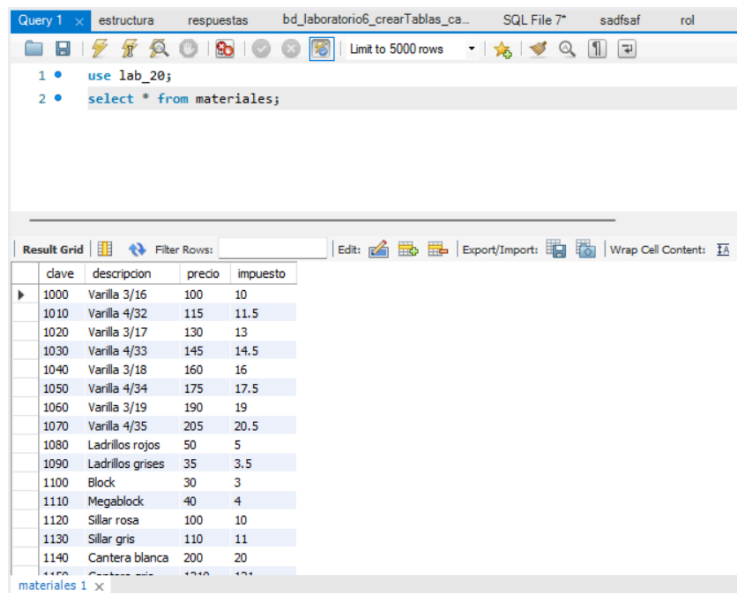


Algebra relacional.
materiales

SQL:



```
Query 1 x estructura respuestas bd_laboratorio6_crearTablas_ca... SQL File 7* sadfsaf rol
1 • use lab_20;
2 • select * from materiales;
```

	clave	descripcion	precio	impuesto
▶	1000	Varilla 3/16	100	10
	1010	Varilla 4/32	115	11.5
	1020	Varilla 3/17	130	13
	1030	Varilla 4/33	145	14.5
	1040	Varilla 3/18	160	16
	1050	Varilla 4/34	175	17.5
	1060	Varilla 3/19	190	19
	1070	Varilla 4/35	205	20.5
	1080	Ladrillos rojos	50	5
	1090	Ladrillos grises	35	3.5
	1100	Block	30	3
	1110	Megablock	40	4
	1120	Sillar rosa	100	10
	1130	Sillar gris	110	11
	1140	Cantera blanca	200	20

Selección

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

SQL

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



```
4 • select * from materiales
5   where clave=1000
6
```

	clave	descripcion	precio	impuesto
▶	1000	Varilla 3/16	100	10
•	NULL	NULL	NULL	NULL

Proyección

Algebra relacional.
 $PR_{\{clave,rfc,fecha\}}(entregan)$

SQL

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

```

6
7 • select clave,rfc,fecha from entregan
8

```

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

Reunión Natural

Algebra relacional.
entregan JN materiales

SQL

```

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

```

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

No aparecerá en el resultado de esta consulta. Esto se debe a que la condición `materiales.clave = entregan.clave` asegura que solo se incluyan los registros donde haya una correspondencia entre las claves en ambas tablas.

Query 1 x estructura respuestas bd_laboratorio6_crearTablas_ca... SQL

```

7 • select clave,rfc,fecha from entregan;
8
9 • select * from materiales,entregan
10 where materiales.clave = entregan.clave;
11
12

```

	clave	descripcion	precio	impuesto	clave	rfc	numero	fecha
▶	1000	Varilla 3/16	100	10	1000	AAAA800101	5000	2001-12-13
	1000	Varilla 3/16	100	10	1000	AAAA800101	5019	1999-04-07
	1010	Varilla 4/32	115	11.5	1010	BBBB800101	5001	1998-07-28
	1010	Varilla 4/32	115	11.5	1010	BBBB800101	5018	1997-09-12
	1020	Varilla 3/17	130	13	1020	CCCC800101	5002	2003-12-16
	1020	Varilla 3/17	130	13	1020	CCCC800101	5017	2000-05-21
	1030	Varilla 4/33	145	14.5	1030	DDDD800101	5003	1998-01-12
	1030	Varilla 4/33	145	14.5	1030	DDDD800101	5016	2005-04-30
	1040	Varilla 3/18	160	16	1040	EEEE800101	5004	1999-12-18
	1040	Varilla 3/18	160	16	1040	EEEE800101	5015	1999-12-18
	1050	Varilla 4/34	175	17.5	1050	FFFF800101	5005	2004-03-15
	1050	Varilla 4/34	175	17.5	1050	FFFF800101	5014	2000-05-18
	1060	Varilla 3/19	190	19	1060	GGGG800101	5006	2004-03-15
	1060	Varilla 3/19	190	19	1060	GGGG800101	5013	2002-11-14

Reunión con criterio específico

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

SQL

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

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

```
11
12 • select * from entregan,proyectos
13     where entregan.numero <= proyectos.numero;
14
```

	clave	rfc	numero	fecha	cantidad	numero	denominador
▶	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

