



Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Querétaro

ISC

Laboratorio 16 : DBMS empresariales y consultas básicas en SQL

Desarrollo de Aplicaciones Web y Bases de Datos

Eduardo Juárez y Ricardo Cortés

Jesús Olmos Larios - A01275595

Quinto Semestre

Investigación sobre DBMS empresariales

ORACLE

Requerimientos Generales de Hardware	<p>Para Windows:</p> <p>Memoria mínima – 1GB</p> <p>Memoria recomendada – 2GB</p> <p>Espacio en disco mínimo – 500 MB de espacio Libre</p> <p>Espacio en disco recomendado – 1GB de espacio Libre</p> <p>JVM - J2SE 5.0 / Java SE 6</p>
Ambientes o plataformas en las que pueden operar	<p>Sun Solaris 9, 10 (SPARC) - Solaris 9, 10 (x86)</p> <p>64-bit Sun Solaris 10 (SPARC, x86)</p> <p>Red Hat Enterprise Linux 3.0 Actualización 1, 4.0 y 5.x</p> <p>Red Hat Enterprise Linux 5.x de 64 bits</p> <p>SUSE Linux Enterprise Server 10 (SP1 y SP2 también son compatibles)</p> <p>SUSE Linux Enterprise Server 10 de 64 bits (SP1 también es compatible)</p> <p>SuSE Linux Enterprise Server 11 de 64 bits</p> <p>Ubuntu Linux 8.04, versión Hardy</p> <p>AIX 5.2, 5.3, 6.1</p> <p>Windows 2000 SP4+ - Advanced Server SP4+ - Windows Server 2003, 2008 - Windows XP Pro SP3 - Windows Vista - Windows 2008 – Windows 7</p> <p>Macintosh OS 10.4, 10.5 (Intel, Power)</p> <p>OpenSolaris</p>
Costos de implementación y mantenimiento	Es gratuito para el desarrollo, la implementación y la redistribución.
Ventajas	<p>Información sobre parches y actualizaciones de corrección de errores</p> <p>Videos de procedimientos en pantalla y tutoriales</p> <p>Noticias y eventos</p> <p>Ofertas de asistencia y formación</p>
Desventajas	<p>Glassfish continuará con la implementación Java EE como referencia y como proyecto Open Source. Esta es la desventaja más importante ya que el gasto para las empresas es muy elevado</p> <p>Glassfish cuenta con Netbeans como IDE, ya incorporado en el propio Netbeans.</p> <p>Glassfish soporta las mas recientes versiones de JSP, Java Servlets y la versión JEE 6.</p>
Porcentaje del mercado que controlan	5.92%

SQL Server

Requerimientos Generales de Hardware	<p>Hard Disk:</p> <p>SQL Server requires a minimum of 6 GB of available hard-disk space. Disk space requirements will vary with the SQL Server components you install. For more information, see Hard Disk Space Requirements later in this article. For information on supported storage types for data files, see Storage Types for Data Files.</p> <p>Monitor:</p> <p>SQL Server requires Super-VGA (800x600) or higher resolution monitor.</p> <p>Internet: Internet functionality requires Internet access (fees may apply).</p> <p>Memory *</p> <p>Minimum: Express Editions: 512 MB</p> <p>All other editions: 1 GB</p> <p>Recommended: Express Editions: 1 GB</p> <p>All other editions: At least 4 GB and should be increased as database size increases to ensure optimal performance.</p> <p>Processor Speed:</p> <p>Minimum: x64 Processor: 1.4 GHz - Recommended: 2.0 GHz or faster</p> <p>Processor Type: x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support</p>
--------------------------------------	---

Ambientes o plataformas en las que pueden operar	Ambiente de Desarrollo Microsoft Windows 2003 Server Microsoft Windows 2000 Professional/Server Microsoft Windows 95/98 Microsoft Windows NT Workstation 4.0 y posteriores Microsoft Windows NT Server 4.0 y posteriores Microsoft Windows NT Server Enterprise Edition 4.0 Windows 3.X MS-DOS Third party Internet browsers		
Costos de implementación y mantenimiento	Ediciones Enterprise Standard, por núcleo Standard: servidor Standard: CAL Developer Web Express	Precio de Open No Level (USD) \$13, 748 \$3,586 \$899 \$209 Gratis Consulte los precios a su socio Gratis	Modelo de licencias Paquete de 2 núcleos Paquete de 2 núcleos Servidor[2] CAL Por usuario No aplicable No aplicable
Ventajas	Es útil para manejar y obtener datos de la red de redes. Nos permite olvidarnos de los ficheros que forman la base de datos. Si trabajamos en una red social nos permite agregar otros servidores de SQL Server. Por ejemplo dos personas que trabajan con SQL Server, uno de ellos se puede conectar al servidor de su otro compañero y así se puede ver las bases de datos del otro compañero con SQL Server. SQL permite administrar permisos a todo. También permite que alguien conecte su SQLO al nuestro pero sin embargo podemos decirle que no puede ver esta base de datos pero otro si.		
Desventajas	Utiliza mucho la memoria RAM para las instalaciones y utilización de software. No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular. La relación, calidad y el precio esta muy debajo comparado con oracle. Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.		
Porcentaje del mercado que controlan	18.07%		

