

SQL (Structured Query Language)

Introducción

SQL (Structured Query Language) es un lenguaje estándar utilizado para gestionar y manipular bases de datos. En SQL Server, se utiliza tanto para la definición de estructuras de bases de datos como para la manipulación de datos, lo que incluye consultas, inserciones, actualizaciones y eliminaciones. Este módulo cubre las principales categorías de comandos SQL, incluyendo **DDL (Data Definition Language)**, **DML (Data Manipulation Language)**, y más.

DDL (Data Definition Language)

El **DDL** es un subconjunto de SQL que se utiliza para definir la estructura de una base de datos. Permite crear, modificar y eliminar objetos de la base de datos como tablas, índices y vistas.

CREATE TABLE

El comando CREATE TABLE se utiliza para definir una nueva tabla en la base de datos.

Restricciones

Las restricciones son reglas aplicadas a las columnas de una tabla para asegurar la integridad de los datos. Ejemplos de restricciones incluyen PRIMARY KEY, FOREIGN KEY, NOT NULL, y CHECK.

- **Ejemplo de restricción CHECK:**
- CREATE TABLE Empleados (
• ID INT PRIMARY KEY,
• Nombre VARCHAR(100) NOT NULL,
• Edad INT CHECK (Edad >= 18)
•);

Tipos de Datos

Los tipos de datos definen la naturaleza de los datos que se almacenarán en cada columna. Algunos tipos comunes son INT, VARCHAR, DATE, y DECIMAL.

CREATE INDEX

El comando CREATE INDEX se utiliza para crear un índice que acelera las consultas en una o más columnas de una tabla.

- **Ejemplo:**
- CREATE INDEX idx_edad ON Empleados (Edad);

DROP INDEX

El comando DROP INDEX elimina un índice previamente creado.

- **Ejemplo:**
- DROP INDEX idx_edad ON Empleados;

ALTER TABLE

El comando ALTER TABLE se utiliza para modificar una tabla existente, como agregar o eliminar columnas.

- **Ejemplo:**
- ALTER TABLE Empleados ADD Salario DECIMAL(10, 2);

DROP TABLE

El comando DROP TABLE elimina una tabla y todos sus datos de la base de datos.

- **Ejemplo:**
- DROP TABLE Empleados;

CREATE VIEW

El comando CREATE VIEW se utiliza para crear una vista, que es una consulta predefinida que presenta datos de una o varias tablas.

- **Ejemplo:**
- CREATE VIEW VistaEmpleados AS
- SELECT Nombre, Edad FROM Empleados WHERE Edad > 30;

ALTER VIEW

El comando ALTER VIEW permite modificar una vista existente.

- **Ejemplo:**
- ALTER VIEW VistaEmpleados AS
- SELECT Nombre, Edad, Salario FROM Empleados WHERE Edad > 30;

DROP VIEW

El comando DROP VIEW elimina una vista.

- **Ejemplo:**
- DROP VIEW VistaEmpleados;

CREATE TRIGGER

El comando CREATE TRIGGER se utiliza para crear un disparador (trigger) que ejecuta automáticamente una acción cuando ocurre un evento específico en la base de datos.

- **Ejemplo:**
- CREATE TRIGGER trg_EmpleadoInsert
- ON Empleados
- FOR INSERT
- AS
- BEGIN
- PRINT 'Nuevo empleado insertado';
- END;

ALTER TRIGGER

El comando ALTER TRIGGER modifica un disparador existente.

- **Ejemplo:**
- ALTER TRIGGER trg_EmpleadoInsert
- ON Empleados
- FOR INSERT, DELETE
- AS
- BEGIN
- PRINT 'Empleado insertado o eliminado';
- END;

DROP TRIGGER

El comando DROP TRIGGER elimina un disparador.

- **Ejemplo:**
- DROP TRIGGER trg_EmpleadoInsert;

DML (Data Manipulation Language)

El **DML** se utiliza para manipular los datos dentro de las tablas, permitiendo operaciones como consultas, inserciones, actualizaciones y eliminaciones.

SELECT

El comando SELECT se utiliza para consultar datos de una o más tablas.

5.3.1.1 Cláusula FROM

La cláusula FROM se utiliza para especificar las tablas de las cuales se obtendrán los datos.

- **Ejemplo:**
- SELECT Nombre, Edad FROM Empleados;

GROUP BY - HAVING

GROUP BY agrupa los resultados de una consulta según una o más columnas, mientras que HAVING filtra los grupos.

- **Ejemplo:**
- SELECT Edad, COUNT(*) FROM Empleados
- GROUP BY Edad
- HAVING COUNT(*) > 1;

COMPUTE

La cláusula COMPUTE realiza cálculos adicionales sobre los resultados.

- **Ejemplo:**
- SELECT Edad, COUNT(*) FROM Empleados
- GROUP BY Edad
- COMPUTE SUM(Edad);

Funciones de Agrupación

SQL Server ofrece funciones de agregación como SUM, MAX, MIN, AVG, y COUNT para realizar cálculos sobre grupos de datos.

- **Ejemplo:**
- SELECT AVG(Edad) FROM Empleados;

Funciones Generales

Funciones como CONVERT, GETDATE, y DATEDIFF son útiles para la conversión de datos y el manejo de fechas.

- **Ejemplo:**
- SELECT GETDATE();

Funciones de Sistema

Las funciones de sistema como ISNULL y COALESCE son usadas para manejar valores nulos.

- **Ejemplo:**
- SELECT ISNULL(Salario, 0) FROM Empleados;

Consultas Avanzadas

SQL permite realizar consultas avanzadas utilizando operaciones como UNION, INTERSECT, EXCEPT, subconsultas y tablas temporales.

- **Ejemplo de UNION:**
- SELECT Nombre FROM Empleados

- UNION
- SELECT Nombre FROM ExEmpleados;

INSERT

El comando INSERT se utiliza para agregar nuevos registros en una tabla.

- **Ejemplo:**
- INSERT INTO Empleados (ID, Nombre, Edad) VALUES (1, 'Juan', 30);

Bulk Insert

BULK INSERT se utiliza para importar grandes volúmenes de datos desde un archivo.

- **Ejemplo:**
- BULK INSERT Empleados
- FROM 'C:\Datos\empleados.txt'
- WITH (FIELDTERMINATOR = ',', ROWTERMINATOR = '\n');

UPDATE

El comando UPDATE se utiliza para modificar los registros existentes en una tabla.

- **Ejemplo:**
- UPDATE Empleados SET Edad = 35 WHERE ID = 1;

DELETE

El comando DELETE elimina registros de una tabla.

- **Ejemplo:**
- DELETE FROM Empleados WHERE Edad < 18;

TRUNCATE TABLE

TRUNCATE TABLE elimina todos los registros de una tabla, pero sin eliminar la estructura de la tabla.

- **Ejemplo:**
- TRUNCATE TABLE Empleados;

Este módulo cubre los aspectos fundamentales de SQL, permitiendo tanto la definición como la manipulación de datos en una base de datos.