CS50's Introduction to Databases with SQL

OpenCourseWare

Donate ⬈ (https://cs50.harvard.edu/donate)

Carter Zenke (https://carterzenke.me)
carter@cs50.harvard.edu
 (https://github.com/carterzenke) (https://www.linkedin.com/in/carterzenke/)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
 (https://www.facebook.com/dmalan) (https://github.com/dmalan) (https://
www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/) (https://
www.reddit.com/user/davidjmalan) (https://www.threads.net/@davidjmalan) (https://
twitter.com/davidjmalan)

DESE

Problem to Solve

You just landed a new job as a data analyst for the State of Massachusetts, working within
the Department of Elementary and Secondary Education (or DESE, for short!). DESE oversees
the state's public education system (https://en.wikipedia.org/wiki/
Education_in_the_United_States). One responsibility of DESE is to ensure every child has a
quality education—one in which they encounter experienced teachers, have access to an
abundance of resources, and ultimately graduate having met all requirements of the state.
Your SQL skills have a part to play in these lofty goals!

In a database called `dese.db`, answer questions about the state of education in

Massachusetts.

Demo

```
$ sqlite3
```

Distribution Code

For this problem, you'll need to download `dese.db`, along with several `.sql` files in which you'll write your queries.

▼ **Download the distribution code**

Log into [cs50.dev (https://cs50.dev/)](https://cs50.dev/), click on your terminal window, and execute `cd` by itself. You should find that your terminal window's prompt resembles the below:

```
$
```

Next execute

```
wget https://cdn.cs50.net/sql/2023/x/psets/1/dese.zip
```

in order to download a ZIP called `dese.zip` into your codespace.

Then execute

```
unzip dese.zip
```

to create a folder called `dese`. You no longer need the ZIP file, so you can execute

```
rm dese.zip
```

and respond with "y" followed by Enter at the prompt to remove the ZIP file you downloaded.

Now type

```
cd dese
```

followed by Enter to move yourself into (i.e., open) that directory. Your prompt should now resemble the below.

```
dese/ $
```

If all was successful, you should execute

```
ls
```

and see a database named `dese.db` alongside several `.sql` files. Executing `sqlite3 dese.db` should open the database in `sqlite3`, via which you'll execute SQL queries. If not, retrace your steps and see if you can determine where you went wrong!

Schema

In Massachusetts, public education is delegated to districts (https://en.wikipedia.org/wiki/School_district), a type of school government often associated with an individual town. These districts, in turn, contain many individual schools. Consider the entity relationship diagram below, which codifies the relationship between districts, schools, and other data DESE collects.
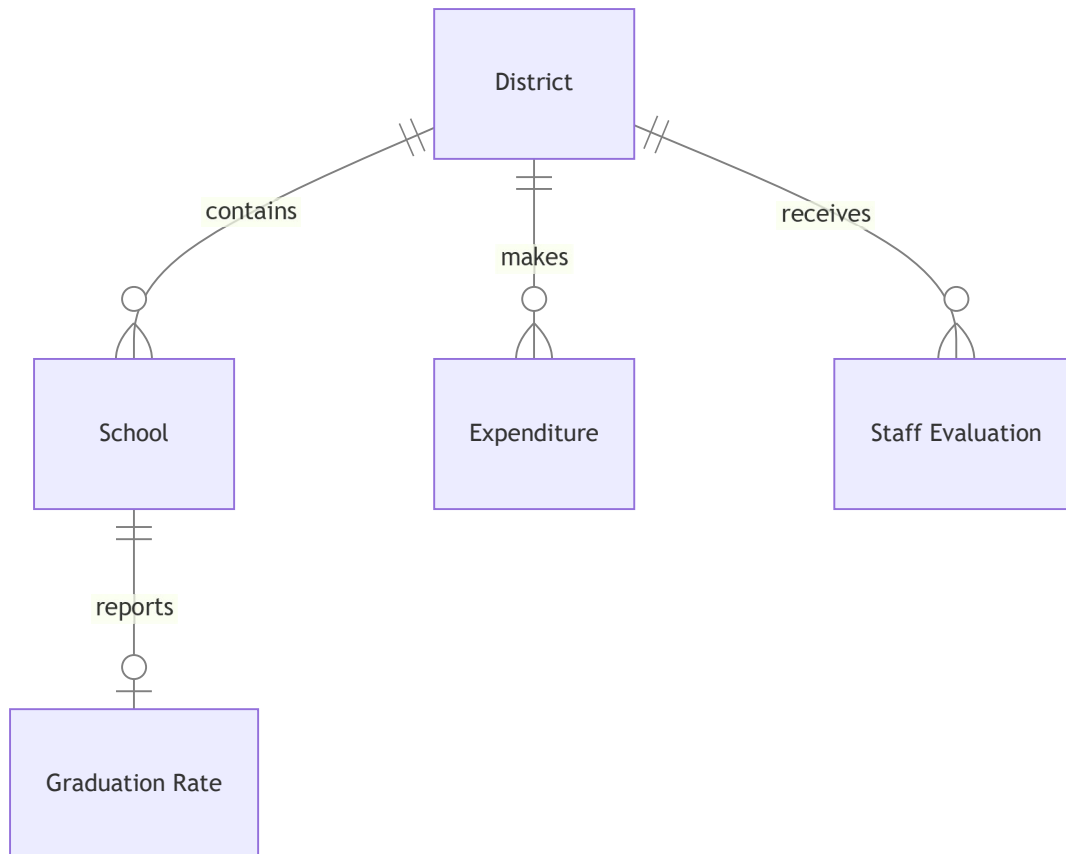
Within `dese.db`, you'll find several tables that implement the relationships described in the diagram above. Click the drop-downs below to learn more about the schema of each individual table.

