

Laboratorio 20 - Consultas en SQL

| | |
|---|----|
| Tuplas..... | 2 |
| Ordenamiento..... | 3 |
| Operadores de cadena..... | 3 |
| Operadores lógicos y compuestos..... | 5 |
| Modificando la estructura de una tabla existente..... | 9 |
| Creación de vistas..... | 9 |
| Consultas..... | 10 |

Tuplas

¿Cómo está definido el número de tuplas de este resultado (`select * from entregan,materiales`) en términos del número de tuplas de `entregan` y de `materiales`?

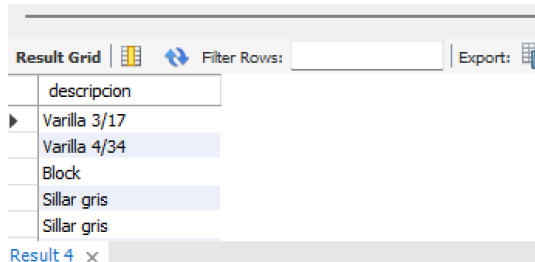
En álgebra relacional, la operación de producto cartesiano (denotada por el símbolo "x") entre dos conjuntos devuelve todas las combinaciones posibles de tuplas, donde cada tupla del primer conjunto se combina con cada tupla del segundo conjunto.

Suponiendo que el conjunto "Entregan" tiene 'm' tuplas y el conjunto "Materiales" tiene 'n' tuplas, el producto cartesiano entre ambos conjuntos producirá 'm * n' tuplas en el resultado.

Entonces, en términos del número de tuplas de `Entregan` (`|Entregan|`) y el número de tuplas de `Materiales` (`|Materiales|`), el número de tuplas en el resultado del producto cartesiano será:

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

```
20 • SELECT m.descripcion
21 FROM entregan e
22 JOIN materiales m ON e.clave = m.clave
23 WHERE YEAR(e.fecha) = 2000;
24
25
```



| descripcion |
|--------------|
| Varilla 3/17 |
| Varilla 4/34 |
| Block |
| Sillar gris |
| Sillar gris |

Si aparecen varias veces algunas descripciones de material en el resultado de la consulta, podría deberse a que hay múltiples entregas de un mismo material en el año 2000. Esto significa que para ese material en particular, ha habido más de una entrada en la tabla **entregan** asociada con entregas realizadas en el año 2000.

Para obtener cada descripción de material solo una vez en el resultado, puedes utilizar la cláusula **DISTINCT** en tu consulta. Esta cláusula elimina duplicados de los resultados devueltos, lo que garantiza que cada descripción de material aparezca solo una vez en la salida:

```

20 • SELECT distinct m.descripcion
21 FROM entregan e
22 JOIN materiales m ON e.clave = m.clave
23 WHERE YEAR(e.fecha) = 2000;

```

| Result Grid | Filter Rows: | Export: |
|----------------|--------------|---------|
| descripcion | | |
| Varilla 3/17 | | |
| Varilla 4/34 | | |
| Block | | |
| Sillar gris | | |
| Cantera blanca | | |

Ordenamiento

Para obtener los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto y presentando las fechas de la más reciente a la más antigua, puedes usar la siguiente consulta SQL:

```

25 • SELECT p.numero, p.denominacion, e.fecha, e.cantidad
26 FROM proyectos p
27 JOIN entregan e ON p.numero = e.numero
28 ORDER BY p.numero, e.fecha DESC;

```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------------|------------|--------------------|
| numero | denominacion | fecha | cantidad |
| 5000 | Vamos Mexico | 2003-03-15 | 177 |
| 5000 | Vamos Mexico | 2001-12-13 | 165 |
| 5000 | Vamos Mexico | 1999-04-07 | 382 |
| 5001 | Aztecon | 2000-05-21 | 43 |
| 5001 | Aztecon | 2000-05-18 | 601 |

Operadores de cadena

`SELECT * FROM productos where Descripcion LIKE 'Si%'` devolverá todas las filas de la tabla **productos** donde el valor de la columna **Descripcion** comienza con la cadena "Si".

