DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Databases Project – Spring 2019

Team No: 13

Names: Wan-Tzu Huang, Mingbo Cui, Futong Liu

Contents

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 1

Assumptions

1. If a host deletes his account, his associated properties will also be deleted from the listing.
2. If a reviewer deletes his account, his associated comments will still stay in the database.
3. If a property on the listing is deleted, the associated comments of it will also be deleted.
4. A host might not have a listing at all times.
5. Location, room_type, bed_type, property_type are all categorical data, and they are all filled up, i.e. they cannot be left blank.
6. One reviewer id can only correspond to one reviewer name. Namely, a reviewer cannot change his name once he obtains his reviewer id.
7. Houses in each original csv file are indeed of the indicated city. For example, houses in madrid_listings.csv are all located in Madrid.

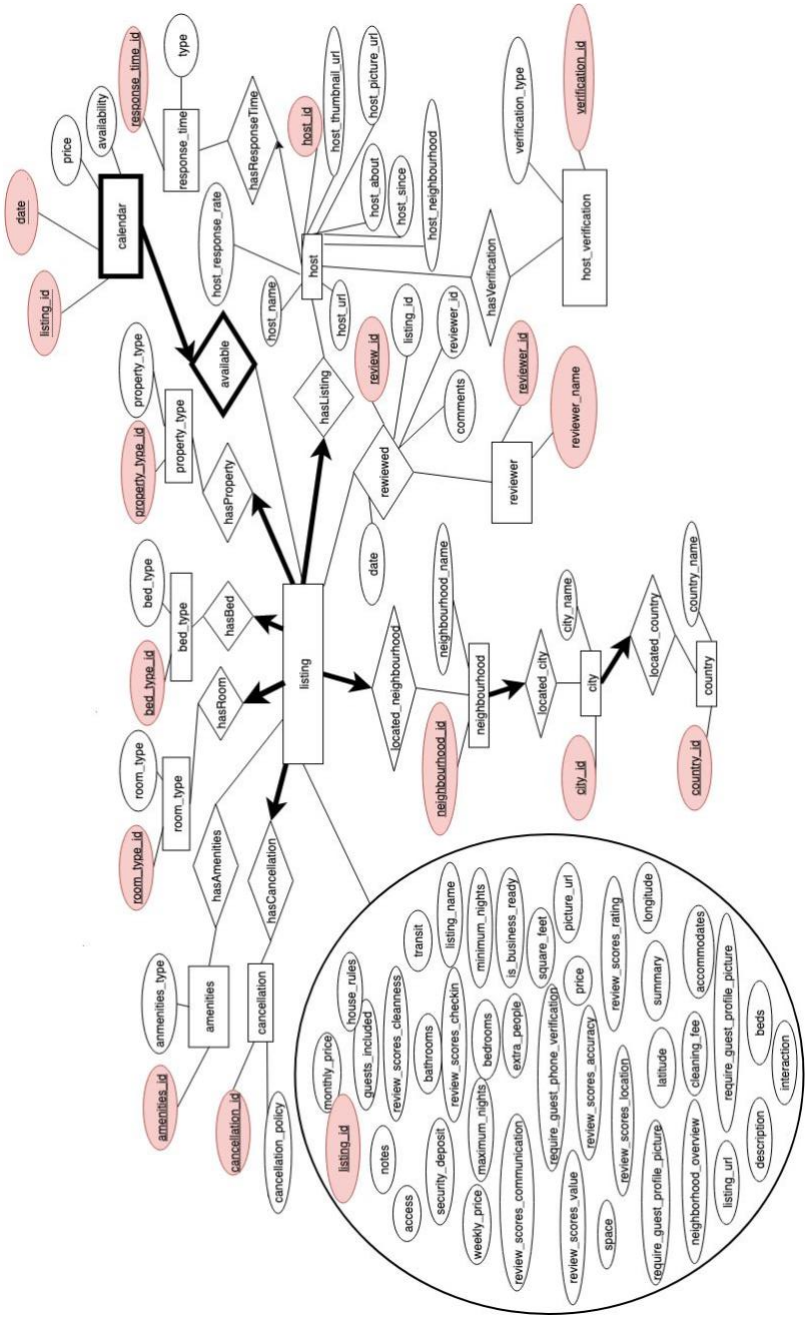
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Entity Relationship Schema

Schema

<Add the figure of the ER schema>



DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Description

Entities (EntityName: primary key**)**:

