Lab 20

Emiliano Valdivia Lara

A01276258

Consulta de un tabla completa

Algebra relacional. materiales

SQL

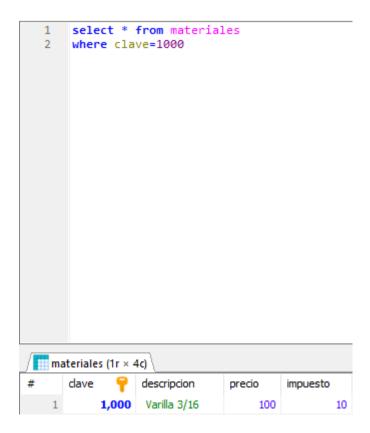
select * from materiales

SELECT * FROM materiales materiales (45r × 4c) dave descripcion precio impuesto 1 1,000 Varilla 3/16 100 10 2 1,010 Varilla 4/32 115 11.5 3 1,020 Varilla 3/17 130 13 4 1,030 Varilla 4/33 145 14.5 5 1,040 Varilla 3/18 160 16 6 1,050 Varilla 4/34 17.5 175 7 1,060 Varilla 3/19 190 19 8 1,070 Varilla 4/35 205 20.5 9 1,080 Ladrillos rojos 5 50 1,090 Ladrillos grises 3.5 10 35 11 1,100 Block 30 3 Megablock 4 12 1,110 40 13 1,120 Sillar rosa 10 100 14 Sillar gris 1,130 110 11 15 1,140 Cantera blanca 200 20 121 16 1,150 Cantera gris 1,210 142 17 1,160 Cantera rosa 1,420 18 1,170 Cantera amarilla 230 23 19 1,180 Recubrimiento P1001 200 20 Recubrimiento P1010 22 20 1,190 220 21 1,200 Recubrimiento P1019 240 24

Selección

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

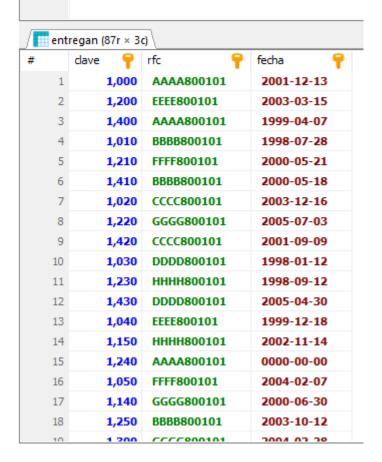
SQL select * from materiales where clave=1000



Proyección

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

SQL select clave,rfc,fecha from entregan



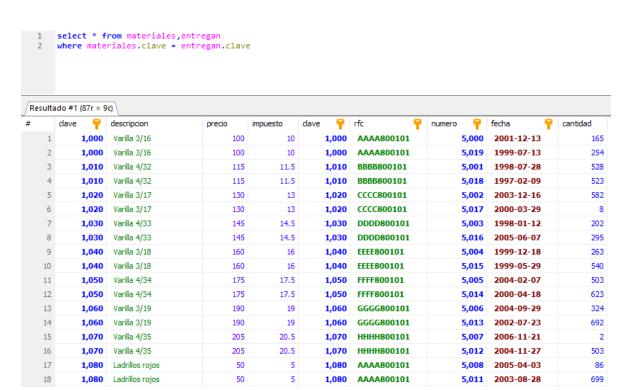
Reunión Natural

Algebra relacional. entregan JN materiales

SQL

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

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



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

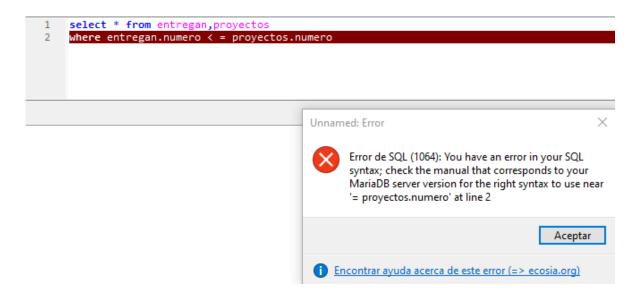
Reunión con criterio específico

Algebra relacional.

