This is CS50

Week 7

Today

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

15.

No. 58. *Nazi-Maruttash, year 13th*

Contents: The payment of grain and dates as temple stipends for twelve months.

	[Nisannu]	*Airu*	*Simânu*	*Dûzu*	*Abu*	*Ulûlu*	*napḫar*	*Tashrîtu*	*Araḫshamna*	*Kislimu*	*Tebitu*	*Shabâṭu*	*Adaru*	*napḫar she'um*	*napḫar*	*a-wi-lu-tum*	*MU-BI-im*
							sha i-na qât	*mIlu*	*nabi*	*maḫ-*	*-rum*			*sha qât libbi abulli i-na libbi she'i sha Zarat-IMki u suluppu maḫrum*			
																KAL	*mSin-da-ma-qu*, "overseer."
	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	4*gur* 144*qa*	*KAL*	*mSin-mu-shab-shi*, "temple servant."
	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	4*gur* 144*qa*	*KAL*	*mIp-pa-e-a* ù(ditto)
5	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	4*gur* 144	*KAL*	*mIdinanni-Shamash*, "keeper."
	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	1 *gur*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	1 *gur*	2 *gur*	*SAL* or *f*	*Tam-bi-Da-du*, "his wife."
	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	144 *qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	144 *gur*	1*gur* 108*qa*	*SAL-TUR*	*Da-li-lu-sha*, "his daughter", "seeress."
	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	108 *qa*									*KAL-TUR*	*mArdu-Nusku*, "his son," *ultu Tashrîtu ḫarrânu.*
	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	144 *qa*	*KAL-TUR-TUR*	*mNusku-ki-na-u-ṣur*, "his [grand] son."
10	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	36 *qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	36 *qa*	72 *qa*	*TUR-GAB*	*mGab-mar-ta-ash*, "his son."
	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	72*qa*	2*gur* 72*qa*	4*gur* 144*qa*	*KAL*	*mA-na-dShe-mi-i-at-kal*, "grinder."
	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	3*gur* 36*qa*	*SAL* or *f*	*Ish-tar-be-li-uṣ-ri.*
																KAL-TUR	*mUshab-shi-uz-ni-a-na-ili*, "her son," *ḫarrânu.*
	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	144 *qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	24*qa*	144 *qa*	1*gur* 108*qa*	*KAL-TUR-TUR*	*mDu-uk-ki-in-ilu*, *hSHI*(?) "her [grand] son."
15	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	1 *gur*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	30*qa*	1 *gur*	2 *gur*	*SAL-TUR*	*Ba-su-un-du*, "her daughter," "seeress."
	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	144 *qa*	*SAL-TUR-GAB*	*Ḫu-la-la-tum*, "her daughter."
	6*qa*	6*qa*	6*qa*	6*qa*	6 *qa*	6*qa*	36 *qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	36*qa*	72 *qa*	*TUR-GAB*	*I-na-rîsh-Marduk-di-nu.* "her son."
	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	3*gur* 36*qa*	*SAL* or *f*	*Bêlit-balâṭu-îrish(-ish)*
	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	108 *qa*	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	18*qa*	108 *qa*	1*gur* 36*qa*	*KAL-TUR-TUR*	*mLul-ta-mar-Nusku*, "her son," "weaver."
20	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	12*qa*	72 *qa*	144 *qa*	*SAL-TUR-GAB*	*Rabâ-sha-dIsh-ḫa-ra*, "her daughter."
	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	36 *qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	6*qa*	36 *qa*	72 *qa*	*SAL-TUR-GAB*	*Di-ni-ili-lu-mur*, "her daughter."
	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	3*gur* 36*qa*	*SAL* or *f*	*Mi-sha-ri-tum.*
	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	48*qa*	1*gur* 108*qa*	3*gur* 36*qa*	*SAL* or *f*	*I-na-Ak-ka-di-rab-bat.*

Apple Numbers

Google Sheets

Microsoft Excel

…

Database

A collection of data organized for creating, reading, updating, and deleting.

Scale

Scale

Frequency

Scale

Frequency

Speed

Database Management System

Software via which you can interact with a database.

MySQL

Oracle

PostgreSQL

SQLite

...

SQL

A language via which you can create, read, update, and delete data in a database.

