### CONSULTA DE UNA TABLA COMPLETA

|  |  |
| --- | --- |
| **Algebra** | materiales |
| **SQL** | select \* from materiales |
| **Número de tuplas** | 45 |
| **Salida** | |
|  | |

### SELECCIÓN

|  |  |
| --- | --- |
| **Algebra** | SL{clave=1000}(materiales) |
| **SQL** | select \* from materiales where clave=1000 |
| **Número de tuplas** | 1 |
| **Salida** | |
|  | |

### PROYECCIÓN

|  |  |
| --- | --- |
| **Algebra** | PR{clave,rfc,fecha} (entregan) |
| **SQL** | select clave,rfc,fecha from entregan |
| **Número de tuplas** | 87 |
| **Salida** | |
|  | |

### REUNIÓN NATURAL

|  |  |
| --- | --- |
| **Algebra** | entregan JN materiales |
| **SQL** | select \* from materiales,entregan where materiales.clave = entregan.clave |
| **Número de tuplas** | 87 |
| **Salida** | |
|  | |

### REUNIÓN CON CRITERIO ESPECÍFICO

|  |  |
| --- | --- |
| **Algebra** | entregan JN {entregan.numero <= proyectos.numero} proyectos |
| **SQL** | select \* from entregan,proyectos  where entregan.numero < = proyectos.numero |
| **Número de tuplas** | 836 |
| **Salida** | |
|  | |

### UNIÓN (SE ILUSTRA JUNTO CON SELECCIÓN)

|  |  |
| --- | --- |
| **Algebra** | SL{clave=1450}(entregan) UN SL{clave=1300}(entregan) |
| **SQL** | (select \* from entregan where clave=1450) union (select \* from entregan where clave=1300) |
| **SQL Opción 2** | select \* from entregan where clave=1450 OR clave=1300 |
| **Número de tuplas** | 2 |
| **Salida** | |
|  | |

### INTERSECCIÓN (SE ILUSTRA JUNTO CON SELECCIÓN Y PROYECCIÓN)

|  |  |
| --- | --- |
| **Algebra** | PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan)) |
| **SQL** | (select clave from entregan where numero=5001) intersect (select clave from entregan where numero=5018) |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

### DIFERENCIA (SE ILUSTRA CON SELECCIÓN )

|  |  |
| --- | --- |
| **Algebra** | entregan - SL{clave=1000}(entregan) |
| **SQL** | (select \* from entregan) **except** (select \* from entregan where clave=1000) |
| **Número de tuplas** | 85 |
| **Salida** | |
|  | |

### PRODUCTO CARTESIANO

|  |  |
| --- | --- |
| **Algebra** | entregan X materiales |
| **SQL** | select \* from entregan, materiales; |
| **Número de tuplas** | 1000 |
| **Salida** | |
|  | |

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

Número de tuplas de entregan \* Número de tuplas de materiales

### CONSTRUCCIÓN DE CONSULTAS A PARTIR DE UNA ESPECIFICACIÓN

Consulta

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

|  |  |
| --- | --- |
| **SQL** | SELECT descripcion from materiales as M, entregan as E where fecha between '2000-01-01' AND '2000-12-31' |
| **Número de tuplas** | 540 |
| **Salida** | |
|  | |

¿Por qué aparecen varias veces algunas descripciones de material?

Porque varias veces se entregarón los mismos materiales

### USO DEL CALIFICADOR DISTINCT

Consulta

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.

|  |  |
| --- | --- |
| **SQL** | SELECT **DISTINCT** descripcion from materiales as M, entregan as E where fecha between '2000-01-01' AND '2000-12-31' |
| **Número de tuplas** | 43 |
| **Salida** | |
|  | |

¿Qué resultado obtienes en esta ocasión?

