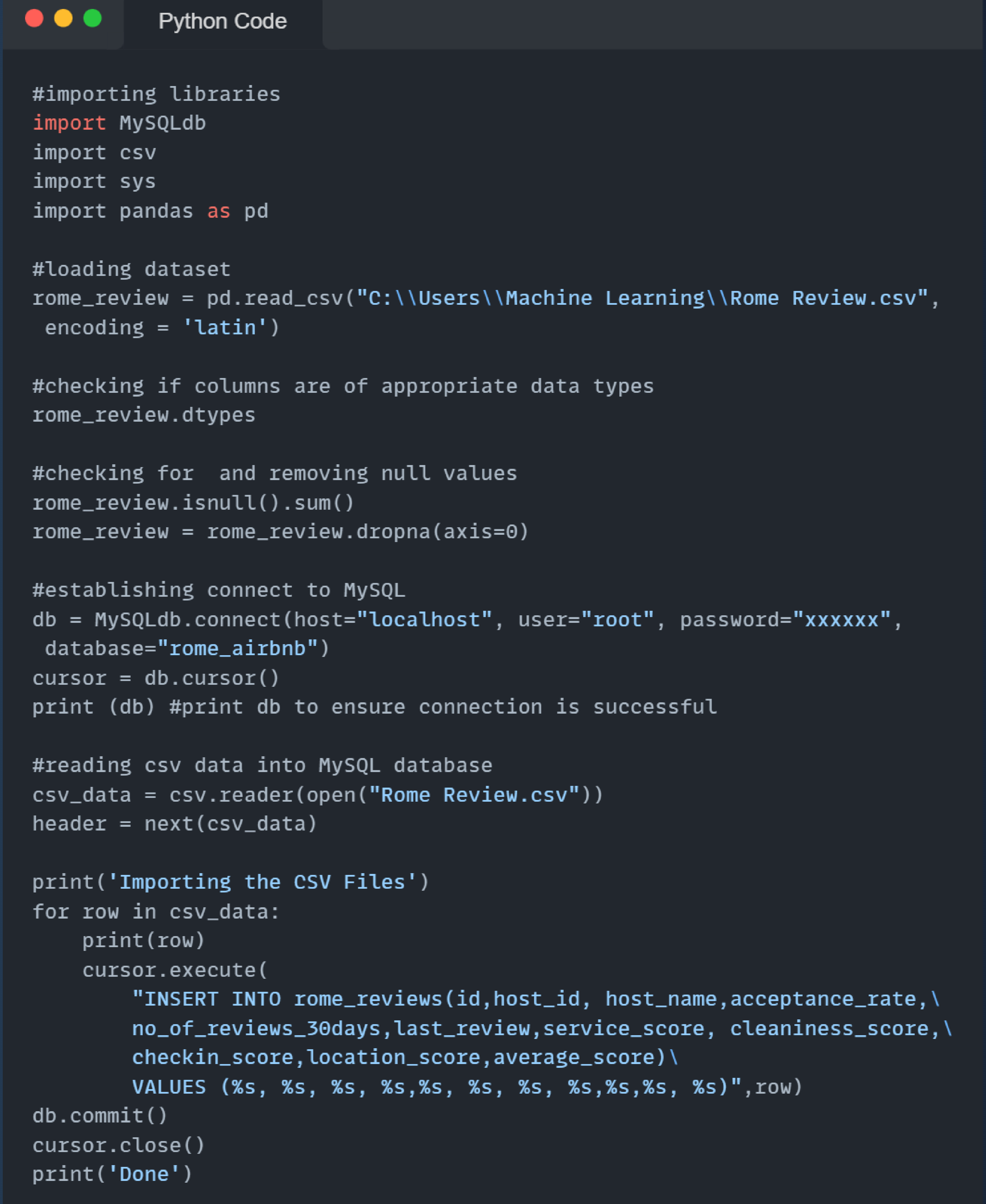
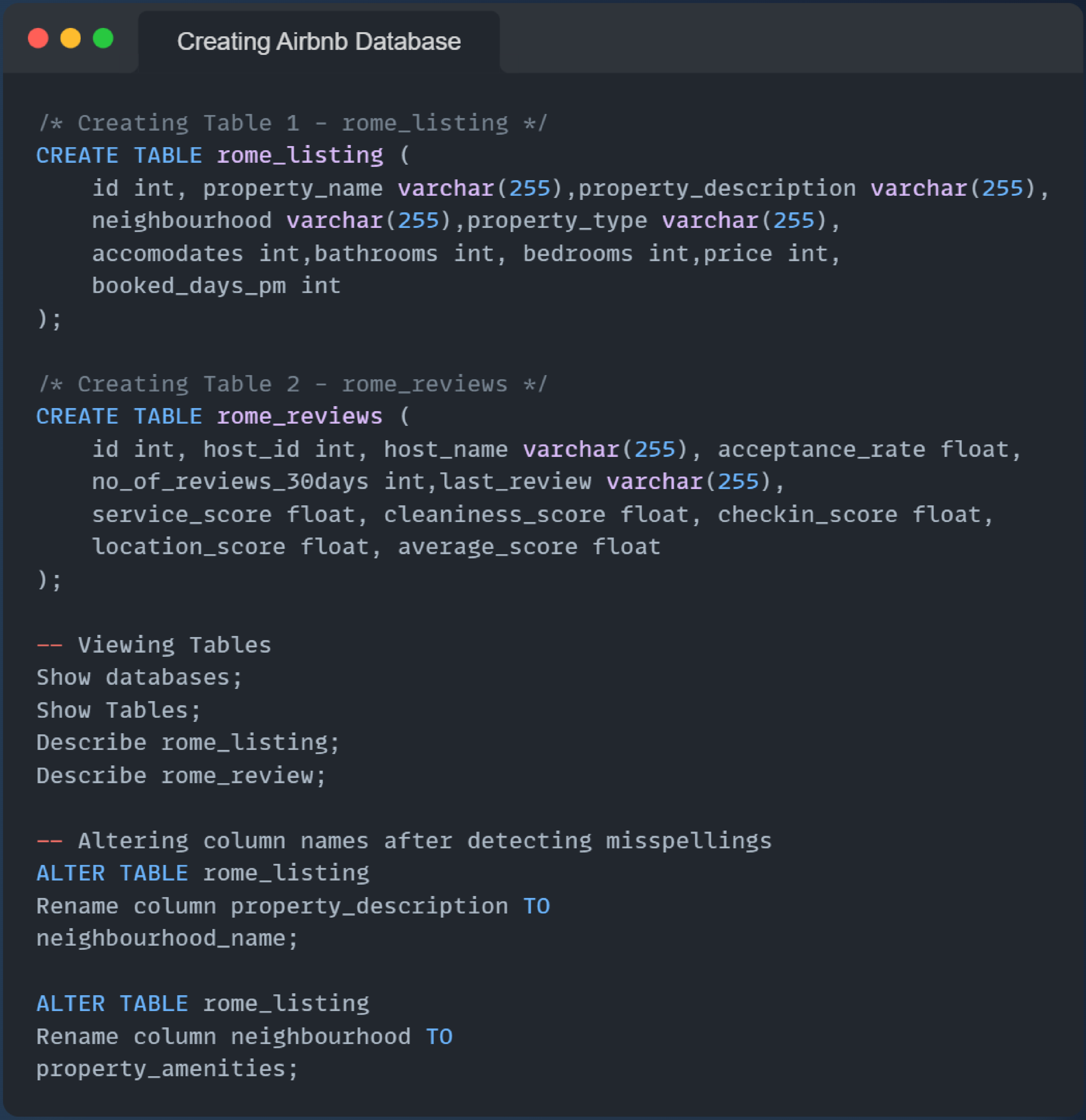
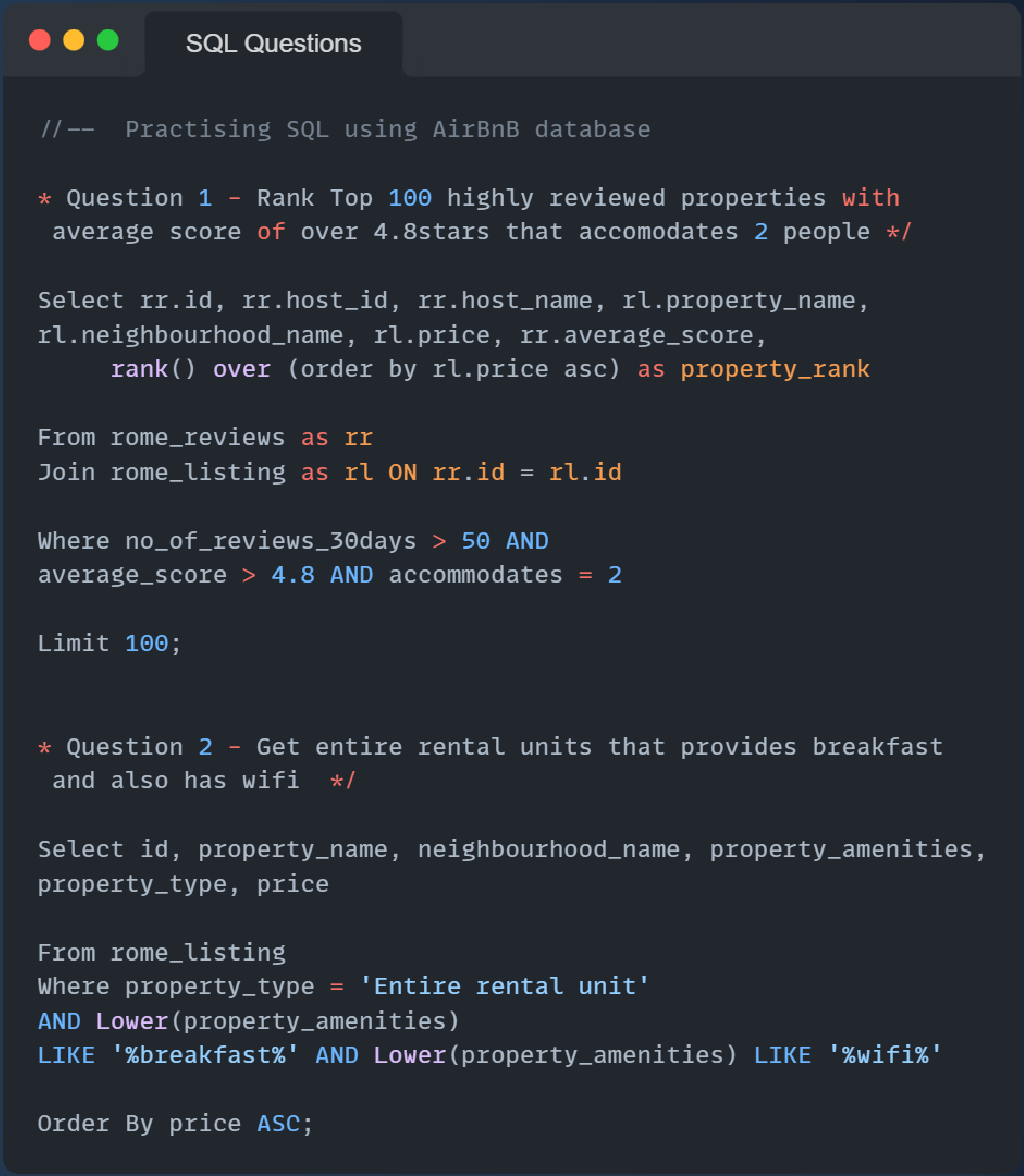
airbnb
PYTHON MYSQL DATABASE
MySQL



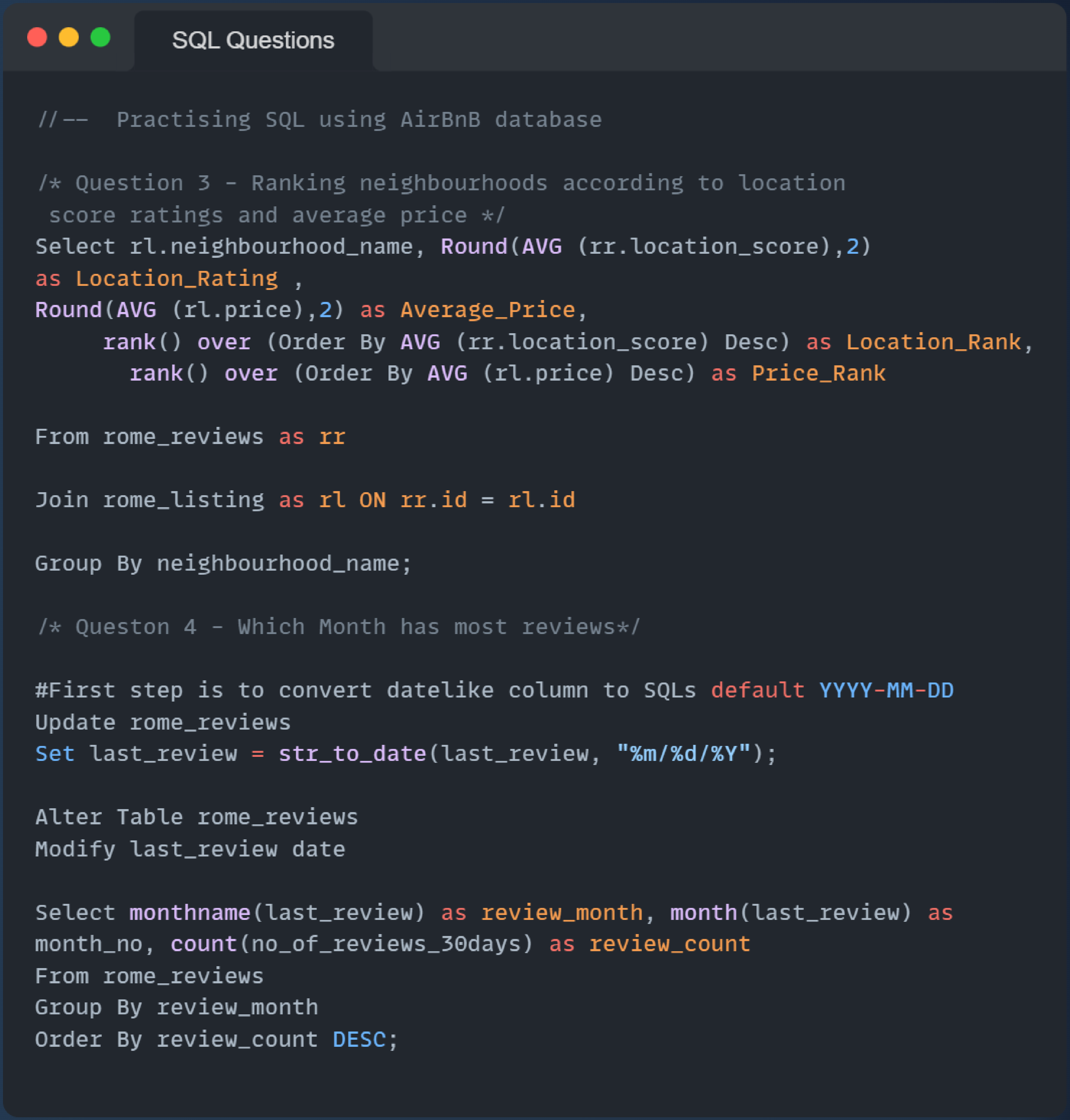
```
#importing libraries
import MySQLdb
import csv
import sys
import pandas as pd

#loading dataset
rome_review = pd.read_csv("C:\\Users\\Machine Learning\\Rome Review.csv",
 encoding = 'latin')

#checking if columns are of appropriate data types
rome_review.dtypes

#checking for  and removing null values
rome_review.isnull().sum()
rome_review = rome_review.dropna(axis=0)

#establishing connect to MySQL
db = MySQLdb.connect(host="localhost", user="root", password="xxxxxx",
 database="rome_airbnb")
cursor = db.cursor()
print (db) #print db to ensure connection is successful

#reading csv data into MySQL database
csv_data = csv.reader(open("Rome Review.csv"))
header = next(csv_data)