Database Design

Organizing information

goodreads

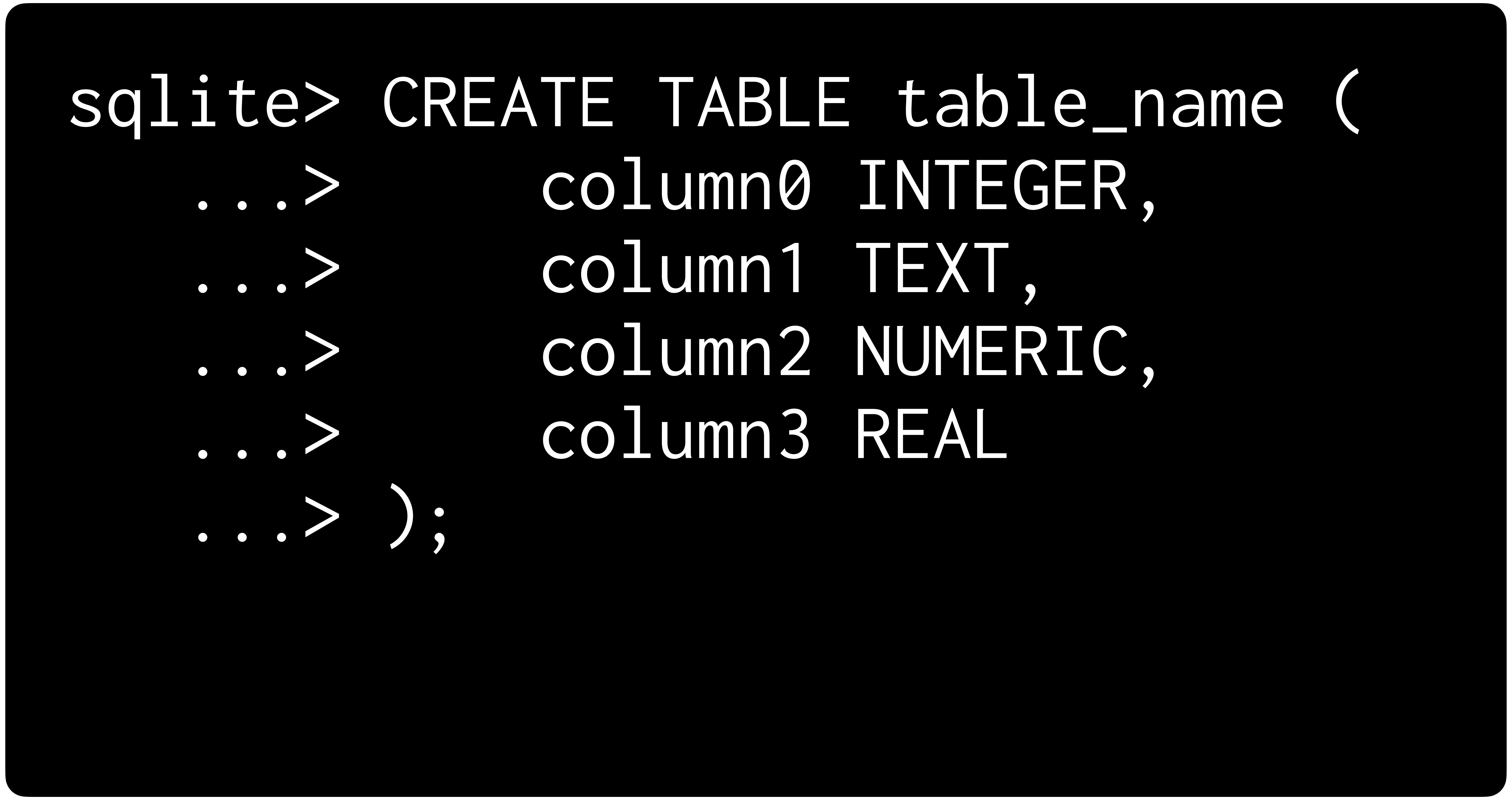
Design principles

- 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*.