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)
```

```
15 • (select * from entregan where clave=1450)
16   union
17   (select * from entregan where clave=1300)
18
```

	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 = 1450 OR clave = 1300;

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

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

Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

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

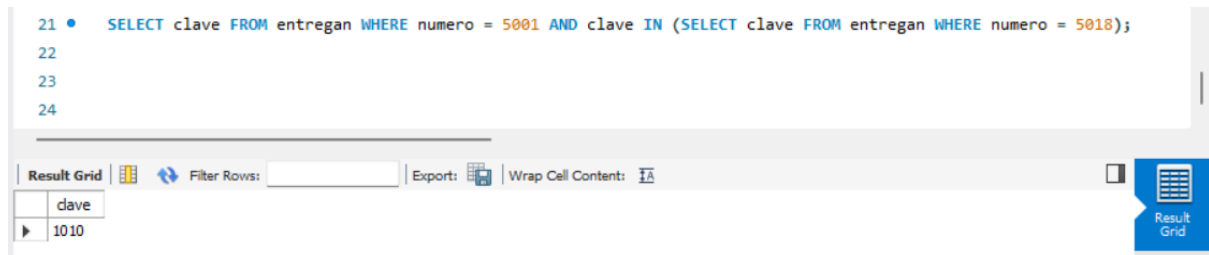
SQL

(select clave from entregan where numero=5001)

intersect

(select clave from entregan where numero=5018)

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



Diferencia (se ilustra con selección)

Algebra relacional.

$entregan - SL\{clave=1000\}(entregan)$

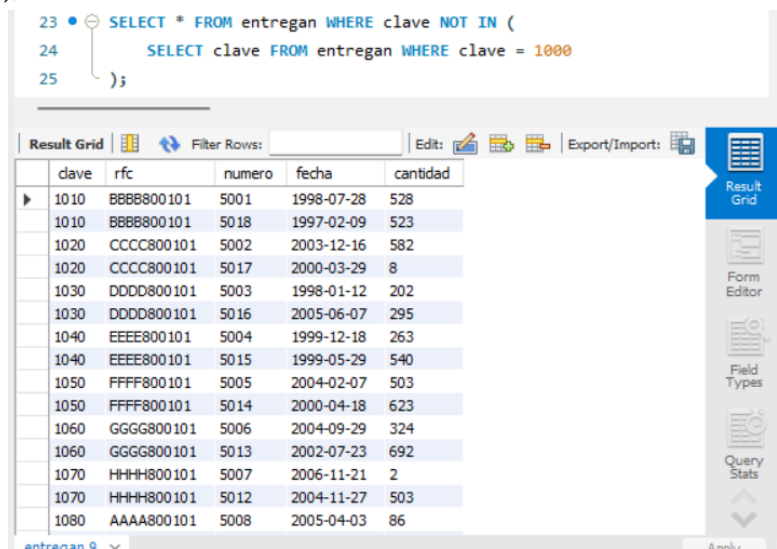
SQL

(select * from entregan)

minus

(select * from entregan where clave=1000)

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



Producto cartesiano

Algebra relacional.

$entregan \times materiales$

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?

si la tabla entregan tiene 'n' tuplas y la tabla materiales tiene 'm' tuplas, el número total de tuplas en el resultado será 'n * m'.

27 • `select * from entregan,materiales;`

	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

Construcción de consultas a partir de una especificación

SET DATEFORMAT dmy;

SELECT m.descripcion

FROM entregan e

JOIN materiales m ON e.clave_material = m.clave

WHERE YEAR(e.fecha) = 2000;

¿Por qué aparecen varias veces algunas descripciones de material?

Ya que hay múltiples entregas de ese mismo material realizadas en el año 2000.

```
31 • SELECT m.descripcion
32 FROM entregan e
33 JOIN materiales m ON e.clave = m.clave
34 WHERE YEAR(e.fecha) = 2000;
35
```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

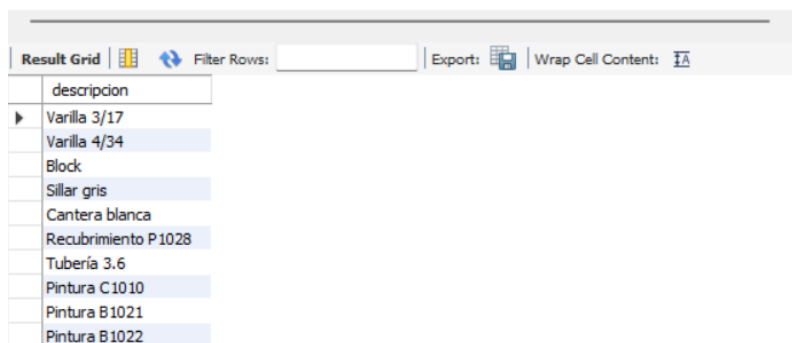
descripcion
Varilla 4/34
Block
Sillar gris
Sillar gris
Cantera blanca
Recubrimiento P1028
Tubería 3.6
Pintura C1010
Pintura B1021
Pintura B1021
Pintura B1022

Uso del calificador distinct

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

¿Qué resultado obtienes en esta ocasión?

