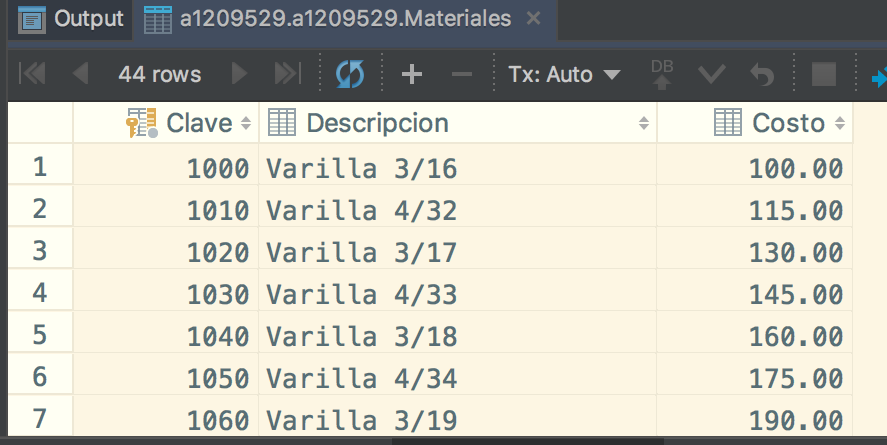
**Michelle Fernanda Anguiano Juárez – A01209529**

**Lab 15**

**Consulta de una tabla completa**

* Sentencia: **select** *\** **from** materiales
* Muestra de Salida



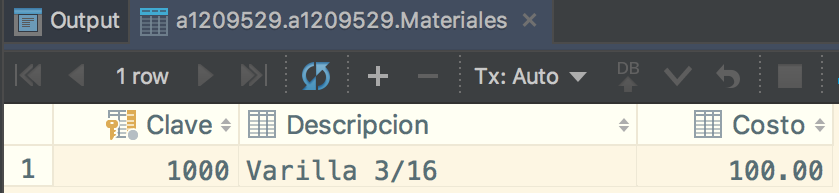
* Número de Renglones: 44

**Selección**

* Sentencia:

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

* Muestra de Salida



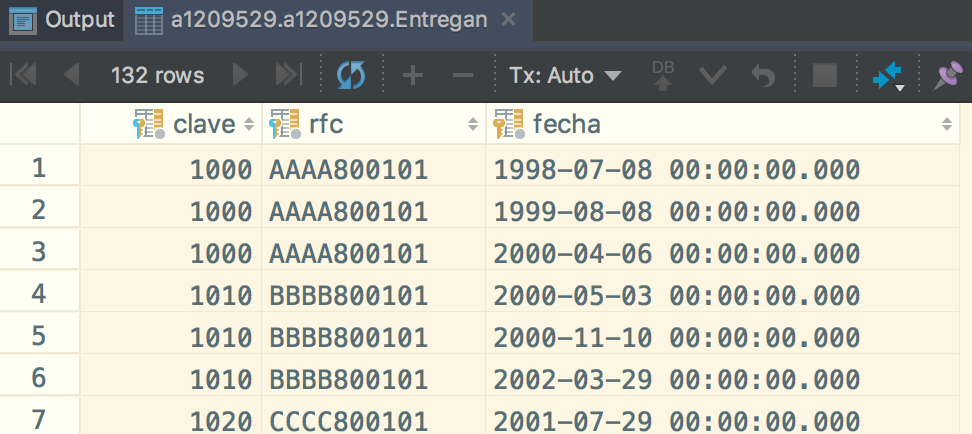
* Número de Renglones: 1

**Proyección**

* Sentencia:

**select clave**,**rfc**,**fecha from** entregan

* Muestra de Salida



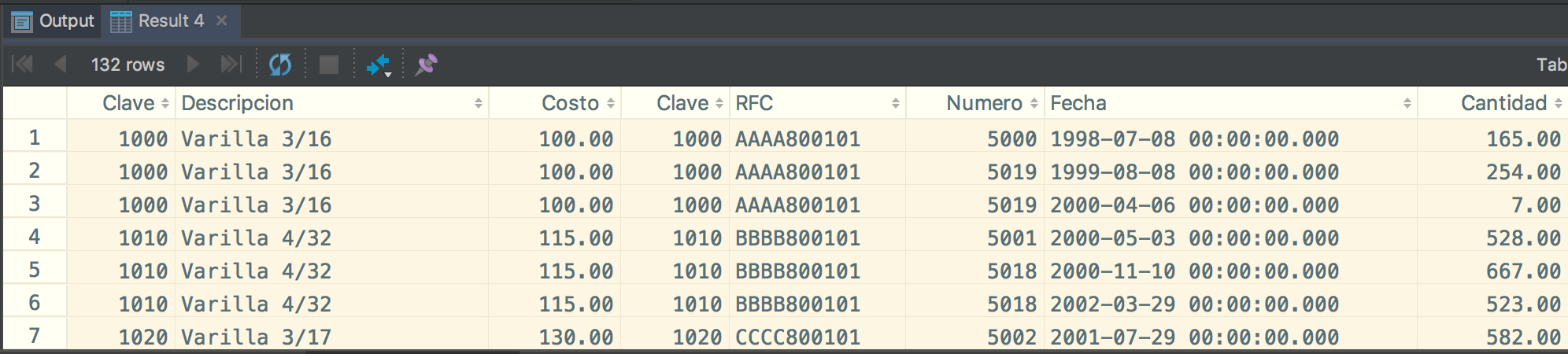
* Número de Renglones: 132

**Reunión Natural**

* Sentencia:

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

* Muestra de Salida



* Número de Renglones: 132
* **Si algún material no se ha entregado, ¿Aparecería en el resultado de esta consulta?**

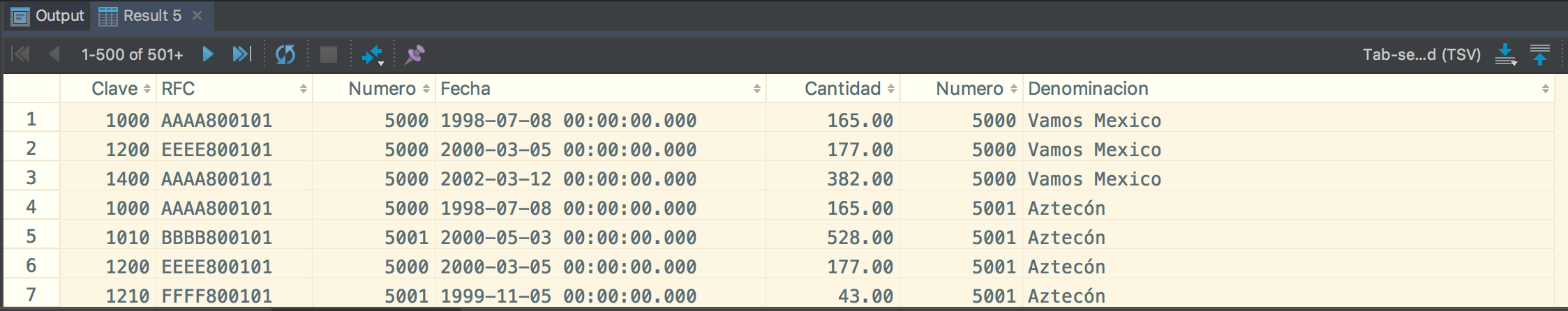
No, no aparecería porque la consulta solamente arroja los materiales que han sido entregados

**Reunión con criterio específico**

* Sentencia:

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

* Muestra de Salida



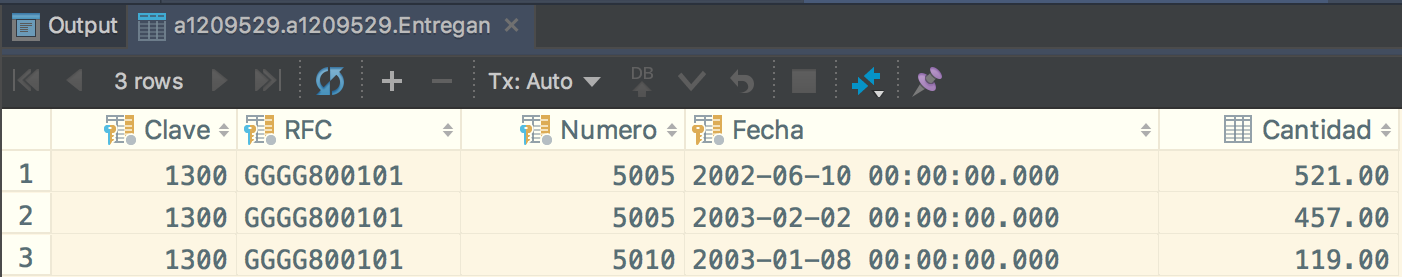
* Número de Renglones: 1188

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

* Sentencia:

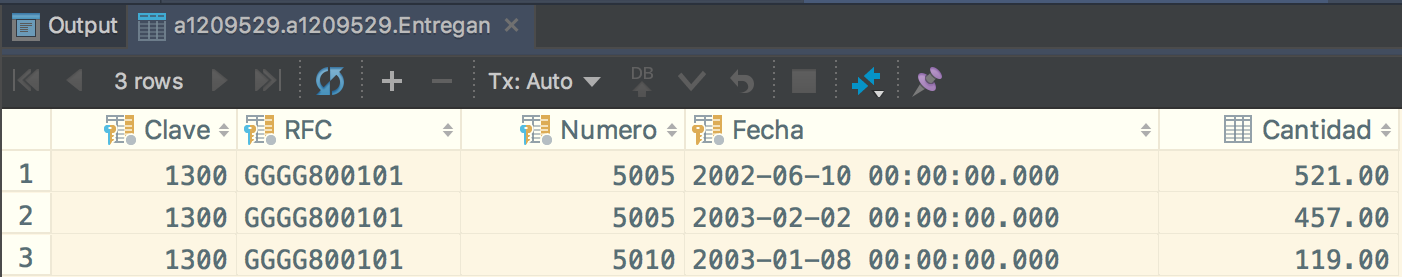
(**select** \* **from** entregan **where** clave=1450)  
**union**(**select** *\** **from** entregan **where clave**=1300)

* Muestra de Salida



* Número de Renglones: 3
* **¿Cuál sería una consulta que obtuviera el mismo resultado sin usar el operador Unión? Compruébalo.**

**SELECT** *\****FROM** Entregan  
**WHERE Clave** = 1450 **OR Clave** = 1300

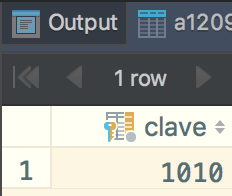


**Intersección (se ilustra junto con selección y proyección)**

* Sentencia:

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

* Muestra de Salida



* Número de Renglones: 1

**Nota**: No debería haber funcionado ya que no existe este comando en SQL Server pero se probó y el resultado es el mostrado anteriormente.

