



**Instituto Tecnológico de Estudios Superiores de Monterrey,
Campus Querétaro**

Desarrollo de Software y Toma de Decisiones

Laboratorio 20

Carlos Vazquez - A01711730

Consulta de un tabla completa

Algebra relacional.
materiales

SQL
select * from materiales

Selección

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

SQL
select * from materiales
where clave=1000

Proyección

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

SQL
select clave,rfc,fecha from entregan

Reunión Natural

Algebra relacional.
entregan JN materiales

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

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

No, solo aparecen los materiales que si han sido entregados

```
1 SELECT * FROM materiales m, entregan e
2 WHERE m.clave = e.clave;
3
```

100% 1:3

Result Grid Filter Rows: Search Export:

	clave	descripci...	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	

Reunión con criterio específico

Algebra relacional.

entregan \Join {entregan.numero <= proyectos.numero} proyectos

SQL

```
select * from entregan,proyectos
```

```
where entregan.numero < = proyectos.numero
```

Unión (se ilustra junto con selección)

Algebra relacional.

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

SQL

```
(select * from entregan where clave=1450)
```

```
union
```

```
(select * from entregan where clave=1300)
```

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

```
3
4  (SELECT * FROM entregan
5   WHERE clave = 1450)
6
7   UNION
8
9   (SELECT * FROM entregan
10  WHERE clave = 1300)
```

100% 25:2

Result Grid Filter Rows: Search Export:

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

```
12 SELECT * FROM entregan
13 WHERE clave=1450 or clave=1300
```

100% 24:12

Result Grid Filter Rows: Search Edit:

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

estas 2 son la misma consulta pero una sin usar unión

Intersección

Algebra relacional.

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

SQL

Nota: Debido a que en SQL server no tiene definida alguna palabra reservada que nos permita hacer esto de una manera entendible, veremos esta sección en el siguiente laboratorio con el uso de Subconsultas. Un ejemplo de un DBMS que si tiene la implementación de una palabra reservada para esta función es Oracle, en él si se podría generar la consulta con una sintaxis como la siguiente:

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

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

```

15 SELECT * FROM entregan
16 WHERE clave
17 NOT IN
18 (SELECT clave FROM entregan
19 WHERE clave = 1000)

```

100% 20:19

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

	clave	rfc	numero	fecha	cantidad	
	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	

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?

Selección de entregas:

```

21 SELECT * FROM entregan;

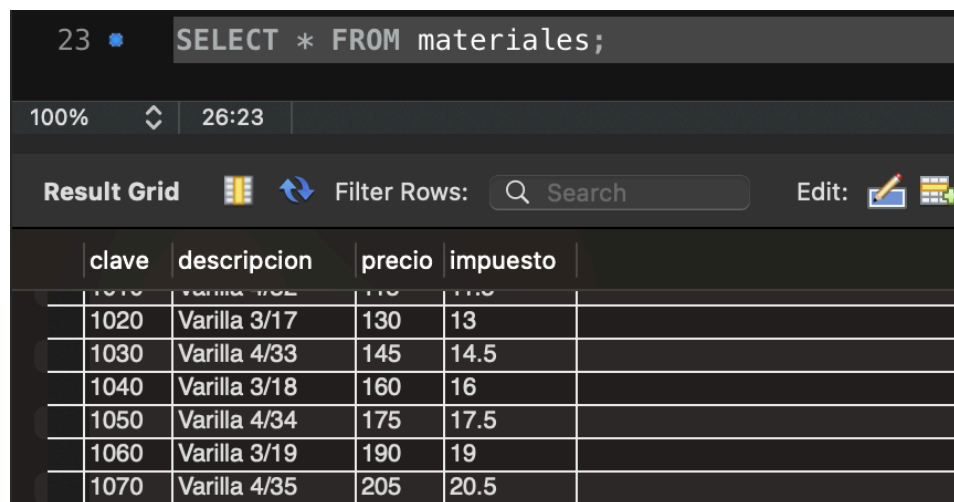
```