El símbolo '%' en la cláusula **LIKE** se utiliza como comodín para representar cero o más caracteres. En este caso, 'Si%' indica que estamos buscando todas las filas donde la columna **Descripcion** comienza con "Si" seguido de cualquier cantidad de caracteres.

Si la consulta fuera `SELECT * FROM productos WHERE Descripcion LIKE 'Si'`, devolvería todas las filas de la tabla **productos** donde el valor de la columna **Descripcion** es exactamente "Si". Si no hay ninguna fila en la tabla productos con Descripcion exactamente igual a "Si", esta consulta no devolverá ningún resultado.

El comportamiento de estas consultas se debe a cómo funciona la comparación de cadenas con el operador LIKE. Cuando se usa `LIKE 'Si%'`, busca todas las cadenas que comienzan con "Si", mientras que `LIKE 'Si'` busca exactamente la cadena "Si"

`SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;` Esta consulta concatena las columnas **Apellido** y **Nombre** de la tabla **Personas**, separadas por una coma y un espacio, y renombra el resultado como **Nombre**.

```
DECLARE @foo varchar(40);
```

```
DECLARE @bar varchar(40);
```

```
SET @foo = '¿Que resultado?';
```

```
SET @bar = ' ¿¿¿??? ';
```

```
SET @foo += ' obtienes?';
```

```
PRINT @foo + @bar;
```

- DECLARE se utiliza para declarar variables en SQL Server. En este caso, se están declarando dos variables de tipo `varchar(40)`, `@foo` y `@bar`.
- `@foo` es una variable que se utiliza para almacenar una cadena de caracteres.
- SET se utiliza para asignar valores a variables. En este caso, se asigna una cadena de caracteres a las variables `@foo` y `@bar`.
- `+=` es un operador de concatenación que se usa para concatenar cadenas de caracteres en SQL Server.
- PRINT se utiliza para mostrar mensajes en la ventana de mensajes de SQL Server Management Studio.

El resultado de ejecutar este código sería: ¿Que resultado obtienes? ¿¿¿???

Esto se debe a que primero se asigna la cadena "¿Que resultado" a la variable `@foo`, luego se agrega la cadena " obtienes?" a `@foo`, y finalmente se imprime el valor de `@foo` seguido por el valor de `@bar`.

```
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';
```

Son consultas que utilizan el operador LIKE en combinación con patrones de búsqueda específicos para filtrar los resultados de la tabla Entregan. Explicaré el comportamiento, función y resultado de cada consulta:

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

- Comportamiento: Esta consulta selecciona todas las filas de la tabla Entregan donde el valor de la columna RFC comienza con una letra entre 'A' y 'D'.
- Función: Utiliza un patrón de búsqueda que incluye un rango de caracteres [A-D], seguido de % que indica cualquier número de caracteres adicionales.
- Resultado: Devolverá todas las filas donde el valor de la columna RFC comienza con una letra entre 'A' y 'D'.

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

- Comportamiento: Esta consulta selecciona todas las filas de la tabla Entregan donde el valor de la columna RFC no comienza con la letra 'A'.
- Función: Utiliza un patrón de búsqueda que incluye [^A], donde ^ nega la coincidencia con el carácter 'A', seguido de % que indica cualquier número de caracteres adicionales.
- Resultado: Devolverá todas las filas donde el valor de la columna RFC no comienza con la letra 'A'.

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

- Comportamiento: Esta consulta selecciona todas las filas de la tabla Entregan donde el valor de la columna Numero tiene tres caracteres seguidos por el número '6'.
- Función: Utiliza el patrón de búsqueda ___6, donde _ representa un único carácter comodín.
- Resultado: Devolverá todas las filas donde el valor de la columna Numero tiene exactamente tres caracteres y termina con el número '6'. Por ejemplo, 1236, 4566, etc.

