

Laboratorio 20: Consultas en SQL

Consulta de un tabla completa

Sentencia: select * from entregan;

clave	rfc	numero	fecha	cantidad
1000	AAAA...	5000	2001-12-13	165
1000	AAAA...	5019	1999-07-13	254

Número de renglones: 87

Selección

Sentencia: select * from entregan
where numero = 5000;

clave	rfc	numero	fecha	cantidad
1000	AAAA...	5000	2001-12-13	165
1200	EEEE...	5000	2003-03-15	177
1400	AAAA...	5000	1999-04-07	382
NULL	NULL	NULL	NULL	NULL

Número de renglones: 3

Proyección

Sentencia: select clave, fecha, cantidad
from entregan
where cantidad > 150;

clave	fecha	cantidad
1000	2001-12-13	165
1000	1999-07-13	254
1010	1998-07-28	528

Número de renglones: 67

Reunión Natural

Sentencia: select *
from proveedores as p, entregan as e
where p.rfc = e.rfc;

rfc	razonsocial	clave	rfc	numero	fecha	cantidad
AAAA...	La fragua	1000	AAAA...	5000	2001-12-13	165
AAAA...	La fragua	1000	AAAA...	5019	1999-07-13	254
AAAA...	La fragua	1080	AAAA...	5008	2005-04-03	86

Número de renglones: 87

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

Reunión con criterio específico

Sentencia: select *
from materiales as m, entregan as e
where e.numero >= e.numero;

clave	descripcion	precio	impuesto	Porcentajelmpues...	clave	rfc	numero	fecha	cantidad
2000	Jabón	125	12.5	4.00	1000	AAAA...	5000	2001-12-13	165
1430	Pintura B1022	125	12.5	2.86	1000	AAAA...	5000	2001-12-13	165
1420	Pintura C1012	125	12.5	2.84	1000	AAAA...	5000	2001-12-13	165
1410	Pintura B1001	125	12.5	2.86	1000	AAAA...	5000	2001-12-13	165

Número de renglones: 1000

Unión (se ilustra junto con selección)

Sentencia:
(select *
from materiales as m
where precio < 150)
unión
(select *
from materiales as m
where impuesto < 12);

clave	descripcion	precio	impuesto	PorcentajImpues...
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

Número de renglones: 22

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

Sentencia:

```
select *
from materiales as m
where precio < 150 OR impuesto < 12;
```

clave	descripcion	precio	impuesto	PorcentajImpues...
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

Número de renglones: 22

Diferencia (se ilustra con selección)

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

Sentencia:

```
select *
from entregan as e1
where not exists
(select *
from entregan as e2
where clave = 1000 AND e1.clave = e2.clave);
```

clave	rfc	numero	fecha	cantidad
1010	BBBB...	5001	1998-07-28	528
1010	BBBB...	5018	1997-02-09	523
1020	CCCC...	5002	2003-12-16	582

Número de renglones: 85

Producto cartesiano

Sentencia: select *
from entregan, proveedores;

clave	rfc	numero	fecha	cantidad	rfc	razonsocial
1000	AAAA...	5000	2001-12-13	165	HHHH...	Tubasa
1000	AAAA...	5000	2001-12-13	165	GGG...	Tabiquera del centro
1000	AAAA...	5000	2001-12-13	165	FFFF8...	Comex

Número de renglones: 696

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales? Es la cantidad de posibilidades de combinaciones que hay entre los dos conjuntos.

Construcción de consultas a partir de una especificación

Sentencia:
select descripcion, fecha
from materiales as m
join entregan as e
on m.clave = e.clave
where fecha between '01/01/00' and '31/12/00';

descripcion	fecha
Varilla 3/16	2001-12-13
Recubrimiento P1019	2003-03-15
Varilla 3/17	2003-12-16

Número de renglones: 47

¿Por qué aparecen varias veces algunas descripciones de material?
Porque fueron entregadas en fechas distintas

Uso del calificador distinct

Sentencia:

```
select distinct descripcion
from materiales as m
join entregan as e
on m.clave = e.clave
where fecha between '01/01/00' and '31/12/00';
```

descripcion
Varilla 3/16
Recubrimiento P1019
Varilla 3/17

Número de renglones: 31

¿Qué resultado obtienes en esta ocasión? La lista de descripciones sin repeticiones.