100% 24:21

Result Grid Filter Rows: Search Edit:

	clave	rfc	numero	fecha	cantidad	
	1000	AAAA800101	5000	2001-12-13	165	
	1000	AAAA800101	5019	1999-07-13	254	
	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	

Selección de materiales:



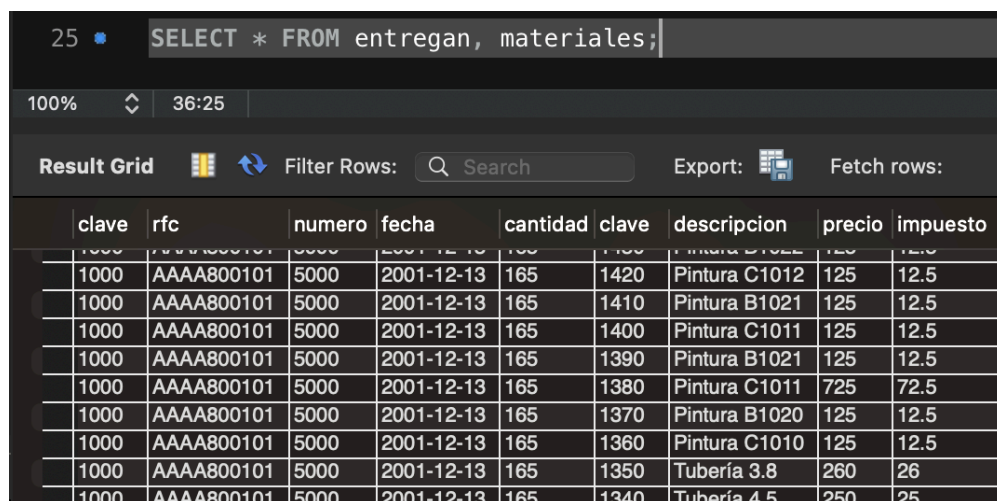
23 • `SELECT * FROM materiales;`

100% 26:23

Result Grid Filter Rows: Search Edit:

	clave	descripcion	precio	impuesto
	1020	Varilla 3/17	130	13
	1030	Varilla 4/33	145	14.5
	1040	Varilla 3/18	160	16
	1050	Varilla 4/34	175	17.5
	1060	Varilla 3/19	190	19
	1070	Varilla 4/35	205	20.5

Selección de ambas:



25 • `SELECT * FROM entregan, materiales;`

100% 36:25

Result Grid Filter Rows: Search Export: Fetch rows:

	clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
	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

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 algunas veces algunas descripciones de material?

Lo que mantiene la unicidad no son las descripciones, sino las claves. Al seleccionar las descripciones, se muestran múltiples que podrían ser iguales

```
27 SELECT DATE_FORMAT(NOW(), '%Y-%m%-d')
28 AS formatted_date;
29
30 SELECT m.descripcion
31 FROM materiales m, entregan e
32 WHERE m.clave = e.clave
33 AND YEAR(e.fecha) = 2000
```

100% 25:33

Result Grid Filter Rows: Search Export:

	descripcion	
	Varilla 3/17	
	Varilla 4/34	
	Block	
	Sillar gris	
	Sillar gris	
	Cantera blanca	
	Recubrimiento P1028	

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?

Los registros no se repiten:

```
35 SELECT DISTINCT (m.descripcion)
36 FROM materiales m, entregan e
37 WHERE m.clave = e.clave AND YEAR (e.fecha) = 2000;
```

100% 51:37

Result Grid Filter Rows: Search Export:

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

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.

```
39 SELECT p.numero, denominacion, e.fecha, e.cantidad
40 FROM proyectos p, entregan e
41 WHERE p.numero = e.numero
42 ORDER BY p.numero ASC, e.fecha DESC;
```

