

## Laboratorio 20: Consultas en SQL

### Consulta de un tabla completa

SQL

```
select * from materiales
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a script editor with the following code:

```
USE Lab6;
select * from materiales;
```

In the bottom-right pane, there is a grid viewer titled "materiales 1" showing the results of the query. The table has three columns: clave, descripcion, and precio. The data is as follows:

	clave	descripcion	precio
1	1,000	Varilla 3/16	100
2	1,010	Varilla 4/32	115
3	1,020	Varilla 3/17	130
4	1,030	Varilla 4/33	145
5	1,040	Varilla 3/18	160
6	1,050	Varilla 4/34	175
7	1,060	Varilla 3/19	190
8	1,070	Varilla 4/35	205
9	1,080	Ladrillos rojos	50
10	1,090	Ladrillos grises	35
11	1,100	Block	30
12	1,110	Megablock	40
13	1,120	Sillar rosa	100
14	1,130	Sillar gris	110
15	1,140	Cantera blanca	200
16	1,150	Cantera gris	1210

### Selección

Algebra relacional.

```
SL{clave=1000}(materiales)
```

SQL

```
select * from materiales
```

```
where clave=1000
```

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a script editor with the following code:

```
*<localhost> Script-5 X
select * from materiales
where clave=1000
```

In the bottom-right pane, there is a grid viewer titled "materiales 1" showing the results of the query. The table has three columns: clave, descripcion, and precio. The data is as follows:

	clave	descripcion	precio
1	1,000	Varilla 3/16	100

## Proyección

Algebra relacional.

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

SQL

`select clave, rfc, fecha from entregan`

The screenshot shows the MySQL Workbench interface with a query editor window titled '<localhost> Script-5'. The query `select clave, rfc, fecha from entregan` is entered. Below the editor is a results pane titled 'entregan 1' showing a table with three columns: 'clave', 'rfc', and 'fecha'. The data consists of 11 rows, each containing a unique value for 'clave' (ranging from 1,000 to 1,230) and corresponding values for 'rfc' and 'fecha'.

	clave	rfc	fecha
1	1,000	AAAA800101	1998-07-08
2	1,200	EEEE800101	2000-03-05
3	1,400	AAAA800101	2002-03-12
4	1,010	BBBB800101	2000-05-03
5	1,210	FFFF800101	1999-11-05
6	1,410	BBBB800101	2000-02-05
7	1,020	CCCC800101	2001-07-29
8	1,220	GGGG800101	2003-02-01
9	1,420	CCCC800101	1998-04-07
10	1,030	DDDD800101	1998-02-21
11	1,230	HHHH800101	2003-01-06

## Reunión Natural

Algebra relacional.

$\text{entregan JN materiales}$

SQL

`select * from materiales, entregan`

`where materiales.clave = entregan.clave`

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

No, porque la información de las tablas se despliega cuando la clave de material se encuentra también en la de entregan.

<localhost> Script-5 ×

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

materiales(+) 1 ×

```
• select * from materiales,entregan where ... Enter a SQL expression to filter results (use Ctrl+Space)
```

	clave	descripcion	precio	clave	rfc	numero	fecha	Valor
1	1,000	Varilla 3/16	100	clave bigint(20)	AAAA800101	5,000	1998-07-08	
2	1,000	Varilla 3/16	100	1,000	BBBB800101	5,019	1999-08-08	
3	1,010	Varilla 4/32	115	1,010	BBBB800101	5,001	2000-05-03	
4	1,010	Varilla 4/32	115	1,010	BBBB800101	5,018	2002-03-29	
5	1,020	Varilla 3/17	130	1,020	CCCC800101	5,002	2001-07-29	
6	1,020	Varilla 3/17	130	1,020	CCCC800101	5,017	1999-02-04	
7	1,030	Varilla 4/33	145	1,030	DDDD800101	5,003	1998-02-21	
8	1,030	Varilla 4/33	145	1,030	DDDD800101	5,016	2000-11-05	
9	1,040	Varilla 3/18	160	1,040	EEEE800101	5,004	1999-12-11	
10	1,040	Varilla 3/18	160	1,040	EEEE800101	5,015	2002-07-12	
11	1,050	Varilla 4/24	175	1,050	FFFF800101	5,005	2000-10-14	
12	1,050	Varilla 4/24	175	1,050	FFFF800101	5,014	1999-03-07	
13	1,060	Varilla 3/19	190	1,060	GGGG800101	5,006	2000-05-04	
14	1,060	Varilla 3/19	190	1,060	GGGG800101	5,013	2000-01-02	
15	1,070	Varilla 4/35	205	1,070	HHHH800101	5,007	1998-02-23	
16	1,070	Varilla 4/35	205	1,070	HHHH800101	5,019	1999-12-02	

## Reunión con criterio específico

Algebra relacional.

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

SQL

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

<localhost> Script-5 ×