```
36 • SELECT DISTINCT m.descripcion
37 FROM entregan e
38 JOIN materiales m ON e.clave = m.clave
39 WHERE YEAR(e.fecha) = 2000;
40
```



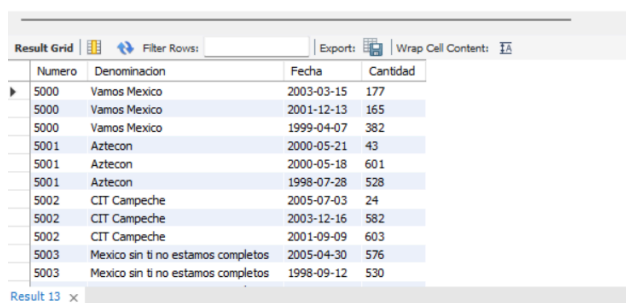
The screenshot shows a database interface with a 'Result Grid' tab. The grid contains a single column labeled 'descripcion'. The data is as follows:

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

Ordenamientos.

```
SELECT p.Numero, p.Denominacion, e.Fecha, e.Cantidad
FROM Proyectos p
JOIN Entregan e ON p.Numero = e.Numero
ORDER BY p.Numero ASC, e.Fecha DESC;
```

```
41 • SELECT p.Numero, p.Denominacion, e.Fecha, e.Cantidad
42 FROM Proyectos p
43 JOIN Entregan e ON p.Numero = e.Numero
44 ORDER BY p.Numero ASC, e.Fecha DESC;
45
46
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains four columns: 'Numero', 'Denominacion', 'Fecha', and 'Cantidad'. The data is as follows:

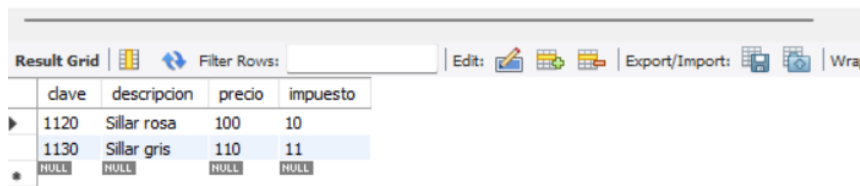
Numero	Denominacion	Fecha	Cantidad
5000	Vamos Mexico	2003-03-15	177
5000	Vamos Mexico	2001-12-13	165
5000	Vamos Mexico	1999-04-07	382
5001	Aztecon	2000-05-21	43
5001	Aztecon	2000-05-18	601
5001	Aztecon	1998-07-28	528
5002	CIT Campeche	2005-07-03	24
5002	CIT Campeche	2003-12-16	582
5002	CIT Campeche	2001-09-09	603
5003	Mexico sin ti no estamos completos	2005-04-30	576
5003	Mexico sin ti no estamos completos	1998-09-12	530

Operadores de cadena

```
SELECT * FROM materiales where Descripcion LIKE 'Si%'
```

¿Qué resultado obtienes?

46 • `SELECT * FROM materiales where Descripcion LIKE 'Si%';`



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

Explica que hace el símbolo '%'.

El símbolo % se utiliza para que en la consulta seleccione todas las filas de la tabla materiales donde la columna descripción comience con la cadena "Si", seguida de cero o más caracteres adicionales.

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

La consulta LIKE 'Si' buscaría solamente coincidencias exactas con la cadena "Si". Solo mostraria aquellos materiales cuya descripción es "Si", sin caracteres adicionales antes o después.

¿Qué resultado obtienes?

48 • `SELECT * FROM materiales where Descripcion LIKE 'Si';`



	clave	descripcion	precio	impuesto
*	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL

Explica a qué se debe este comportamiento.

Esto se debe a que no existen materiales cuya descripción sea "si"

```
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? error al ejecutar el código

¿Para qué sirve DECLARE? Se utiliza para declarar una variable local.

¿Cuál es la función de @foo? Se utiliza para almacenar una parte del texto.

¿Que realiza el operador SET? Se utiliza para asignar un valor a una variable

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

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

- Comportamiento: Esta consulta selecciona todos los valores de la columna RFC de la tabla Entregan donde el valor de RFC comienza con cualquier letra entre A y D.
- Función: El operador Like se utiliza para realizar comparaciones de patrones de texto. [A-D] especifica un rango de caracteres, por lo que la consulta seleccionará valores de RFC que comiencen con cualquier letra entre A y D.
- Resultado: Seleccionará todos los valores de RFC de la tabla Entregan que comienzan con una letra entre A y D.

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

- Comportamiento: Esta consulta selecciona todos los valores de la columna RFC de la tabla Entregan donde el valor de RFC no comienza con la letra A.
- Función: [^A] especifica que no debe coincidir con el carácter A al principio del valor de RFC.
- Resultado: Seleccionará todos los valores de RFC de la tabla Entregan que no comiencen con la letra A.

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

- Comportamiento: Esta consulta selecciona todos los valores de la columna Numero de la tabla Entregan donde el valor de Numero tiene tres caracteres seguidos por el número 6 al final.
- Función: ____6 especifica un patrón donde _ representa cualquier carácter y 6 representa el número 6.
- Resultado: Seleccionará todos los valores de Numero de la tabla Entregan que tengan tres caracteres seguidos por el número 6 al final, por ejemplo, 'ABC6', '1236', etc.

Operadores Lógicos.

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