1. host: host_id
2. reponse_time: response_time_id
   Response time is in fact a categorical attribute. So it can be normalized into a small entity to compress.
3. host_verification: verification_id
   Host verification is in fact a categorical attribute. So it can be normalized into a small entity.
4. listing: listing_id
5. neighbourhood: neighbourhood_id
   Neighbourhood is in fact a categorical attribute. So it can be normalized into a small entity.
6. city: city_id
   City is in fact a categorical attribute. So it can be normalized into a small entity.
7. country: country_id
   Country is in fact a categorical attribute. So it can be normalized into a small entity.
8. cancellation: cancellation_id
   Cancellation policy is in fact a categorical attribute. So it can be normalized into a small entity.
9. amenities: amenities_id
   Amenities is in fact a categorical attribute. So it can be normalized into a small entity.
10. room_type: room_type_id
    Room type is in fact a categorical attribute. So it can be normalized into a small entity.
11. bed_type: bed_type_id
    Bed type is in fact a categorical attribute. So it can be normalized into a small entity.
12. property_type: property_type_id
    Property type is in fact a categorical attribute. So it can be normalized into a small entity.
13. calendar: (calendar_date, listing_id)

Relationships:

1. hasListing
   Each house in table listing can only have one host and every house has a host to manage it.
2. hasReponseTime
3. hasVerification
4. reviewed
5. located_neighbourhood
   Each house in table listing is located in one neighbourhood.
6. located_city
   Each neighbourhood in table neighbourhood is located in one city.
7. located_country
   Each city in table city is located in one country
8. hasCancellation
   Each house in table listing has one cancellation policy.
9. hasAmenities

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

10. hasRoom
    Each house in table listing has one room type.
11. hasBed
    Each house in table listing has one bed type.
12. hasProperty
    Each house in table listing has one property type.
13. Available
    Each record in table calendar belongs to one listing.

Relational Schema

ER schema to Relational schema

Intuitively, each entity FROM the above ER model graph is created as a table with the same attributes. However, some entity and relationship could be combined into one single table because of the constraint they have, which can hence save space and improve performance. Therefore, we finally come to 18 tables shown as below.

1. host: host_id
2. listing: listing_id
   Relationship hasListing is merged into this table because of the total participation constraint and the one-to-many constraint between listing and host. Similarly, relationship located_neighbourhood, hasCancellation, hasAmenities, hasRoom, hasBed, hasProperty are also merged into this table.
3. reviewer: reviewer_id
4. reviewed: review_id
5. neighbourhood: neighbourhood_id
6. city: city_id
7. country: country_id
8. amenities: amenities_id
9. hasAmenities: (listing_id, amenities_id)
10. cancellation: cancellation_id
11. calendarAvailable: (calendar_date, listing_id)
    Entity calendar and relationship available are combined together because of the total participation constraint and the one-to-many constraint between them. The primary key of this new table is the combination of calendar_date and listing_id, which is the same as entity calendar, with listing_id as the foreign key referencing table listing.
12. response_time: response_time_id
13. hasResponseTime: (response_time_id, host_id)
14. host_verification: verification_id
15. has_verification: (verification_id, host_id)
16. property_type: property_type_id
17. bed_type: bed_type_id

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

18. room_type: room_type_id

DDL

```
CREATE TABLE reviewed(
 review_id INTEGER,
 reviewer_id INTEGER,
 listing_id INTEGER,
 review_date DATE,
 comments CLOB,
 PRIMARY KEY (review_id),
 FOREIGN KEY (listing_id) REFERENCES Listing,
 FOREIGN KEY (reviewer_id) REFERENCES reviewer
 );

CREATE TABLE reviewer(
 reviewer_id INTEGER,
 reviewer_name CHAR(80),
 PRIMARY KEY(reviewer_id, reviewer_name)
);

CREATE TABLE host(
      host_id INTEGER,
      host_name NCHAR(70),
      host_since DATE,
      host_about CLOB,
      host_picture_url VARCHAR2(4000),
      host_thumbnail_url VARCHAR2(4000),
      host_neighborhood VARCHAR2(4000),
      host_url VARCHAR2(4000),
      host_response_rate INTEGER,
      PRIMARY KEY (host_id)
);

CREATE TABLE response_time(
    response_time_id INTEGER,
    type CHAR(20),
    PRIMARY KEY (response_time_id)
);

CREATE TABLE hasResponseTime(
      response_time_id INTEGER,
      host_id INTEGER,
      PRIMARY KEY (response_time_id, host_id),
      FOREIGN KEY (host_id) REFERENCES host,
      FOREIGN KEY (response_time_id) REFERENCES response_time
);

CREATE TABLE host_verification(
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

```
      verification_id INTEGER,
      verification_type CHAR(30),
      PRIMARY KEY (verification_id)
);

CREATE TABLE has_verification(
      verification_id INTEGER,
      host_id INTEGER,
      PRIMARY KEY (verification_id, host_id),
      FOREIGN KEY (host_id) REFERENCES host,
      FOREIGN KEY (verification_id) REFERENCES host_verification
);

