

Consulta de una tabla completa

Algebra relacional: materiales

SQL: select * from materiales

```
1 SELECT *
2 FROM materiales
```

✓ Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0009 segundos.)

```
SELECT * FROM materiales;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

1 ▾

>

>>

☐

Mostrar todo

Número de filas:

25 ▾

Filtrar filas:

+ Opciones

				clave	descripcion	precio	impuesto
<input type="checkbox"/>		Editar		Copiar		Borrar	1000 Varilla 3/16 100 10
<input type="checkbox"/>		Editar		Copiar		Borrar	1010 Varilla 4/32 115 11.5
<input type="checkbox"/>		Editar		Copiar		Borrar	1020 Varilla 3/17 130 13
<input type="checkbox"/>		Editar		Copiar		Borrar	1030 Varilla 4/33 145 14.5
<input type="checkbox"/>		Editar		Copiar		Borrar	1040 Varilla 3/18 160 16
<input type="checkbox"/>		Editar		Copiar		Borrar	1050 Varilla 4/34 175 17.5

Selección

Algebra relacional: $SL_{\{clave=1000\}}(\text{materiales})$

SQL:

select * from materiales

where clave=1000

```
1 SELECT *
2 FROM materiales
3 WHERE clave=1000
```

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0015 segundos.)

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

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Ac](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

	clave	descripcion	precio	impuesto
<input type="checkbox"/>  Editar  Copiar  Borrar	1000	Varilla 3/16	100	10

Proyección

Algebra relacional: $\text{PR}\{\text{clave}, \text{rfc}, \text{fecha}\}$ (entregan)

SQL: select clave,rfc,fecha from entregan

```
1 SELECT clave,rfc,fecha
2 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 PHP](#)]

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

+ Opciones

	clave	rfc	fecha
<input type="checkbox"/>  Editar  Copiar  Borrar	1000	AAAA800101	2001-12-13
<input type="checkbox"/>  Editar  Copiar  Borrar	1000	AAAA800101	1999-07-13
<input type="checkbox"/>  Editar  Copiar  Borrar	1010	BBBB800101	1998-07-28

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?

```

1 SELECT *
2 FROM materiales,entregan
3 WHERE materiales.clave = entregan.clave

```

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

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: Filtrar filas:

+ Opciones

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

Reunión con criterio específico

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

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

```

1 SELECT *
2 FROM entregan,proyectos
3 WHERE entregan.numero <= proyectos.numero

```

✓ Mostrando filas 0 - 835 (total de 836, La consulta tardó 0,0021 segundos.)

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

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

Número de filas: Filtrar filas:

+ Opciones

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

Unión (se ilustra junto con selección)

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

SQL: (select * from entregan where clave=1450) unión (select * from entregan where clave=1300)

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

```
1 (SELECT *
2 FROM entregan
3 WHERE clave=1450)
4 UNION
5 (SELECT *
6 FROM entregan
7 WHERE clave=1300)
```

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0013 segundos.)

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

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

☐ Mostrar todo | Número de filas: 25 ▾ Filtrar filas: Sort by key:

+ Opciones

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

Intersección (se ilustra junto con selección y proyección)

Algebra relacional: $PR\{clave\}(SL\{numero=5001\}(entregan)) \cap$

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

SQL: (select clave from entregan where numero=5001) intersect (select clave from entregan where numero=5018)

```
1 SELECT clave
2 FROM entregan
3 WHERE numero=5001 IN (SELECT clave
4                       FROM entregan
5                       WHERE numero=5018)
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas)

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

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Compartir](#)]

clave

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.

```
1 (SELECT *
2 FROM entregan)
3 EXCEPT
4 (SELECT *
5 FROM entregan
6 WHERE clave=1000);
```

✓ Mostrando filas 0 - 24 (total de 85, La consulta tardó 0,0014 segundos.)

```
(SELECT * FROM entregan) EXCEPT (SELECT * FROM entregan WHERE clave=1000);
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

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

+ Opciones

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

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?

```
1 SELECT *
2 FROM entregan,materiales
```

✓ Mostrando filas 0 - 3914 (total de 3915, La consulta tardó 0,0040 segundos.)

```
SELECT * FROM entregan,materiales;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

Número de filas: Filtrar filas:

+ Opciones

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

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?

```
1 SELECT M.descripcion
2 FROM materiales M, entregan E
3 WHERE M.clave = E.clave AND (fecha BETWEEN '01/01/00' AND '31/12/00')
```

✓ Mostrando filas 0 - 24 (total de 47, La consulta tardó 0,0018 segundos.)

