Consulta de un tabla completa

Algebra relacional.
materiales

SQL
select * from materiales

Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0008 segundos.)

```
SELECT* FROM materiales;
```

[Editar en línea] [Editar] [Crea

1 > >> | Mostrar todo | Número de filas: 25

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
Consola	itar	Copiar	Borrar	1120	Sillar rosa	100	10

Selección

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

SQL
select * from materiales
where clave=1000

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

```
SELECT* FROM materiales WHERE clave = 1000;
```

[Editar en línea] [Editar] [Crea

Mostrar todo | Número de filas: 25

Opciones extra

←T→	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

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

```
SELECT clave, rfc, fecha FROM entregan;
```

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

1 > >> | Mostrar todo | Número de filas: 25

Opciones extra

←T→			clave	rfc	fecha
Editar	Copiar	Borrar	1000	AAAA800101	2001-12-13
Editar	Copiar	Borrar	1000	AAAA800101	1999-07-13
Editar	Copiar	Borrar	1010	BBBB800101	1998-07-28
Editar	Copiar	Borrar	1010	BBBB800101	1997-02-09
Editar	Copiar	Borrar	1020	CCCC800101	2003-12-16
Editar	Copiar	Borrar	1020	CCCC800101	2000-03-29
Editar	Copiar	Borrar	1030	DDDD800101	1998-01-12
Editar	Copiar	Borrar	1030	DDDD800101	2005-06-07
Editar	Copiar	Borrar	1040	EEEE800101	1999-12-18
Editar	Copiar	Borrar	1040	EEEE800101	1999-05-29

Reunión Natural

Algebra relacional.
entregan JN materiales

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

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

- No

Mostrando filas 0 - -1 (total de 0, La consulta tardó 0,0010 segundos.)

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

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

Mostrar todo | Número de filas: 25

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

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 0,0016 segundos.)

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

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

Mostrar todo | Número de filas: 25

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

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 0,0065 segundos.)

```
SELECT* FROM entregan WHERE clave = 1450 UNION SELECT* FROM entregan WHERE clave = 1300;
```

[Editar en línea] [Editar] [Crear código P

Mostrar todo | Número de filas: 25

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.

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

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

[Editar en línea] [Editar] [Crear c

Mostrar todo | Número de filas: 25

Opciones extra

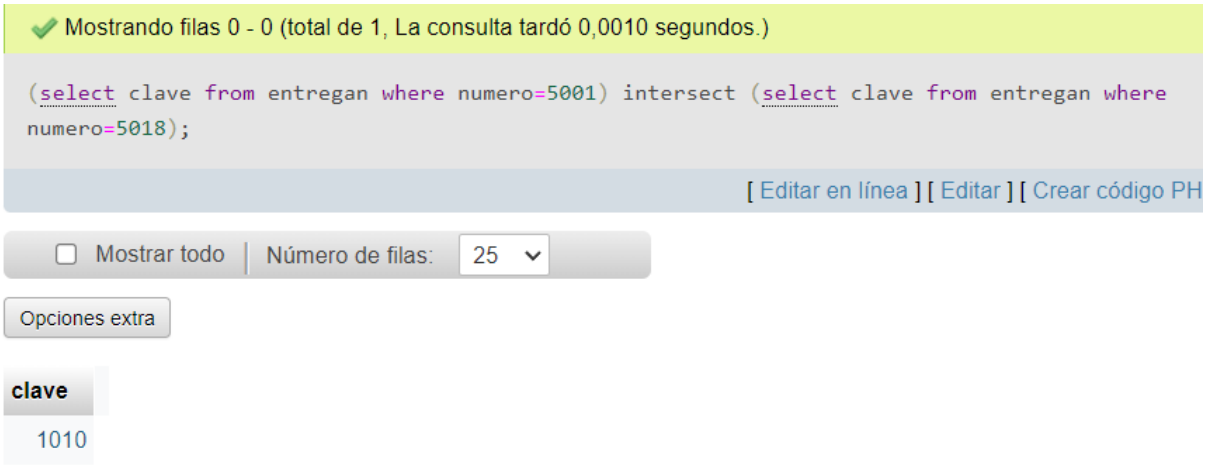
←T→	clave	rfc	numero	fecha	cantidad
Editar Copiar Borrar	1300	GGGG800101	5005	2004-02-28	521
Editar Copiar Borrar	1300	GGGG800101	5010	2001-02-10	119

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)



Diferencia (se ilustra con selección)

Algebra relacional.
entregan - SL{clave=1000}(entregan)

SQL
(select * from entregan)
EXCEPT
(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.

