

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 1 de 9		

GUIA N° 4 – IMPLEMENTACIÓN DE BASES DE DATOS CON SQL SERVER

Objetivo: Conocer y aplicar correctamente los principales comandos, cláusulas y operadores del lenguaje SQL en el sistema gestor de base de datos Sql Server.

INTRODUCCIÓN

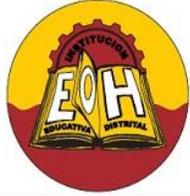
El diseño de bases de datos se descompone en tres etapas: diseño conceptual, diseño lógico y diseño físico. El diseño conceptual es el proceso por el cual se construye un modelo de la información que se utiliza en una empresa u organización, independientemente del SGBD que se vaya a utilizar para implementar el sistema y de los equipos informáticos o cualquier otra consideración física. Un modelo conceptual es un conjunto de conceptos que permiten describir la realidad mediante representaciones lingüísticas y gráficas.

Los modelos conceptuales deben poseer una serie de propiedades: expresividad, simplicidad, minimalidad y formalidad. El modelo conceptual más utilizado es el modelo entidad-relación, que se basa en los conceptos de entidades, relaciones, atributos, dominios de atributos, identificadores y jerarquías de generalización. En la metodología del diseño conceptual se construye un esquema conceptual local para cada vista de cada usuario o grupo de usuarios.

En el diseño lógico se obtiene un esquema lógico local para cada esquema conceptual local. Estos esquemas lógicos se integran después para formar un esquema lógico global que represente todas las vistas de los distintos usuarios de la empresa. Por último, en el diseño físico, se construye la implementación de la base de datos sobre un SGBD determinado. Ya que este diseño debe adaptarse al SGBD, es posible que haya que introducir cambios en el esquema lógico para mejorar las prestaciones a nivel físico.

Cada vista de usuario comprende los datos que un usuario maneja para llevar a cabo una determinada tarea. Normalmente, estas vistas corresponden a las distintas áreas funcionales de la empresa, y se pueden identificar examinando los diagramas de flujo de datos o entrevistando a los usuarios, examinando los procedimientos, informes y formularios, y observando el funcionamiento de la empresa.

Cada esquema conceptual local está formado por entidades, relaciones, atributos, dominios de atributos, identificadores y puede haber también jerarquías de generalización. Además, estos esquemas se completan documentándolos en el diccionario de datos

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 2 de 9		

LENGUAJE DE CONSULTA ESTRUCTURADO – SQL

El lenguaje de consulta estructurado SQL (por sus siglas en inglés Structured Query Language); es un lenguaje normalizado que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas. SQL

El lenguaje SQL está conformado por una serie de comandos, cláusulas, operadores y funciones de agregado que al ser combinadas genera una “sentencia” con la que se le indica al SGBD la tarea, proceso o transacción que ha de realizar. En el lenguaje SQL existen fundamentalmente dos tipos de comandos, comandos DDL y Comandos DML.

El Lenguaje de Definición de datos (LDD) (en inglés - Data Definition Language o DDL), es el que se encarga de la modificación de la estructura de los objetos de la base de datos. Se tienen cuatro comandos u operaciones básicas DDL: CREATE, ALTER, DROP y TRUNCATE.

1. CREATE Este comando crea un objeto dentro de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte.
2. ALTER Este comando permite modificar la estructura de un objeto– Se pueden agregar / quitar campos a una tabla, modificar el tipo de un campo, agregar / quitar índices a una tabla, modificar un trigger, etc.
3. DROP Este comando elimina un objeto de la base de datos. Puede ser una tabla, vista, índice, trigger, función, procedimiento o cualquier otro objeto que el motor de la base de datos soporte. Se puede combinar con la sentencia ALTER.
4. TRUNCATE Este comando trunca todo el contenido de una tabla. La ventaja sobre el comando DELETE, es que si se quiere borrar todo el contenido de la tabla, es mucho más rápido, especialmente si la tabla es muy grande, este borra la tabla y la vuelve a crear y no ejecuta ninguna Transacción.