CREATE TABLE property_type(
    property_type_id INTEGER,
    type CHAR(40),
    PRIMARY KEY (property_type_id)
);

CREATE TABLE bed_type(
    bed_type_id INTEGER,
    type CHAR(20),
    PRIMARY KEY (bed_type_id)
);

CREATE TABLE room_type(
    room_type_id INTEGER,
    type CHAR(20),
    PRIMARY KEY (room_type_id)
);

CREATE TABLE listing (
    listing_id INTEGER,
    host_id INTEGER NOT NULL,
    listing_name CHAR(150),
    amenities INTEGER,
    cancellation_policy_id INTEGER NOT NULL,
    room_type_id INTEGER NOT NULL,
    bed_type_id INTEGER NOT NULL,
    property_type_id INTEGER NOT NULL,
    neighbourhood_id NOT NULL,
    review_scores_rating INTEGER,
    cleaning_fee FLOAT,
    minimum_nights  INTEGER,
    square_feet FLOAT,
    review_scores_cleanliness INTEGER,
    review_scores_accuracy INTEGER,
    review_scores_value INTEGER,
    review_scores_location INTEGER,
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
    require_guest_phone_verification CHAR(1),
    price FLOAT,
    accommodates INTEGER,
    security_deposit FLOAT,
    review_scores_checkin INTEGER,
    review_scores_communication INTEGER,
    maximum_nights INTEGER,
    guests_included INTEGER,
    monthly_price FLOAT,
    bathrooms INTEGER,
    bedrooms INTEGER,
    weekly_price FLOAT,
    extra_people FLOAT,
    is_business_travel_ready CHAR(1),
    longitude FLOAT,
    latitude FLOAT,
    house_rules VARCHAR2(4000),
    listing_summary VARCHAR2(4000),
    transit VARCHAR2(4000),
    space VARCHAR2(4000),
    access VARCHAR2(4000),
    neighborhood_overview VARCHAR2(4000),
    picture_url VARCHAR2(4000),
    notes VARCHAR2(4000),
    listing_url VARCHAR2(4000),
    description VARCHAR2(4000),
    interaction VARCHAR2(4000),
    beds INTEGER,
    require_guest_profile_picture CHAR(1),

    PRIMARY KEY (listing_id),
    FOREIGN KEY (cancellation_policy_id) REFERENCES cancellation,
    FOREIGN KEY (room_type_id) REFERENCES room_type,
    FOREIGN KEY (bed_type_id) REFERENCES bed_type,
    FOREIGN KEY (property_type_id) REFERENCES property_type,
    FOREIGN KEY (host_id) REFERENCES host,
    FOREIGN KEY (neighbourhood_id) REFERENCES neighbourhood
);

CREATE TABLE neighbourhood(
    neighbourhood_id INTEGER,
    city_id INTEGER NOT NULL,
    neighbourhood_name CHAR(40),
    PRIMARY KEY (neighbourhood_id),
    FOREIGN KEY (city_id) REFERENCES city
);
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

```
CREATE TABLE city(
    city_id INTEGER,
    country_id INTEGER NOT NULL,
    city_name CHAR(30),
    PRIMARY KEY (city_id)
    FOREIGN KEY (country_id) REFERENCES country,
);

CREATE TABLE country(
    country_id INTEGER,
    country_name CHAR(20),
    PRIMARY KEY (country_id)
);

CREATE TABLE hasAmenities(
      amenities_id INTEGER,
      listing_id INTEGER,
      PRIMARY KEY (listing_id, amenities_id),
      FOREIGN KEY (amenities_id) REFERENCES amenities,
      FOREIGN KEY (listing_id) REFERENCES listing,
);

CREATE TABLE amenities(
    amenities_id INTEGER,
    amenities_type CHAR(90),
    PRIMARY KEY (amenities_id)
);

CREATE TABLE cancellation(
    cancellation_id INTEGER,
    cancellation_policy CHAR(50),
    PRIMARY KEY (cancellation_id)
);

CREATE TABLE calendarAvailable(
    listing_id INTEGER,
    calendar_date DATE,
    price FLOAT,
    availability CHAR(1),
    PRIMARY KEY (calendar_date, listing_id)
    FOREIGN KEY (listing_id) REFERENCES listing,
);
```

General Comments

ER model was designed all three of us. ER model is drawn by Futong, DDL generated by Wan-Tzu, report written by Mingbo.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 2

Regarding Deliverable 1, we updated the assumption part according to the TA's review. And we also made some modification to the ER-model and the DDL code above.

