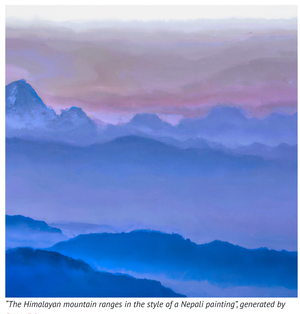
Census Taker



"The Himalayan mountain ranges in the style of a Nepali painting", generated by DALL·E 2

Problem to Solve

You are a census taker working for the Nepali government. As you crest one final hill, your breath catches at the sight of a Himalayan sunrise, casting a glow on the village you've journeyed so far to reach. Your guide, a local, halts abruptly. Underneath the steady rustle of your census papers, you feel an itch of curiosity. After all, it's not every day your job takes you to a village like this one.

In `census.db`, process your data into views the Nepali government can use for record-keeping.

Demo

Recorded with **asciinema**

Distribution Code

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

▶ **Download the distribution code**

Schema

In `census.db` you'll find a single table, `census`. In the `census` table, you'll find the following columns:

- `id`, which uniquely identifies each census record
- `district`, which is the name of the census record's district
- `locality`, which is the name of the census record's locality within the district
- `families`, which is the number of families associated with the census record
- `households`, which is the total number of households associated with the census record (multiple families may live in the same household)
- `population`, which is the population associated with the census record
- `male`, which is the number of people associated with the census record who have identified as male
- `female`, which is the number of people associated with the census record who have identified as female

Specification

In each of the corresponding `.sql` files, write a SQL statement to create each of the following views of the data in `census.db`. Note that, while views can be created from other views, each of your views should stand alone (i.e., not rely on a prior view).

Rural

In `rural.sql`, write a SQL statement to create a view named `rural`. This view should contain all census records relating to a rural municipality (identified by including "rural" in their name). Ensure the view contains all of the columns from the `census` table.

Total

In `total.sql`, write a SQL statement to create a view named `total`. This view should contain the sums for each numeric column in `census`, across all districts and localities. Ensure the view contains each of the following columns:

- `families`, which is the sum of families from every locality in Nepal.
- `households`, which is the sum of households from every locality in Nepal.
- `population`, which is the sum of the population from every locality in Nepal.
- `male`, which is the sum of people identifying as male from every locality in Nepal.
- `female`, which is the sum of people identifying as female from every locality in Nepal.

By District

In `by_district.sql`, write a SQL statement to create a view named `by_district`. This view should contain the sums for each numeric column in `census`, grouped by `district`. Ensure the view contains each of the following columns:

- `district`, which is the name of the district.
- `families`, which is the total number of families in the district.
- `households`, which is the total number of households in the district.
- `population`, which is the total population of the district.
- `male`, which is the total number of people identifying as male in the district.
- `female`, which is the total number of people identifying as female in the district.

Most Populated

In `most_populated.sql`, write a SQL statement to create a view named `most_populated`. This view should contain, in order from greatest to least, the most populated *districts* in Nepal. Ensure the view contains each of the following columns:

- `district`, which is the name of the district.
- `families`, which is the total number of families in the district.
- `households`, which is the total number of households in the district.
- `population`, which is the total population of the district.
- `male`, which is the total number of people identifying as male in the district.
- `female`, which is the total number of people identifying as female in the district.

Usage

To test your views as you write them in your `.sql` files, you can run a query on the database by running

```
.read FILENAME
```

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

```
.read rural.sql
```

Keep in mind you can also use

```
DROP VIEW name;
```

where `name` is the name of your view, to remove a view before creating it anew.

How to Test

While `check50` is available for this problem, you're encouraged to also test your code on your own. You might try queries like the below:

- How many rural districts are there? How many families live in rural districts? Using your `rural` view, you should find there are 461 rural districts with 2,229,834 families.
- How many households are in Nepal? Using your `total` view, you should find there are 5,642,674.
- Which district has the second lowest number of families? And how many does it have? Using your `by_district` view, you should find that the Mustang district has only 3,751 families.
- Which district has the highest population? And how many households are in that district? Using your `most_populated` view, you should find that the most populated is Kathmandu with 275,806 households.

Correctness

```
check50 cs50/problems/2024/sql/census
```

How to Submit

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

```
submit50 cs50/problems/2024/sql/census
```

Acknowledgements

Data retrieved from Open Data Nepal, opendatanepal.com.