

Tema 39

Lenguajes para definición y manipulación de datos en BBDD relacionales

1. Introducción

- SQL (Structured Query Language) nace en los años 70.
- Lenguaje **declarativo**: se indica *qué* se desea obtener, no *cómo*.
- Estándar universal en bases de datos relacionales.
- Se integra con lenguajes como:
 - Python, R, JavaScript, Spark...

1.2 Componentes de SQL

Subtipo	Función	Ejemplos
DDL	Definición de estructuras	CREATE TABLE , ALTER TABLE
DML	Manipulación de datos	INSERT , UPDATE , DELETE
DQL	Consultas	SELECT , JOIN , GROUP BY
DCL	Control de permisos	GRANT , REVOKE , RLS
TCL	Transacciones	BEGIN , COMMIT , ROLLBACK

1.3 DDL – Definición de datos

- Define estructuras: tablas, vistas, tipos personalizados.

Ejemplo:

```
CREATE TABLE empleados (  
  id INT PRIMARY KEY,  
  nombre VARCHAR(100),  
  salario DECIMAL(8,2)  
);
```

- Admite tipos avanzados: JSON, GEOMETRY
- Usa TRIGGERS , VISTAS MATERIALIZADAS
- Rol: arquitecto de la base de datos

1.4 DML – Manipulación de datos

- Inserta, actualiza y elimina registros

Ejemplo:

```
INSERT INTO empleados (id, nombre, salario)
VALUES (1, 'Ana', 2500.00);
```

- Window functions:

```
SELECT nombre, salario,
AVG(salario) OVER (PARTITION BY depto)
FROM empleados;
```

1.5 DQL – Lenguaje de consulta

Consulta y extracción de datos con:

- `SELECT` , `WHERE` , `JOIN` , `GROUP BY` , `HAVING`
- Subconsultas, `WITH` (CTEs)

Ejemplo:

```
SELECT nombre  
FROM empleados  
WHERE salario > 2000;
```

1.6 DCL – Control de acceso

- Gestiona permisos sobre objetos de la BD

Ejemplo:

GRANT SELECT ON empleados TO lectura;

- Seguridad avanzada:
 - Row-Level Security (RLS)
 - Column Masking

1.7 TCL – Transacciones

- Agrupan operaciones para garantizar atomicidad

BEGIN;

UPDATE cuentas SET saldo = saldo - 100 WHERE id = 1;

UPDATE cuentas SET saldo = saldo + 100 WHERE id = 2;

COMMIT;

- Evita inconsistencias en operaciones críticas

1.8 Integración externa

- SQL embebido en:
 - Python (`sqlite3` , `sqlalchemy`)
 - Java (`JDBC`)
 - C / R / JS
- Soporte en ORMs: Django, Hibernate
- Conecta **lógica de negocio** con la **persistencia de datos**

1.9 Alternativas y extensiones

- **NoSQL:** MongoDB, Cassandra
 - Escalabilidad horizontal, flexibilidad
- **NewSQL:** TiDB, CockroachDB
 - SQL + ACID distribuido
- **GraphQL:** desde frontend
- **DSLs:**
 - `pandas.query()` , `dplyr` , `Flink SQL`



1.10 Aplicaciones modernas

- **Business Intelligence:**
 - Power BI, Tableau
- **Aprendizaje automático:**
 - Preprocesamiento de datos
- **Learning Analytics:**
 - Análisis de comportamiento educativo
- **Fintech / Sanidad / Educación:**
 - Gestión segura y masiva de datos estructurados

1.11 Conclusión técnica

- SQL es **fundamental** en entornos técnicos y de negocio.
- Su dominio permite:
 - Manipular datos
 - Crear estructuras robustas
 - Aplicar seguridad, transacciones, eficiencia
- Habilidades clave para perfiles de:
 - Desarrollo, análisis de datos, sistemas y seguridad