Tecnológico de Monterrey

Bases de Datos

Laboratorio 16

Reporte

Miguel Ángel Marines Olvera | A01705317

1. **Investiga Sobre DBMS Empresariales**
    1. **Oracle**
        1. **Requerimientos generales de hardware**

            Procesador de 550 MHz

            Memoria RAM de 1 a 2 GB

            Memoria virtual de 2 a 4 GB

            Instalación avanzada 4.92 GB
        2. **Ambientes o plataformas en las que pueden operar.**

            Multiplataforma: Windows, MAC OS, Linux, AIX, HP-UX, Solaris, etc.
        3. **Costos de implementación y mantenimiento.**

Licencia de Oracle Database	Unit Price
Database Standard Edition	$5.7312
Database Enterprise Edition	$6.1613
Database Enterprise Edition High Performance	$7.0753
Database Enterprise Edition Extreme Performance	$7.9892

        4. **Ventajas y desventajas de su uso.**

            <u>Ventajas</u>:

            - Portabilidad
            - Soporte
            - Multiples Bases de Datos
            - Backup
            - Recovery
            - Interfaz gráfica intuitiva y cómoda de utilizar.
            - Tiene control de acceso y control de entrada de datos.
            - Protección de dato.
            - Particiones.
            - Copias de seguridad.

            <u>Desventajas</u>:

            - Caro para empresas pequeñas
            - La implementación es compleja y requiere de amplios conocimientos.
            - Necesita de muchos recursos.
        5. **Porcentaje del mercado que controlan**

            46 %
        6. **Referencia**

            ORACLE. (2020). Oracle Database Service. 2020, de ORACLE Sitio web: https://go.oracle.com

2. SQL Server

1. Requerimientos generales de hardware

Memoria RAM 1GB
Disco Duro 6GB
Framework .NET
Procesador de 64 bits

2. Ambientes o plataformas en las que pueden operar.

Multiplataforma: Windows

3. Costos de implementación y mantenimiento.

SQL Server Enterprise Edition: $7,128 per core
SQL Server Standard Edition: $1,859 per core
SQL Server Standard Edition Server Licensing: $931 plus $209 per named user client access license (CAL)

4. Ventajas y desventajas de su uso.

Ventajas:

- Se pueden cambiar permisos de muchas cosas
- Permite olvidarse de los ficheros de la BD
- Permite varios usuarios trabajando al mismo tiempo Usa mucha RAM
- Permite administrar los permisos de todos los usuarios.

Desventajas:

- Usa mucha RAM
- La relación calidad precio está por debajo de la de Oracle DB.
- No tiene buena implementación de tipos de datos en variables .

5. Porcentaje del mercado que controlan

17 %

6. Referencias

Microsoft. (2020). SQL Server. 2020, de Microsoft Sitio web: https://www.microsoft.com/es-mx/sql-server/sql-server

3. MySql

1. Requerimientos generales de hardware

Memoria RAM 512 MB

Memoria Virtual 1024 MB

Disco Duro 1 GB

Arquitectura de 32 o 64 bits

Protocolo Red TCP/IP

2. Ambientes o plataformas en las que pueden operar.

Multiplataforma: Windows, AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware.

3. Costos de implementación y mantenimiento.

Name	Price
MySQL Cluster Carrier Grade Edition	$10,000
MySQL Enterprise Edition	$5,000
MySQL Standard Edition	$2,000

4. Ventajas y desventajas de su uso.

Ventajas:

- Es de uso libre, (Open source).
- No necesita recursos de alto rendimiento.
- Fácil instalación y configuración.
- Costos bajos en requerimientos para la elaboración de bases de datos
- Las operaciones se realizan de forma rápida, lo que lo hace uno de los gestores con mejor rendimiento.

Desventajas:

- Carece de documentación oficial.
- No es intuitivo.

5. Porcentaje del mercado que controlan

24 %

6. Referencias

MySQL. (2020). MySQL. 2020, de MySQL Sitio web: https://dev.mysql.com

4. MongoDB

1. Requerimientos generales de hardware

Por cada asset a usar 10MB
UnSSD

2. Ambientes o plataformas en las que pueden operar.

Multiplataforma: Windows, Linux, MAC OS, Solaris.

