

Laboratorio 20

1. Construcción de consultas a partir de una especificación

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

```
4 • SELECT M.descripcion, fecha
5 FROM materiales M, entregan E
6 WHERE M.clave = E.clave
7 AND YEAR(fecha) = 2000;
```

Result Grid		Filter Rows:
	descripcion	fecha
▶	Varilla 3/17	2000-03-29
	Varilla 4/34	2000-04-18
	Block	2000-12-07

✓ 44 11:27:45 SELECT M.descripcion, fecha FROM materiales M, e... 12 row(s) returned

¿Por qué aparecen varias veces algunas descripciones de material?

Porque algunos materiales fueron entregados varias veces en diferentes fechas en el año 2000.

2. Uso del calificador distinct

```
9 • SELECT DISTINCT M.descripcion
10 FROM materiales M, entregan E
11 WHERE M.clave = E.clave
12 AND YEAR(fecha) = 2000;
```

Result Grid			Filter Rows:	<input type="text"/>
	descripcion			
▶	Varilla 3/17			
	Varilla 4/34			
	Block			
✓	46	11:33:38	SELECT DISTINCT M.descripcion FROM materiales ...	10 row(s) returned

¿Qué resultado obtienes en esta ocasión?

Todas las descripciones de materiales entregados en 2000 sin repeticiones.

3. Ordenamientos.

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.

```

16 • SELECT P.numero, denominacion, fecha, cantidad
17 FROM proyectos P, entregan E
18 WHERE P.numero = E.numero
19 ORDER BY P.numero, fecha DESC;

```

Result Grid			Filter Rows:	<input type="text"/>	Export:	Wrap Cell Cont
	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		
✓	58	11:44:54	SELECT P.numero, denominacion, fecha, cantidad F...	87 row(s) returned		

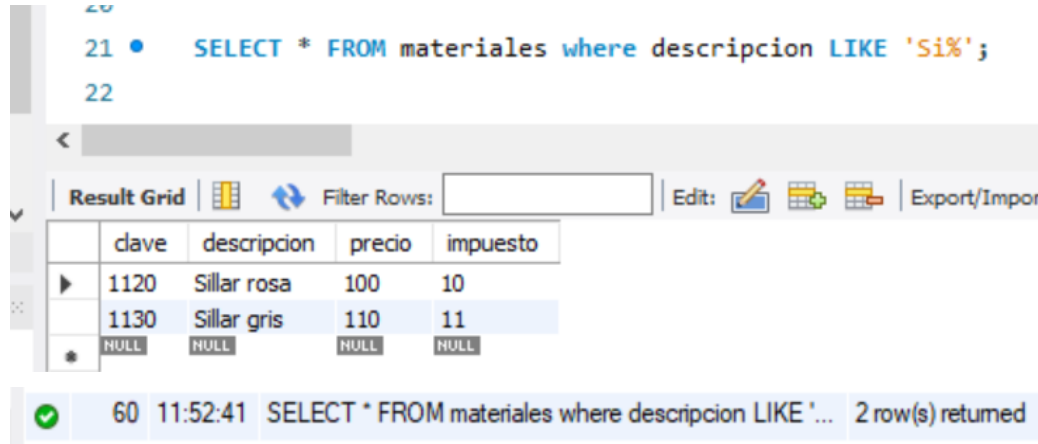
4. Uso de expresiones

- + Suma
- Resta
- * Producto
- / División

5. Operadores de cadena

SELECT * FROM productos where Descripcion LIKE 'Si%'

¿Qué resultado obtienes? Todas las descripciones que empiezan con Si



```
21 • SELECT * FROM materiales where descripcion LIKE 'Si%';
22
```


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

60 11:52:41 SELECT * FROM materiales where descripcion LIKE '... 2 row(s) returned

Explica que hace el símbolo '%': Ignora el valor de caracteres previos o subsecuentes.

¿Qué sucede si la consulta fuera : LIKE 'Si' ? : Da todos los resultados donde la descripción es 'Si'

¿Qué resultado obtienes?



```
21 • SELECT * FROM materiales where descripcion LIKE 'Si';
22
```

	clave	descripcion	precio	impuesto
*	NULL	NULL	NULL	NULL

Explica a qué se debe este comportamiento: No existen descripciones=Si

6. 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?