```
58 • SELECT Clave,RFC,Numero,Fecha,Cantidad
59 FROM Entregan
60 WHERE Numero Between 5000 and 5010;
61
62
```

Clave	RFC	Numero	Fecha	Cantidad
1000	AAAA800101	5000	2001-12-13	165
1010	BBBB800101	5001	1998-07-28	528
1020	CCCC800101	5002	2003-12-16	582
1030	DDDD800101	5003	1998-01-12	202
1040	EEEE800101	5004	1999-12-18	263
1050	FFFF800101	5005	2004-02-07	503
1060	GGGG800101	5006	2004-09-29	324
1070	HHHH800101	5007	2006-11-21	2
1080	AAAA800101	5008	2005-04-03	86
1090	BBBB800101	5009	1997-03-13	73
1090	BBBB800101	5010	1998-11-17	421

¿Cómo filtrarías rangos de fechas?

```
SELECT Clave, RFC, Numero, Fecha, Cantidad
FROM Entregan
WHERE Fecha BETWEEN '2003-01-01' AND '2005-01-31';
```

```
62 • SELECT Clave, RFC, Numero, Fecha, Cantidad
63 FROM Entregan
64 WHERE Fecha BETWEEN '2003-01-01' AND '2005-01-31';
65
66
```

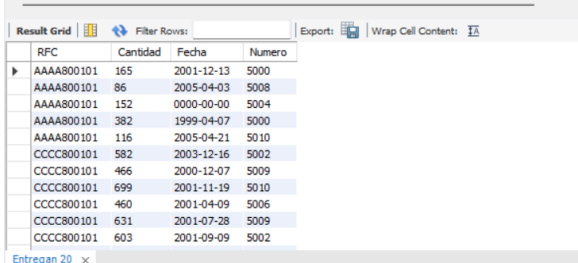
Clave	RFC	Numero	Fecha	Cantidad
1020	CCCC800101	5002	2003-12-16	582
1050	FFFF800101	5005	2004-02-07	503
1060	GGGG800101	5006	2004-09-29	324
1070	HHHH800101	5012	2004-11-27	503
1080	AAAA800101	5011	2003-08-28	699
1120	EEEE800101	5007	2003-11-21	692
1160	AAAA800101	5019	2003-05-05	244
1190	DDDD800101	5019	2004-11-19	94
1200	EEEE800101	5000	2003-03-15	177
1230	HHHH800101	5012	2004-08-23	312
1250	BBBB800101	5005	2003-10-12	71

```
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?

```
66 • SELECT RFC,Cantidad, Fecha,Numero
67 FROM Entregan
68 WHERE Numero Between 5000 and 5010 AND
69 Exists ( SELECT RFC
70 FROM Proveedores
71 WHERE RazonSocial LIKE 'La%' and Entregan.RFC = Proveedores.RFC );
72
```



The screenshot shows a SQL query result grid with columns: RFC, Cantidad, Fecha, and Numero. The data is filtered to show records where the 'Numero' is between 5000 and 5010. The 'RFC' values are a mix of 'AAAA800101' and 'CCCC800101'. The 'Fecha' values range from 2001-12-13 to 2001-09-09. The 'Numero' values are 5000, 5008, 5004, 5000, 5010, 5002, 5009, 5010, 5006, 5009, and 5002.

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
CCCC800101	631	2001-07-28	5009
CCCC800101	603	2001-09-09	5002

La consulta selecciona columnas de la tabla `Entregan` donde el número esté en el rango de 5000 a 5010 y donde exista al menos una fila en la tabla `Proveedores` donde la Razón Social del proveedor comienza con "La" y coincide con el RFC de la tabla `Entregan`.

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

La función del paréntesis `()` después de `EXISTS` se utiliza para delimitar una subconsulta. EXISTS verifica si al menos una fila devuelve un resultado de esta subconsulta para cada fila de la tabla `[Entregan]`. Si la subconsulta devuelve algún resultado para una fila específica de `[Entregan]`, entonces `EXISTS` devuelve verdadero para esa fila, lo que significa que se incluirá en el resultado final de la consulta principal.