3. Costos de implementación y mantenimiento.

$57.00 Dólares al mes

4. Ventajas y desventajas de su uso.

Ventajas:

- Valida documentos
- Motores de almacenamiento ya integrados
- Menor tiempo de recuperación de información

Desventajas:

- Tecnología muy joven.
- No es adecuado para transacciones complejas
- No tiene un equivalente a herencia

5. Porcentaje del mercado que controlan

0.4 %

6. Referencias

MongoDB Atlas. (2020). MongoDB Atlas. 2020, de MongoDB Atlas Sitio web: https://www.mongodb.com

5. Conclusión Sobre DBMS Empresariales

Hay varias empresas que brindan servicios de bases de datos de calidad, sin embargo a la hora de contratar una de ellas tenemos que analizar la que mejor se ajuste a las necesidades de la empresa que lo contrata, ya que se tiene que tomar en cuenta varios factores, por ejemplo: la cantidad de información que maneja la empresa, el tipo de información que maneja la empresa, los recursos de los dispositivos con los que se cuentan dentro de la empresa, el costo del servicio, etc.

1. Consulta de un Tabla Completa

Algebra Relacional

Materiales

SQL

select * from Materiales

Sentencia

select * from Materiales

Salida

44 Rows

Muestra Salida

	Clave	Descripcion	Costo
1	1000	Varilla 3/16	100.00
2	1010	Varilla 4/32	115.00
3	1020	Varilla 3/17	130.00

2. Selección

Algebra Relacional

SL{clave=1000}(materiales)

SQL

```
select * from materiales
where clave=1000
```

Sentencia

```
select * from materiales
where clave=1000
```

Salida

1 Row

Muestra Salida

	Clave	Descripcion	Costo
1	1000	Varilla 3/16	100.00

3. Proyección

Algebra Relacional

PR{clave,rfc,fecha} (entregan)

SQL

select clave,rfc,fecha from entregan

Sentencia

```
select clave, rfc, fecha
from entregan
```

Salida

132 Rows

Muestra Salida

	clave	rfc	fecha
1	1000	AAAA800101	1998-07-08 00:00:00.000
2	1000	AAAA800101	1999-08-08 00:00:00.000
3	1000	AAAA800101	2000-04-06 00:00:00.000

4. Reunión Natural

Algebra Relacional

entregan JN materiales

SQL

```
select * from materiales, entregan
where materiales.clave = entregan.clave
```

Sentencia

```
select * from materiales, entregan
where materiales.clave = entregan.clave
```

Salida

132 Rows

Muestra Salida

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

	Clave	Descripcion	Costo	Clave	RFC	Numero	Fecha	Cantidad
1	1000	Varilla 3/16	100.00	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00
2	1000	Varilla 3/16	100.00	1000	AAAA800101	5019	1999-08-08 00:00:00.000	254.00
3	1000	Varilla 3/16	100.00	1000	AAAA800101	5019	2000-04-06 00:00:00.000	7.00

5. Reunión con criterio específico

Algebra Relacional

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

SQL

```
select * from entregan, proyectos
where entregan.numero < = proyectos.numero
```

Sentencia

```
select * from entregan, proyectos
where entregan.numero < = proyectos.numero
```

Salida

1188 Rows

Muestra Salida

	Clave	RFC	Numero	Fecha	Cantidad	Numero	Denominacion
1	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00	5000	Vamos Mexico
2	1200	EEEE800101	5000	2000-03-05 00:00:00.000	177.00	5000	Vamos Mexico
3	1400	AAAA800101	5000	2002-03-12 00:00:00.000	382.00	5000	Vamos Mexico

6. Unión (se ilustra junto con selección)

Algebra Relacional

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

SQL

(select * from entregan where clave=1450)
union
(select * from entregan where clave=1300)

Sentencia

(select * from entregan where clave=1450)
union
(select * from entregan where clave=1300)

Salida

3 Rows

Muestra Salida

	Clave	RFC	Numero	Fecha	Cantidad
1	1300	GGGG800101	5005	2002-06-10 00:00:00.000	521.00
2	1300	GGGG800101	5005	2003-02-02 00:00:00.000	457.00
3	1300	GGGG800101	5010	2003-01-08 00:00:00.000	119.00

