MySQL
Cheat Sheet

Start with MySql

```
mysql -u [username] -p;  // Connect to MySQL server

// Connect to MySQL Server with a database using a username and password
mysql -u [username] -p [database];

exit; // Exit mysql command-line client:

// Export data using mysqldump tool
mysqldump -u [username] -p [database] > data_backup.sql;

mysql> system clear; // To clear MySQL screen console window on Linux
```

Working with databases

```
// Create a database with a specified name if it does not exist
CREATE DATABASE [IF NOT EXISTS] database_name;

// Use a database or change the current database to another databas
USE database_name;

// Drop a database with a specified name permanently.
DROP DATABASE [IF EXISTS] database_name;

// Show all available databases in the current database server
SHOW DATABASE;
```

Working with tables

```sql
// Show all tables in a current database.
SHOW TABLES;

//Create a new table
CREATE TABLE [IF NOT EXISTS] table_name(
  field1 type1, field2 type2
);

// Add a new column into a table:
ALTER TABLE table ADD [COLUMN] column_name;

// Drop a column from a table:
ALTER TABLE table_name DROP [COLUMN] column_name;

// Add index with a specific name to a table on a column:
ALTER TABLE table ADD INDEX [name](column, ...);

// Add primary key into a table:
ALTER TABLE table_name ADD PRIMARY KEY (column_name,...);

// Remove the primary key of a table:
ALTER TABLE table_name DROP PRIMARY KEY;

// Drop a table:
DROP TABLE [IF EXISTS] table_name;

// Show the columns of a table:
DESCRIBE table_name;

//Show the information of a column in a table:
DESCRIBE table_name column_name;
```

Querying data from tables

```sql
SELECT * FROM table_name; // Query all data from a table:

// Query data from one or more column of a table:
SELECT column1, column2, ... FROM table_name;

// Remove duplicate rows from the result of a query:
SELECT DISTINCT (column) FROM table_name;

// Query data with a filter using a WHERE clause:
SELECT select_list FROM table_name WHERE condition;

// Query data from multiple tables using inner join:
SELECT select_list FROM table1 INNER JOIN table2 ON condition;

// Query data from multiple tables using left join:
SELECT select_list FROM table1 LEFT JOIN table2 ON condition;

// Query data from multiple tables using right join:
SELECT select_list FROM table1 RIGHT JOIN table2 ON condition;

// Make a Cartesian product of rows:
SELECT select_list FROM table1 CROSS JOIN table2;

SELECT COUNT(*) FROM table_name;  // Counting rows in a table.

// Group rows using the GROUP BY clause.
SELECT select_list FROM table_name
GROUP BY column_1, column_2, ...;

// Filter group using the HAVING clause:
SELECT select_list FROM table_name
GROUP BY column1 HAVING condition;
```

Working with indexes

```
// Creating an index with the specified name on a table:
CREATE INDEX index_name ON table_name (column,...);

DROP INDEX index_name; // Drop an index:

CREATE UNIQUE INDEX index_name // Create a unique index:
ON table_name (column,...);
```

Working with views

```
CREATE VIEW [IF NOT EXISTS] view_name  // Create a new view:
AS select_statement;

// Create a new view with the WITH CHECK OPTION:
CREATE VIEW [IF NOT EXISTS] view_name
AS select_statement WITH CHECK OPTION;

// Create or replace a view:
CREATE OR REPLACE view_name AS select_statement;

DROP VIEW [IF EXISTS] view_name; // Drop a view:

DROP VIEW [IF EXISTS] view1, view2, ...; // Drop multiple views:

RENAME TABLE view_name TO new_view_name; // Rename a view:

SHOW FULL TABLES  // Show views from a database:
[{FROM | IN } database_name] WHERE table_type = 'VIEW';
```

Modifying data in tables

```sql
// Insert a new row into a table:
INSERT INTO table_name(column_list) VALUES(value_list);

// Insert multiple rows into a table:
INSERT INTO table_name(column_list)
VALUES(value_list1), (value_list2), (value_list3), ...;

// Update all rows in a table:
UPDATE table_name SET column1 = value1, ...;

// Update data for a set of rows specified
// by a condition in WHERE clause.
UPDATE table_name
SET column_1 = value_1, ...
WHERE condition

// Update with join
UPDATE table1, table2
INNER JOIN table1 ON table1.column1 = table2.column2
SET column1 = value1, WHERE condition;

DELETE FROM table_name;  // Delete all rows in a table

// Delete rows specified by a condition:
DELETE FROM table_name WHERE condition;

// Delete with join
DELETE table1, table2 FROM table1 INNER JOIN table2
ON table1.column1 = table2.column2
WHERE condition;
```

Searching

```
// Search for data using the LIKE operator:
SELECT select_list
FROM table_name
WHERE column LIKE '%pattern%';

// Text search using a regular expression
// with RLIKE operator.
SELECT select_list
FROM table_name
WHERE column RLIKE 'regular_expression';
```

- Link In Bio To Get Cheat Sheet PDF.

Follow us for a daily dose of tech insights

Drop us a line and share your suggestions for next posts.

If you have any queries then let me know in the comment box.

Was it helpful?

follow for more!

Like

Comment

Share

Save

Muhammad Zubair

@iamzubairarain

Did you know? I'm a freelancer too! 💼 Visit my Upwork profile: [https://www.upwork.com/freelancers/~014ffd655fc0791835]