MySql

Requerimientos Generales de Hardware	512 Mb de memoria Ram 1024 Mb maquina virtual 1 GB de espacio de disco duro Sistema operativo:Windows, Linux y Unix Arquitectura del sistema 32/64 bit Protocolo de red TCP/IP				
Ambientes o plataformas en las que pueden operar	AIX GNU/Linux OS/2 Warp SunOS Tru64 Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10 y Windows Server (2000, 2003, 2008 y 2012). OpenVMS	BSD Mac OS X QNX SCO OpenServer eBD	FreeBSD Mac OS X QNX	HP-UX NetBSD SGI IRIX SCO UnixWare	Kurisu OS OpenBSD Solaris

Costos de implementación y mantenimiento	<p>Hay dos versiones de las licencias comerciales perpetuas que se venden a los socios la MySQL Classic y MySQL Pro.</p> <p>El costo para la licencia MySQL Classic es \$399</p> <p>El coste para MySQL Pro es \$699.</p> <p>La version Classic y Pro de la licencia de terceros se licencia por base del servidor, permitiendo que un número ilimitado de usuarios, los dispositivos, las máquinas, etc. conecten con el servidor, sin hacia fuera un límite de CPU's en esa máquina.</p>
Ventajas	<p>MySQL software es Open Source</p> <p>Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.</p> <p>Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.</p> <p>Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos</p> <p>Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.</p> <p>Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet</p> <p>El software MySQL usa la licencia GPL</p>
Desventajas	<p>Un gran porcentaje de las utilidades de MySQL no están documentadas.</p> <p>No es intuitivo, como otros programas (ACCESS).</p>
Porcentaje del mercado que controlan	15.36%

MongoDB

Requerimientos Generales de Hardware	Los componentes principales de MongoDB se ejecutan en hardware little-endian, principalmente los procesadores x86 / x86_64. Las bibliotecas cliente (es decir, los controladores) pueden ejecutarse en sistemas endian grandes o pequeños.
Ambientes o plataformas en las que pueden operar	Desde Foursquare y LinkedIn o empresas de telecomunicaciones como Orange y Telefónica. Empresas como Cisco, Bosch o plataformas de formación como Codecademy. Otras son eBay, Expedia. Forbes, IBM, Windows Azure, McAfee o periódicos como The Guardian, Le Figaro, The NewYork Times, etc. Incluso el CERN (Organización Europea para la Investigación Nuclear) utiliza MongoDB para los grandes volúmenes de datos que genera el acelerador de partículas. De igual manera: C ADP, Adobe, AstraZeneca, BBVA, eBay,, Facebook's Parse, MetLife, Pearson, The Weather Channel, Ticketmaster.
Costos de implementación y mantenimiento	Gratuito a menos de que se requiera soporte
Ventajas	<p>Es ideal para entornos con pocos recursos de computación</p> <p>Es una herramienta con un coste bajo</p> <p>Tiene una gran documentación</p> <p>Es un complemento perfecto para JavaScript</p>
Desventajas	<p>No es una base de datos adecuada para aplicaciones con transacciones complejas</p> <p>Es una tecnología joven</p> <p>No tiene Joins para consultas</p>
Porcentaje del mercado que controlan	3.28%

Consultas básicas en SQL

Consulta de una tabla completa

select * from materiales

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

44

Selección

select * from materiales

where clave=1000

	Clave	Descripcion	Costo
1	1000	Varilla 3/16	100.00

1

Proyección

select clave,rfc,fecha from entregan

	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

132

Reunión Natural

select * from materiales,entregan

where materiales.clave = entregan.clave

	Clave	Descripcion	Costo	Clave	RFC	Numero	Fecha	Cantidad
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
4	1010	Varilla 4/32	115.00	1010	BBBB800101	5001	2000-05-03 00:00:00.000	528.00

132

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

Reunión con criterio específico

select * from entregan,proyectos

where entregan.numero < = proyectos.numero

	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

1,188

Unión (se ilustra junto con selección)

(select * from entregan where clave=1450)

union

(select * from entregan where clave=1300)

	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

3

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

SELECT * FROM ENTREGAN WHERE clave = 1300 OR clave = 1450

Intersección (se ilustra junto con selección y proyección)

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)

	clave
1	1010

1

Diferencia (se ilustra con selección)

(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.

