Raymundo Romero Arenas

Matrícula: A00570654

Bases de datos – Grupo 1

Profesor: Ricardo Cortés Espinosa

Miércoles 2 de octubre del 2019

**Laboratorio #16**

1. SELECT \* FROM Materiales

|  |  |  |
| --- | --- | --- |
| Clave | Descripción | Costo |
| 1000 | Varilla 3/16 | 100.00 |
| 1010 | Varilla 4/32 | 115.00 |
| 1020 | Varilla 3/17 | 130.00 |

R = 44 filas

1. SELECT \* FROM Materiales

WHERE Clave = 1000

|  |  |  |
| --- | --- | --- |
| Clave | Descripción | Costo |
| 1000 | Varilla 3/16 | 100.00 |

R = 1 fila

1. SELECT Clave, RFC, Fecha FROM Entregan

|  |  |  |
| --- | --- | --- |
| 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 |

R = 132 filas

1. SELECT \* FROM Materiales, Entregan

WHERE Materiales.Clave = Entregan.Clave

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | Descripción | Costo | RFC | Fecha |
| 1000 | Varilla 3/16 | 100.00 | AAAA800101 | 1998-07-08 00:00:00.000 |
| 1000 | Varilla 3/16 | 100.00 | AAAA800101 | 1999-08-08 00:00:00.000 |
| 1000 | Varilla 3/16 | 100.00 | AAAA800101 | 2000-04-06 00:00:00.000 |

|  |  |  |
| --- | --- | --- |
| Clave | Número | Cantidad |
| 1000 | 5000 | 165.00 |
| 1000 | 5019 | 254.00 |
| 1000 | 5019 | 7.00 |

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

R = No

1. SELECT \* FROM Entregan, Proyectos

WHERE Entregan.Numero < = Proyectos.Numero

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | RFC | Número | Fecha | Cantidad |
| 1000 | AAAA800101 | 5000 | 1998-07-08 00:00:00.000 | 165.00 |
| 1200 | EEEE800101 | 5000 | 2000-03-05 00:00:00.000 | 177.00 |
| 1400 | AAAA800101 | 5000 | 2002-03-12 00:00:00.000 | 382.00 |

|  |
| --- |
| Denominación |
| Vamos México |
| Vamos México |
| Vamos México |

R = 1,188 filas

1. (SELECT \* FROM Entregan WHERE Clave = 1450)

UNION

(SELECT \* FROM Entregan WHERE Clave = 1300)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | RFC | Número | Fecha | Cantidad |
| 1300 | GGGG800101 | 5005 | 2002-06-10 00:00:00.000 | 521.00 |
| 1300 | GGGG800101 | 5005 | 2003-02-02 00:00:00.000 | 457.00 |
| 1300 | GGGG800101 | 5010 | 2003-01-08 00:00:00.000 | 119.00 |

R = 3 filas

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

R = SELECT \* FROM Entregan WHERE Clave = 1450 OR Clave = 1300

1. (SELECT Clave FROM Entregan WHERE Numero = 5001)

INTERSECT

(SELECT Clave FROM Entregan WHERE Numero = 5018)

|  |
| --- |
| Clave |
| 1010 |

R = 1 fila

1. SELECT Clave, RFC, Numero, Fecha, Cantidad

FROM Entregan

WHERE Clave NOT IN

(SELECT Clave

FROM Entregan

WHERE Clave = 1000)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | RFC | Número | 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 |

R = 129 filas

1. SELECT \* FROM Entregan, Materiales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | RFC | Número | 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 |

|  |  |
| --- | --- |
| Descripción | Costo |
| Varilla 3/16 | 100.00 |
| Varilla 3/16 | 100.00 |
| Varilla 3/16 | 100.00 |

R = 5,808 filas

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

R = Filas (Entregan) \* Filas (Materiales)

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

SET DateFormat DMY

SELECT Descripcion

FROM Materiales as M, Entregan as E

WHERE M.Clave = E.Clave AND E.Fecha BETWEEN '01/01/00' AND '31/12/00'

|  |
| --- |
| Descripción |
| Varilla 3/16 |
| Varilla 4/32 |
| Varilla 4/32 |

R = 28 filas

¿Por qué aparecen varias veces algunas descripciones de material?

R = Porque se hicieron varias entregas del mismo material el mismo día

1. Agrega la palabra DISTINCT inmediatamente después de la palabra SELECT a la consulta que planteaste antes. ¿Qué resultado obtienes en esta ocasión?

R = Una instancia única de la descripción de los materiales entregados

|  |
| --- |
| Descripción |
| Arena |
| Block |
| Cantera rosa |

R = 22 filas

1. 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, Denominacion, E.Cantidad

FROM Proyectos P, Entregan E

WHERE P.Numero = E.Numero

ORDER BY P.Numero ASC, E.Fecha DESC

|  |  |  |
| --- | --- | --- |
| Número | Denominación | Cantidad |
| 5000 | Vamos Mexico | 382.00 |
| 5000 | Vamos Mexico | 177.00 |
| 5000 | Vamos Mexico | 165.00 |

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

¿Qué resultado obtienes?

