


Consulta de un tabla completa


```
select * from materiales
```

materiales (45r × 4c)				
#	clave 	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

Comentario: En lo subrayado amarillo tiene la notación numero de renglones y número de columnas




Selección

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

materiales (1r × 4c)				
#	clave 	descripcion	precio	impuesto
1	1,000	Varilla 3/16	100	10

Proyección

```
select clave, rfc, fecha from entregan
```

entregan (87r × 3c)			
#	clave 	rfc 	fecha 
1	1,000	AAAA800101	2001-12-13
2	1,000	AAAA800101	1999-07-13
3	1,010	BBBB800101	1998-07-28
4	1,010	BBBB800101	1997-02-09

Reunión Natural

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

entregan (87r × 3c)			
#	clave	rfc	fecha
1	1,000	AAAA800101	2001-12-13
2	1,000	AAAA800101	1999-07-13
3	1,010	BBBB800101	1998-07-28
4	1,010	BBBB800101	1997-02-09

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

No porque es un join y solo mostrará un registro que cumpla la igualdad en la otra tabla.

Reunión con criterio específico

```
select * from entregan,proyectos
WHERE entregan.numero <= proyectos.numero
```

Resultado #1 (836r × 7c)							
#	clave	rfc	numero	fecha	cantidad	numero	denominacion
1	1,000	AAAA800101	5,000	2001-12-13	165	5,000	Vamos Mexico
2	1,200	EEEE800101	5,000	2003-03-15	177	5,000	Vamos Mexico
3	1,400	AAAA800101	5,000	1999-04-07	382	5,000	Vamos Mexico

Unión (se ilustra junto con selección)

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

entregan (2r × 5c)					
#	clave	rfc	numero	fecha	cantidad
1	1,300	GGGG800101	5,005	2004-02-28	521
2	1,300	GGGG800101	5,010	2001-02-10	119

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

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

```
SELECT *
FROM entregan
WHERE clave = 1450 OR clave = 1300;
```

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

entregan (2r × 5c)						
#	clave	rfc	numero	fecha	cantidad	
1	1,300	GGGG800101	5,005	2004-02-28	521	
2	1,300	GGGG800101	5,010	2001-02-10	119	

Intersección (se ilustra junto con selección y proyección)

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

entregan (1r × 1c)	
#	clave
1	1,010

Diferencia (se ilustra con selección)

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

entregan (85r × 5c)					
#	clave	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

Producto cartesiano

```
select * from entregan, materiales
```

Resultado #1 (3,915r × 9c)

#	clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	imp
1	1,000	AAAA800101	5,000	2001-12-13	165	1,000	Varilla 3/16	100	
2	1,000	AAAA800101	5,000	2001-12-13	165	1,010	Varilla 4/32	115	
3	1,000	AAAA800101	5,000	2001-12-13	165	1,020	Varilla 3/17	130	
4	1,000	AAAA800101	5,000	2001-12-13	165	1,030	Varilla 4/32	145	

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales? No se si se refiera la pregunta a que es la combinación posible de cada tupla.

Construcción de consultas a partir de una especificación

```
select materiales.descripcion, entregan.fecha
FROM Materiales
INNER JOIN Entregan ON Materiales.Clave = Entregan.Clave
WHERE YEAR(Entregan.Fecha) = 2000;
```

Materiales (12r × 2c)

#	descripcion	fecha
1	Varilla 3/17	2000-03-29
2	Varilla 4/34	2000-04-18
3	Block	2000-12-07
4	Sillar gris	2000-04-13
5	Sillar gris	2000-03-08

¿Por qué aparecen varias veces algunas descripciones de material? Porque es el mismo material entregado en un diferente espacio a lo largo del tiempo.

Uso del calificador distinct

¿Qué resultado obtienes en esta ocasión? Pues me sale lo mismo

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 Proyectos.Numero, Proyectos.Denominacion, entregan.fecha,
SUM(Entregan.Cantidad) AS Total_Cantidad
FROM Proyectos, Entregan
WHERE Proyectos.Numero = Entregan.Numero
GROUP BY Proyectos.Numero, Proyectos.Denominacion, entregan.fecha
ORDER BY Proyectos.Numero, entregan.fecha DESC;
```

Proyectos (87r × 4c)				
#	Numero	Denominacion	fecha	Total_Cantidad
1	5,000	Vamos Mexico	2003-03-15	177
2	5,000	Vamos Mexico	2001-12-13	165
3	5,000	Vamos Mexico	1999-04-07	382
4	5,001	Aztecon	2000-05-21	43
5	5,001	Aztecon	2000-05-18	601

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 Materiales
where Descripcion LIKE 'Si%'
```

¿Qué resultado obtienes?