Assumptions

1. If a host deletes his account, his associated properties will also be deleted from the listing.
2. If a reviewer deletes his account, his associated comments will still stay in the database.
3. If a property on the listing is deleted, the associated comments of it will also be deleted.
4. A host might not have a listing at all times.
5. Location, room_type, bed_type, property_type are all categorical data, and they are all filled up, i.e. they cannot be left blank.
6. One reviewer id can only correspond to one reviewer name. Namely, a reviewer cannot change his name once he obtains his reviewer id.
7. Houses in each original csv file are indeed of the indicated city. For example, houses in madrid_listings.csv are all located in Madrid.

Data Loading

To load the data, we just use oracle developer to load csv file. Before that, we have done a lot of data cleaning work:

1: We refreshed the "city" and "country" attributes with the name of different csv file, for example, the "city" attribute and "country" attribute of certain listing contained in "**barcelona_listings.csv**" will be updated to "Barcelona" and "Spain" respectively.

2: Some digits in the listing is expressed with comma, to load the data we have removed the comma and normalize the type from String to Integer and Float accordingly.

3: In the REVIEWER table, we have two attributes reviewer_id and reviewer_name and the primary key is reviewer_id, but we found there are two reviewers who share with same reviewer_id but with different reviewer_name. We take the recommended suggestion on MOODLE and delete one of them.

4: To save the memory and improve our query performance, we have done a lot of normalization work:

1) the "amenities" attribute in the original file is a dirty list which contains all accessible facilities, we seperate the list to many single elements;
2) Location: we have normalized "neighbourhood", "city" and "country" attributes with three other tables. After normalization, LISTING only contains neighbourhood id.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

3) Attributes of Categorial Type: We have normalized bed_type, room_type, property_type, cancellation_policy...almost all attributes with categorical type have been normalized to single digit

Query Implementation

We still have not fully finished yet the loading procedure of the relations so we did not attach the result of each query to the report.

Query 1:

What is the average price for a listing with 8 bedrooms?

Description of logic:

Filter to obtain all the listings that have 8 bedrooms and then get the average price of them.

SQL statement:

```
SELECT AVG(price)
FROM listing
WHERE listing.bedrooms = 8;
```

	AVG(PRICE)
1	313.153846153846153846153846153846153846

Results for Query 1

Query 2:

What is the average cleaning review score for listings with TV?

Description of logic:

Average value of review_scores_cleanness attribute is calculated by only consider the listing which has TV in their amenities

SQL statement

```
SELECT AVG(l.REVIEW_SCORES_CLEANLINESS)
FROM listing l, HASAMENITIES ha
WHERE l.listing_id = ha.listing_id AND
      ha.AMENITIES_ID = (SELECT am.AMENITIES_ID FROM AMENITIES am
                         WHERE am.AMENITIES_TYPE = 'TV');
```

	AVG(L.REVIEW_SCORES_CLEANLINESS)
1	9.398500936914428482198625858838226108 68

Results for Query 2

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 3:

Print all the hosts who have an available property between date 03.2019 and 09.2019.

Description of logic:

By joining table listing and calendar we could filter to get the required host ids. By looking up table host, we can print out the host ids and host names.

SQL statement

```
SELECT H.host_id, H.host_name
FROM host H
WHERE H.host_id IN (SELECT L.host_id
                    FROM listing L, calendaravailable C
                    WHERE L.listing_id = C.listing_id AND
                          C.availability = 't' AND
                          C.CALENDAR_DATE >= '01-MAR-19' AND
                          C.CALENDAR_DATE <= '30-SEP-19');
```

	HOST_ID	HOST_NAME
1	612665	Oscar
2	1541784	Aga
3	1570247	Irene
4	2741652	Rodrigo
5	2745490	Bori
6	2201772	Dimitry
7	2747795	Stephanie
8	1391607	Aline
9	3920790	Carla
10	3922188	Marta

Results for Query 3

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 4:

Print how many listing items exist that are posted by two different hosts but the hosts have the same name.

Description of logic:

COUNT is used to determine how many different listing with different host_id but the host_id has same host_name

SQL statement

```
SELECT count(*)
FROM listing l
WHERE l.host_id in (select h1.host_id
                    from host h1, host h2
                    where h1.host_name = h2.host_name and h1.host_id != h2.host_id);
```

	COUNT(*)
1	30393

Results for Query 4

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 5:

Description of logic:

Make a join between listing and calendar. Then get all the available listings and all the listings under that name of Viajes Eco. Then project the dates.

SQL statement

```
SELECT c.CALENDAR_DATE
FROM listing, calendaravailable c
WHERE listing.listing_id = c.listing_id AND
      c.availability = 't' AND
      listing.host_id = (SELECT host_id
                         FROM host
                         WHERE host.host_name = 'Viajes Eco');
```

	CALENDAR_DATE
1	03.03.19
2	02.03.19
3	01.03.19
4	28.02.19
5	27.02.19
6	26.02.19
7	25.02.19
8	24.02.19
9	23.02.19
10	22.02.19

Results for Query 5

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 6:

Find all the hosts (host_ids, host_names) that have only one listing.

