



## Sesión 21

Tema	Lenguaje de Definición de Datos - DDL
Propósito	Dar a conocer a los participantes los fundamentos y técnicas de uso del sub-lenguaje SQL para la definición de datos, mediante la correspondiente presentación y demostración del docente en una sesión expositiva-demonstrativa.
Fecha	C.17.11.2025
Hora	17:00

### ¿Qué es el DDL?

El **Lenguaje de Definición de Datos (DDL)** es un subconjunto de SQL dedicado exclusivamente a la definición, modificación y eliminación de estructuras de base de datos. A diferencia del DML (Lenguaje de Manipulación de Datos) que trabaja con los datos, el DDL trabaja con la **estructura** donde se almacenan esos datos.

#### Características Principales:

- **Definición de esquemas:** Crear la arquitectura de la base de datos.
- **Metadatos:** Trabaja con información sobre los datos.
- **Auto-commit:** Las operaciones DDL confirman automáticamente las transacciones.
- **Impacto estructural:** Cambia la organización de la base de datos.

### Comandos DDL Principales

#### 1. CREATE - Crear objetos

-- *Crear base de datos*

```
CREATE DATABASE nombre_base_datos;
```

-- *Crear tabla*

```
CREATE TABLE nombre_tabla (
    columna1 tipo_dato restricciones,
    columna2 tipo_dato restricciones,
    ...
);
```



## 2. ALTER - Modificar estructura

-- Agregar columna

```
ALTER TABLE tabla ADD columna tipo_dato;
```

-- Modificar columna

```
ALTER TABLE tabla MODIFY columna nuevo_tipo_dato;
```

-- Eliminar columna

```
ALTER TABLE tabla DROP COLUMN columna;
```

## 3. DROP - Eliminar objetos

-- Eliminar tabla

```
DROP TABLE nombre_tabla;
```

-- Eliminar base de datos

```
DROP DATABASE nombre_base_datos;
```

## 4. TRUNCATE - Vaciar tabla

-- Eliminar todos los registros

```
TRUNCATE TABLE nombre_tabla;
```

## 5. RENAME - Renombrar objetos

-- Renombrar tabla

```
RENAME TABLE viejo_nombre TO nuevo_nombre;
```



## Ejemplos Prácticos

### Ejemplo 1: Creación de Base de Datos y Tablas

**Enunciado:** Crear una base de datos "Universidad" con tablas para Estudiantes y Cursos.

**Solución:**

-- *Crear base de datos*

```
CREATE DATABASE Universidad;
```

```
USE Universidad;
```

-- *Tabla Estudiantes*

```
CREATE TABLE Estudiantes (
    id_estudiante INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    apellido VARCHAR(50) NOT NULL,
    fecha_nacimiento DATE,
    email VARCHAR(100) UNIQUE,
    activo BOOLEAN DEFAULT TRUE
);
```

-- *Tabla Cursos*

```
CREATE TABLE Cursos (
    id_curso INT PRIMARY KEY AUTO_INCREMENT,
    nombre_curso VARCHAR(100) NOT NULL,
    creditos INT DEFAULT 3,
    profesor VARCHAR(100),
    cupo_maximo INT CHECK (cupo_maximo > 0)
);
```



## Ejemplo 2: Modificación de Estructura

**Enunciado:** Agregar una columna "semestre" a la tabla Estudiantes y modificar el tipo de dato de "creditos" en Cursos.

**Solución:**

-- Agregar columna semestre

```
ALTER TABLE Estudiantes ADD semestre INT DEFAULT 1;
```

-- Modificar tipo de dato y agregar restricción

```
ALTER TABLE Cursos MODIFY creditos DECIMAL(3,1) CHECK (creditos BETWEEN 1 AND 10);
```

-- Agregar columna fecha\_creacion

```
ALTER TABLE Cursos ADD fecha_creacion DATE DEFAULT CURRENT_DATE;
```

