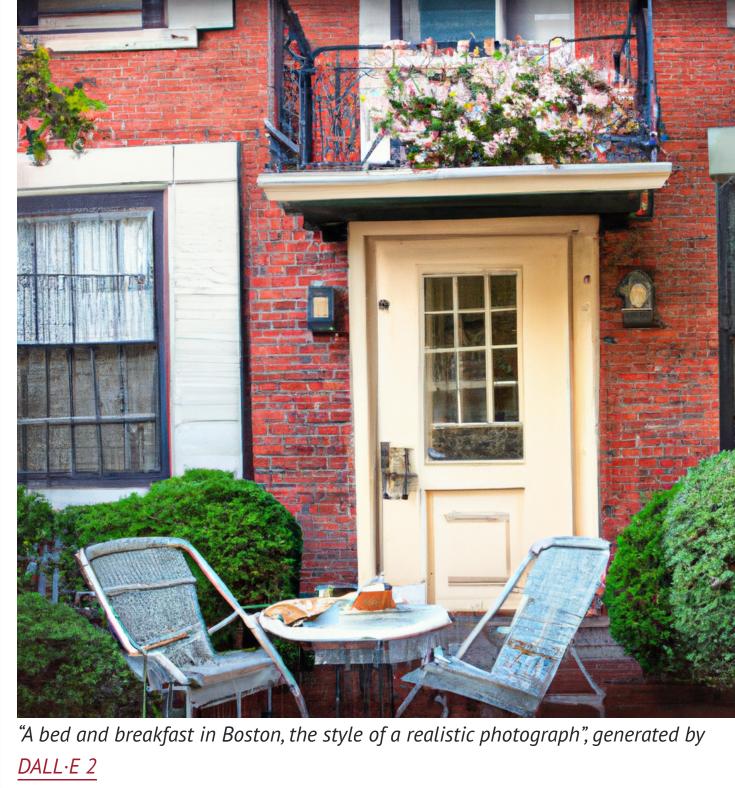


Bed and Breakfast



Problem to Solve

A Bed and Breakfast ("BnB" for short!) is a short-term place one might stay and pay the owner for the service, similar to a hotel. Over the past few years, AirBnB

has allowed most anyone to rent out their place, whether it's a home, a cute cottage, or even a treehouse. You're a data analyst for the City of Boston and you're interested in discovering how the rise of AirBnB has changed the local tourist scene. You've even

compiled a database, bnb.db, filled with data directly from AirBnB. In bnb.db, whip up a few views that will paint a clearer picture of AirBnB's influence on the city of Boston.

Demo

```
$ sqlite3 bnb.db
sqlite> SELECT "property_type", "host_name", "bedrooms"
   ···> FROM
```

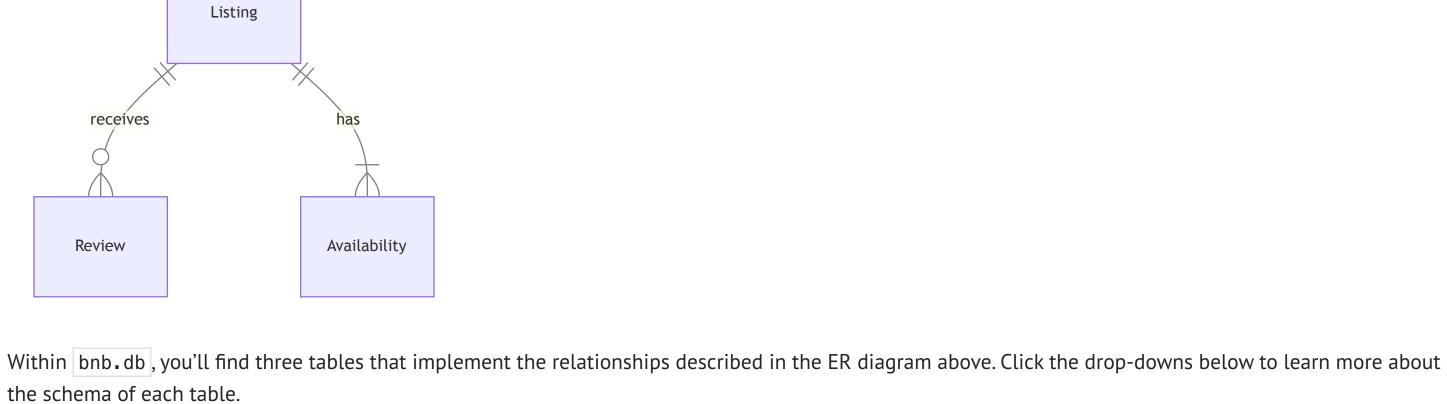
Recorded with asciinema

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

Distribution Code

▶ Download the distribution code

Schema



▼ listings table

• id, which is the ID of the listing.