▼ **`districts` table**

The `districts` table contains the following columns:

- `id`, which is the ID of the district
- `name`, which is the name of the district
- `type`, which denotes the type of district. In Massachusetts, there are public school districts (denoted "Public School District") and charter districts (denoted "Charter District"). No need to learn the difference, though you can do so from Wikipedia (https://en.wikipedia.org/wiki/Charter_school) if curious!
- `city`, which is the city in which the district is located
- `state`, which is the state in which the district is located
- `zip`, which is the ZIP Code (https://en.wikipedia.org/wiki/ZIP_Code) in which the district is located

▼ **`schools` table**

The `schools` table contains the following columns:

- `id`, which is the ID of the school

- `district_id`, which is the ID of the district to which the school belongs
- `name`, which is the name of the school
- `type`, which denotes the type of school. In Massachusetts, there are public schools (denoted "Public School") and charter schools (denoted "Charter School"). No need to learn the difference, though you can do so from Wikipedia (https://en.wikipedia.org/wiki/Charter_school) if curious!
- `city`, which is the city in which the school is located
- `state`, which is the state in which the school is located
- `zip`, which is the ZIP Code (https://en.wikipedia.org/wiki/ZIP_Code) in which the school is located

▼ `graduation_rates` table

The `graduation_rates` table contains the following columns:

- `id`, which is the ID of the graduation rate
- `school_id`, which is the ID of the school with which the graduation is associated
- `graduated`, which is the percentage of students, 0–100, who graduated on time
- `dropped`, which is the percentage of students, 0–100, who dropped out of school before graduation
- `excluded`, which is the percentage of students, 0–100, who were "excluded" (i.e., expelled)

▼ `expenditures` table

The `expenditures` table contains the following columns:

- `id`, which is the ID of the expenditure
- `district_id`, which is the ID of the district with which the expenditure is associated
- `pupils`, which is the number of pupils attending the given district
- `per_pupil_expenditure`, which is the amount of money spent, in dollars, on each student attending the district

▼ `staff_evaluations` table

The `staff_evaluations` table contains the following columns:

- `id`, which is the ID of the evaluation report
- `district_id`, which is the ID of the district with which the evaluation is associated
- `evaluated`, which is the percentage of district staff, 0–100, formally evaluated
- `exemplary`, which is the percentage of district staff, 0–100, evaluated as "exemplary"
- `proficient`, which is the percentage of district staff, 0–100, evaluated as "proficient"

- `needs_improvement`, which is the percentage of district staff, 0–100, evaluated as "needing improvement"
- `unsatisfactory`, which is the percentage of district staff, 0–100, evaluated as "unsatisfactory"

You might (cleverly!) note that some of these relationships could be implemented with a single table. You'd be correct, though in this case, you'll simply have to work with what your colleagues at DESE created!

Specification

For each of the following questions, you should write a single SQL query that outputs the results specified by each problem. Your response must take the form of a single SQL query. You **should not** assume anything about the `id`s of any particular rows: your queries should be accurate even if the `id`s were different. Finally, each query should return only the data necessary to answer the question.

`1.sql`

Your colleague is preparing a map of all public schools in Massachusetts. In `1.sql`, write a SQL query to find the names and cities of all public schools in Massachusetts.

- Keep in mind that not all schools in the `schools` table are considered traditional public schools. Massachusetts also recognizes charter schools (https://en.wikipedia.org/wiki/Charter_schools_in_the_United_States), which (according to DESE!) are considered distinct.

`2.sql`

Your team is working on archiving old data. In `2.sql`, write a SQL query to find the names of districts that are no longer operational.

- Districts that are no longer operational have "(non-op)" at the end of their name.