print('Importing the CSV Files')
for row in csv_data:
    print(row)
    cursor.execute(
        "INSERT INTO rome_reviews(id,host_id, host_name,acceptance_rate,\
        no_of_reviews_30days,last_review,service_score, cleaniness_score,\
        checkin_score,location_score,average_score)\
        VALUES (%s, %s, %s, %s,%s, %s, %s, %s,%s,%s, %s)",row)
db.commit()
cursor.close()
print('Done')
```



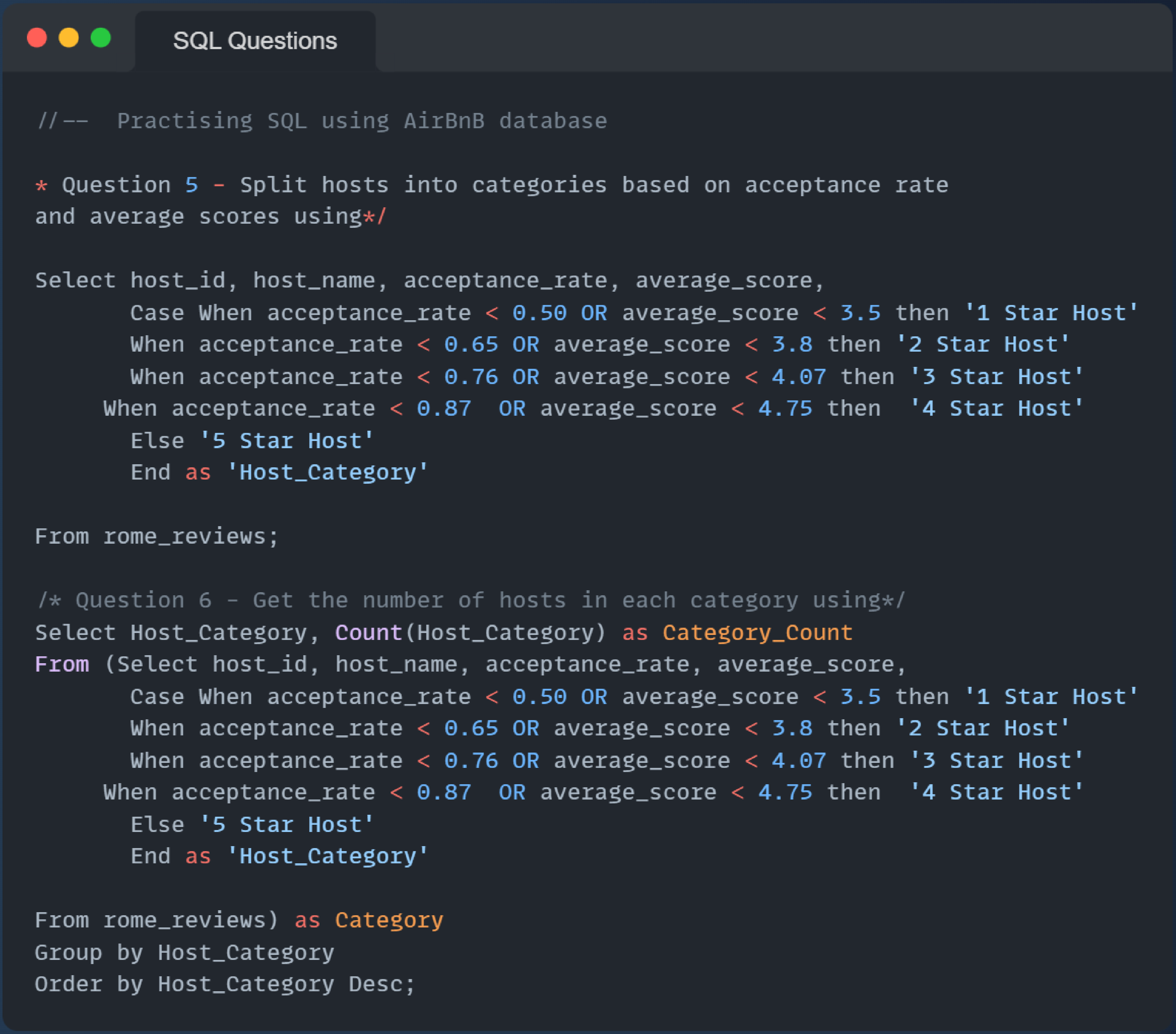
```
/* Creating Table 1 - rome_listing */
CREATE TABLE rome_listing (
    id int, property_name varchar(255),property_description varchar(255),
    neighbourhood varchar(255),property_type varchar(255),
    accomodates int,bathrooms int, bedrooms int,price int,
    booked_days_pm int
);

/* Creating Table 2 - rome_reviews */
CREATE TABLE rome_reviews (
    id int, host_id int, host_name varchar(255), acceptance_rate float,
    no_of_reviews_30days int,last_review varchar(255),
