SQL BOOTCAMP: CREATE/DELETE DATA & DATABASES

GENERAL ASSEMBLY

DATABASES

MySQL contains a number of databases, databases contain a number of *tables*. Tables are like pages in Excel spreadsheets, made from *rows* and *columns*.

FINDING DATABASES & TABLES

The following commands are used to navigate MySQL and find databases and tables inside of MySQL:

- `SHOW DATABASES;` — lists all databases
- `USE DATABASE` *`database`*`;` — makes the database *`database`* active.
- `SHOW TABLES;` — lists all tables in the current database.
- `DESCRIBE` *`table`*`;` — Describes the properties of the table named *`table`*.

DATA TYPES

Numbers: `INT` (whole numbers), `FLOAT` (numbers with a decimal)

Dates: `DATE` (e.g. 2015-12-12), `TIME` (e.g. 16:49:59), `DATETIME` (e.g. 2015-12-12 16:49:59)

Strings: `VARCHAR(len)`, stores up to `len` characters of text

THE NULL TYPE

The null type (represented as NULL, with no quotes) is a special datatype that means "we don't know what goes here.

CREATE/DELETE A DATABASE

Use **CREATE DATABASE** to create a new database

```
CREATE DATABASE [IF NOT EXISTS]
database;
```

You can optionally add **IF NOT EXISTS** to this command, which stops it from failing if the table already exists.

CREATE/DELETE A TABLE

Use **CREATE TABLE** to create a new table.

```
CREATE TABLE table name (
   col1 col_type constraints,
   col2 col_type constraints
);
```

For each column you need to specify the column name, the datatype of the column and optionally any constraints.

TABLE CONSTRAINTS

Table constraints specify rules on a table

- `NOT NULL` — makes it an error to have **NULL** values
- `UNIQUE` — makes it an error to have duplicate values
- `DEFAULT` *`value`* — sets values to the default if they are not specified on insert
- `PRIMARY KEY` — sets this column as the main identifier for the table
- `FOREIGN KEY col REFERENCES table(other_col)` — makes this column depend on a column in another table

INSERTING DATA

Use **INSERT INTO** to add any number of new rows to a table

```
INSERT INTO table
   (col1, col2, col3)
VALUES
   (val1, val2, val3),
   (val1, val2, val3);
```

You can miss out columns, they will be filled in with `NULL` unless the `NOT NULL` constraint is set, in which case the `NULL` value is considered an error.

UPDATING DATA

Use **UPDATE** to change values in a table

```
UPDATE
    table
SET
    col = value
[WHERE condition];
```

If you don't add a **WHERE** clause, the **UPDATE** will apply to *all* rows. Be careful!

DELETING DATA

Use **DELETE FROM** to delete rows from a table.

```
DELETE FROM
    table
[WHERE condition];
```

If you don't add a **WHERE** clause, the **DELETE FROM** will delete all rows in the table.