### Laboratorio 15 : Creación de consultas utilizando SQL Básico

Angélica Güemes Estrada

A01421467

Materiales(Clave, Descripción, Costo)

Proveedores(RFC, RazonSocial)

Proyectos(Numero,Denominacion)

Entregan(Clave, RFC, Numero, Fecha, Cantidad)

Convenio: para evitar las letras griegas originales del álgebra relacional, en esta lectura se utiliza la siguiente notación:

SL{condición} : selección con el criterio condición.

PR{lista de columnas}: proyección de lista de columnas.

JN: reunión natural (natural join).

JN{condición}: reunión con el criterio condición (teta join).

UN: unión.

IN: intersección.

- : diferencia

X: producto cartesiano.

<https://miscursos.tec.mx/bbcswebdav/pid-8193681-dt-content-rid-27665457_1/courses/QRO.TC2026.2.1811.20480/lectura4_Mr_Ar/lectura_Mr_Ar.html>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Consultas y sus resultados\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Incluye el número de renglones que SQL Server reporta al final de la consulta.

***1) Álgebra relacional***

Materiales

***1) SQL***

select \* from materiales

***1) Resultado (44 renglones)***

*1000 Varilla 3/16 100.00*

*1010 Varilla 4/32 115.00*

*1020 Varilla 3/17 130.00*

***2) Álgebra relacional (Selección)***

SL{clave=1000}(materiales)

***2) SQL***

select \* from materiales

where clave=1000

***2) Resultado (1 renglón)***

1000 Varilla 3/16 100.00

***3) Álgebra relacional (Proyección)***

PR{clave,rfc,fecha} (entregan)

***3) SQL***

select clave,rfc,fecha

from entregan

***3) Resultado (132 renglones)***

Clave RFC FECHA

1000 AAAA800101 1998-07-08 00:00:00.000

1000 AAAA800101 1999-08-08 00:00:00.000

1000 AAAA800101 2000-04-06 00:00:00.000

1010 BBBB800101 2000-05-03 00:00:00.000

1010 BBBB800101 2000-11-10 00:00:00.000

***4) Álgebra relacional (Reunión natural)***

entregan JN materiales

***4) SQL***

select \* from materiales,entregan

where materiales.clave = entregan.clave

***4) Resultado (132 renglones)***

clave descripcion costo clave rfc numero fecha cantidad

1000 Varilla 3/16 100.00 1000 AAAA800101 5000 1998-07-08 00:00:00.000 165.00

1000 Varilla 3/16 100.00 1000 AAAA800101 5019 1999-08-08 00:00:00.000 254.00

1000 Varilla 3/16 100.00 1000 AAAA800101 5019 2000-04-06 00:00:00.000 7.00

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

No

***5) Álgebra relacional (Reunión con criterio específico)***

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

***5) SQL***

select \* from entregan,proyectos

where entregan.numero < = proyectos.numero

***5) Resultado (500 rebglones)***

1000 AAAA800101 5000 1998-07-08 00:00:00.000 165.00 5000 Vamos Mexico

1200 EEEE800101 5000 2000-03-05 00:00:00.000 177.00 5000 Vamos Mexico

1400 AAAA800101 5000 2002-03-12 00:00:00.000 382.00 5000 Vamos Mexico

***6) Álgebra relacional (Unión y selección)***

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

***6) SQL***

SQL

(select \* from entregan where clave=1450)

union

(select \* from entregan where clave=1300)

***6) Resultado (3 renglones)***

Clave RFC Numero Fecha Cantidad

1300 GGGG800101 5005 2002-06-10 00:00:00.000 521.00

1300 GGGG800101 5010 2003-01-08 00:00:00.000 119.00

1300 GGGG800101 5005 2003-02-02 00:00:00.000 457.00

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

select \*

from entregan

where (clave = 1300) or (clave = 1450)

***7) Álgebra relacional (Intersección con proyección y selección)***

PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan))

***7) SQL***

(select clave from entregan where numero=5001)

intersect

(select clave from entregan where numero=5018)

***7) Resultado***

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:

***8) Álgebra relacional (Diferencia con selección)***

entregan - SL{clave=1000}(entregan)

***8) SQL***

(select \* from entregan)

minus

(select \* from entregan where clave=1000)

***8) Resultado***

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

(select \* from entregan)

except

(select \* from entregan where clave=1000)