100% 37:42

Result Grid Filter Rows: Search Export:

numero	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

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

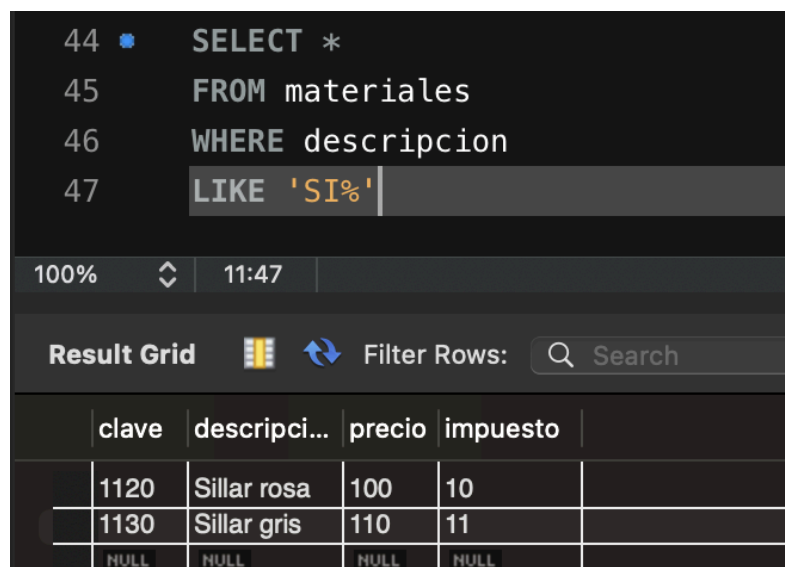
Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre"). Para SQL Server también pueden 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%'
```



The screenshot shows a SQL query editor with a dark theme. The query is as follows:

```
44 SELECT *
45 FROM materiales
46 WHERE descripcion
47 LIKE 'SI%'
```

Below the query editor, there is a toolbar with a zoom level of 100%, a refresh icon, and a timestamp of 11:47. Below the toolbar is a 'Result Grid' section with a search bar and a 'Filter Rows' button. The result grid displays the following data:

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

¿Qué resultado obtienes?

Explica que hace el símbolo '%'. Es forma de representar una secuencia

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

¿Qué resultado obtienes? Busca exactamente con el texto '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? Para declarar variables en bloque de código

¿Cuál es la función de @foo? Es el nombre de una variable

¿Que realiza el operador SET? asignar un valor

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]%';
```

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

52




100% 49:51

Result Grid   Filter Rows: Export: 

RFC

```
SELECT Numero FROM Entregan WHERE Numero LIKE '__6';
```

```
53 SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
54
100% 54:53
```

Result Grid   Filter Rows: Export: 