La descripción de los materiales que se han entregado en el año 2000 pero sin repeticiones

### ORDENAMIENTOS

Consulta

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.

|  |  |
| --- | --- |
| **SQL** | SELECT P.numero, P.denominacion, E.fecha, E.cantidad FROM entregan as E, proyectos as P ORDER BY P.numero, E.fecha DESC |
| **Número de tuplas** | 1740 |
| **Salida** | |
|  | |

### USO DE EXPRESIONES

En una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

* + Suma
* - Resta
* \* Producto
* / División

### OPERADORES DE CADENA

Like

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.

|  |  |
| --- | --- |
| **SQL** | SELECT \* FROM materiales where descripcion LIKE 'Si%'; |
| **Número de tuplas** | 2 |
| **Salida** | |
|  | |

¿Qué resultado obtienes?

Los materiales los cuales su descripción inicie con ‘Si’

Explica que hace el símbolo '%'.

Es un comodín que determina la posición de los caracteres por los cuales deseamos establecer el patrón de filtrado

En el caso anterior, indicamos que solo deseamos materiales cuya descripción inicie con ‘Si’

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

Explica a qué se debe este comportamiento.

En este caso, nos arrojaría una lista de 0 tuplas, debido a que el operador LIKE actuaria como un ‘=’ especificando que solo deseamos los materiales cuya descripción sea igual a ‘Si’, los cuales no existen.

Operador de concatenación

El operador de concatenación (+, +=), concatena dos o más cadenas de caracteres.  
Su sintaxis es: Expresión + Expresión.

*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?

¿Que resultado obtienes? ¿¿¿???

¿Para qué sirve DECLARE?

La instrucción DECLARE sirve para indicar o declarar una o más variables.

¿Cuál es la función de @foo?

Es el alias de una tabla

¿Que realiza el operador SET? El comando SET sirve para inicializar o asignar un valor a la variable

Otros operadores

[ ] - 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.

|  |  |
| --- | --- |
| **SQL** | SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%'; |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

Comportamiento:

Muestra los rfc de entregas donde el primer carácter del rfc es la letra en el rango de A a la D.

|  |  |
| --- | --- |
| **SQL** | SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%'; |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

Comportamiento:

Muestra los rfc de entregas donde el primer carácter del rfc no es la letra A.

|  |  |
| --- | --- |
| **SQL** | SELECT Numero FROM Entregan WHERE Numero LIKE '\_\_\_6'; |
| **Número de tuplas** | 9 |
| **Salida** | |
|  | |

Comportamiento:

Muestra el número de entregas cuyo número tenga una longitud de 4 caracteres y contenga en la cuarta posición de la cadena el núm. 6, independientemente de que caracteres estén en las primeras 3 posiciones.

### OPERADORES COMPUESTOS.

Los operadores compuestos ejecutan una operación y establecen un valor.

* + = (Suma igual)
* - = (Restar igual)
* \* = (Multiplicar igual)
* / = (Dividir igual)
* % = (Módulo igual)

### OPERADORES LÓGICOS.

ALL

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

Any/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

Between es un operador para especificar intervalos. Una aplicación muy común de dicho operador son intervalos de fechas.

|  |  |
| --- | --- |
| **SQL** | SELECT Clave,RFC,Numero,Fecha,Cantidad  FROM Entregan  WHERE Numero Between 5000 and 5010; |
| **Número de tuplas** | 43 |
| **Salida** | |
|  | |

¿Cómo filtrarías rangos de fechas?

Con Between ‘Fecha’ AND ‘Fecha’

fecha between '2000-01-01' AND '2000-12-31'

|  |  |
| --- | --- |
| **SQL** | SELECT Clave,RFC,Numero,Fecha,Cantidad  FROM Entregan  WHERE fecha between '2000-01-01' AND '2000-12-31' |
| **Número de tuplas** | 12 |
| **Salida** | |
|  | |

EXISTS

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] )

|  |  |
| --- | --- |
| **SQL** | 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 ) |
| **Número de tuplas** | 12 |
| **Salida** | |
|  | |

¿Qué hace la consulta?