Result Grid	Filter Rows:	Export:	Wrap Cell
resultado			
¿Que resultado obtienes? ¿¿¿???			

¿Para qué sirve **DECLARE**? Para declarar variables, su tipo y longitud máxima en caso de aplicar. En MySQL, para declarar y asignar variables se usa SET.

¿Cuál es la función de **@foo**?

@foo es una variable que almacena una cadena de texto.

¿Que realiza el operador **SET**?

Asigna un valor a una variable.

[] - 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 carácter 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.

7. Explica el comportamiento, función y resultado de cada una de las siguientes consultas:

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

Devuelve todos los RFC de 'Entregan' donde el valor comienza con una letra entre A y D.

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

Devuelve todas las entradas de la columna RFC donde el valor no comienza con la letra A

SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

Devuelve todos los registros de Número en Entregan donde el valor tiene tres caracteres seguidos por el número 6 en la cuarta posición.

8. Operadores compuestos.

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

Operadores Lógicos.

9. 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?

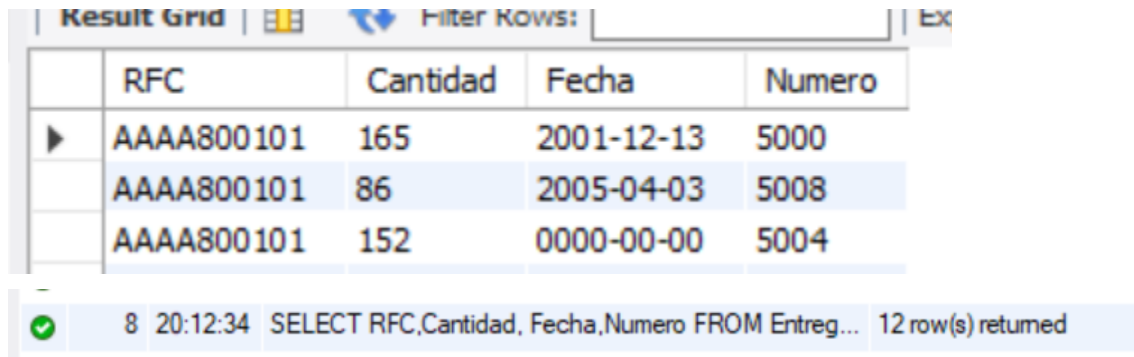
```
AND fecha BETWEEN '2000/01/01' AND '2001/12/12';
```

10. **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?

Muestra el RFC, Cantidad, Fecha y Numero de Entregan, donde los números están entre 5000 y 5010 y para los cuales hay al menos un proveedor cuya razón social comienza con 'La'.



	RFC	Cantidad	Fecha	Numero
▶	AAAA800101	165	2001-12-13	5000
	AAAA800101	86	2005-04-03	5008
	AAAA800101	152	0000-00-00	5004

8 20:12:34 SELECT RFC,Cantidad, Fecha,Numero FROM Entreg... 12 row(s) returned

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


El paréntesis agrupa la subconsulta para claridad sintáctica y lógica.

11. **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

```
37 • SELECT RFC,Cantidad, Fecha,Numero
38 FROM Entregan
39 WHERE numero Between 5000 and 5010
40 AND RFC IN ( SELECT RFC
41 FROM Proveedores
42 WHERE RazonSocial LIKE 'La%');
```



	RFC	Cantidad	Fecha	Numero
▶	AAAA800101	165	2001-12-13	5000
	AAAA800101	86	2005-04-03	5008
	AAAA800101	152	0000-00-00	5004

9 20:13:32 SELECT RFC, Cantidad, Fecha, Numero FROM Entre... 12 row(s) returned

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.

```
44 • SELECT RFC, Cantidad, Fecha, Numero
45 FROM Entregan
46 WHERE numero Between 5000 and 5010
47 AND RFC NOT IN ( SELECT RFC
48 FROM Proveedores
49 WHERE razonsocial NOT LIKE 'La%');
```

	RFC	Cantidad	Fecha	Numero
▶	AAAA800101	165	2001-12-13	5000
	CCCC800101	582	2003-12-16	5002
	AAAA800101	86	2005-04-03	5008

✓ 10 20:14:41 SELECT RFC, Cantidad, Fecha, Numero FROM Entre... 12 row(s) returned

12. 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

Selecciona las primeras dos filas de la tabla Proyectos. La cláusula TOP limita el número de filas que se devuelven en un resultado. En este caso, se seleccionan las dos primeras filas de la tabla Proyectos.

