Code First Girls 2023 : Introduction to Data & SQL

Sponsored by Deloitte

Final Data Analytics Project Report

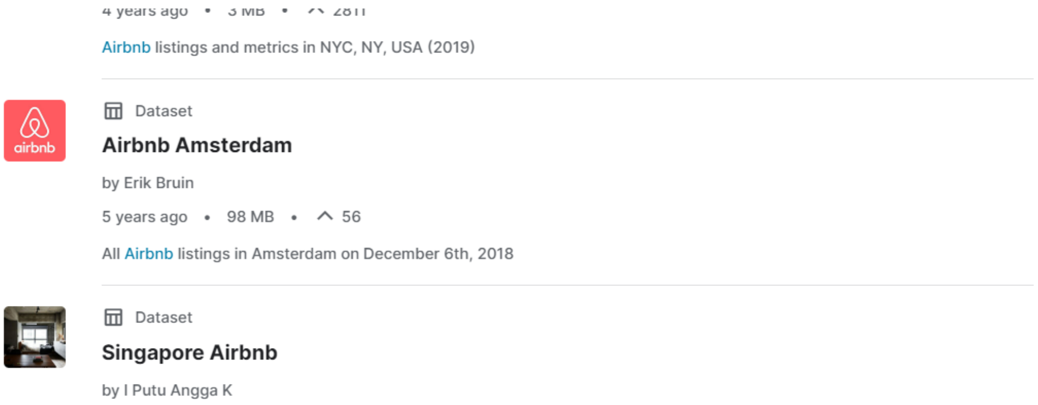
Sharmada Nagarajan

Project Topic: **Airbnb Amsterdam**

TABLE OF CONTENTS

STEP 1: DOWNLOAD DATA & PREP CSV. FILES

- Select & download the dataset from kaggle.com.
- For this assignment, **I chose to work on an airbnb Amsterdam dataset**.
- I chose this dataset as it had multiple tables, columns that I could work with and create a relational database on MySQL.



4 years ago • 3 MB • ^ 2811

Airbnb listings and metrics in NYC, NY, USA (2019)

Dataset

Airbnb Amsterdam

by Erik Bruin

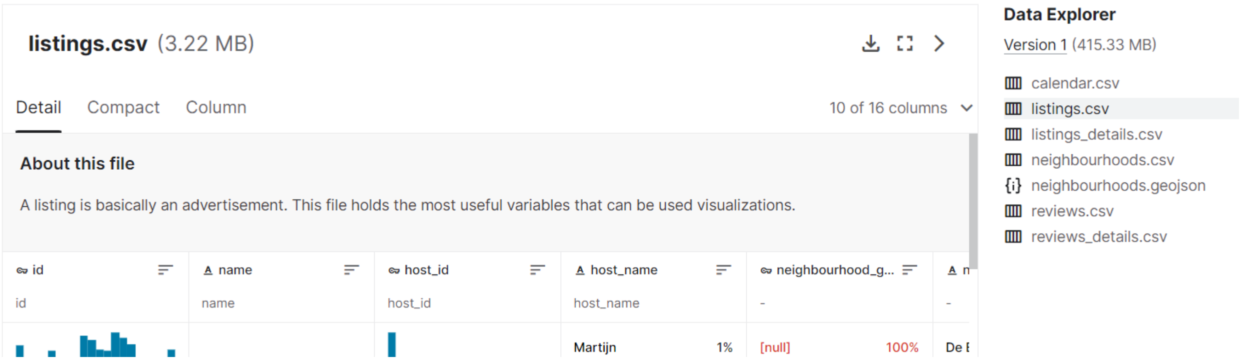
5 years ago • 98 MB • ^ 56

All Airbnb listings in Amsterdam on December 6th, 2018

Dataset

Singapore Airbnb

by I Putu Angga K



listings.csv (3.22 MB)

About this file

A listing is basically an advertisement. This file holds the most useful variables that can be used visualizations.

id	name	host_id	host_name	neighbourhood_g...	n
id	name	host_id	host_name	-	-
			Martijn 1%	[null] 100%	De E

Data Explorer

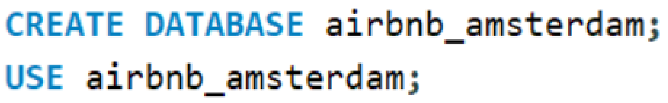
Version 1 (415.33 MB)

- calendar.csv
- listings.csv
- listings_details.csv
- neighbourhoods.csv
- neighbourhoods.geojson
- reviews.csv
- reviews_details.csv

- After downloading the dataset, I cleaned up the data & selected the relevant columns into separate csv files.
- Upon inspection, I was able to group the data into 3 bigger tables - listings, listing_details & reviews_details.
- For the 4th table, I decided to focus on host_details.
- As the data was organised around listings as the main focus, I will need to filter/ derive host_details separately using SELECT DISTINCT syntax on MySQL & export the results to create the new table. (code shown in STEP 3)

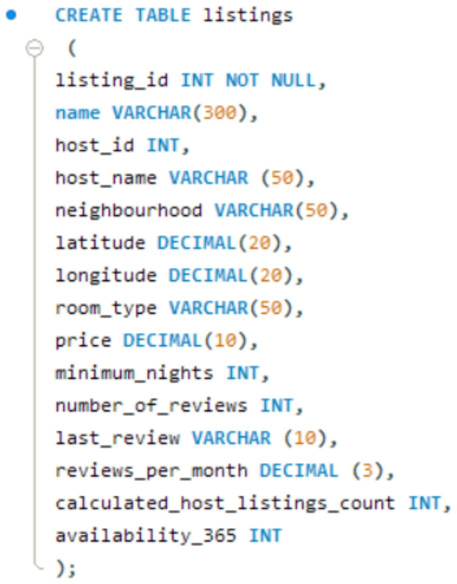
STEP 2: CREATE THE RELATIONAL DB & TABLES

- First, I created the database to use as shown below:

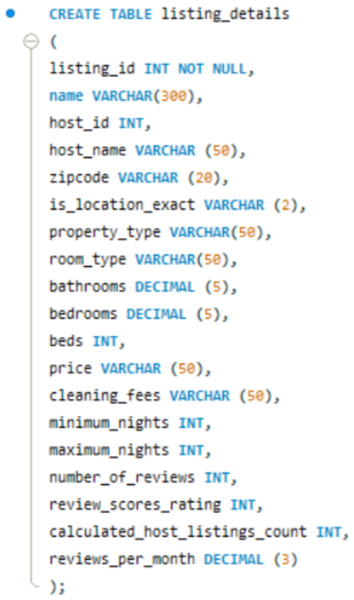


```
CREATE DATABASE airbnb_amsterdam;
USE airbnb_amsterdam;
```

