

LAB20 OUTPUTS

```
/* Manuel Villalpando y Dante Péres */

-- Materiales(Clave, Descripción, Costo)
-- Proveedores(RFC, RazonSocial)
-- Proyectos(Numero,Denominacion)
-- Entregan(Clave, RFC, Numero, Fecha, Cantidad)

/* Consulta de un tabla completa

Algebra relacional.
materiales

SQL */
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

45 Lineas */

/* Selección

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

SQL */
```

LAB20 OUTPUTS

```
SELECT * FROM materiales
WHERE clave=1000

/*clave
descripcion
precio
impuesto

1000
Varilla 3/16
100
10

1 Linea */

/* Proyección

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

SQL*/
SELECT clave,rfc,fecha FROM entregan

/* clave
rfc
fecha

1000
AAAA800101
2001-12-13

1000
AAAA800101
1999-07-13

1010
BBBB800101
1998-07-2

87 Lineas*/

/*
Reunión Natural
```

LAB20 OUTPUTS

```
Algebra relacional.  
entregan JN materiales
```

```
SQL*/  
SELECT * FROM materiales,entregan  
WHERE materiales.clave = entregan.clave
```

```
/* clave  
descripcion  
precio  
impuesto  
clave  
rfc  
numero  
fecha  
cantidad  
1000  
Varilla 3/16  
100  
10  
1000  
AAAA800101  
5000  
2001-12-13  
165  
1000  
Varilla 3/16  
100  
10  
1000  
AAAA800101  
5019  
1999-07-13  
254
```

```
87 Lineas
```

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

No, ya que en esta consulta solamente se recuperan los registros que tienen una clave en común en ambas tablas,

LAB20 OUTPUTS

por lo que si no hay una clave en común, no se recuperará ningún registro de la tabla materiales.

*/

/*

Reunión con criterio específico

Algebra relacional.

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

SQL*/

SELECT * FROM entregan,proyectos

WHERE entregan.numero <= proyectos.numero

/* clave

rfc

numero

fecha

cantidad

numero

denominacion

1000

AAAA800101

5000

2001-12-13

165

5000

Vamos Mexico

1200

EEEE800101

5000

2003-03-15

177

5000

Vamos Mexico

1400

AAAA800101

5000

1999-04-07

382

5000

Vamos Mexico

LAB20 OUTPUTS

```
87 Lineas */
```

```
/*
```

```
Unión (se ilustra junto con selección)
```

```
Algebra relacional.
```

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

```
SQL */
```

```
(SELECT * FROM entregan WHERE clave=1450)
```

```
UNION
```

```
(SELECT * FROM entregan WHERE clave=1300)
```

```
/*
```

```
clave
```

```
rfc
```

```
numero
```

```
fecha
```

```
cantidad
```

```
1300
```

```
GGGG800101
```

```
5005
```

```
2004-02-28
```

```
521
```

```
1300
```

```
GGGG800101
```

```
5010
```

```
2001-02-10
```

```
119
```

```
2 Lineas
```

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

```
SELECT * FROM entregan WHERE clave IN (1450, 1300)
```

```
*/
```

```
/*
```

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

```
Algebra relacional.
```

LAB20 OUTPUTS

```
PR{clave}(SL{numero=5001}(entregan)) IN
```

```
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)
```

```
/* clave
```

```
1010
```

```
1 Linea */
```

```
/*
```

```
Diferencia (se ilustra con selección )
```

Algebra relacional.

```
entregan - SL{clave=1000}(entregan)
```

```
SQL*/
```

```
(SELECT * FROM entregan)
```

```
minus
```

```
(SELECT * FROM entregan where clave=1000)
```

```
/*
```

```
clave
```

```
rfc
```

```
numero
```

```
fecha
```

```
cantidad
```

```
1010
```

```
BBBB800101
```

```
5001
```

```
1998-07-28
```

```
528
```

