



# Tecnológico de Monterrey

## Lab 20: Consultas en SQL

### **Materia:**

Construcción de Software y Toma de Decisiones (Gpo 402)

### **Maestros:**

Ricardo Cortés Espinosa

Eduardo Juárez Pineda

### **Fecha:**

13/3/2024

### **Equipo:**

Jesús Alejandro Cedillo Zertuche - A01705442

## Preguntas de la lectura

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

No aparecería, ya que en entregan solo se encuentran los registros de lo que sí se entregó  
Habría que usar una subconsulta para poder hacer esto por ejemplo.

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

**Compruébalo.**

```
(select * from entregan where clave=1430)
union
(select * from entregan where clave=1300)
```

Una consulta que tiene el mismo resultado sería usar un OR, ya que está buscando una de las claves o la otra.

```
SELECT * FROM entregan WHERE clave=1430 OR clave=1300;
```

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

Se suman las tuplas que tiene materiales con las de entrega, y pone todas las combinaciones posibles (un proveedor, con cada material, precio, etc.).

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

```
SELECT Descripcion
FROM materiales AS M, entregan AS E
WHERE M.clave = E.clave AND
E.fecha BETWEEN '2000-01-01' AND '2000-12-31';
```

Descripcion	
Recubrimiento P1028	
Pintura B1021	
Cantera blanca	
Sillar gris	
Pintura B1022	
Block	
Tubería 3.6	
Sillar gris	
Varilla 4/34	
Pintura C1010	
Varilla 3/17	
Pintura B1021	

Hay varios registros que aparecen varias veces (porque tienen varias entregas a lo largo del tiempo).

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

```
SELECT DISTINCT Descripcion
FROM materiales AS M, entregan AS E
WHERE M.clave = E.clave AND
E.fecha BETWEEN '2000-01-01' AND '2000-12-31';
```

Descripcion
Recubrimiento P1028
Pintura B1021
Cantera blanca
Sillar gris
Pintura B1022
Block
Tubería 3.6
Varilla 4/34
Pintura C1010
Varilla 3/17

Ahora solo se obtiene una vez el resultado, por la palabra **distinct**.

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

```
SELECT P.numero, P.denominacion, E.fecha, E.cantidad
FROM proyectos AS P, entregan AS E
WHERE P.numero = E.numero
ORDER BY fecha DESC;
```

numero	denominacion	fecha	cantidad
5007	Construcción de plaza Magnolias	2006-11-21	2
5018	Tu cambio por la educación	2006-07-12	53
5017	Reparación de la carretera del sol	2006-06-10	93
5018	Tu cambio por la educación	2006-04-15	302
5015	CIT Yucatan	2006-02-16	458
5012	Restauración de instalaciones del CEA	2005-12-05	645
5016	Ampliación de la carretera a la huasteca	2005-11-07	356
5002	CIT Campeche	2005-07-03	24
5016	Ampliación de la carretera a la huasteca	2005-06-07	295
5007	Construcción de plaza Magnolias	2005-06-06	506
5008	Televisa en acción	2005-06-03	337
5003	Mexico sin ti no estamos completos	2005-04-30	576
5010	Construcción de Hospital Infantil	2005-04-21	116
5008	Televisa en acción	2005-04-03	86
5012	Restauración de instalaciones del CEA	2004-11-27	503
5019	Queretaro limpio	2004-11-19	94
5006	Reconstrucción del templo de Guadalupe	2004-09-29	324
5012	Restauración de instalaciones del CEA	2004-08-23	312

Modifique el ejemplo de la lectura para que se adecue a la base de datos que se tiene.

```
SELECT * FROM proyectos where denominacion LIKE 'Re%';
```

**¿Qué resultado obtienes?** Obtengo todos los proyectos cuya denominación empieza con Re.

**Explica que hace el símbolo '%'**. Es con comodín para que no importa que haya después salga el registro que se pidió En mi caso importa que empiece con Re pero no con que acaba (ya que él % está al principio).

```
SELECT * FROM proyectos where denominacion LIKE 'Re';
```

**¿Qué resultado obtienes?** No sale nada en la consulta.

**Explica a qué se debe este comportamiento.** Se debe a que está buscando una denominación que sea Re, y como no existe sale vacía la consulta.

