

DATA DEFINITION LANGUAGE (DDL)

DDL è l'acronimo di **Data Definition Language** (Linguaggio di Definizione Dati). È un sottoinsieme del linguaggio SQL (Structured Query Language) che si concentra sulla **definizione e gestione della struttura del database**. In altre parole, il DDL è utilizzato per **creare, modificare ed eliminare oggetti del database**, come:

- **Database stessi:** Creare e eliminare database.
- **Tabelle:** Creare, modificare (aggiungere/rimuovere colonne, cambiare tipi di dati, etc.), e eliminare tabelle.
- **Indici:** Creare e eliminare indici per migliorare le prestazioni delle query.
- **Viste:** Creare e eliminare viste (tabelle virtuali basate su query).
- **Utenti e permessi:** Gestire utenti e i loro permessi di accesso al database (anche se a volte questo è considerato parte del DCL - Data Control Language, c'è sovrapposizione).
- **Vincoli:** Definire vincoli di integrità dei dati (chiavi primarie, chiavi esterne, vincoli di unicità, etc.).
- **Sequenze:** Creare e gestire sequenze per generare valori univoci.
- **Sinonimi:** Creare sinonimi per oggetti del database per semplificare la referenziazione.

Le principali istruzioni DDL in SQL sono:

- **CREATE:** Utilizzata per **creare nuovi oggetti** del database. Esempi: CREATE DATABASE, CREATE TABLE, CREATE INDEX, CREATE VIEW.
- **ALTER:** Utilizzata per **modificare la struttura di oggetti esistenti**. Esempi: ALTER TABLE (per aggiungere, modificare o eliminare colonne), ALTER DATABASE.
- **DROP:** Utilizzata per **eliminare oggetti** del database in modo permanente. Esempi: DROP TABLE, DROP DATABASE, DROP INDEX, DROP VIEW.
- **TRUNCATE:** Utilizzata per **eliminare *tutti i dati da una tabella***, ma mantenendo la struttura della tabella intatta. È più veloce di DELETE per eliminare tutti i dati e non registra le singole eliminazioni nel log delle transazioni (in alcuni DBMS).
- **RENAME:** Utilizzata per **rinominare oggetti** del database. Esempio: RENAME TABLE.

Caratteristiche Principali del DDL:

1. **Definizione della Struttura del Database:** La caratteristica fondamentale del DDL è la sua focalizzazione sulla **definizione della struttura del database**. Non si occupa direttamente dei dati contenuti nelle tabelle (quello è il ruolo del DML), ma piuttosto di come le tabelle e gli altri oggetti sono organizzati e definiti.
2. **Operazioni sulla Schema:** Le operazioni DDL modificano la **schema** del database, ovvero la sua struttura logica. Ogni comando DDL altera il modo in cui i dati sono organizzati e gestiti dal DBMS.
3. **Operazioni Generalmente Irreversibili (o Difficili da Invertire):** Comandi come DROP sono **irreversibili** e comportano la perdita definitiva degli oggetti (e dei dati contenuti, nel caso di tabelle). Anche ALTER TABLE può essere difficile da invertire in alcuni casi, specialmente se comporta la rimozione di colonne o vincoli. Pertanto, è cruciale utilizzare i comandi DDL con cautela e pianificazione.
4. **Operazioni Automaticamente Commesse (Auto-Commit):** La maggior parte dei sistemi di gestione di database relazionali esegue i comandi DDL in modalità **auto-commit**. Questo significa che le modifiche apportate da un comando DDL vengono **immediatamente e permanentemente salvate** nel database. Non è necessario un comando COMMIT esplicito come per le transazioni DML (anche se alcuni DBMS permettono di incorporare DDL in transazioni più ampie).
5. **Spesso Richiede Privilegi di Amministrazione:** L'esecuzione di comandi DDL che modificano la struttura del database (come CREATE DATABASE, DROP TABLE, ALTER TABLE) spesso richiede **privilegi di amministrazione** o permessi specifici. Questo è per proteggere l'integrità e la stabilità del database e prevenire modifiche non autorizzate alla sua struttura.
6. **Parte Standard del Linguaggio SQL:** Il DDL è una parte **integrante e standardizzata del linguaggio SQL**. Le istruzioni DDL sono generalmente **compatibili tra diversi sistemi di database relazionali** (con possibili piccole variazioni sintattiche specifiche del dialetto SQL).
7. **Utilizzato per l'Impostazione Iniziale e l'Evoluzione del Database:** Il DDL è fondamentale per due fasi chiave nella vita di un database:
 - **Impostazione Iniziale:** Quando si crea un nuovo database, il DDL viene utilizzato per definire la struttura iniziale, creando database, tabelle, indici, vincoli, etc.
 - **Evoluzione del Database:** Nel tempo, le esigenze applicative possono cambiare, richiedendo modifiche alla struttura del database. Il DDL viene utilizzato per adattare la schema, aggiungendo nuove tabelle, colonne, indici, o modificando quelli esistenti.

8. **Fondamentale per la Progettazione del Database:** Il DDL è strettamente legato alla **progettazione del database**. Una buona progettazione del database si traduce in un DDL ben strutturato e efficiente, che riflette correttamente il modello dei dati e supporta le esigenze dell'applicazione.

In sintesi, il DDL è un componente **essenziale** di SQL per la **definizione e gestione dell'architettura del database**. Permette di creare, modificare e gestire tutti gli oggetti che compongono la struttura del database, garantendo una base solida e ben definita per la memorizzazione e la gestione dei dati.