Description of logic:

Select the host_id and host_name attributes FROM host table and find the host_id that has only one listing in listing table

SQL statement

```
SELECT h.host_id, h.host_name
FROM host h
WHERE h.host_id IN (SELECT l.host_id
                    FROM listing l
                    GROUP BY l.host_id
                    HAVING COUNT(*) = 1);
```

	HOST_ID	HOST_NAME
1	1541784	Aga
2	1570247	Irene
3	2741652	Rodrigo
4	2747795	Stephanie
5	2764476	Sol
6	2769482	Mariona
7	3920790	Carla
8	3922188	Marta
9	3987152	Yeni
10	4145430	Pau

Results for Query 6

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 7:

What is the difference in the average price of listings with and without Wifi.

Description of logic:

A view is created to contain all the listing ids that are equipped with Wifi. Then, we use AVG to calculated the two average price and find the difference between them. The listings with and without wifi is found by using IN and NOT IN. The syntax 'SELECT * FROM dual' is used to avoid an unnecessary join.

SQL statement

```
CREATE VIEW with_wifi as (
            SELECT l.listing_id
            FROM hasamenities ha, listing l, amenities am
            WHERE l.listing_id = ha.listing_id AND ha.amenities_id =
            am.amenities_type = 'Wifi');

SELECT (
      (SELECT avg(l.price) FROM listing l
      WHERE l.listing_id in (SELECT * from with_wifi) ) -
      (SELECT avg(l.price) FROM listing l
      WHERE l.listing_id not in (SELECT * from with_wifi)) )
AS dif FROM dual;
```

	DIF
1	3.213881387155044444047001591758625006 71

Results for Query 7

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 8:

How much more (or less) costly to rent a room with 8 beds in Berlin compared to Madrid on average?

Description of logic:

Number of bedrooms and price are all attributes of listing, we have normalized the city type so it exsists in another table, but it is easy to access by simply joining them.

SQL statement

```
SELECT AVG(L1.price) - AVG(L2.price)
FROM listing L1, listing L2
WHERE L1.Bedrooms = 8 AND
      L1.NEIGHBOURHOOD_ID IN
        (SELECT n.NEIGHBOURHOOD_ID
         FROM NEIGHBOURHOOD n
         WHERE n.CITY_ID = (SELECT c.CITY_ID
                            FROM city c
                            WHERE c.CITY_NAME = 'Berlin')) AND
                            L2.Bedrooms = 8 AND
                            L2.NEIGHBOURHOOD_ID IN
                              (SELECT n.NEIGHBOURHOOD_ID
                               FROM NEIGHBOURHOOD n
                               WHERE n.CITY_ID = (SELECT c.CITY_ID
                                                  FROM city c
                                                  WHERE c.CITY_NAME = 'Madrid')) ;
```

	AVG(L1.PRICE)-AVG(L2.PRICE)
1	144.375

Results for Query 8

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 9:

Find the top-10 (in terms of the number of listings) hosts (host_ids, host_names) in Spain.

Description of logic:

First we join country, listing and host tables and group them with host_id and host_name with descending order with regard to the number of records. By constraining the number of rows to be less than 10 we could get the host_id and host_names of top 10 hosts.

SQL statement

```
SELECT * FROM
  (SELECT H.host_id, H.host_name
   FROM listing L, host H
   WHERE L.NEIGHBOURHOOD_ID IN
     (SELECT n.NEIGHBOURHOOD_ID
      FROM NEIGHBOURHOOD n
      WHERE n.CITY_ID IN (SELECT c.CITY_ID
                          FROM city c, country ct
                          WHERE ct.COUNTRY_NAME = 'Spain' AND
                                c.COUNTRY_ID = ct.COUNTRY_ID))
      AND L.host_id = h.host_id
      GROUP BY L.host_id, H.host_name, H.host_id
      ORDER BY COUNT(*) DESC)
WHERE ROWNUM <=10;
```

	HOST_ID	HOST_NAME
1	4459553	Eva&Jacques
2	99018982	Apartamentos
3	32046323	Juan
4	28038703	Luxury Rentals Madrid
5	1391607	Aline
6	299462	Stay U-Nique
7	10704	Weflating
8	3566146	Home Club
9	1408525	Mad4Rent
10	5878688	Yaiza

Results for Query 9

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 10:

Find the top-10 rated (review_score_rating) apartments (id,name) in Barcelona.

