

## **Laboratorio 20: Consultas en SQL**

Juan Enrique Ayala Zapata  
A01711235

### 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 se ha entregado ¿Aparecería en el resultado de esta consulta?

```
1 • select * from materiales,entregan
2   where materiales.clave = entregan.clave;
3
```

Result Grid     Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 									
	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
	1010	Varilla 4/32	115	11.5	1010	BBBB800101	5001	1998-07-28	528
	1010	Varilla 4/32	115	11.5	1010	BBBB800101	5018	1997-02-09	523
	1020	Varilla 3/17	130	13	1020	CCCC800101	5002	2003-12-16	582
	1020	Varilla 3/17	130	13	1020	CCCC800101	5017	2000-03-29	8
	1030	Varilla 4/33	145	14.5	1030	DDDD800101	5003	1998-01-12	202
	1030	Varilla 4/33	145	14.5	1030	DDDD800101	5016	2005-06-07	295
	1040	Varilla 3/18	160	16	1040	EEEE800101	5004	1999-12-18	263
	1040	Varilla 3/18	160	16	1040	EEEE800101	5015	1999-05-29	540
	1050	Varilla 4/34	175	17.5	1050	FFFF800101	5005	2004-02-07	503
	1050	Varilla 4/34	175	17.5	1050	FFFF800101	5014	2000-04-18	623
	1060	Varilla 3/19	190	19	1060	GGGG800101	5006	2004-09-29	324
	1060	Varilla 3/19	190	19	1060	GGGG800101	5013	2002-07-23	692
	1070	Varilla 4/35	205	20.5	1070	HHHH800101	5007	2006-11-21	2
	1070	Varilla 4/35	205	20.5	1070	HHHH800101	5012	2004-11-27	503

- No, solo aparecen todos los materiales que han sido entregados

### Reunión con criterio específico

Algebra relacional.

entregan  $\Join$  {entregan.numero  $\leq$  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)  $\cup$  SL{clave=1300}(entregan)

SQL


```
(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)
4   |
```

Result Grid					
		Filter Rows:			Export:  Wrap
	clave	rfc	numero	fecha	cantidad
▶	1300	GGGG800101	5005	2004-02-28	521
	1300	GGGG800101	5010	2001-02-10	119

```
1 • select * from entregan where clave=1450 or clave=1300;
2
```

Result Grid					
Filter Rows:					
	clave	rfc	numero	fecha	cantidad
▶	1300	GGGG800101	5005	2004-02-28	521
	1300	GGGG800101	5010	2001-02-10	119
*	NULL	NULL	NULL	NULL	NULL

### 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 *
2 FROM entregan
3 WHERE clave NOT IN (SELECT clave FROM entregan WHERE clave = 1000);
4
```

Result Grid					
Filter Rows:					
	clave	rfc	numero	fecha	cantidad
▶	1010	BBBB800101	5001	1998-07-28	528
	1010	BBBB800101	5018	1997-02-09	523
	1020	CCCC800101	5002	2003-12-16	582
	1020	CCCC800101	5017	2000-03-29	8
	1030	DDDD800101	5003	1998-01-12	202
	1030	DDDD800101	5016	2005-06-07	295
	1040	EEEE800101	5004	1999-12-18	263
	1040	EEEE800101	5015	1999-05-29	540
	1050	FFFF800101	5005	2004-02-07	503
	1050	FFFF800101	5014	2000-04-18	623
	1060	GGGG800101	5006	2004-09-29	324
	1060	GGGG800101	5013	2002-07-22	502

### 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?

- Si solo se selecciona de entregan:

3 • `select * from entregan;`

Result Grid    Filter Rows: <input type="text"/> Edit:					
	clave	rfc	numero	fecha	cantidac
▶	1000	AAAA800101	5000	2001-12-13	165
	1000	AAAA800101	5019	1999-07-13	254
	1010	BBBB800101	5001	1998-07-28	528
	1010	BBBB800101	5018	1997-02-09	523
	1020	CCCC800101	5002	2003-12-16	582
	1020	CCCC800101	5017	2000-03-29	8
	1030	DDDD800101	5003	1998-01-12	202
	1030	DDDD800101	5016	2005-06-07	295
	1040	EEEE800101	5004	1999-12-18	263
	1040	EEEE800101	5015	1999-05-29	540
	1050	FFFF800101	5005	2004-02-07	503
	1050	FFFF800101	5014	2000-04-10	622




- Si solo se seleccionan de materiales:

3 • `select * from materiales;`

Result Grid    Filter Rows: <input type="text"/>				
	clave	descripcion	precio	impuesto
▶	1000	Varilla 3/16	100	10
	1010	Varilla 4/32	115	11.5
	1020	Varilla 3/17	130	13
	1030	Varilla 4/33	145	14.5
	1040	Varilla 3/18	160	16
	1050	Varilla 4/34	175	17.5
	1060	Varilla 3/19	190	19
	1070	Varilla 4/35	205	20.5
	1080	Ladrillos rojos	50	5
	1090	Ladrillos grises	35	3.5
	1100	Block	30	3
	1110	Menablock	40	4

- Si se seleccionan ambas:

```
1 • select * from entregan,materiales;
2
3
```

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content:  Fetch rows:									
	clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
▶	1000	AAAA800101	5000	2001-12-13	165	2000	Jabón	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1430	Pintura B1022	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1420	Pintura C1012	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1410	Pintura B1021	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1400	Pintura C1011	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1390	Pintura B1021	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1380	Pintura C1011	725	72.5
	1000	AAAA800101	5000	2001-12-13	165	1370	Pintura B1020	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1360	Pintura C1010	125	12.5
	1000	AAAA800101	5000	2001-12-13	165	1350	Tubería 3.8	260	26
	1000	AAAA800101	5000	2001-12-13	165	1340	Tubería 4.5	250	25
	1000	AAAA800101	5000	2001-12-13	165	1330	Tubería 3.7	240	24




### 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.

```
1 • SELECT DATE_FORMAT(NOW(), '%Y-%m-%d') AS formatted_date;
2
3 • SELECT m.descripcion
4 FROM materiales m, entregan e
5 WHERE m.clave = e.clave AND YEAR(e.fecha) = 2000;
6
7
```

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content: 	
	descripcion
▶	Varilla 3/17
	Varilla 4/34
	Block
	Sillar gris
	Sillar gris
	Cantera blanca
	Recubrimiento P1028
	Tubería 3.6
	Pintura C1010
	Pintura B1021

¿Por qué aparecen varias veces algunas descripciones de material?

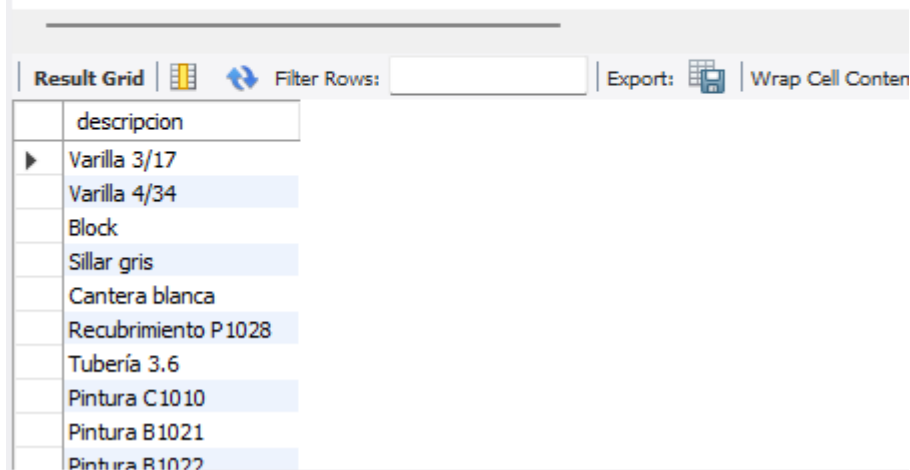

- Lo que mantiene la unicidad de los materiales no son las descripciones, son las claves. Como seleccionamos las descripciones, se muestran múltiples que pueden ser iguales.

### 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?