Materiales (2r × 4c)				
#	clave	descripcion	precio	impuesto
1	1,120	Sillar rosa	100	10
2	1,130	Sillar gris	110	11

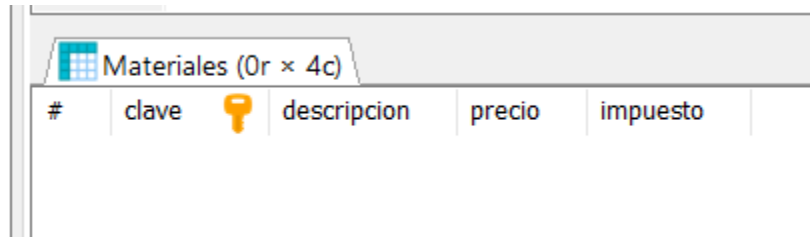
Explica que hace el símbolo '%'.

En este caso se esta ocupando como sufijo y estamos buscando las descripciones que empiecen con Si sin importar que este delante de ello.

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

```
SELECT *  
FROM Materiales  
where Descripcion LIKE 'Si'
```

¿Qué resultado obtienes?



#	clave	descripcion	precio	impuesto
		Si		

Explica a qué se debe este comportamiento.

Porque esta buscando la descripcion que literalmente solo tenga Si

Aquí según yo no tenemos Personas a menos que no lo haya visto, pero voy a darme a la tarea de investigar estos comandos.

Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:
SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;

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

¿Qué resultado obtienes de ejecutar el siguiente código?

Se declaran dos variables, después se reasignan y por ultimo se concatenan

¿Para qué sirve DECLARE? Declara variables y se debe de poner un @

¿Cuál es la función de @foo? Pues es una variable de tipo varchar con una longitud de 40 caracteres.

¿Que realiza el operador SET? Es asignar un nuevo valor a tal variable

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

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

SELECT RFC **FROM** Entregan **WHERE** RFC **LIKE** '[A-D]>'; Aquí se buscan los rfc's que comiencen desde A hasta D

SELECT RFC **FROM** Entregan **WHERE** RFC **LIKE** '[^A]>'; Busca cuales no coinciden con A pero no se porque no imprime nada si debería de no?

SELECT Numero **FROM** Entregan **WHERE** Numero **LIKE** '____6'; Este según es el buscar números que tengan 4 dígitos y el cuarto sea el número 6

Entregan (0r x 1c)		Entregan (0r x 1c)		Entregan (9r x 1c)	
#	Numero				
1	5,016				
2	5,006				
3	5,006				
4	5,016				
5	5,016				

Los operadores compuestos ejecutan una operación y establecen un valor.

+ = (Suma igual)

- = (Restar igual)

* = (Multiplicar igual)

/ = (Dividir igual)

% = (Módulo igual)

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;
```

Entregan (43r × 5c)						
#	Clave	RFC	Numero	Fecha	Cantidad	
1	1,000	AAAA800101	5,000	2001-12-13	165	
2	1,010	BBBB800101	5,001	1998-07-28	528	
3	1,020	CCCC800101	5,002	2003-12-16	582	
4	1,030	DDDD800101	5,003	1998-01-12	202	
5	1,040	EEEE800101	5,004	1999-12-10	262	

¿Cómo filtrarías rangos de fechas?

```
SELECT Clave, RFC, Numero, Fecha, Cantidad
FROM Entregan
WHERE Fecha BETWEEN '2000-01-01' AND '2005-12-31';
```

Entregan (54r × 5c)						
#	Clave	RFC	Numero	Fecha	Cantidad	
1	1,000	AAAA800101	5,000	2001-12-13	165	
2	1,020	CCCC800101	5,002	2003-12-16	582	
3	1,020	CCCC800101	5,017	2000-03-29	8	
4	1,030	DDDD800101	5,016	2005-06-07	295	
5	1,050	EEEE800101	5,005	2004-02-07	502	

```
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 )
```

Entregan (12r × 4c)				
#	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165	2001-12-13	5,000
2	AAAA800101	86	2005-04-03	5,008
3	AAAA800101	152	0000-00-00	5,004
4	AAAA800101	382	1999-04-07	5,000
5	AAAA800101	116	2005-04-21	5,010

¿Qué hace la consulta? Primero busca la razón social que empiece con La y después selecciona las que su número este entre 5000 y 5010

¿Qué función tiene el paréntesis () después de EXISTS? Hacer la 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 RFC IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
);
```

