

GarraTT's MySQL Cheatsheet

A practical MySQL reference for building and querying relational data. Focused on the queries and patterns you'll use in real apps: selects, joins, indexes, and safety.

SELECT Essentials

Goal	Pattern	Example
Select columns	SELECT ... FROM ...	SELECT id, title, created_at FROM posts;
Filter rows	WHERE + operators	SELECT * FROM posts WHERE published = 1 AND created_at >= '2026-01-01';
Sort + limit	ORDER BY + LIMIT	SELECT * FROM posts ORDER BY created_at DESC LIMIT 10;
Distinct	Remove duplicates	SELECT DISTINCT author_id FROM posts;
Like search	Prefix/infix match	SELECT * FROM posts WHERE title LIKE '%mysql%';

Aggregations

Function	Use	Example
COUNT	How many rows	SELECT COUNT(*) AS n FROM posts WHERE published = 1;
SUM / AVG	Totals + averages	SELECT AVG(read_time_min) AS avg_min FROM posts;
GROUP BY	Group aggregates	SELECT author_id, COUNT(*) AS posts FROM posts GROUP BY author_id;
HAVING	Filter grouped results	SELECT author_id, COUNT(*) AS posts FROM posts GROUP BY author_id HAVING posts >= 5;

Joins

Join	Use	Example
INNER JOIN	Only matching rows	SELECT p.title, u.name FROM posts p JOIN users u ON u.id = p.author_id;
LEFT JOIN	Keep left side even if missing	SELECT u.name, p.title FROM users u LEFT JOIN posts p ON p.author_id = u.id;
Many to many	Join through pivot table	SELECT p.title, t.name FROM posts p JOIN post_tags pt ON pt.post_id = p.id JOIN tags t ON t.id = pt.tag_id;

Tables and Columns

Task	Command	Example
Create table	CREATE TABLE	<pre>CREATE TABLE posts (id BIGINT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(200) NOT NULL, body TEXT, published TINYINT(1) NOT NULL DEFAULT 0, created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP);</pre>
Alter table	ALTER TABLE	<pre>ALTER TABLE posts ADD COLUMN slug VARCHAR(200) NOT NULL;</pre>
Change column	MODIFY / CHANGE	<pre>ALTER TABLE posts MODIFY COLUMN title VARCHAR(255) NOT NULL;</pre>

Constraints

Constraint	Use	Example
PRIMARY KEY	Unique row id	<pre>id BIGINT PRIMARY KEY AUTO_INCREMENT</pre>
UNIQUE	No duplicates	<pre>UNIQUE KEY uq_posts_slug (slug)</pre>
FOREIGN KEY	Relational integrity	<pre>author_id BIGINT, CONSTRAINT fk_posts_author FOREIGN KEY (author_id) REFERENCES users(id)</pre>
NOT NULL	Required fields	<pre>title VARCHAR(200) NOT NULL</pre>

Indexes and Performance

Tool	Use	Example
Add index	Speed up WHERE / JOIN	<pre>CREATE INDEX idx_posts_created ON posts(created_at);</pre>
Composite index	Match multi-column filters	<pre>CREATE INDEX idx_posts_pub_date ON posts(published, created_at);</pre>
EXPLAIN	See query plan	<pre>EXPLAIN SELECT * FROM posts WHERE published = 1 ORDER BY created_at DESC LIMIT 10;</pre>
Covering index	Avoid table lookups	<pre>CREATE INDEX idx_posts_pub_date_title ON posts(published, created_at, title);</pre>

Writes

Action	Pattern	Example
INSERT	Add a row	INSERT INTO posts (title, body, published) VALUES ('Hello', 'Content...', 1);
UPDATE	Change rows	UPDATE posts SET published = 1 WHERE id = 42;
DELETE	Remove rows	DELETE FROM posts WHERE id = 42;
Upsert	Insert or update	INSERT INTO tags (name) VALUES ('mysql') ON DUPLICATE KEY UPDATE name = VALUES(name);

Transactions

Command	Use	Example
START TRANSACTION	Begin atomic work	START TRANSACTION;
COMMIT	Save changes	COMMIT;
ROLLBACK	Undo on error	ROLLBACK;
Isolation	Tune consistency	SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

Date and Time

Need	Function	Example
Now	NOW()	SELECT NOW();
Date part	DATE()	SELECT DATE(created_at) AS day FROM posts;
Range filter	BETWEEN / >= <	SELECT * FROM posts WHERE created_at >= '2026-01-01' AND created_at < '2026-02-01';
Intervals	DATE_ADD	SELECT DATE_ADD(NOW(), INTERVAL 7 DAY);

JSON

Tool	Use	Example
JSON type	Store structured data	ALTER TABLE posts ADD COLUMN meta JSON;
Extract	Read JSON fields	SELECT meta->>'\$.readingTime' AS rt FROM posts;
Set	Update a JSON path	UPDATE posts SET meta = JSON_SET(meta, '\$.featured', true) WHERE id = 42;

Handy Admin

Task	Command	Example
Show databases	SHOW DATABASES	SHOW DATABASES;
Use database	USE	USE mydb;
Show tables	SHOW TABLES	SHOW TABLES;
Describe table	DESCRIBE	DESCRIBE posts;