LAB20 OUTPUTS

```
1010
BBBB800101
5018
1997-02-09
523
```

```
1020
CCCC800101
5002
2003-12-16
582
```

```
85 Lineas
```

Nuevamente, "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.

```
SELECT * FROM entregan WHERE clave <> 1000
*/
```

```
/*
```

```
Producto cartesiano
```

```
Algebra relacional.
entregan X materiales
```

```
SQL */
SELECT *
FROM entregan, materiales;
/* clave
```

```
rfc
numero
fecha
cantidad
clave
descripcion
precio
impuesto
1000
AAAA800101
5000
2001-12-13
```

LAB20 OUTPUTS

```
165
1000
Varilla 3/16
100
10
1000
AAAA800101
5000
2001-12-13
165
1010
Varilla 4/32
115
11.5
1000
AAAA800101
5000
2001-12-13
165
1020
Varilla 3/17
130
13
```

ESTE QUERRY ESTA MAL, YA QUE NO DESPLIEGA CORRECTAMENTE LAS TABLAS, INCLUSO NO SE MUESTRA EL NUMERO DE TABLAS DESPLEGADAS EN LA CONSULTA, PERO SI SE MUESTRAN LOS DATOS DE LAS TABLAS.

EN DADO CASO DE QUE QUIERA CORREGIR ESTE ERROR Y DESPLEGAR LOS ELEMENTOS QUE SE CRUZAN CON LA MISMA CLAVE, SE TENDRIA QUE UTILIZAR EL SIGUIENTE QUERRY:

```
*/
```

```
SELECT *
FROM entregan
JOIN materiales
ON entregan.clave = materiales.clave;
/*
clave
rfc
numero
```


LAB20 OUTPUTS

```
fecha
cantidad
clave
descripcion
precio
impuesto
1000
AAAA800101
5000
2001-12-13
165
1000
Varilla 3/16
100
10
1000
AAAA800101
5019
1999-07-13
254
1000
Varilla 3/16
100
10
1010
BBBB800101
5001
1998-07-28
528
1010
Varilla 4/32
115
11.5

87 Lineas

¿Cómo está definido el número de tuplas de este resultado en términos
del número de tuplas de entregan y de materiales?
El número de tuplas de este resultado es igual al número de tuplas de
entregan multiplicado por el número de tuplas de materiales.

*/

/*
```

LAB20 OUTPUTS

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.

*/

```
SELECT descripcion, fecha
FROM entregan, materiales
WHERE entregan.clave = materiales.clave
AND YEAR(STR_TO_DATE(fecha, '%Y-%m-%d')) = 2000;
```

/*

Varilla 3/17

2000-03-29

Varilla 4/34

2000-04-18

Block

2000-12-07

12 Lineas

¿Por qué aparecen varias veces algunas descripciones de material?

Porque no se ha puesto la cláusula DISTINCT, por lo que se muestran todas las tuplas que cumplen con la condición. O en su defecto la cláusula GROUP BY, que agrupa las tuplas que cumplen con la condición.

*/

/*

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),

LAB20 OUTPUTS

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.

```
*/
```

```
    SELECT proyectos.numero, proyectos.denominacion, entregan.fecha,  
entregan.cantidad
```

```
    FROM proyectos
```

```
    INNER JOIN entregan ON proyectos.numero = entregan.numero
```

```
    ORDER BY proyectos.numero, entregan.fecha DESC;
```

```
/*
```

```
5000
```

```
Vamos Mexico
```

```
2003-03-15
```

```
177
```

```
5000
```

```
Vamos Mexico
```

```
2001-12-13
```

```
165
```

```
87 Lineas
```

```
*/
```

```
/* Operadores de cadena */
```

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

```
/* clave
```

```
descripcion
```

```
precio
```

```
impuesto
```

```
1120
```

```
Sillar rosa
```

```
100
```

```
10
```

```
1130
```

```
Sillar gris
```

```
110
```

LAB20 OUTPUTS

11

2 Lineas

Explica que hace el símbolo '%'.
El símbolo '%' indica que puede haber cualquier número de caracteres antes de la cadena que se está buscando.