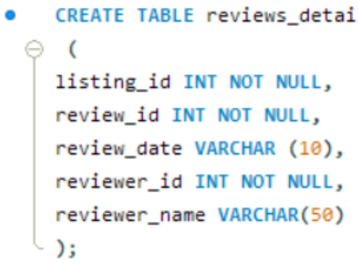
- Next, I created the tables in alignment with the csv files.
  - **listings**: an overview of airbnb listings in Amsterdam
  - **listings_details**: a detailed table giving information on the listing & property specifications
  - **reviews_details**: an overview of each review of every airbnb listing in the **listings** table



```
CREATE TABLE listings
 (
listing_id INT NOT NULL,
name VARCHAR(300),
host_id INT,
host_name VARCHAR (50),
neighbourhood VARCHAR(50),
latitude DECIMAL(20),
longitude DECIMAL(20),
room_type VARCHAR(50),
price DECIMAL(10),
minimum_nights INT,
number_of_reviews INT,
last_review VARCHAR (10),
reviews_per_month DECIMAL (3),
calculated_host_listings_count INT,
availability_365 INT
);
```



```
CREATE TABLE listing_details
(
listing_id INT NOT NULL,
name VARCHAR(300),
host_id INT,
host_name VARCHAR (50),
zipcode VARCHAR (20),
is_location_exact VARCHAR (2),
property_type VARCHAR(50),
room_type VARCHAR(50),
bathrooms DECIMAL (5),
bedrooms DECIMAL (5),
beds INT,
price VARCHAR (50),
cleaning_fees VARCHAR (50),
minimum_nights INT,
maximum_nights INT,
number_of_reviews INT,
review_scores_rating INT,
calculated_host_listings_count INT,
reviews_per_month DECIMAL (3)
);
```



```
CREATE TABLE reviews_details
 (
listing_id INT NOT NULL,
review_id INT NOT NULL,
review_date VARCHAR (10),
reviewer_id INT NOT NULL,
reviewer_name VARCHAR(50)
);
```

STEP 3: IMPORTING CSV. DATA INTO MYSQL WORKBENCH

- On the Schemas tab, I right clicked on the relevant database and selected the Data Import Wizard.

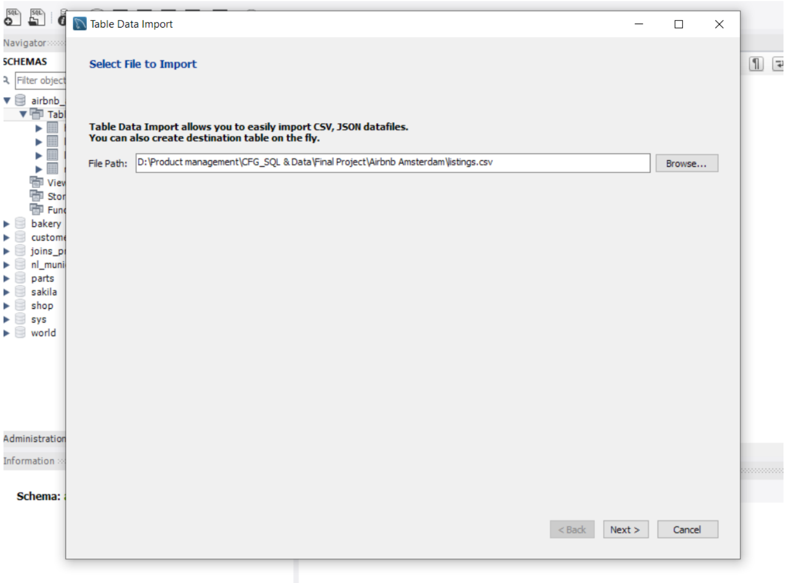


Table Data Import

Select File to Import

Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.

File Path: D:\Product management\CFG_SQL & Data\Final Project\Airbnb Amsterdam\listings.csv

- I then selected the path for the csv file and selected the correct destination. In this case, I used an existing table as I had already created the necessary tables within the **airbnb_amsterdam** database.
- Following this, I checked whether the columns match and proceeded to import the data.

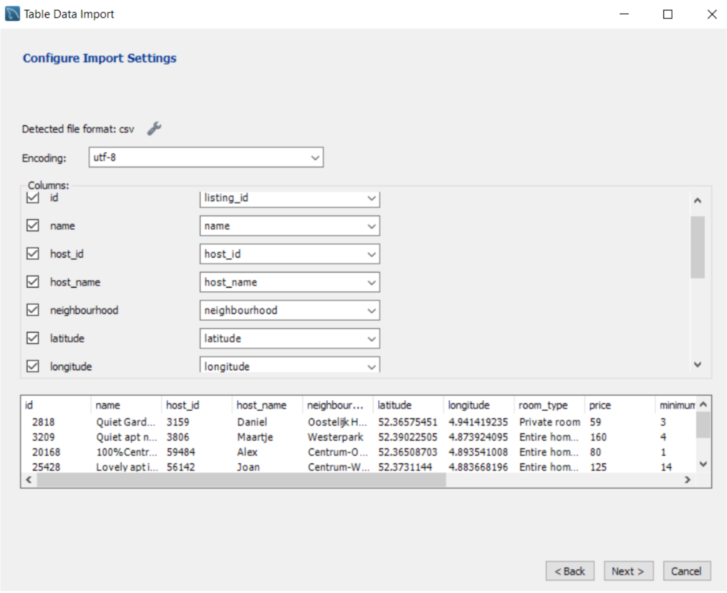


Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

☑ id	listing_id
☑ name	name
☑ host_id	host_id
☑ host_name	host_name
☑ neighbourhood	neighbourhood
☑ latitude	latitude
☑ longitude	longitude