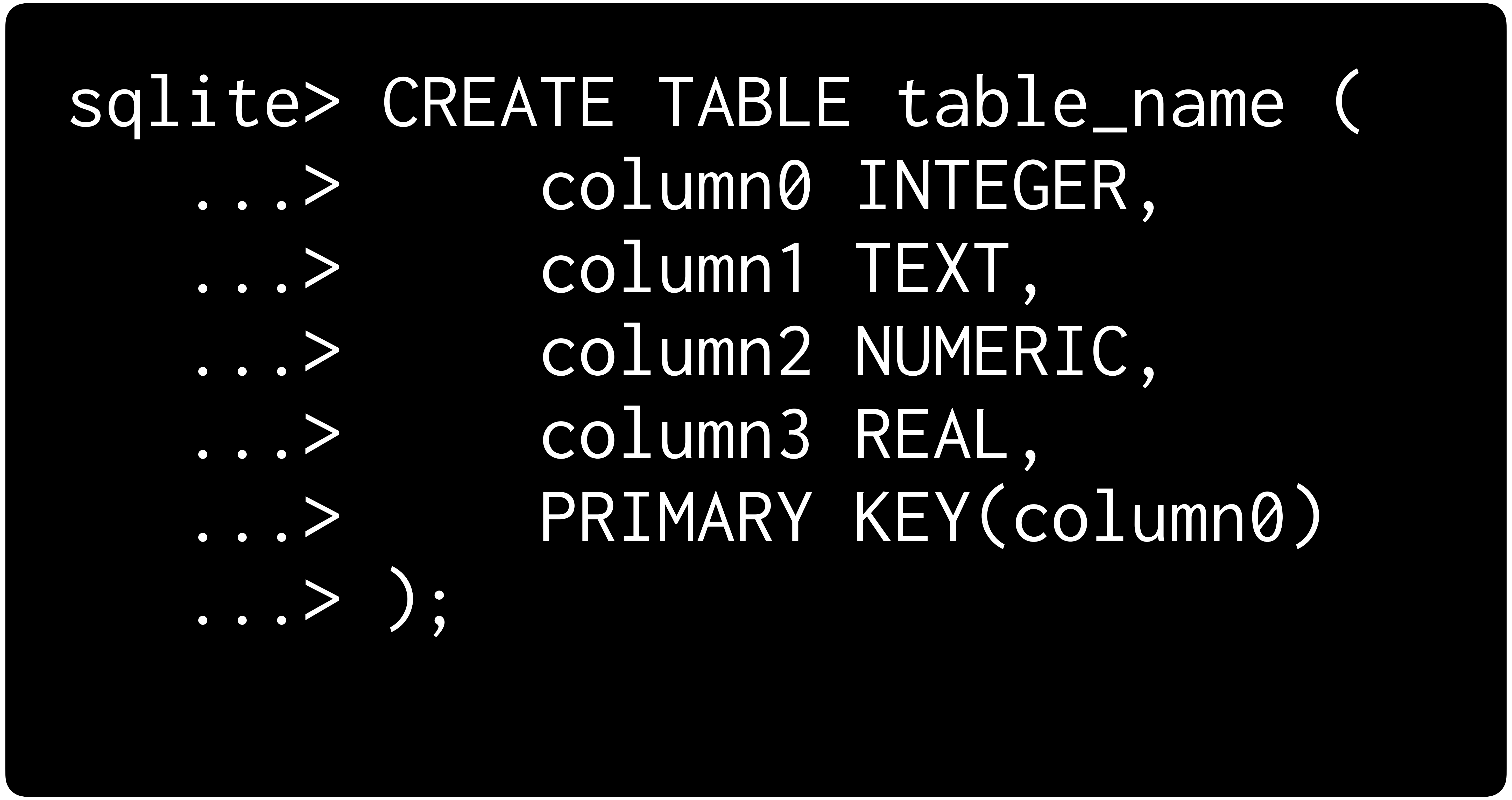
Creating a table

- In your terminal, create a database called **reads.db**
  - `sqlite3 reads.db`

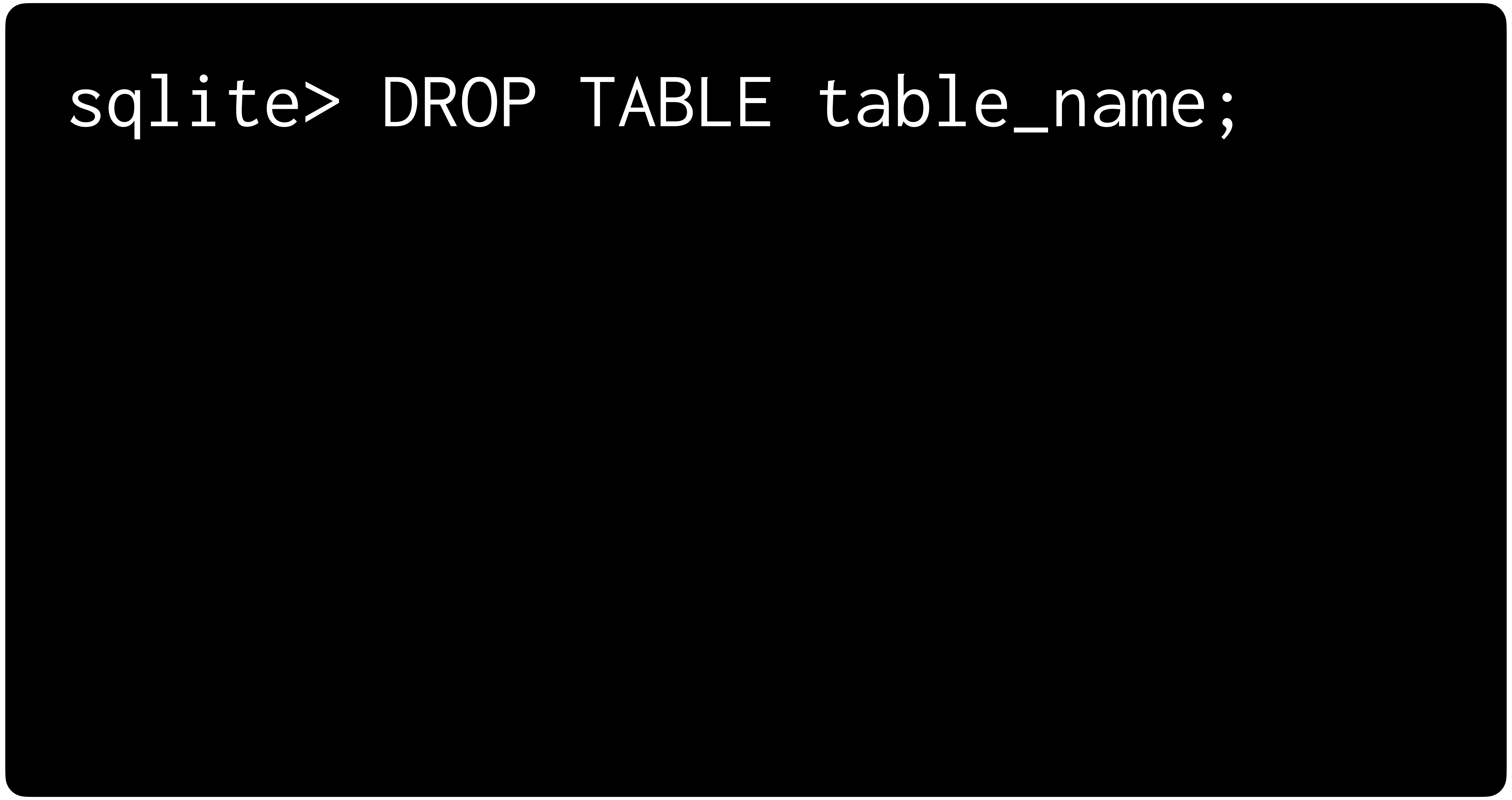
```
sqlite> CREATE TABLE table_name (
   ...>     column0 TYPE,
   ...>     column1 TYPE,
   ...>     column2 TYPE,
   ...>     column3 TYPE
   ...> );
```



```
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;
```