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

Sentencia

Select *from entregan where clave=1450 or clave=1300

Salida

3 Rows

Muestra Salida

	Clave	RFC	Numero	Fecha	Cantidad
1	1300	GGGG800101	5005	2002-06-10 00:00:00.000	521.00
2	1300	GGGG800101	5005	2003-02-02 00:00:00.000	457.00
3	1300	GGGG800101	5010	2003-01-08 00:00:00.000	119.00

7. Intersección (se ilustra junto con selección y proyección)

Algebra Relacional

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

SQL

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:

```
(select clave from entregan where numero=5001)
intersect
(select clave from entregan where numero=5018)
```

Sentencia

```
(select clave from entregan where numero=5001)
intersect
(select clave from entregan where numero=5018)
```

Salida

1 Row

Muestra Salida

	clave
1	1010

8. Diferencia (se ilustra con selección)

Algebra Relacional

entregan - SL{clave=1000}(entregan)

SQL

(select * from entregan)
minus
(select * from entregan where clave=1000)

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 where clave=1000

Salida

3 Rows

Muestra Salida

	Clave	RFC	Numero	Fecha	Cantidad
1	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00
2	1000	AAAA800101	5019	1999-08-08 00:00:00.000	254.00
3	1000	AAAA800101	5019	2000-04-06 00:00:00.000	7.00

9. Producto cartesiano

Algebra Relacional

entregan X materiales

SQL

select * from entregan, materiales

Sentencia

select * from entregan, materiales

Salida

5808 Rows

Muestra Salida

	Clave	RFC	Numero	Fecha	Cantidad	Clave	Descripcion	Costo
1	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00	1000	Varilla 3/16	100.00
2	1000	AAAA800101	5019	1999-08-08 00:00:00.000	254.00	1000	Varilla 3/16	100.00
3	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 una multiplicación de cada una de las tuplas de materiales por cada una de las duplas de entregan.

10. Construcción de consultas a partir de una especificación

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

Recuerda que la fecha puede indicarse como '01-JAN-2000' o ’01/01/00'.

Importante: Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa qué vas a trabajar con ese formato de fechas.

Sentencia

```
SET DATEFORMAT dmy
Select Descripción
From Materiales, Entregan
Where Materiales.clave = Entregan.clave and Entregan.fecha >= ’01/01/00’ and
Entregan.fecha <= ’31/12/00’
```

Salida

28 Rows

¿Por qué aparecen varias veces algunas descripciones de material?

Porque hubo varias entregas de ese material.

	Descripcion
1	Varilla 3/16
2	Varilla 4/32
3	Varilla 4/32

11. Uso del calificador distinct

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.

Sentencia

```
SET DATEFORMAT dmy
Select distinct Descripción
From Materiales, Entregan
Where Materiales.clave = Entregan.clave and
Entregan.fecha >= '01/01/00' and Entregan.fecha <= '31/12/00'
```

Salida

22 Rows

Muestra Salida

	Descripcion
1	Arena
2	Block
3	Cantera rosa

¿Qué resultado obtienes en esta ocasión?

Se eliminan las filas que se repetían.

12. Ordenamientos.

Si al final de una sentencia select se agrega la cláusula

order by campo [desc] [,campo [desc] …]

donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.

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 P, Entregan E
Where E.Numero = P.Numero
Order by P.Numero, E.Fecha DESC
```

Salida

132 Rows

Muestra Salida

	Numero	denominacion	Fecha	Cantidad
1	5000	Vamos Mexico	2002-03-12 00:00:00.000	382.00
2	5000	Vamos Mexico	2000-03-05 00:00:00.000	177.00
3	5000	Vamos Mexico	1998-07-08 00:00:00.000	165.00

Uso de expresiones.

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo 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

Las columnas con expresiones pueden renombrarse escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre"). Para SQL Server también pueden utilizarse comillas simples.

13. Operadores de cadena

El operador 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.

También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.

SELECT * FROM Materiales where Descripcion LIKE 'Si%'

¿Qué resultado obtienes?

2 Rows

	Clave	Descripcion	Costo
1	1120	Sillar rosa	100.00
2	1130	Sillar gris	110.00

Explica que hace el símbolo '%'.

Funciona como un comodín para encontrar registros que tengan esa combinación de caracteres .

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

No devuelve nada porque no hay ninguna descripción que sea solo Si

¿Qué resultado obtienes?

Se obtiene una tabla o conjunto vacío

	Clave	Descripcion	Costo

Explica a qué se debe este comportamiento.
La única parte que es tomada en cuenta es el Si y cómo no hay ninguno de esta manera, no se obtiene nada y ningún resultado.

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 = '¿Que resultado';
SET @bar = ' ¿¿¿??? '
SET @foo += ' obtienes?';
PRINT @foo + @bar;
```