Description of logic:

Making a join between listing and located_city with a filter will get all the listings FROM Barcelona. Making a joint between listing and hasProperty with a filter will get all the listings of type Apartment. Combining the two will result in all the apartments in Barcelona. Sort them by descending order and get the top ten row in the relation. Project them on listing_id and listing_name will get the result.

SQL statement

```
SELECT * FROM(
       SELECT L.listing_id,  L.listing_name
       FROM listing l
       WHERE l.NEIGHBOURHOOD_ID IN
          (SELECT n.NEIGHBOURHOOD_ID
           FROM NEIGHBOURHOOD n
           WHERE n.CITY_ID IN
             (SELECT c.CITY_ID
              FROM city c
              WHERE c.CITY_NAME = 'Barcelona'))AND
                    l.PROPERTY_TYPE_ID IN
                      (SELECT pt.property_type_id
                       FROM property_type pt
                       WHERE pt.TYPE = 'Apartment') AND
                    l.REVIEW_SCORES_RATING IS NOT NULL
                    ORDER BY l.REVIEW_SCORES_RATING DESC)
WHERE rownum <= 10;
```

	LISTING_ID	LISTING_NAME
1	29308955	RAMBLAS SUPER CENTRAL XXL APARTAMENT
2	18913823	design loft close to sagrada familia_Ber
3	13645248	Espacioso apart 3hab en el Eixample
4	26366403	PEACEFUL COSY BEDROOM/ CHAMBRE PAISIBLE
5	27893741	Apartamento Bonanova – Sant Gervasi
6	22767977	Long Stay Apartment beautiful area of Barcelona
7	29424816	[Las Ramblas] Modernist Apartment/DoubleBedroom
8	29416416	[Las Ramblas] Modernist Apartment/Bedroom/Balcony
9	29415802	[Las Ramblas] Modernist Big Flat/Grand Bedroom
10	29336270	Triple Bedroom walk to Ramblas

Results for Query 10

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

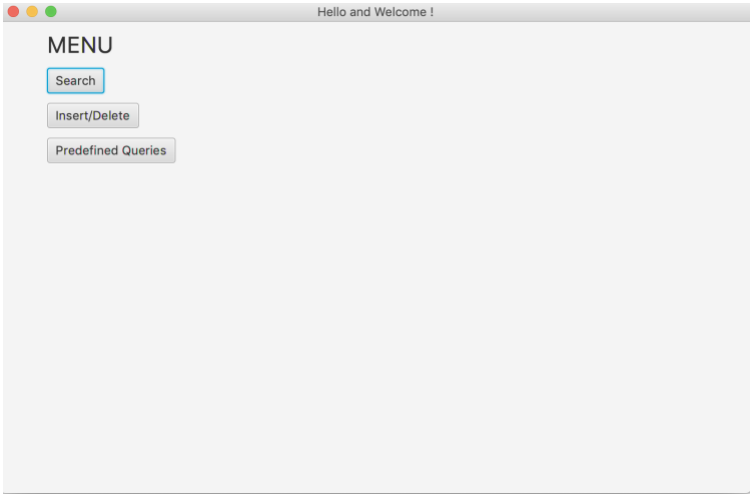
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