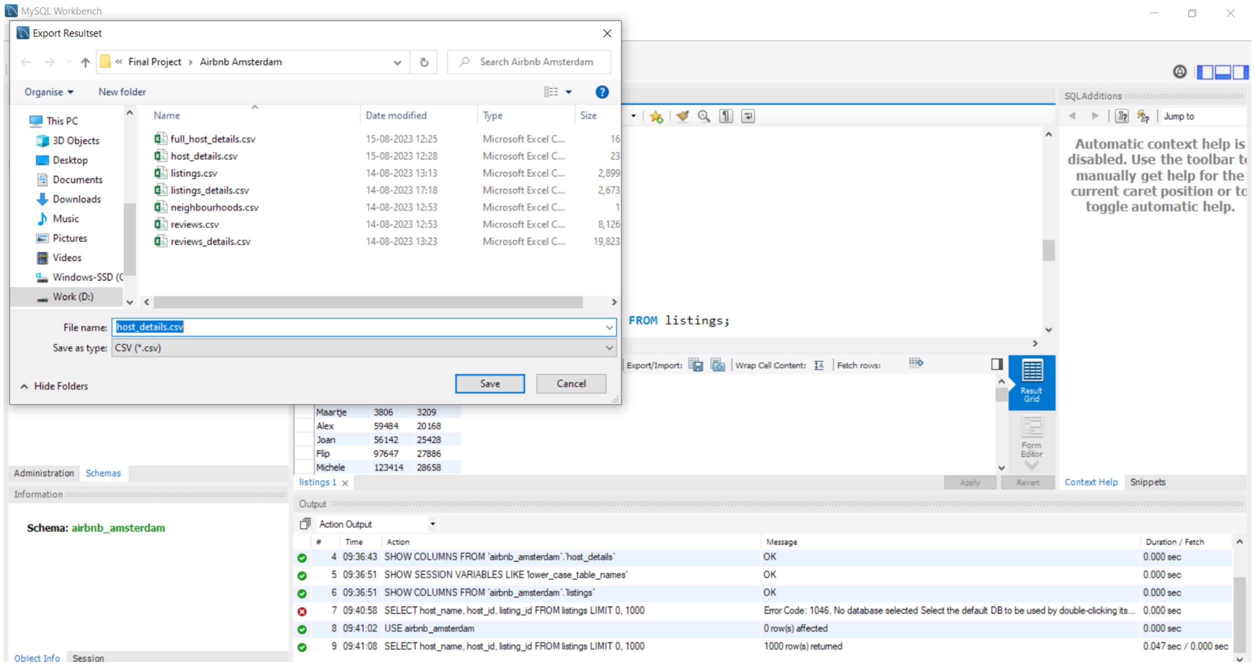
id	name	host_id	host_name	neighbour...	latitude	longitude	room_type	price	minimum
2818	Quiet Gard...	3159	Daniel	Oostelijk H...	52.36575451	4.941419235	Private room	59	3
3209	Quiet apt n...	3806	Maartje	Westerpark	52.39022505	4.873924095	Entire hom...	160	4
20168	100%Centr...	59484	Alex	Centrum-O...	52.36508703	4.893541008	Entire hom...	80	1
25428	Lovely apt i...	56142	Joan	Centrum-W...	52.3731144	4.883668196	Entire hom...	125	14

- Upon completion, the window displays the number of records imported.

- I repeat this for the other two tables as well.
- At this point, I had created and filled the main three tables. However, for the 4th table (host_details), I used the following SELECT query to get the necessary data and export as a new table:

```sql
SELECT DISTINCT host_id, host_name, listing_id FROM host_details;
```

- Following this, I exported the results as a new csv file, which I then imported back into the database after creating a new host_details table as shown below:



Name	Date modified	Type	Size
full_host_details.csv	15-08-2023 12:25	Microsoft Excel C...	16
host_details.csv	15-08-2023 12:28	Microsoft Excel C...	23
listings.csv	14-08-2023 13:13	Microsoft Excel C...	2,899
listings_details.csv	14-08-2023 17:18	Microsoft Excel C...	2,673
neighbourhoods.csv	14-08-2023 12:53	Microsoft Excel C...	1
reviews.csv	14-08-2023 12:53	Microsoft Excel C...	8,126
reviews_details.csv	14-08-2023 13:23	Microsoft Excel C...	19,823

File name: host_details.csv

Save as type: CSV (*.csv)

```
FROM listings;
```

Schema: airbnb_amsterdam

Maartje	3806	3209
Alex	59484	20168
Joan	56142	25428
Flip	97647	27886
Michele	123414	28658

listings 1

Output

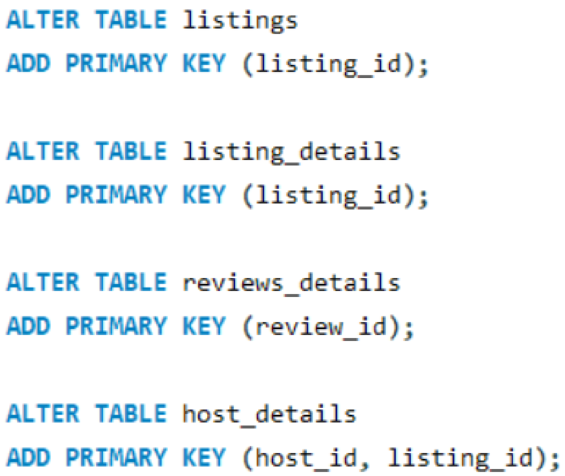
Action Output

#	Time	Action	Message	Duration / Fetch
4	09:36:43	SHOW COLUMNS FROM \`airbnb_amsterdam\`.\`host_details\`	OK	0.000 sec
5	09:36:51	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
6	09:36:51	SHOW COLUMNS FROM \`airbnb_amsterdam\`.\`listings\`	OK	0.000 sec
7	09:40:58	SELECT host_name, host_id, listing_id FROM listings LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its...	0.000 sec
8	09:41:02	USE airbnb_amsterdam	0 row(s) affected	0.000 sec
9	09:41:08	SELECT host_name, host_id, listing_id FROM listings LIMIT 0, 1000	1000 row(s) returned	0.047 sec / 0.000 sec

```sql
CREATE TABLE host_details
(
host_id INT NOT NULL,
host_name VARCHAR (50),
listing_id INT NOT NULL
);
```

STEP 4: SETTING PRIMARY & FOREIGN KEYS

- I added the PKs for the three tables - listings, listing_details, reviews_details. This was possible as these tables contained a unique id for each row which could be used as the PK.
- The host_details table cannot hold a single column as PK as there is no unique identifier for each row. So I created a composite PK using the combination of host_id, listing_id.