Muestra el rfc, cantidad, fecha y numero de los registros en entregan que tienen un numero entre 5000 y 5010, pero además deben existir dentro de la subconsulta que especifica que la razón social de los proveedores debe iniciar con las letras ‘La’

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

Especifica una consulta, es decir, determina una subconsulta de una consulta.

IN

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

|  |  |
| --- | --- |
| **SQL** | 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 ) |
| **Número de tuplas** | 12 |
| **Salida** | |
|  | |

NOT

NOT 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.

|  |  |
| --- | --- |
| **SQL** | SELECT RFC,Cantidad, Fecha,Numero FROM Entregan  WHERE Numero Between 5000 and 5010 AND RFC  NOT IN  (SELECT ALL RFC FROM Proveedores  WHERE RazonSocial NOT LIKE 'La%' and Entregan.RFC = Proveedores.RFC); |
| **Número de tuplas** | 12 |
| **Salida** | |
|  | |

TOP

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 por qué.

|  |  |
| --- | --- |
| **SQL** | SELECT TOP 2 \* FROM Proyectos |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

Explicación:

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

|  |  |
| --- | --- |
| **SQL** | SELECT TOP Numero FROM Proyectos |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

Explicación:

### MODIFICANDO LA ESTRUCTURA DE UNA TABLA EXISTENTE

|  |  |
| --- | --- |
| **SQL** | ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);  UPDATE materiales SET PorcentajeImpuesto = 2\*clave/1000; |
| **Salida** | |
|  | |

¿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?

|  |  |
| --- | --- |
| **SQL** | SELECT ((m.costo\*(1+(m.PorcentajeImpuesto/100)))\*e.cantidad) 'Importe' FROM materiales as m, entregan as e WHERE m.clave = e.clave; |
| **Número de tuplas** | 87 |
| **Salida** | |
|  | |

### CREACIÓN DE VISTAS

Una vista es una tabla virtual cuyo contenido está definido por una consulta. Al igual que una tabla, una vista consta de un conjunto de columnas y filas de datos con un nombre.

Sentencia

*Create view nombrevista (nombrecolumna1 , nombrecolumna2 ,..., nombrecolumna3 )  
as select...*

*select \* from nombrevista*

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

|  |  |
| --- | --- |
| **SQL** | CREATE VIEW vista\_importe  AS SELECT ((m.costo\*(1+(m.PorcentajeImpuesto/100)))\*e.cantidad) 'Importe' FROM materiales as m, entregan as e WHERE m.clave = e.clave; |
| **Número de tuplas** | 87 |
| **Salida** | |
|  | |

|  |  |
| --- | --- |
| **SQL** | CREATE VIEW vista\_materiales\_clave1000 AS select \* from materiales where clave=1000;  SELECT \* from vista\_materiales\_clave1000; |
| **Número de tuplas** | 1 |
| **Salida** | |
|  | |

|  |  |
| --- | --- |
| **SQL** | CREATE VIEW vista\_MaterialesEntregados\_c1450\_1300 as select \* from entregan where clave=1450 or clave=1300; |
| **Número de tuplas** | 2 |
| **Salida** | |
|  | |

|  |  |
| --- | --- |
| **SQL** | CREATE VIEW vista\_diferencia as (select \* from entregan) except (select \* from entregan where clave=1000) |
| **Número de tuplas** | 85 |
| **Salida** | |
|  | |

|  |  |
| --- | --- |
| **SQL** | CREATE VIEW vista\_calificador\_DISTINCT as SELECT DISTINCT descripcion from materiales as M, entregan as E where fecha between '2000-01-01' AND '2000-12-31'; |
| **Número de tuplas** | 43 |
| **Salida** | |
|  | |

### CONSULTAS

* Materiales(Clave, Descripción, Costo)
* Proveedores(RFC, RazonSocial)
* Proyectos(Numero,Denominacion)
* Entregan(Clave, RFC, Numero, Fecha, Cantidad)