Mostrando filas 0 - 24 (total de 85, La consulta tardó 0,0016 segundos.)

```
select * from entregan EXCEPT (select * from entregan where clave=100
```

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

1 > >> | Mostrar todo | Número de filas: 25

Opciones extra

clave	rfc	numero	fecha	cantidad
1010	BBBB800101	5001	1998-07-28	528
1010	BBBB800101	5018	1997-02-09	523
1020	CCCC800101	5002	2003-12-16	582
1020	CCCC800101	5017	2000-03-29	8
1030	DDDD800101	5003	1998-01-12	202
1030	DDDD800101	5016	2005-06-07	295
1040	EEEE800101	5004	1999-12-18	263
1040	EEEE800101	5015	1999-05-29	540
1050	FFFF800101	5005	2004-02-07	503
1050	FFFF800101	5014	2000-04-18	623
[illegible]	GGG800101	5006	2004-09-29	324

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?
La interfaz tiene un error pero deberia ser 87*45 = 3915

Mostrando filas 0 - -1 (total de 0, La consulta tardó 0,0009 segundos.)

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

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

Mostrar todo | Número de filas: 25

Opciones extra

clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
1000	AAAA800101	5000	2001-12-13	165	1000	Varilla 3/16	100	10
1000	AAAA800101	5000	2001-12-13	165	1010	Varilla 4/32	115	11.5
1000	AAAA800101	5000	2001-12-13	165	1020	Varilla 3/17	130	13
1000	AAAA800101	5000	2001-12-13	165	1030	Varilla 4/33	145	14.5
1000	AAAA800101	5000	2001-12-13	165	1040	Varilla 3/18	160	16
1000	AAAA800101	5000	2001-12-13	165	1050	Varilla 4/34	175	17.5
1000	AAAA800101	5000	2001-12-13	165	1060	Varilla 3/19	190	19
1000	AAAA800101	5000	2001-12-13	165	1070	Varilla 4/35	205	20.5
1000	AAAA800101	5000	2001-12-13	165	1080	Ladrillos rojos	50	5
1000	AAAA800101	5000	2001-12-13	165	1090	Ladrillos grises	35	3.5

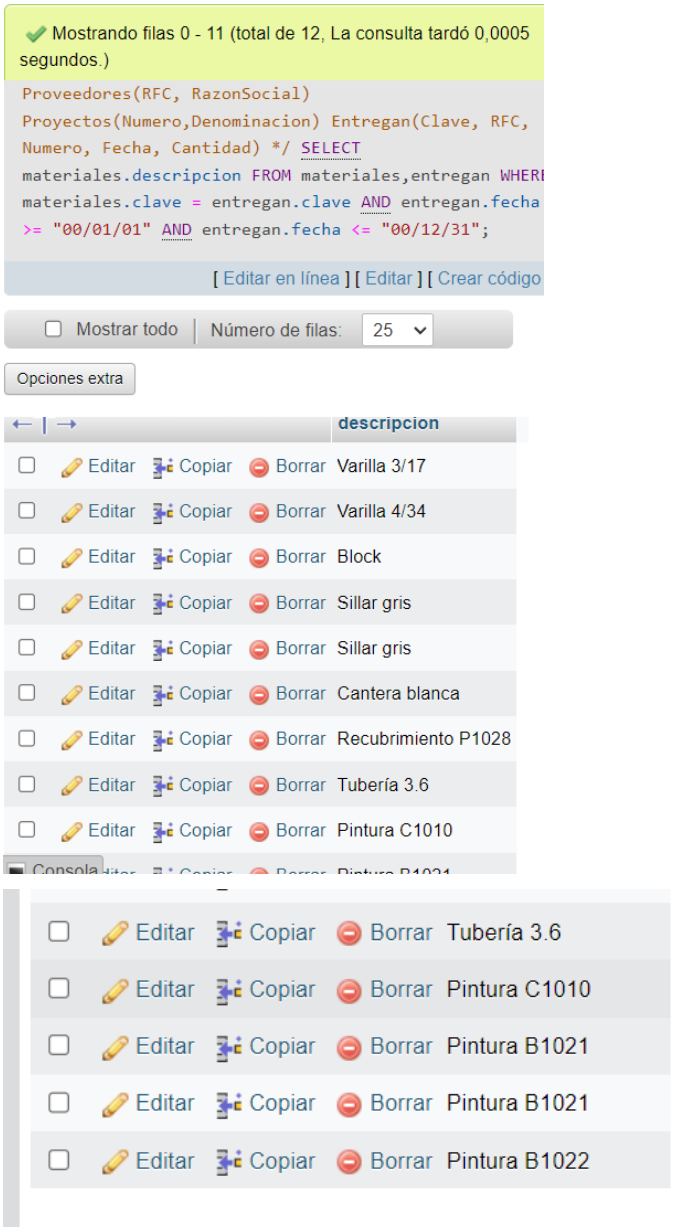
Construcción de consultas a partir de una especificación

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.