SELECT * FROM ENTREGAN WHERE clave != 1000

	Clave	RFC	Numero	Fecha	Cantidad
1	1010	BBBB800101	5001	2000-05-03 00:00:00.000	528.00
2	1010	BBBB800101	5018	2000-11-10 00:00:00.000	667.00
3	1010	BBBB800101	5018	2002-03-29 00:00:00.000	523.00

129

Producto cartesiano

select * from entregan,materiales

	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
4	1010	BBBB800101	5001	2000-05-03 00:00:00.000	528.00	1000	Varilla 3/16	100.00

5,808

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?: se multiplicaron los números de los registros

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 que vas a trabajar con ese formato de fechas.

set dateformat dmy

SELECT M.Descripcion FROM MATERIALES M, ENTREGAN E WHERE M.Clave = E.Clave AND Fecha >= '01/01/2000' AND Fecha < '01/01/2001'

	Descripcion
1	Varilla 3/16
2	Varilla 4/32
3	Varilla 4/32
4	Varilla 4/33

28

¿Por qué aparecen varias veces algunas descripciones de material?: Porque no utilizamos DISTINCT

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.

¿Qué resultado obtienes en esta ocasión?

set dateformat dmy

SELECT DISTINCT M.Descripcion FROM MATERIALES M, ENTREGAN E WHERE M.Clave = E.Clave AND Fecha >= '01/01/2000' AND Fecha < '01/01/2001'

	Descripcion
1	Arena
2	Block
3	Cantera rosa
4	Ladrillos grises

22

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.

```
SELECT P.Numero, P.Denominacion, E.Fecha, E.Cantidad FROM PROYECTOS P, ENTREGAN E WHERE P.Numero = E.Numero ORDER BY P.Numero, E.Fecha DESC
```

	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
4	5001	Aztlán	2000-05-03 00:00:00.000	528.00

132

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.

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 productos where Descripcion LIKE 'Si%'
```

¿Qué resultado obtienes?: Ninguno, marca error puesto que no existe una tabla llamada “productos”

Explica que hace el símbolo '%': Buscará cualquier sentencia de caracteres, incluyendo NULL que empiecen con el patrón “Si”

¿Qué sucede si la consulta fuera : LIKE 'Si' ? : Solo buscaría el patrón específico “Si”, no aceptaría nada que no fuera parecido.

¿Qué resultado obtienes?: Puesto que sigue sin existir la tabla productos, nada. Me veo en la necesidad de cambiar el nombre de “productos” a “MATERIALES”, obteniendo los siguientes resultados:

```
SELECT * FROM MATERIALES where Descripcion LIKE 'Si%'
```

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

2

```
SELECT * FROM MATERIALES where Descripcion LIKE 'Si'
```

Clave	Descripcion	Costo
-------	-------------	-------

0

Explica a qué se debe este comportamiento: Solo se está buscando específicamente el patrón “SI”

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:

```
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;
```

¿Qué resultado obtienes de ejecutar el siguiente código?: ¿Que resultado obtienes? ¿¿¿???

¿Para qué sirve DECLARE?: ¿Para crear variables?

¿Cuál es la función de @foo?: Tiene función de servir como un texto

¿Que realiza el operador SET?: ¿Asignar valores?

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 caracter que no se encuentre dentro del intervalo o del conjunto especificado.

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

Se va a poner únicamente el RFC de la tabla ENTREGAN que inicie ya sea con las letras de la A hasta la D

	RFC
1	AAAA800101
2	AAAA800101
3	AAAA800101
4	BBBB800101

72

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

Se va a poner únicamente el RFC de la tabla ENTREGAN que no empiece con la Letra A

	RFC
1	BBBB800101
2	BBBB800101
3	BBBB800101
4	CCCC800101

114

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

Se va a poner únicamente el Numero de la tabla ENTREGAN n donde el número tenga una longitud de 4 caracteres y que termine en 6

	Numero
1	5016
2	5016
3	5006

14

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;
```

	Clave	RFC	Numero	Fecha	Cantidad
1	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00
2	1010	BBBB800101	5001	2000-05-03 00:00:00.000	528.00
3	1020	CCCC800101	5002	2001-07-29 00:00:00.000	582.00
4	1030	DDDD800101	5003	1998-02-21 00:00:00.000	202.00

