

SQL Cheat Sheet

Concetti di Base

```
USE sql_store;  
SELECT * FROM customers WHERE state = 'CA' ORDER BY first_name LIMIT 3;
```

- **SQL non è sensibile alle maiuscole/minuscole.**
Puoi usare maiuscole o minuscole indifferentemente.
- **In MySQL, ogni istruzione deve terminare con un punto e virgola (;).**

Commenti

I commenti servono per aggiungere note al codice senza influenzarne l'esecuzione.

```
-- Questo è un commento e non verrà eseguito.
```

Clausola **SELECT**

Uso delle espressioni

```
SELECT (points * 10 + 20) AS discount_factor FROM customers;
```

Ordine di esecuzione delle operazioni matematiche:

1. Parentesi
2. Moltiplicazione/Divisione
3. Addizione/Sottrazione

Rimuovere duplicati

```
SELECT DISTINCT state FROM customers;
```

Usa **DISTINCT** per ottenere solo valori unici.

Clausola WHERE

Usiamo **WHERE** per filtrare i dati.

Operatori di confronto:

- Maggiore di: **>**
- Maggiore o uguale a: **>=**
- Minore di: **<**
- Minore o uguale a: **<=**
- Uguale a: **=**
- Diverso da: **<>** o **!=**

Operatori Logici

AND (entrambe le condizioni devono essere vere)

```
SELECT *  
FROM customers  
WHERE birthdate > '1990-01-01' AND points > 1000;
```

OR (almeno una delle condizioni deve essere vera)

```
SELECT *  
FROM customers  
WHERE birthdate > '1990-01-01' OR points > 1000;
```

NOT (nega una condizione)

```
SELECT *  
FROM customers  
WHERE NOT (birthdate > '1990-01-01');
```

Operatore IN

Trova valori in una lista specifica.

```
SELECT *  
FROM customers  
WHERE state IN ('VA', 'NY', 'CA');
```

Operatore BETWEEN

Trova valori in un intervallo.

```
SELECT *  
FROM customers  
WHERE points BETWEEN 100 AND 200;
```

Operatore LIKE

Trova valori che corrispondono a un pattern.

```
SELECT *  
FROM customers  
WHERE first_name LIKE 'b%';
```

- % = qualsiasi numero di caratteri.
- _ = esattamente un carattere.

Operatore REGEXP

Fondamenti di basi di dati

Trova valori usando espressioni regolari.

```
SELECT *  
FROM customers  
WHERE first_name REGEXP '^a';
```

Caratteri utili:

- `^` : inizio della stringa.
- `$` : fine della stringa.
- `|` : OR logico.
- `[abc]` : qualsiasi carattere tra le parentesi.
- `[a-d]` : qualsiasi carattere nel range specificato.

Esempi REGEXP:

- Nomi che terminano con "EY" o "ON":

```
SELECT * FROM customers WHERE first_name REGEXP 'ey$|on$';
```

- Nomi che iniziano con "MY" o contengono "SE":

```
SELECT * FROM customers WHERE first_name REGEXP '^my|se';
```


Operatore IS NULL

Trova righe con valori null.

```
SELECT *  
FROM customers  
WHERE phone IS NULL;
```

Clausola ORDER BY

Ordina i risultati.

```
SELECT *  
FROM customers  
ORDER BY state, first_name DESC;
```

- **ASC** (ordine crescente) è predefinito.
- **DESC** (ordine decrescente) deve essere specificato.

Clausola LIMIT

Limita il numero di righe restituite.

```
SELECT * FROM customers LIMIT 3;
```

Per saltare righe e limitare il risultato:

```
SELECT * FROM customers LIMIT 6, 3;
```

JOIN

Inner Join

Restituisce solo le righe con corrispondenze in entrambe le tabelle.

```
SELECT *  
FROM customers c JOIN orders o  
ON c.customer_id = o.customer_id;
```

Outer Join

Restituisce tutte le righe da una tabella, anche se non ci sono corrispondenze.

```
SELECT *  
FROM customers c LEFT JOIN orders o  
ON c.customer_id = o.customer_id;
```

Clausola USING

Semplifica il join quando i nomi delle colonne sono identici.

```
SELECT *  
FROM customers c JOIN orders o USING (customer_id);
```

Cross Join

Combina ogni riga di una tabella con ogni riga di un'altra tabella.

```
SELECT *  
FROM colors CROSS JOIN sizes;
```


UNION

Unisce i risultati di più query.

```
SELECT name, address  
FROM customers  
UNION  
SELECT name, address  
FROM clients;
```

Inserimento di Dati

Inserire un singolo record

```
INSERT INTO customers(first_name, phone, points)  
VALUES ('Brad Pitt', NULL, DEFAULT);
```

Inserire più record

```
INSERT INTO customers(first_name, phone, points) VALUES  
('Brad Pitt', NULL, DEFAULT),  
('Bob', '1234', 10);
```