TL;DR BASIC STATEMENTS

Objectives

- Use the `INSERT` statement to create data
- Use the `SELECT` statement to read data
- Use the `UPDATE` statement to update data
- Use the `DELETE` statement to delete data
- Introduce the `WHERE` clause
- Understand MySQL operators

Overview

- 2 sets of commands
    - Create/Manage the structure of your database
    - Manipulate data in your database
- CRUD Operations
    - **C**reate: `Insert`
    - **R**ead: `Selcect`
    - **U**pdate: `Update`
    - **D**elete: `Delete`
- Standard SQL Convention: Single Quote `(')`
    - Numeric values do not need to be quoted
    - Inner string quotes can be escaped `(\')` or you can use two single quotes in a row `('')`

Insert

```
INSERT INTO table_name (field1, field2, ...)
VALUES ('value1', 'value2', ...);
```

- The set of values must match up exactly with the set of columns
- Should almost never specify a column that has `AUTO_INCREMENT` like your primary key.

Select

```
SELECT column1, column2, ... FROM table_name;
```

- Very powerful statement to get what you are looking for

Where Clause

```
SELECT column1, column2, ...
FROM table_name
WHERE column_name = 'value';
```

- We can use `WHERE` to specify what data we want returned
- Notice - SQL uses a single `=`
- The fastest way to find a single record is to use the table's primary key

MySQL Operators

- `=` Eqaul
- `!=` or `<>` Not Equal
- `<` Less than
- `>` Greater than
- `<=` Less than or equal to
- `>=` Greater than or equal to
- `BETWEEN value1 AND value2` Greater than or equal to value1 and less than or equal to value2

Update

```
UPDATE table_name
SET column1 = 'value1', column2 = 'value2', ...
WHERE columnA = 'valueA';
```

- Only updates existing records

Delete

```
DELETE FROM table_name WHERE column_name = 'value';
```

- Table `id` s will not automatically reorder
    - This is intentional - as primary keys should be consistent and predictable
- !!! **CAUTION** !!!
    - Delete query is very very dangerous. **THERE IS NO GOING BACK!**
    - As a developer, you want to make sure you are deleting the records you want to!
    - Safest approach is to start with a `select` statement... then convert to a `delete` statement

```
-- First:
SELECT * FROM quotes WHERE id = 3;
-- Convert to:
DELETE FROM quotes WHERE id = 3;
```

Truncate

```
TRUNCATE table_name;
```

- Truncate **DELETES EVERYTHING!**

Additional Resources

- How to use Quotes in SQL
- MySQL Operators