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

SQL

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



Unión (se ilustra junto con selección)

Algebra relacional.

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

SOL

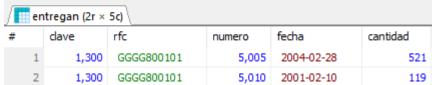
(select * from entregan where clave=1450)

union

(select * from entregan where clave=1300)

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

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



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

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



Intersección (se ilustra junto con selección y proyección)

Algebra relacional.

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)

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

# entregan (1r × 1c)

# dave
1 1,010
```

Diferencia (se ilustra con selección)

Algebra relacional. entregan - SL{clave=1000}(entregan)

SQL

(select * from entregan)

minus

(select * from entregan where clave=1000)

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

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

entregan (85r × 5c)								
#	dave	rfc	numero	fecha	cantidad			
	1 1,010	BBBB800101	5,001	1998-07-28	528			
	2 1,010	BBBB800101	5,018	1997-02-09	523			
	3 1,020	CCCC800101	5,002	2003-12-16	582			
	4 1,020	CCCC800101	5,017	2000-03-29	8			
	5 1,030	DDDD800101	5,003	1998-01-12	202			
	5 1,030	DDDD800101	5,016	2005-06-07	295			
	7 1,040	EEEE800101	5,004	1999-12-18	263			
	B 1,040	EEEE800101	5,015	1999-05-29	540			
	9 1,050	FFFF800101	5,005	2004-02-07	503			
1	1,050	FFFF800101	5,014	2000-04-18	623			
1	1 1,060	GGGG800101	5,006	2004-09-29	324			
1	2 1,060	GGGG800101	5,013	2002-07-23	692			
1	3 1,070	HHHH800101	5,007	2006-11-21	2			
1	4 1,070	HHHH800101	5,012	2004-11-27	503			
1	5 1,080	AAAA800101	5,008	2005-04-03	86			
1	5 1,080	AAAA800101	5,011	2003-08-28	699			
1	7 1,090	BBBB800101	5,009	1997-03-13	73			
1	1,090	BBBB800101	5,010	1998-11-17	421			
- 1	1 100	CCCC000101	E 000	2000 12 07	ACC			

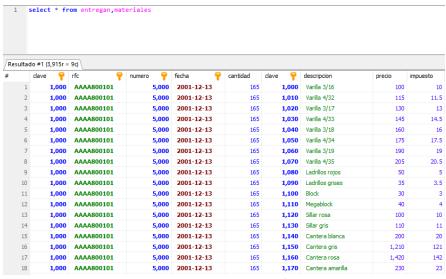
Producto cartesiano

Algebra relacional. entregan X materiales

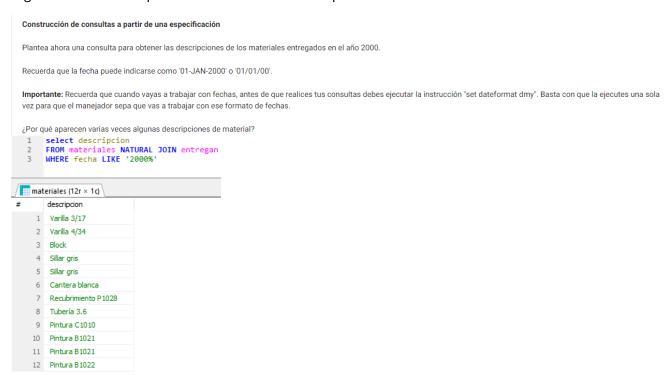
SQL

select * from entregan,materiales

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



Se entrega el producto de cruzar cada registro de ambas tablas, por lo tanto, el total de registros es una multiplicación de la cantidad de tuplas de cada tabla.



Algunas descripciones aparecen más de una vez porque se entregaron más de una vez durante el año 2000.

Uso del calificador distinct

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

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

¿Qué resultado obtienes en esta ocasión?



Ya no se repiten los registros que antes se repetían.

Ordenamientos.

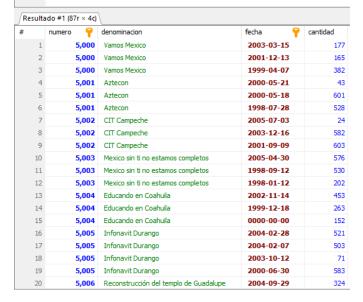
Si al final de una sentencia select se agrega la cláusula

order by campo [desc] [,campo [desc] ...]

donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.

Obtén los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto, presentando las fechas de la más reciente a la más antiqua





Uso de expresiones.

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo en una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

- + Suma
- Resta
- * Producto

/ División

Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre"). Para SQL Server también pueden utilizarse comillas simples.

Operadores de cadena

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.

SELECT * FROM productos where Descripcion LIKE 'Si%'

¿Qué resultado obtienes? Explica que hace el símbolo '%'. ¿Qué sucede si la consulta fuera : LIKE 'Si' ? ¿Qué resultado obtienes? Explica a qué se debe este comportamiento.



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

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

Sin embargo, tenemos otros operadores como [], [1] y _.

[] - Busca coincidencia dentro de un intervalo o conjunto dado. Estos caracteres se pueden utilizar para buscar coincidencias de patrones como sucede con LIKE.

[1] - En contra parte, este operador coincide con cualquier caracter 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.

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

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';

Operadores Lógicos.

Los operadores lógicos comprueban la verdad de una condición, al igual que los operadores de comparación, devuelven un tipo de dato booleano (True, false o unknown).

ALL Es un operador que compara un valor numérico con un conjunto de valores representados por un subquery. La condición es verdadera cuando todo el conjunto cumple la condición.

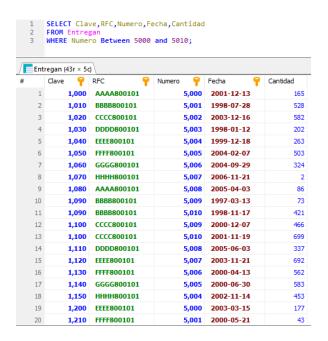
ANY o SOME Es un operador que compara un valor numérico con un conjunto de valores. La condición es verdadera cuando al menos un dato del conjunto cumple la condición.

La sintaxis para ambos es: valor_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?



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

