfc
JOIN proyectos p ON e.numero = p.numero
WHERE p.denominacion = 'Televisa en acción';

CREATE VIEW ProveedoresEducando AS
SELECT DISTINCT pr.rfc, pr.razonsocial
FROM entregan e
JOIN proveedores pr ON e.rfc = pr.rfc
JOIN proyectos p ON e.numero = p.numero
```

WHERE p.denominacion = 'Educando en Coahuila';

SELECT pt.rfc, pt.razonsocial
FROM ProveedoresTelevisa pt
LEFT JOIN ProveedoresEducando pe ON pt.rfc = pe.rfc
WHERE pe.rfc IS NULL;

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0107 segundos.)

```
CREATE VIEW ProveedoresTelevisa AS SELECT DISTINCT pr.rfc, pr.razonsocial FROM entregan e JOIN
proveedores pr ON e.rfc = pr.rfc JOIN proyectos p ON e.numero = p.numero WHERE p.denominacion = 'Televisa
en acción';
```

[Editar en línea] [Editar] [Crear código PHP]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0035 segundos.)

```
CREATE VIEW ProveedoresEducando AS SELECT DISTINCT pr.rfc, pr.razonsocial FROM entregan e JOIN
proveedores pr ON e.rfc = pr.rfc JOIN proyectos p ON e.numero = p.numero WHERE p.denominacion = 'Educando
en Coahuila';
```

[Editar en línea] [Editar] [Crear código PHP]

La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0030 segundos.)

```
SELECT pt.rfc, pt.razonsocial FROM ProveedoresTelevisa pt LEFT JOIN ProveedoresEducando pe ON pt.rfc =
pe.rfc WHERE pe.rfc IS NULL;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

rfc	razonsocial
DDDD800101	Cecoferre
CCCC800101	La Ferre

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).

SELECT p.denominacion, e.rfc, pr.razonsocial

```
FROM entregan e
JOIN proyectos p ON e.numero = p.numero
JOIN proveedores pr ON e.rfc = pr.rfc
WHERE p.denominacion = 'Televisa en acción'
AND e.rfc NOT IN (
        SELECT e2.rfc
        FROM entregan e2
        JOIN proyectos p2 ON e2.numero = p2.numero
        WHERE p2.denominacion = 'Educando en Coahuila'
);
```

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0012 segundos.)

```
SELECT p.denominacion, e.rfc, pr.razonsocial FROM entregan e JOIN proyectos p ON e.numero = p.numero JOIN
proveedores pr ON e.rfc = pr.rfc WHERE p.denominacion = 'Televisa en acción' AND e.rfc NOT IN ( SELECT
e2.rfc FROM entregan e2 JOIN proyectos p2 ON e2.numero = p2.numero WHERE p2.denominacion = 'Educando en
Coahuila' );
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

denominacion	rfc	razonsocial
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	CCCC800101	La Ferre

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.

```
SELECT m.clave, m.descripcion, m.precio AS costo
FROM materiales m
JOIN entregan e1 ON m.clave = e1.clave
JOIN proyectos p1 ON e1.numero = p1.numero
WHERE p1.denominacion = 'Televisa en acción'
AND EXISTS (
        SELECT 1
        FROM entregan e2
        JOIN proyectos p2 ON e2.numero = p2.numero
        WHERE e2.rfc = e1.rfc
        AND p2.denominacion = 'Educando en Coahuila'
);
```

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0017 segundos.)

```
SELECT m.clave, m.descripcion, m.precio AS costo FROM materiales m JOIN entregan e1 ON m.clave = e1.clave
JOIN proyectos p1 ON e1.numero = p1.numero WHERE p1.denominacion = 'Televisa en acción' AND EXISTS (
SELECT 1 FROM entregan e2 JOIN proyectos p2 ON e2.numero = p2.numero WHERE e2.rfc = e1.rfc AND
p2.denominacion = 'Educando en Coahuila' );
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ningu

Opciones extra

clave	descripcion	costo
1080	Ladrillos rojos	50
1280	Tepetate	34

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.

```
SELECT m.descripcion AS nombre_material,
        COUNT(e.clave) AS cantidad_entregas,
        SUM(e.cantidad * m.precio) AS total_costo
FROM materiales m
JOIN entregan e ON m.clave = e.clave
GROUP BY m.descripcion;
```

```sql
SELECT m.descripcion AS nombre_material, COUNT(e.clave) AS cantidad_entregas, SUM(e.cantidad * m.precio)
AS total_costo FROM materiales m JOIN entregan e ON m.clave = e.clave GROUP BY m.descripcion;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

nombre_material	cantidad_entregas	total_costo
Varilla 3/16	2	41900
Varilla 4/32	2	120865
Varilla 3/17	2	76700
Varilla 4/33	2	72065
Varilla 3/18	2	128480
Varilla 4/34	2	197050
Varilla 3/19	2	193040
Varilla 4/35	2	103525
Ladrillos rojos	2	39250
Ladrillos grises	2	17290
Block	2	34950
Megablock	2	28200
Sillar rosa	2	90700
Sillar gris	2	68750
Cantera blanca	2	160400
Cantera gris	2	1102310
Cantera rosa	2	576520
Cantera amarilla	2	53590
Recubrimiento P1001	2	148200
Recubrimiento P1010	2	99000
Recubrimiento P1019	2	182880