En MySQL, se utiliza LIMIT.

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

SELECT TOP Numero FROM Proyectos

No es válida porque TOP se utiliza para especificar un número fijo de filas a seleccionar, no puede aceptar una columna como argumento.

13. 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);
```

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

Para que a cada material se le asigne 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
	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

¿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, m.descripcion, m.Precio, e.Cantidad, e.Fecha,  
       SUM(e.Cantidad * (m.Precio * (1 + m.PorcentajeImpuesto / 100))) AS ImporteTotal  
FROM Entregan e, Materiales m  
WHERE e.Clave = m.Clave  
GROUP BY e.Clave, e.RFC, e.Numero, e.Fecha;
```

	Clave	descripcion	Precio	Cantidad	Fecha	ImporteTotal
▶	1000	Varilla 3/16	100	165	2001-12-13	16830
	1000	Varilla 3/16	100	254	1999-07-13	25908
	1010	Varilla 4/32	115	528	1998-07-28	61946.543999999994

14. Creación de vistas

Create view nombrevista (nombrecolumna1 , nombrecolumna2 ,..., nombrecolumna3)

as select... 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.

*Las vistas no pueden incluir la cláusula order by.

Crea vistas para cinco de las consultas que planteaste anteriormente en la práctica. Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

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

```
109 • SELECT RFC
110 FROM Entregan
111 WHERE YEAR(Fecha) = 2000
112 GROUP BY RFC
113 HAVING AVG(Cantidad) >= 300;
114
```

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

	RFC	AVG(Cantidad)
▶	BBBB800101	601.0000
	FFFF800101	322.7500
	GGGG800101	583.0000

97 13:56:32 SELECT RFC, AVG(Cantidad) FROM Entregan WHE... 3 row(s) returned

4. El Total entregado por cada material en el año 2000.

```
115 • SELECT M.clave, descripcion, SUM(cantidad) as Total_entregado
116 FROM materiales M, Entregan E
117 WHERE M.clave=E.clave
118 AND YEAR(Fecha) = 2000
119 GROUP BY clave;
```

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

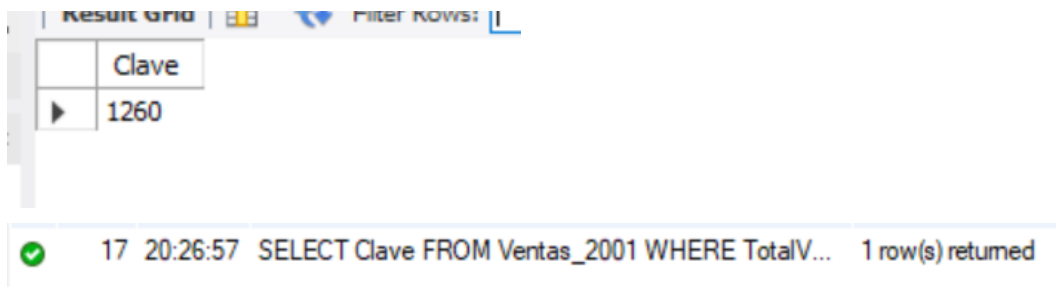
	clave	descripcion	Total_entregado
▶	1020	Varilla 3/17	8
	1050	Varilla 4/34	623
	1100	Block	466
	1130	Sillar gris	625

98 14:00:06 SELECT M.clave, descripcion, SUM(cantidad) as Tot... 11 row(s) returned