Operadores lógicos y compuestos

Para filtrar rangos de fechas en SQL, puedes utilizar la cláusula BETWEEN junto con las fechas de inicio y fin del rango que deseas filtrar. Por ejemplo, si deseas filtrar las fechas entre el 1 de enero de 2023 y el 31 de diciembre de 2023, puedes hacerlo de la siguiente manera:

29

30 • `SELECT Clave, RFC, Numero, Fecha, Cantidad`

31 `FROM Entregan`

32 `WHERE Fecha BETWEEN '1998-01-01' AND '2005-12-31';`

33

Result Grid

| | Clave | RFC | Numero | Fecha | Cantidad |
|---|-------|------------|--------|------------|----------|
| ▶ | 1000 | AAAA800101 | 5000 | 2001-12-13 | 165 |
| | 1000 | AAAA800101 | 5019 | 1999-07-13 | 254 |
| | 1010 | BBBB800101 | 5001 | 1998-07-28 | 528 |
| | 1020 | CCCC800101 | 5002 | 2003-12-16 | 582 |
| | 1020 | CCCC800101 | 5017 | 2000-03-29 | 8 |

Fecha BETWEEN '2023-01-01' AND '2023-12-31' filtra las filas donde el valor de la columna Fecha está entre el 1 de enero de 2023 y el 31 de diciembre de 2023, inclusivo.

Uso de EXISTS

```
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] )
```

Esta consulta busca recuperar información de la tabla **Entregan** donde el número de entrega (**Numero**) esté entre 5000 y 5010, y donde exista un proveedor cuya razón social (**RazonSocial**) comience con "La" y cuyo RFC esté presente en la tabla **Proveedores**. ¿Qué función tiene el paréntesis () después de EXISTS?

La función de EXISTS es verificar si la subconsulta dentro de los paréntesis devuelve algún resultado. Si la subconsulta devuelve al menos una fila, la condición de EXISTS se evalúa como verdadera y se seleccionan las filas correspondientes de la tabla principal (en este caso, la tabla Entregan). Si la subconsulta no devuelve filas, la condición de EXISTS se evalúa como falsa y las filas no se seleccionan.

El paréntesis () después de EXISTS simplemente indica que el EXISTS está tomando una subconsulta como argumento. Es una práctica común en SQL rodear la subconsulta con paréntesis para mayor claridad y legibilidad del código. Sin embargo, los paréntesis son opcionales en este caso y pueden ser omitidos sin cambiar el resultado de la consulta.

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

```

9 • SELECT RFC, Cantidad, Fecha, Numero
10 FROM Entregan
11 WHERE Numero BETWEEN 5000 AND 5010 AND
12 RFC IN (
13     SELECT RFC
14     FROM Proveedores
15     WHERE RazonSocial LIKE 'La%'
16 );

```

| RFC | Cantidad | Fecha | Numero |
|------------|----------|------------|--------|
| AAAA800101 | 165 | 2001-12-13 | 5000 |
| AAAA800101 | 86 | 2005-04-03 | 5008 |
| AAAA800101 | 152 | 0000-00-00 | 5004 |
| AAAA800101 | 382 | 1999-04-07 | 5000 |
| AAAA800101 | 116 | 2005-04-21 | 5010 |

En esta consulta, la subconsulta dentro del operador IN devuelve todos los RFC de los proveedores cuya razón social (RazonSocial) comienza con "La". Luego, se seleccionan todas las filas de la tabla Entregan donde el número de entrega (Numero) esté entre 5000 y 5010 y cuyo RFC esté presente en la lista de RFC obtenida de la subconsulta. Este enfoque logra el mismo resultado que la consulta original que utiliza EXISTS.

Ahora, usando el operador NOT IN Realizar un ejemplo donde apliques algún operador : ALL, SOME o ANY.

```

SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 5000 AND 5010
AND RFC NOT IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
)
AND Cantidad > ALL (
    SELECT Cantidad
    FROM Entregan AS e
    WHERE e.RFC IN (
        SELECT RFC
        FROM Proveedores
        WHERE RazonSocial LIKE 'La%'
    )
);

```

El primer NOT IN selecciona todas las entregas cuyos RFC no están asociados con proveedores cuya razón social comienza con "La".