```
SELECT M.descripcion FROM materiales M, entregan E WHERE M.clave = E.clave AND (fecha BETWEEN '01/01/00' AND '31/12/00');
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

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

+ Opciones

	descripcion
<input type="checkbox"/> Editar Copiar Borrar	Varilla 3/16
<input type="checkbox"/> Editar Copiar Borrar	Varilla 3/17
<input type="checkbox"/> Editar Copiar Borrar	Varilla 4/33
<input type="checkbox"/> Editar Copiar Borrar	Varilla 4/34

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?

```
1 SELECT DISTINCT M.descripcion
2 FROM materiales M, entregan E
3 WHERE M.clave = E.clave AND (fecha BETWEEN '01/01/00' AND '31/12/00')
```

✓ Mostrando filas 0 - 24 (total de 31, La consulta tardó 0,0020 segundos.)

```
SELECT DISTINCT M.descripcion FROM materiales M, entregan E WHERE M.clave = E.clave AND (fecha BETWEEN '01/01/00' AND '31/12/00');
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

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

+ Opciones

	descripcion
<input type="checkbox"/> Editar Copiar Borrar	Varilla 3/16
<input type="checkbox"/> Editar Copiar Borrar	Varilla 3/17
<input type="checkbox"/> Editar Copiar Borrar	Varilla 4/33
<input type="checkbox"/> Editar Copiar Borrar	Varilla 4/34
<input type="checkbox"/> Editar Copiar Borrar	Varilla 3/19
<input type="checkbox"/> Editar Copiar Borrar	Varilla 4/35
<input type="checkbox"/> Editar Copiar Borrar	Ladrillos rojos

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.

```
1 SELECT P.numero, P.denominacion, E.fecha, E.cantidad
2 FROM proyectos P, entregan E
3 ORDER BY E.fecha ASC;
```

✓ Mostrando filas 0 - 24 (total de 1740, La consulta tardó 0,0024 segundos.) [fecha: 0000-00-00... - 1997-01-06...]

SELECT P.numero, P.denominacion, E.fecha, E.cantidad FROM proyectos P, entregan E ORDER BY E.fecha ASC;

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

1 > >>

Número de filas: 25

Filtrar filas:

+ Opciones

numero	denominacion	fecha	1	cantidad
5012	Restauración de instalaciones del CEA	0000-00-00		152
5003	Mexico sin ti no estamos completos	0000-00-00		152
5017	Reparación de la carretera del sol	0000-00-00		152
5008	Televisa en acción	0000-00-00		152
5013	Reparación de la plaza Sonora	0000-00-00		152

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%'
```

```
1 SELECT *
2 FROM materiales
3 WHERE descripcion LIKE 'Si%'
```

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0010 segundos.)

SELECT * FROM materiales WHERE descripcion LIKE 'Si%';

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Acción](#)]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas:

+ Opciones

← T →


▼

clave

descripcion

precio

impuesto

<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1120	Sillar rosa	100	10
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1130	Sillar gris	110	11

¿Qué resultado obtienes?

Explica que hace el símbolo '%'.

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

¿Qué resultado obtienes?

Explica a qué se debe este comportamiento.

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?

¿Cuál es la función de @foo?

¿Que realiza el operador SET?

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

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

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

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

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

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

```

1 SELECT RFC
2 FROM Entregan
3 WHERE RFC
4 LIKE '[A-D]%'

```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas).

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

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [

RFC

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

```

1 SELECT RFC
2 FROM Entregan
3 WHERE RFC
4 LIKE '[^A]%' ;

```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas)

```

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

```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [

SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

```

1 SELECT Numero
2 FROM Entregan
3 WHERE Numero
4 LIKE '___6';

```

✓ Mostrando filas 0 - 8 (total de 9, La consulta tardó 0,0011 segundos.)

```




























SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código P](#)

☐ Mostrar todo | Número de filas: 25 Filtrar filas: Bu

+ Opciones

				Numero
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5016
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5006
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5006
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5016
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5016
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5006
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5006
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5016
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5016

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;

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










✓ Mostrando filas 0 - 24 (total de 43, La consulta tardó 0,0015 segundos.)

`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:

+ Opciones

			Clave	RFC	Numero	Fecha	Cantidad
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1000	AAAA800101	5000	2001-12-13 165
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1200	EEEE800101	5000	2003-03-15 177
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1400	AAAA800101	5000	1999-04-07 382

¿Cómo filtrarías rangos de fechas?

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])