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

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

Se obtiene '¿Que resultado obtienes? ¿??'

**¿Para qué sirve DECLARE?** Para declarar una variable dentro de SQL

**¿Cuál es la función de @foo?** Foo guarda '¿Qué resultado obtienes?' (de forma separada)

**¿Qué realiza el operador SET?** El operador SET actualiza un registro con lo que le defines.

Esto también se usa en update Table por ejemplo

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

La primera consulta busca en la tabla entregan RFC que incluyan de la A a la D, mientras que la segunda consulta es que empiecen con A.

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

Esta consulta busca registros en entrega de 4 dígitos que acaben en 6.

Esto se logra por el carácter \_ que es comodín aunque solo es un carácter. Como hay 3 \_ son 3 números y un 6. Por eso entrega los números que acaban en 6 de 4 dígitos.

### ¿Cómo filtrarías rangos de fechas?

Between es una palabra reservada que busca registros entre los dos rangos (incluyendo el número).

```
SELECT Clave,RFC,Numero,Fecha,Cantidad  
FROM Entregan  
WHERE Numero Between 5000 and 5010;
```

Para filtrar rangos de fecha se puede usar el between y poner la fecha de inicio, y la fecha del fin usando el between.

```
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?** La consulta regresa registros donde el número de entrega este entre 5000 y 5010 donde aparte la razón social del proveedor inicie con La (como la fragua).

**¿Qué función tiene el paréntesis ( ) después de EXISTS?** Se usa una subconsulta donde ahí saca el RFC de los proveedores que inician con La. El paréntesis se pone para ejecutar la subconsulta y que regrese registros que estén en ambas.

**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%' and Entregan.RFC = Proveedores.RFC);
```

Hay que pasarle el atributo que queremos que busque en el subquery.

	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	
	CCCC800101	603	2001-09-09	5002	
	CCCC800101	278	1999-05-05	5008	

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

```
SELECT RFC,Cantidad, Fecha,Numero
FROM Entregan
WHERE Numero Between 5000 and 5010 AND
RFC NOT In (SELECT RFC
FROM Proveedores
WHERE RazonSocial NOT LIKE 'La%' and Entregan.RFC = Proveedores.RFC);
```

Con esto como estamos devolviendo en la subconsulta los que no empiezan con La, usando el NOT IN regresa el mismo resultado.

**¿Qué hace la siguiente sentencia? Explica por qué.**

```
SELECT TOP 2 * FROM Proyectos;
```

Saca los primeros dos registros de la tabla de proyectos. Es una palabra reservada que no funciona en todos los sistemas DB (como MYSQL). Ahí se tiene que usar top.

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

```
SELECT TOP Numero FROM Proyectos;
```

Esta consulta está incorrecta ya que Número lo toma como variable y es una columna de la tabla por lo que regresa error.

**Modifica la tabla de materiales y le agrega columna PorcentajeImpuesto Numérica.**

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

Como los registros están vacíos, hacemos update de materiales donde con set ponemos el cálculo de impuestos.

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

**¿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.RFC, SUM((precio + impuesto) * cantidad) AS "Total  
entregado"  
FROM Materiales AS M, Entregan AS E  
WHERE M.clave = E.clave  
GROUP BY E.RFC;
```

## Consultas con views

### Consulta #1:

```
CREATE VIEW Proveedores_QueEmpiezancon_La
AS SELECT Entregan.RFC, Entregan.Cantidad, Entregan.Fecha,
Entregan.numero
FROM Proveedores, Entregan
WHERE RazonSocial LIKE 'La%' and Entregan.RFC = Proveedores.RFC;

/* Query anterior con la view que regresa lo mismo */
SELECT RFC,Cantidad, Fecha,Numero
FROM Proveedores_QueEmpiezancon_La
WHERE Numero Between 5000 and 5010;
```

Aquí creas una vista con todos los proveedores cuyo nombre empieza con La, y después de esa tabla o vista agarras los proyectos con cierto número.

### Consulta #2:

```
CREATE VIEW Top2_Proyectos
AS SELECT * FROM Proyectos LIMIT 2;

SELECT *
FROM Top2_Proyectos;
```

Creas la view y la guardas para después consultar directamente de la view.

### Consulta #3:

```
CREATE VIEW Numero_Proyectos_Acaban_en_6
AS SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

SELECT *
FROM Numero_Proyectos_Acaban_en_6;
```

Aquí guardas en una view todos los proyectos cuyo número acaba en 6, y después haces la consulta directo del view.



#### Consulta #4:

```
CREATE VIEW Entregas_1430_OR_1300
AS SELECT * FROM entregan WHERE clave=1430 OR clave=1300;

SELECT M.descripcion, M.clave, E.RFC, fecha, cantidad
FROM Entregas_1430_OR_1300 AS E, Materiales AS M
WHERE E.clave = M.clave;
```

Con esta view sacas una tabla de entrega mucho más chica, y cuando se hace el join optimiza la consulta ya que solo tiene que comparar con 4 registros (que con los de la tabla del join).

#### Consulta #5: El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales (usando un view)

```
CREATE VIEW Entregas_2000
AS SELECT *
FROM Entregan AS E
WHERE fecha BETWEEN '2000-01-01' AND '2000-12-31';

SELECT RFC, Avg(cantidad) AS 'Promedio de entregas'
FROM Entregas_2000
GROUP BY RFC;
```

El query que está más abajo (en las consultas que había que hacer) es el original, y ambos regresan exactamente el mismo resultado. Cree un view para practicar hacer las consultas de ambas formas.

### Consultas a generar

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

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

clave	descripcion
1030	Varilla 4/33
1230	Cemento
1430	Pintura B1022

Genera 3 renglones la consulta.

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

```
SELECT M.clave, descripcion
FROM materiales AS M, Entregan AS E, Proveedores AS Prov
WHERE M.clave = E.Clave AND E.rfc = Prov.rfc AND
Prov.razonsocial = 'Acme tools';
```

clave	descripci...

Esta consulta regresa 0 renglones, ya que en la base de datos no existe Acme Tools.

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

```
SELECT RFC, Avg(cantidad) AS 'Promedio de entregas'
FROM Entregan
WHERE fecha Between '2000-01-01' AND '2000-12-31'
GROUP BY RFC
HAVING Avg(cantidad) < 300;
```

RFC	Promedio de entregas
CCCC800101	237.0000
DDDD800101	13.0000
EEEE800101	265.0000
HHHH800101	89.5000

La consulta regresa los 4 renglones de la foto.

El total entregado por cada material en el año 2000.

```
SELECT SUM(E.cantidad) AS 'Total entregado', M.descripcion
FROM Entregan AS E
WHERE E.fecha Between '2000-01-01' AND '2000-12-31'
GROUP BY E.clave;
```

Total entregado	clave
8	1020
623	1050
466	1100
625	1130
583	1140
43	1210
72	1310
265	1360
107	1390
601	1410
13	1430

La consulta regresa 11 renglones, que es lo que se ve en la foto

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

```
SELECT SUM(cantidad), clave
FROM Entregan
WHERE fecha Between '2001-01-01' AND '2001-12-31'
GROUP BY clave
ORDER BY SUM(cantidad) DESC
Limit 1;
```

SUM(cantidad)	clave
1091	1260

La consulta regresa solo 1 renglón, ya que estoy usando limit

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

```
SELECT descripcion
FROM Materiales
WHERE descripcion LIKE '%ub%'
```

descripcion
Recubrimiento P1001
Recubrimiento P1010
Recubrimiento P1019
Recubrimiento P1028
Recubrimiento P1037
Tubería 3.5
Tubería 4.3
Tubería 3.6
Tubería 4.4
Tubería 3.7
Tubería 4.5
Tubería 3.8

La consulta regresa los 12 renglones que se pueden observar en la foto.

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

