MySQL Quick Reference

NOTE: [table] indicates a place you put the name of a table. For example, describe [table]; is the <u>form</u> of the command. To <u>actually use</u> the command on a table named CUSTOMER, type describe CUSTOMER; Similarly, where you see [column], put the actual name of a column. Where you see [value], put an actual value.

Show all tables: show tables;
Show table structure: describe [table];
List all indexes on a table: show index from [table];
Create new table with columns: CREATE TABLE [table] ([column] VARCHAR(120), [another-column] DATETIME);
Adding a column: ALTER TABLE [table] ADD COLUMN [column] VARCHAR(120);
Adding a column with a unique, auto-incrementing ID: ALTER TABLE [table] ADD COLUMN [column] INT NOT NULL AUTO_INCREMENT PRIMARY KEY;

Inserting a record: INSERT INTO [table] ([column], [column]) VALUES ('[value]', [value]');

Selecting records: SELECT * FROM [table];
Selecting parts of records: SELECT [column], [another-column] FROM [table];
Selecting specific records: SELECT * FROM [table] WHERE [column] = [value];
        (Selectors: <, >, !=; combine multiple selectors with AND, OR)
Select records containing [value]: SELECT * FROM [table] WHERE [column] LIKE '%[value]%';
Select records starting with [value]: SELECT * FROM [table] WHERE [column] LIKE '[value]%';
Select records starting with val and ending with ue: SELECT * FROM [table] WHERE [column] LIKE '[val_ue]';
Select a range: SELECT * FROM [table] WHERE [column] BETWEEN [value1] and [value2];
Select with custom order and only limit: SELECT * FROM [table] WHERE [column] ORDER BY [column] ASC LIMIT [value];(Order: DESC, ASC)

Custom column output names: SELECT [column] AS [custom-column] FROM [table];

Updating records: UPDATE [table] SET [column] = '[updated-value]' WHERE [column] = [value];

Deleting records: DELETE FROM [table] WHERE [column] = [value];
Delete *all records* from a table (without dropping the table itself): DELETE FROM [table];

Removing table columns: ALTER TABLE [table] DROP COLUMN [column];
Deleting tables: DROP TABLE [table];
Deleting databases: DROP DATABASE [database];

MySQL function for date-time input: NOW()

Logout: exit;

Aggregate functions

Select but without duplicates: SELECT DISTINCT [column] FROM [table];
Calculate total number of records: SELECT COUNT([column]) FROM [table];
Counting and selecting grouped records: SELECT *, (SELECT COUNT([column]) FROM [table]) AS count FROM [table] GROUP BY [column];
Count total number of [column] and group by [category-column]: SELECT [category-column], SUM([column]) FROM [table] GROUP BY [category-column];
Get largest value in [column]: SELECT MAX([column]) FROM [table];
Get smallest value: SELECT MIN([column]) FROM [table];
Get average value: SELECT AVG([column]) FROM [table];
Get rounded average value and group by [category-column]: SELECT [category-column], ROUND(AVG([column]), 2) FROM [table] GROUP BY [category-column];

Multiple tables

Select from multiple tables: SELECT [table1].[column], [table1].[another-column], [table2].[column] FROM [table1], [table2];

Combine rows from 2 different tables: SELECT * FROM [table1], [table2] WHERE [table1].[column] = [table2].[column];

Combine rows from 2 different tables (another way): SELECT * FROM [table1] INNER JOIN [table2] ON [table1].[column] = [table2].[column];

Combine rows from 3 different tables: SELECT * FROM [table1], [table2], [table3] WHERE [table1].[column] = [table2].[column] AND [table2].[column] = [table3].[column];

Combine rows from 2 different tables but do not require the join condition: SELECT * FROM [table1] LEFT OUTER JOIN [table2] ON [table1].[column] = [table2].[column]; (The left table is the first table that appears in the statement. Use RIGHT OUTER JOIN to specify the second table.)

Rename column or table using an *alias*: SELECT [table1].[column] AS '[value]', [table2].[column] AS '[value]' FROM [table1], [table2];