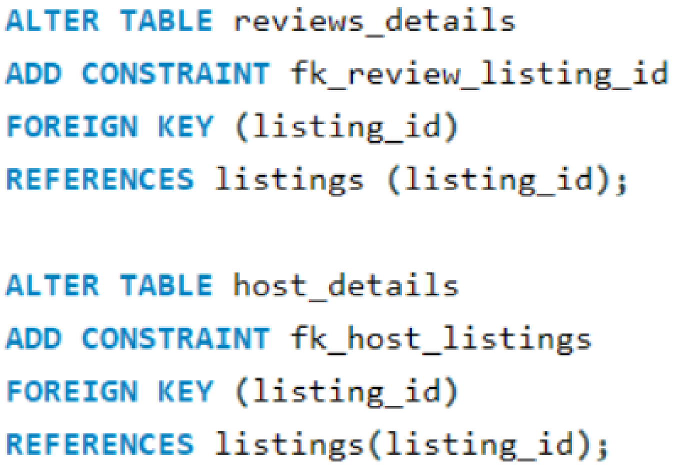
```
ALTER TABLE listings
ADD PRIMARY KEY (listing_id);

ALTER TABLE listing_details
ADD PRIMARY KEY (listing_id);

ALTER TABLE reviews_details
ADD PRIMARY KEY (review_id);

ALTER TABLE host_details
ADD PRIMARY KEY (host_id, listing_id);
```

- I then proceeded to set listing_id as the foreign keys for the reviews_details & host_details tables as shown below:



```
ALTER TABLE reviews_details
ADD CONSTRAINT fk_review_listing_id
FOREIGN KEY (listing_id)
REFERENCES listings (listing_id);

ALTER TABLE host_details
ADD CONSTRAINT fk_host_listings
FOREIGN KEY (listing_id)
REFERENCES listings(listing_id);
```

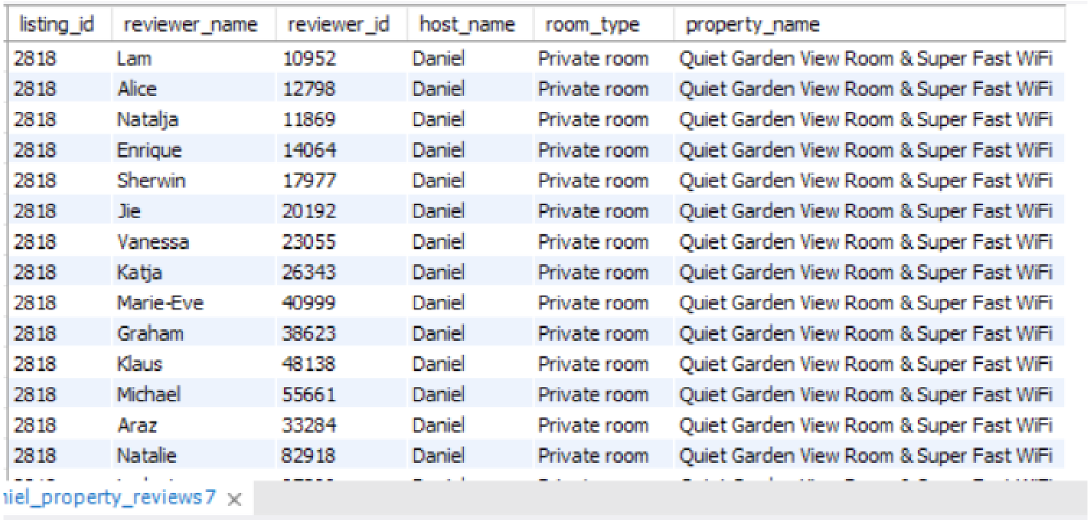
STEP 5: DATABASE ANALYSIS

1. **<u>Creating views using joins</u>**

- For this step, I wanted to capture **an overview of the reviewers who had reviewed the listings by a host named Daniel.** For this, I needed the listing id, reviewer names & id, host name (Daniel), property name (in listings table as 'name', and the room type.

```
CREATE VIEW daniel_property_reviews
  AS
   SELECT
    listings.listing_id,
    reviews_details.reviewer_name,
    reviews_details.reviewer_id,
    listings.host_name,
    listings.room_type,
    listings.name
      AS property_name
        FROM listings
          INNER JOIN
             reviews_details
                 ON
                   listings.listing_id = reviews_details.listing_id
                         WHERE
                         listings.host_name = 'Daniel';
```

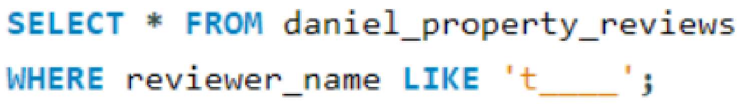
- The query/ view above returned a result of 248 rows - indicating the total number of reviews that a property advertised by Daniel had received.
- The results are as follows:



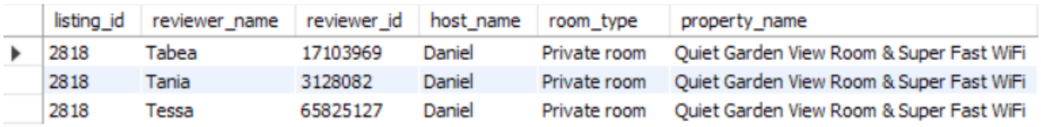
listing_id	reviewer_name	reviewer_id	host_name	room_type	property_name
2818	Lam	10952	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Alice	12798	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Natalja	11869	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Enrique	14064	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Sherwin	17977	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Jie	20192	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Vanessa	23055	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Katja	26343	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Marie-Eve	40999	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Graham	38623	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Klaus	48138	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Michael	55661	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Araz	33284	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Natalie	82918	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi

2. **Creating a query to use the generated view for data analysis**

   a) **Find out the names of reviewers starting with T and having 5 alphabets in their names:**

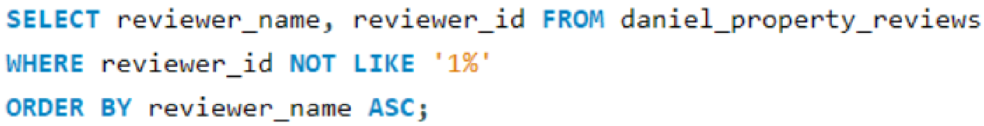


```
SELECT * FROM daniel_property_reviews
WHERE reviewer_name LIKE 't____';
```