¿Qué sucede si la consulta fuera : LIKE 'Si' ?
No se mostrarían los resultados, ya que no se cumpliría la condición.

¿Qué resultado obtienes?
Ninguno.

Explica a qué se debe este comportamiento.
Porque no se cumpliría la condición debido a que no se le esta pidiendo que busque una cadena que contenga la cadena 'Si', sino que busque una cadena que sea igual a 'Si'.

*/

/*

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

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene letras de la A a la D
Funcion: Esto nos sirve para filtrar por un rango de caracteres especificos
Resultado: Por si solo retorna cero columnas, esto se puede deber a que el operador Like es sensible a mayusculas y minusculas o que no sea compatible con la version de SQL que se esta utilizando.

*/

SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
/* Comportamiento: Toma la tabla rfc de 'entregan' y obtienen valores de cuando rfc contiene al final el numero 6

*/

LAB20 OUTPUTS

```
Funcion: Esto nos sirve para filtrar por un rango de caracteres
especificos
Resultado:
Numero
5016
5006
5006

9 Lineas
*/

/*
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? */
SELECT Clave,RFC,Numero,Fecha,Cantidad
FROM Entregan
WHERE Fecha BETWEEN '2000-01-01' and '2000-12-31';

/* 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?
La consulta muestra los RFC, Cantidad, Fecha y Numero de la tabla
Entregan, donde el numero de proyecto se encuentra entre 5000 y 5010 y
el RFC de la tabla Entregan se encuentra en la tabla Proveedores y la
razon social de la tabla Proveedores empieza con la letra 'La'.
¿Qué función tiene el paréntesis ( ) después de EXISTS?
El paréntesis es para indicar que la subconsulta se encuentra dentro de
la consulta principal.
```

LAB20 OUTPUTS

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

```
*/  
SELECT RFC, Cantidad, Fecha, Numero  
FROM Entregan  
WHERE Numero BETWEEN 5000 AND 5010  
AND RFC IN (  
    SELECT RFC  
    FROM Proveedores  
    WHERE RazonSocial LIKE 'La%'  
)
```

/* Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN

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. */

```
SELECT RFC, Cantidad, Fecha, Numero  
FROM Entregan  
WHERE Numero BETWEEN 5000 AND 5010  
AND RFC = ANY (  
    SELECT RFC  
    FROM Proveedores  
    WHERE RazonSocial LIKE 'La%'  
)
```

/* ¿Qué hace la siguiente sentencia? Explica por qué.

Despliega las dos primeras filas de la tabla de proyectos porque se le esta indicando con top que nada más

despliegue las primeras 2 o sea las del top 2*/

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

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

No es valida su sintaxis porque requiere que se le especifique la cantidad de tablas a desplegar*/

LAB20 OUTPUTS

```
SELECT TOP Numero FROM Proyectos
```

```
/* Modificando la estructura de un 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.

¿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.Numero, m.Descripcion, e.Cantidad, m.Precio,
m.PorcentajeImpuesto,
      e.Cantidad * m.Precio * (1 + m.PorcentajeImpuesto / 100) AS
Importe
FROM entregan e
JOIN materiales m ON e.clave = m.clave;
```

```
/* HACER 5 VIEWS */
```

```
CREATE VIEW Materiales1000 AS
```

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

```
CREATE VIEW ProveedoresEntregan AS
```

```
SELECT * FROM Proveedores, Entregan
WHERE Proveedores.RFC = Entregan.RFC;
```

```
CREATE VIEW MatsEntregados AS
```

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

```
CREATE VIEW ProveedoresEntreganMats AS
```

