Memoria Técnica - Capítulos 1 y 2 del SQL eBook de Steve Nouri

# Capítulo 1: Introducción a SQL

SQL (Structured Query Language) es un lenguaje utilizado para gestionar datos en sistemas de bases de datos relacionales (RDBMS).  
Permite la definición de estructuras, la manipulación de datos, la consulta y el control de acceso a la información.

Evolución del estándar:  
- SQL-86: Primera versión ANSI.  
- SQL-92: Ampliación importante con subconsultas y operaciones JOIN.  
- SQL:1999: Introduce expresiones regulares y CTEs.  
- SQL:2003 - SQL:2016: Se incluyen XML, JSON, funciones analíticas y temporales.

Sub-lenguajes de SQL:  
- DDL (Data Definition Language): Define la estructura (CREATE, ALTER, DROP).  
- DML (Data Manipulation Language): Manipula los datos (INSERT, UPDATE, DELETE).  
- DCL (Data Control Language): Administra permisos (GRANT, REVOKE).  
- TCL (Transaction Control Language): Gestiona transacciones (COMMIT, ROLLBACK).

Operaciones CRUD:  
- CREATE: INSERT  
- READ: SELECT  
- UPDATE: UPDATE  
- DELETE: DELETE

# Capítulo 2: ALTER TABLE

El comando ALTER TABLE permite modificar la estructura de una tabla existente.  
  
Operaciones comunes:  
- ADD COLUMN: Agrega nuevas columnas.  
- DROP COLUMN: Elimina columnas.  
- ALTER COLUMN: Modifica tipo de datos o restricciones.  
- ADD CONSTRAINT: Agrega restricciones como PRIMARY KEY o DEFAULT.  
- DROP CONSTRAINT: Elimina restricciones existentes.

Ejemplo: Añadir columnas

ALTER TABLE Empleados ADD FechaIngreso DATE NOT NULL DEFAULT GETDATE(), FechaNacimiento DATE NULL;

Ejemplo: Eliminar una columna

ALTER TABLE Empleados DROP COLUMN Salario;

Ejemplo: Añadir restricción DEFAULT

ALTER TABLE Empleados ADD CONSTRAINT SalarioPorDefecto DEFAULT (100) FOR Salario;

Ejemplo: Modificar columna

ALTER TABLE Empleados ALTER COLUMN FechaIngreso DATETIME NOT NULL DEFAULT GETDATE();

Ejemplo: Agregar clave primaria

ALTER TABLE Empleados ADD CONSTRAINT PK\_Empleado PRIMARY KEY (ID);

# Contenido Adicional

---  
  
Capítulo 3: Buenas Prácticas con ALTER TABLE  
  
Al trabajar con ALTER TABLE en entornos productivos, es fundamental seguir ciertas prácticas para evitar errores y garantizar la integridad de los datos.  
  
1. Realizar respaldos antes de aplicar ALTER TABLE.  
2. Verificar la existencia de restricciones antes de eliminarlas.  
3. Planificar los cambios en horarios de baja actividad.  
4. Documentar todos los cambios aplicados a las estructuras.  
  
---  
  
Capítulo 4: Tipos de Restricciones  
  
Las restricciones son reglas aplicadas a las columnas para garantizar la validez de los datos. Algunas comunes:  
  
- PRIMARY KEY: Asegura unicidad y no permite nulos.  
- FOREIGN KEY: Mantiene integridad referencial con otra tabla.  
- UNIQUE: Evita valores duplicados en una columna.  
- NOT NULL: Obliga a que la columna tenga valor.  
- DEFAULT: Asigna un valor por defecto si no se especifica.  
- CHECK: Define un rango válido de valores.  
  
---  
  
Capítulo 5: Comparación entre Motores de Base de Datos  
  
A continuación, se muestra una comparación básica sobre el soporte de ALTER TABLE entre algunos motores populares:  
  
| Motor | Soporta ALTER TABLE | Comentarios adicionales |  
|---------------|---------------------|---------------------------------------|  
| MySQL | Sí | Muy flexible y ampliamente usado |  
| PostgreSQL | Sí | Gran soporte para cambios complejos |  
| SQL Server | Sí | Soporta operaciones en lotes |  
| SQLite | Limitado | No soporta DROP COLUMN directamente |  
  
---  
  
Capítulo 6: Ejemplo práctico completo  
  
Caso: Agregar historial de contratación a la tabla de empleados  
  
```sql  
ALTER TABLE Empleados  
ADD FechaContratacion DATETIME NOT NULL DEFAULT GETDATE();  
ALTER TABLE Empleados  
ADD CONSTRAINT PK\_Empleados PRIMARY KEY (ID);  
```  
  
---  
  
Capítulo 7: Consideraciones de rendimiento  
  
Al modificar grandes tablas con ALTER TABLE, se recomienda:  
- Usar índices para acelerar operaciones dependientes.  
- Aplicar ALTER por lotes si el sistema lo permite.  
- Validar mediante EXPLAIN PLAN o herramientas equivalentes.  
  
---  
  
Capítulo 8: Conclusión  
  
Los comandos SQL y especialmente ALTER TABLE son herramientas poderosas para la evolución de esquemas de datos. Comprender su sintaxis, restricciones y efectos secundarios es esencial para garantizar una administración segura y eficiente de bases de datos relacionales.