El lenguaje de manipulación de datos (en inglés - Data Manipulation Language, o DML) es un lenguaje que permite especificarle al SGBD las tareas de consulta o manipulación de los datos a realizar. Los principales comandos DML del lenguaje SQL son SELECT, INSERT, UPDATE y DELETE

1. SELECT: Permite consultar registros en la base de datos de una o más tablas
2. INSERT: Permite cargar lotes de datos y crear nuevos registros en la base de datos
3. UPDATE: Permite modificar valores en campos y registros ya existente
4. DELETE: Permite eliminar registros de una determinada tabla de la base de datos

COMANDOS, CLAUSULAS Y OPERADORES BASICOS DE SQL SERVER

I. CREAR – SELECCIONAR BASE DE DATOS

1. CREAR BASE DE DATOS

```
CREATE DATABASE Nombre_Base_Datos;
```

2. USAR UNA BASE DE DATOS

```
USE NOMBRE_BASE_DATOS;
```

II. CREAR UNA TABLA

```
CREATE TABLE Nombre_Tabla
(
    Campo_1 Tipo1 Primary Key IDENTITY(0,1),
    Campo_2 Tipo2,
    .
    .
    Campo_N TipoN
);
```

Nota: `IDENTITY(0,1)` se utiliza solo cuando el campo se necesita definir como *AUTONUMERICO*

III. TIPO DE DATOS BASICOS EN TABLAS – SQL SERVER

Tipo	Bytes	Descripción
INT	4	Números enteros. Existen otros tipos de mayor (BIGINT) o menor longitud (SMALLINT)
REAL	8	Números reales (grandes y con decimales). Permiten almacenar todo tipo de número no entero.
NUMERIC (P,S)	Variable	Números en general donde “P” es la precisión (dígitos antes de la coma) y “S” la escala (dígitos después de la coma)
CHAR (N)	1/caracter	Alfanuméricos de longitud fija predefinida
VARCHAR(N)	1/caracter	Alfanuméricos de longitud variable hasta N
DATE/TIME/ DATETIME	3	Fecha y/o Hora, existen múltiples formatos
BIT	1	Almacenan un bit de información (verdadero o falso)
MONEY	8	Números de características monetarias

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 4 de 9		

IV. ALTERACIONES SOBRE UNA TABLA

1. ELIMINAR UNA TABLA

```
DROP TABLE Nombre_tabla;
```

2. REINICIAR UNA TABLA

```
TRUNCATE TABLE Nombre_tabla;
```

3. AGREGAR UN CAMPO A UNA TABLA

```
ALTER TABLE Nombre_tabla ADD Campo tipo;
```

4. ELIMINAR UN CAMPO DE UNA TABLA

```
ALTER TABLE Nombre_tabla DROP COLUMN Campo;
```

5. CAMBIAR EL TIPO DE UN CAMPO DE UNA TABLA

```
ALTER TABLE Nombre_tabla ALTER COLUMN Campo Nuevo_tipo;
```

6. CAMBIAR EL NOMBRE DE UNA TABLA

```
SP_RENAME 'Nombre_tabla', 'Nuevo_Nombre_tabla';
```

7. CAMBIAR EL NOMBRE DE UN CAMPO DE UNA TABLA

```
SP_RENAME ' Nombre_tabla.Nombre_Campo ', 'Nuevo_Nombre_Campo ';
```

V. RELACIONES ENTRE TABLAS

1. CREAR UNA RELACIÓN ENTRE DOS TABLAS

```
ALTER TABLE Tabla_Foranea ADD  
CONSTRAINT FK_TablaForanea_TablaPrimaria  
FOREIGN KEY (CampoFK)  
REFERENCES Tabla_Primary (CampoPK);
```

Nota:

- ✓ CampoFK corresponde al campo llave foránea de la relación
- ✓ CampoPK corresponde al campo llave primaria de la relación