***8) Resultado (129 renglones)***

Clave RFC Numero Fecha Cantidad

1010 BBBB800101 5001 2000-05-03 00:00:00.000 528.00

1010 BBBB800101 5018 2000-11-10 00:00:00.000 667.00

1010 BBBB800101 5018 2002-03-29 00:00:00.000 523.00

***9) Álgebra relacional (Producto cartesiano)***

entregan X materiales

***9) SQL***

select \* from entregan,materiales

***9) Resultado (5808 rebglones)***

Clave RFC Numero Fecha Cantidad Clave Descripcion Costo

1000 AAAA800101 5000 1998-07-08 00:00:00.000 165.00 1000 Varilla 3/16 100.00

1000 AAAA800101 5019 1999-08-08 00:00:00.000 254.00 1000 Varilla 3/16 100.00

1000 AAAA800101 5019 2000-04-06 00:00:00.000 7.00 1000 Varilla 3/16 100.00

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

Es un producto cruz. 132\*44

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Construcción de consultas\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Construcción de consultas a partir de una especificación**

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

set dateformat dmy

SELECT Descripcion

FROM Materiales m, Entregan e

WHERE m.Clave = e.Clave

AND e.Fecha BETWEEN ‘01/01/2000’ AND ‘31/12/2000’

***Resultado (28 renglones)***

Descripcion

Varilla 3/16

Varilla 4/32

Varilla 4/32

¿Por qué aparecen varias veces algunas descripciones de material?

Porque se entregaron más de una vez en el año 2000 lo cual se pudo aver evitado con un GROUP BY Descripcion o DISTINCT.

**Uso del calificador distinct**

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

set dateformat dmy

SELECT DISTINCT Descripcion

FROM Materiales m, Entregan e

WHERE m.Clave = e.Clave

AND e.Fecha BETWEEN ‘01/01/2000’ AND ‘31/12/2000’

***Resultado (22 renglones)***

Descripcion

Arena

Block

Cantera rosa

Ladrillos grises

¿Qué resultado obtienes en esta ocasión?

Ya no hay descripciones repetidas.

**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.

set dateformat dmy

SELECT p.Numero, p.Denominacion, e.Fecha, e.Cantidad

FROM Proyecto p, Entregan e

WHERE p.Numero = e.Numero

ORDER BY p.Numero, e.Fecha DESC

***Resultado (132 renglones)***

Numero Denominacion Fecha Cantidad

5000 Vamos Mexico 2002-03-12 00:00:00.000 382.00

5000 Vamos Mexico 2000-03-05 00:00:00.000 177.00

5000 Vamos Mexico 1998-07-08 00:00:00.000 165.00

**Operadores de cadena**

SELECT \* FROM Materiales where Descripcion LIKE 'Si%'

¿Qué resultado obtienes?

Obtiene todos los datos de la tabla productos donde su descripción inicie con la palabra Si, seguido de cualquier tipo de texto o caracter.

***Resultado (2 renglones)***

Clave Descripcion Costo

1120 Sillar rosa 100.00

1130 Sillar gris 110.00

Explica que hace el símbolo '%'

Es un comodín, expresa que puede haber cualquier cosa (texto o caracter) antes (en caso de usarlo como prefijo) o después (como sufijo) de un caracter o texto.

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

Mostraria los datos donde la descripción fuera únicamente un ‘Si’, en este caso no hay resultados.

¿Qué resultado obtienes?

Nada

Explica a qué se debe este comportamiento.

A que no hay ningún producto con descripción ‘Si’ y no se utilizó ningún comodín (%)

**Concatenación, (+, +=)**

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?**

Obtienes ¿Que resultado obtienes? ¿¿¿???

**¿Para qué sirve DECLARE?**

Para declarar una variable

**¿Cuál es la función de @foo?**

Es el nombre que se le puso a la variable de tipo varchar y longitud 40. Su funcion es guardar el valor de esa variable.

**¿Que realiza el operador SET?**

Le asigna valor a variables.

**Operador [ ] , [^] 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 contra parte, este operador coincide con cualquier carácter que no se encuentre dentro del intervalo o del conjunto especificado.

\_ - El operador \_ o guion bajo, se utiliza para coincidir con un carácter de una comparación de cadenas.

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

Comportamiento: Muestra los RFC de Entregan que inicien con A,B,C o D seguido por lo que sea.

Resultado (72 renglones):

RFC

AAAA800101

AAAA800101

AAAA800101

BBBB800101

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

Comportamiento: Muestra los RFC de Entregan que inicien con cualquier letra que no sea A, seguido por lo que sea

Resultado (114 renglones):

RFC

BBBB800101

BBBB800101

BBBB800101

CCCC800101

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

Comportamiento: Muestra el RFC de Entregan que terminen con un 6.

Resultado (14 renglones):

Numero

5016

5016

5006

5006

**Operadores Lógicos**

Los operadores lógicos comprueban la verdad de una condición.