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador 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%'
);
```

```

73 • SELECT RFC, Cantidad, Fecha, Numero
74 FROM Entregan
75 WHERE Numero BETWEEN 5000 AND 5010
76 AND RFC IN (
77     SELECT RFC
78     FROM Proveedores
79     WHERE RazonSocial LIKE 'La%'
80 );
81

```

Result Grid				
Filter Rows:				
Export: Wrap Cell Content:				
RFC	Cantidad	Fecha	Numero	
AAA800101	165	2001-12-13	5000	
AAA800101	86	2005-04-03	5008	
AAA800101	152	0000-00-00	5004	
AAA800101	382	1999-04-07	5000	
AAA800101	116	2005-04-21	5010	
CCC800101	582	2003-12-16	5002	
CCC800101	466	2000-12-07	5009	
CCC800101	699	2001-11-19	5010	
CCC800101	460	2001-04-09	5006	
CCC800101	631	2001-07-28	5009	
CCC800101	603	2001-09-09	5002	

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.

```

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%'
)

```

```

82 • SELECT RFC, Cantidad, Fecha, Numero
83 FROM Entregan
84 WHERE Numero BETWEEN 5000 AND 5010
85 AND RFC NOT IN (
86     SELECT RFC
87     FROM Proveedores
88     WHERE RazonSocial LIKE 'La%'
89 );

```

Result Grid				
Filter Rows:				
Export: Wrap Cell Content:				
RFC	Cantidad	Fecha	Numero	
BBB800101	528	1998-07-28	5001	
DDD800101	202	1998-01-12	5003	
EEE800101	263	1999-12-18	5004	
FFF800101	503	2004-02-07	5005	
GGG800101	324	2004-09-29	5006	
HHH800101	2	2006-11-21	5007	
BBB800101	73	1997-03-13	5009	
BBB800101	421	1998-11-17	5010	
DDD800101	337	2005-06-03	5008	
EEE800101	692	2003-11-21	5007	
FFF800101	562	2000-04-13	5006	

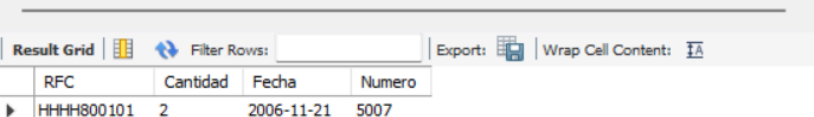
```

SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Cantidad > ALL (
    SELECT Cantidad
    FROM Entregan
    WHERE RFC IN (
        SELECT RFC
        FROM Proveedores
        WHERE RazonSocial LIKE 'La%'
    )
)

```

)
);

```
91 • SELECT RFC, Cantidad, Fecha, Numero
92 FROM Entregan
93 WHERE Cantidad < ALL (
94     SELECT Cantidad
95     FROM Entregan
96     WHERE RFC IN (
97         SELECT RFC
98         FROM Proveedores
99         WHERE RazonSocial LIKE 'La%'
100     )
101 );
```



The screenshot shows a SQL query editor with a query that selects columns from 'Entregan' and filters them based on a subquery. Below the query, there is a 'Result Grid' with columns 'RFC', 'Cantidad', 'Fecha', and 'Numero'. The first row of data is 'HHHH800101', '2', '2006-11-21', and '5007'.

RFC	Cantidad	Fecha	Numero
HHHH800101	2	2006-11-21	5007

¿Qué hace la siguiente sentencia? Explica por qué.

```
SELECT TOP 2 * FROM Proyectos
```

Devuelve las primeras 2 filas de la tabla Proyectos. La cláusula TOP se utiliza para limitar el número de filas devueltas por una consulta. Por eso en este ejemplo, se seleccionan las primeras 2 filas de la tabla Proyectos sin ningún criterio de ordenamiento específico, por lo que las filas seleccionadas pueden variar dependiendo de la implementación de la base de datos.

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

```
SELECT TOP Numero FROM Proyectos
```

Por otro lado, la consulta `SELECT TOP Numero FROM Proyectos` dará como resultado un error. La cláusula `TOP` requiere un valor numérico constante que indique el número de filas a seleccionar, pero en este caso, se está intentando usar el valor de una columna (`Numero`) para determinar cuántas filas se deben seleccionar. Esto no es válido en SQL. La cláusula `TOP` no puede tomar un valor de columna como argumento. Debe ser un número constante o una expresión numérica constante.

Modificando la estructura de un tabla existente.

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

```
ALTER TABLE materiales ADD PorcentajImpuesto 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 PorcentajImpuesto = 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.

	clave	descripcion	precio	impuesto	PorcentajeImpuesto
▶	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
	1040	Varilla 3/18	160	16	2.08
	1050	Varilla 4/34	175	17.5	2.10
	1060	Varilla 3/19	190	19	2.12
	1070	Varilla 4/35	205	20.5	2.14
	1080	Ladrillos rojos	50	5	2.16
	1090	Ladrillos grises	35	3.5	2.18
	1100	Block	30	3	2.20
	1110	Megablock	40	4	2.22
	1120	Sillar rosa	100	10	2.24
	1130	Sillar gris	110	11	2.26
	1140	Cantera blanca	200	20	2.28
	1150	Cantera gris	1210	121	2.30
	1160	Cantera rosa	1420	142	2.32
	1170	Cantera amarilla	230	23	2.34

materiales 1 x

¿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.Clave,
e.RFC,
e.Numero,
e.Fecha,
e.Cantidad,
m.Precio AS Precio_Unitario,
m.Precio * (1 + (m.PorcentajeImpuesto / 100)) AS Precio_Con_Impuesto,
e.Cantidad * (m.Precio * (1 + (m.PorcentajeImpuesto / 100))) AS Importe_Total