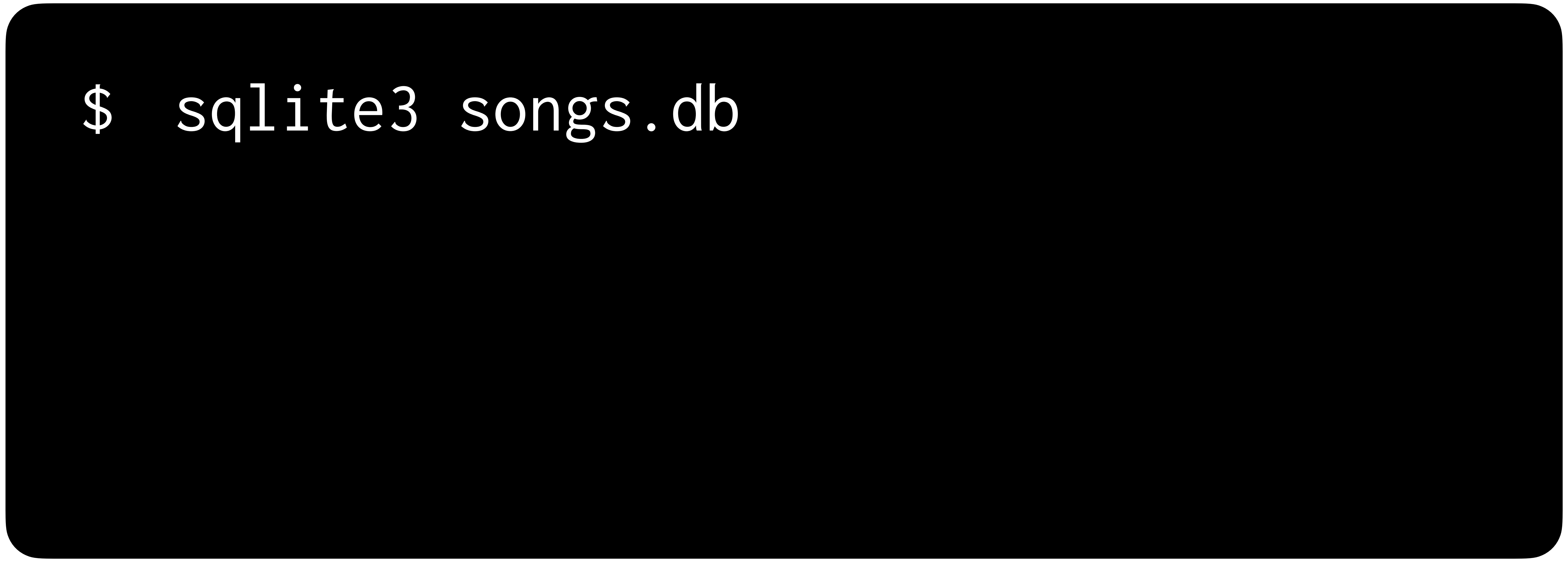
Songs

Querying a database of songs

Schema

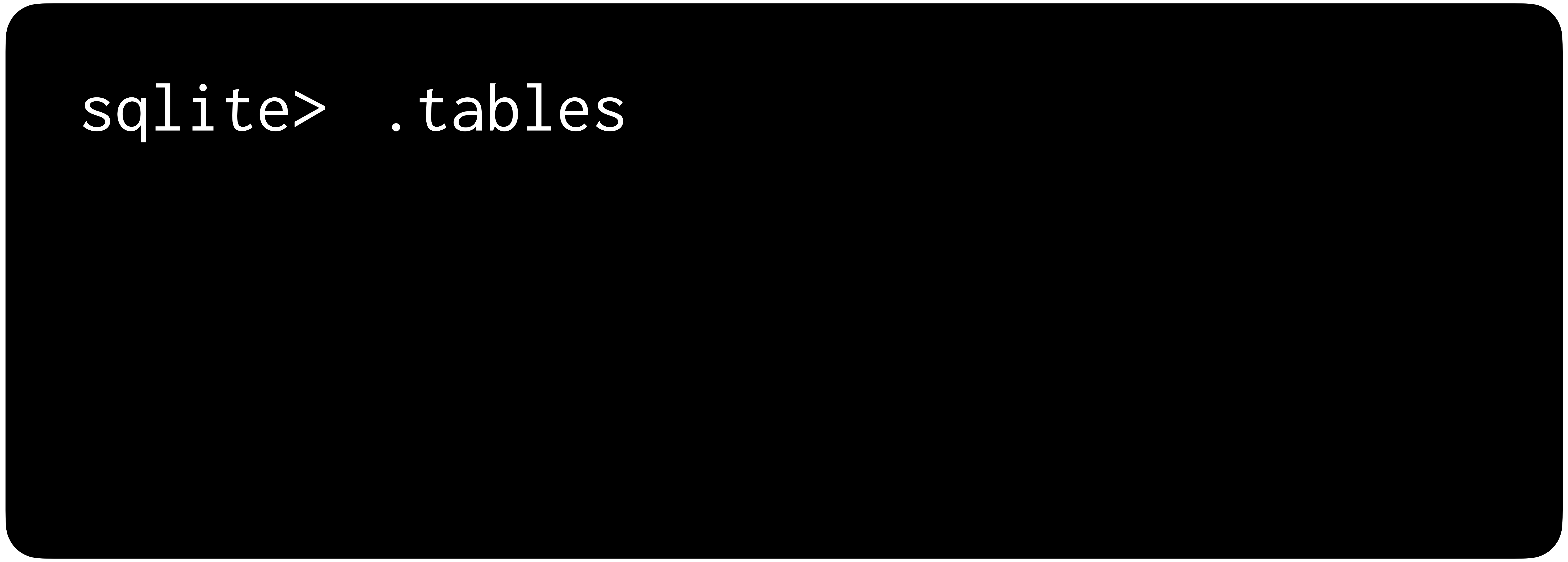
How data is organized in a database

```
$  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

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

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