**ALL** 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** 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;

(60 renglones)

Clave RFC Numero Fecha Cantidad

1000 AAAA800101 5000 1998-07-08 00:00:00.000 165.00

1010 BBBB800101 5001 2000-05-03 00:00:00.000 528.00

1020 CCCC800101 5002 2001-07-29 00:00:00.000 582.00

1030 DDDD800101 5003 1998-02-21 00:00:00.000 202.00

¿Cómo filtrarías rangos de fechas?

Con el operador BETWEEN seguido por la fecha de inicio después un AND seguido por la fecha final.

**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] )

***Resultado (16 renglones)***

RFC Cantidad Fecha Numero

AAAA800101 165.00 1998-07-08 00:00:00.000 5000

AAAA800101 86.00 1999-01-12 00:00:00.000 5008

AAAA800101 152.00 2003-01-12 00:00:00.000 5004

¿Qué hace la consulta?

Devuelve RFC, Cantidad, Fecha y Numero de las filas que en el campo Numero tienen un valor entre 5000 y 5010 y que además su azón social (En la tabla proveedores) inicie con La seguido de cualquier cosa.

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

Crea una 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 parametro. 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**

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] )

***Resultado (16 renglones)***

RFC Cantidad Fecha Numero

AAAA800101 165.00 1998-07-08 00:00:00.000 5000

AAAA800101 86.00 1999-01-12 00:00:00.000 5008

AAAA800101 152.00 2003-01-12 00:00:00.000 5004

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

SELECT RFC, Cantidad, Fecha, Numero

FROM [Entregan]

WHERE [Numero] Between 5000 and 5010

AND RFC NOT IN ( SELECT [RFC]

FROM [Proveedores]

WHERE NOT(RazonSocial LIKE 'La%') and NOT( [Entregan].[RFC] = [Proveedores].[RFC])

***Resultado (60 renglones)***

RFC Cantidad Fecha Numero

AAAA800101 165.00 1998-07-08 00:00:00.000 5000

BBBB800101 528.00 2000-05-03 00:00:00.000 5001

CCCC800101 582.00 2001-07-29 00:00:00.000 5002

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

SELECT \*

FROM Entregan

WHERE RFC = ALL(SELECT RFC

FROM Entregan

WHERE RFC LIKE ‘A%’)

***Resultado (18 renglones)***

Clave RFC Numero Fecha Cantidad

1000 AAAA800101 5000 1998-07-08 00:00:00.000 165.00

1000 AAAA800101 5019 1999-08-08 00:00:00.000 254.00

1000 AAAA800101 5019 2000-04-06 00:00:00.000 7.00

El Operador TOP, es un operador que recorre la entrada, un query, y sólo devuelve el primer número o porcentaje especifico de filas basado en un criterio de ordenación si es posible.

¿Qué hace la siguiente sentencia? Explica porque

SELECT TOP 2 \* FROM Proyectos

Regresa las dos primeras filas de los resultados

***Resultado (2 renglones)***

Numero Denominacion

5000 Vamos Mexico

5001 Aztecón

¿Qué sucede con la siguiente consulta? Explica porque

SELECT TOP Numero FROM Proyectos

Te regresa la primera fila del resultado de la consulta.

***Resultado (1 renglones)***

Numero Denominacion

5000 Vamos Mexico

**Modificando la estructura de un tabla existente**

Agrega a la tabla materiales:

ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2); --Agrega una columna

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.

¿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 p.Denominacion, SUM(e.Cantidad\*m.Costo\*(1 + m.PorcentajeImpuestos/100)) as Importe

FROM Materiales m, Proyectos p, Entregan e

WHERE e.Clave=m.Clave AND e.Entregan=p.Entregan

***Resultado (20 renglones)***

Denominacion Importe

Ampliación de la carretera a la huasteca 742461.1940000000

Aztecón 150200.2190000000

CIT Campeche 161603.0820000000

**Creación de vistas**

La sentencia

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

as select...

Permite definir una vista. 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.

La parte (nombrecolumna1,nombrecolumna2,.de la sentencia create view puede ser omitida si no hay ambigüedad en los nombres de las columnas de la sentencia select asociada.

Importante: Las vistas no pueden incluir la cláusula order by.

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

**Ejercicios vistas**

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

**Sentencia**

SELECT m.Clave, m.Descripcion

FROM Materiales m, Entregan e, Proyectos p

WHERE m.Clave=e.Clave AND p.Numero=e.Numero

AND p.Denominacion LIKE ‘México sin ti no estamos completos’

**Renglones**

3

**Resultado**

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".