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.

Sentencia:

```
select p.numero, p.denominacion, e.fecha, e.cantidad
from proyectos as p
join entregan as e
on p.numero = e.numero
order by p.numero, e.fecha desc;
```

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

Número de renglones: 87

Operadores de cadena

¿Qué resultado obtienes? Todas las descripciones de materiales que empiecen con Si.

Explica que hace el símbolo '%'.

Permite ser intercambiado por cualquier otro carácter al momento de realizar la consulta.

Sentencia: SELECT * FROM materiales where Descripcion LIKE 'Si%';

clave	descripci...	precio	impuesto	PorcentajImpues...
1120	Sillar rosa	100	10	2.24
1130	Sillar gris	110	11	2.26
NULL	NULL	NULL	NULL	NULL

Número de renglones: 2

¿Qué sucede si la consulta fuera : LIKE 'Si' ? Se buscaría descripciones que únicamente sean 'Si'.

Sentencia: SELECT * FROM materiales where Descripcion LIKE 'Si';

clave	descripci...	precio	impuesto	PorcentajImpues...
NULL	NULL	NULL	NULL	NULL

Número de renglones: 0

¿Qué resultado obtienes? Todas las descripciones de materiales que sean Si. Es decir, ninguna.

Explica a qué se debe este comportamiento.

A qué al % funcionar de comodín permite que la base de datos reemplace ese carácter en la consulta por cualquier otro en cualquier cantidad.

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 = '¿Qué resultado?'
```

```
SET @bar = '¿¿¿??'
```

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

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

No Funcionaron en MySQL Workbench

¿Qué resultado obtienes al ejecutar el siguiente código?

¿Que resultado obtienes? ¿¿¿???

¿Para qué sirve DECLARE?

Declarar una variable local.

¿Cuál es la función de @foo?

Almacenar temporalmente la cadena de texto '¿Que resultado'.

¿Qué realiza el operador SET?

Asigna un valor a una variable dada.

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

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

[^] - En contraparte, este operador coincide con cualquier carácter que no se encuentre dentro del intervalo o del conjunto especificado.

_ - El operador _ o guión bajo, se utiliza para coincidir con un carácter 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]%';
```

Debería devolver la lista de RFCs que comiencen con A, B, C o D. Esto debido a que '[A-D]%' es para seleccionar los valores de RFC que comiencen con cualquier letra entre A y D, seguido de cualquier combinación de caracteres.

Número de renglones: ?

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%' ; No funcionó
```

Debería devolver la lista de RFCs que no comiencen con la letra A. Esto debido al carácter de negación que impide recibir resultados que empiecen en A y a % que permite que haya cualquier combinación de caracteres posteriores.

Número de renglones: ?

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

Devuelve los valores de la columna Numero que tienen exactamente cuatro caracteres y acaba con el número 6. El guión bajo sirve de comodín pero limita al espacio que en sí te da _.

Numero
5006
5006
5006

Número de renglones: 9

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,Número,Fecha,Cantidad
FROM Entregan
WHERE Numero Between 5000 and 5010;
```

¿Cómo filtrarías rangos de fechas?

where fecha between '??/??/??' and '??/??/??'

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] )
```

No Funcionó

¿Qué hace la consulta?

Busca registros en Entregan con un número entre 5000 y 5010, pero solo incluye aquellos registros para los cuales también hay un proveedor en Proveedores cuya Razón Social comienza con 'La' y que comparten el mismo RFC.

¿Qué función tiene el paréntesis () después de EXISTS?

Contener la subconsulta.

IN Especifica si un valor dado tiene coincidencias con algún valor de una subconsulta. NOTA: Se utiliza dentro del WHERE pero debe contener un parámetro. Ejemplo: Where proyecto.id IN Lista_de_Proyectos_Subquery

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

Sentencia:

```
SELECT RFC, Cantidad, Fecha, Numero
FROM Entregan
WHERE Numero BETWEEN 5000 AND 5010
AND RFC IN (SELECT RFC FROM Proveedores WHERE RazonSocial LIKE 'La%')
```

NOT Simplemente niega la entrada de un valor booleano.

Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

Sentencia:

```
select *
from entregan e1
where e1.cantidad > any (
select avg(cantidad)
from entregan e2
where e2.cantidad > 100 and e2.cantidad < 500
);
```

clave	rfc	numero	fecha	cantidad
1010	BBBB...	5001	1998-07-28	528
1010	BBBB...	5018	1997-02-09	523
1020	CCCC...	5002	2003-12-16	582

Número de renglones: 51

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
```

Selecciona los dos primeros registros de proyectos en orden ascendente.

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

```
SELECT TOP Numero FROM Proyectos
```

No funcionó

Modificando la estructura de una tabla existente.

Agrega a la tabla materiales la columna PorcentajImpuesto con la instrucción:

```
ALTER TABLE materiales ADD PorcentajImpuesto 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 PorcentajImpuesto = 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.

clave	descripcion	precio	impuesto	PorcentajImpues...
1000	Varilla 3/16	100	10	2.00
1010	Varilla 4/32	115	11.5	2.02

¿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?

Selección:

```
select sum(cantidad * (precio + impuesto)) as "Importe Total"
```

```
from materiales as m
```

```
join entregan as e
```

```
on m.clave = e.clave
```

group by e.clave;

Impor...
46090
13295...
84370

Número de renglones: 44

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
```

¡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.

Sentencia:

```
create view importetotal (importe) as
select sum(cantidad * (precio + impuesto)) as "Importe Total"
from materiales as m
join entregan as e
on m.clave = e.clave
group by e.clave;
```

select * from importetotal;

Importe Total
46090
132951.5
84370

Sentencia:

```
create view ejem1 (clave, descripcion, precio, impuesto, porcentajeimpuestos) as
SELECT * FROM materiales where Descripcion LIKE 'Si%';
```

clave	descripci...	precio	impuesto	porcentajeimpuest...
1120	Sillar rosa	100	10	2.24
1130	Sillar gris	110	11	2.26

Sentencia:

```
create view ejem2 (numero, denominacion, fecha, cantidad) as
select p.numero, p.denominacion, e.fecha, e.cantidad
from proyectos as p
join entregan as e
on p.numero = e.numero
order by p.numero, e.fecha desc;
```

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

Sentencia:

```
create view ejem3 (clave, descripcion, precio, impuesto, porcentajeimpuestos) as
(select *
from materiales as m
where precio < 150)
union
(select *
from materiales as m
where impuesto < 12);
```

clave	descripcion	precio	impuesto	porcentajeimpuest...
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

Sentencia:

```
create view ejem4 (clave, descripcion, precio, impuesto, porcentajeimpuestos) as
select *
from materiales as m
where precio < 150 OR impuesto < 12;
```

clave	descripcion	precio	impuesto	porcentajeimpuestos
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

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

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".

Sentencia:

```
select distinct m.clave, m.descripcion
from materiales as m
join entregan as e on m.clave = e.clave
```

join proyectos as p on e.numero = p.numero
where p.denominacion = 'México sin ti no estamos completos';

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

Número de renglones: 3

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

Sentencia:

```
select distinct m.clave, m.descripcion
from materiales as m
join entregan as e on m.clave = e.clave
join proveedores as p on e.rfc = p.rfc
where p.razonsocial = 'Acme tools';
```

clave	descripci...

Número de renglones: 0

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

Sentencia:

```
select p.rfc
from proveedores as p
join entregan as e on p.rfc = e.rfc
where year(e.fecha) = 2000
group by p.rfc
having avg(e.cantidad) >= 300;
```

rfc
BBBB...
FFFF8...
GGG...

Número de renglones: 3

El total entregado por cada material en el año 2000.

