SQL

- In order to build increasingly complex websites, we depend on a **database** to store information long-term. The simplest form of a database with which we are all likely familiar is a basic spreadsheet, organized into rows and columns, tabs (tables), and individual files (databases).

- SQL is a programming language whose purpose is to *query* databases (perform operations on them).

- After you create a database, you create one or more **tables**.
- For each table, you specify all of the **columns** in the table.
- When new information is added to the database, the new information (typically) goes into a new **row**.
- There are many data types that can be stored in a SQL database. This is just a small sample.

INT	SMALLINT	TINYINT	MEDIUMINT	BIGINT
DECIMAL	FLOAT	BIT	DATE	TIME
DATETIME	TIMESTAMP	CHAR	VARCHAR	BINARY
BLOB	TEXT	ENUM	GEOMETRY	LINESTRING

- After you create a database, you create one or more **tables**.
- For each table, you specify all of the **columns** in the table.
- When new information is added to the database, the new information (typically) goes into a new **row**.
- In SQLite, which we'll use in this course, we can consolidate these various datatypes into a few more general classes (though underlying types still exist)

`NULL`	`INTEGER`	`REAL`	`TEXT`	`BLOB`

- Another consideration is choosing a column to be a **primary key**, guaranteed to be unique across rows. A good primary key makes subsequent table operations much easier.
    - You can also have a *joint primary key*, a combination of two or more columns where the combination is guaranteed to be unique.

- SQL is a programming language, but it has a limited vocabulary that we’ll use.

- Another consideration is choosing a column to be a **primary key**, guaranteed to be unique across rows. A good primary key makes subsequent table operations much easier.
  - You can also have a *joint primary key*, a combination of two or more columns where the combination is guaranteed to be unique.

- SQL is a programming language, but it has a limited vocabulary that we'll use.

```
INSERT
SELECT
UPDATE
DELETE
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza

- An `INSERT` query adds information to a table.

```
INSERT INTO
<table>
(<columns>)
VALUES
(<values>)
```

- An INSERT query adds information to a table.

```
INSERT INTO
users
(username, password, fullname)
VALUES
('newman', 'USMAIL', 'Newman')
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza

- When defining the column that ultimately is your primary key, it’s usually a good idea for that column to be an integer.

- Moreover, you can configure that column to **autoincrement**, so it will pre-populate that column for you automatically when rows are added, eliminating the risk that you’ll accidentally try to insert something with a duplicate value.

- An `INSERT` query adds information to a table.

```
INSERT INTO
moms
(username, mother)
VALUES
('kramer', 'Babs Kramer')
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

- A `SELECT` query extracts information from a table.

```
SELECT
<columns>
FROM
<table>
WHERE
<predicate>
```

- A SELECT query extracts information from a table.

```
SELECT
idnum, fullname
FROM
users
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

- A SELECT query extracts information from a table.

```
SELECT
password
FROM
users
WHERE
idnum < 12
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

- A SELECT query extracts information from a table.

```
SELECT
*
FROM
moms
WHERE
username = 'jerry'
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

- Databases empower us to organize information into tables efficiently.

- We don’t always need to store every possible relevant piece of information in the same table, but rather we can use relationships across tables to connect all the pieces of data we need.

- Let’s imagine we need to get a user’s full name (from the *users* table) and their mother’s name (from the *moms* table).

- A `SELECT (JOIN)` query extracts information from multiple tables.

```
SELECT
<columns>
FROM
<table1>
JOIN
<table2>
ON
<predicate>
```

- A `SELECT (JOIN)` query extracts information from multiple tables.

```
SELECT
users.fullname, moms.mother
FROM
users
JOIN
moms
ON
users.username = moms.username
```

- A `SELECT (JOIN)` query extracts information from multiple tables.

```sql
SELECT
users.fullname, moms.mother
FROM
users
JOIN
moms
ON
users.username = moms.username
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

users & moms

users.idnum	users.username moms.username	users.password	users.fullname	moms.mother
10	jerry	fus!ll!	Jerry Seinfeld	Helen Seinfeld
11	gcostanza	b0sc0	George Costanza	Estelle Costanza

users & moms

users.idnum	users.username moms.username	users.password	users.fullname	moms.mother
10	jerry	fus!ll!	Jerry Seinfeld	Helen Seinfeld
11	gcostanza	b0sc0	George Costanza	Estelle Costanza

- An UPDATE query modifies information in a table.

```
UPDATE
<table>
SET
<column> = <value>
WHERE
<predicate>
```

- An UPDATE query modifies information in a table.

```
UPDATE
users
SET
password = 'yadayada'
WHERE
idnum = 10
```

users

idnum	username	password	fullname
10	jerry	yadayada	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

- A `DELETE` query removes information from a table.

```
DELETE FROM
<table>
WHERE
<predicate>
```

- A DELETE query removes information from a table.

```
DELETE FROM
users
WHERE
username = 'newman'
```

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza
12	newman	USMAIL	Newman

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

users

idnum	username	password	fullname
10	jerry	fus!ll!	Jerry Seinfeld
11	gcostanza	b0sc0	George Costanza

moms

username	mother
jerry	Helen Seinfeld
gcostanza	Estelle Costanza
kramer	Babs Kramer

SQL in Python

CS50 Library allows us to use SQL in Python!

```
from cs50 import SQL
```

```
db = SQL("sqlite:///DATABASE_NAME.db")
```

```
db.execute("some SQL command")
```

Use ? in place of variables:

- `results = db.execute("SELECT id FROM courses WHERE code = ?", code)`

Lab!

Problem Set Preview

Exercises

Harvard Database

- We have information about people, courses, which people are enrolled in which courses, adn which people teach which courses
- What is the best way to store this information in a SQL database?
- How can we load this information from a CSV into SQL?
- We'll use Python to complete these exercises

What is Amia's student id?

What is the course title for CS51?

What are the course codes and titles for all of the CS courses? (Assume that all CS courses have a course code that begins with 'CS')

How many courses are there?

How many students are taking CS50?

What are the names of all of the instructors?

Generate a table with all instructors' names and the course they teach.

Switch Caroline from CS50 to CS187