**Diferencia (se ilustra con selección)**

* Sentencia:

(**select** \* **from** entregan)  
minus  
(**select** \* **from** entregan **where** clave=1000)

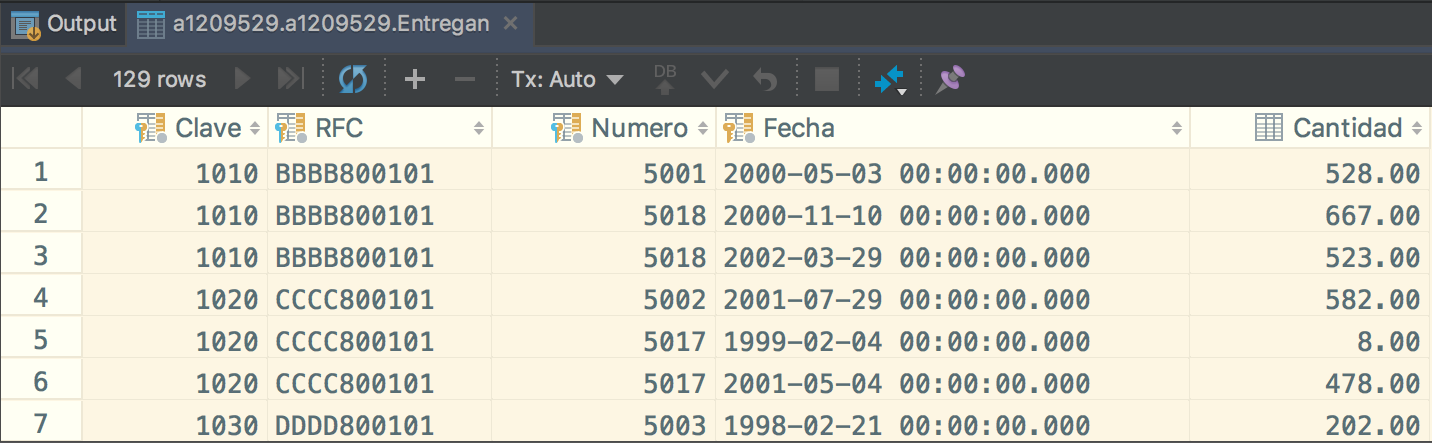
* Muestra de Salida

No hay

* Número de Renglones: -

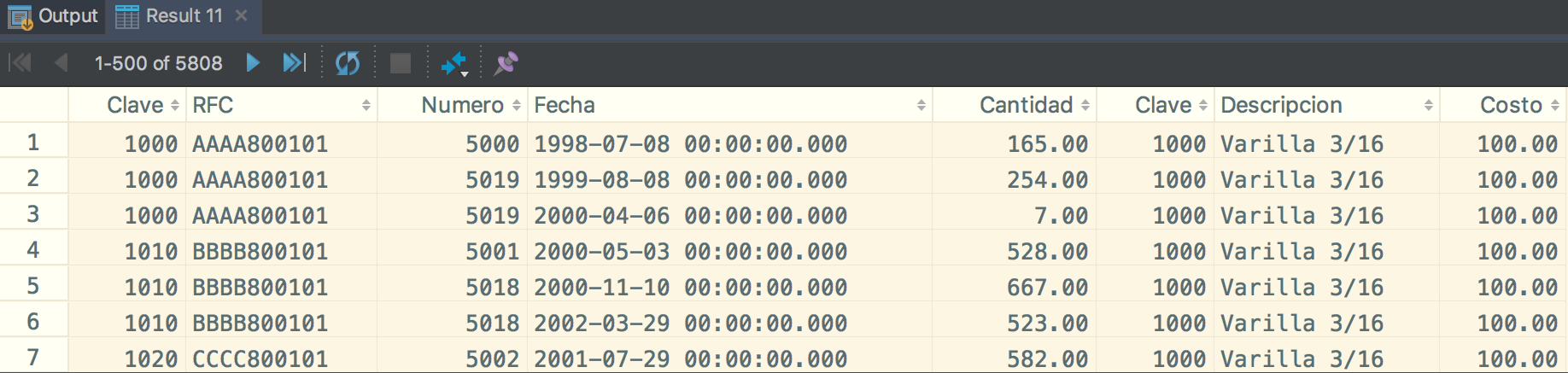
**La palabra "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.**

**SELECT** *\****FROM** Entregan  
**WHERE clave**!=1000



**Producto cartesiano**

* Sentencia: **select** *\** **from** entregan,materiales
* Muestra de Salida

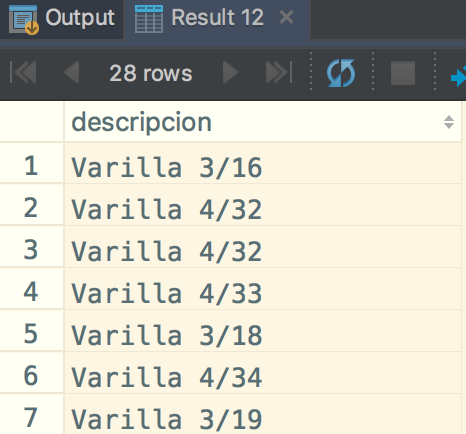


* Número de Renglones: 5808

**¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?**

Son 5808 filas, que representan todas las combinaciones posibles entre la tabla Entregan y la tabla Materiales. Entregan tiene 132 filas y materiales 44, al multiplicar 132\*44 el resultado es de 5808 filas.