```
• select * from entregan,proyecto  
  where entregan.numero <= proyecto.numero
```

entregan(+) 1 ×

```
• select * from entregan,proyecto where ... Enter a SQL expression to filter results (use Ctrl+Space)
```

	clave	rfc	numero	fecha	cantidad	numero	denominacion	Valor
1	1,000	AAAA800101	5,000	1998-07-08	165	5,000	Vamos Mexico	
2	1,200	EEEE800101	5,000	2000-03-05	177	5,000	Vamos Mexico	
3	1,400	AAAA800101	5,000	2002-03-12	382	5,000	Vamos Mexico	
4	1,000	AAAA800101	5,000	1998-07-08	165	5,001	AztecÚn	
5	1,200	EEEE800101	5,000	2000-03-05	177	5,001	AztecÚn	
6	1,400	AAAA800101	5,000	2002-03-12	382	5,001	AztecÚn	
7	1,010	BBBB800101	5,001	2000-05-03	528	5,001	AztecÚn	
8	1,210	FFFF800101	5,001	1999-11-05	43	5,001	AztecÚn	
9	1,410	BBBB800101	5,001	2000-02-05	601	5,001	AztecÚn	
10	1,000	AAAA800101	5,000	1998-07-08	165	5,002	CIT Campeche	
11	1,200	EEEE800101	5,000	2000-03-05	177	5,002	CIT Campeche	
12	1,400	AAAA800101	5,000	2002-03-12	382	5,002	CIT Campeche	
13	1,010	BBBB800101	5,001	2000-05-03	528	5,002	CIT Campeche	
14	1,210	FFFF800101	5,001	1999-11-05	43	5,002	CIT Campeche	
15	1,410	BBBB800101	5,001	2000-02-05	601	5,002	CIT Campeche	

## Unión (se ilustra junto con selección)

Algebra relacional.

SL{clave=1450}(entregan) UN SL{clave=1300}(entregan)

SQLº

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

union

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

The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a script editor window titled "localhost > Script-5" containing the following SQL code:

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

Below the script editor is a results pane titled "Resultados 2 X". It displays a table with the following data:

Grilla	clave	rfc	numero	fecha	cantidad
1	1,300	GGGG800101	5,005	2002-06-10	521
2	1,300	GGGG800101	5,010	2003-01-08	119

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

Una consulta que arroja un resultado idéntico sin usar unión es la siguiente: (select \* from entregan where clave=1300), esto se debe a que la clave 1450 no existe.

Compruébalo.

The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a results pane titled "entregan 1 X" containing the following SQL code:

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

Below the results pane is a results pane titled "Estadísticas 1". It displays a table with the same data as the previous screenshot:

Grilla	clave	rfc	numero	fecha	cantidad
1	1,300	GGGG800101	5,005	2002-06-10	521
2	1,300	GGGG800101	5,010	2003-01-08	119

### Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

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

SQL

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

The screenshot shows a MySQL Workbench interface. The top window is titled 'Script-5' and contains the following SQL code:

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

The bottom window is titled 'Resultados 1' and displays the query results in a grid:

Grilla	clave
1	1,010

### Diferencia (se ilustra con selección )

Algebra relacional.

entregan - SL{clave=1000}(entregan)

SQL

(select \* from entregan)

minus

(select \* from entregan where clave=1000)

The screenshot shows a MySQL Workbench interface. The top window is titled 'Script-5' and contains the following SQL code:

```
(select * from entregan)
except
(select * from entregan where clave=1000)
```

The bottom window is titled 'Resultados 1' and displays the query results in a grid:

clave	RFC	numero	fecha	cantidad
1,010	BBBB800101	5,001	2000-05-03	528
1,010	BBBB800101	5,018	2002-03-29	523
1,020	CCCC800101	5,002	2001-07-29	582
1,020	CCCC800101	5,017	1999-02-04	8
1,030	DDDD800101	5,003	1998-02-21	202
1,030	DDDD800101	5,016	2000-11-05	295
1,040	EEEE800101	5,004	1999-12-11	263
1,040	EEEE800101	5,015	2002-07-12	540
1,050	FFFF800101	5,005	2000-10-14	503
1,050	FFFF800101	5,014	1999-03-07	623
1,060	GGGG800101	5,006	2000-05-04	324
1,060	GGGG800101	5,013	2000-01-02	692
1,070	HHHH800101	5,007	1998-02-23	2

### 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?

R= Se multiplican el numero de tuplas en cada tabla

	clave	rfc	numero	fecha	cantidad	clave	descripcion
1	1,000	AAAAA800101	5,000	1998-07-08	165	1,000	Varilla 3/16
2	1,000	AAAAA800101	5,000	1998-07-08	165	1,010	Varilla 4/32
3	1,000	AAAAA800101	5,000	1998-07-08	165	1,020	Varilla 3/17
4	1,000	AAAAA800101	5,000	1998-07-08	165	1,030	Varilla 4/33
5	1,000	AAAAA800101	5,000	1998-07-08	165	1,040	Varilla 3/18
6	1,000	AAAAA800101	5,000	1998-07-08	165	1,050	Varilla 3/34
7	1,000	AAAAA800101	5,000	1998-07-08	165	1,060	Varilla 3/19
8	1,000	AAAAA800101	5,000	1998-07-08	165	1,070	Varilla 4/35
9	1,000	AAAAA800101	5,000	1998-07-08	165	1,080	Ladrillos rojos
10	1,000	AAAAA800101	5,000	1998-07-08	165	1,090	Ladrillos grises
11	1,000	AAAAA800101	5,000	1998-07-08	165	1,100	Block
12	1,000	AAAAA800101	5,000	1998-07-08	165	1,110	Meablock

### Construcción de consultas a partir de una especificación

Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

SQL

```
Select m.descripcion from materiales m natural join entregan e where year(fecha) = 2000;
```

¿Por qué aparecen varias veces algunas descripciones de material?

R= Porque hay varias tuplas repetidas con la misma información

descripcion
Varilla 4/32
Varilla 4/33
Varilla 4/34
Varilla 3/19
Varilla 3/19
Ladrillos grises
Block
Cantera rosa
Recubrimiento P1010
Recubrimiento P1019
Recubrimiento P1019
Arena
Tepetate
Tuberla 4.4
Tuberla 3.7
Pintura C1010
Pintura B1020
Pintura B1020
Pintura B1021
Pintura C1012

## Uso del calificador distinct

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

The screenshot shows the MySQL Workbench interface. In the top panel, there is a script editor window titled '<localhost> Script-5' containing the SQL query:Select distinct m.descripcion from materiales m natural join entregan e where year(fecha) = 2000;Below the script editor is a results grid titled 'materiales 1'. The grid has a single column labeled 'abc descripcion' and contains 17 rows, each with a number from 1 to 17 and a corresponding material description. To the right of the grid is a 'Valor' panel showing the value '73'. The bottom of the interface features a status bar with the text 'Enter a SQL expression to filter results (use Ctrl+Space)'.

	abc descripcion
1	Varilla 4/32
2	Varilla 4/33
3	Varilla 4/34
4	Varilla 3/19
5	Ladrillos grises
6	Block
7	Cantera rosa
8	Recubrimiento P1010
9	Recubrimiento P1019
10	Arena
11	Tepetate
12	Tubería 4.4
13	Tubería 3.7
14	Pintura C1010
15	Pintura B1020
16	Pintura B1021
17	Pintura C1012

## 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.

	numero	denominacion	fecha	cantidad
1	5,000	Vamos Mexico	2002-03-12	382
2	5,000	Vamos Mexico	2000-03-05	177
3	5,000	Vamos Mexico	1998-07-08	165
4	5,001	AztecUn	2000-05-03	528
5	5,001	AztecUn	2000-02-05	601
6	5,001	AztecUn	1999-11-05	43
7	5,002	CIT Campeche	2003-02-01	24
8	5,002	CIT Campeche	2001-07-29	582
9	5,002	CIT Campeche	1998-04-07	603
10	5,003	Mexico sin ti no estamos	2003-01-06	530
11	5,003	Mexico sin ti no estamos	1999-09-02	576
12	5,003	Mexico sin ti no estamos	1998-02-21	202
13	5,004	Educando en Coahuila	2003-01-12	152
14	5,004	Educando en Coahuila	2001-08-10	453
15	5,004	Educando en Coahuila	1999-12-11	263
16	5,005	Infonavit Durango	2002-07-08	71
17	5,005	Infonavit Durango	2002-06-10	521

## 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%'
```

Grilla	123 clave	RBC descripción	123 precio
1	1,120	Sillar rosa	100
2	1,130	Sillar gris	110
3			

## ¿Qué resultado obtienes?

Explica que hace el símbolo '%'.