60

¿Cómo filtrarías rangos de fechas?: Dependiendo del rango de fecha que se desee comparar, usaría la fecha inicial y la compararía con la última fecha a ser filtrada

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] )
```

	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
4	CCCC800101	466.00	2000-08-06 00:00:00.000	5009

16

¿Qué hace la consulta?: Filtra Numero entre 5000 y 5010 y realiza una subconsulta del RFC de PROVEEDORES en donde filtra que la razón social empiece con “La...”

¿Qué función tiene el paréntesis () después de EXISTS?: Hacer 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] )
```

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

```
SELECT RFC,Cantidad, Fecha,Numero
FROM [Entregan] E
```

```
WHERE [Numero] Between 5000 and 5010 AND E.RFC NOT IN
( SELECT [RFC] FROM [Proveedores] WHERE RazonSocial NOT LIKE 'La%')
```

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
```

	Numero	Denominacion
1	5000	Vamos Mexico
2	5001	Aztecón

2

Selecciona los 2 primeros registros de la consulta, gracias a la utilización de TOP 2

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

```
SELECT TOP Numero FROM Proyectos
```

```
Msg 102, Level 15, State 1, Line 77
Incorrect syntax near 'Numero'.
```

```
Completion time: 2020-10-05T20:29:07.1218955-05:00
```

No sucede nada ya que no se especificó cuantos registros se querían limitar

Modificando la estructura de una tabla existente.

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

```
ALTER TABLE materiales ADD PorcentajeImpuesto 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 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.

¿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 M.Clave, (SUM(E.Cantidad)*M.Costo)+((SUM(E.Cantidad)*M.Costo)*M.PorcentajeImpuesto) AS "IMPORTE"
FROM MATERIALES M, ENTREGAN E WHERE E.Clave = M.Clave GROUP BY E.Clave, M.Clave
```

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.

```
CREATE VIEW AD (RFC) AS SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%;'
SELECT * FROM AD;
```

	RFC
1	AAAA800101
2	AAAA800101
3	AAAA800101
4	BBBB800101

72

```
CREATE VIEW NOT_A (RFC) AS SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%;'
SELECT * FROM NOT_A;
```

	RFC
1	BBBB800101
2	BBBB800101
3	BBBB800101
4	CCCC800101

114

```
CREATE VIEW Cuatro (Numero) AS SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
SELECT * FROM Cuatro;
```

	Numero
1	5016
2	5016
3	5006
4	5006

14

```
CREATE VIEW LIMITE (Clave,RFC,Numero,Fecha,Cantidad ) AS SELECT Clave,RFC,Numero,Fecha,Cantidad FROM
Entregan WHERE Numero Between 5000 and 5010;
SELECT * FROM LIMITE;
```

	Clave	RFC	Numero	Fecha	Cantidad
1	1000	AAAA800101	5000	1998-07-08 00:00:00.000	165.00
2	1010	BBBB800101	5001	2000-05-03 00:00:00.000	528.00
3	1020	CCCC800101	5002	2001-07-29 00:00:00.000	582.00
4	1030	DDDD800101	5003	1998-02-21 00:00:00.000	202.00

60

```
CREATE VIEW COMPLETO (RFC,Cantidad, Fecha, Numero) AS 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] )
SELECT * FROM COMPLETO;
```

	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
4	CCCC800101	466.00	2000-08-06 00:00:00.000	5009

16

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.

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".

```
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'
```

	Clave	Descripcion
1	1030	Varilla 4/33
2	1230	Cemento
3	1430	Pintura B1022

3

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

```
SELECT E.Clave, M.Descripcion FROM MATERIALES M, ENTREGAN E, PROVEEDORES P WHERE M.Clave = E.Clave AND
P.RFC = E.RFC AND P.RazonSocial = 'Acme tools'
```

Clave	Descripcion
-------	-------------

0

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