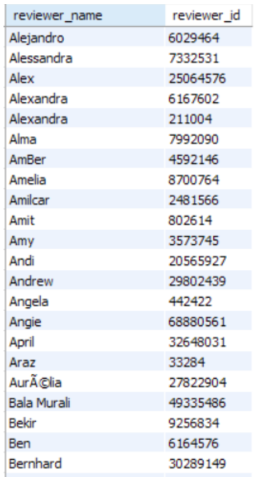


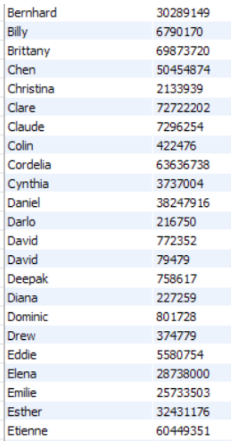
listing_id	reviewer_name	reviewer_id	host_name	room_type	property_name
2818	Tabea	17103969	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Tania	3128082	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi
2818	Tessa	65825127	Daniel	Private room	Quiet Garden View Room & Super Fast WiFi

   b) **Find out the details of reviewers in alphabetical order of their names whose reviewer_id doesn't begin with 1:**



```
SELECT reviewer_name, reviewer_id FROM daniel_property_reviews
WHERE reviewer_id NOT LIKE '1%'
ORDER BY reviewer_name ASC;
```

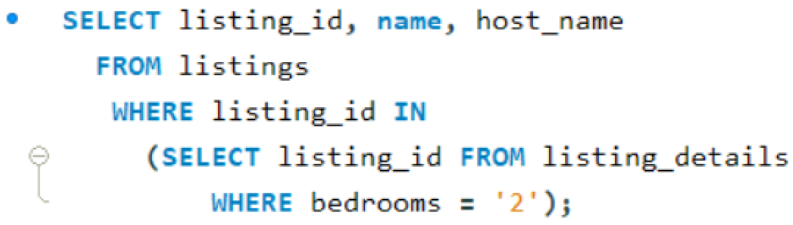




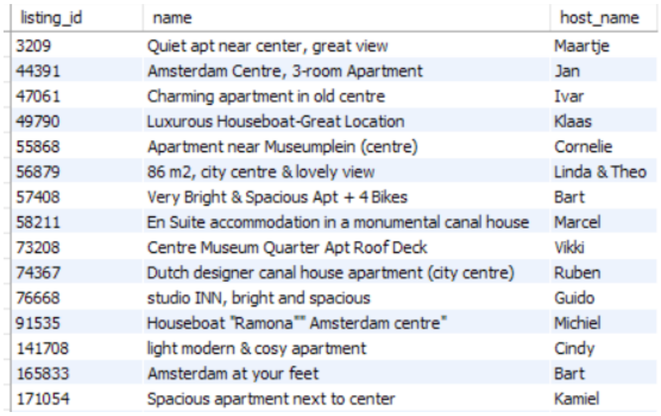
reviewer_name	reviewer_id
Alejandro	6029464
Alessandra	7332531
Alex	25064576
Alexandra	6167602
Alexandra	211004
Alma	7992090
AmBer	4592146
Amelia	8700764
Amilcar	2481566
Amit	802614
Amy	3573745
Andi	20565927
Andrew	29802439
Angela	442422
Angie	68880561
April	32648031
Araz	33284
AurÃ©lia	27822904
Bala Murali	49335486
Bekir	9256834
Ben	6164576
Bernhard	30289149
Bernhard	30289149
Billy	6790170
Brittany	69873720
Chen	50454874
Christina	2133939
Clare	72722202
Claude	7296254
Colin	422476
Cordelia	63636738
Cynthia	3737004
Daniel	38247916
Darlo	216750
David	772352
David	79479
Deepak	758617
Diana	227259
Dominic	801728
Drew	374779
Eddie	5580754
Elena	28738000
Emilie	25733503
Esther	32431176
Etienne	60449351

3. **Creating a subquery to extract data**

   a) **Get property listing_id, name and respective host name of properties that have 2 bedrooms:**



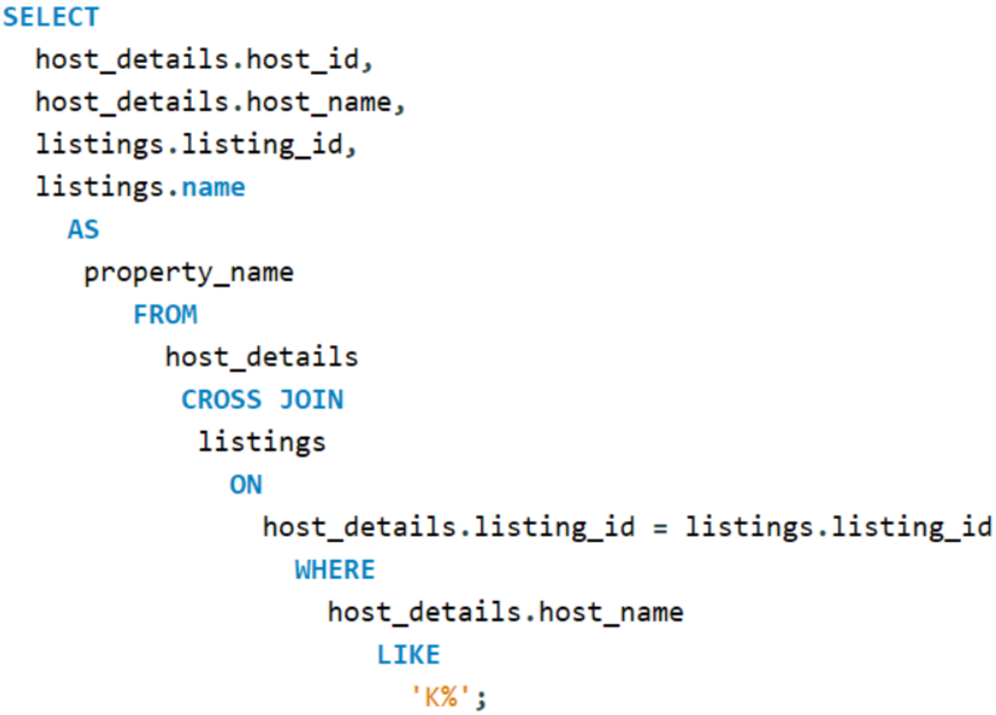
```
• SELECT listing_id, name, host_name
    FROM listings
     WHERE listing_id IN
       (SELECT listing_id FROM listing_details
           WHERE bedrooms = '2');
```