    service_score float, cleaniness_score float, checkin_score float,
    location_score float, average_score float
);

-- Viewing Tables
Show databases;
Show Tables;
Describe rome_listing;
Describe rome_review;

-- Altering column names after detecting misspellings
ALTER TABLE rome_listing
Rename column property_description TO
neighbourhood_name;

ALTER TABLE rome_listing
Rename column neighbourhood TO
property_amenities;
```



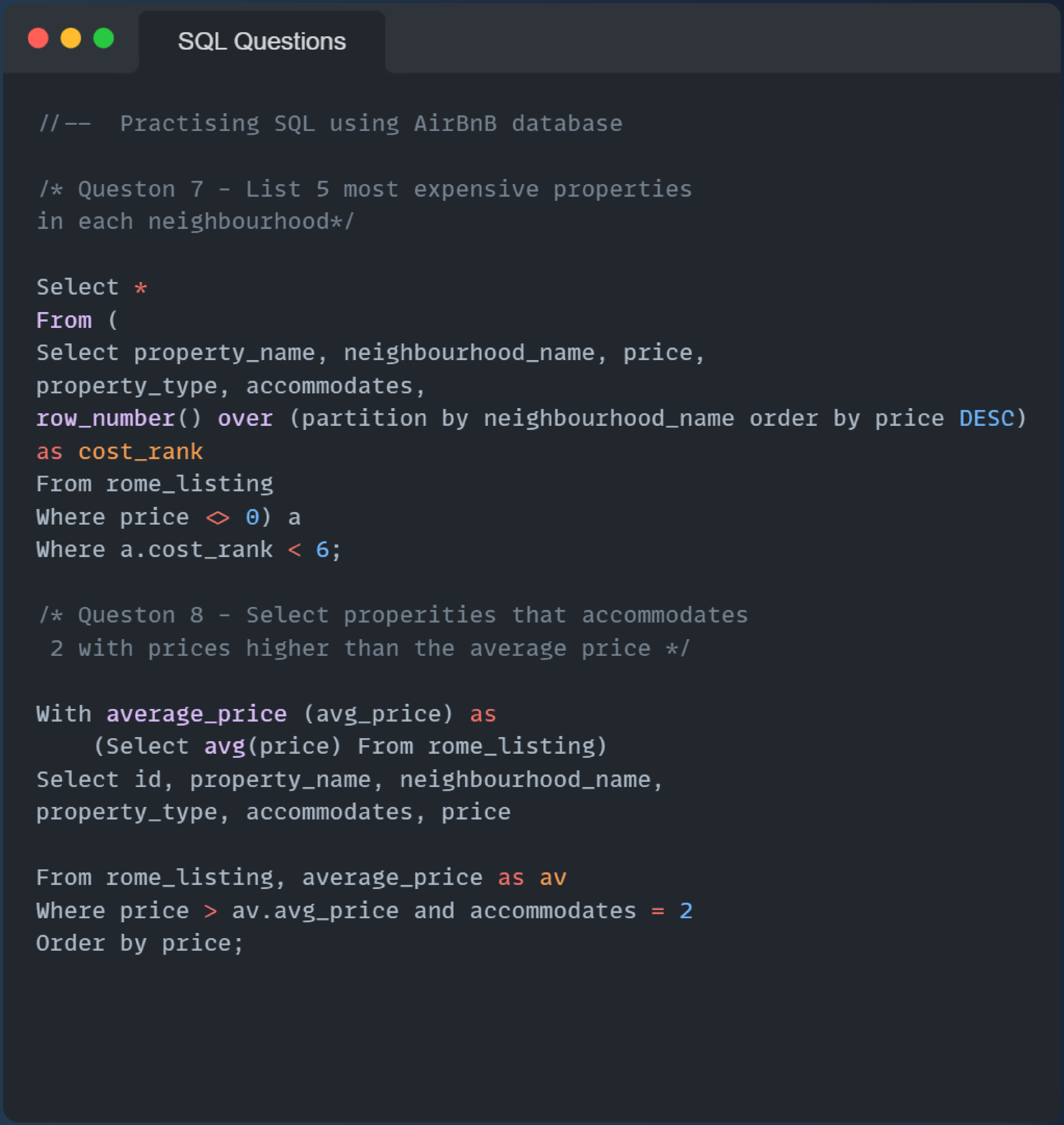
```
//--  Practising SQL using AirBnB database

* Question 1 - Rank Top 100 highly reviewed properties with
 average score of over 4.8stars that accomodates 2 people */

Select rr.id, rr.host_id, rr.host_name, rl.property_name,
rl.neighbourhood_name, rl.price, rr.average_score,
     rank() over (order by rl.price asc) as property_rank

From rome_reviews as rr
Join rome_listing as rl ON rr.id = rl.id

Where no_of_reviews_30days > 50 AND