	Numero	
	5006	
	5006	
	5006	
	5006	
	5016	
	5016	
	5016	
	5016	
	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;
```

¿Cómo filtrarías rangos de fechas?

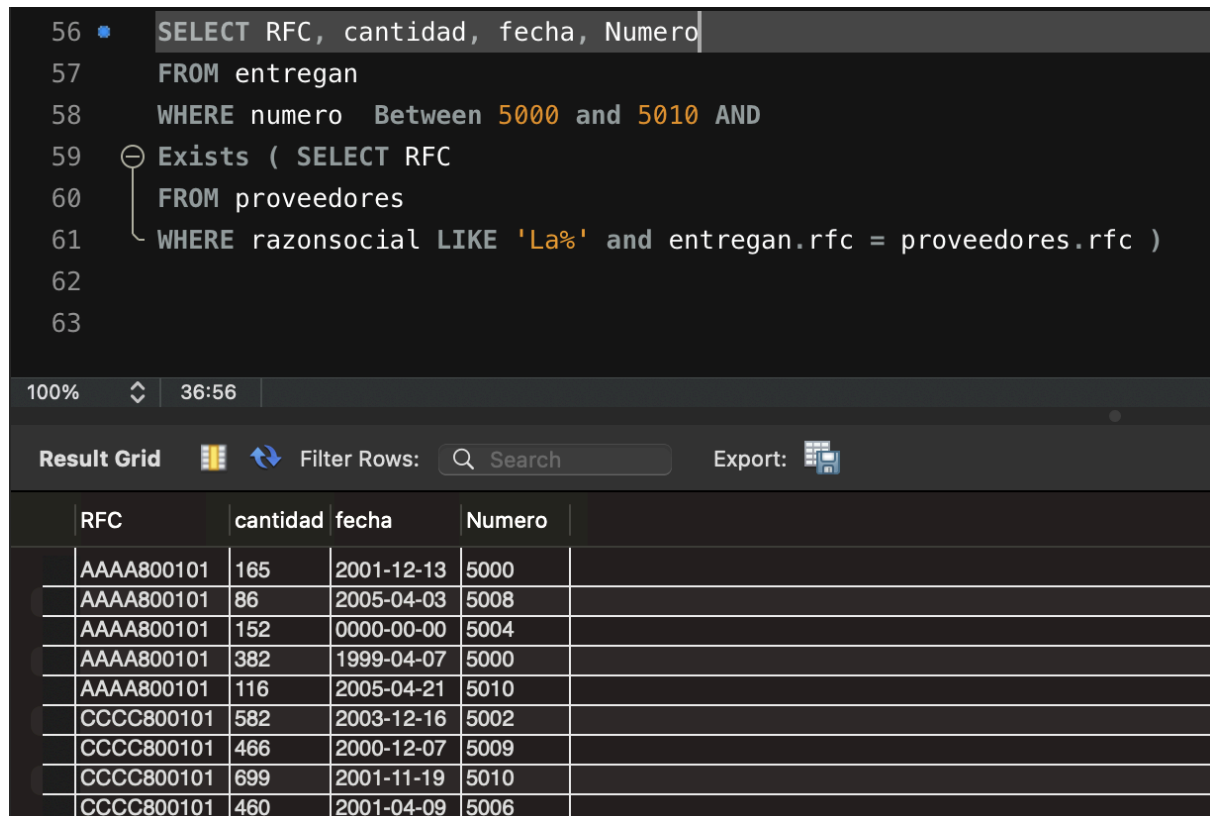
EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.

```
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] )
```

¿Qué hace la consulta?

regresa los datos donde el valor va de 5000 a 5010 y existe un registro en la tabla proveedores con la misma RFC y la razón social empieza con “la”



```
56 SELECT RFC, cantidad, fecha, Numero
57 FROM entregan
58 WHERE numero Between 5000 and 5010 AND
59 EXISTS ( SELECT RFC
60 FROM proveedores
61 WHERE razonsocial LIKE 'La%' and entregan.rfc = proveedores.rfc )
62
63
```

100% 36:56

Result Grid Filter Rows: Search Export:

	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
	CCCC800101	582	2003-12-16	5002
	CCCC800101	466	2000-12-07	5009
	CCCC800101	699	2001-11-19	5010
	CCCC800101	460	2001-04-09	5006

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

```
63 SELECT RFC, cantidad, fecha, numero
64 FROM entregan
65 WHERE numero between 5000 and 5010
66 AND RFC IN (
67     SELECT RFC
68     FROM proveedores
69     WHERE razonsocial like 'La%')
```

100% 14:64

Result Grid Filter Rows: Search Export:

	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	
	CCCC800101	582	2003-12-16	5002	
	CCCC800101	466	2000-12-07	5009	
	CCCC800101	699	2001-11-19	5010	

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.