```
EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.
 SELECT RFC, Cantidad, Fecha, Numero
 FROM [Entregan]
 WHERE [Numero] Between 5000 and 5010 AND
 Exists (SELECT [RFC]
 FROM [Proveedores]
 WHERE RazonSocial LIKE 'La%' and [Entregan].[RFC] = [Proveedores].[RFC] )
 ¿Qué hace la consulta?
 ¿Qué función tiene el paréntesis () después de EXISTS?
      SELECT RFC, Cantidad, Fecha, Numero
     WHERE Numero Between 5000 and 5010 AND
     Exists ( SELECT RFC
     WHERE RazonSocial LIKE 'La%' and Entregan.RFC = Proveedores.RFC )
Entregan (12r × 4c)
      RFC
                      Cantidad
   1 AAAA800101
                            165 2001-12-13
                                                      5,000
                                                      5,002
   2 CCCC800101
                            582 2003-12-16
                                                      5,008
   3 AAAA800101
                             86 2005-04-03
   4 CCCC800101
                            466 2000-12-07
                                                      5,009
   5 CCCC800101
                            699 2001-11-19
                                                      5,010
   6 AAAA800101
                            152 0000-00-00
                                                      5,004
      CCCC800101
                            460 2001-04-09
                                                      5,006
   8 CCCC800101
                            631 2001-07-28
                                                      5,009
   9 AAAA800101
                            382 1999-04-07
                                                      5,000
  10 AAAA800101
                            116 2005-04-21
                                                      5.010
  11 CCCC800101
                            603 2001-09-09
                                                      5.002
  12 CCCC800101
                            278 1999-05-05
                                                      5,008
```

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

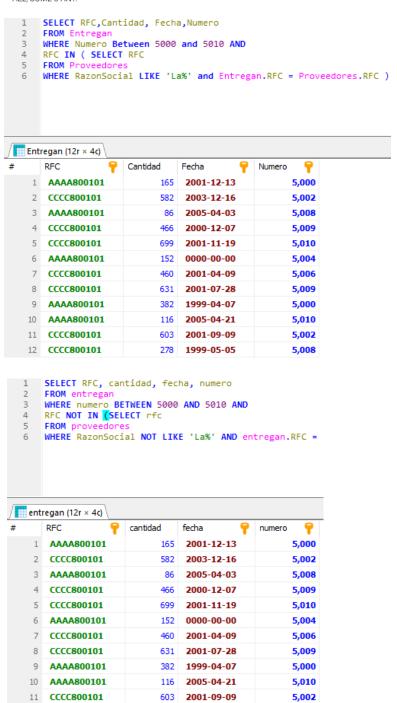
El paréntesis después de exists encapsula a la subconsulta sobre la que se ejecutará la función para determinar lo que se mostrará en la consulta original.

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.

12 CCCC800101

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.



278 1999-05-05

5,008

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.

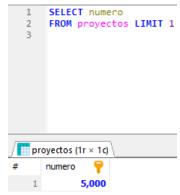
¿Qué hace la siguiente sentencia? Explica por qué.

SELECT TOP 2 * FROM Proyectos



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

SELECT TOP Numero FROM Proyectos



Creación de vistas

La sentencia:

Create view nombrevista (nombrecolumna 1, nombrecolumna 2,..., nombrecolumna 3) as select...

Permite definir una vista. Una vista puede pensarse como una consulta etiquetada con un nombre, ya que en realidad al referirnos a una vista el DBMS realmente ejecuta la consulta asociada a ella, pero por la cerradura del álgebra relacional, una consulta puede ser vista como una nueva relación o tabla, por lo que es perfectamente válido emitir la sentencia:

select * from nombrevista

¡Como si nombrevista fuera una tabla!

Comprueba lo anterior, creando vistas para cinco de las consultas que planteaste anteriormente en la práctica. Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