¿Para qué sirve DECLARE?

DECLARE sirve para declarar variables.

¿Cuál es la función de @foo?

La función de @foo es ser una variable que guarda un string, (cadena de caracteres).

¿Que realiza el operador SET?

Asigna un valor a una variable

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 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.

Ahora explica el comportamiento, función y resultado de cada una de las siguientes consultas:

15.

Sentencia

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

Salida

Muestra todos los RFC's que empiezan desde A hasta D.

72 Rows

Muestra Salida

	RFC
1	AAAA800101
2	AAAA800101
3	AAAA800101

16.

Sentencia

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

Salida

Muestra todos los RFC's que NO empiezan en A.

114 Rows

Muestra Salida

Results Messages

	RFC
1	BBBB800101
2	BBBB800101
3	BBBB800101

17.

Sentencia

SELECT Numero FROM Entregan WHERE Numero LIKE '___6';

Salida

Muestra los números de los proyectos que que su cuarto carácter sea un 6.

14 Rows

Muestra Salida

	Numero
1	5016
2	5016
3	5006

Operadores compuestos.

Los operadores compuestos ejecutan una operación y establecen un valor.

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

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

18. ¿Cómo filtrarías rangos de fechas?

Sentencia

```
Select Distinct Description
From Materiales M, Entregan E
Where M.Clave = E.Clave and E.Fecha Between '01-01-200' and '31-12-2000'
```

Salida

22 Rows

Muestra Salida

	Descripcion
1	Arena
2	Block
3	Cantera rosa

EXISTS Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.

19. ¿Qué hace la consulta?

Sentencia

```
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] )
```

Salida

Muestra algunos datos de las entregas donde el numero esté entre 5000 y 5010, y además la razón social de los proveedores empiece con La.

16 Rows

Muestra Salida

	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165.00	1998-07-08 00:00:00.000	5000
2	CCCC800101	582.00	2001-07-29 00:00:00.000	5002
3	AAAA800101	86.00	1999-01-12 00:00:00.000	5008

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

Agrupa una consulta para que se ejecute primero

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

20. 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 E
Where E.RFC IN (Select RFC from Proveedores where RazonSocial Like 'La%' and
Numero Between 5000 and 5010)
```

Salida

	RFC	Cantidad	Fecha	Numero
1	AAAA800101	165.00	1998-07-08 00:00:00.000	5000
2	CCCC800101	582.00	2001-07-29 00:00:00.000	5002
3	AAAA800101	86.00	1999-01-12 00:00:00.000	5008

NOT Simplemente niega la entrada de un valor booleano.

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

Sentencia

```
Select E.RFC, Cantidad, Fecha, Numero
From Entregan E, Proveedores P
WHERE E.RFC = P.RFC AND E.RFC NOT IN (select E.RFC from Proveedores where
Numero < 5000 Union SELECT RFC from Proveedores where numero > 5010) AND
P.RazonSocial LIKE 'La%'
```

Salida

16 Rows

	RFC	Cantidad	Fecha	Numero
1	AAA…	165.00	1998-07-08 00…	5000
2	CCC…	582.00	2001-07-29 00…	5002
3	AAA…	86.00	1999-01-12 00…	5008

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

Sentencia