LAB20 OUTPUTS

```
SELECT * FROM ProveedoresEntregan, MatsEntregados
WHERE ProveedoresEntregan.Numero = MatsEntregados.Numero;

CREATE VIEW ProveedoresEntreganMatsProyectos AS
SELECT * FROM ProveedoresEntreganMats, Proyectos
WHERE ProveedoresEntreganMats.Numero = Proyectos.Numero;

-- Los materiales (clave y descripción) entregados al proyecto "México
sin ti no estamos completos".
SELECT m.Clave, m.Descripcion
FROM Materiales m
INNER JOIN Entregan e ON m.Clave = e.Clave
INNER JOIN Proyectos p ON e.Numero = p.Numero
WHERE p.Denominacion = 'México sin ti no estamos completos';

-- Los materiales (clave y descripción) que han sido proporcionados por
el proveedor "Acme tools".
SELECT m.Clave, m.Descripcion
FROM Materiales m
INNER JOIN Entregan e ON m.Clave = e.Clave
INNER JOIN Proveedores p ON e.RFC = p.RFC
WHERE p.RazonSocial = 'Acme tools';

-- El RFC de los proveedores que durante el 2000 entregaron en promedio
cuando menos 300 materiales.
SELECT e.RFC
FROM Entregan e
INNER JOIN Proveedores p ON e.RFC = p.RFC
WHERE YEAR(e.Fecha) = 2000
GROUP BY e.RFC
HAVING AVG(e.Cantidad) >= 300;

-- El Total entregado por cada material en el año 2000.
SELECT e.Clave, SUM(e.Cantidad * (m.Precio * (1 + m.PorcentajeImpuesto
/ 100))) AS TotalEntregado
FROM Entregan e
INNER JOIN Materiales m ON e.Clave = m.Clave
WHERE YEAR(e.Fecha) = 2000
GROUP BY e.Clave;

-- La Clave del material más vendido durante el 2001. (se recomienda
usar una vista intermedia para su solución)
CREATE VIEW MaterialMasVendido AS
```


LAB20 OUTPUTS

```
SELECT e.Clave, SUM(e.Cantidad) AS TotalCantidad
FROM Entregan e
WHERE YEAR(e.Fecha) = 2001
GROUP BY e.Clave;

SELECT m.Clave, m.Descripcion
FROM Materiales m
INNER JOIN MaterialMasVendido mv ON m.Clave = mv.Clave
WHERE mv.TotalCantidad = (SELECT MAX(TotalCantidad) FROM
MaterialMasVendido);

-- Productos que contienen el patrón 'ub' en su nombre.
SELECT Clave, Descripcion
FROM Materiales
WHERE Descripcion LIKE '%ub%';

-- Denominación y suma del total a pagar para todos los proyectos.
SELECT p.Denominacion, SUM(e.Cantidad * (m.Precio * (1 +
m.PorcentajeImpuesto / 100))) AS TotalPagar
FROM Entregan e
INNER JOIN Materiales m ON e.Clave = m.Clave
INNER JOIN Proyectos p ON e.Numero = p.Numero
GROUP BY p.Denominacion;

-- 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 MaterialesTelevisa AS
SELECT DISTINCT e.RFC, m.Clave
FROM Entregan e
INNER JOIN Materiales m ON e.Clave = m.Clave
INNER JOIN Proyectos p ON e.Numero = p.Numero
WHERE p.Denominacion = 'Televisa en acción';

CREATE VIEW MaterialesCoahuila AS
SELECT DISTINCT e.RFC

-- La cantidad total entregada por proveedor para el proyecto "Televisa
en acción".
SELECT Proveedores.RFC, Proveedores.RazonSocial, SUM(Entregan.Cantidad)
as CantidadTotal
```

LAB20 OUTPUTS

```
FROM Proveedores, Entregan
WHERE Proveedores.RFC = Entregan.RFC
AND Entregan.Numero = (SELECT Numero FROM Proyectos WHERE Denominacion
= 'Televisa en acción')
GROUP BY Proveedores.RFC, Proveedores.RazonSocial;

-- Nombre del material, cantidad de veces entregados y total del costo
de dichas entregas por material de todos los proyectos.
SELECT m.Descripcion AS Material, COUNT(*) AS CantidadEntregas,
SUM(e.Cantidad * m.Precio + e.Cantidad * m.Precio *
m.PorcentajeImpuesto / 100) AS TotalCosto
FROM materiales m
INNER JOIN entregan e ON m.Clave = e.Clave
GROUP BY m.Descripcion
ORDER BY TotalCosto DESC;
```