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

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

Queries 5–7

Combining Tables

Methods to reference data from other tables

Subqueries

Joins

Subqueries

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
...	...

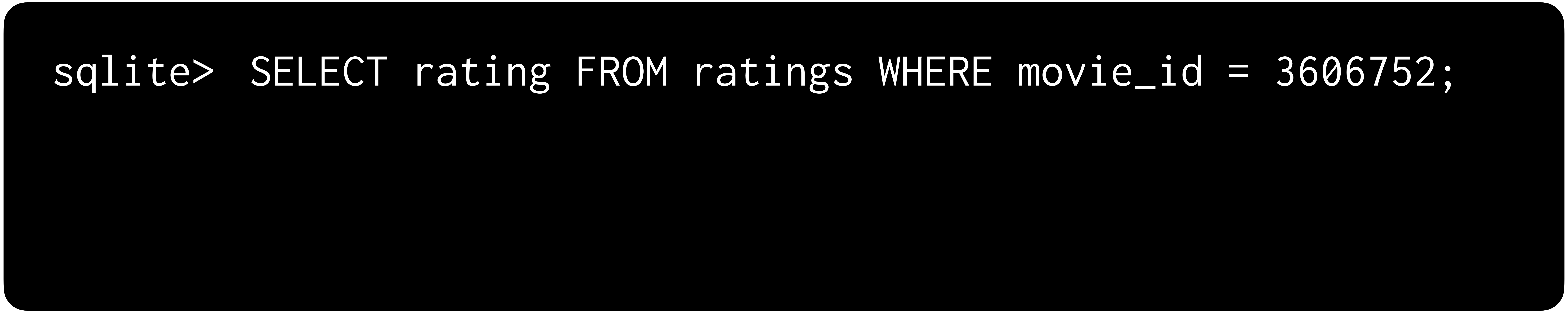
```