**Construcción de consultas a partir de una especificación**

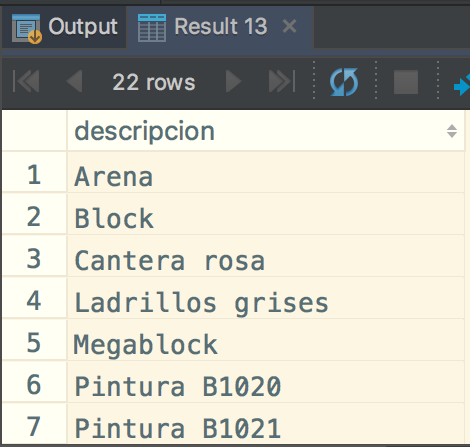
Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

**SET DATEFORMAT** DMY  
**SELECT M**.**descripcion  
FROM** Entregan E, Materiales **M  
WHERE Fecha BETWEEN '01/01/2000' AND '31/12/2000'  
AND** E.**Clave** = **M**.**Clave**

**¿Por qué aparecen varias veces algunas descripciones de material?**

Porque la consulta esta mostrando la descripción de todos los materiales que han sido entregados entre las fechas que se indican, pero no los esta agrupando, por eso se muestran repetidas.

**Uso del calificador distinct**

Agrega la palabra distinct inmediatamente después de la palabra select a la consulta que planteaste antes.

**SET DATEFORMAT** DMY  
**SELECT DISTINCT M**.**descripcion  
FROM** Entregan E, Materiales **M  
WHERE Fecha BETWEEN '01/01/2000' AND '31/12/2000'  
AND** E.**Clave** = **M**.**Clave**

¿Qué resultado obtienes en esta ocasión?

Esta consulta ya no arroja materiales repetidos, ya que con la palabra “DISTINCT” los esta separando, en lugar de ser 28 renglones, ahora muestra 22.

**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.

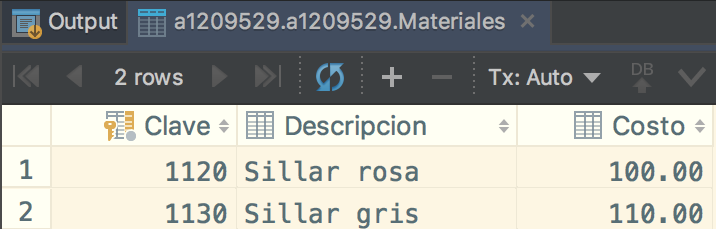
**Select** p.**numero**, p.**denominacion**, e.**fecha**, e.**cantidad  
FROM** Proyectos p, Entregan e  
**WHERE** p.**Numero** = e.**Numero  
ORDER BY** p.**Numero**, **fecha desc**



**Operadores de cadena**

**SELECT** *\** **FROM** Materiales **where Descripcion LIKE 'Si%'**

**¿Qué resultado obtienes?**



Todos los materiales que empiezan con las letras “Si”, no importa si hay mas letras después.

**Explica que hace el símbolo '%'**

Actúa como un filtro, que dentro de la consulta le indica que después de ese símbolo solo filtre las palabras que empiecen con las letras o caracteres subsecuentes.

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

No muestra nada, porque solamente busca los materiales que contengan esas letras única y específicamente.

**¿Qué resultado obtienes?**

cero filas

**Explica a qué se debe este comportamiento.**

Ningún material tiene una descripción que sea idéntica a ‘Si’

**¿Qué resultado obtienes de ejecutar el siguiente código?**

**DECLARE** @foo **varchar**(40);  
**DECLARE** @bar **varchar**(40);  
**SET** @foo = **'¿Que resultado'**;  
**SET** @bar = **' ¿¿¿??? '  
SET** @foo += **' obtienes?'**;  
**PRINT** @foo + @bar;

La cadena “¿Que resultado obtienes? ¿¿¿???”