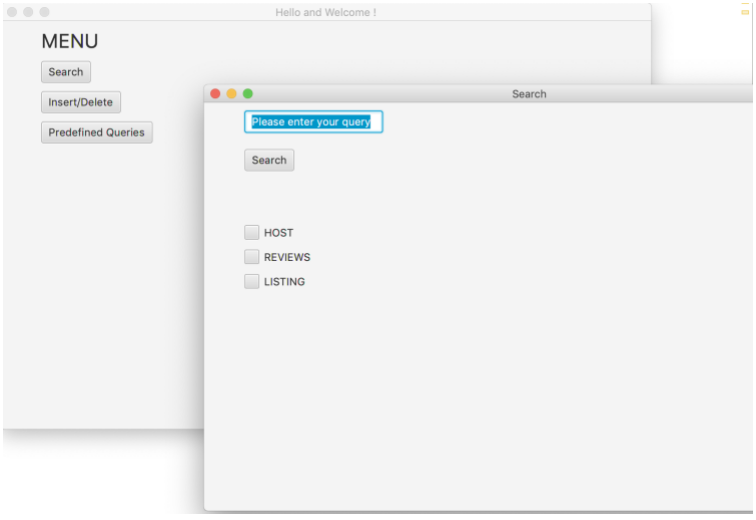
Interface

Design logic Description

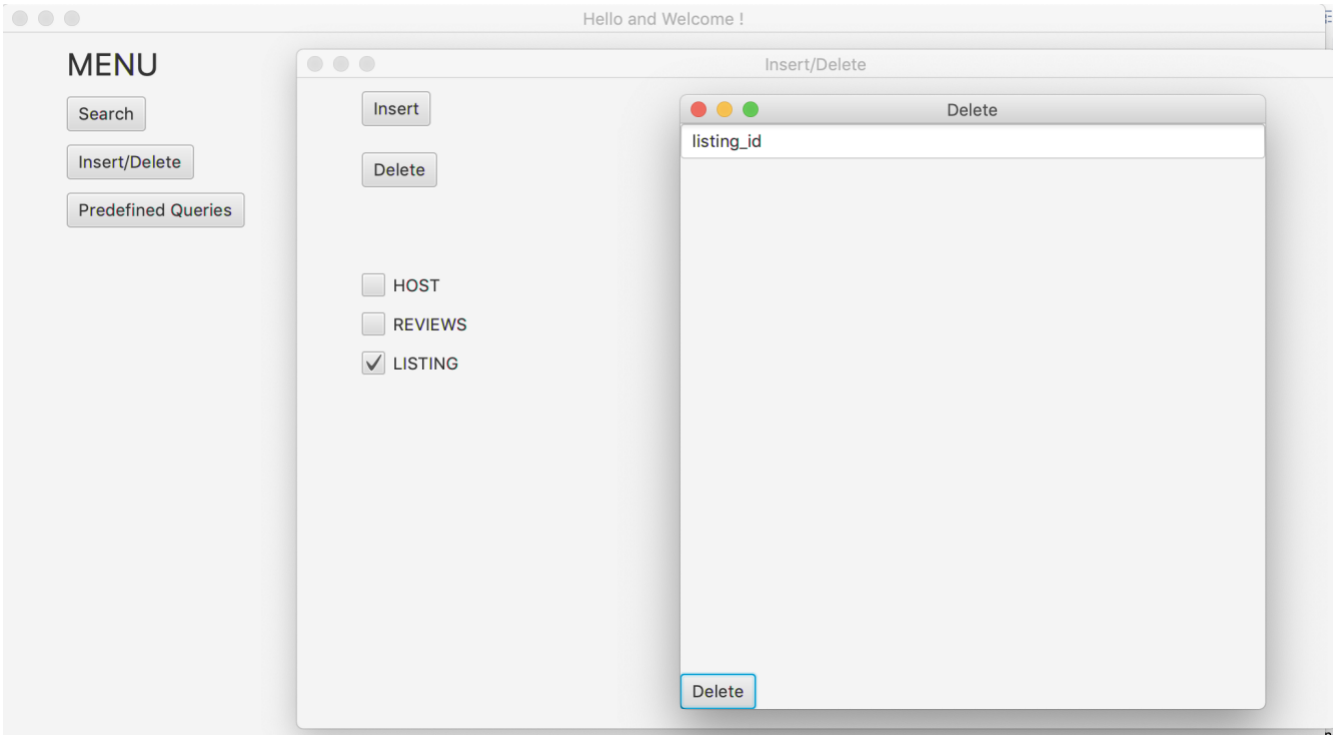
JavaFX is used to design the interface. The front window includes 3 buttons which are Search button to do basic keyword-based search, Insert/Delete button allows insert and delete and Predefined Queries button is used to show the predefined queries required by the project.

Screenshots





DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/



General Comments

Data cleaning is done by Mingbo
Data loading by Futong
Queries writing is written by all members
User interface by wan-tzu

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 3

Assumptions

<In this section write down the assumptions you made about the data. Write a sentence for each assumption you made>

1. If a host deletes his account, his associated properties will also be deleted from the listing.
2. If a reviewer deletes his account, his associated comments will still stay in the database.
3. If a property on the listing is deleted, the associated comments of it will also be deleted.
4. A host might not have a listing at all times.
5. Location, room_type, bed_type, property_type are all categorical data, and they are all filled up, i.e. they cannot be left blank.
6. One reviewer id can only correspond to one reviewer name. Namely, a reviewer cannot change his name once he obtains his reviewer id.
7. Houses in each original csv file are indeed of the indicated city. For example, houses in madrid_listings.csv are all located in Madrid.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query Implementation

Query 1:

Print how many hosts in each city have declared the area of their property in square feet. Sort the output based on the city name in ASCending order.

Description of logic:

Firstly, filter out host_id who dont havesquare_feet attributes and then join table listing, neighbourhood and city. Use "group by" to group city_name and count the number is each group and print then in ascending order.