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

```
CREATE VIEW Ventas_2001 AS  
SELECT Clave, SUM(Cantidad) AS TotalVentas  
FROM Entregan  
WHERE YEAR(Fecha) = 2001  
GROUP BY Clave;
```

```
SELECT Clave  
FROM Ventas_2001  
WHERE TotalVentas = (SELECT MAX(TotalVentas) FROM Ventas_2001);
```

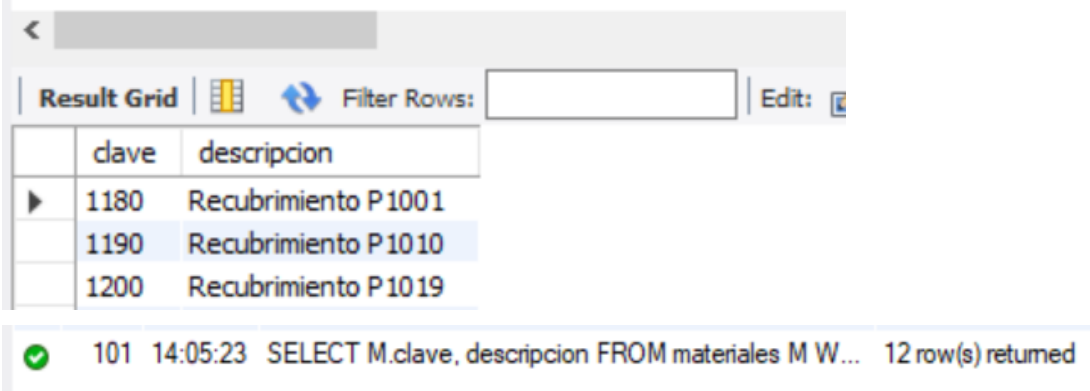


The screenshot shows a 'Result Grid' with one column labeled 'Clave' and one row containing the value '1260'. Below the grid, a status bar indicates '17 20:26:57 SELECT Clave FROM Ventas_2001 WHERE TotalV... 1 row(s) returned'.

Clave
1260

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

```
131 • SELECT M.clave, descripcion  
132 FROM materiales M  
133 WHERE descripcion LIKE '%ub%';
```

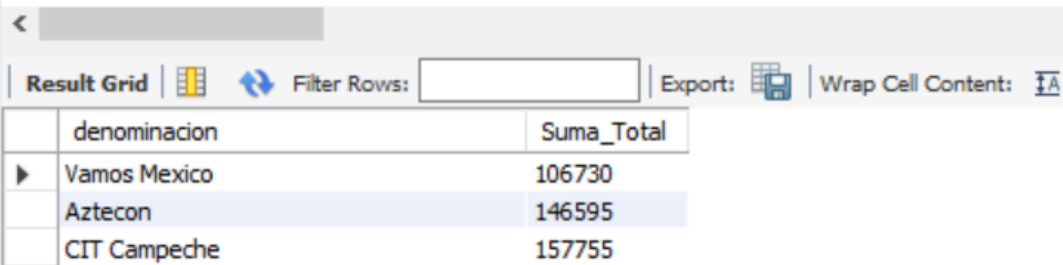


The screenshot shows a 'Result Grid' with two columns: 'clave' and 'descripcion'. It contains three rows of data. Below the grid, a status bar indicates '101 14:05:23 SELECT M.clave, descripcion FROM materiales M W... 12 row(s) returned'.

clave	descripcion
1180	Recubrimiento P1001
1190	Recubrimiento P1010
1200	Recubrimiento P1019

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

```
135 • SELECT denominacion, SUM(cantidad*precio) as Suma_Total
136 FROM proyectos P, entregan E, materiales M
137 WHERE E.clave=M.clave
138 AND E.numero=P.numero
139 GROUP BY P.numero;
```



The screenshot shows a database interface with a query result grid. The grid has two columns: 'denominacion' and 'Suma_Total'. There are three rows of data: 'Vamos Mexico' with a sum of 106730, 'Aztecon' with a sum of 146595, and 'CIT Campeche' with a sum of 157755. Above the grid, there are controls for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'.

denominacion	Suma_Total
Vamos Mexico	106730
Aztecon	146595
CIT Campeche	157755

✓ 103 14:12:15 SELECT denominacion, SUM(cantidad*precio) as Su... 20 row(s) returned

