MySQL Commands, Data Types, and Constraints Cheat Sheet

Command/Concept	Syntax	Description
Connecting to MySQL	mysql -u [username] -p	Connects to the MySQL server with a specified username. The `-p` flag prompts for the user's password.
Show Databases	SHOW DATABASES;	Lists all the databases available on the MySQL server.
Select Database	USE [database_name];	Sets the specified database as the current database for subsequent queries.
Show Tables	SHOW TABLES;	Lists all the tables in the currently selected database.
Describe Table Structure	DESCRIBE [table_name];	Provides a detailed description of the table's columns, their data types, and other attributes.
Create Database	CREATE DATABASE [database_name];	Creates a new database with the specified name.
Drop Database	DROP DATABASE [database_name];	Deletes the specified database and all its contents.

Create Table	CREATE TABLE [table_name] (
	    column1 datatype constraints,
	    column2 datatype constraints,
	    …
	);
		Creates a new table with the specified columns, data types, and constraints.

Drop Table	DROP TABLE [table_name];	Deletes the specified table and its data.
Add Column	ALTER TABLE [table_name] ADD [column_name] datatype constraints;	Adds a new column to the specified table.
Modify Column	ALTER TABLE [table_name] MODIFY [column_name] new_datatype new_constraints;	Modifies the data type or constraints of an existing column.
Drop Column	ALTER TABLE [table_name] DROP COLUMN [column_name];	Deletes a column from the specified table.

Insert Data	INSERT INTO [table_name] (column1, column2, ...) VALUES (value1, value2, ...);	Inserts new data into the specified columns of a table.
Update Data	UPDATE [table_name] SET column1 = value1, column2 = value2, ... WHERE condition;	Updates existing data in a table based on a condition.
Delete Data	DELETE FROM [table_name] WHERE condition;	Deletes data from a table based on a condition.
Select Data	SELECT column1, column2, ... FROM [table_name] WHERE condition;	Retrieves data from specified columns in a table based on a condition.
WHERE Clause	SELECT column1, column2, ... FROM [table_name] WHERE condition;	Filters records that meet the specified condition.
ORDER BY Clause	SELECT column1, column2, ... FROM [table_name] ORDER BY column1 [ASC|DESC];	Sorts the result set by specified columns in ascending (ASC) or descending (DESC) order.
GROUP BY Clause	SELECT column1, COUNT(column2) FROM [table_name] GROUP BY column1;	Groups rows that have the same values in specified columns into summary rows.
HAVING Clause	SELECT column1, COUNT(column2) FROM [table_name] GROUP BY column1 HAVING condition;	Filters groups based on a condition, used with GROUP BY.
INNER JOIN	SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;	Retrieves records that have matching values in both tables.
LEFT JOIN	SELECT columns FROM table1 LEFT JOIN table2 ON table1.column = table2.column;	Retrieves all records from the left table, and the matched records from the right table. Returns NULL for non-matching rows from the right table.
RIGHT JOIN	SELECT columns FROM table1 RIGHT JOIN table2 ON table1.column = table2.column;	Retrieves all records from the right table, and the matched records from the left table. Returns NULL for non-matching rows from the left table.
FULL JOIN	SELECT columns FROM table1 LEFT JOIN table2 ON table1.column = table2.column UNION SELECT columns FROM	Retrieves records when there is a match in one of the tables. MySQL does not support FULL JOIN natively, so a combination of LEFT JOIN and RIGHT JOIN is

	table1 RIGHT JOIN table2 ON table1.column = table2.column;	used with UNION.
COUNT()	SELECT COUNT(column_name) FROM [table_name];	Returns the number of rows that match a specified condition.
SUM()	SELECT SUM(column_name) FROM [table_name];	Returns the total sum of a numeric column.
AVG()	SELECT AVG(column_name) FROM [table_name];	Returns the average value of a numeric column.
MAX()	SELECT MAX(column_name) FROM [table_name];	Returns the maximum value in a column.
MIN()	SELECT MIN(column_name) FROM [table_name];	Returns the minimum value in a column.
CONCAT()	SELECT CONCAT(column1, ' ', column2) FROM [table_name];	Concatenates two or more strings.
SUBSTRING()	SELECT SUBSTRING(column_name, start_position, length) FROM [table_name];	Extracts a substring from a string (starting at any position).
LENGTH()	SELECT LENGTH(column_name) FROM [table_name];	Returns the length of a string.
REPLACE()	SELECT REPLACE(column_name, 'search_string', 'replace_string') FROM [table_name];	Replaces all occurrences of a substring within a string with a new substring.
CURDATE()	SELECT CURDATE();	Returns the current date.
NOW()	SELECT NOW();	Returns the current date and time.
DATE_ADD()	SELECT DATE_ADD(column_name, INTERVAL value unit) FROM [table_name];	Adds a specified time interval to a date.
DATE_SUB()	SELECT DATE_SUB(column_name, INTERVAL value unit) FROM [table_name];	Subtracts a specified time interval from a date.
Create Index	CREATE INDEX index_name ON [table_name](column_name);	Creates an index on a table for faster query performance.
Drop Index	DROP INDEX index_name ON [table_name];	Deletes an index from a table.
Start Transaction	START TRANSACTION;	Begins a new transaction.
Commit	COMMIT;	Commits the current transaction, making all changes permanent.

Rollback	ROLLBACK;	Rolls back the current transaction, undoing all changes made since the transaction began.
Create User	CREATE USER 'username'@'host' IDENTIFIED BY 'password';	Creates a new MySQL user with the specified username and password.
Grant Privileges	GRANT ALL PRIVILEGES ON [database_name].* TO 'username'@'host';	Grants all privileges on the specified database to the user.
Revoke Privileges	REVOKE ALL PRIVILEGES ON [database_name].* FROM 'username'@'host';	Revokes all privileges on the specified database from the user.
Drop User	DROP USER 'username'@'host';	Deletes a MySQL user.
Backup Database	mysqldump -u [username] -p [database_name] > backup.sql	Creates a backup of the specified database and saves it to a file.
Restore Database	mysql -u [username] -p [database_name] < backup.sql	Restores a database from a backup file.
PRIMARY KEY	PRIMARY KEY (column_name)	Uniquely identifies each row in a table. Ensures no NULL values.
FOREIGN KEY	FOREIGN KEY (column_name) REFERENCES other_table(column_name)	Creates a relationship between two tables, ensuring referential integrity.
UNIQUE	UNIQUE (column_name)	Ensures all values in a column are unique.
NOT NULL	column_name datatype NOT NULL	Ensures a column cannot have NULL values.
DEFAULT	column_name datatype DEFAULT default_value	Sets a default value for a column if no value is provided.
CHECK	CHECK (condition)	Ensures all values in a column satisfy a specific condition.
AUTO_INCREMENT	column_name datatype AUTO_INCREMENT	Automatically generates a unique number for new rows.