El operador ALL se utiliza para comparar la cantidad de cada entrega con la cantidad mínima de todas las entregas asociadas a proveedores cuya razón social comienza con "La". Esto significa que solo se seleccionarán las entregas donde la cantidad sea mayor que todas las cantidades mínimas de las entregas asociadas a proveedores con razón social que comienza con "La".

SELECT TOP 2 * FROM Proyectos

La sentencia SELECT TOP 2 * FROM Proyectos selecciona las dos primeras filas de la tabla Proyectos.

TOP 2 indica que queremos seleccionar solo las primeras dos filas.

* significa que queremos seleccionar todas las columnas de la tabla.

FROM Proyectos especifica de qué tabla queremos seleccionar las filas.

La razón por la que esta sentencia selecciona las dos primeras filas es porque SQL Server no garantiza el orden en el que devuelve las filas a menos que se especifique explícitamente mediante una cláusula ORDER BY. En ausencia de una cláusula ORDER BY, SQL Server devolverá las filas en un orden que puede ser aleatorio o basado en la estructura interna de la tabla. Por lo tanto, las dos primeras filas que devuelve la consulta pueden variar a menos que se especifique un orden específico con ORDER BY.

SELECT TOP Numero FROM Proyectos

La consulta SELECT TOP Numero FROM Proyectos intenta seleccionar un número específico de filas de la tabla Proyectos basado en el valor de la columna Numero. Sin embargo, esta consulta generará un error, ya que el valor que se espera después de TOP debe ser un número entero constante que indique el número de filas que se desean seleccionar, y no una columna de la tabla.

Modificando la estructura de una tabla existente.

```
7
8 ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);
9 • UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;
```

| clave | descripcion | precio | impuesto | PorcentajeImpuesto |
|-------|--------------|--------|----------|--------------------|
| 1000 | Varilla 3/16 | 100 | 10 | 2.00 |
| 1010 | Varilla 4/32 | 115 | 11.5 | 2.02 |
| 1020 | Varilla 3/17 | 130 | 13 | 2.04 |
| 1030 | Varilla 4/33 | 145 | 14.5 | 2.06 |
| 1040 | Varilla 3/18 | 160 | 16 | 2.08 |

Para obtener el importe total de las entregas, es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega, el precio del material y el impuesto asignado, puedes usar una consulta que realice el cálculo multiplicando la cantidad entregada por el precio del material, sumando el impuesto y multiplicando por la cantidad entregada:

```
56 • SELECT e.Numero, m.descripcion AS Material, e.Cantidad,
57       (e.Cantidad * m.Precio * (1 + m.PorcentajeImpuesto / 100)) AS Importe_Total
58 FROM Entregan e
59 JOIN materiales m ON e.clave = m.clave;
60
```

| Numero | Material | Cantidad | Importe_Total |
|--------|--------------|----------|-------------------|
| 5000 | Varilla 3/16 | 165 | 16830 |
| 5019 | Varilla 3/16 | 254 | 25908 |
| 5001 | Varilla 4/32 | 528 | 61946.544 |
| 5018 | Varilla 4/32 | 523 | 61359.929 |
| 5002 | Varilla 3/17 | 582 | 77203.46399999999 |

Creación de vistas

Para crear las vistas basadas en las consultas anteriores, podemos hacer lo siguiente

```
CREATE VIEW Descripciones_Materiales_2000 AS
SELECT DISTINCT m.descripcion
FROM Entregan e
JOIN materiales m ON e.clave = m.clave
WHERE YEAR(e.fecha) = 2000;
```

Vista para la consulta que obtiene las descripciones de los materiales entregados en el año 2000

```
CREATE VIEW Detalles_Entregas_Proyectos AS
SELECT p.numero, p.denominacion, e.fecha, e.cantidad
FROM proyectos p
JOIN entregan e ON p.numero = e.numero;
```