```
3 • SELECT distinct(m.descripcion)
4     FROM materiales m, entregan e
5     WHERE m.clave = e.clave AND YEAR(e.fecha) = 2000;
6
7
```

Result Grid |  Filter Rows:  | Export:  Wrap Cell Content

	descripcion
▶	Varilla 3/17
	Varilla 4/34
	Block
	Sillar gris
	Cantera blanca
	Recubrimiento P1028
	Tubería 3.6
	Pintura C1010
	Pintura B1021
	Pintura R1022

- Ahora no se repiten los registros

### 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 antigua.

```

1 • SELECT p.Numero, p.Denominacion, e.Fecha, e.Cantidad
2 FROM Proyectos p
3 JOIN Entregan e ON p.Numero = e.Numero
4 ORDER BY p.Numero ASC, e.Fecha DESC;
5

```

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
	5001	Aztecon	1998-07-28	528
	5002	CIT Campeche	2005-07-03	24
	5002	CIT Campeche	2003-12-16	582
	5002	CIT Campeche	2001-09-09	603
	5003	Mexico sin tino estamos completos	2005-04-30	576

### 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?

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

Result Grid

Filter Rows:

Edit:

Export/

	clave	descripcion	precio	impuesto
▶	1120	Sillar rosa	100	10
	1130	Sillar gris	110	11
*	NULL	NULL	NULL	NULL

Explica que hace el símbolo '%'.

- Es una forma de representar una secuencia de cero o más caracteres



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

```
1 • SELECT * FROM materiales where Descripcion LIKE 'Si'
```

Result Grid

Filter Rows:

Edit:

Export

	clave	descripcion	precio	impuesto
*	NULL	NULL	NULL	NULL

Explica a qué se debe este comportamiento.

- esto no utilizaría el operador % y buscaría una coincidencia exacta con el texto 'Si'

Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.

Su sintaxis es : Expresión + Expresión.

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?

```

1 • SELECT CONCAT(descripcion, ', ', denominacion) AS Nombre
2   FROM Materiales m
3   JOIN Proyectos p ON m.Clave = p.Numero
4   LIMIT 1;
5 • SET @foo = '¿Qué resultado';
6 • SET @bar = ' ¿¿¿???';
7 • SET @foo = CONCAT(@foo, ' obtienes?');
8 • SELECT CONCAT(@foo, @bar) AS ResultadoFinal;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ResultadoFinal			
¿Qué resultado obtienes? ¿¿¿???			

¿Para qué sirve DECLARE?

- Para declarar variables dentro de un bloque de código

¿Cuál es la función de @foo?

- Es el nombre de una variable. La variable se llama 'foo'.

¿Que realiza el operador SET?

- En este contexto, es para asignar 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]%' ;**

```

1 • SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%' ;
2   -- SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%' ;
3   -- SELECT Numero FROM Entregan WHERE Numero LIKE '___6' ;
4
5

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
RFC			

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

```

1  -- SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%' ;
2  • SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%' ;
3  -- SELECT Numero FROM Entregan WHERE Numero LIKE '___6' ;
4
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [I/A](#)

RFC
-----

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

```

3  • SELECT Numero FROM Entregan WHERE Numero LIKE '___6' ;
4
5

```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

Numero
5006
5006
5006
5006
5016
5016
5016
5016
5016
5016

**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?

- WHERE Fecha BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD';

**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?

```

1 • SELECT RFC,Cantidad, Fecha,Numero
2   FROM entregan
3   WHERE Numero Between 5000 and 5010 AND
4   EXISTS ( SELECT RFC
5             FROM Proveedores
6             WHERE RazonSocial LIKE 'La%' and Entregan.RFC = Proveedores.RFC )