FROM

Entregan e

JOIN

Materiales m ON e.Clave = m.Clave;

106 • SELECT

107 e.Clave,

108 e.RFC,

109 e.Numero,

110 e.Fecha,

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

	Clave	RFC	Numero	Fecha	Cantidad	Precio_Unitario	Precio_Con_Impuesto	Importe_Total
▶	1000	AAAA800101	5000	2001-12-13	165	100	102	16830
	1000	AAAA800101	5019	1999-07-13	254	100	102	25908
	1010	BBBB800101	5001	1998-07-28	528	115	117.323	61946.543999999994
	1010	BBBB800101	5018	1997-02-09	523	115	117.323	61359.929
	1020	CCCC800101	5002	2003-12-16	582	130	132.652	77203.463999999999
	1020	CCCC800101	5017	2000-03-29	8	130	132.652	1061.216
	1030	DDDD800101	5003	1998-01-12	202	145	147.987	29893.374
	1030	DDDD800101	5016	2005-06-07	295	145	147.987	43656.165
	1040	EEEE800101	5004	1999-12-18	263	160	163.32799999999997	42955.263999999996
	1040	EEEE800101	5015	1999-05-29	540	160	163.32799999999997	88197.119999999998
	1050	FFFF800101	5005	2004-02-07	503	175	178.67499999999998	89873.525
	1050	FFFF800101	5014	2000-04-18	623	175	178.67499999999998	111314.525
	1060	GGGG800101	5006	2004-09-29	324	190	194.028000000000002	62865.072000000001
	1060	GGGG800101	5013	2002-07-23	692	190	194.028000000000002	134267.376000000002
	1070	HHHH800101	5007	2006-11-21	2	205	209.387000000000003	418.77400000000006
	1070	HHHH800101	5012	2004-11-27	503	205	209.387000000000003	105321.661000000001
	1080	AAAA800101	5008	2005-04-03	86	50	51.080000000000005	4392.88
	1080	AAAA800101	5011	2003-08-28	699	50	51.080000000000005	35704.92000000000006

Creación de vistas

Vista 1: Importe total de las entregas basado en la cantidad, el precio del material y el impuesto asignado

CREATE VIEW VistaImporteEntregas AS
SELECT

e.Clave,
 e.RFC,
 e.Numero,
 e.Fecha,
 e.Cantidad,
 m.Costo AS Precio_Unitario,
 m.Costo * (1 + (m.PorcentajImpuesto / 100)) AS Precio_Con_Impuesto,
 e.Cantidad * (m.Costo * (1 + (m.PorcentajImpuesto / 100))) AS Importe_Total

FROM

Entregan e

JOIN

Materiales m ON e.Clave = m.Clave;

Clave	RFC	Numero	Fecha	Cantidad	Precio_Unitario	Precio_Con_Impuesto	Importe_Total
1000	AAAA800101	5000	2001-12-13	165	100	102	16830
1000	AAAA800101	5019	1999-07-13	254	100	102	25908
1010	BBBB800101	5001	1998-07-28	528	115	117.323	61946.543999999994
1010	BBBB800101	5018	1997-02-09	523	115	117.323	61359.929
1020	CCCC800101	5002	2003-12-16	582	130	132.652	77203.463999999999
1020	CCCC800101	5017	2000-03-29	8	130	132.652	1061.216
1030	DDDD800101	5003	1998-01-12	202	145	147.987	29893.374
1030	DDDD800101	5016	2005-06-07	295	145	147.987	43656.165
1040	EEEE800101	5004	1999-12-18	263	160	163.32799999999997	42955.263999999996
1040	EEEE800101	5015	1999-05-29	540	160	163.32799999999997	88197.119999999998
1050	FFFF800101	5005	2004-02-07	503	175	178.67499999999998	89873.525
1050	FFFF800101	5014	2000-04-18	623	175	178.67499999999998	111314.525
1060	GGGG800101	5006	2004-09-29	324	190	194.028000000000002	62865.072000000001
1060	GGGG800101	5013	2002-07-23	692	190	194.028000000000002	134267.376000000002
1070	HHHH800101	5007	2006-11-21	2	205	209.387000000000003	418.77400000000006
1070	HHHH800101	5012	2004-11-27	503	205	209.387000000000003	105321.661000000001
1080	AAAA800101	5008	2005-04-03	86	50	51.080000000000005	4392.88
1080	AAAA800101	5011	2003-08-28	699	50	51.080000000000005	35704.920000000006

Vista 2 Lista de todos los materiales con su precio unitario y porcentaje de impuesto