```
71 SELECT RFC, cantidad, fecha, numero
72 FROM entregan
73 WHERE numero between 5000 and 5010
74 AND RFC NOT IN (
75     SELECT RFC
76     FROM proveedores
77     WHERE razonsocial like 'La%');
```

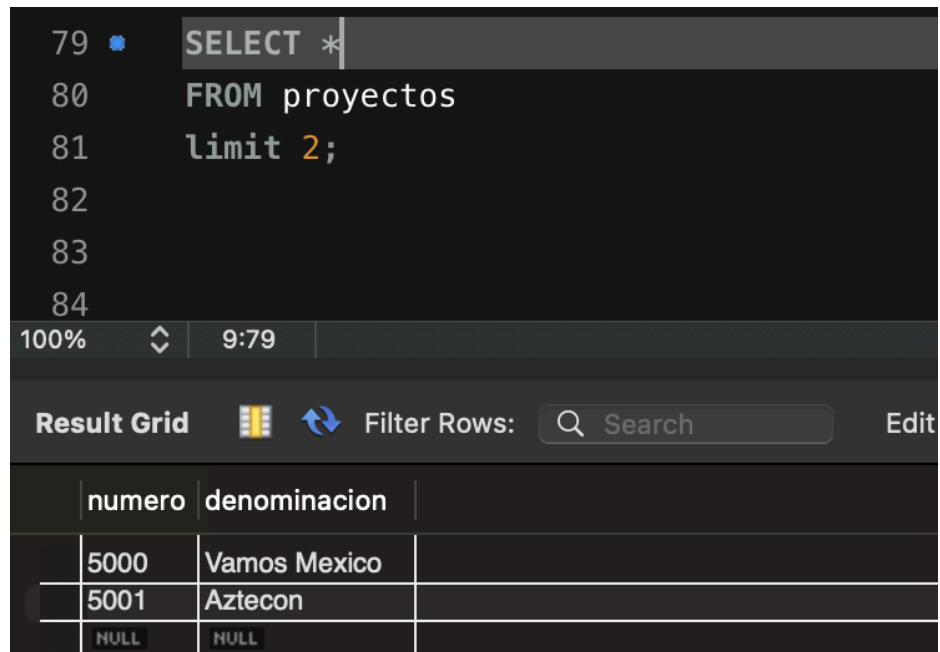
100% 31:77

Result Grid Filter Rows: Search Export:

	RFC	cantidad	fecha	numero	
	EEEE800101	263	1999-12-18	5004	
	FFFF800101	503	2004-02-07	5005	
	GGGG800101	324	2004-09-29	5006	
	HHHH800101	2	2006-11-21	5007	
	BBBB800101	73	1997-03-13	5009	
	BBBB800101	421	1998-11-17	5010	
	DDDD800101	337	2005-06-03	5008	
	EEEE800101	692	2003-11-21	5007	

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.

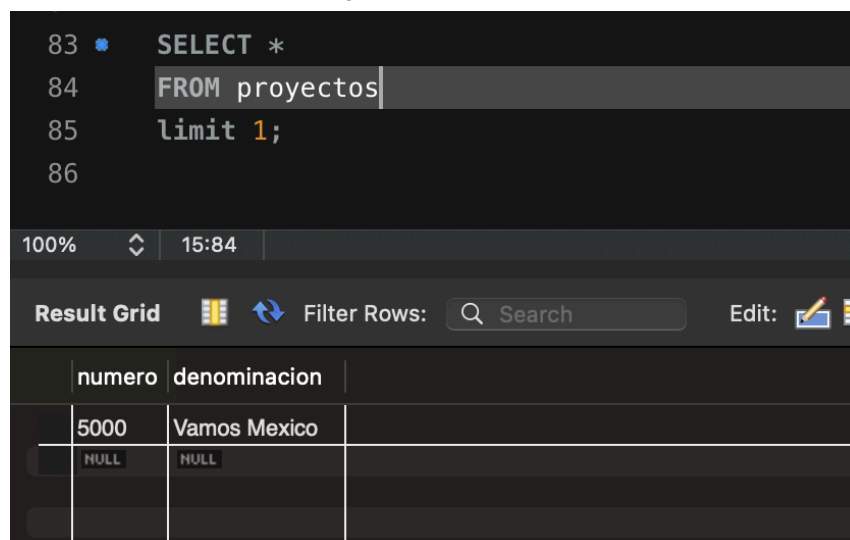
¿Qué hace la siguiente sentencia? Explica por qué. selección de las primeras 2 filas de tabla proyectos
SELECT TOP 2 * FROM Proyectos



The screenshot shows a SQL query editor with a dark theme. The query is: `SELECT * FROM proyectos limit 2;`. Below the query, there is a toolbar with 'Result Grid', 'Filter Rows', and a search bar. The results are displayed in a table with columns 'numero' and 'denominacion'.

numero	denominacion
5000	Vamos Mexico
5001	Aztecon
NULL	NULL

¿Qué sucede con la siguiente consulta? Explica por qué. Selecciona un valor de la columna número de la tabla proyectos
SELECT TOP Numero FROM Proyectos



The screenshot shows a SQL query editor with a dark theme. The query is: `SELECT * FROM proyectos limit 1;`. Below the query, there is a toolbar with 'Result Grid', 'Filter Rows', and an 'Edit' button. The results are displayed in a table with columns 'numero' and 'denominacion'.

numero	denominacion
5000	Vamos Mexico
NULL	NULL

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);
```