Entregan (12r x 4c)					
#	RFC	Cantidad	Fecha	Numero	
1	AAAA800101	165	2001-12-13	5,000	
2	AAAA800101	86	2005-04-03	5,008	
3	AAAA800101	152	0000-00-00	5,004	
4	AAAA800101	382	1999-04-07	5,000	
5	AAAA800101	116	2005-04-21	5,010	

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 NOT IN (
    SELECT RFC
    FROM Proveedores
    WHERE RazonSocial LIKE 'La%'
);
```

Selecciona las entregas donde la cantidad es mayor que al menos una de las cantidades entregadas por los proveedores cuya razón social comienza con "La".

```
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Cantidad > SOME (
    SELECT Cantidad
    FROM Entregan
    WHERE RFC IN (
        SELECT RFC
        FROM Proveedores
        WHERE RazonSocial LIKE 'La%'
    )
);
```

Entregan (85r × 4c)				
#	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165	2001-12-13	5,000
2	AAAA800101	254	1999-07-13	5,019
3	BBBB800101	528	1998-07-28	5,001
4	BBBB800101	523	1997-02-09	5,018
5	CCCC800101	522	2002-12-15	5,002

SELECT Limit 2 * FROM Proyectos

```
SELECT * FROM proyectos
LIMIT 2;
```

proyectos (2r × 2c)		
#	numero	denominacion
1	5,000	Vamos Mexico
2	5,001	Aztecon

¿Qué hace la siguiente sentencia? Explica por qué.

Se imprimen las primeras dos tuplas porque el operador imprime esos dos primeros que encuentra y pues son los dos primeros porque no hay una condición.

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

#	clave	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	175	17.5
7	1,060	Varilla 3/19	190	19

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

#	clave	descripcion	precio	impuesto	PorcentajeImpuesto
1	1000	Varilla 3/16	100	10	2.0
2	1010	Varilla 4/32	115	11.5	2.02
3	1020	Varilla 3/17	130	13	2.04
4	1030	Varilla 4/33	145	14.5	2.06
5	1040	Varilla 3/18	160	16	2.08
6	1050	Varilla 4/34	175	17.5	2.1
7	1060	Varilla 3/19	190	19	2.12
8	1070	Varilla 4/35	205	20.5	2.14

¿Qué consulta usarías para obtener el importe de las entregas es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega y el precio del material y el impuesto asignado?

```
SELECT e.Clave,
       e.Cantidad * m.Precio * (1 + (m.Impuesto / 100)) AS Importe_Total
FROM Entregan e
JOIN Materiales m ON e.Clave = m.Clave;
```

Entregan (87r x 2c)		
#	Clave	Importe_Total
1	1000	18150
2	1000	27940.000000000004
3	1010	67702.8
4	1010	67061.675
5	1020	85405.70000000000

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

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

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

materiales (3r × 2c)		
#	clave	descripcion
1	1030	Varilla 4/33
2	1230	Cemento
3	1430	Pintura B1022

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

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

materiales (0r × 2c)		
#	clave	descripcion

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

```
SELECT Pr.rfc, E.fecha, AVG(E.cantidad) AS Promedio
FROM entregan E, proveedores Pr
WHERE E.rfc = Pr.rfc
AND YEAR(E.fecha) = 2000
GROUP BY Pr.rfc
HAVING AVG(E.cantidad) >= 300
```

entregan (3r × 3c)			
#	rfc	fecha	Promedio
1	BBBB800101	2000-05-18	601.0
2	FFFF800101	2000-04-18	322.75
3	GGGG800101	2000-06-30	583.0

El Total entregado por cada material en el año 2000.

```
SELECT M.descripcion, SUM(E.cantidad) AS Entregado
FROM entregan E, materiales M
WHERE E.clave = M.clave
AND YEAR(E.fecha) = 2000
GROUP BY M.descripcion
HAVING AVG(E.cantidad)
ORDER BY Entregado DESC
```

entregan (10r × 2c)		
#	descripcion	Entregado
1	Pintura B1021	708
2	Sillar gris	625
3	Varilla 4/34	623
4	Cantera blanca	583
5	Block	466
6	Pintura C1010	265
7	Tubería 3.6	72
8	Recubrimiento P1028	43
9	Pintura B1022	13
10	Varilla 3/17	8

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

