SQL Common Commands List

Mauro Bogliaccino

Contents

- SQL Commands List for the Most Common Actions
  - Contents
    - SQL Commands List
      - CREATE VIEW
      - DELETE
      - GRANT
      - REVOKE
      - COMMIT
      - ROLLBACK
      - SAVEPOINT
      - DROP DATABASE

- SQL Commands List for the Most Common Actions
  - Contents
    - SQL Commands List
      - DROP INDEX
        - SQL Server
        - MS Access
        - DB2/Oracle
        - MySQL
      - DROP TABLE
      - EXISTS
      - GROUP BY
      - HAVING

- SQL Commands List for the Most Common Actions
  - Contents
    - SQL Commands List
      - IN
      - INSERT INTO
      - INNER JOIN
      - LEFT JOIN
      - RIGHT JOIN
      - FULL JOIN
      - LIKE
      - ORDER BY

- SQL Commands List for the Most Common Actions
  - Contents
    - SQL Commands List
      - SELECT
      - SELECT
      - SELECT DISTINCT
      - SELECT INTO
      - SELECT TOP
      - TRUNCATE TABLE
      - UNION
      - UNION ALL
      - UPDATE
      - WHERE

SQL Commands List

AND|OR

AND combina due o più condizioni in una singola query. Tutte le condizioni devono essere soddisfatte per mostrare i risultati.
OR restituisce risultati che soddisfano almeno una delle condizioni.

```
SELECT * FROM Developers

WHERE Country='France' AND City='Paris';
```

```
SELECT * FROM Developers

WHERE City='London' OR City='Paris';
```

ALTER TABLE

Permette di aggiungere o rimuovere colonne da una tabella.

```
ALTER TABLE Developers
ADD BirthDate date;
```

```
ALTER TABLE Developers

DROP COLUMN BirthDate;
```

AS (alias)

Permette di rinominare una colonna o una tabella con un alias più conveniente senza modificare i nomi originali nel database. Questo rende più semplici le query.

```
SELECT ID as CustomerID, Name AS Customers
FROM Customers;
```

```
SELECT o.ID, c.Name
FROM Customers AS c, Customer_orders AS o

WHERE c.id = 2 AND c.ID = o.customer_id;
```

BETWEEN

Filtra i risultati restituendo solo quelli che rientrano in un intervallo specificato (es. date, numeri o testo).

```
SELECT * FROM Orders

WHERE Price BETWEEN 10 AND 15;
```

CREATE DATABASE

Crea un nuovo database. È necessario avere diritti di amministratore per eseguire questa operazione.

```
CREATE DATABASE testingDB;
```

CREATE TABLE

Crea una nuova tabella all'interno di un database.

```
CREATE TABLE Suppliers (
    SupplierID int,
    FirstName varchar(255),
    LastName varchar(255),
    City varchar(255),
    Country varchar(255)

);
```

CREATE INDEX

Crea un indice per una tabella, rendendo più veloce il recupero dei dati. Gli indici non sono visibili agli utenti.

```
CREATE INDEX idx_lastname

ON Persons (LastName);
```

CREATE VIEW

Crea una vista basata su una query specifica. Una vista è simile a una tabella ma contiene solo i campi rilevanti per uno scopo specifico.

```
CREATE VIEW [Present List Products] AS
SELECT ID, Name
FROM Products

WHERE Discontinued = No;
```

DELETE

Elimina righe specifiche da una tabella.

```
DELETE FROM Developers

WHERE Name='Brad Pitt';
```

```
DELETE * FROM Developers;
```

GRANT

Concede permessi agli utenti per accedere a un database.

```
GRANT SELECT, UPDATE ON YOUR_TABLE TO FIRST_USER, SECOND_USER;
```

REVOKE

Revoca i permessi precedentemente concessi agli utenti.

```
REVOKE SELECT, UPDATE ON YOUR_TABLE FROM FIRST_USER, SECOND_USER;
```

COMMIT

Salva ogni transazione nel database.

```
DELETE FROM CUSTOMERS
WHERE AGE = 18;
```

COMMIT

ROLLBACK

Annulla le transazioni non ancora salvate nel database.

```
DELETE FROM CUSTOMERS
WHERE AGE = 18;

ROLLBACK;
```

SAVEPOINT

Crea un punto di riferimento all'interno di una transazione per poter tornare a quel punto senza annullare l'intera transazione.

```
SAVEPOINT SAVEPOINT_NAME;
```

DROP DATABASE

Elimina un intero database e tutti i suoi dati. Questa operazione deve essere usata con estrema cautela.

```
DROP DATABASE db_name
```

