This is CS50

Week 7

Open carterzenke.me/section for attendance.
Open code.cs50.io and log in!

carterzenke.me/section

Think, Pair, Share

- What are you excited about from this week's lecture?
- What do you want to learn more about?

https://carterzenke.me/section

Today

- What are **databases**? What makes for good database design?
- What is **SQL**?
- Problem Set 7

Database Design

Organizing information beautifully


goodreads

Design principles

- Create one table for each **entity** in your dataset.
- All tables should have a **primary key**.
- The information in the table should depend on the primary key *only*.

Creating a table

- In your terminal, create a database called **reads.db**
  - `sqlite3 reads.db`
- Create a table in your database to represent a book, including columns for title, subject, and publication date.

```
sqlite> CREATE TABLE table_name (
   ...>     column0 TYPE,
   ...>     column1 TYPE,
   ...>     column2 TYPE,
   ...>     column3 TYPE
   ...> );
```

```
sqlite> CREATE TABLE table_name (
   ...>     column0 INTEGER,
   ...>     column1 TEXT,
   ...>     column2 NUMERIC,
   ...>     column3 REAL
   ...> );
```

```
sqlite> CREATE TABLE table_name (
   ...>     column0 INTEGER,
   ...>     column1 TEXT,
   ...>     column2 NUMERIC,
   ...>     column3 REAL,
   ...>     PRIMARY KEY(column0)
   ...> );
```

```
sqlite> DROP TABLE table_name;
```

Inserting, Deleting

```
sqlite> INSERT INTO table (column0, column1)
   ...> VALUES (value0, value1);
```

```
sqlite> DELETE FROM table
   ...> WHERE condition;
```

Songs

Querying a database of songs

Schema

How data is organized in a database

```
$  sqlite3 DB_NAME
```

```
$  sqlite3 songs.db
```

```
sqlite> ...
```

```
sqlite>  .tables
```

songs.db

songs.db

songs

artists

```
sqlite>  .schema songs
```

```
sqlite>  SELECT * FROM songs LIMIT 3;
```

songs.db

songs

id	name	artist_id	...
1	God's Plan	23	...
2	SAD!	67	...
3	rockstar (feat. 21 Savage)	54	...
...	...	...	...

artists

songs.db

artists

id	name
23	Drake
67	XXXTENTACION
54	Post Malone
...	...

songs

Queries 1–5

SELECT

WHERE

LIKE

ORDER BY

```
SELECT column
FROM table
WHERE condition;
```

```
SELECT column
FROM table
WHERE column LIKE pattern;
```

```
SELECT column
FROM table
WHERE condition
ORDER BY column;
```

Aggregate Functions

Keywords to calculate data from multiple rows

```
SELECT column
FROM table
WHERE condition;
```

```
SELECT COUNT(column)
FROM table
WHERE condition;
```

```
SELECT AVG(column)
FROM table
WHERE condition;
```

```
SELECT MIN(column)
FROM table
WHERE condition;
```

Queries 5–7

IMDb

Querying a database of movies

movies.db

movies.db

movies

stars

people

ratings

directors

movies.db

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

stars people ratings directors

movies.db

people

id	name	birth
158	Tom Hanks	1956
5562	Owen Wilson	1968
68338	Kristen Bell	1980
...	...	...

movies stars ratings directors

movies.db

stars

movie_id	person_id
114709	158
3606752	5562
2294629	68338
...	...

movies people ratings directors

Queries 1–5

SELECT

WHERE

LIKE

ORDER BY

```
SELECT column
FROM table
WHERE condition;
```

```sql
SELECT title
FROM movies
WHERE title = 'Cars 3';
```

```sql
SELECT rating, movie_id
FROM ratings
WHERE rating >= 9.8;
```

```sql
SELECT rating, movie_id
FROM ratings
WHERE rating >= 9.8 AND votes > 100;
```

```
SELECT column
FROM table
WHERE column LIKE pattern;
```

```sql
SELECT title
FROM movies
WHERE title LIKE 'Cars%';
```

```
SELECT title
FROM movies
WHERE title LIKE '%Cars';
```

```
SELECT title
FROM movies
WHERE title LIKE '%Cars%';
```

```
SELECT column
FROM table
WHERE condition
ORDER BY column;
```

```
SELECT rating, movie_id
FROM ratings
WHERE rating > 9.8
ORDER BY rating;
```

```sql
SELECT rating, movie_id
FROM ratings
WHERE rating > 9.8
ORDER BY rating ASC;
```

```sql
SELECT rating, movie_id
FROM ratings
WHERE rating > 9.8
ORDER BY rating DESC;
```

```sql
SELECT rating, movie_id
FROM ratings
WHERE rating > 9.8
ORDER BY rating DESC, movie_id;
```

Queries 6–10

Combining Tables

Methods to reference data from other tables

SELECTs (nested)

JOINs

SELECTs (nested)

JOINs

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

```
sqlite>  SELECT id FROM movies WHERE title = "Cars 3";
```

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

```
sqlite>  SELECT id FROM movies WHERE title = "Cars 3";
```

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

```
sqlite>  SELECT rating FROM ratings WHERE movie_id = 3606752;
```

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

```
sqlite>  SELECT rating FROM ratings WHERE movie_id = 3606752;
```

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

```
sqlite>  SELECT rating FROM ratings WHERE movie_id = ?;
```

```
sqlite> SELECT rating
        FROM ratings
        WHERE movie_id = ?;
```

```
sqlite> SELECT rating
        FROM ratings
        WHERE movie_id = (
            SELECT id
            FROM movies
            WHERE title = "Cars 3"
        );
```

```
sqlite> SELECT rating
        FROM ratings
        WHERE movie_id = (
            SELECT id
            FROM movies
            WHERE title = "Cars 3"
        );
```

```
sqlite> SELECT rating
        FROM ratings
        WHERE movie_id = (
            3606752
        );
```

SELECTs (nested)

JOINs

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

movies

id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...

movies JOIN ratings

id	title	year	movie_id	rating
114709	Toy Story	1995	114709	8.3
3606752	Cars 3	2017	3606752	6.7
2294629	Frozen	2013	2294629	7.4
...	...	...	...	...

movies JOIN ratings

id	title	year	rating
114709	Toy Story	1995	8.3
3606752	Cars 3	2017	6.7
2294629	Frozen	2013	7.4
...	...	...	...

*movie_id column hidden for visualization

Queries 11–13

LIMIT

Capping the number of rows returned

```
SELECT column
FROM table
WHERE condition
LIMIT number;
```

```
SELECT column
FROM table
WHERE condition
ORDER BY column
LIMIT number;
```

```sql
SELECT movie_id, rating
FROM ratings
WHERE votes > 100
ORDER BY rating DESC
LIMIT 10;
```

INTERSECT

Returning common rows between 2 queries

```
SELECT column
FROM table
WHERE condition;
```

```
SELECT column
FROM table
WHERE condition
INTERSECT
SELECT column
FROM table
WHERE condition;
```

The week ahead

- **Submit Problem Set 7** by Sunday, March 26, 11:59 PM.
- Attend **office hours**.
- Complete `https://cs50.ly/studybuddy` to be paired with a classmate if you'd like!