CREATE VIEW VistaMateriales AS

SELECT

Clave,
 Descripción,
 Costo AS Precio_Unitario,
 PorcentajImpuesto

FROM

Materiales;

1 • SELECT * FROM lab_20.vistamateriales;

Clave	Descripción	Precio_Unitario	PorcentajImpuesto
1000	Varilla 3/16	100	2.00
1010	Varilla 4/32	115	2.02
1020	Varilla 3/17	130	2.04
1030	Varilla 4/33	145	2.06
1040	Varilla 3/18	160	2.08
1050	Varilla 4/34	175	2.10
1060	Varilla 3/19	190	2.12
1070	Varilla 4/35	205	2.14
1080	Ladrillos rojos	50	2.16
1090	Ladrillos grises	35	2.18
1100	Block	30	2.20
1110	Megablock	40	2.22
1120	Sillar rosa	100	2.24
1130	Sillar gris	110	2.26
1140	Cantera blanca	200	2.28
1150	Cantera gris	1210	2.30
1160	Cantera rosa	1420	2.32
1170	Cantera amarilla	230	2.34

Vista 3 para obtener proveedores con la cantidad total de entregas realizadas por cada uno

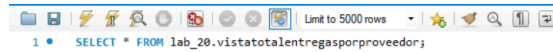
CREATE VIEW VistaTotalEntregasPorProveedor AS

SELECT

```

RFC,
COUNT(*) AS Total_Entregas
FROM
Entregan
GROUP BY
RFC;

```



RFC	Total_Entregas
AAAA800101	12
BBBB800101	12
CCCC800101	12
DDDD800101	11
EEEE800101	10
FFFF800101	10
GGGG800101	10
HHHH800101	10

Vista 4 para obtener la lista de proyectos con la cantidad total de materiales entregados en cada uno:

```

CREATE VIEW VistaTotalMaterialesPorProyecto AS
SELECT
Numero,
SUM(Cantidad) AS Total_Materiales_Entregados
FROM
Entregan
GROUP BY
Numero;

```

Numero	Total_Materiales_Entregados
5001	1172
5002	1209
5003	1308
5004	868
5005	1678
5006	1625
5007	1544
5008	1354
5009	1769
5010	2046
5011	2202
5012	2373
5013	2542
5014	1501
5015	1855

Vista 5 para la lista de materiales con su costo y el proveedor que suministra cada uno:

```

CREATE VIEW VistaMaterialesProveedores AS
SELECT
m.Clave,
m.Descripcion,
m.Precio,
p.RazonSocial AS Proveedor
FROM
Materiales m
JOIN
Entregan e ON m.Clave = e.Clave
JOIN
Proveedores p ON e.RFC = p.RFC;

```

Limit to 5000 rows

1 • `SELECT * FROM lab_20.vistamaterialesproveedores;`

Clave	Descripcion	Precio	Proveedor
1000	Varilla 3/16	100	La fragua
1000	Varilla 3/16	100	La fragua
1080	Ladrillos rojos	50	La fragua
1080	Ladrillos rojos	50	La fragua
1160	Cantera rosa	1420	La fragua
1160	Cantera rosa	1420	La fragua
1240	Arena	200	La fragua
1240	Arena	200	La fragua
1320	Tubería 4.4	230	La fragua
1320	Tubería 4.4	230	La fragua
1400	Pintura C1011	125	La fragua
1400	Pintura C1011	125	La fragua
1010	Varilla 4/32	115	Oviedo
1010	Varilla 4/32	115	Oviedo
1090	Ladrillos grises	35	Oviedo

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 repor

Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos". **3 resultados**

177 • `SELECT m.Clave, m.Descripcion`
 178 `FROM Materiales m`
 179 `JOIN Entregan e ON m.Clave = e.Clave`
 180 `JOIN Proyectos p ON e.Numero = p.Numero`
 181 `WHERE p.Denominacion = 'México sin ti no estamos completos';`
 182
 183

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". **0 resultados**

183 • `SELECT Clave, Descripcion`
 184 `FROM Materiales`
 185 `WHERE Clave IN (`
 186 `SELECT Clave`
 187 `FROM Entregan`
 188 `WHERE RFC = 'Acme tools'`
 189 `);`
 190

Clave	Descripcion
-------	-------------

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

```

191 • SELECT RFC
192 FROM Entregan
193 WHERE YEAR(Fecha) = 2000
194 GROUP BY RFC
195 HAVING AVG(Cantidad) >= 300;
196
197

```

Result Grid	Filter Rows:	Export:	Wrap C
RFC			
BBB800101			
FFF800101			
GGG800101			

El Total entregado por cada material en el año 2000. **11 resultados**

```

197 • SELECT Clave, SUM(Cantidad) AS Total_Entregado
198 FROM Entregan
199 WHERE YEAR(Fecha) = 2000
200 GROUP BY Clave;
201

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Clave	Total_Entregado		
1020	8		
1050	623		
1100	466		
1130	625		
1140	583		

La Clave del material más vendido durante el 2001. (se recomienda usar una vista intermedia para su solución) **7 resultados**

