

Elenco di Comandi SQL

1. Comandi per la gestione dei dati (DML - Data Manipulation Language)

- SELECT - Recupera dati da una tabella o più tabelle.
- INSERT INTO - Aggiunge nuovi record in una tabella.
- UPDATE - Modifica i dati esistenti in una tabella.
- DELETE - Elimina dati da una tabella.

2. Comandi per la definizione dello schema (DDL - Data Definition Language)

- CREATE TABLE - Crea una nuova tabella.
- ALTER TABLE - Modifica una tabella esistente (aggiunge, modifica o rimuove colonne).
- DROP TABLE - Elimina una tabella esistente (con tutti i suoi dati).
- CREATE DATABASE - Crea un nuovo database.
- DROP DATABASE - Elimina un intero database.

3. Comandi per il controllo dei permessi (DCL - Data Control Language)

- GRANT - Concede permessi a un utente.
- REVOKE - Revoca permessi concessi a un utente.

4. Comandi per il controllo delle transazioni (TCL - Transaction Control Language)

- START TRANSACTION - Inizia una nuova transazione.
- COMMIT - Applica i cambiamenti effettuati nella transazione.
- ROLLBACK - Annulla i cambiamenti effettuati nella transazione.
- SAVEPOINT - Crea un punto intermedio nella transazione.
- RELEASE SAVEPOINT - Elimina un punto intermedio creato con SAVEPOINT.

5. Comandi per la gestione delle tabelle (operazioni avanzate)

- TRUNCATE TABLE - Elimina tutti i dati di una tabella, ma mantiene la struttura.
- RENAME TABLE - Rinomina una tabella.

6. Comandi per i vincoli e la gestione della struttura

- PRIMARY KEY - Definisce una colonna o più colonne come chiave primaria (valori univoci).
- FOREIGN KEY - Definisce una chiave esterna per creare relazioni tra tabelle.
- UNIQUE - Impone che i valori di una colonna siano univoci.

7. Comandi per query avanzate

- JOIN - Combina righe da più tabelle basate su una relazione.
- GROUP BY - Raggruppa i risultati per una o più colonne.
- HAVING - Filtra i gruppi creati da GROUP BY.

- ORDER BY - Ordina i risultati in base a una o più colonne.
- LIMIT - Limita il numero di righe restituite da una query.

8. Altri comandi utili

- DISTINCT - Elimina i duplicati dai risultati.
- CASE - Crea condizioni all'interno di una query.

Esempi di utilizzo reale per SELECT

1. Esempio 1: Seleziona tutti gli utenti:
`SELECT * FROM utenti;`
2. Esempio 2: Seleziona utenti maggiorenni:
`SELECT nome, età FROM utenti WHERE età >= 18;`

Esempi di utilizzo reale per INSERT INTO

3. Esempio 1: Aggiungi un nuovo utente:
`INSERT INTO utenti (nome, età) VALUES ('Mario', 25);`
4. Esempio 2: Aggiungi un utente con email:
`INSERT INTO utenti (nome, età, email) VALUES ('Anna', 22, 'anna@example.com');`

Esempi di utilizzo reale per UPDATE

5. Esempio 1: Aggiorna l'età di un utente:
`UPDATE utenti SET età = 30 WHERE nome = 'Mario';`
6. Esempio 2: Modifica più record:
`UPDATE utenti SET età = età + 1 WHERE età < 18;`

Esempi di utilizzo reale per DELETE

7. Esempio 1: Elimina un utente specifico:
`DELETE FROM utenti WHERE nome = 'Mario';`
8. Esempio 2: Elimina tutti gli utenti minorenni:
`DELETE FROM utenti WHERE età < 18;`

Esempi di utilizzo reale per CREATE TABLE

9. Esempio: Creazione di una tabella 'utenti':
`CREATE TABLE utenti (
 id INT AUTO_INCREMENT PRIMARY KEY,
 nome VARCHAR(100),
 età INT
);`

Esempi di utilizzo reale per ALTER TABLE

10. Esempio 1: Aggiungi una colonna email:
`ALTER TABLE utenti ADD email VARCHAR(255);`
11. Esempio 2: Rimuovi una colonna:
`ALTER TABLE utenti DROP COLUMN email;`

Esempi di utilizzo reale per DROP TABLE

12. Esempio: Elimina la tabella 'utenti':
DROP TABLE utenti;

Esempi di utilizzo reale per JOIN

13. Esempio: Combina utenti e ordini:
SELECT utenti.nome, ordini.id
FROM utenti
INNER JOIN ordini ON utenti.id = ordini.id_utente;

Esempi di utilizzo reale per GROUP BY

14. Esempio: Conta utenti per età:
SELECT COUNT(*), età
FROM utenti
GROUP BY età;

Esempi di utilizzo reale per ORDER BY

15. Esempio: Ordina utenti per età decrescente:
SELECT * FROM utenti ORDER BY età DESC;

Esempi di utilizzo reale per LIMIT

16. Esempio: Mostra solo i primi 5 utenti:
SELECT * FROM utenti LIMIT 5;

Esempi di utilizzo reale per DISTINCT

17. Esempio: Mostra età uniche degli utenti:
SELECT DISTINCT età FROM utenti;

Esempi di utilizzo reale per CASE

18. Esempio: Assegna categorie di età:
SELECT nome,
CASE
WHEN età < 18 THEN 'Minorenne'
ELSE 'Maggiorenne'
END AS categoria
FROM utenti;