R = Los materiales que empiezan con las letras 'Si'

Explica que hace el símbolo '%'.

R = Define la cadena de caracteres a buscar en la columna

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

R = No arrojaría un resultado

Explica a qué se debe este comportamiento.

R = Sólo buscaría aquellos materiales que se llamen 'Si’

1. 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?

R = ¿Que resultado obtienes? ¿¿¿???

¿Para qué sirve DECLARE?

R = Declara una variable junto con su tipo de dato

¿Cuál es la función de @foo?

R = Guarda la cadena '¿Que resultado?'

¿Que realiza el operador SET?

R = Asigna un valor a una variable

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

Resultado = Muestra los RFCs que empiezan en A,B,C,D

Función = Obtiene los RFCs que empiezan con A,B,C,D

Comportamiento = Busca aquellos RFC que empiecen con los caracteres dentro del rango de corchetes

|  |
| --- |
| RFC |
| AAAA800101 |
| AAAA800101 |
| AAAA800101 |

R = 72 filas

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

Resultado = Muestra los RFCs que no empiezan con A

Función = Obtiene el complemento de los RFCs que empiezan con A

Comportamiento = Busca aquellos RFC que no empiecen con los caracteres dentro del rango de corchetes

|  |
| --- |
| RFC |
| BBBB800101 |
| BBBB800101 |
| BBBB800101 |

R = 114 filas

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

Resultado = Muestra los números que terminan en 6

Función = Obtiene los números que tienen como último dígito un 6

Comportamiento = Compara si el último dígito del número es el indicado

|  |
| --- |
| Número |
| 5016 |
| 5016 |
| 5006 |

R = 14 filas

1. SELECT Clave,RFC,Numero,Fecha,Cantidad

FROM Entregan

WHERE Numero Between 5000 and 5010;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clave | RFC | Número | 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 |

¿Cómo filtrarías rangos de fechas?

R = Aplica un Set DateFormat dmy y compara la fecha como un String

1. 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)

|  |  |  |  |
| --- | --- | --- | --- |
| RFC | Número | Fecha | Cantidad |
| AAAA800101 | 165.00 | 1998-07-08 00:00:00.000 | 5000 |
| CCCC800101 | 582.00 | 2001-07-29 00:00:00.000 | 5002 |
| AAAA800101 | 86.00 | 1999-01-12 00:00:00.000 | 5008 |

R = 16 filas

¿Qué hace la consulta?

R = Muestra las entregas con número entre 5000-5010 cuyo RFC empieza con 'La'

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

R = Guarda la subconsulta que el EXISTS usará para hacer la consulta principal

1. Query del 19) usando 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)

1. Query del 19) usando 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%' OR Entregan.RFC != Proveedores.RFC)

1. Query del 19) usando SOME/ANY/ALL

SELECT RFC,Cantidad,Fecha,Numero

FROM Entregan

WHERE Numero BETWEEN 5000 AND 5010 AND RFC = ANY

(SELECT RFC

FROM Proveedores

WHERE RazonSocial LIKE 'La%' AND Entregan.RFC = Proveedores.RFC)

1. SELECT TOP 2 \* FROM Proyectos

|  |  |
| --- | --- |
| Número | Denominación |
| 5000 | Vamos Mexico |
| 5001 | Aztecón |

R = 2 filas

¿Qué hace la siguiente sentencia?

R = Obtiene las primeras dos filas de la tabla Proyectos

1. SELECT TOP Numero FROM Proyectos

¿Qué sucede con la siguiente consulta?

R = No funciona porque no especifica cuántas filas quiere extraer

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

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

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

SELECT SUM(E.Cantidad\*(M.Costo+1/M.PorcentajeImpuesto))

FROM Entregan as E, Materiales as M

1. Crea vistas para cinco de las consultas que planteaste anteriormente en la práctica

* Número 2)

CREATE VIEW NumeroMenor AS (

SELECT \* FROM Materiales

WHERE Clave = 1000

)

SELECT \* FROM NumeroMenor

* Número 6)

CREATE VIEW Entregan1 AS (

SELECT \* FROM Entregan WHERE Clave = 1450

)

CREATE VIEW Entregan2 AS (

SELECT \* FROM Entregan WHERE Clave = 1300

)

(SELECT \* FROM Entregan1)

UNION

(SELECT \* FROM Entregan2)

* Número 13)

CREATE VIEW MaterialSI AS (

SELECT \* FROM Materiales

WHERE Descripcion LIKE 'Si%'

)

SELECT \* FROM MaterialSI

* Número 23

CREATE VIEW TOP2 AS (

SELECT TOP 2 \* FROM Proyectos

)

SELECT \* FROM TOP2

* Número 15

CREATE VIEW AD AS (

SELECT RFC FROM Entregan

WHERE RFC LIKE '[A-D]%'

)

SELECT \* FROM AD