```

1 • SELECT * FROM lab_20.materialmasvendido2001;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
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. **12 resultados**

```

212 • SELECT Clave, Descripcion
213 FROM Materiales
214 WHERE Descripcion LIKE '%ub%';
215

```

Result Grid	Filter Rows:	Export:
Clave	Descripcion	
1180	Recubrimiento P1001	
1190	Recubrimiento P1010	
1200	Recubrimiento P1019	

Denominación y suma del total a pagar para todos los proyectos. **20 resultados**


```

216 • SELECT p.Numero, p.Denominacion, SUM(e.Cantidad * m.Precio) AS Total_Pagar
217 FROM Entregan e
218 JOIN Materiales m ON e.Clave = m.Clave
219 JOIN Proyectos p ON e.Numero = p.Numero
220 GROUP BY p.Numero, p.Denominacion;
221
222

```

Numero	Denominacion	Total_Pagar
5000	Vamos Mexico	106730
5001	Aztecon	146595
5002	CIT Campeche	157755

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). **2 resultados**

```

CREATE VIEW ProveedoresTelevisa AS
SELECT DISTINCT e.RFC
FROM Entregan e
JOIN Proyectos p ON e.Numero = p.Numero
WHERE p.Denominacion = 'Televisa en acción';

```

```

CREATE VIEW ProveedoresEducandoCoahuila AS
SELECT DISTINCT e.RFC
FROM Entregan e
JOIN Proyectos p ON e.Numero = p.Numero
WHERE p.Denominacion = 'Educando en Coahuila';

```

```

SELECT DISTINCT p.RFC, p.RazonSocial
FROM Proveedores p
JOIN ProveedoresTelevisa pt ON p.RFC = pt.RFC
LEFT JOIN ProveedoresEducandoCoahuila pec ON p.RFC = pec.RFC
WHERE pec.RFC IS NULL;

```

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). **2 resultados**

```

241 • SELECT DISTINCT p.RFC, p.RazonSocial
242 FROM Proveedores p
243 JOIN Entregan e ON p.RFC = e.RFC
244 JOIN Proyectos pr ON e.Numero = pr.Numero
245 WHERE pr.Denominacion = 'Televisa en acción'
246 AND p.RFC NOT IN (SELECT DISTINCT p2.RFC
247 FROM Proveedores p2
248 JOIN Entregan e2 ON p2.RFC = e2.RFC
249 JOIN Proyectos pr2 ON e2.Numero = pr2.Numero
250 WHERE pr2.Denominacion = 'Educando en Coahuila');
251

```

RFC	RazonSocial
CCCC800101	La Ferre
DDDD800101	Cecoferre

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. **2 resultados**

```

252 • SELECT DISTINCT m.Clave, m.Precio, m.Descripcion
253 FROM Materiales m
254 JOIN Entregan e ON m.Clave = e.Clave
255 JOIN Proyectos p ON e.Numero = p.Numero
256 JOIN Proveedores pr ON e.RFC = pr.RFC
257 WHERE p.Denominacion = 'Televisa en acción'
258 AND pr.RFC IN (SELECT DISTINCT pr2.RFC
259                FROM Proveedores pr2
260                JOIN Entregan e2 ON pr2.RFC = e2.RFC
261                JOIN Proyectos p2 ON e2.Numero = p2.Numero
262                WHERE p2.Denominacion = 'Educando en Coahuila');
263
264

```

Result Grid		
Filter Rows:	Export:	Wrap Cell Content:
Clave	Precio	Descripcion
1080	50	Ladrillos rojos
1280	34	Tepetate

Reto: Usa solo el operador NOT IN en la consulta anterior (No es parte de la entrega).

```

SELECT DISTINCT m.Clave, m.Costo, m.Descripcion
FROM Materiales m
JOIN Entregan e ON m.Clave = e.Clave
JOIN Proyectos p ON e.Numero = p.Numero
JOIN Proveedores pr ON e.RFC = pr.RFC
WHERE p.Denominacion = 'Televisa en acción'
AND pr.RFC NOT IN (SELECT DISTINCT pr2.RFC
                   FROM Proveedores pr2
                   JOIN Entregan e2 ON pr2.RFC = e2.RFC
                   JOIN Proyectos p2 ON e2.Numero = p2.Numero
                   WHERE p2.Denominacion = 'Educando en Coahuila');

```

Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos. **42 resultados**

```

264 • SELECT m.Descripcion AS Nombre_Material,
265          COUNT(*) AS Veces_Entregados,
266          SUM(e.Cantidad * m.Precio) AS Total_Costo
267 FROM Materiales m
268 JOIN Entregan e ON m.Clave = e.Clave
269 GROUP BY m.Descripcion;
270
271

```

Result Grid		
Filter Rows:	Export:	Wrap Cell Content:
Nombre_Material	Veces_Entregados	Total_Costo
Varilla 3/16	2	41900
Varilla 4/32	2	120865
Varilla 3/17	2	76700
Varilla 4/33	2	72065