El símbolo % indica que este puede ser sustituido por cualquier cadena de caracteres de cualquier longitud, incluyendo caracteres nulos.

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

Si la consulta fuera LIKE 'Si', solo se buscarían esos 2 caracteres específicamente y no como parte de un patrón, por lo que no recibiríamos un resultado.

¿Qué resultado obtienes?

Explica a qué se debe este comportamiento.

Intenta buscar solamente los que tengan la palabra Si, como no hay ninguno no se muestra nada

## 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\_numérico {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;
```

	Clave	RFC	Numero	Fecha	Cantidad
1	1,000	AAAA800101	5,000	1998-07-08	165
2	1,010	BBBB800101	5,001	2000-05-03	528
3	1,020	CCCC800101	5,002	2001-07-29	582
4	1,030	DDDD800101	5,003	1998-02-21	202
5	1,040	EEEE800101	5,004	1999-12-11	263
6	1,050	FFFF800101	5,005	2000-10-14	503
7	1,060	GGGG800101	5,006	2000-05-04	324
8	1,070	HHHH800101	5,007	1998-02-23	2
9	1,080	AAAA800101	5,008	1999-01-12	86
10	1,090	BBBB800101	5,009	2000-08-01	73
11	1,090	BBBB800101	5,010	1998-01-03	421
12	1,100	CCCC800101	5,009	2000-08-06	466
13	1,100	CCCC800101	5,010	2001-09-10	699
14	1,110	DDDD800101	5,008	1999-05-10	337
15	1,120	EEEE800101	5,007	2001-07-07	692
16	1,130	FFFF800101	5,006	2002-07-06	562
17	1,140	GGGG800101	5,005	2001-09-02	583
18	1,150	HHHH800101	5,004	2001-08-10	453
19	1,200	EEEE800101	5,000	2000-03-05	177

¿Cómo filtrarías rangos de fechas?

Filtraría rangos de fecha usando las instrucciones YEAR(), MONTH() y DAY() junto con BETWEEN.

**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] )
```

entregan 1 × Estadísticas 1

SELECT RFC,Cantidad, Fecha,Numero FROM Entregar | Enter a SQL expression to filter results (use Ctrl+Space)

	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165	1998-07-08	5,000
2	CCCC800101	582	2001-07-29	5,002
3	AAAA800101	86	1999-01-12	5,008
4	CCCC800101	466	2000-08-06	5,009
5	CCCC800101	699	2001-09-10	5,010
6	AAAA800101	152	2003-01-12	5,004
7	CCCC800101	460	1999-05-10	5,006
8	CCCC800101	631	1999-08-09	5,009
9	AAAA800101	382	2002-03-12	5,000
10	AAAA800101	116	1998-06-05	5,010
11	CCCC800101	603	1998-04-07	5,002
12	CCCC800101	278	2000-08-02	5,008

¿Qué hace la consulta?

La consulta recupera el RFC, cantidad, fecha y numero de la tabla entregan cuando el numero está entre 5000 y 5010 y el RFC sea de una razón social que comience con La y continue con cualquier secuencia de caracteres.

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

El paréntesis después de exists encapsula a la subconsulta sobre la que se ejecutará la función para determinar lo que se mostrará en la consulta original.

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**

entregan 1 X Estadísticas 1

SELECT RFC,Cantidad, Fecha,Numero PRO | Enter a SQL expression to filter results (use Ctrl+Space)

	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165	1998-07-08	5,000
2	CCCC800101	582	2001-07-29	5,002
3	AAAA800101	86	1999-01-12	5,008
4	CCCC800101	466	2000-08-06	5,009
5	CCCC800101	699	2001-09-10	5,010
6	AAAA800101	152	2003-01-12	5,004
7	CCCC800101	460	1999-05-10	5,006
8	CCCC800101	631	1999-08-09	5,009
9	AAAA800101	382	2002-03-12	5,000
10	AAAA800101	116	1998-06-05	5,010
11	CCCC800101	603	1998-04-07	5,002
12	CCCC800101	278	2000-08-02	5,008

NOT Simplemente niega la entrada de un valor booleano.