average_score > 4.8 AND accommodates = 2

Limit 100;


* Question 2 - Get entire rental units that provides breakfast
 and also has wifi  */

Select id, property_name, neighbourhood_name, property_amenities,
property_type, price

From rome_listing
Where property_type = 'Entire rental unit'
AND Lower(property_amenities)
LIKE '%breakfast%' AND Lower(property_amenities) LIKE '%wifi%'

Order By price ASC;
```



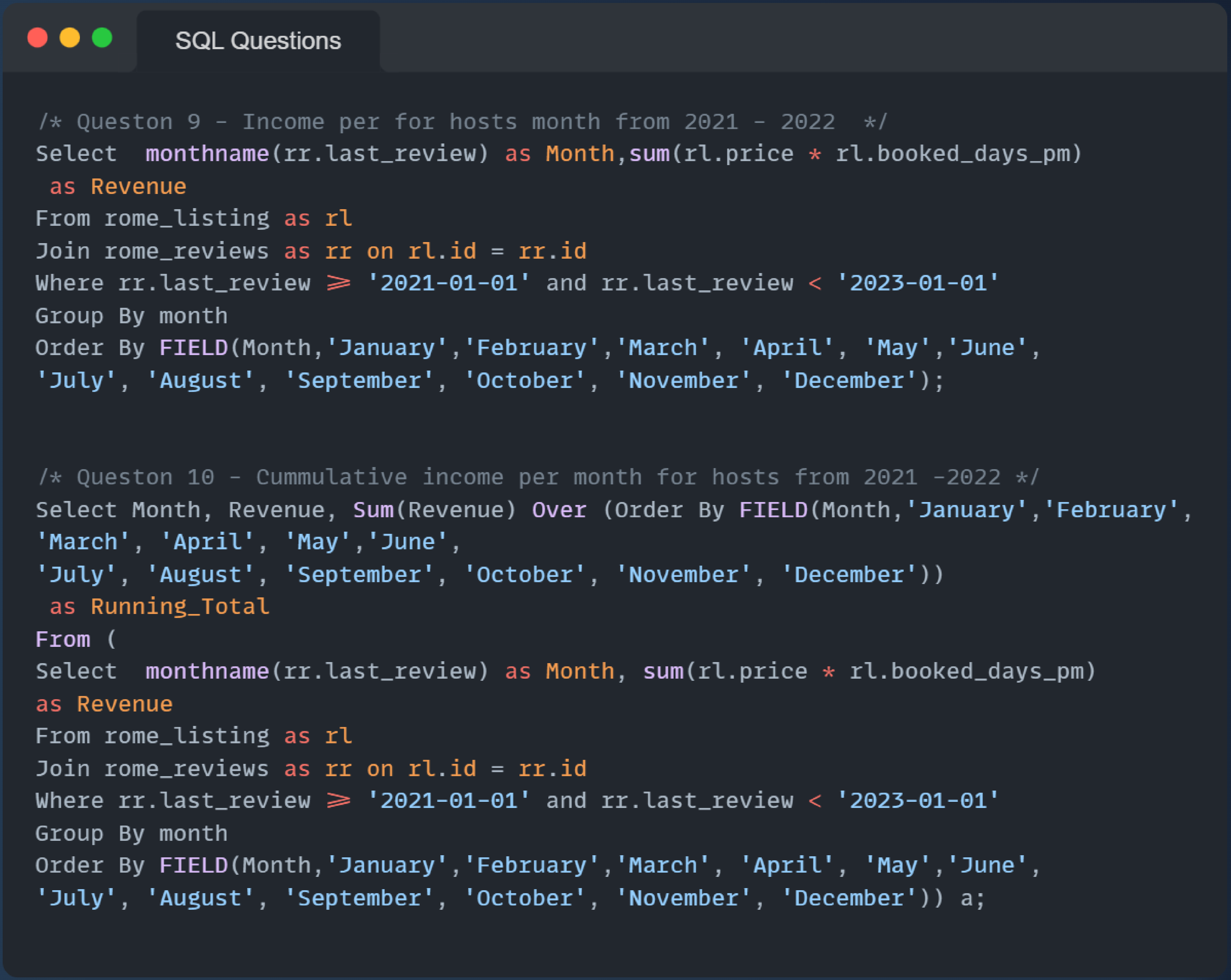
```
//--  Practising SQL using AirBnB database

/* Question 3 - Ranking neighbourhoods according to location
 score ratings and average price */
Select rl.neighbourhood_name, Round(AVG (rr.location_score),2)
as Location_Rating ,
Round(AVG (rl.price),2) as Average_Price,
     rank() over (Order By AVG (rr.location_score) Desc) as Location_Rank,
       rank() over (Order By AVG (rl.price) Desc) as Price_Rank

From rome_reviews as rr