DROP INDEX

Elimina un indice specifico

SQL Server

```
DROP INDEX tbl_name.index_name
```

MS Access

```
DROP INDEX index_name ON tbl_name
```

DB2/Oracle

```
DROP INDEX index_name
```

MySQL

```
ALTER TABLE tbl_name DROP INDEX index_name
```

DROP TABLE

Elimina una tabella con tutti i suoi parametri. Se si vogliono eliminare solo i contenuti mantenendo la tabella, si usa il comando **TRUNCATE TABLE**.

```
DROP TABLE tbl_name
```

EXISTS

Verifica l'esistenza di un record utilizzando una sottoquery.

```
SELECT id, name
FROM customers

WHERE EXISTS (SELECT id FROM customer_orders WHERE customer_orders.customer_id = customers.id AND customers.city = "Rome");
```

GROUP BY

Organizza dati identici in gruppi utilizzando una funzione di aggregazione.

```
SELECT COUNT(ID), City
FROM Developers

GROUP BY City;
```

HAVING

Filtra i risultati di una query con funzioni di aggregazione. Si usa al posto di **WHERE** per queste funzioni.

```sql
SELECT COUNT(ID), Country
FROM Pets
GROUP BY Country

HAVING COUNT(ID) > 2;
```

IN

Include più valori nella clausola **WHERE**.

```
SELECT * FROM Developers

WHERE Country IN ('USA', 'France', 'India');
```

INSERT INTO

Inserisce nuove righe in una tabella.

```
INSERT INTO Developers (Name, City, Country)

VALUES ('Luke Christon', 'London', 'UK');
```

INNER JOIN

Combina righe provenienti da tabelle diverse.

```sql
SELECT Orders.ID, Developers.Name
FROM Orders

INNER JOIN Developers ON Orders.ID = Developers.ID;
```

LEFT JOIN

Recupera righe dalla tabella di sinistra con corrispondenze nella tabella di destra. Le righe senza corrispondenze contengono valori nulli.

```
SELECT Developers.Name, Customer_orders.ID
FROM Developers
LEFT JOIN Customer_orders ON Developers.ID = Customer_orders.customer_id

ORDER BY Developers.Name;
```

RIGHT JOIN

Recupera righe dalla tabella di destra con corrispondenze nella tabella di sinistra.

```
SELECT Customer_orders.ID, Employees.Last_name, Employees.First_name
FROM Customer_orders
RIGHT JOIN Employees ON Customer_orders.employee_id = Employees.ID

ORDER BY Customer_orders.ID;
```

FULL JOIN

Restituisce tutte le righe che corrispondono, sia nella tabella di sinistra che in quella di destra.

```
SELECT Customers.Name, Customer_orders.ID
FROM Customers
FULL OUTER JOIN Orders ON Customers.ID=Customer_orders.customer_id

ORDER BY Customers.Name;
```

LIKE

Cerca specifici pattern in una colonna.

```
SELECT * FROM users WHERE email LIKE '%gmail%';
```

ORDER BY

Ordina i risultati della query (ascendente per impostazione predefinita).

```
SELECT * FROM users ORDER BY email DESC;
```

SELECT

Seleziona dati da un database e restituisce una tabella di risultati.

```sql
SELECT username,email

FROM users;
```

```sql
SELECT * FROM Customers;
```

SELECT DISTINCT

Restituisce solo dati unici, escludendo i duplicati.

```
SELECT DISTINCT City FROM Developers;
```

SELECT INTO

Copia dati selezionati da una tabella a un’altra.

```
SELECT * INTO CustomerBackup2018

FROM Customers;
```

```
SELECT Name, Contact INTO CustomerBackup2017

FROM Customers;
```

SELECT TOP

Specifica il numero massimo o la percentuale di dati da restituire nei risultati.

```sql
SELECT * FROM Customers

LIMIT 3;
```

```sql
SELECT TOP 50 PERCENT * FROM Customers;
```

TRUNCATE TABLE

Elimina i dati di una tabella mantenendo la struttura e i parametri.

```
TRUNCATE TABLE tbl_name
```

UNION

Combina più set di risultati di query eliminando i duplicati.

```
SELECT City FROM Developers
UNION
SELECT City FROM Customers

ORDER BY City;
```

UNION ALL

Combina più set di risultati mantenendo i duplicati.

```
SELECT City FROM Developers
UNION ALL
SELECT City FROM Customers

ORDER BY City;
```

UPDATE

Aggiorna i dati di una tabella.

```
UPDATE Developers
SET City = 'Paris', Country= 'France'

WHERE Name = 'Brad Pitt';
```

WHERE

Filtra i risultati di una query restituendo solo quelli che soddisfano la condizione specificata. Non può essere usata con funzioni di aggregazione; in questo caso, si usa **HAVING**.

```
SELECT * FROM Developers

WHERE Country='France';
```