**¿Para qué sirve DECLARE?**

Para declarar una variable

**¿Cuál es la función de @foo?**

Asignar el nombre a la variable

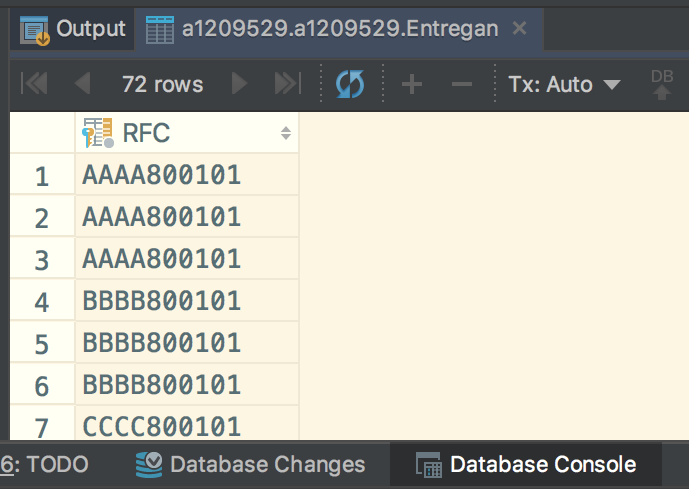
**¿Que realiza el operador SET?**

Asigna un valor a una variable

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

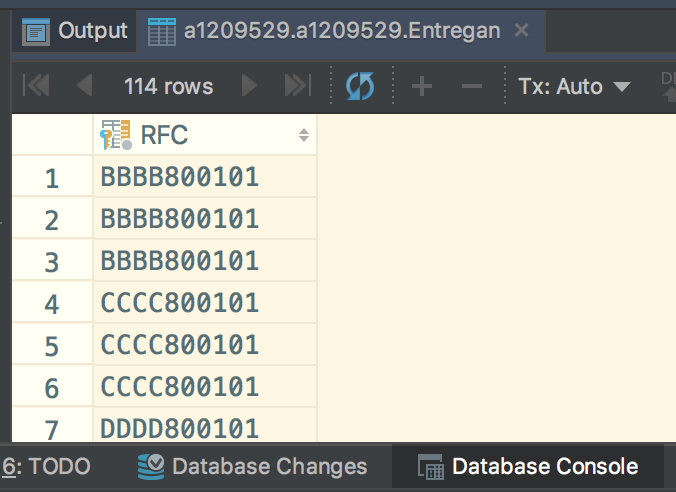
**SELECT RFC FROM** Entregan **WHERE RFC LIKE '[A-D]%'**;

Devuelve los RFC de la tabla entregan que empiecen con un carácter que este entre A y D y que después tenga cualquier cadena (incluso la cadena vacía), regresa 72 filas.



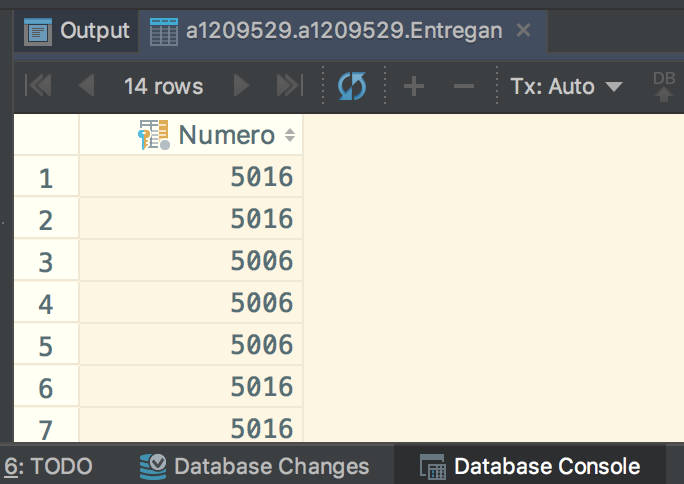
**SELECT RFC FROM** Entregan **WHERE RFC LIKE '[^A]%'**;

Devuelve los RFC de la tabla entregan que empiecen con cualquier carácter que no sea ‘A’ y que después tengan cualquier cadena (incluso la cadena vacía), devuelve 114 filas



**SELECT Numero FROM** Entregan **WHERE Numero LIKE '\_\_\_6'**;

Devuelve los números de la tabla entregan con una longitud de 4 caracteres/dígitos, donde el 4to carácter tiene que ser un 6, regresa 14 filas.



**¿Cómo filtrarías rangos de fechas?**

**SET DATEFORMAT** DMY  
**SELECT** *\** **FROM** Entregan  
**WHERE Fecha BETWEEN '01/01/1998' AND '31/12/1999'**

**¿Qué hace la siguiente consulta?**

**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]** )

Primero busca que el número del proyecto este entre 5000 y 5010, después de esas entregas devuelve el RFC, la cantidad, fecha y el numero si la Razon Social del proveedor empieza con ‘La’ y seguido tiene cualquier cadena.

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

Engloba una subconsulta.

**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** Entregan.**RFC IN** (**SELECT [RFC]  
FROM** [Proveedores]  
**WHERE RazonSocial LIKE 'La%' and** [Entregan].**[RFC]** = [Proveedores].**[RFC]**)

**Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN.**

**SELECT RFC**,**Cantidad**, **Fecha**,**Numero  
FROM** [Entregan]  
**WHERE [Numero] Between** 5000 **and** 5010  
**AND** Entregan.**RFC NOT IN** (**SELECT [RFC]  
FROM** [Proveedores]  
**WHERE RazonSocial NOT LIKE 'La%' and** [Entregan].**[RFC]** = [Proveedores].**[RFC]**)

**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%' and** [Entregan].**[RFC]** = [Proveedores].**[RFC]** )

**¿Qué hace la siguiente sentencia? Explica porque**

**SELECT TOP** 2 *\** **FROM** Proyectos

Selecciona los dos primeros registros de todos los Proyectos. Top 2 escoge los dos primeros y con \* se están seleccionando todos los registros.

**¿Qué sucede con la siguiente consulta? Explica porque**

**SELECT TOP Numero FROM** Proyectos

Esta consulta en realidad no funciona, tuve que poner:

**SELECT TOP** 1 **Numero FROM** Proyectos

Selecciona el primer número de la tabla proyectos. Con top le decimos que seleccione desde arriba y 1 le dice que sea el primero.

**¿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.**Cantidad**, **M**.**Costo**, **M**.**PorcentajeImpuesto**, (**Cantidad**\***Costo**)\*(1+**PorcentajeImpuesto**/100) **as** Importe  
**FROM** Materiales **M**, Entregan E  
**WHERE M**.**Clave** = E.**Clave**

**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** [Entregas entre 1998 y 1999] **AS  
SELECT** *\** **FROM** Entregan  
**WHERE Fecha BETWEEN '01/01/1998' AND '31/12/1999'**

**CREATE VIEW** [Entregas con RFC que inicia con letra entre A y D] **AS  
SELECT RFC FROM** Entregan **WHERE RFC LIKE '[A-D]%'**;