listing_id	name	host_name
3209	Quiet apt near center, great view	Maartje
44391	Amsterdam Centre, 3-room Apartment	Jan
47061	Charming apartment in old centre	Ivar
49790	Luxurous Houseboat-Great Location	Klaas
55868	Apartment near Museumplein (centre)	Cornelie
56879	86 m2, city centre & lovely view	Linda & Theo
57408	Very Bright & Spacious Apt + 4 Bikes	Bart
58211	En Suite accommodation in a monumental canal house	Marcel
73208	Centre Museum Quarter Apt Roof Deck	Vikki
74367	Dutch designer canal house apartment (city centre)	Ruben
76668	studio INN, bright and spacious	Guido
91535	Houseboat "Ramona"" Amsterdam centre"	Michiel
141708	light modern & cosy apartment	Cindy
165833	Amsterdam at your feet	Bart
171054	Spacious apartment next to center	Kamiel

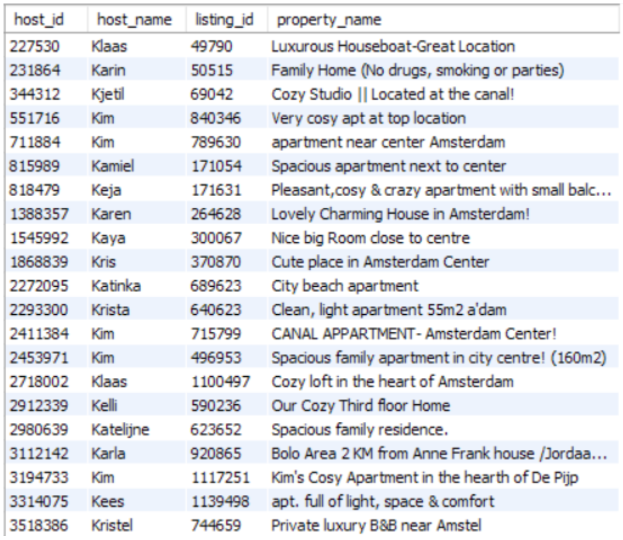
4. Creating queries using JOINS, WHERE, GROUP BY, HAVING

- To demonstrate subqueries, I extracted specific information as shown in the examples below:
  a) **Get the listing id & property names with the respective host id of the hosts whose names begin with the alphabet K :**



```
SELECT
  host_details.host_id,
  host_details.host_name,
  listings.listing_id,
  listings.name
    AS
     property_name
        FROM
          host_details
           CROSS JOIN
            listings
              ON
                host_details.listing_id = listings.listing_id
                  WHERE
                    host_details.host_name
                       LIKE
                         'K%';
```

The query returned a result of 37 hosts whose names begin with K as below:

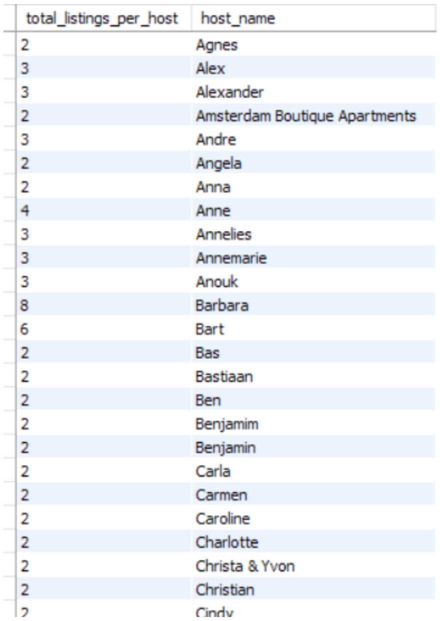


host_id	host_name	listing_id	property_name
227530	Klaas	49790	Luxurous Houseboat-Great Location
231864	Karin	50515	Family Home (No drugs, smoking or parties)
344312	Kjetil	69042	Cozy Studio || Located at the canal!
551716	Kim	840346	Very cosy apt at top location
711884	Kim	789630	apartment near center Amsterdam
815989	Kamiel	171054	Spacious apartment next to center
818479	Keja	171631	Pleasant,cosy & crazy apartment with small balc...
1388357	Karen	264628	Lovely Charming House in Amsterdam!
1545992	Kaya	300067	Nice big Room close to centre
1868839	Kris	370870	Cute place in Amsterdam Center
2272095	Katinka	689623	City beach apartment
2293300	Krista	640623	Clean, light apartment 55m2 a'dam
2411384	Kim	715799	CANAL APPARTMENT- Amsterdam Center!
2453971	Kim	496953	Spacious family apartment in city centre! (160m2)
2718002	Klaas	1100497	Cozy loft in the heart of Amsterdam
2912339	Kelli	590236	Our Cozy Third floor Home
2980639	Katelijne	623652	Spacious family residence.
3112142	Karla	920865	Bolo Area 2 KM from Anne Frank house /Jordaa...
3194733	Kim	1117251	Kim's Cosy Apartment in the hearth of De Pijp
3314075	Kees	1139498	apt. full of light, space & comfort
3518386	Kristel	744659	Private luxury B&B near Amstel

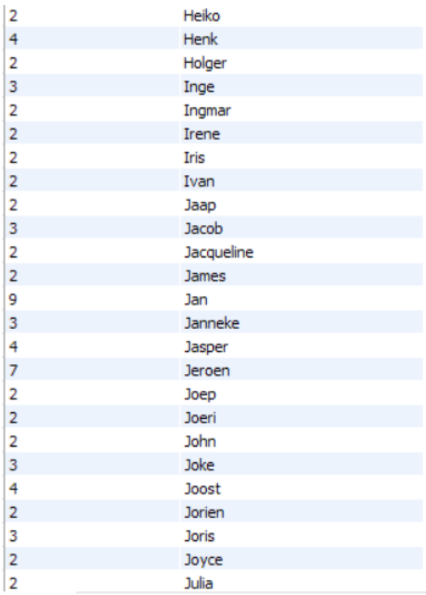
b) Get the number of listings per host of those having more than one listing in alphabetical order :

```
SELECT COUNT(listing_id) AS total_listings_per_host, host_name FROM listings
GROUP BY host_name
HAVING COUNT(listing_id) > 1
ORDER BY host_name ASC;
```