¿Por qué aparecen varias veces algunas descripciones de material?
Porque se entregaron en diferentes fechas



Uso del calificador distinct

En el resultado anterior, observamos que una misma descripción de material aparece varias veces.

Agrega la palabra distinct inmediatamente después de la palabra select a la consulta que planteaste antes.

¿Qué resultado obtienes en esta ocasión?

Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0005 segundos.)

```
Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC,
Numero, Fecha, Cantidad) */ SELECT DISTINCT
materiales.descripcion FROM materiales,entregan WHERE
materiales.clave = entregan.clave AND entregan.fecha
>= "00/01/01" AND entregan.fecha <= "00/12/31";
```

[Editar en línea] [Editar] [Crear código P

Mostrar todo | Número de filas: 25

Opciones extra

← | →			descripcion
Editar	Copiar	Borrar	Varilla 3/17
Editar	Copiar	Borrar	Varilla 4/34
Editar	Copiar	Borrar	Block
Editar	Copiar	Borrar	Sillar gris
Editar	Copiar	Borrar	Cantera blanca
Editar	Copiar	Borrar	Recubrimiento P1028
Editar	Copiar	Borrar	Tubería 3.6
Editar	Copiar	Borrar	Pintura C1010
Editar	Copiar	Borrar	Pintura B1021
Editar	Copiar	Borrar	Pintura B1022

Consola

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.

Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0009 segundos.) [fecha: **2006-11-21... - 2003-07-13...**]

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELECT
proyectos.numero, proyectos.denominacion, entregan.cantidad, entregan.fecha FROM
entregan, proyectos WHERE proyectos.numero = entregan.numero ORDER BY entregan.fecha
DESC;
```

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

1 > >> | Mostrar todo | Número de filas: 25

Opciones extra

numero	denominacion	cantidad	fecha 1
5007	Construcción de plaza Magnolias	2	2006-11-21
5018	Tu cambio por la educación	53	2006-07-12
5017	Reparación de la carretera del sol	93	2006-06-10
5018	Tu cambio por la educación	302	2006-04-15
5015	CIT Yucatan	458	2006-02-16
5012	Restauración de instalaciones del CEA	645	2005-12-05
5016	Ampliación de la carretera a la huasteca	356	2005-11-07
5002	CIT Campeche	24	2005-07-03
5016	Ampliación de la carretera a la huasteca	295	2005-06-07
5007	Construcción de plaza Magnolias	506	2005-06-06
5008	Televisa en acción	337	2005-06-03
[illegible]	Mexico sin ti no estamos completos	576	2005-04-30

Consola

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?

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.)

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad)
FROM materiales where Descripcion LIKE 'Si%';
```

[Editar en línea] [Editar] [C

Mostrar todo | Número de filas: 25

Opciones extra

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

Explica que hace el símbolo '%'.
Que empieze con los caracteres antes de % y lo que sigue despues del mismo no importa
¿Qué sucede si la consulta fuera : LIKE 'Si' ?
Buscar las coincidencias exactas con 'Si'
¿Qué resultado obtienes?

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

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELECT *
FROM materiales where Descripcion LIKE 'Si';
```

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

clave	descripcion	precio	impuesto

Operaciones sobre los resultados de la consulta

Crear vista

Explica a qué se debe este comportamiento.
No existe una descripcion que se llame exactamente '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?
Declarar una funcion o variable
¿Cuál es la función de @foo?
Establecer que se esta creando una funcion
¿Que realiza el operador SET?
Agrega un valor a la 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]%';

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

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELECT
RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

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

RFC

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

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

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial) Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ /*SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';*/ SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
```

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

RFC

SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0003 segundos.)