Join rome_listing as rl ON rr.id = rl.id

Group By neighbourhood_name;

/* Queston 4 - Which Month has most reviews*/

#First step is to convert datelike column to SQLs default YYYY-MM-DD
Update rome_reviews
Set last_review = str_to_date(last_review, "%m/%d/%Y");

Alter Table rome_reviews
Modify last_review date

Select monthname(last_review) as review_month, month(last_review) as
month_no, count(no_of_reviews_30days) as review_count
From rome_reviews
Group By review_month
Order By review_count DESC;
```



```
//--  Practising SQL using AirBnB database

* Question 5 - Split hosts into categories based on acceptance rate
and average scores using*/

Select host_id, host_name, acceptance_rate, average_score,
       Case When acceptance_rate < 0.50 OR average_score < 3.5 then '1 Star Host'
       When acceptance_rate < 0.65 OR average_score < 3.8 then '2 Star Host'
       When acceptance_rate < 0.76 OR average_score < 4.07 then '3 Star Host'
     When acceptance_rate < 0.87  OR average_score < 4.75 then  '4 Star Host'
       Else '5 Star Host'
       End as 'Host_Category'

From rome_reviews;

/* Question 6 - Get the number of hosts in each category using*/
Select Host_Category, Count(Host_Category) as Category_Count
From (Select host_id, host_name, acceptance_rate, average_score,
       Case When acceptance_rate < 0.50 OR average_score < 3.5 then '1 Star Host'
       When acceptance_rate < 0.65 OR average_score < 3.8 then '2 Star Host'
       When acceptance_rate < 0.76 OR average_score < 4.07 then '3 Star Host'
     When acceptance_rate < 0.87  OR average_score < 4.75 then  '4 Star Host'
       Else '5 Star Host'
       End as 'Host_Category'

From rome_reviews) as Category
Group by Host_Category
Order by Host_Category Desc;
```



```
//--  Practising SQL using AirBnB database

