Intro to SQL

Overview

- SQL is a DDL (data definition language) and querying language
- We will be using it for database work in this class
- It can create defitions, add/remove data and query the database

Basic Commands

- CREATE TABLE – creates a new table
- INSERT INTO – inserts rows into a table
- DROP TABLE – deletes an entire table
- SELECT – returns rows from a table
- DELETE – deletes rows from a table
- UPDATE – changes contents of rows in a table

CREATE Command

```
CREATE TABLE `bsg_planets` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `population` bigint,
  `language` varchar(255),
  `capitol` varchar(255),
  PRIMARY KEY (`id`),
  UNIQUE KEY (`name`)
) ENGINE=InnoDB;
```

CREATE

- CREATE TABLE `bsg_planets`
  - Creates a table called bsg_planets
  - Attributes follow within the ()'s
- `id` int NOT NULL AUTO_INCREMENT
  - Creates an integer attribute called `id` that can not be null and will automatically increment on every new row

CREATE

- `name` varchar(255) NOT NULL,
  - Creates a variable length string attributed called `name` which may not be null
- `population` bigint,
  - Creates a big integer attribute called `population`
- `language` varchar(255),
  `capitol` varchar(255),
  - Creates variable length strings `language` and `capitol`

CREATE

- PRIMARY KEY (`id`),
  - Sets the attribute `id` to be the primary key
- UNIQUE KEY (`name`)
  - Sets `name` to be a unique field
- ENGINE=InnoDB
  - Sets the database engine to be InnoDB
  - Always do this for this class
  - Allows foreign keys

INSERT INTO

- INSERT INTO bsg_planets (name,population, language, capitol) values (“Gemenon”,2800000000, “Old Gemenese", “Oranu");
- INSERT INTO bsg_planets
  - Inserts rows into bsg_planets
- (name,population, language, capitol)
  - Specifies columns we will be inerting data into

INSERT INTO

- values (“Gemenon”,2800000000, “Old Gemenese", “Oranu")
  - Specifies values we are inserting
  - We could insert multiple rows like so:
    - Values (“a”,1,”b”,”c”),(“z”,2,”y”,”x”);
    - Each tuple is a row, so this would add two new rows

DROP TABLE

- DROP TABLE bsg_planets;
- Deletes the table and all rows
- Don't do this unless you are sure

SELECT

- SELECT name, population FROM bsg_planets WHERE population > 2700000000;
- SELECT name, population
  - Specifies the columns to select
  - Can use * to select all columns, don't do this other if an application will use values from returned rows
- FROM bsg_planets
  - Table to select from

SELECT

- WHERE population > 2700000000;
  - Condition to select rows on
  - Optional
  - Normal equality operators work
  - Use = for equality and <> for not equals
  - Strings need to be in quotes
  - More advanced selection techniques will come later

DELETE

- DELETE FROM bsg_planets WHERE name = “blah”;
- DELETE FROM bsg_planets
  - Chooses the table to delete rows from
- WHERE name = “blah”;
  - Condition used to pick rows to delete
  - If you are not sure you are deleting the right rows, use the same condition in a SELECT statement

UPDATE

- UPDATE bsg_planets SET name = ‘Caprica’ WHERE id=3;
- UPDATE bsg_planets
  - Picks table to update
- SET name = ‘Caprica’
  - The field to change in the selected rows
- WHERE id=3
  - Condition to find rows to update

Some attribute types

- INT - -2147483648 to 2147483647
- BIGINT – Bigger than int
- FLOAT – Floating point
- DECIMAL (M,D) – Contains M digits with D right of the decimal point
- VARCHAR(M) – Variable length string of max length M
- CHAR(M) – Fixed length string of length M