sqlite> SELECT id FROM movies WHERE title = 'Cars 3';
```

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
...	...

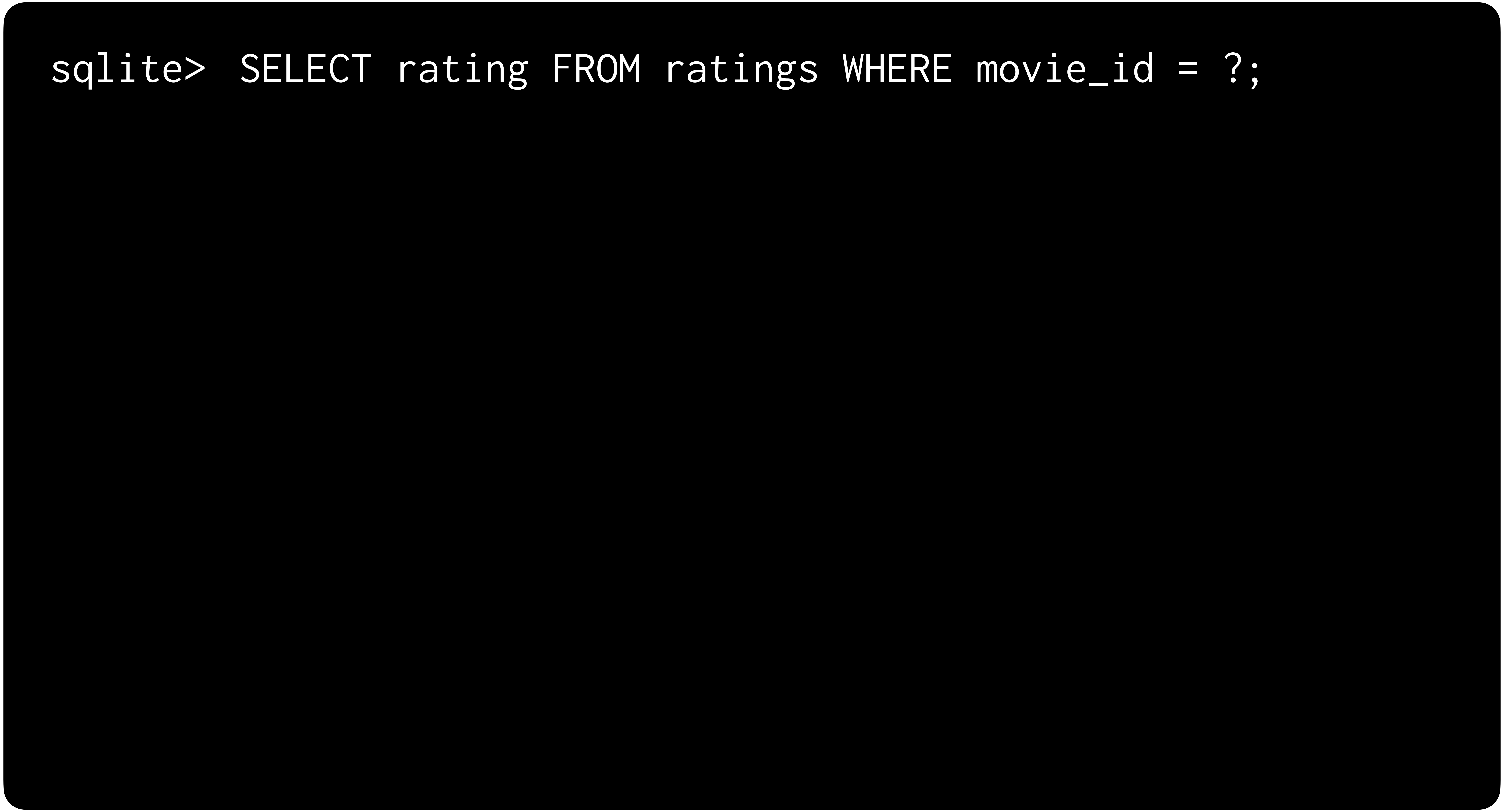
```
sqlite>  SELECT rating FROM ratings WHERE movie_id = 3606752;
```

movies

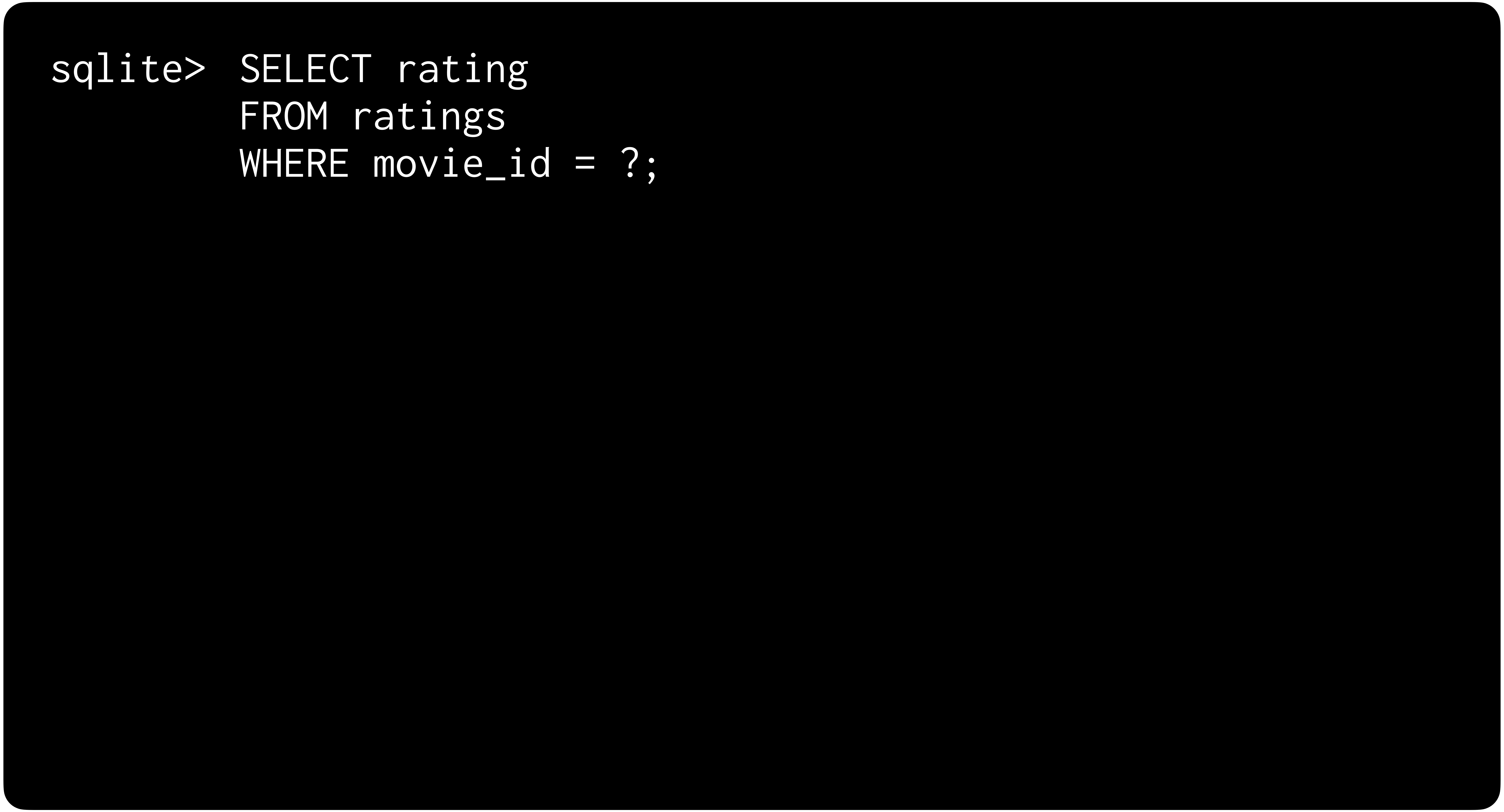
id	title	year
114709	Toy Story	1995
3606752	Cars 3	2017
2294629	Frozen	2013
...	...	...