/\*

Materiales(Clave, Descripción, Costo)

Proveedores(RFC, RazonSocial)

Proyectos(Numero,Denominacion)

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

\*/

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

|  |  |
| --- | --- |
| **SQL** | SELECT m.clave, m.descripcion FROM materiales as m, entregan as e WHERE m.clave = e.clave AND numero IN  (SELECT numero FROM proyectos WHERE denominacion = "México sin ti no estamos completos"); |
| **SQL option2** | SELECT m.clave, m.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'; |
| **Número de tuplas** | 3 |
| **Salida** | |
|  | |

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

|  |  |
| --- | --- |
| **SQL** | SELECT m.clave, m.descripcion FROM materiales as m, entregan as e WHERE m.clave = e.clave AND RFC IN  (SELECT RFC FROM proveedores WHERE razonsocial= 'Acme tools'); |
| **Número de tuplas** | 0 |

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

|  |  |
| --- | --- |
| **SQL** | SELECT rfc FROM entregan WHERE EXTRACT(year FROM fecha) = 2000 GROUP BY rfc HAVING AVG(cantidad) >= 300; |
| **Número de tuplas** | 3 |
| **Salida** | |
|  | |

El Total entregado por cada material en el año 2000.

|  |  |
| --- | --- |
| **SQL** | SELECT clave, SUM(cantidad) 'Total' FROM entregan WHERE EXTRACT(year FROM fecha) = 2000 GROUP BY clave HAVING SUM(cantidad); |
| **Número de tuplas** | 11 |
| **Salida** | |
|  | |

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

|  |  |
| --- | --- |
| **SQL** | SELECT clave FROM entregan WHERE EXTRACT(year FROM fecha) = 2001 GROUP BY clave ORDER BY SUM(cantidad) DESC LIMIT 1; |
| **Número de tuplas** | 1 |
| **Salida** | |
|  | |

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

|  |  |
| --- | --- |
| **SQL** | SELECT \* FROM materiales WHERE descripcion LIKE '%ub%'; |
| **Número de tuplas** | 12 |
| **Salida** | |
|  | |

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

|  |  |
| --- | --- |
| **SQL** | SELECT p.denominacion, ((m.costo\*(1+(m.PorcentajeImpuesto/100)))\*e.cantidad) 'Total por proyecto' FROM materiales as m, entregan as e, proyectos as p WHERE m.clave = e.clave AND e.numero = p.numero group BY p.denominacion; |
| **Número de tuplas** | 20 |
| **Salida** | |
|  | |

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).

|  |  |
| --- | --- |
| **SQL** |  |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

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).

|  |  |
| --- | --- |
| **SQL** | SELECT p.denominacion, e.rfc, s.razonsocial FROM entregan as e, proveedores as s, proyectos as p WHERE e.rfc= s.rfc AND e.numero = p.numero AND denominacion = 'Televisa en acción' AND p.denominacion IN (SELECT p.denominacion FROM proyectos as p WHERE p.denominacion != 'Educando en Coahuila'); |
| **Número de tuplas** | 5 |
| **Salida** | |
|  | |

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.

|  |  |
| --- | --- |
| **SQL** |  |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

Reto: Usa solo el operador NOT IN en la consulta anterior (No es parte de la entrega).

|  |  |
| --- | --- |
| **SQL** |  |
| **Número de tuplas** |  |
| **Salida** | |
|  | |

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

|  |  |
| --- | --- |
| **SQL** | SELECT m.descripcion, SUM(e.cantidad) 'Veces entregado', ((m.costo\*(1+(m.PorcentajeImpuesto/100)))\*(SUM(e.cantidad))) 'Total por material' FROM materiales as m, entregan as e WHERE m.clave = e.clave GROUP BY m.descripcion; |
| **Número de tuplas** | 42 |
| **Salida** | |
|  | |