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

-- Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos". [SELECT](#) m.Clave, m.Descripción FROM Materiales m INNER JOIN Entregan e ON m.Clave = e.Clave INNER JOIN Proyectos p ON e.Numero = p.Numero WHERE p.Denominacion = 'México sin ti no estamos completos';

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

Clave	Descripción
1030	Varilla 4/33
1230	Cemento
1430	Pintura B1022

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

-- Los materiales (clave y descripción) que han sido proporcionados por el proveedor "Acme tools". [SELECT](#) m.Clave, m.Descripción FROM Materiales m INNER JOIN Entregan e ON m.Clave = e.Clave INNER JOIN Proveedores p ON e.RFC = p.RFC WHERE p.RazonSocial = 'Acme tools';

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Clave Descripción

Query results operations

Create view

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

-- El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales. [SELECT](#) e.RFC FROM Entregan e INNER JOIN Proveedores p ON e.RFC = p.RFC WHERE YEAR(e.Fecha) = 2000 GROUP BY e.RFC HAVING AVG(e.Cantidad) >= 300;

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

RFC
BBBB800101
FFFF800101
GGGG800101

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

LAB20 OUTPUTS

✓ Showing rows 0 - 10 (11 total, Query took 0.0017 seconds.)

```
-- El Total entregado por cada material en el año 2000. SELECT e.Clave, SUM(e.Cantidad * (m.Precio * (1 + m.PorcentajeImpuesto / 100))) AS TotalEntregado FROM Entregan e INNER JOIN Materiales m ON e.Clave = m.Clave WHERE YEAR(e.Fecha) = 2000 GROUP BY e.Clave;
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Clave	TotalEntregado
1020	1061.216
1050	111314.525
1100	14287.56
1130	70303.75
1140	119258.47999999998
1210	11010.15
1310	16255.008000000002
1360	34025.99999999999
1390	13746.824999999999
1410	77243.525000000001
1430	1671.475

☐ Show all | Number of rows: 25 | Filter rows: Search this table

✓ Showing rows 0 - 0 (1 total, Query took 0.0029 seconds.)

```
SELECT m.Clave, m.Descripcion FROM Materiales m INNER JOIN MaterialMasVendido mv ON m.Clave = mv.Clave WHERE mv.TotalCantidad = (SELECT MAX(TotalCantidad) FROM MaterialMasVendido);
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Clave	Descripcion
1260	Gravilla

☐ Show all | Number of rows: 25 | Filter rows: Search this table

LAB20 OUTPUTS

✓ Showing rows 0 - 11 (12 total, Query took 0.0003 seconds.)

```
-- Productos que contienen el patrón 'ub' en su nombre. SELECT Clave, Descripcion FROM Materiales WHERE Descripcion LIKE '%ub%';
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	Clave	Descripcion
<input type="checkbox"/> Edit Copy Delete	1180	Recubrimiento P1001
<input type="checkbox"/> Edit Copy Delete	1190	Recubrimiento P1010
<input type="checkbox"/> Edit Copy Delete	1200	Recubrimiento P1019
<input type="checkbox"/> Edit Copy Delete	1210	Recubrimiento P1028
<input type="checkbox"/> Edit Copy Delete	1220	Recubrimiento P1037
<input type="checkbox"/> Edit Copy Delete	1290	Tubería 3.5
<input type="checkbox"/> Edit Copy Delete	1300	Tubería 4.3
<input type="checkbox"/> Edit Copy Delete	1310	Tubería 3.6
<input type="checkbox"/> Edit Copy Delete	1320	Tubería 4.4
<input type="checkbox"/> Edit Copy Delete	1330	Tubería 3.7
<input type="checkbox"/> Edit Copy Delete	1340	Tubería 4.5
<input type="checkbox"/> Edit Copy Delete	1350	Tubería 3.8

↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

✓ Showing rows 0 - 19 (20 total, Query took 0.0008 seconds.)