```
SELECT E.Cantidad
FROM Entregan E
WHERE E.Clave = ANY (SELECT M.Clave FROM Materiales M WHERE M.Costo > 450)
```

Salida

	Cantidad
1	453.00
2	270.00
3	458.00

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.

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

Sentencia

SELECT TOP 2 * FROM Proyectos

Salida

Muestra las dos primeras filas de proyectos.

	Numero	Denominacion
1	5000	Vamos Mexico
2	5001	Aztecón

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

Sentencia

SELECT TOP Numero FROM Proyectos

Salida

No se lleva a cabo la consulta porque no se especifican las filas a mostrar.

Modificando la estructura de un tabla existente.

25. Agrega a la tabla materiales la columna PorcentajeImpuesto con la instrucción:

Sentencia

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

26. A fin de que los materiales tengan un impuesto, les asignaremos impuestos ficticios basados en sus claves con la instrucción:

Sentencia

```
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.

Revisa la tabla de materiales para que compruebes lo que hicimos anteriormente.

27. ¿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?

Sentencia

```
Select SUM(Costo) + SUM(PorcentajeImpuesto) as 'TOTAL'
FROM Materiales M, Entregan E
WHERE E.Clave = M.Clave
```

Crea vistas para cinco de las consultas que planteaste anteriormente en la práctica . Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

1.

```
CREATE VIEW TERMINANEN6
AS SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
```

2.

```
CREATE VIEW EntreganTodo
AS SELECT Clave, RFC, Fecha from Entregan
```

3.

```
CREATE VIEW Materiales1000
AS SELECT * from Materiales WHERE Clave=1000
```

4.

```
CREATE VIEW Descripcionanio2000
AS SELECT DISTINCT Descripcion
FROM Materiales M, Entregan E
WHERE M.Clave = E.Clave and E.Fecha BETWEEN '01_01_2000' and '31-12-2000'
```

5.

```
CREATE VIEW DescMateriales
AS SELECT * FROM Materiales WHERE Descripcion LIKE 'Si%'
```

6.

```
CREATE VIEW PROYPORFECHA
AS SELECT P.Numero, P.Denominacion, E.Fecha, E.Cantidad
FROM Proyectos P, Entregan  E
ORDER BY P.Numero, E.Fecha DESC
```

7.

```
CREATE VIEW MaterialesEnEl2000
AS SELECT DISTINCT M.Descripcion
FROM Materiales M,Entregan E
WHERE M.Clave = E.Clave AND E.Fecha > '2000-01-01' AND E.Fecha < '2000-12-31'
```

8.

```
CREATE VIEW ENTREGAS_EN2000
AS SELECT Clave, RFC, Numero, Fecha, Cantidad
FROM Entregan
WHERE Fecha BETWEEN '2000-01-01' AND '2000-12-21'
```

9.

```
CREATE VIEW EMPIEZANCONSI
AS SELECT *
FROM Materiales WHERE Descripcion LIKE 'Si%'
```

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.

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

Sentencia

```
SELECT E.Clave, M.Descripcion
FROM Materiales M, Entregan E, Proyectos P
WHERE M.Clave = E.Clave and P.Numero = E.Numero AND
P.Denominacion = 'Mexico sin ti no estamos completos'
```

Salida

3 Rows

	Clave	Descripcion
1	1030	Varilla 4…
2	1230	Cemento
3	1430	Pintura B…

29. 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, Proyectos P
Where M.Clave = E.Clave AND E.RFC = P.Numero and E.RFC IN(Select RFC From Proveedores P Where P.RazonSocial like 'ac%')

```
SELECT M.Descripcion, E.Clave
FROM Materiales M, Entregan E, Proveedores P
WHERE M.Clave = E.Clave AND E.RFC = P.RFC AND P.RazonSocial
= 'Acme tools'
```

Salida

0 rows

	Clave	Descripcion

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

Sentencia

```
SET DATEFORMAT dmy
SELECT RFC, SUM(Cantidad) AS SUM FROM Entregan E
WHERE E.Fecha BETWEEN '01-01-2000' AND '31-12-2000'
GROUP BY RFC
HAVING SUM(Cantidad) >= 300
```

Select RFC, Sum(Cantidad) as sum
From Entregan E
Where E.Fecha Between '01-01-2000' and '31-12-2000'
Group By RFC
Having sum(Cantidad) >=300