ratings

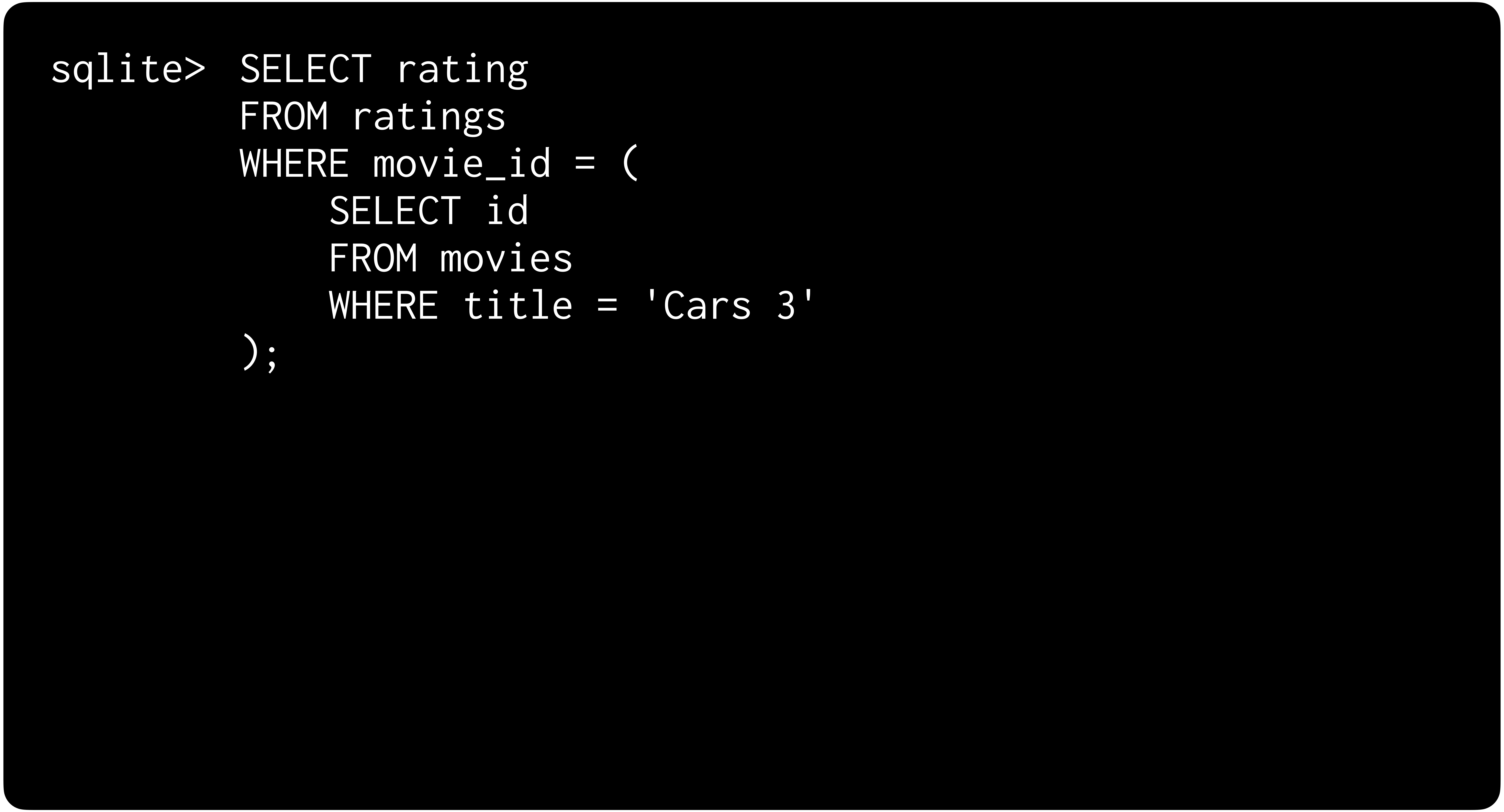
movie_id	rating
114709	8.3
3606752	6.7
2294629	7.4
...	...



```
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
         );
```

Subqueries

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

Introduction to

Databases with SQL

https://cs50.harvard.edu/sql

This is CS50

Week 7