```
CREATE VIEW Material_Ventas AS
SELECT clave, SUM(Cantidad) AS Total_vendido
FROM entregan
WHERE YEAR(Fecha) = 2001
GROUP BY clave;
```

```
SELECT clave
FROM Material_ventas
ORDER BY Total_vendido DESC
LIMIT 1;
```

Material_ventas (1r × 1c)	
#	clave
1	1260

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

```
SELECT Descripcion
FROM Materiales
where Descripcion LIKE '%ub%'
```

Materiales (12r × 1c)	
#	Descripcion
1	Recubrimiento P1001
2	Recubrimiento P1010
3	Recubrimiento P1019
4	Recubrimiento P1028
5	Recubrimiento P1037
6	Tubería 3.5
7	Tubería 4.3
8	Tubería 3.6
9	Tubería 4.4
10	Tubería 3.7
11	Tubería 4.5
12	Tubería 3.8

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

```
SELECT Denominacion, SUM(E.cantidad * M.precio) AS total_a_pagar
FROM materiales M, proyectos Pr, entregan E
WHERE M.clave = E.clave
AND E.numero = Pr.numero
GROUP BY Pr.denominacion
ORDER BY total_a_pagar DESC
```

materiales (20r × 2c)		
#	Denominacion	total_a_pagar
1	CIT Yucatan	851700
2	Restauración de instalaciones del CEA	846380
3	Educando en Coahuila	620610
4	Ampliación de la carretera a la huasteca	565135
5	Reparación de la plaza Sonora	527485

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 vista_televisa AS
SELECT Numero, Denominacion
FROM Proyectos
WHERE Denominacion = 'Televisa en acción';
```

```
CREATE VIEW vista_coahuila AS
SELECT Numero, Denominacion
FROM Proyectos
WHERE Denominacion = 'Educando en Coahuila';
```

```
SELECT DISTINCT pr.Denominacion, e.RFC, p.RazonSocial
FROM Entregan e, Proveedores p, Proyectos pr
WHERE e.Numero IN (SELECT Numero FROM vista_televisa)
AND e.Numero NOT IN (SELECT Numero FROM vista_coahuila)
AND e.RFC = p.RFC
AND e.Numero = pr.Numero;
```

Entregan (4r × 3c)			
#	Denominacion	RFC	RazonSocial
1	Televisa en acción	AAAA800101	La fragua
2	Televisa en acción	DDDD800101	Cecoferre
3	Televisa en acción	EEEE800101	Alvin
4	Televisa en acción	CCCC800101	La Ferre

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 pr.Denominacion, e.RFC, p.RazonSocial
FROM Entregan e, Proveedores p, Proyectos pr
WHERE e.Numero = pr.Numero
AND pr.Denominacion = 'Televisa en acción'
```

```
AND e.Numero NOT IN (SELECT Numero FROM Proyectos WHERE Denominacion =
'Educando en Coahuila')
AND e.RFC = p.RFC;
```

Entregan (4r × 3c)			
#	Denominacion	RFC	RazonSocial
1	Televisa en acción	AAAA800101	La fragua
2	Televisa en acción	CCCC800101	La Ferre
3	Televisa en acción	DDDD800101	Cecoferre
4	Televisa en acción	EEEE800101	Alvin

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 DISTINCT m.descripcion, m.precio, pr.denominacion
FROM Entregan e, Proveedores p, Proyectos pr, materiales m
WHERE e.Numero IN (SELECT Numero FROM vista_televisa)
AND m.clave = e.clave
AND e.RFC = p.RFC
AND e.Numero = pr.Numero;
```

```
SELECT DISTINCT m.descripcion, m.precio, pr.denominacion
FROM Entregan e, Proveedores p, Proyectos pr, materiales m
WHERE e.Numero IN (SELECT Numero FROM vista_coahuila)
AND m.clave = e.clave
AND e.RFC = p.RFC
AND e.Numero = pr.Numero;
```

Entregan (5r × 3c)		Entregan (3r × 3c)	
#	descripcion	precio	denominacion
1	Ladrillos rojos	50	Televisa en acción
2	Megablock	40	Televisa en acción
3	Tezontle	80	Televisa en acción
4	Tepetate	34	Televisa en acción
5	Pintura C1012	125	Televisa en acción

Entregan (5r × 3c)			
Entregan (3r × 3c)			
#	descripcion	precio	denominacion
1	Varilla 3/18	160	Educando en Coahuila
2	Cantera gris	1210	Educando en Coahuila
3	Arena	200	Educando en Coahuila