/* Queston 7 - List 5 most expensive properties
in each neighbourhood*/

Select *
From (
Select property_name, neighbourhood_name, price,
property_type, accommodates,
row_number() over (partition by neighbourhood_name order by price DESC)
as cost_rank
From rome_listing
Where price <> 0) a
Where a.cost_rank < 6;

/* Queston 8 - Select properities that accommodates
 2 with prices higher than the average price */

With average_price (avg_price) as
    (Select avg(price) From rome_listing)
Select id, property_name, neighbourhood_name,
property_type, accommodates, price

From rome_listing, average_price as av
Where price > av.avg_price and accommodates = 2
Order by price;
```



SQL Questions

```sql
/* Queston 9 - Income per for hosts month from 2021 - 2022  */
Select  monthname(rr.last_review) as Month,sum(rl.price * rl.booked_days_pm)
 as Revenue
From rome_listing as rl
Join rome_reviews as rr on rl.id = rr.id
Where rr.last_review >= '2021-01-01' and rr.last_review < '2023-01-01'
Group By month
Order By FIELD(Month,'January','February','March', 'April', 'May','June',
'July', 'August', 'September', 'October', 'November', 'December');


/* Queston 10 - Cummulative income per month for hosts from 2021 -2022 */
Select Month, Revenue, Sum(Revenue) Over (Order By FIELD(Month,'January','February',
'March', 'April', 'May','June',
'July', 'August', 'September', 'October', 'November', 'December'))
 as Running_Total
From (
Select  monthname(rr.last_review) as Month, sum(rl.price * rl.booked_days_pm)
as Revenue
From rome_listing as rl
Join rome_reviews as rr on rl.id = rr.id
Where rr.last_review >= '2021-01-01' and rr.last_review < '2023-01-01'
Group By month
Order By FIELD(Month,'January','February','March', 'April', 'May','June',
'July', 'August', 'September', 'October', 'November', 'December')) a;
```