Vista para la consulta que obtiene los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas

```
CREATE VIEW Importe_Total_Entregas AS
SELECT e.Numero, m.descripcion AS Material, e.Cantidad,
       (e.Cantidad * m.Precio * (1 + m.PorcentajeImpuesto / 100)) AS Importe_Total
FROM Entregan e
JOIN materiales m ON e.clave = m.clave;
```

Vista para la consulta que calcula el importe total de las entregas

```
CREATE VIEW Materiales_Con_Impuesto_Mayor AS
SELECT *
FROM materiales
WHERE PorcentajeImpuesto > 10; -- Por ejemplo,
```

Vista para la consulta que selecciona los materiales entregados cuyo impuesto es mayor que un valor específico

```
CREATE VIEW Proyectos_Con_Entregas_La AS
SELECT DISTINCT p.*
FROM proyectos p
JOIN entregan e ON p.numero = e.numero
JOIN proveedores pr ON e.RFC = pr.RFC
WHERE pr.RazonSocial LIKE 'La%';
```

Vista para la consulta que obtiene los proyectos con entregas realizadas por proveedores cuya razón social comienza con "La"

Consultas

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

```

86 • SELECT m.clave, m.descripcion
87 FROM Materiales m
88 JOIN Entregan e ON m.clave = e.clave
89 JOIN Proyectos p ON e.numero = p.numero
90 WHERE p.denominacion = 'Mexico sin ti no estamos completos';
91

```

| Result Grid | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|-------|---------------|---------|--------------------|
| | clave | descripcion | | |
| ▶ | 1030 | Varilla 4/33 | | |
| | 1230 | Cemento | | |
| | 1430 | Pintura B1022 | | |

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

```

92 • SELECT m.clave, m.descripcion
93 FROM materiales m
94 JOIN entregan e ON m.clave = e.clave
95 JOIN proveedores p ON e.rfc = p.rfc
96 WHERE p.razonsocial = 'Acme Tools';

```

| Result Grid | | Filter Rows: | Export: |
|-------------|-------|--------------|---------|
| | clave | descripcion | |

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

```

98 • SELECT e.rfc, AVG(e.cantidad) AS promedio_entregas
99 FROM entregan AS e
100 WHERE YEAR(e.fecha) = 2000
101 GROUP BY e.rfc
102 HAVING AVG(e.cantidad) >= 300;

```

| Result Grid | | Filter Rows: | Export: | Wrap Cell Coi |
|-------------|------------|-------------------|---------|---------------|
| | rfc | promedio_entregas | | |
| ▶ | BBBB800101 | 601.0000 | | |
| | FFFF800101 | 322.7500 | | |
| | GGGG800101 | 583.0000 | | |

El Total entregado por cada material en el año 2000.

```

104 • SELECT m.clave, m.descripcion, SUM(e.cantidad) AS total_entregado
105 FROM entregan e
106 JOIN materiales m ON e.clave = m.clave
107 WHERE YEAR(e.fecha) = 2000
108 GROUP BY e.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 | |
| 1140 | Cantera blanca | 583 | |
| 1210 | Recubrimiento P1028 | 43 | |

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

- Crear vista intermedia:

```

111 • CREATE VIEW entregas_2001 AS
112 SELECT clave, SUM(cantidad) AS total_entregado
113 FROM entregan
114 WHERE YEAR(fecha) = 2001
115 GROUP BY clave;

```

- Consultar la vista para encontrar el material más vendido

```

117 • SELECT clave
118 FROM entregas_2001
119 ORDER BY total_entregado DESC
120 LIMIT 1; #mostrar el material mas vendido

```

| Result Grid | Filter Rows: | Export: | V |
|-------------|--------------|---------|---|
| clave | | | |
| 1260 | | | |

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

```

122 • SELECT clave, descripcion
123 FROM materiales
124 WHERE descripcion LIKE '%ub%'

```

| Result Grid | | | Filter Rows: | |
|-------------|-------|---------------------|--------------|--|
| | clave | descripcion | | |
| ▶ | 1180 | Recubrimiento P1001 | | |
| | 1190 | Recubrimiento P1010 | | |
| | 1200 | Recubrimiento P1019 | | |
| | 1210 | Recubrimiento P1028 | | |
| | 1220 | Recubrimiento P1037 | | |

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