```
CREATE VIEW vistal(clave,descripcion)
AS SELECT clave, descripcion
FROM materiales NATURAL JOIN entregan
WHERE fecha LIKE '2000%';
       SELECT *
        FROM vistal
wista1 (12r × 2c)
                  descripcion
        dave
              1,020 Varilla 3/17
              1,050 Varilla 4/34
              1.100 Block
              1,130 Sillar gris
              1,130 Sillar gris
              1.140 Cantera blanca
             1,210 Recubrimiento P1028
              1.310 Tubería 3.6
    9
              1,360 Pintura C1010
   11
               1,410 Pintura B1021
               1,430 Pintura B1022
   12
```

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".

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

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

El Total entregado por cada material en el año 2000.

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

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

1,030 Varilla 4/33

1,430 Pintura B1022

1,230 Cemento

1

3

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

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).

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).

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.

Reto: Usa solo el operador NOT IN en la consulta anterior (No es parte de la entrega).

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

```
1 SELECT clave, descripcion
2 FROM materiales NATURAL JOIN entregan NATURAL JOIN proyectos
3 WHERE denominacion = "México sin ti no estamos completos"

| Mexico sin ti no estamos completos | Mexi
```

```
SELECT clave, descripcion
      FROM materiales NATURAL JOIN entregan NATURAL JOIN proveedores WHERE razonsocial = "Acme tools"
materiales (0r × 2c)
    dave
            descripcion
  1 SELECT rfc
      FROM entregan NATURAL JOIN proveedores
  3
       WHERE YEAR(fecha) = 2000
      GROUP BY rfc
  4
      HAVING AVG(cantidad)>=300
entregan (3r × 1c)
    1 BBBB800101
    2 FFFF800101
   3 GGGG800101
      SELECT descripcion, SUM(cantidad)
FROM entregan NATURAL JOIN materiales
      WHERE YEAR(fecha) = 2000
      GROUP BY clave
materiales (11r × 2c)
      descripcion
                             SUM(cantidad)
    1 Varilla 3/17
                                           8
    2 Varilla 4/34
                                         623
    3 Block
                                         466
    4 Sillar gris
    5 Cantera blanca
                                         583
    6 Recubrimiento P1028
                                          43
    7 Tubería 3.6
                                          72
    8 Pintura C1010
                                         265
    9 Pintura B1021
                                         107
   10 Pintura B1021
                                         601
  11 Pintura B1022
                                          13
```