```

Result Grid				
	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
	CCCC800101	582	2003-12-16	5002
	CCCC800101	466	2000-12-07	5009
	CCCC800101	699	2001-11-19	5010
	CCCC800101	460	2001-04-09	5006
	CCCC800101	631	2001-07-28	5009

- El paréntesis permite que se ejecute la subconsulta

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

```

1 • SELECT RFC, Cantidad, Fecha, Numero
2   FROM entregan
3   WHERE Numero BETWEEN 5000 AND 5010
4   AND RFC IN (
5       SELECT RFC
6       FROM Proveedores
7       WHERE RazonSocial LIKE 'La%'
8   )
9

```

Result Grid				
	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
	CCCC800101	582	2003-12-16	5002

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.

```

1 • SELECT RFC, Cantidad, Fecha, Numero
2   FROM entregan
3  WHERE Numero BETWEEN 5000 AND 5010
4  AND RFC NOT IN (
5      SELECT RFC
6      FROM proveedores
7      WHERE RazonSocial LIKE 'La%'
8  )
9

```

Result Grid				
Filter Rows:				
Export:				
	RFC	Cantidad	Fecha	Numero
▶	BBBB800101	528	1998-07-28	5001
	DDDD800101	202	1998-01-12	5003
	EEEE800101	263	1999-12-18	5004
	FFFF800101	503	2004-02-07	5005
	GGGG800101	324	2004-09-29	5006
	HHHH800101	2	2006-11-21	5007
entregan 47 x				

¿Qué hace la siguiente sentencia? Explica por qué.

SELECT TOP 2 \* FROM Proyectos

```

1 SELECT * FROM Proyectos
2 LIMIT 2;
3

```

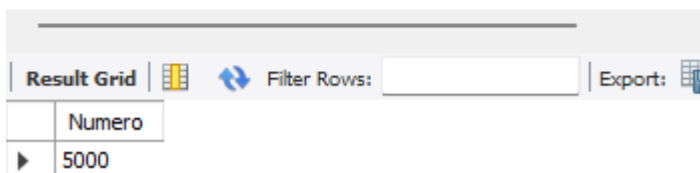
Result Grid	
Filter Rows:	
Export:	
numero	denominacion
▶ 5000	Vamos Mexico
5001	Aztecon
* NULL	NULL

- La consulta selecciona las primeras 2 filas (o registros) de la tabla Proyectos, devolviendo todas las columnas de esas filas. Esa sintaxis no existe en MySQL.

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

SELECT TOP Numero FROM Proyectos

```
1 • SELECT Numero FROM Proyectos LIMIT 1;  
2
```



Numero
5000

- Selecciona un solo valor de la columna Numero de la tabla 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.

```

3 • ALTER TABLE Materiales ADD PorcentajeImpuesto NUMERIC(6,2);
4 • UPDATE Materiales SET PorcentajeImpuesto = 2 * Clave / 1000;
5
6
7 • SELECT * FROM Materiales

```

Result Grid					
		Filter Rows:			
		Edit:			
		Export/Import:			
	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
	1050	Varilla 4/34	175	17.5	2.10
	1060	Varilla 3/19	190	19	2.12
	1070	Varilla 4/35	205	20.5	2.14
	1080	Ladrillos rojos	50	5	2.16
	1090	Ladrillos grises	35	3.5	2.18
	1100	Block	30	3	2.20
	1110	Menablock	40	4	2.22

¿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?

```

12      E.rfc,
13      E.numero,
14      E.fecha,
15      E.cantidad,
16      M.precio,
17      M.porcentajeImpuesto,
18      E.cantidad * M.precio * (1 + M.PorcentajeImpuesto) AS ImporteEntrega
19 FROM Entregan E
20 JOIN Materiales M ON E.clave = M.clave;