**CREATE VIEW** [Descripcion de materiales entregados en el año 2000] **AS  
SELECT DISTINCT M**.**descripcion  
FROM** Entregan E, Materiales **M  
WHERE Fecha BETWEEN '01/01/2000' AND '31/12/2000'  
AND** E.**Clave** = **M**.**Clave**

**CREATE VIEW** [Entregas con clave 1450 o 1300] **AS  
SELECT** *\****FROM** Entregan  
**WHERE Clave** = 1450 **OR Clave** = 1300

**CREATE VIEW** [Entregas con clave distinta de 1000] **AS   
SELECT** *\****FROM** Entregan  
**WHERE clave**!=1000

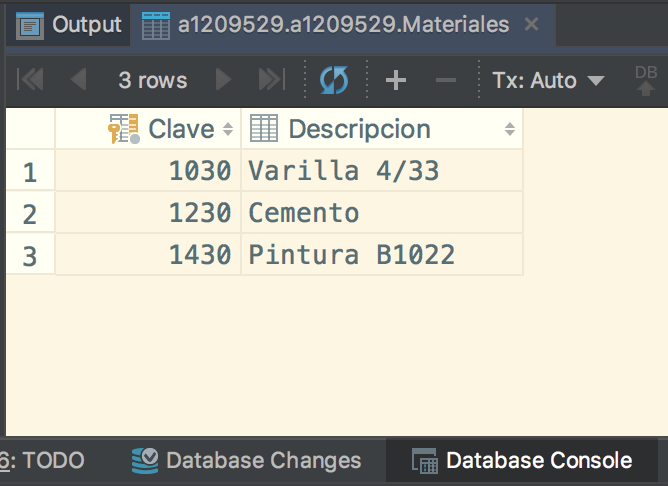
**A continuación se te dan muchos enunciados de los cuales deberás generar su correspondiente consulta.**

**En el reporte incluye la sentencia, una muestra de la salida (dos o tres renglones) y el número de renglones que SQL Server reporta al final de la consulta.**

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

**SELECT m**.**Clave**, **Descripcion  
FROM** Proyectos p, Materiales **m**, Entregan e  
**WHERE Denominacion LIKE '%Mexico sin ti no estamos completos%'  
AND** p.**Numero** = e.**Numero  
AND m**.**Clave** = e.**Clave**

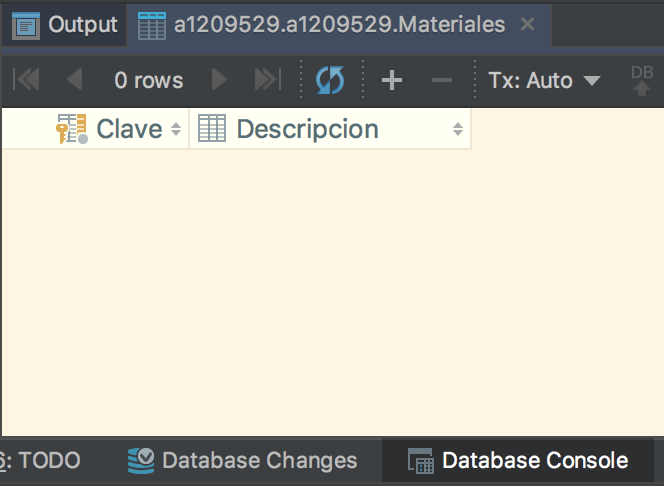
Regresa 3 filas

****

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

**SELECT m**.**Clave**, **Descripcion  
FROM** Proveedores p, Materiales **m**, Entregan e  
**WHERE RazonSocial LIKE '%Acme tools%'  
AND** p.**RFC** = e.**RFC  
AND m**.**Clave** = e.**Clave**

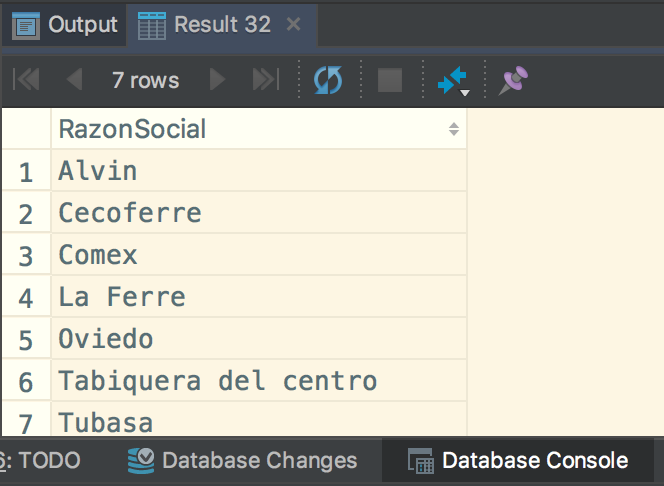
Regresa 0 filas



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

**SET DATEFORMAT** DMY  
**SELECT RazonSocial  
FROM** Proveedores p, Entregan e  
**WHERE** p.**RFC** = e.**RFC  
AND Fecha BETWEEN '01/01/2000' AND '31/12/2000'  
GROUP BY RazonSocial  
HAVING** *AVG*(**Cantidad**) > 300