```
1 SELECT e.rfc,e.cantidad, e.fecha,e.numero
2 FROM entregan E
3 WHERE E.numero Between 5000 and 5010 AND Exists (SELECT p.rfc
4 FROM proveedores P
5 WHERE p.razonsocial LIKE 'La%' and E.rfc = p.rfc )
```

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0030 segundos.)

SELECT e.rfc,e.cantidad, e.fecha,e.numero FROM entregan E WHERE E.numero Between 5000 and 5010 AND Exists (SELECT p.rfc FROM proveedores P WHERE p.razonsocial LIKE 'La%' and E.rfc = p.rfc);

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

rfc	cantidad	fecha	numero
AAAA800101	165	2001-12-13	5000
AAAA800101	86	2005-04-03	5008
AAAA800101	152	0000-00-00	5004
AAAA800101	382	1999-04-07	5000
AAAA800101	116	2005-04-21	5010

¿Qué hace la consulta?

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

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

```

1 SELECT e.rfc,e.cantidad, e.fecha,e.numero
2 FROM entregan E
3 WHERE E.numero Between 5000 and 5010 and e.rfc IN (SELECT p.rfc
4
5                                FROM proveedores P
                                WHERE p.razonsocial LIKE 'La%' and E.rfc = p.rfc )

```

✓ Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0032 segundos.)

SELECT e.rfc,e.cantidad, e.fecha,e.numero FROM entregan E WHERE E.numero Between 5000 and 5010 and e.rfc IN (SELECT p.rfc FROM proveedores P WHERE p.razonsocial LIKE 'La%' and E.rfc = p.rfc);

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

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

rfc	cantidad	fecha	numero
AAAA800101	165	2001-12-13	5000
AAAA800101	86	2005-04-03	5008
AAAA800101	152	0000-00-00	5004

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.

```

1 SELECT e.rfc,e.cantidad, e.fecha,e.numero
2 FROM entregan E
3 WHERE E.numero Between 5000 and 5010 and e.rfc NOT IN (SELECT p.rfc
4
5                                FROM proveedores P
                                WHERE p.razonsocial NOT LIKE 'La%' and E.rfc = p.rfc )

```

✓ Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0036 segundos.)

SELECT e.rfc,e.cantidad, e.fecha,e.numero FROM entregan E WHERE E.numero Between 5000 and 5010 and e.rfc NOT IN (SELECT p.rfc FROM proveedores P WHERE p.razonsocial NOT LIKE 'La%' and E.rfc = p.rfc);

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

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

rfc	cantidad	fecha	numero
AAAA800101	165	2001-12-13	5000
AAAA800101	382	1999-04-07	5000
CCCC800101	582	2003-12-16	5002
CCCC800101	603	2001-09-09	5002
AAAA800101	152	0000-00-00	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

```

1 SELECT *
2 FROM Proyectos
3 LIMIT 2

```

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

```
SELECT * FROM Proyectos LIMIT 2;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código P](#)

+ Opciones

			numero	denominacion
<input type="checkbox"/>	Editar	Copiar	Borrar	5000 Vamos Mexico
<input type="checkbox"/>	Editar	Copiar	Borrar	5001 Aztecon

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

SELECT TOP Numero FROM Proyectos

```
1 SELECT Numero
2 FROM Proyectos
3 LIMIT 1
```

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

```
SELECT Numero FROM Proyectos LIMIT 1;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código P](#)

+ Opciones

				Numero
<input type="checkbox"/>	Editar	Copiar	Borrar	5000

Modificando la estructura de una tabla existente.

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

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

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

<div><div><div></div><div></div><div></div></div></div>				clave	descripcion	precio	impuesto	PorcentajeImpuesto
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1000	Varilla 3/16	100	10	NULL
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1010	Varilla 4/32	115	11.5	NULL
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1020	Varilla 3/17	130	13	NULL
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1030	Varilla 4/33	145	14.5	NULL
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1040	Varilla 3/18	160	16	NULL

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.

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

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

+ Opciones

<div><div><div>←</div><div>T</div><div>→</div></div></div>						clave	descripcion	precio	impuesto	PorcentajeImpuesto	
<input type="checkbox"/>		Editar		Copiar		Borrar	1000	Varilla 3/16	100	10	2.00
<input type="checkbox"/>		Editar		Copiar		Borrar	1010	Varilla 4/32	115	11.5	2.02
<input type="checkbox"/>		Editar		Copiar		Borrar	1020	Varilla 3/17	130	13	2.04
<input type="checkbox"/>		Editar		Copiar		Borrar	1030	Varilla 4/33	145	14.5	2.06
<input type="checkbox"/>		Editar		Copiar		Borrar	1040	Varilla 3/18	160	16	2.08
<input type="checkbox"/>		Editar		Copiar		Borrar	1050	Varilla 4/34	175	17.5	2.10
<input type="checkbox"/>		Editar		Copiar		Borrar	1060	Varilla 3/19	190	19	2.12

¿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".