2. Eliminar una relacion

```
Alter Table Tabla_Foranea DROP Constraint Nombre_Relacion;
```

Nota: Por lo general "Nombre_Relacion" es "FK_TablaForanea_TablaPrimaria"

VI. MANIPULACION DE DATOS

1. INSERTAR DATOS

```
INSERT INTO Nombre_tabla (campo1, campo2...)  
VALUES (valor 1, 'valor 2'...);
```

Nota:

- ✓ Si el campo es alfanumérico (char/varchar) el valor a ingresar debe ir entre comillas sencillas ('valor')
- ✓ Si el campo es numérico (int/numeric) el valor a ingresar va sin comillas (valor)

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 5 de 9		

2. MODIFICAR DATOS

`UPDATE Nombre_tabla SET campo1 = valor1, campo2 = 'valor2' ...`
`WHERE Condición;`

Nota:

- ✓ Si el campo a actualizar es alfanumérico (char/varchar) el valor a ingresar debe ir entre comillas sencillas ('valor')
- ✓ Si el campo a actualizar es numérico (int/numeric) el valor a ingresar va sin comillas (valor)
- ✓ La condición puede ser simple (Campo = Valor) o compuesta (Campo1 = Valor1 AND Campo2 = Valor2 ...)

3. ELIMINAR DATOS

`DELETE FROM Nombre_tabla WHERE Condición;`

Nota:

- ✓ La condición puede ser simple (Campo = Valor) o compuesta (Campo1 = Valor1 AND Campo2 = Valor2 ...)

4. CONSULTAR DATOS

`4.1 SELECT * FROM Nombre_tabla;`

`4.2 SELECT campo1,campo2,...campoN FROM Nombre_tabla WHERE Condición;`

Nota:

- ✓ La condición puede ser simple (Campo = Valor) o compuesta (Campo1 = Valor1 AND Campo2 = Valor2 ...)

4.1 CONSULTAS ESPECIALIZADAS

Las consultas en SQL Server se pueden hacer de forma especializada usando los comandos mostrados a continuación:

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 6 de 9		

<p><u>AND/OR (Para Unir N Condiciones)</u> SELECT "nombre_columna" FROM "nombre_tabla" WHERE "condición 1" {{AND OR} "condición n"}}</p> <p><u>DISTINCT (Para quitar valores repetidos en una columna)</u> SELECT DISTINCT "nombre_columna" FROM "nombre_tabla"</p> <p><u>BETWEEN (Para Rangos de valores)</u> SELECT "nombre_columna" FROM "nombre_tabla" WHERE "nombre_columna" BETWEEN 'valor1' AND 'valor2'</p> <p><u>ORDER BY (Para Ordenar De Forma Ascendente o Descendente)</u> SELECT "nombre_columna" FROM "nombre_tabla" [WHERE "condición"] ORDER BY "nombre_columna" [ASC, DESC]</p> <p><u>COUNT (Para Contar Registros)</u> SELECT COUNT("nombre_columna") FROM "nombre_tabla"</p>	<p><u>IN (Para Filtrar Por Valores Fijos)</u> SELECT "nombre_columna" FROM "nombre_tabla" WHERE "nombre_columna" IN ('valor1', 'valor2', ...)</p> <p><u>GROUP BY (Para Formar Grupos)</u> SELECT "nombre_columna 1", SUM ("nombre_columna 2") FROM "nombre_tabla" GROUP BY "nombre_columna 1"</p> <p><u>HAVING (Para Aplicar Una Condición a Un Grupo)</u> SELECT "nombre_columna 1", SUM("nombre_columna 2") FROM "nombre_tabla" GROUP BY "nombre_columna 1" HAVING (condición de función aritmética)</p> <p><u>LIKE (Para Filtrar por un patrón, por ejemplo por la primera letra)</u> SELECT "nombre_columna" FROM "nombre_tabla" WHERE "nombre_columna" LIKE {patrón}</p>
---	---