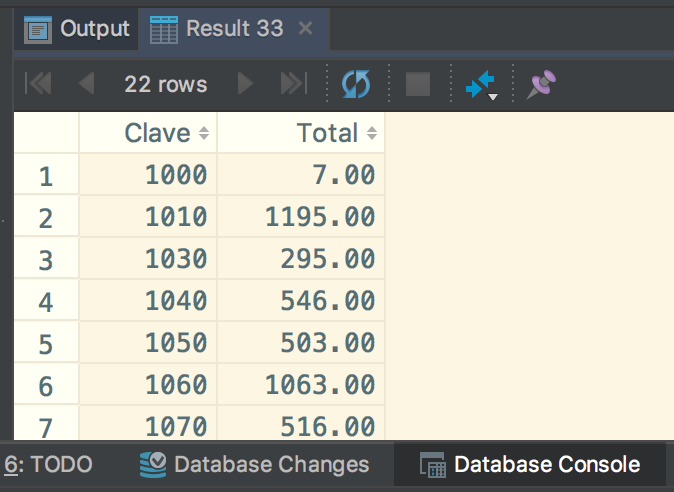
Regresa 7 filas



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

**SET DATEFORMAT** DMY  
**SELECT Clave**, *sum*(**Cantidad**) **as** Total  
**FROM** Entregan e  
**WHERE Fecha BETWEEN '01/01/2000' AND '31/12/2000'  
GROUP BY Clave**

Regresa 22 filas



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

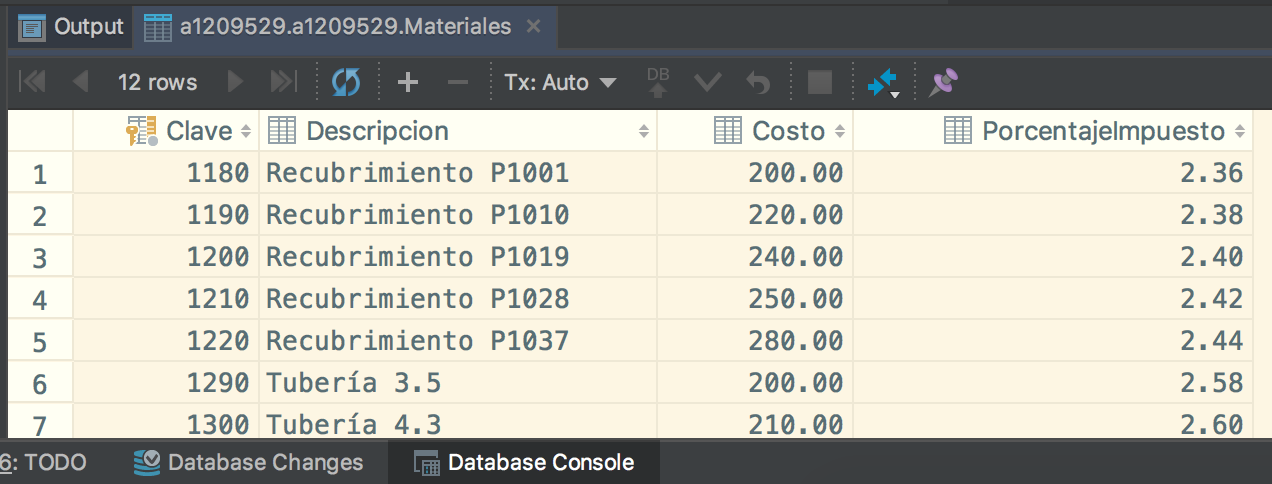
Resultado = 1020

Regresa 1 fila

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

**SELECT** *\****FROM** Materiales  
**WHERE Descripcion like '%ub%'**

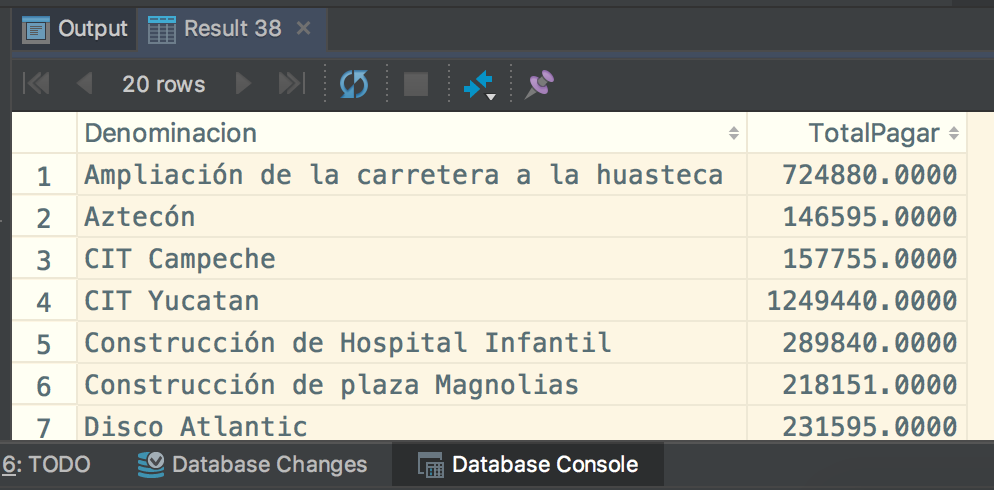
Regresa 12 filas



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

**SELECT Denominacion**, *SUM*(**Cantidad**\***Costo**) **as** TotalPagar  
**FROM** Materiales **m**, Proyectos p, Entregan e  
**WHERE m**.**clave** = e.**clave  
AND** p.**Numero** = e.**Numero  
GROUP BY Denominacion**

Regresa 20 filas



* **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 que apoyan Televisa] (**Denominacion**, **RFC**, **RazonSocial**) **AS  
SELECT DISTINCT Denominacion**, P.**RFC**, **RazonSocial  
FROM** Proveedores P, Proyectos Pr, Entregan E  
**WHERE** P.**RFC** = E.**RFC  
AND** Pr.**Numero** = E.**Numero  
AND Denominacion LIKE 'Televisa%'  
  