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial) Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ /*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';
```

[Editar en línea] [Editar] [Crear código PH

Mostrar todo | Número de filas: 25

Opciones extra

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

Consola

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;

Mostrando filas 0 - 24 (total de 43, La consulta tardó 0,0004 segundos.)

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELECT
Clave,RFC,Numero,Fecha,Cantidad FROM Entregan WHERE Numero Between 5000 and 5010;
```

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

1 > >> | Mostrar todo | Número de filas: 25

Opciones extra

←T→	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

¿Cómo filtrarías rangos de fechas?

Mostrando filas 0 - 16 (total de 17, La consulta tardó 0,0004 segundos.)

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELI
Clave,RFC,Numero,Fecha,Cantidad FROM Entregan WHERE Fecha Between "2001/01/01" and
"2003/01/01";
```

[Editar en línea] [Editar] [Crear códi

Mostrar todo | Número de filas: 25

Opciones extra

←T→				Clave	RFC	Numero	Fecha	Cantidad
	Editar	Copiar	Borrar	1000	AAAA800101	5000	2001-12-13	165
	Editar	Copiar	Borrar	1060	GGGG800101	5013	2002-07-23	692
	Editar	Copiar	Borrar	1100	CCCC800101	5010	2001-11-19	699
	Editar	Copiar	Borrar	1110	DDDD800101	5011	2002-01-13	368
	Editar	Copiar	Borrar	1150	HHHH800101	5004	2002-11-14	453
	Editar	Copiar	Borrar	1170	BBBB800101	5017	2002-01-31	180
	Editar	Copiar	Borrar	1180	CCCC800101	5017	2002-04-17	334
	Editar	Copiar	Borrar	1180	CCCC800101	5018	2001-10-24	407
	Editar	Copiar	Borrar	1210	FFFF800101	5014	2002-09-26	70
	Editar	Copiar	Borrar	1260	CCCC800101	5006	2001-04-09	460
	Editar	Copiar	Borrar	1260	CCCC800101	5009	2001-07-28	631

Consola

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?
De los proveedores que tienen una razon social que inicia con La mostrar RFC, Cantidad, Fecha y Numero de Entregan donde el numero debe estar en un rango de 5000 a 5010

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0007 segundos.)

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

[Editar en línea] [Editar] [Crear código P

Mostrar todo | Número de filas: 25

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

Consola

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

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.

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.
Muestra los primeros 2 elementos

SELECT TOP 2 * FROM Proyectos

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.)

```
/* Materiales(Clave, Descripción, Costo) Proveedores(RFC, RazonSocial)
Proyectos(Numero,Denominacion) Entregan(Clave, RFC, Numero, Fecha, Cantidad) */ SELECT*
FROM Proyectos LIMIT 2;
```

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

Opciones extra

←T→			numero	denominacion
☐ Editar	Copiar	Borrar	5000	Vamos Mexico
☐ Editar	Copiar	Borrar	5001	Aztecon

¿Qué sucede con la siguiente consulta? Explica por qué.
Nuemro: variable a reemplazar

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.

Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0003 segundos.)

```
SELECT * FROM `materiales`
```

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	descripcion	precio	impuesto	PorcentajeImpuesto
Editar Copiar Borrar		1000	Varilla 3/16	100	10	2.00
Editar Copiar Borrar		1010	Varilla 4/32	115	11.5	2.02
Editar Copiar Borrar		1020	Varilla 3/17	130	13	2.04
Editar Copiar Borrar		1030	Varilla 4/33	145	14.5	2.06
Editar Copiar Borrar		1040	Varilla 3/18	160	16	2.08
Editar Copiar Borrar		1050	Varilla 4/34	175	17.5	2.10
Editar Copiar Borrar		1060	Varilla 3/19	190	19	2.12
Editar Copiar Borrar		1070	Varilla 4/35	205	20.5	2.14
Editar Copiar Borrar		1080	Ladrillos rojos	50	5	2.16

¿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?

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.

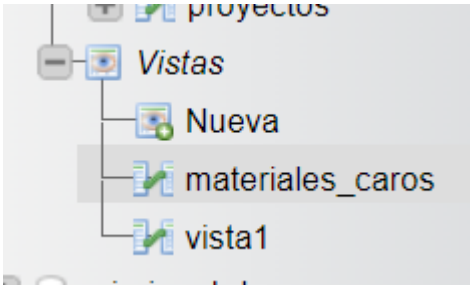
Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Solo usando vistas).

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.

NOTAS

```
CREATE VIEW materiales_caros as
SELECT *
FROM materiales
WHERE precio > 1000
```



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.