```

Result Grid								
		Filter Rows:						
		Export:						
		Wrap Cell Content:						
	clave	rfc	numero	fecha	cantidad	precio	porcentajeImpuesto	ImporteEntrega
▶	1000	AAAA800101	5000	2001-12-13	165	100	2.00	49500
	1000	AAAA800101	5019	1999-07-13	254	100	2.00	76200
	1010	BBBB800101	5001	1998-07-28	528	115	2.02	183374.4
	1010	BBBB800101	5018	1997-02-09	523	115	2.02	181637.9
	1020	CCCC800101	5002	2003-12-16	582	130	2.04	230006.4
	1020	CCCC800101	5017	2000-03-29	8	130	2.04	3161.6
	1030	DDDD800101	5003	1998-01-12	202	145	2.06	89627.40000000001
	1030	DDDD800101	5016	2005-06-07	295	145	2.06	130891.5

## Creación de vistas

La sentencia:

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:

```
select * from nombrevista
```

**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:**

```
1 • CREATE VIEW v_materiales_con_impuesto AS
2 SELECT
3     clave,
4     descripcion,
5     precio,
6     PorcentajeImpuesto,
7     precio * (1 + PorcentajeImpuesto) AS CostoConImpuesto
8 FROM Materiales;
9
```

Result Grid					
		Filter Rows:	Export:		Wrap Cell Content:
	clave	descripcion	precio	PorcentajeImpuesto	CostoConImpuesto
▶	1000	Varilla 3/16	100	2.00	300
	1010	Varilla 4/32	115	2.02	347.3
	1020	Varilla 3/17	130	2.04	395.2
	1030	Varilla 4/33	145	2.06	443.7
	1040	Varilla 3/18	160	2.08	492.8
	1050	Varilla 4/34	175	2.10	542.5
	1060	Varilla 3/19	190	2.12	592.8000000000001
	1070	Varilla 4/35	205	2.14	643.7



```

4      E.rfc,
5      E.numero,
6      E.fecha,
7      E.cantidad,
8      M.precio,
9      M.PorcentajeImpuesto,
10     E.cantidad * M.precio * (1 + M.PorcentajeImpuesto) AS ImporteEntrega
11 FROM Entregan E
12 JOIN Materiales M ON E.Clave = M.Clave;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:



	clave	rfc	numero	fecha	cantidad	precio	PorcentajeImpuesto	ImporteEntrega
▶	1000	AAAA800101	5000	2001-12-13	165	100	2.00	49500
	1000	AAAA800101	5019	1999-07-13	254	100	2.00	76200
	1010	BBBB800101	5001	1998-07-28	528	115	2.02	183374.4
	1010	BBBB800101	5018	1997-02-09	523	115	2.02	181637.9
	1020	CCCC800101	5002	2003-12-16	582	130	2.04	230006.4
	1020	CCCC800101	5017	2000-03-29	8	130	2.04	3161.6
	1030	DDDD800101	5003	1998-01-12	202	145	2.06	89627.40000000001
	1030	DDDD800101	5016	2005-06-07	295	145	2.06	130891.5


```


1 CREATE VIEW v_entregas_detalle AS
2 SELECT
3     E.clave,
4     M.descripcion,
5     E.rfc,
6     P.razonsocial,
7     E.numero,
8     PR.denominacion,
9     E.fecha,

```

Result Grid

  Filter Rows:

Export: 




Wrap Cell Content: 

	clave	descripcion	rfc	razonsocial	numero	denominacion	fecha	cantidad	pr
▶	1160	Cantera rosa	AAAA800101	La fragua	5019	Queretaro limpio	2003-05-05	244	14
	1000	Varilla 3/16	AAAA800101	La fragua	5019	Queretaro limpio	1999-07-13	254	10
	1320	Tubería 4.4	AAAA800101	La fragua	5018	Tu cambio por la educación	2002-06-23	413	23
	1160	Cantera rosa	AAAA800101	La fragua	5016	Ampliación de la carretera a la huasteca	1997-08-09	162	14
	1320	Tubería 4.4	AAAA800101	La fragua	5012	Restauración de instalaciones del CEA	1999-04-27	698	23
	1240	Arena	AAAA800101	La fragua	5011	Remodelación de aulas del IPP	1997-03-13	366	20
	1080	Ladrillos rojos	AAAA800101	La fragua	5011	Remodelación de aulas del IPP	2003-08-28	699	50