Sentencia:

```
select m.descripcion, count(e.clave) as "Total Entregado"
from entregan as e
join materiales as m on e.clave = m.clave
where year(e.fecha) = 2000
group by (m.descripcion);
```

descripcion	Total Entregado
Varilla 3/17	1
Varilla 4/34	1
Block	1

Número de renglones: 10

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

Sentencia:

```
select clave from ventas2001
order by totalvendido desc
limit 1;
```

clave
1260

Número de renglones: 1

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

Sentencia:

```
select *
```


from materiales

where descripcion like '%ub%';

clave	descripcion	precio	impuesto	PorcentajImpues...
1180	Recubrimiento P1001	200	20	2.36
1190	Recubrimiento P1010	220	22	2.38
1200	Recubrimiento P1019	240	24	2.40

Número de renglones: 12

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

Sentencia:

```
select p.denominacion, sum(e.cantidad * (m.precio + m.precio *  
m.PorcentajImpuesto)/100) as "Total A Pagar"  
from proyectos as p  
join entregan e on p.numero = e.numero  
join materiales as m on e.clave = m.clave  
group by (p.denominacion);
```

denominacion	Total A Pagar
Vamos Mexico	3753.8199999999997
Aztecon	5071.169
CIT Campeche	5425.632

Número de renglones: 20

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).

Sentencia:

```
create view provTelevisa as  
select p.denominacion, pr.rfc, pr.razonsocial  
from proveedores as pr  
join entregan as e on pr.rfc = e.rfc  
join proyectos as p on e.numero = p.numero  
where p.denominacion = 'Televisa en acción';
```

```
create view provEduCoahulia as  
select pr2.rfc  
from proveedores as pr2
```

```

join entregan as e2 on e2.rfc = pr2.rfc
join proyectos as p2 on p2.numero = e2.numero
where p2.denominacion = 'Educando en Coahuila';

```

```

select pt.denominacion, pt.rfc, pt.razonsocial
from provTelevisa as pt
where pt.rfc not in (select pe.rfc from provEduCoahuila as pe);

```

denominacion	rfc	razonsocial
Televisa en acción	CCCC...	La Ferre
Televisa en acción	DDDD...	Cecoferre
Televisa en acción	DDDD...	Cecoferre

Número de renglones: 3

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).

Sentencia:

```

select p.denominacion, pr.rfc, pr.razonsocial
from proveedores as pr
join entregan as e on pr.rfc = e.rfc
join proyectos as p on e.numero = p.numero
where p.denominacion = 'Televisa en acción' and
pr.rfc not in (
select pr2.rfc
from proveedores as pr2
join entregan as e2 on e2.rfc = pr2.rfc
join proyectos as p2 on p2.numero = e2.numero
where p2.denominacion = 'Educando en Coahuila'
);

```

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

Número de renglones: 3

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.

Sentencia:

```
SELECT m.descripcion, (m.precio + m.precio * m.PorcentajeImpuesto/100) as
Costo
FROM materiales as m
JOIN entregan as e ON m.clave = e.clave
JOIN proyectos as p ON e.numero = p.numero
WHERE p.denominacion = 'Televisa en acción'
AND e.rfc IN (
    SELECT e2.rfc
    FROM entregan e2
    JOIN proyectos p2 ON e2.numero = p2.numero
    WHERE p2.denominacion = 'Educando en Coahuila'
);
```

descripcion	Costo
Ladrillos rojos	51.08
Tepetate	34.870400000000004

Número de renglones: 2

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

Sentencia:

```
select p.denominacion, m.descripcion, count(m.clave) as "Cantidad de Entregas",
(m.precio + m.precio * m.PorcentajeImpuesto/100) as "Costo Total"
from materiales as m
join entregan as e on m.clave = e.clave
join proyectos as p on e.numero = p.numero
group by m.descripcion;
```

denominacion	descripcion	Cantidad de Entreg...	Costo Total
Vamos Mexico	Varilla 3/16	2	102
Vamos Mexico	Recubrimiento P1019	2	245.76
Vamos Mexico	Pintura C1011	4	128.5

Número de renglones: 42