Results as shown below:



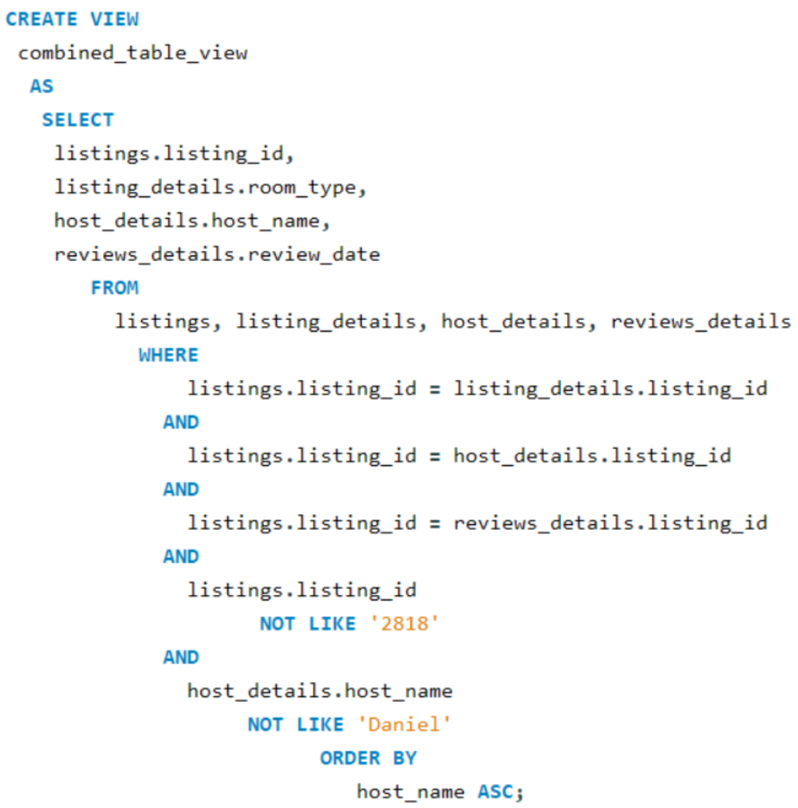
total_listings_per_host	host_name
2	Agnes
3	Alex
3	Alexander
2	Amsterdam Boutique Apartments
3	Andre
2	Angela
2	Anna
4	Anne
3	Annelies
3	Annemarie
3	Anouk
8	Barbara
6	Bart
2	Bas
2	Bastiaan
2	Ben
2	Benjamim
2	Benjamin
2	Carla
2	Carmen
2	Caroline
2	Charlotte
2	Christa & Yvon
2	Christian
2	Cindy



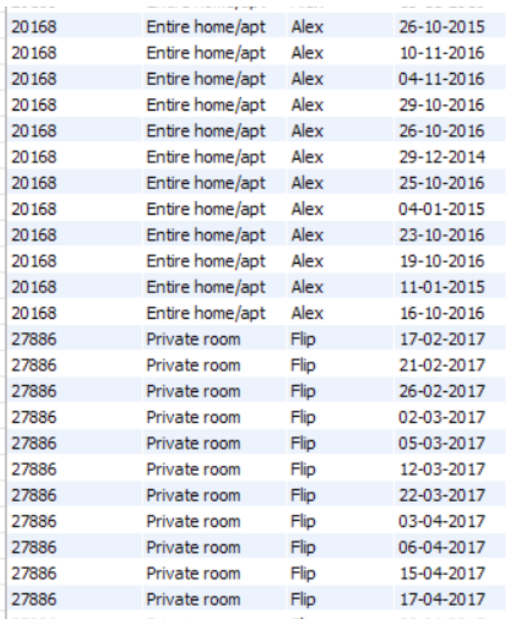
2	Heiko
4	Henk
2	Holger
3	Inge
2	Ingmar
2	Irene
2	Iris
2	Ivan
2	Jaap
3	Jacob
2	Jacqueline
2	James
9	Jan
3	Janneke
4	Jasper
7	Jeroen
2	Joep
2	Joeri
2	John
3	Joke
4	Joost
2	Jorien
3	Joris
2	Joyce
2	Julia

5. **Creating a view with all 4 base tables**

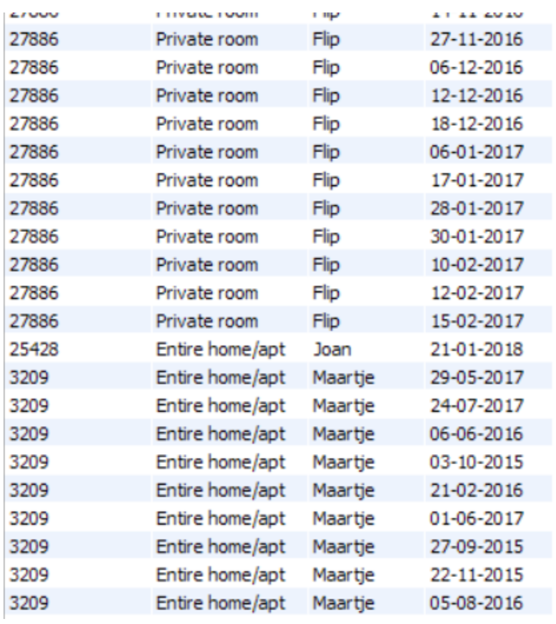
- I created a view of the airbnb listings with details of the listing_id, room type, host name and review dates which span across all four base tables as follows:



```
CREATE VIEW
 combined_table_view
  AS
   SELECT
    listings.listing_id,
    listing_details.room_type,
    host_details.host_name,
    reviews_details.review_date
       FROM
         listings, listing_details, host_details, reviews_details
           WHERE
               listings.listing_id = listing_details.listing_id
             AND
               listings.listing_id = host_details.listing_id
             AND
               listings.listing_id = reviews_details.listing_id
             AND
               listings.listing_id
                     NOT LIKE '2818'
             AND
               host_details.host_name
                    NOT LIKE 'Daniel'
                          ORDER BY
                             host_name ASC;
```