**Sentencia**

SELECT m.Clave, m.Descripcion

FROM Materiales m, Entregan e, Proveedor p

WHERE m.Clave=e.Clave AND p.RFC=e.RFC

AND p.RazonSocial LIKE ‘Acme tools’

**Renglones**

0

**Resultado**

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

**Sentencia**

set dateformat dmy

SELECT RFC

FROM Entregan e

WHERE Fecha BETWEEN ‘01/01/2000’ AND ‘31/12/2000’

HAVING AVG(Cantidad) >299

GROUP BY RFC

**Renglones**

7

**Resultado**

RFC

BBBB800101

CCCC800101

DDDD800101

EEEE800101

El Total entregado por cada material en el año 2000.

**Sentencia**

set dateformat dmy

SELECT Descripcion, SUM(Cantidad) as ‘Total entregado’

FROM Materiales m, Entregan e

WHERE e.Clave=m.Clave

AND Fecha BETWEEN ‘1/01/2000’ AND ‘31/12/2000’

GROUP BY Descripcion

**Renglones**

22

**Resultado**

Descripcion Total

Arena 366.00

Block 466.00

Cantera rosa 162.00

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

**Sentencia**

set dateformat dmy

SELECT TOP 1 Clave

FROM Entregan

WHERE Fecha BETWEEN ‘1/01/2001’ AND ‘31/12/2001’

GROUP BY Clave

ORDER BY SUM(Cantidad) DESC

**Renglones**

1

**Resultado**

**Clave**

1020

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

**Sentencia**

SELECT Descripcion

FROM Materiales

WHERE Descripcion LIKE ‘%ub%’

**Renglones**

12

**Resultado**

Descripcion

Recubrimiento P1001

Recubrimiento P1010

Recubrimiento P1019

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

**Sentencia**

SELECT pro.Denominacion, SUM(m.Costo\*e.Cantidad\*(1+m.PorcentajeImpuestos/100)) as ‘Total a pagar’

FROM Materiales m, Proyectos pro, Entrega e

WHERE e.Clave=m.Clave AND e.Numero=pro.Numero

GROUP BY Denominacion

**Renglones**

20

**Resultado**

Ampliación de la carretera a la huasteca 742461.1940000000

Aztecón 150200.2190000000

CIT Campeche 161603.0820000000

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 Problema1 as

SELECT Denominacion, p.RFC, p.RazonSocial

FROM Proyectos pro, Proveedores p, Entregan e

WHERE e.RFC=p.RFC AND e.Numero=pro.Numero

WHERE Denominacion LIKE ‘Televisa en acción’

AND Denominacion NOT IN (SELECT Denominacion

FROM Proyectos pro

WHERE Denominacion LIKE ‘Educando en Coahuila’

)

SELECT \*

FROM Problema1

**Renglones**

8

**Resultado**

Denominacion RFC Razon Social

Televisa en acción AAAA800101 La fragua

Televisa en acción DDDD800101 Cecoferre

Televisa en acción DDDD800101 Cecoferre

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 Denominacion, e.RFC, p.RazonSocial

FROM Proyectos p, Proveedores pro, Entregan e

WHERE e.RFC=pr.RFC AND e.Numero=p.Numero

AND Denominacion LIKE ‘Televisa en acción’

**Renglones**

4

**Resultado**

Educando en Coahuila EEEE800101 Alvin

Educando en Coahuila EEEE800101 Alvin

Educando en Coahuila EEEE800101 Alvin

Educando en Coahuila 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.

**Sentencia**

SELECT Costo, Descripcion

FROM Materiales m, Proveedores pro, Proyectos p, Entregan e

WHERE e.Clave = m.Clave AND e.RFC = pro.RFC AND e.Numero = p.Numero

AND (p.Denominacion LIKE ‘Televisa en acción’ AND pro.RFC IN (

SELECT Costo, Descripcion

FROM Materiales m, Proveedores pro, Proyectos p, Entregan e

WHERE e.Clave = m.Clave AND e.RFC = pro.RFC AND e.Numero = p.Numero

AND p.Denominacion LIKE ‘Educando en Coahuila))

**Renglones**

0

**Resultado**

**-**

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

**Sentencia**

SELECT Descripcion, COUNT(e.Clave) as ‘Veces entregado’, SUM(e.Cantidad\*m.Costo) as ‘Costo total de entregas’

FROM Materiales m, Entrega e

WHERE e.Clave=m.Clave

GROUP BY Descripcion

**Renglones**

42

**Resultado**

Arena 3 213400.0000

Block 3 50640.0000

Cantera amarilla 3 172500.0000