```
SELECT denominacion, SUM((precio + impuesto) * cantidad) AS "Total a pagar"
FROM Materiales AS M, Entregan AS E, Proyectos AS P
WHERE M.clave = E.clave AND E.numero = P.numero
GROUP BY denominacion
ORDER BY denominacion;
```

denominacion	Total a pagar
Aztecon	161254.5
CIT Campeche	173530.5
CIT Yucatan	936870
Construcción de Hospital Infantil	158724.5
Construcción de plaza Magnolias	135265.90003156662
Disco Atlantic	173910
Educando en Coahuila	682671
Infonavit Durango	353248.5

La consulta regresa 20 renglones, el total de los proyectos que están en la base de datos.

**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 Proyectos_Televisa AS
SELECT DISTINCT denominacion, Prov.RFC, Prov.RazonSocial
FROM proveedores AS Prov, entregan AS E, proyectos AS P
WHERE Prov.rfc = E.rfc AND E.numero = P.numero AND
denominacion = 'Televisa en acción';

CREATE VIEW Proyectos_Educando_Coahuila AS
SELECT Prov.RazonSocial
FROM proveedores AS Prov, entregan AS E, proyectos AS P
WHERE Prov.rfc = E.rfc AND E.numero = P.numero AND
P.denominacion = 'Educando en Coahuila';

SELECT DISTINCT denominacion, RFC, RazonSocial
FROM Proyectos_Televisa
WHERE RazonSocial NOT IN (SELECT RazonSocial FROM
Proyectos_Educando_Coahuila);
```

denominacion	RFC	RazonSocial
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	CCCC800101	La Ferre

La consulta regresa dos renglones (los de la imagen).

**Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Sin usar vistas, utiliza not in, in o exists).**

```
SELECT DISTINCT denominacion, Prov.RFC, Prov.RazonSocial
FROM proveedores AS Prov, entregan AS E, proyectos AS P
WHERE Prov.rfc = E.rfc AND E.numero = P.numero AND
denominacion = 'Televisa en acción' AND
Prov.RazonSocial NOT IN (SELECT Prov.RazonSocial
FROM proveedores AS Prov, entregan AS E, proyectos AS P
WHERE Prov.rfc = E.rfc AND E.numero = P.numero AND
P.denominacion = 'Educando en Coahuila');
```

denominacion	RFC	RazonSocial
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	CCCC800101	La Ferre

Regresa dos renglones, que son los que se ven en la foto.

**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.**

```
CREATE VIEW Materiales_Televisa AS
SELECT precio, descripcion, Prov.RazonSocial
FROM Materiales AS M, entregan AS E, proyectos AS P, proveedores AS
Prov
WHERE M.clave = E.clave AND E.numero = P.numero AND E.rfc = Prov.rfc
AND
P.denominacion = 'Televisa en acción';

CREATE VIEW Proveedores_coahuila AS
SELECT Prov.RazonSocial
FROM Materiales AS M, entregan AS E, proyectos AS P, proveedores AS
Prov
WHERE M.clave = E.clave AND E.numero = P.numero AND E.rfc = Prov.rfc
AND
P.denominacion = 'Educando en Coahuila';

SELECT precio, descripcion, RazonSocial
FROM Materiales_Televisa
WHERE RazonSocial IN (SELECT RazonSocial FROM Proveedores_coahuila);
```

precio	descripcion	RazonSocial
34	Tepetate	Alvin
50	Ladrillos rojos	La fragua

En esta consulta se generan dos renglones.

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

```
SELECT M.descripcion, Count(E.clave) AS 'Cantidad Entregado',
Sum((M.precio + M.impuesto) * E.cantidad) AS 'Total de costo'
FROM Entregan AS E, Materiales AS M
WHERE E.clave = M.clave
GROUP BY E.clave
ORDER BY M.descripcion;
```

descripcion	Cantidad Entregado	Total de costo
Arena	2	113960
Block	2	38445
Cantera amarilla	2	58949
Cantera blanca	2	176440
Cantera gris	2	1212541
Cantera rosa	2	634172
Cemento	2	277860
Grava	2	83820
Gravilla	2	108009
Ladrillos grises	2	19019
Ladrillos rojos	2	43175
Megablock	2	31020

Se regresan 44 renglones de esta consulta.