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.

```
88 • ALTER TABLE materiales ADD porcentajeimpuesto NUMERIC (6,2);
89 • UPDATE materiales
90   SET porcentajeimpuesto = (2 * clave / 1000)
91   WHERE clave IS NOT NULL;
92 • SELECT * FROM materiales;
```

100% 19:89

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

	clave	descripcion	precio	impuesto	porcentajeimpue...
	1000	Varilla 3/16	100	10	2.00
	1010	Varilla 4/32	115	11.5	2.02
	1020	Varilla 3/17	130	13	2.04
	1030	Varilla 4/33	145	14.5	2.06

¿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?

```
94 • SELECT e.clave, e.rfc, e.numero, e.fecha, e.cantidad,
95        m.precio, m.porcentajeimpuesto, e.cantidad * m.precio * (1 + m.porcentajeimpuesto)
96        AS importentrega
97        FROM entregan e, materiales m
98        where e.clave = m.clave
```

100% 24:98

Result Grid Filter Rows: Search Export:

	clave	rfc	numero	fecha	cantidad	precio	porcentajeimpue...	importentrega
	1000	AAAA800101	5000	2001-12-13	165	100	2.00	49500
	1000	AAAA800101	5019	1999-07-13	254	100	2.00	76200
	1010	BBBB800101	5001	1998-07-28	528	115	2.02	183374.4
	1010	BBBB800101	5018	1997-02-09	523	115	2.02	181637.9
	1020	CCCC800101	5002	2003-12-16	582	130	2.04	230006.4

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.

```
124 • SELECT DISTINCT m.clave, m.descripcion
125 FROM materiales m
126 JOIN entregan e ON m.clave = e.clave
127 JOIN proyectos p ON e.numero = p.numero
128 WHERE p.denominacion = 'México sin ti no estamos completos';
129
```

100% 1:123

Result Grid Filter Rows: Search Export:

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

```
130 • SELECT DISTINCT m.clave, m.descripcion
131 FROM materiales m
132 JOIN entregan e ON m.clave = e.clave
133 JOIN proveedores p ON e.rfc = p.rfc
134 WHERE p.razonsocial = 'Acme tools';
135
136
```

100% 39:130

Result Grid Filter Rows: Search Export:

clave	descripci...

```
136 • SELECT e.rfc
137 FROM entregan e
138 WHERE YEAR(e.fecha) = 2000
139 GROUP BY e.rfc
140 HAVING AVG(e.cantidad) >= 300;
141
142
143
```

100% 13:136

Result Grid Filter Rows: Search Export:

rfc
BBBB...
FFFF8...
GGG...

```
142 • SELECT e.clave, SUM(e.cantidad) AS total_entregado
143 FROM entregan e
144 WHERE YEAR(e.fecha) = 2000
145 GROUP BY e.clave;
146
147
148
```

100% 31:140

Result Grid Filter Rows: Search Export:

clave	total_entrega...
1020	8
1050	623
1100	466
1130	625
1140	583

```

147 CREATE VIEW vista_material_2001 AS
148 SELECT e.clave, SUM(e.cantidad) AS total_vendido
149 FROM entregan e
150 WHERE YEAR(e.fecha) = 2001
151 GROUP BY e.clave;
152
153 SELECT clave
154 FROM vista_materiales_2001
155 ORDER BY total_vendido DESC