■ property_type, which is the type of the listing (e.g., "Entire rental unit", "Private room in rental unit", etc.). host_name , which is the AirBnB username of the listing's host.

The listings table contains the following columns:

- accommodates, which is the listing's maximum number of occupants. bedrooms , which is the listing's number of bedrooms.
- description, which is the description of the listing on AirBnB.
- **▼** reviews table
- The reviews table contains the following columns: • id, which is the ID of the review.

listing_id, which is the ID of the listing which received the review.

- date, which is the date the review was posted. reviewer_name, which is the AirBnB username of the reviewer.
- comments , which is the content of the review.
- **▼** availabilities table
- id, which is the id of the availability.

listing_id , which is the listing ID associated with the availability.

The availabilities table contains the following columns:

date, which is the date of the availability.

SELECT * FROM "listings" LIMIT 5;

- available , which is whether the date is still available to be booked (TRUE or FALSE). price, which is the price of staying on the given date.
- **Specification**

created from other views, each of your views should stand alone (i.e., not rely on a prior view).

No Descriptions

You might notice that when running

In each of the corresponding sql files, write a SQL statement to create each of the following views of the data in bnb.db. Note that, while views can be

the results look quite wonky! The description column contains descriptions with many line breaks, each of which are printed to your terminal. In no_descriptions.sql, write a SQL statement to create a view named no_descriptions that includes all of the columns in the listings table except

for description.

One-Bedrooms

In one_bedrooms.sql, write a SQL statement to create a view named one_bedrooms. This view should contain all listings that have exactly one bedroom.

property_type, from the listings table. host_name , from the listings table.

■ id, which is the id of the listing from the listings table.

accommodates, from the listings table. **Available**

Ensure the view contains the following columns:

view contains the following columns:

• id, which is the id of the listing from the listings table. property_type, from the listings table. host_name , from the listings table.

In available.sql, write a SQL statement to create a view named available. This view should contain all dates that are available at all listings. Ensure the

date, from the availabilities table, which is the date of the availability. **Frequently Reviewed**

• id, which is the id of the listing from the listings table.

In frequently_reviewed.sql, write a SQL statement to create a view named frequently_reviewed. This view should contain the 100 most frequently reviewed listings, sorted from most- to least-frequently reviewed. Ensure the view contains the following columns:

property_type, from the listings table.

- host_name , from the listings table. reviews , which is the number of reviews the listing has received. If any two listings have the same number of reviews, sort by property_type (in alphabetical order), followed by host_name (in alphabetical order).
- **June Vacancies**

In june_vacancies.sql, write a SQL statement to create a view named june_vacancies. This view should contain all listings and the number of days in June of 2023 that they remained vacant. Ensure the view contains the following columns: • id, which is the id of the listing from the listings table.

property_type, from the listings table.

host_name , from the listings table. days_vacant, which is the number of days in June of 2023, that the given listing was marked as available.

- 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 no_descriptions.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.

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 listings are there in total? Use your no_descriptions view to find that there are 3,973.

■ How many one-bedroom listings are there? And how many can accommodate at least 4 guests? Use your one_bedrooms view to find that of the 1,228 one-bedrooms, 222 of them can accommodate your group of 4.

How to Test

■ How many listings have availability for December 31st, 2023 (i.e., "2023-12-31")? Use your available view to find that there are 2,251. How many of those are available on any type of boat? You should find that there are 7. Enjoy your New Year's Eve afloat! ■ How many reviews does the most frequently reviewed property have? And who is the host of that property? Use your | frequently_reviewed | view to

- find that Tiffany's property has 860 reviews. ■ How many listings were available in June 2023? Use your june_vacancies view to find that there were 1,895 vacancies.
- your query runs correctly on the database you've downloaded, it will also run correctly on check50 's database! If your query runs incorrectly on check50 's database, it's also running incorrectly on your own database.

For performance's sake, check50 uses a smaller database than you've downloaded. For that reason, results may not match between the two databases. If

Correctness

How to Submit

Acknowledgements

Data retrieved from insideairbnb.com.

check50 cs50/problems/2024/sql/bnb

submit50 cs50/problems/2024/sql/bnb

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