TALLER

1. Abra el software manejador de SQL SERVER y usando los comandos mostrados en esta guía implemente una base de datos que corresponda con el Modelo Relacional propuesto
2. Una vez terminado realice las siguientes tareas sobre las tablas correspondientes:
 - a) Ingresar, consultar, actualizar y eliminar los datos de la institución educativa(Colegio)
 - b) Ingresar, consultar, actualizar y eliminar los datos de la biblioteca
 - c) Ingresar, consultar, actualizar y eliminar materias o asignaturas
 - d) Ingresar, consultar, actualizar y eliminar libros
 - e) Ingresar, consultar, actualizar y eliminar los clientes (estudiantes)
 - f) Ingresar, consultar, actualizar y eliminar préstamos de libros



GUÍA DE TRABAJO N° 4

Educación Media Fortalecida
SED/SENA

Programación de Software
Grado 11

Ing. Néstor Raúl Suarez Perpiñan

Página 7 de 9



	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 8 de 9		

3. Cuando cada una de las tablas tengan información almacenada ejecute las consultas que se muestran a continuación, explique qué datos arrojan y saque sus propias conclusiones sobre en qué situaciones o bajo qué circunstancias se podrían usar cada una de las consultas.

✓ **Consulta N° 1**

```

Select
Estudiantes.identificacion,
Estudiantes.Nombre,
Estudiantes.Apellido,
Estudiantes.grado,
Colegios.Nombre
FROM
Estudiantes,Colegios

Where Estudiantes.NitColegio = Colegios.NitColegio
AND Estudiantes.Grado = 1101
AND Colegios.Nombre = 'Colegio Enrique Olaya Herrera'

```

✓ **Consulta N° 2**

```

Select
Count(*) as Numero_De_Estudiantes
FROM
Estudiantes,Colegios
Where Estudiantes.NitColegio = Colegios.NitColegio
AND Estudiantes.Grado = 1101
AND Colegios.Nombre = 'Colegio Enrique Olaya Herrera'

```

✓ **Consulta N° 3**

```

Select
Estudiantes.identificacion,
Estudiantes.Nombre,
Estudiantes.Apellido,
Estudiantes.grado,
Colegios.Nombre
FROM
Estudiantes,Colegios
Where Estudiantes.NitColegio = Colegios.NitColegio
AND Estudiantes.Apellido Like '%o%'

```

	GUÍA DE TRABAJO N° 4		
	Educación Media Fortalecida SED/SENA	Programación de Software Grado 11	
	Ing. Néstor Raúl Suarez Perpiñan Página 9 de 9		

✓ **Consulta N° 4**

Select

*

FROM

Libros, Materias, Biblioteca

Where Libros.Id_materia = Materias.Id_Materia

AND Libros.Id_Biblioteca = Biblioteca.Id_Biblioteca

✓ **Consulta N° 5**

Select

Libros.nombre As Nombre_Libro,

Libros.autor,

Libros.NumEjemplares,

Materias.nombre As Nombre_Materia,

Biblioteca.Nombre AS Nombre_Biblioteca

FROM

Libros, Materias, Biblioteca

Where Libros.Id_materia = Materias.Id_Materia

AND Libros.Id_Biblioteca = Biblioteca.Id_Biblioteca

✓ **Consulta N° 6**

Select

Count(*) as Cantidad_Libros,

Max(Libros.Numejemplares) AS Mayor_Numero_Ejemplares,

Min(Libros.Numejemplares) AS Menor_Numero_Ejemplares,

Sum(Libros.Numejemplares) AS Total_Ejemplares,

Avg(Libros.Numejemplares) AS Promedio_Ejemplares

FROM

Libros, Materias, Biblioteca

Where Libros.Id_materia = Materias.Id_Materia

AND Libros.Id_Biblioteca = Biblioteca.Id_Biblioteca