```

100% 18:151

Result Grid Filter Rows: Search Export: Fetch rows:

clave
1260

```

158 SELECT *
159 FROM materiales
160 WHERE descripcion LIKE '%ub%';
161

```

100% 31:160

Result Grid Filter Rows: Search Edit:

	clave	descripcion	precio	impuesto	porcentajeimpue...
	1180	Recubrimiento P1001	200	20	2.36
	1190	Recubrimiento P1010	220	22	2.38
	1200	Recubrimiento P1019	240	24	2.40
	1210	Recubrimiento P1028	250	25	2.42
	1220	Recubrimiento P1037	280	28	2.44
	1290	Tubería 3.5	200	20	2.58
	1300	Tubería 4.3	210	21	2.60
	1310	Tubería 3.6	220	22	2.62
	1320	Tubería 4.4	230	23	2.64
	1330	Tubería 3.7	240	24	2.66
	1340	Tubería 4.5	250	25	2.68
	1350	Tubería 3.8	260	26	2.70

```

162 SELECT p.denominacion, SUM(e.cantidad * m.precio) AS total_pagar
163 FROM proyectos p
164 JOIN entregan e ON p.numero = e.numero
165 JOIN materiales m ON e.clave = m.clave
166 GROUP BY p.denominacion;
167

```

100% 25:166

Result Grid Filter Rows: Search Export:

denominacion	total_pagar
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

```

168 • CREATE VIEW vista_proveedores_televisa AS
169 SELECT DISTINCT pr.rfc, pr.razonsocial, p.denominacion
170 FROM proveedores pr
171 JOIN entregan e ON pr.rfc = e.rfc
172 JOIN proyectos p ON e.numero = p.numero
173 WHERE p.denominacion = 'Televisa en acción';
174
175 • CREATE VIEW vista_proveedores_coahuila AS
176 SELECT DISTINCT pr.rfc
177 FROM proveedores pr
178 JOIN entregan e ON pr.rfc = e.rfc
179 JOIN proyectos p ON e.numero = p.numero
180 WHERE p.denominacion = 'Educando en Coahuila';
181
182 • SELECT *
183 FROM vista_proveedores_televisa
184 WHERE rfc NOT IN (SELECT rfc FROM vista_proveedores_coahuila);
185

```

100% 42:175

Result Grid Filter Rows: Search Export:

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

```

186 • SELECT DISTINCT pr.rfc, pr.razonsocial, p.denominacion
187 FROM proveedores pr
188 JOIN entregan e ON pr.rfc = e.rfc
189 JOIN proyectos p ON e.numero = p.numero
190 WHERE p.denominacion = 'Televisa en acción'
191 AND pr.rfc NOT IN (
192     SELECT DISTINCT pr2.rfc
193     FROM proveedores pr2
194     JOIN entregan e2 ON pr2.rfc = e2.rfc
195     JOIN proyectos p2 ON e2.numero = p2.numero
196     WHERE p2.denominacion = 'Educando en Coahuila'
197 );

```

100% 20:187

Result Grid Filter Rows: Search Export:

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

```

199 • SELECT DISTINCT m.clave, m.descripcion, m.precio, e.cantidad, (m.precio * e.cantidad) AS costo_total
200 FROM materiales m
201 JOIN entregan e ON m.clave = e.clave
202 JOIN proyectos p ON e.numero = p.numero
203 JOIN proveedores pr ON e.rfc = pr.rfc
204 WHERE p.denominacion = 'Televisa en acción'
205 AND pr.rfc IN (
206     SELECT DISTINCT e2.rfc
207     FROM entregan e2
208     JOIN proyectos p2 ON e2.numero = p2.numero
209     WHERE p2.denominacion = 'Educando en Coahuila'
210 );

```

100% 51:209

Result Grid Filter Rows: Search Export:

	clave	descripcion	precio	cantidad	costo_total
	1080	Ladrillos rojos	50	86	4300
	1280	Tepetate	34	107	3638