## Ejemplo 3: Gestión de Restricciones y Claves

**Enunciado:** Crear una tabla de Matrículas que relacione Estudiantes con Cursos.

**Solución:**

```
CREATE TABLE Matriculas (
```

```
    id_matricula INT PRIMARY KEY AUTO_INCREMENT,
```

```
    id_estudiante INT,
```

```
    id_curso INT,
```

```
    fecha_matricula DATE DEFAULT CURRENT_DATE,
```

```
    calificacion DECIMAL(3,2) CHECK (calificacion BETWEEN 0 AND 10),
```

-- Claves foráneas

```
    FOREIGN KEY (id_estudiante) REFERENCES Estudiantes(id_estudiante),
```

```
    FOREIGN KEY (id_curso) REFERENCES Cursos(id_curso),
```



-- Restricción única

```
UNIQUE KEY unique_matricula (id_estudiante, id_curso)
```

```
);
```

#### Ejemplo 4: Operaciones de Mantenimiento

**Enunciado:** Realizar diversas operaciones DDL sobre las tablas existentes.

**Solución:**

-- Renombrar tabla

```
RENAME TABLE Matriculas TO Inscripciones;
```

-- Agregar índice para mejorar rendimiento

```
CREATE INDEX idx_estudiante_email ON Estudiantes(email);
```

-- Eliminar columna

```
ALTER TABLE Estudiantes DROP COLUMN activo;
```

-- Vaciar tabla (eliminar todos los registros)

```
TRUNCATE TABLE Inscripciones;
```

-- Eliminar tabla

```
DROP TABLE Inscripciones;
```



### Ejemplo 5: Estructura Compleja con Varias Restricciones

**Enunciado:** Crear una base de datos para biblioteca con tablas normalizadas.

**Solución:**

```
CREATE DATABASE Biblioteca;
```

```
USE Biblioteca;
```

```
CREATE TABLE Autores (
```

```
    id_autor INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100) NOT NULL,  
    nacionalidad VARCHAR(50),  
    fecha_nacimiento DATE
```

```
);
```

```
CREATE TABLE Libros (
```

```
    id_libro INT PRIMARY KEY AUTO_INCREMENT,  
    titulo VARCHAR(200) NOT NULL,  
    isbn VARCHAR(20) UNIQUE,  
    año_publicacion INT,  
    id_autor INT,  
    precio DECIMAL(8,2) CHECK (precio >= 0),  
    stock INT DEFAULT 0 CHECK (stock >= 0),
```

```
    FOREIGN KEY (id_autor) REFERENCES Autores(id_autor)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE
```

```
);
```



```
CREATE TABLE Categorias (
    id_categoria INT PRIMARY KEY AUTO_INCREMENT,
    nombre_categoria VARCHAR(50) NOT NULL UNIQUE,
    descripcion TEXT
);
```

-- Tabla intermedia para relación muchos a muchos

```
CREATE TABLE Libro_Categorias (
    id_libro INT,
    id_categoria INT,
    PRIMARY KEY (id_libro, id_categoria),
    FOREIGN KEY (id_libro) REFERENCES Libros(id_libro),
    FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria)
);
```

### 💡 Buenas Prácticas en DDL

1. **Nomenclatura consistente:** Usar snake\_case o camelCase consistentemente
2. **Claves primarias:** Siempre definir PRIMARY KEY
3. **Restricciones:** Aplicar NOT NULL, UNIQUE, CHECK cuando sea necesario
4. **Claves foráneas:** Definir ON DELETE y ON UPDATE apropiados
5. **Documentación:** Comentar tablas y columnas importantes
6. **Backup:** Siempre hacer backup antes de operaciones DDL destructivas



## Bibliografía Recomendada

- "Fundamentos de Bases de Datos" de Abraham Silberschatz, Henry F. Korth y S. Sudarshan
- "Sistemas de Bases de Datos: un enfoque práctico" de Thomas M. Connolly y Carolyn Begg
- "Desarrollo de Bases de Datos: casos prácticos desde el análisis a la implementación" de Dolores Cuadra, Elena Castro, Ana M. Iglesias
- "Tecnología y Diseño de Bases de Datos" de Marcos, C. Calero y B. Vela
- <https://docs.oracle.com/en/database/>