CREATE VIEW** [Proveedores que apoyan Educando a Coahuila] (**Denominacion**, **RFC**, **RazonSocial**) **AS  
SELECT DISTINCT Denominacion**, P.**RFC**, **RazonSocial  
FROM** Proveedores P, Proyectos Pr, Entregan E  
**WHERE** P.**RFC** = E.**RFC AND** Pr.**Numero** = E.**Numero AND Denominacion LIKE '%Coahuila%'  
  
CREATE VIEW** [Proveedores que apoyan Televisa y no Coahuila] **as  
SELECT RazonSocial FROM** [Proveedores que apoyan Televisa]  
**EXCEPT  
SELECT RazonSocial FROM** [Proveedores que apoyan Educando a Coahuila]  
**SELECT** *\** **FROM** [Proveedores que apoyan Televisa]  
**WHERE RazonSocial IN** (**SELECT** *\** **FROM** [Proveedores que apoyan Televisa y no Coahuila])

Resultado:

Televisa en acción CCCC800101 La Ferre

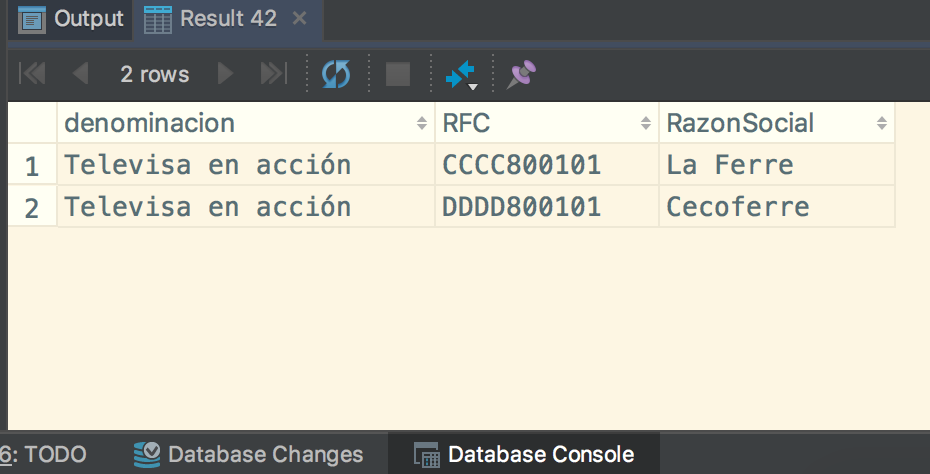
Televisa en acción DDDD800101 Cecoferre

Regresa 2 filas

* **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 DISTINCT denominacion**, pv.**RFC**, pv.**RazonSocial  
FROM** Proyectos py, Proveedores pv, Entregan e  
**WHERE** pv.**RFC** = e.**RFC  
AND** py.**Numero** = e.**Numero  
AND** py.**Denominacion LIKE 'Televisa%'  
AND** pv.**RazonSocial NOT IN** (**SELECT** pv.**RazonSocial  
 FROM** Proveedores pv, Entregan e, Proyectos py  
 **WHERE** pv.**RFC** = e.**RFC  
 AND** py.**Numero** = e.**Numero  
 AND** py.**denominacion LIKE 'Educando en Coahuila%'  
 GROUP BY** pv.**RazonSocial** )

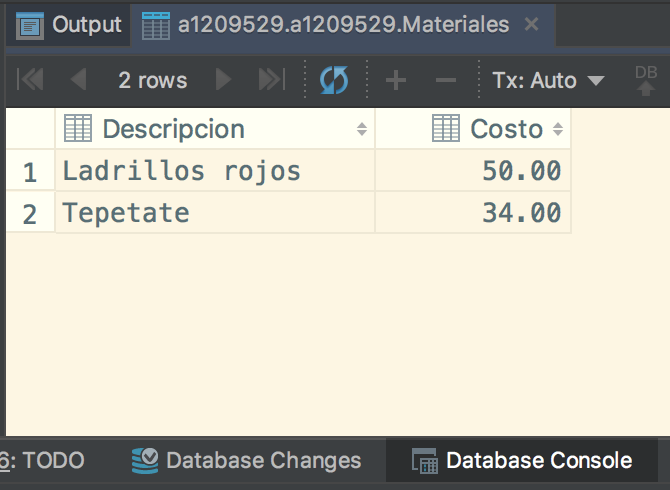
Regresa 2 filas



* **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**.**Descripcion**, **m**.**Costo  
FROM** Proyectos py, Proveedores pv, Entregan e, Materiales **m  
WHERE** pv.**RFC** = e.**RFC  
AND** py.**Numero** = e.**Numero  
AND m**.**Clave** = e.**Clave  
AND** py.**Denominacion LIKE 'Televisa%'  
AND** pv.**RazonSocial IN** (**SELECT** pv.**RazonSocial  
 FROM** Proveedores pv, Entregan e, Proyectos py  
 **WHERE** pv.**RFC** = e.**RFC  
 AND** py.**Numero** = e.**Numero  
 AND** py.**denominacion LIKE 'Educando en Coahuila%'  
 GROUP BY** pv.**RazonSocial** )

Regresa 2 filas



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

**SELECT Descripcion**, *count*(**Descripcion**), *SUM*(**Costo**\***Cantidad**)  
**FROM** Entregan e, Materiales **m  
WHERE** e.**Clave** = **m**.**Clave  
GROUP BY Descripcion**

Regresa 42 filas