```
SELECT RFC FROM ENTREGAN WHERE FECHA BETWEEN '01/01/2000' and '01/01/2001' GROUP BY RFC HAVING
AVG(Cantidad)>=300
```

	RFC
1	BBBB800101
2	CCCC800101
3	DDDD800101
4	FFFF800101

7

El Total entregado por cada material en el año 2000.

SELECT Clave, SUM(Cantidad) FROM ENTREGAN WHERE FECHA BETWEEN '01/01/2000' and '01/01/2001' GROUP BY Clave

	Clave	(No column name)
1	1000	7.00
2	1010	1195.00
3	1030	295.00
4	1040	546.00

22

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

SELECT TOP 1 Clave, SUM(Cantidad) AS "Total" FROM ENTREGAN WHERE FECHA BETWEEN '01/01/2001' and '01/01/2002' GROUP BY CLAVE ORDER BY Total DESC

	Clave	Total
1	1020	1060.00

1

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

SELECT * FROM Materiales WHERE Descripcion LIKE '%ub%'

	Clave	Descripcion	Costo	PorcentajeImpuesto
1	1180	Recubrimiento P1001	200.00	2.36
2	1190	Recubrimiento P1010	220.00	2.38
3	1200	Recubrimiento P1019	240.00	2.40

12

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

SELECT P.Denominacion, SUM(E.Cantidad*(M.Costo*(PorcentajeImpuesto/100)+1)) AS "TOTAL" FROM ENTREGAN E, PROYECTOS P, MATERIALES M WHERE P.Numero = E.Numero AND M.Clave = E.Clave GROUP BY P.Denominacion

	Denominacion	TOTAL
1	Ampliación de la carretera a la huasteca	20204.1940000000
2	Aztecón	4777.2190000000
3	CIT Campeche	5057.0820000000
4	CIT Yucatan	33194.4880000000

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).

CREATE VIEW televisaNoCoahuila AS (SELECT DISTINCT P.Denominacion, PR.RFC, PR.RazonSocial FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND P.Denominacion LIKE 'Te1%' AND PR.RFC NOT IN (SELECT PR.RFC FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND P.Denominacion LIKE 'Edu%'));

SELECT * FROM televisaNoCoahuila;

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

2

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).

SELECT DISTINCT P.Denominacion, PR.RFC, PR.RazonSocial FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND P.Denominacion LIKE 'Te1%' AND PR.RFC NOT IN (SELECT PR.RFC FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND P.Denominacion LIKE 'Edu%');

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

2

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.

SELECT M.Costo, M.Descripcion FROM ENTREGAN E, PROYECTOS P, MATERIALES M, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND M.Clave = E.Clave AND P.Denominacion LIKE 'Te1%' AND PR.RFC IN (SELECT PR.RFC FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PR WHERE P.Numero = E.Numero AND PR.RFC = E.RFC AND P.Denominacion LIKE 'Edu%')

	Costo	Descripcion
1	50.00	Ladrillos rojos
2	34.00	Tepetate

2

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

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

```
SELECT M.Descripcion, P.Denominacion, SUM(E.Cantidad),
SUM(E.Cantidad*(M.Costo*(PorcentajeImpuesto/100)+1)) FROM ENTREGAN E, MATERIALES M, PROYECTOS P WHERE
M.Clave = E.Clave AND P.Numero = E.Numero GROUP BY M.Descripcion, P.Denominacion
```

	Descripcion	Denominacion	(No column name)	(No column name)
1	Cantera rosa	Ampliación de la carretera a la huasteca	162.00	5498.9280000000
2	Pintura C1010	Ampliación de la carretera a la huasteca	364.00	1601.6000000000
3	Recubrimiento P1010	Ampliación de la carretera a la huasteca	978.00	6098.8080000000
4	T. L. L. 4 E	Ampliación de la carretera a la huasteca	685.00	5374.5000000000

87

Muchas de estas consultas requieren la utilización de funciones agregadas...
Se recomienda que revises nuevamente la lectura.

Referencias:

<https://docs.oracle.com/cd/E19879-01/821-1040/abpak/index.html>
<https://docs.oracle.com/cd/E19879-01/821-1040/relnotessges/index.html>
<https://docs.oracle.com/cd/E19879-01/820-7426/gidij/index.html>
<https://core.ac.uk/download/pdf/13325903.pdf>
<https://www.datanyze.com/market-share/databases--272>
<https://docs.microsoft.com/en-us/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-ver15?view=sql-server-ver15>
<https://docs.microsoft.com/es-es/sql/sql-server/editions-and-components-of-sql-server-2016?redirectedfrom=MSDN&view=sql-server-ver15>
<https://www.monografias.com/trabajos73/microsoft-sql-server/microsoft-sql-server2.shtml>
<https://www.microsoft.com/es-mx/sql-server/sql-server-2019-pricing>
<https://sqlserver4b.weebly.com/ventajas-y-desventajas.html>
<https://sabecerra7.wordpress.com/2014/08/22/costos-y-soporte-tecnico-de-las-bases-de-datos/>
<https://mysqldaniel.wordpress.com/ventajas-y-desventajas/>
<http://admbditz2017.blogspot.com/2017/02/5-mongodb.html#:~:text=Requisitos%20de%20hardware&text=Los%20componentes%20principales%20de%20MongoDB,sistemas%20endian%20grandes%20o%20peque%C3%B1os>
<https://openwebinars.net/blog/ventajas-y-desventajas-de-mongodb/>