```
1 SELECT M.clave, M.descripcion
2 FROM materiales M, Proyectos P, entregan E
3 WHERE M.clave = E.clave AND E.numero = P.numero AND P.denominacion = 'México sin ti no estamos completos'
```

						clave	descripcion
<input type="checkbox"/>		Editar		Copiar		Borrar	1030 Varilla 4/33
<input type="checkbox"/>		Editar		Copiar		Borrar	1230 Cemento
<input type="checkbox"/>		Editar		Copiar		Borrar	1430 Pintura B1022

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

```
1 SELECT M.clave, M.descripcion
2 FROM materiales M, proveedores P, entregan E
3 WHERE M.clave = E.clave AND E.rfc = P.rfc AND P.razonsocial = 'Acme tools'
```

✔ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0029 segundos.)

```
SELECT M.clave, M.descripcion FROM materiales M, proveedores P, entregan E WHERE M.clave = E.clave
```

☐ 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.

```
1 SELECT P.rfc, AVG(E.cantidad) as 'Promedio 2000'
2 FROM proveedores P, entregan E
3 WHERE E.fecha BETWEEN '01/01/00' AND '31/12/00'
4 HAVING AVG(E.cantidad) > 300
```

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0019 segundos.)

SELECT P.rfc, AVG(E.cantidad) as 'Promedio 2000' FROM prove

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

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Bu

+ Opciones

	rfc	Promedio 2000
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	AAAA800101	360.5745

El Total entregado por cada material en el año 2000.

```
1 SELECT M.descripcion, SUM(E.cantidad) as 'Total 2000'
2 FROM materiales M, entregan E
3 WHERE E.fecha BETWEEN '01/01/00' AND '31/12/00' AND E.clave = M.clave
4 GROUP BY M.descripcion
```

	descripcion	Total 2000
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Block	699
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Cantera amarilla	233
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Cantera gris	911
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Cantera rosa	244
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Cemento	312

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







```
1 SELECT M.clave
2 FROM materiales M, entregan E
3 WHERE E.fecha BETWEEN '01/01/01' AND '31/12/01' AND E.clave = M.clave
4 ORDER BY E.cantidad DESC
5 LIMIT 1
```

+ Opciones

	clave
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1080

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

```
1 SELECT m.descripcion
2 FROM materiales m
3 WHERE m.descripcion LIKE '%ub%'
```

	descripcion
<input type="checkbox"/>   	Recubrimiento P1001
<input type="checkbox"/>   	Recubrimiento P1010
<input type="checkbox"/>   	Recubrimiento P1019
<input type="checkbox"/>   	Recubrimiento P1028
<input type="checkbox"/>   	Recubrimiento P1037

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

```
1 SELECT pr.denominacion, SUM(m.PorcentajeImpuesto*(e.cantidad*m.precio)) as Total
2 FROM materiales m, proyectos pr, entregan e
3 WHERE m.clave = e.clave AND pr.numero = e.numero
4 GROUP BY pr.denominacion
```

	denominacion	Total
<input type="checkbox"/>   	Ampliación de la carretera a la huasteca	1383550.9
<input type="checkbox"/>   	Aztecon	360521.9
<input type="checkbox"/>   	CIT Campeche	384808.19999999995
<input type="checkbox"/>   	CIT Yucatan	1982230
<input type="checkbox"/>   	Construcción de Hospital Infantil	356580.30000000005

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).

```
1 SELECT pr.denominacion, p.rfc, p.razonsocial
2 FROM proyectos pr, entregan e, proveedores p
3 WHERE pr.denominacion = 'Televisa en acción' AND pr.denominacion NOT IN (SELECT pr.denominacion
4                                     FROM proyectos pr, entregan e, proveedores p
5                                     WHERE p.rfc = e.rfc AND pr.numero = e.numero AND pr.denominacion = 'Educando en Coahuila')
6 GROUP BY p.razonsocial
```

denominacion	rfc	razonsocial
Televisa en acción	EEEE800101	Alvin
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	FFFF800101	Comex
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.

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