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

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

¿Qué hace la siguiente sentencia? Explica por qué.

Escoje los primeros dos resultados

SELECT TOP 2 \* FROM Proyectos

proyecto 1 X Estadísticas 1

SELECT \* FROM Proyecto LIMIT 2 | Enter a SQL expression t

	numero	denominacion
1	5,000	Vamos Mexico
2	5,001	AztecUn

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

Escoje solo el primer resultado

SELECT TOP Numero FROM Proyectos

	numero	denominacion
1	5,000	Vamos Mexico

### 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.

```
CREATE VIEW vista1(clave, descripcion)  
AS SELECT clave, descripcion  
FROM materiales NATURAL JOIN entregan  
WHERE fecha LIKE '2000%';  
SELECT *
```

FROM vista1

The screenshot shows a SQL Server Management Studio window titled 'vista1 1 X'. The results pane displays a table with two columns: 'clave' and 'descripcion'. The data consists of 20 rows, each containing a number from 1 to 20 followed by a description of a material.

	clave	descripcion
1	1,010	Varilla 4/32
2	1,030	Varilla 4/33
3	1,050	Varilla 4/34
4	1,060	Varilla 3/19
5	1,060	Varilla 3/19
6	1,090	Ladrillos grises
7	1,100	Block
8	1,160	Cantera rosa
9	1,190	Recubrimiento P1010
10	1,200	Recubrimiento P1019
11	1,200	Recubrimiento P1019
12	1,240	Arena
13	1,280	Tepetate
14	1,320	Tubería 4.4
15	1,330	Tubería 3.7
16	1,360	Pintura C1010
17	1,370	Pintura B1020
18	1,370	Pintura B1020
19	1,410	Pintura B1021
20	1,420	Pintura C1012

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 "Méjico sin ti no estamos completos".

The screenshot shows a SQL Server Management Studio window titled 'materiales 1 X'. The results pane displays a table with two columns: 'clave' and 'descripcion'. The data consists of 3 rows, each containing a number from 1 to 3 followed by a description of a material. Above the results, a code editor shows a T-SQL script using joins between 'materiales' and 'entregan' tables.

	clave	descripcion
1	1,030	Varilla 4/33
2	1,230	Cemento
3	1,430	Pintura B1022

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

```

▶ use lab6;
▶+ ┌─ SELECT clave, descripcion
  ┌─ FROM materiales NATURAL JOIN entregan NATURAL JOIN proveedores
  ┌─ WHERE razonsocial = "Acme tools"
  └─
  
```

materiales 1 X Estadísticas 1

SELECT clave, descripcion FROM materiales | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	clave	descripcion

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

```

▶ use lab6;
▶+ ┌─ SELECT rfc
  ┌─ FROM entregan NATURAL JOIN proveedores
  ┌─ WHERE YEAR(fecha) = 2000
  ┌─ GROUP BY rfc
  ┌─ HAVING AVG(cantidad)>=300
  ....
  
```

entregan 1 X Estadísticas 1

SELECT rfc FROM entregan NATURAL JOIN proveedores | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	rfc
1	AAAA800101
2	BBBB800101
3	CCCC800101
4	DDDD800101
5	EEEE800101
6	FFFF800101
7	GGGG800101

El Total entregado por cada material en el año 2000.

```

▶ use lab6;
▶ SELECT descripcion, SUM(cantidad)
  FROM entregan NATURAL JOIN materiales
 WHERE YEAR(fecha) = 2000
 GROUP BY clave
  
```

**materiales 1** × **Estadísticas 1**

SELECT descripcion, SUM(cantidad) FROM entregan NATURAL JOIN materiales WHERE YEAR(fecha) = 2000 Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	descripcion	SUM(cantidad)
1	Varilla 4/32	528
2	Varilla 4/33	295
3	Varilla 4/34	503
4	Varilla 3/19	1,016
5	Ladrillos grises	73
6	Block	466

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

```

▶ use lab6;
▶ SELECT clave
  FROM entregan NATURAL JOIN materiales
 WHERE YEAR(fecha) = 2001
 GROUP BY clave
 HAVING SUM(cantidad)
 ORDER BY cantidad DESC
 LIMIT 1
  
```

**entregan 1** × **Estadísticas 1**

SELECT clave FROM entregan NATURAL JOIN materiales WHERE YEAR(fecha) = 2001 Enter a SQL expression to filter results