8. 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).

```
CREATE VIEW Proveedores_EducandoenCoahuila AS
SELECT denominacion, E.RFC, razonsocial
FROM proveedores P, entregan E, proyectos Pr
WHERE P.rfc=E.rfc AND Pr.numero=E.numero
AND denominacion='Educando en Coahuila';
```

```
SELECT * FROM Proveedores_EducandoenCoahuila;
```

```
CREATE VIEW Proveedores_TelevisaenAcción AS
SELECT denominacion, E.RFC, razonsocial
FROM proveedores P, entregan E, proyectos Pr
WHERE P.rfc=E.rfc AND Pr.numero=E.numero
AND denominacion='Televisa en acción';
```

```
SELECT * FROM Proveedores_TelevisaenAcción;
```

```
SELECT * FROM Proveedores_TelevisaenAcción
WHERE (rfc) NOT IN (
    SELECT rfc
    FROM Proveedores_EducandoenCoahuila
);
```

Result Grid			
	denominacion	RFC	razonsocial
▶	Televisa en acción	CCCC800101	La Ferre
	Televisa en acción	DDDD800101	Cecoferre
	Televisa en acción	DDDD800101	Cecoferre

✓ 27 11:25:49 SELECT * FROM Proveedores_TelevisaenAcción W... 3 row(s) returned

9. 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).

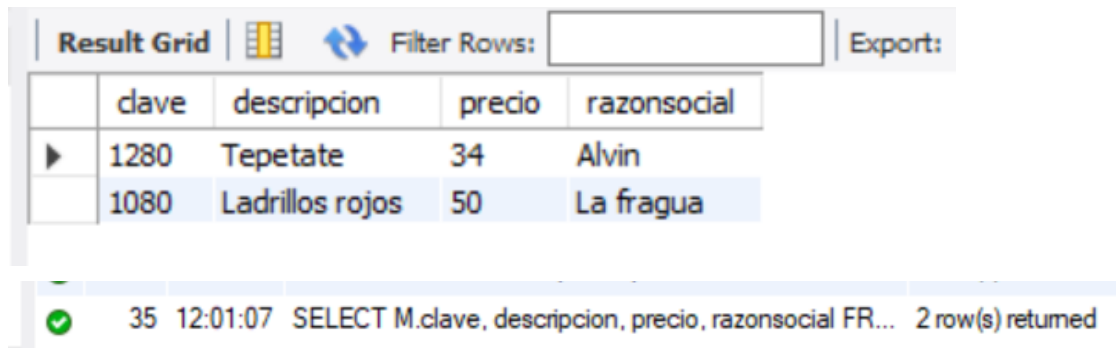
```
SELECT denominacion, E.RFC, razonsocial
FROM proveedores P, entregan E, proyectos Pr
WHERE P.rfc=E.rfc AND Pr.numero=E.numero
AND denominacion='Televisa en acción'
AND (E.rfc) NOT IN (
    SELECT E.RFC
    FROM proveedores P, entregan E, proyectos Pr
    WHERE P.rfc=E.rfc AND Pr.numero=E.numero
    AND denominacion='Educando en Coahuila'
);
```

Result Grid			
	denominacion	RFC	razonsocial
▶	Televisa en acción	CCCC800101	La Ferre
	Televisa en acción	DDDD800101	Cecoferre
	Televisa en acción	DDDD800101	Cecoferre

✓ 30 11:52:54 SELECT denominacion, E.RFC, razonsocial FROM pr... 3 row(s) returned