Salida

8 Rows

	RFC	sum
1	AAAA800101	948.00
2	BBBB800101	1962.00
3	CCCC800101	744.00

31. El Total entregado por cada material en el año 2000.

Sentencia

```
SET DATEFORMAT dmy
SELECT Clave, SUM(E.Cantidad) AS 'Total'
FROM Entregan E
WHERE E.Fecha BETWEEN '01-01-2000' AND '31-12-2000' GROUP
BY Clave
```

Salida

22 Rows

	Clave	Total
1	1000	7.00
2	1010	1195.00
3	1030	295.00

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

Sentencia

```
CREATE VIEW masVendido
AS SELECT Clave
FROM Entregan
WHERE Fecha BETWEEN '2001-01-01' AND '2001-12-31' GROUP BY
Clave
ORDER BY Cantidad DESC
LIMIT 1
```

Salida

0 Rows

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

Sentencia

```
Select Descripcion
FROM Materiales
WHERE Descripcion LIKE '%ub%'
```

Salida

12 rows

	Descripcion
1	Recubrimiento P1001
2	Recubrimiento P1010
3	Recubrimiento P1019

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

Sentencia

```
SELECT Denominacion, SUM(Cantidad)
FROM Proyectos P, Entregan E
WHERE P.Numero = E.Numero
GROUP BY Denominacion
```

Salida

20 Rows

	Denominacion	Suma Total
1	Ampliación de la carreter…	2623.00
2	Aztecón	1172.00
3	CIT Campeche	1209.00

35. A. 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 VTelevisa
AS SELECT Pro.Denominacion, E.RFC, P.RazonSocial
FROM Entregan E, Proveedores P, Proyectos Pro
WHERE P.RFC = E.RFC AND E.Numero = Pro.Numero AND
Pro.Denominacion LIKE 'Televisa%'

CREATE VIEW Educando
AS SELECT Pro.Denominacion, E.RFC, P.RazonSocial
FROM Entregan E, Proveedores P, Proyectos Pro
WHERE P.RFC = E.RFC AND E.Numero = Pro.Numero AND
Pro.Denominacion LIKE 'Educando%'

SELECT T.Denominacion, T.RFC, T.RazonSocial
FROM vtelevisa T
WHERE T.RFC
NOT IN (SELECT RFC FROM educando)
```

Salida

	Denominacion	RFC	RazonSocial
1	Televisa en acción	CCCC800101	La Ferre
2	Televisa en acción	CCCC800101	La Ferre
3	Televisa en acción	DDDD800101	Cecoferre

35.B. 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 Pro.Denominacion, E.RFC, P.RazonSocial
FROM Entregan E, Proveedores P, Proyectos Pro
WHERE P.RFC = E.RFC and E.Numero = Pro.Numero and
Pro.Denominacion
LIKE 'Televisa%' AND P.RFC NOT IN(SELECT E.RFC FROM
Entregan E, Proyectos Pro WHERE E.Numero = Pro.Numero and
Pro.Denominacion LIKE ‘Educando%')
```

Salida

	Denominacion	RFC	RazonSocial
1	Televisa en acción	CCCC800101	La Ferre
2	Televisa en acción	CCCC800101	La Ferre
3	Televisa en acción	DDDD800101	Cecoferre

36. 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

```
CREATE VIEW ProvEdCuahuila
AS SELECT * FROM Entregan
WHERE numero = 5004;

SELECT DISTINCT M.Costo, M.Descripcion
FROM Materiales M, Proyectos P, Entregan E, Proveedores Pv
WHERE E.Clave = M.Clave AND E.Numero = P.Numero AND E.RFC =
Pv.RFC AND P.Denominacion AND P.Denominacion LIKE
'Televisa%' AND Pv.RFC IN(SELECT RFC FROM ProvEdCuahuila)
```

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

Sentencia

```
SELECT M.Descripcion, SUM(E.Cantidad) AS Total, COUNT(*) AS
Entregas
FROM Entregan E, Materiales M
WHERE E.Clave = M.Clave GROUP BY E.Clave, M.Descripcion
```

Salida

	Descripcion	Total	Entregas
1	Varilla 3/16	426.00	3
2	Varilla 4/32	1718.00	3
3	Varilla 3/17	1068.00	3