```
1
       SELECT clave
       FROM entregan NATURAL JOIN materiales
       WHERE YEAR(fecha) = 2001
       GROUP BY clave
  4
       HAVING SUM(cantidad)
       ORDER BY cantidad DESC
       LIMIT 1
entregan (1r × 1c)
      dave
          1,100
1
       SELECT clave
  1
       FROM entregan NATURAL JOIN materiales
       WHERE YEAR(fecha) = 2001
       GROUP BY clave
  4
       HAVING SUM(cantidad)
       ORDER BY cantidad DESC
       LIMIT 1
 entregan (1r × 1c)
       dave
           1,100
      SELECT denominacion, SUM(precio*cantidad)
      FROM materiales NATURAL JOIN entregan NATURAL JOIN proyectos
      GROUP BY numero
proyectos (20r × 2c)
      denominacion
                                          SUM(precio*cantidad)
    1 Vamos Mexico
                                                      106,730
    2 Aztecon
                                                      146,595
    3 CIT Campeche
                                                       157,755
    4 Mexico sin ti no estamos completos
                                                      260,290
    5 Educando en Coahuila
                                                      620,610
    6 Infonavit Durango
                                                      321,135
    7 Reconstrucción del templo de Guadalupe
                                                      220,580
    8 Construcción de plaza Magnolias
                                                       122,969
    9 Televisa en acción
                                                       99,848
   10 Disco Atlantic
                                                      158,100
   11 Construcción de Hospital Infantil
                                                      144,295
   12 Remodelación de aulas del IPP
                                                      225,835
   13 Restauración de instalaciones del CEA
                                                      846,380
   14 Reparación de la plaza Sonora
                                                       527,485
   15 Remodelación de Soriana
                                                       284,450
```

```
SELECT denominacion, rfc, razonsocial
FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyectos
WHERE denominacion = "Televisa en acción" AND rfc NOT IN (SELECT rfc
 1
  2
  3
                                                                                                     FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyectos
  4
   5
                                                                                                     WHERE denominacion = "Educando en Coahuila")
Resultado #1 (3r × 3c)
        denominacion
                                                     razonsocial
    1 Televisa en acción
                                   CCCC800101
                                                           La Ferre
    2 Televisa en acción
                                    DDDD800101
                                                           Cecoferre
                                    DDDD800101
    3 Televisa en acción
                                                           Cecoferre
         SELECT SUM(precio*cantidad), descripcion
FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyectos NATURAL JOIN materiales
WHERE denominacion = "Televisa en acción" AND rfc IN (SELECT rfc
   2
                                                                                                    FROM proveedores NATURAL JOIN entregan NATURAL JOIN proyectos WHERE denominacion = "Educando en Coahuila")
materiales (1r × 2c)
        SUM(precio*cantidad)
#
                                     descripcion
                            7,938 Tepetate
```

1	SELECT denominacion, descripcion, COUNT(cantidad), precio*cantidad
2	FROM entregan NATURAL JOIN proyectos NATURAL JOIN materiales
3	GROUP BY denominacion, descripcion
4	

Resulta	Resultado #1 (87r × 4c)							
#	denominacion	descripcion	COUNT(cantidad)	precio*cantidad				
1	Ampliación de la carretera a la huasteca	Cantera rosa	1	230,040				
2	Ampliación de la carretera a la huasteca	Pintura C1010	1	45,500				
3	Ampliación de la carretera a la huasteca	Recubrimiento P1010	1	78,320				
4	Ampliación de la carretera a la huasteca	Tubería 4.5	1	168,500				
5	Ampliación de la carretera a la huasteca	Varilla 4/33	1	42,775				
6	Aztecon	Pintura B1021	1	75,125				
7	Aztecon	Recubrimiento P1028	1	10,750				
8	Aztecon	Varilla 4/32	1	60,720				
9	CIT Campeche	Pintura C1012	1	75,375				
10	CIT Campeche	Recubrimiento P1037	1	6,720				
11	CIT Campeche	Varilla 3/17	1	75,660				
12	CIT Yucatan	Cantera gris	1	554,180				
13	CIT Yucatan	Recubrimiento P1019	1	140,400				
14	CIT Yucatan	Tubería 3.8	1	70,720				
15	CIT Yucatan	Varilla 3/18	1	86,400				