`3.sql`

The Massachusetts Legislature would like to learn how much money, on average, districts spent per-pupil last year. In `3.sql`, write a SQL query to find the average per-pupil expenditure. Name the column "Average District Per-Pupil Expenditure".

- Note the `per_pupil_expenditure` column in the `expenditures` table contains the average amount, per pupil, each district spent last year. You've been asked to find the average of this set of averages, weighting all districts equally regardless of their size.

`4.sql`

Some cities have more public schools than others. In `4.sql`, write a SQL query to find the 10 cities with the most public schools. Your query should return the names of the cities and the number of public schools within them, ordered from greatest number of public schools to least. If two cities have the same number of public schools, order them alphabetically.

`5.sql`

DESE would like you to determine in what cities additional public schools might be needed. In `5.sql`, write a SQL query to find cities with 3 or fewer public schools. Your query should return the names of the cities and the number of public schools within them, ordered from greatest number of public schools to least. If two cities have the same number of public schools, order them alphabetically.

`6.sql`

DESE wants to assess which schools achieved a 100% graduation rate. In `6.sql`, write a SQL query to find the names of schools (public or charter!) that reported a 100% graduation rate.

`7.sql`

DESE is preparing a report on schools in the city of Cambridge. In `7.sql`, write a SQL query to find the names of schools (public or charter!) in the Cambridge school district.

`8.sql`

A parent wants to send their child to a district with many other students. In `8.sql`, write a SQL query to display the names of all school districts and the number of pupils enrolled in each.

`9.sql`

Another parent wants to send their child to a district with few other students. In `9.sql`, write a SQL query to find the name (or names) of the school district(s) with the single least number of pupils. Report only the name(s).

`10.sql`

In Massachusetts, school district expenditures are in part determined by local taxes on property (e.g., home) values. In `10.sql`, write a SQL query to find the 10 public school districts with the highest per-pupil expenditures. Your query should return the names of the districts and the per-pupil expenditure for each.

`11.sql`

Is there a relationship between school expenditures and graduation rates? In `11.sql`, write a SQL query to display the names of schools, their per-pupil expenditure, and their graduation rate. Sort the schools from greatest per-pupil expenditure to least. If two schools have the same per-pupil expenditure, sort by school name.

- You should assume a school spends the same amount per-pupil their district as a whole spends.

`12.sql`

A parent asks you for advice on finding the best public school districts in Massachusetts. In `12.sql`, write a SQL query to find public school districts with above-average per-pupil expenditures *and* an above-average percentage of teachers rated "exemplary". Your query should return the districts' names, along with their per-pupil expenditures and percentage of teachers rated exemplary. Sort the results first by the percentage of teachers rated exemplary (high to low), then by the per-pupil expenditure (high to low).

▶ **Hint**

`13.sql`

In `13.sql`, write a SQL query to answer a question you have about the data! The query should:

- Involve at least one `JOIN` or subquery

Usage

To test your queries as you write them in your `.sql` files, you can query the database by running

```
.read FILENAME
```

where `FILENAME` is the name of the file containing your SQL query. For example,

```
.read 1.sql
```

You can also run

```
$ cat FILENAME | sqlite3 dese.db > output.txt
```

to redirect the output of the query to a text file called `output.txt`. (This can be useful for checking how many rows are returned by your query!)

How to Test

While `check50` is available for this problem, you're encouraged to instead test your code on your own for each of the following. If you're using the `dese.db` database provided in this problem's distribution, you should find that...

- Executing `1.sql` results in a table with 2 columns and 1,761 rows.
- Executing `2.sql` results in a table with 1 column and 121 rows.
- Executing `3.sql` results in a table with 1 column and 1 row.
- Executing `4.sql` results in a table with 2 columns and 10 rows.
- Executing `5.sql` results in a table with 2 columns and 201 rows.
- Executing `6.sql` results in a table with 1 column and 9 rows.
- Executing `7.sql` results in a table with 1 column and 17 rows.
- Executing `8.sql` results in a table with 2 columns and 396 rows.
- Executing `9.sql` results in a table with 1 column and 1 row.
- Executing `10.sql` results in a table with 2 columns and 10 rows.
- Executing `11.sql` results in a table with 3 columns and 391 rows.
- Executing `12.sql` results in a table with 3 columns and 65 rows.

`13.sql` is up to you!

Note that row counts do not include header rows that only show column names.

Correctness

```
check50 cs50/problems/2023/sql/dese
```

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2023/sql/dese
```

Acknowledgements

Data assembled from the [Massachusetts Department of Elementary and Secondary](#)

Education (https://www.doe.mass.edu/)'s public data sets (https://profiles.doe.mass.edu/).