```

1 • CREATE VIEW v_resumen_proveedor AS
2 SELECT
3     E.rfc,
4     P.razonsocial,
5     COUNT(*) AS NumEntregas,
6     SUM(E.cantidad * M.precio * (1 + M.PorcentajeImpuesto)) AS TotalImporte
7 FROM Entregan E
8 JOIN Materiales M ON E.clave = M.clave
9 JOIN Proveedores P ON E.rfc = P.rfc




```

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content: 				
	rfc	razonsocial	NumEntregas	TotalImporte
▶	AAAA800101	La fragua	12	3690983.6
	BBB8800101	Oviedo	12	1943979.9
	CCCC800101	La Ferre	12	2529628.8
	DDDD800101	Cecoferre	11	1489725.8
	EEEE800101	Alvin	10	1656878.92
	FFFF800101	Comex	10	1515253.5
	GGGG800101	Tabiquera del centro	10	4850661.2
	HHHH800101	Tubasa	10	5432401.899999999

```

1 • CREATE VIEW v_resumen_proyecto AS
2 SELECT
3     E.numero,
4     PR.denominacion,
5     COUNT(*) AS NumEntregas,
6     SUM(E.Cantidad * M.precio * (1 + M.PorcentajeImpuesto)) AS TotalImporte
7 FROM Entregan E
8 JOIN Materiales M ON E.clave = M.clave
9 JOIN Proyectos PR ON E.numero = PR.numero

```

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content: 				
	numero	denominacion	NumEntregas	TotalImporte
▶	5000	Vamos Mexico	3	375382
	5019	Queretaro limpio	5	1505453.0999999999
	5001	Aztecon	3	507116.9
	5018	Tu cambio por la educación	5	1664872.1
	5002	CIT Campeche	3	542563.2
	5017	Reparación de la carretera del sol	5	468146.8
	5003	Mexico sin ti no estamos completos	3	917687.4
	5016	Ampliación de la carretera a la huasteca	5	1948685.9

v\_resumen\_proyecto 6 ×

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

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

```

1 • SELECT DISTINCT M.Clave, M.descripcion
2   FROM Entregan E
3  JOIN Proyectos P ON E.Numero = P.Numero
4  JOIN Materiales M ON E.Clave = M.Clave
5  WHERE P.Denominacion = 'México sin ti no estamos completos';
6

```

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".

```

1 • SELECT DISTINCT M.Clave, M.descripcion
2   FROM Entregan E
3  JOIN Proveedores P ON E.rfc = P.rfc
4  JOIN Materiales M ON E.clave = M.clave
5  WHERE P.RazonSocial = 'Acme tools';
6

```



Result Grid			Filter Rows:	Export:
	Clave	descripcion		

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

```

1 • SELECT E.RFC
2   FROM Entregan E
3  WHERE YEAR(E.Fecha) = 2000
4  GROUP BY E.RFC
5  HAVING AVG(E.Cantidad) >= 300;
6

```

Result Grid |   Filter Rows:




RFC
BBBB800101
FFFF800101
GGGG800101

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


```

1 • SELECT Clave, SUM(Cantidad) AS TotalEntregado
2   FROM Entregan
3  WHERE YEAR(Fecha) = 2000
4  GROUP BY Clave;
5

```

Result Grid |   Filter Rows:  | Export:  | Wrap C:

Clave	TotalEntregado
1020	8
1050	623
1100	466
1130	625
1140	583
1210	43
1310	72
1360	265

Result 12 x 

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