10. 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.

```
SELECT M.clave, descripcion, precio, razonsocial
FROM materiales M, entregan E, proyectos P, proveedores Pro
WHERE m.clave=E.clave AND P.numero=E.numero AND Pro.rfc=E.rfc
AND denominacion='Televisa en acción'
AND razonsocial IN(
    SELECT razonsocial
    FROM materiales M, entregan E, proyectos P, proveedores Pro
    WHERE m.clave=E.clave AND P.numero=E.numero AND Pro.rfc=E.rfc
    AND denominacion='Educando en Coahuila'
);
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has columns for 'clave', 'descripcion', 'precio', and 'razonsocial'. Two rows are visible: one for 'Tepetate' with a price of 34 and provider 'Alvin', and another for 'Ladrillos rojos' with a price of 50 and provider 'La fragua'. Below the grid, a status bar indicates that 2 rows were returned.

	clave	descripcion	precio	razonsocial
▶	1280	Tepetate	34	Alvin
	1080	Ladrillos rojos	50	La fragua

35 12:01:07 SELECT M.clave, descripcion, precio, razonsocial FR... 2 row(s) returned

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

```
SELECT descripcion AS Nombre_Material,
    COUNT(*) AS Veces_Entregado,
    SUM(cantidad * precio) AS Total_Costo
FROM materiales M, entregan E, proyectos P
WHERE M.clave=E.clave AND P.numero=E.numero
GROUP BY descripcion
ORDER BY Nombre_Material;
```

Result Grid | Filter Rows: | Export: |

	Nombre_Material	Veces_Entregado	Total_Costo
	Arena	2	103600
	Block	2	34950
	Cantera amarilla	2	53590
	Cantera blanca	2	160400



65 12:22:03 SELECT descripcion AS Nombre_Material, COU... 42 row(s) returned

SCRIPT SQL

```
USE lab20materiales;  
SET lc_time_names = 'es_ES';
```

```
SELECT M.descripcion, fecha  
FROM materiales M, entregan E  
WHERE M.clave = E.clave  
AND YEAR(fecha) = 2000;
```

```
SELECT DISTINCT M.descripcion  
FROM materiales M, entregan E  
WHERE M.clave = E.clave  
AND YEAR(fecha) = 2000;
```

```
SET lc_time_names = 'es_ES';
```

```
SELECT P.numero, denominacion, fecha, cantidad  
FROM proyectos P, entregan E  
WHERE P.numero = E.numero  
ORDER BY P.numero, fecha DESC;
```

```
SELECT * FROM materiales where descripcion LIKE 'Si';
```

```
SET @foo := '¿Que resultado';  
SET @bar := '¿¿¿???';  
SET @foo := CONCAT(@foo, ' obtienes?');  
SELECT CONCAT(@foo, @bar) AS resultado;
```

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;'  
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%;'  
SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
```

```
SELECT M.descripcion, fecha  
FROM materiales M, entregan E  
WHERE M.clave = E.clave  
AND fecha BETWEEN '2000/01/01' AND '2001/12/12';
```

```
SELECT RFC, Cantidad, Fecha, Numero  
FROM Entregan
```



```
WHERE numero Between 5000 and 5010
AND RFC IN ( SELECT RFC
FROM Proveedores
WHERE RazonSocial LIKE 'La%');
```

```
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE numero Between 5000 and 5010
AND RFC NOT IN ( SELECT RFC
FROM Proveedores
WHERE razonsocial NOT LIKE 'La%');
```

```
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 );
```

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

```
UPDATE materiales SET PorcentajImpuesto = 2*clave/1000;
```

```
SELECT e.Clave, m.descripcion, m.Precio, e.Cantidad, e.Fecha,
      SUM(e.Cantidad * (m.Precio * (1 + m.PorcentajImpuesto / 100))) AS ImporteTotal
FROM Entregan e, Materiales m
WHERE e.Clave = m.Clave
GROUP BY e.Clave, e.RFC, e.Numero, e.Fecha;
```

---->

```
CREATE VIEW Materiales_Entregados_2000 AS
SELECT DISTINCT descripcion, fecha
FROM entregan E, materiales M
WHERE E.clave = M.clave
AND YEAR(fecha) = 2000;
```

```
SELECT * FROM Materiales_Entregados_2000;
```

```
CREATE VIEW Materiales_Descripcion_Si AS
SELECT * FROM materiales WHERE Descripcion LIKE 'Si%';
```

```
SELECT * FROM Materiales_Descripcion_Si;
```