```
-- Denominación y suma del total a pagar para todos los proyectos. SELECT p.Denominacion, SUM(e.Cantidad * (m.Precio * (1 + m.PorcentajeImpuesto / 100))) AS TotalPagar FROM Entregan e INNER JOIN Materiales m.Clave INNER JOIN Proyectos p ON e.Numero = p.Numero GROUP BY p.Denominacion;
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Denominacion	TotalPagar
Ampliación de la carretera a la huasteca	578970.509
Aztecon	150200.21899999998
CIT Campeche	161603.082
CIT Yucatan	871522.3
Construcción de Hospital Infantil	147860.80299999999
Construcción de plaza Magnolias	125890.62340000001
Disco Atlantic	162221.662
Educando en Coahuila	634846.174
Infonavit Durango	328664.165
Mexico sin ti no estamos completos	266863.974
Queretaro limpio	460272.3810000001
Reconstrucción del templo de Guadalupe	225765.124
Remodelación de aulas del IPP	231569.147
Remodelación de Soriana	291233.46499999997
Reparación de la carretera del sol	140370.86800000002
Reparación de la plaza Sonora	540430.561
Restauración de instalaciones del CEA	868515.527
Televisa en acción	102429.64080000002

■ Console

LAB20 OUTPUTS

Showing rows 0 - 3 (4 total. Query took 0.0029 seconds)

-- La cantidad total entregada por proveedor para el proyecto "Televisa en acción".
`SELECT Proveedores.RFC, Proveedores.RazonSocial, SUM(Entregan.Cantidad) as CantidadTotal FROM Proveedores, Entregan WHERE Proveedores.RFC = Entregan.RFC AND Entregan.Numero = (SELECT Numero FROM Proyectos WHERE Denominacion = "Televisa en acción") GROUP BY Proveedores.RFC, Proveedores.RazonSocial;`

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

← T →

		RFC	RazonSocial	CantidadTotal		
<input type="checkbox"/>				AAAA800101	La fragua	86
<input type="checkbox"/>				CCCC800101	La Ferre	278
<input type="checkbox"/>				DDDD800101	Cecoferre	883
<input type="checkbox"/>				EEEE800101	Alvin	107

☐ Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table

Showing rows 0 - 24 (42 total. Query took 0.0005 seconds)

-- Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos.
`SELECT m.Descripcion AS Material, COUNT(*) AS CantidadEntregas, SUM(e.Cantidad * m.Precio + e.Cantidad * m.Precio * m.PorcentajeImpuesto / 100) AS TotalCosto FROM materiales m INNER JOIN entregan e ON m.Clave = e.Clave GROUP BY m.Descripcion ORDER BY TotalCosto DESC;`

[Edit inline] [Edit] [Create PHP code]

1 > >> | Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Material	CantidadEntregas	TotalCosto
Cantera gris	2	1127663.13
Pintura C1011	4	769517.47
Cantera rosa	2	589895.264
Tubería 4.4	2	262275.99199999997
Cemento	2	258813.96
Tubería 4.5	2	256186.59999999998
Pintura B1021	4	240558.59999999998
Varilla 4/34	2	201188.05
Varilla 3/19	2	197132.44799999997
Recubrimiento P1037	2	195619.424
Recubrimiento P1019	2	187269.12
Cantera blanca	2	164057.12
Tubería 3.7	2	159410.448
Recubrimiento P1001	2	151697.52
Tubería 4.3	2	137894.4
Varilla 3/18	2	131152.384
Varilla 4/32	2	123306.473
Pintura C1012	2	113252.54999999999

Console