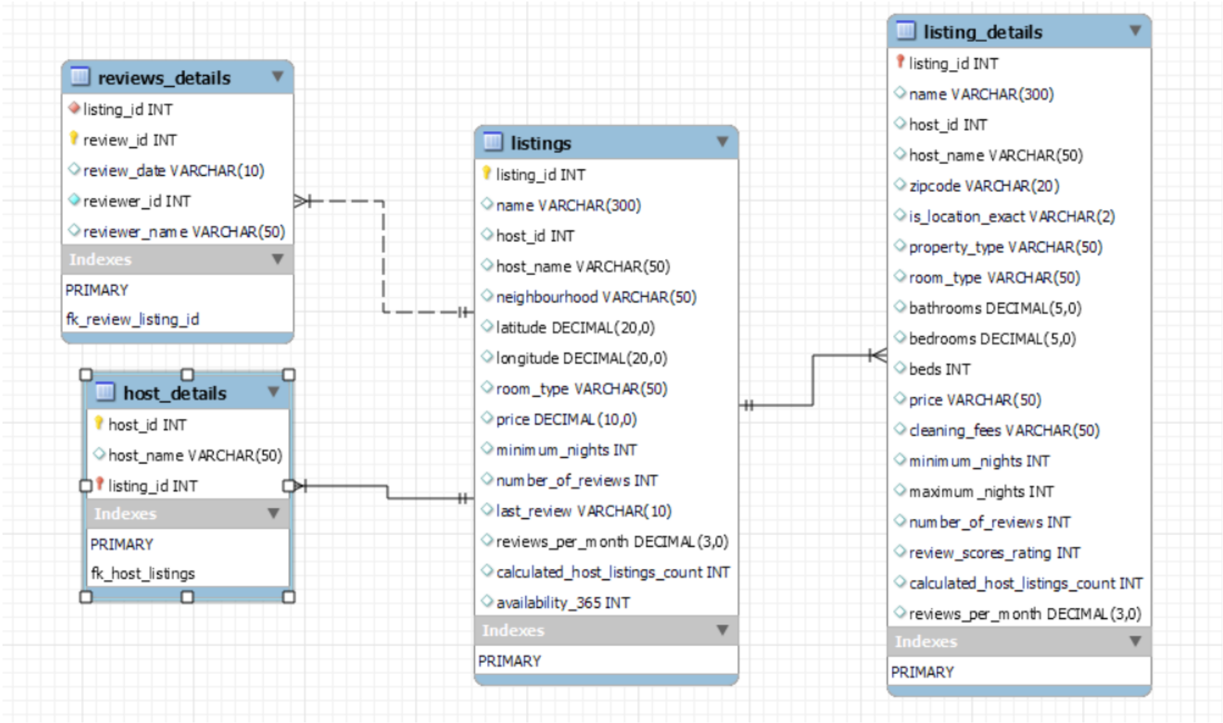


20168	Entire home/apt	Alex	26-10-2015
20168	Entire home/apt	Alex	10-11-2016
20168	Entire home/apt	Alex	04-11-2016
20168	Entire home/apt	Alex	29-10-2016
20168	Entire home/apt	Alex	26-10-2016
20168	Entire home/apt	Alex	29-12-2014
20168	Entire home/apt	Alex	25-10-2016
20168	Entire home/apt	Alex	04-01-2015
20168	Entire home/apt	Alex	23-10-2016
20168	Entire home/apt	Alex	19-10-2016
20168	Entire home/apt	Alex	11-01-2015
20168	Entire home/apt	Alex	16-10-2016
27886	Private room	Flip	17-02-2017
27886	Private room	Flip	21-02-2017
27886	Private room	Flip	26-02-2017
27886	Private room	Flip	02-03-2017
27886	Private room	Flip	05-03-2017
27886	Private room	Flip	12-03-2017
27886	Private room	Flip	22-03-2017
27886	Private room	Flip	03-04-2017
27886	Private room	Flip	06-04-2017
27886	Private room	Flip	15-04-2017
27886	Private room	Flip	17-04-2017



27886	Private room	Flip	27-11-2016
27886	Private room	Flip	06-12-2016
27886	Private room	Flip	12-12-2016
27886	Private room	Flip	18-12-2016
27886	Private room	Flip	06-01-2017
27886	Private room	Flip	17-01-2017
27886	Private room	Flip	28-01-2017
27886	Private room	Flip	30-01-2017
27886	Private room	Flip	10-02-2017
27886	Private room	Flip	12-02-2017
27886	Private room	Flip	15-02-2017
25428	Entire home/apt	Joan	21-01-2018
3209	Entire home/apt	Maartje	29-05-2017
3209	Entire home/apt	Maartje	24-07-2017
3209	Entire home/apt	Maartje	06-06-2016
3209	Entire home/apt	Maartje	03-10-2015
3209	Entire home/apt	Maartje	21-02-2016
3209	Entire home/apt	Maartje	01-06-2017
3209	Entire home/apt	Maartje	27-09-2015
3209	Entire home/apt	Maartje	22-11-2015
3209	Entire home/apt	Maartje	05-08-2016

6. Creating a DB diagram using reverse engineer



reviews_details

listing_id INT
review_id INT
review_date VARCHAR(10)
reviewer_id INT
reviewer_name VARCHAR(50)

Indexes

PRIMARY
fk_review_listing_id

host_details

host_id INT
host_name VARCHAR(50)
listing_id INT

Indexes

PRIMARY
fk_host_listings

listings

listing_id INT
name VARCHAR(300)
host_id INT
host_name VARCHAR(50)
neighbourhood VARCHAR(50)
latitude DECIMAL(20,0)
longitude DECIMAL(20,0)
room_type VARCHAR(50)
price DECIMAL(10,0)
minimum_nights INT
number_of_reviews INT
last_review VARCHAR(10)
reviews_per_month DECIMAL(3,0)
calculated_host_listings_count INT
availability_365 INT

Indexes

PRIMARY

listing_details

listing_id INT
name VARCHAR(300)
host_id INT
host_name VARCHAR(50)
zipcode VARCHAR(20)
is_location_exact VARCHAR(2)
property_type VARCHAR(50)
room_type VARCHAR(50)
bathrooms DECIMAL(5,0)
bedrooms DECIMAL(5,0)
beds INT
price VARCHAR(50)
cleaning_fees VARCHAR(50)
minimum_nights INT
maximum_nights INT
number_of_reviews INT
review_scores_rating INT
calculated_host_listings_count INT
reviews_per_month DECIMAL(3,0)

Indexes

PRIMARY