```

126 • SELECT
127     p.numero,
128     p.denominacion,
129     SUM((m.precio + m.precio * m.impuesto / 100) * e.cantidad) AS total_a_pagar
130 FROM proyectos p
131 JOIN entregan e ON p.numero = e.numero
132 JOIN materiales m ON e.clave = m.clave
133 GROUP BY p.numero, p.denominacion

```

| Result Grid | | | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|--------|------------------------------------|---------------------|---------|--------------------|
| | numero | denominacion | total_a_pagar | | |
| ▶ | 5000 | Vamos Mexico | 124543.950000000001 | | |
| | 5001 | Aztecon | 165655.925 | | |
| | 5002 | CIT Campeche | 178894.275 | | |
| | 5003 | Mexico sin ti no estamos completos | 321237.05 | | |
| | 5004 | Educando en Coahuila | 1296660.1 | | |
| | 5005 | Infraavit Durango | 383545.475 | | |

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).

```

136 -- Vista para el proyecto "Televisa en acción"
137 • CREATE VIEW vista_televisa AS
138 SELECT e.rfc
139 FROM entregan e
140 JOIN proyectos p ON e.numero = p.numero
141 WHERE p.denominacion = 'Televisa en acción';
---

143 -- Vista para el proyecto "Educando en Coahuila"
144 • CREATE VIEW vista_educando AS
145 SELECT e.rfc
146 FROM entregan e
147 JOIN proyectos p ON e.numero = p.numero
148 WHERE p.denominacion = 'Educando en Coahuila';

```

```

150 • SELECT p.razonsocial, p.rfc, pro.denominacion
151 FROM proveedores p
152 JOIN vista_televisa vt ON p.rfc = vt.rfc
153 LEFT JOIN vista_educando ve ON p.rfc = ve.rfc
154 JOIN proyectos pro ON pro.denominacion = 'Televisa en acción'
155 WHERE ve.rfc IS NULL;
156

```

| razonsocial | rfc | denominacion |
|-------------|------------|--------------------|
| Cecoferre | DDDD800101 | Televisa en acción |
| Cecoferre | DDDD800101 | Televisa en acción |
| La Ferre | CCCC800101 | Televisa en acción |

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).

```

157 • SELECT DISTINCT p.razonsocial, p.rfc, pr.denominacion
158 FROM proveedores p
159 JOIN entregan e ON p.rfc = e.rfc
160 JOIN proyectos pr ON e.numero = pr.numero
161 WHERE pr.denominacion = 'Televisa en acción'
162 AND p.rfc NOT IN (
163     SELECT e2.rfc
164     FROM entregan e2
165     JOIN proyectos pr2 ON e2.numero = pr2.numero
166     WHERE pr2.denominacion = 'Educando en Coahuila')
167

```

| razonsocial | rfc | denominacion |
|-------------|------------|--------------------|
| Cecoferre | DDDD800101 | Televisa en acción |
| La Ferre | CCCC800101 | Televisa en acción |

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

```

168 • SELECT
169     m.clave,
170     m.descripcion,
171     m.precio + m.precio * m.impuesto / 100 AS costo
172 FROM materiales m
173 JOIN entregan e ON m.clave = e.clave
174 JOIN proyectos p ON e.numero = p.numero
175 WHERE p.denominacion = 'Televisa en acción'
176 AND EXISTS (
177     SELECT 1
178     FROM entregan e2
179     JOIN proyectos p2 ON e2.numero = p2.numero
180     WHERE p2.denominacion = 'Educando en Coahuila'
181     AND e2.rfc = e.rfc);
182

```

| Result Grid | | | |
|-------------|-------|-----------------|-------------------|
| | clave | descripcion | costo |
| ▶ | 1080 | Ladrillos rojos | 52.5 |
| | 1280 | Tepetate | 35.15600003242493 |