```

1 • CREATE VIEW v_total_2001 AS
2   SELECT Clave, SUM(Cantidad) AS TotalCantidad
3   FROM Entregan
4   WHERE YEAR(Fecha) = 2001
5   GROUP BY Clave;
6
7 • SELECT * FROM v_total_2001

```

Result Grid			Filter Rows:	Export:	Wi
	Clave	TotalCantidad			
▶	1000	165			
	1100	699			
	1180	407			
	1260	1091			
	1300	119			
	1330	554			
	1420	603			

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

```

1 • SELECT Clave, descripcion
2   FROM Materiales
3   WHERE descripcion LIKE '%ub%';
4

```

Result Grid			Filter Rows:
	Clave	descripcion	
▶	1180	Recubrimiento P1001	
	1190	Recubrimiento P1010	
	1200	Recubrimiento P1019	
	1210	Recubrimiento P1028	
	1220	Recubrimiento P1037	
	1290	Tubería 3.5	
	1300	Tubería 4.3	
	1310	Tubería 3.6	

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

```

1 • SELECT P.Denominacion,
2         SUM(E.Cantidad * M.precio * (1 + M.PorcentajeImpuesto)) AS TotalPagar
3 FROM Entregan E
4 JOIN Materiales M ON E.Clave = M.Clave
5 JOIN Proyectos P ON E.Numero = P.Numero
6 GROUP BY P.denominacion;
7

```

Denominacion	TotalPagar
Vamos Mexico	375382
Aztecon	507116.9
CIT Campeche	542563.2
Mexico sin ti no estamos completos	917687.4
Educando en Coahuila	2044227.4
Infonavit Durango	1074051.5
Reconstrucción del templo de Guadalupe	739092.4
Construcción de plaza Magnolias	415131.34

- 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).

```

1 • CREATE VIEW v_proveedores_televisa AS
2 SELECT DISTINCT E.RFC, P.razonsocial, PR.denominacion
3 FROM Entregan E
4 JOIN Proyectos PR ON E.Numero = PR.Numero
5 JOIN Proveedores P ON E.RFC = P.RFC
6 WHERE PR.Denominacion = 'Televisa en acción';
7
8
9 • CREATE VIEW v_proveedores_educando AS
10 SELECT DISTINCT E.RFC
11 FROM Entregan E

```

Denominacion	RFC	RazonSocial
Televisa en acción	DDDD800101	Cecoferre
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).

```

1 • SELECT DISTINCT 'Televisa en acción' AS Denominacion, P.RFC, P.RazonSocial
2 FROM Entregan E
3 JOIN Proyectos PR ON E.Numero = PR.Numero
4 JOIN Proveedores P ON E.RFC = P.RFC
5 WHERE PR.Denominacion = 'Televisa en acción'
6 AND E.RFC NOT IN (
7     SELECT E2.RFC
8     FROM Entregan E2
9     JOIN Proyectos PR2 ON E2.Numero = PR2.Numero
10    WHERE PR2.Denominacion = 'Educando en Coahuila'
11 );

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Denominacion	RFC	RazonSocial
▶	Televisa en acción	DDDD800101	Cecoferre
	Televisa en acción	CCCC800101	La Ferre

- 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.

```

1 • SELECT DISTINCT M.Clave, M.Descripcion, M.Precio
2 FROM Entregan E
3 JOIN Materiales M ON E.Clave = M.Clave
4 JOIN Proyectos PR ON E.Numero = PR.Numero
5 WHERE PR.Denominacion = 'Televisa en acción'
6 AND E.RFC IN (
7     SELECT E2.RFC
8     FROM Entregan E2
9     JOIN Proyectos PR2 ON E2.Numero = PR2.Numero
10    WHERE PR2.Denominacion = 'Educando en Coahuila'
11 );

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Clave	Descripcion	Precio
▶	1080	Ladrillos rojos	50
	1280	Tepetate	34