Lab 5

CS50 for MBAs

carterzenke.me/lab

movies.db

IMDb

Querying a database of movies

Schema

How data is organized in a database

```
wget https://cdn.cs50.net/hbs/2023/spring/labs/5/movies.zip
```

```
$  sqlite3 DB_NAME
```

```
$  sqlite3 movies.db
```

```
sqlite> ...
```

```
sqlite>  .schema
```

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;
```

```sql
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;
```

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

Queries 6–10

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;
```

```
SELECT MIN(rating)
FROM ratings;
```

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;
```

Indexes

Strategically speeding up queries

```
CREATE INDEX indexName ON
tableName(columnName)
```

Submission

- **Submit code files to Gradescope** by Thursday, February 16, 3:10 PM.
- Graded based on completion, but please double check to be sure your files are named correctly:
  - `1.sql` **not** `1 (1).sql`