Grilla	clave
1	1,100

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

The screenshot shows a MySQL Workbench environment. The top part displays a SQL query:

```
use lab6;  
SELECT *  
from MATERIALES m  
where m.descripcion LIKE "%sub%"
```

The bottom part shows the results of the query in a grid format:

	clave	descripcion	precio
1	1,180	Recubrimiento P1001	200
2	1,190	Recubrimiento P1010	220
3	1,200	Recubrimiento P1019	240
4	1,210	Recubrimiento P1028	250
5	1,220	Recubrimiento P1037	280
6	1,290	Tubería 3.5	200
7	1,300	Tubería 4.3	210
8	1,310	Tubería 3.6	220
9	1,320	Tubería 4.4	230
10	1,330	Tubería 3.7	240
11	1,340	Tubería 4.5	250
12	1,350	Tubería 3.8	260

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

```
▶ use lab6;
▶ SELECT denominacion, SUM(precio*cantidad)
  FROM materiales NATURAL JOIN entregan NATURAL JOIN proyecto
 GROUP BY numero
```

proyecto 1 × Estadísticas 1

SELECT denominacion, SUM(precio\*cantidad) | Enter a SQL expression to filter results (use Ctrl+Space)

	denominacion	SUM(precio*cantidad)
1	Vamos Mexico	106,730
2	AztecUn	146,595
3	CIT Campeche	157,755
4	Mexico sin ti no estamos	260,290
5	Educando en Coahuila	620,610
6	Infonavit Durango	321,135
7	ReconstrucciUn del temp	220,580
8	ConstrucciUn de plaza N	122,969
9	Televisa en acciUn	99,848
10	Disco Atlantic	158,100
11	ConstrucciUn de Hospita	144,295
12	RemodelaciUn de aulas c	225,835

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).

```
▶ use lab6;
▶ ┌─•SELECT denominacion, rfc, razonsocial
  FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyecto
  WHERE denominacion LIKE "Televisa en acci_"
  AND rfc NOT IN (SELECT rfc
  FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyecto
  WHERE denominacion = "Educando en Coahuila")
```

proyecto(+) 1 × Estadísticas 1  
dT SELECT denominacion, rfc, razonsocial FROM Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	rfc denominacion	rfc rfc	rfc razonsocial
1	Televisa en acciÚn	CCCC800101	La Ferre
2	Televisa en acciÚn	DDDD800101	Cecoferre
3	Televisa en acciÚn	DDDD800101	Cecoferre

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).

```
▶ use lab6;
▶ ┌─•SELECT SUM(precio*cantidad), descripcion
  FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyecto NATURAL JOIN materiales
  WHERE denominacion LIKE "Televisa en acci_"
  AND rfc IN (SELECT rfc
  FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyecto
  WHERE denominacion = "Educando en Coahuila")
```

materiales 1 × Estadísticas 1  
dT SELECT SUM(precio\*cantidad), descripcion Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	SUM(precio*cantidad)	rfc descripcion
1	7,938	Tepetate

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.

```
▶ use lab6;
▶ ┌─•SELECT denominacion, descripcion, COUNT(cantidad), precio*cantidad
  FROM entregan NATURAL JOIN proyecto NATURAL JOIN
  materiales
  GROUP BY denominacion, descripcion
```

proyecto(+) 1 × Estadísticas 1  
dT SELECT denominacion, descripcion, COUNT Enter a SQL expression to filter results (use Ctrl+Space)

Grilla	rfc denominacion	rfc descripcion	COUNT(cantidad)	rfc precio*cantidad
1	AmpliaciÚn de la carretera a la huasteca	Cantera rosa	1	230,040
2	AmpliaciÚn de la carretera a la huasteca	Pintura C1010	1	45,500
3	AmpliaciÚn de la carretera a la huasteca	Recubrimiento P1010	1	78,320
4	AmpliaciÚn de la carretera a la huasteca	Tuberla 4.5	1	168,500
5	AmpliaciÚn de la carretera a la huasteca	Varilla 4/33	1	42,775
6	AztecÚn	Pintura B1021	1	75,125
7	AztecÚn	Recubrimiento P1028	1	10,750
8	AztecÚn	Varilla 4/32	1	60,720
9	CIT Campeche	Pintura C1012	1	75,375
10	CIT Campeche	Recubrimiento P1037	1	6,720