---->

```
SELECT e.clave, descripcion
FROM materiales M, entregan E, proyectos P
WHERE P.numero=E.numero
AND M.clave=E.clave
AND denominacion="México sin ti no estamos completos";
```

```
SELECT e.clave, descripcion
FROM materiales M, entregan E, proveedores P
WHERE P.rfc=E.rfc
AND M.clave=E.clave
AND razonsocial="Acme tools";
```

```
SELECT e.clave, descripcion, razonsocial
FROM materiales M, entregan E, proveedores P
WHERE P.rfc=E.rfc
AND M.clave=E.clave
AND razonsocial="Acme tools";
```

```
SELECT RFC
FROM Entregan
WHERE YEAR(Fecha) = 2000
GROUP BY RFC
HAVING AVG(Cantidad) >= 300;
```

```
SELECT M.clave, descripcion, SUM(cantidad) as Total_entregado
FROM materiales M, Entregan E
WHERE M.clave=E.clave
AND YEAR(Fecha) = 2000
GROUP BY clave;
```

```
CREATE VIEW Ventas_2001 AS
SELECT Clave, SUM(Cantidad) AS TotalVentas
FROM Entregan
WHERE YEAR(Fecha) = 2001
GROUP BY Clave;
```

```
SELECT Clave
FROM Ventas_2001
```

```
WHERE TotalVentas = (SELECT MAX(TotalVentas) FROM Ventas_2001);
```

```
SELECT M.clave, descripcion  
FROM materiales M  
WHERE descripcion LIKE '%ub%';
```

```
SELECT denominacion, SUM(cantidad*precio) as Suma_Total  
FROM proyectos P, entregan E, materiales M  
WHERE E.clave=M.clave  
AND E.numero=P.numero  
GROUP BY P.numero;
```

8.

```
CREATE VIEW Proveedores_EducandoenCoahuila AS  
SELECT denominacion, E.RFC, razonsocial  
FROM proveedores P, entregan E, proyectos Pr  
WHERE P.rfc=E.rfc AND Pr.numero=E.numero  
AND denominacion='Educando en Coahuila';
```

```
SELECT * FROM Proveedores_EducandoenCoahuila;
```

```
CREATE VIEW Proveedores_TelevisaenAcción AS  
SELECT denominacion, E.RFC, razonsocial  
FROM proveedores P, entregan E, proyectos Pr  
WHERE P.rfc=E.rfc AND Pr.numero=E.numero  
AND denominacion='Televisa en acción';
```

```
SELECT * FROM Proveedores_TelevisaenAcción;
```

```
SELECT * FROM Proveedores_TelevisaenAcción  
WHERE (rfc) NOT IN (  
    SELECT rfc  
    FROM Proveedores_EducandoenCoahuila  
);
```

9.

```
SELECT denominacion, E.RFC, razonsocial  
FROM proveedores P, entregan E, proyectos Pr  
WHERE P.rfc=E.rfc AND Pr.numero=E.numero  
AND denominacion='Televisa en acción'  
AND (E.rfc) NOT IN (  
    SELECT E.RFC  
    FROM proveedores P, entregan E, proyectos Pr
```

```
WHERE P.rfc=E.rfc AND Pr.numero=E.numero  
AND denominacion='Educando en Coahuila'  
);
```

10.

```
SELECT M.clave, descripcion, precio, razonsocial  
FROM materiales M, entregan E, proyectos P, proveedores Pro  
WHERE m.clave=E.clave AND P.numero=E.numero AND Pro.rfc=E.rfc  
AND denominacion='Televisa en acción'  
AND razonsocial IN(  
    SELECT razonsocial  
    FROM materiales M, entregan E, proyectos P, proveedores Pro  
    WHERE m.clave=E.clave AND P.numero=E.numero AND Pro.rfc=E.rfc  
    AND denominacion='Educando en Coahuila'  
);
```

11.

```
SELECT descripcion AS Nombre_Material,  
    COUNT(*) AS Veces_Entregado,  
    SUM(cantidad * precio) AS Total_Costo  
FROM materiales M, entregan E, proyectos P  
WHERE M.clave=E.clave AND P.numero=E.numero  
GROUP BY descripcion  
ORDER BY Nombre_Material;
```