SQL statement

```
SELECT ct.city_name, count(*) FROM LISTING l1, NEIGHBOURHOOD n, city ct
      WHERE l1.HOST_ID IN (SELECT distinct(l2.HOST_ID) FROM LISTING l2 WHERE
      l2.square_feet is not null) AND
      l1.NEIGHBOURHOOD_ID = n.NEIGHBOURHOOD_ID AND
      n.CITY_ID = ct.CITY_ID
GROUP BY ct.city_name
ORDER BY ct.city_name ASC;
```

	CITY_NAME	COUNT(*)
1	Barcelona	1660
2	Berlin	524
3	Madrid	869

Results for Query 1

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 2:

The quality of a neighborhood is defined based on the number of listings and the review score of these listings, one way for computing that is using the median of the review scores, as medians are more robust to outliers. Find the top-5 neighborhoods using median review scores of listings in Madrid. Note: Implement the median operator on your own, and do not use the available built-in operator.

Description of logic:

<What does the query do and how do I decide to solve it>
The query requires to sort the neighbourhoods in Madrid according to each neighbourhood's median review scores, and select the top five neighbourhoods.
The median of each neighbourhood is calculated as following. Firstly, for each house in table listing, we append two more attributes, one is it's median index, indicating what the median's index within its neighbourhood should be, and the other one is it's rank within its neighbourhood. Therefore, the median of review scores in each neighbourhood is when the rank equals the median index. The top five neighbourhoods are found out by sorting the neighbourhoods according to medians.

SQL statement

```
SELECT neighbourhood_name, review_scores_rating FROM(
      SELECT nb.NEIGHBOURHOOD_NAME, l.REVIEW_SCORES_RATING,
      floor((count(all *) over(partition by (l.NEIGHBOURHOOD_ID))+1)/2) AS
      median_index,
      row_number() over(partition by l.neighbourhood_id ORDER BY
      l.review_scores_rating DESC) AS rank
      FROM listing l, neighbourhood nb, city
      WHERE l.neighbourhood_id = nb.neighbourhood_id AND
            city.CITY_ID=nb.CITY_ID AND
            city.CITY_NAME = 'Madrid' AND
            l.REVIEW_SCORES_RATING is not null) medians
WHERE medians.median_index = medians.rank
ORDER BY medians.REVIEW_SCORES_RATING DESC
fetch first 5 rows only;
```

	NEIGHBOURHOOD_NAME	REVIEW_SCORES_RATING
1	Tetuán	100
2	Estrella	100
3	Hispanoamérica	98
4	Vallehermosa	98
5	Vicálvaro	98

Results for Query 2

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 3:

Find all the hosts (host_ids, host_names) with the highest number of listings.

Description of logic:

Firstly, count the number of listing host_id it has and rank the host base on the number of listing it won and call this as rankingTable. Secondly, join rankingTable and host table and filter out whose rankingTable.rank is not 1

SQL statement

```
SELECT h.host_id, h.host_name FROM
       (SELECT host_id, rank() over(ORDER BY numListingPerHost DESC) AS rank FROM
             (SELECT l.host_id, count(*)AS numListingPerHost FROM
                   listing l
                   GROUP BY l.host_id)) rankingTable,
       host h
WHERE h.host_id = rankingTable.host_id AND rankingTable.rank = 1;
```

	HOST_ID	HOST_NAME
1	4459553	Eva&Jacques

Results for Query 3

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 4:

Find the 5 most cheapest Apartments (based on average price within the available dates) in Berlin available between 01-03-2019 and 30-04-2019 having at least 2 beds, a location review score of at least 8, flexible cancellation, and listed by a host with a verifiable government id.

Description of logic:

First we got filtered CALENDAR with the constraints on date and filtered LISTING with the constraints on the number of beds , location review scores, verification type by joining HOST_VERIFICATION, cancellation type by joining with CANCELLATION, located in Berlin by join NEIGHBOURHOOD and CITY, then by group the LISTING_ID we could calculate the average price for every listing, finally we could got the top-5 cheapest apartments in Berlin by ordering the average price with an ascending order and only select the first 5 records to output

SQL statement

```
SELECT * FROM
 (SELECT AVG(cal.Price) AS avgPrice, cal.listing_id FROM
  (SELECT l.listing_id FROM
    listing l, NEIGHBOURHOOD neigh, city c, property_type p, cancellation canc
    WHERE neigh.city_id = c.city_id AND
          c.city_name = 'Berlin' AND
          l.property_type_id = p.property_type_id AND
          p.type = 'Apartment' AND
          l.beds>=2 AND
         l.review_scores_location >=8 AND
         l.neighbourhood_id = neigh.neighbourhood_id AND
          canc.cancellation_id = l.cancellation_policy_id AND
          canc.cancellation_policy = 'flexible' AND
          l.host_id IN
           (SELECT h.host_id FROM host h, host_verification hv, hasverification hhv
             WHERE h.host_id = hhv.host_id AND
                  hv.verification_id = hhv.verification_id AND
                  hv.verification_type LIKE '%government_id%')) accepted_listing,
 CALENDARAVAILABLE cal
 WHERE cal.CALENDAR_DATE >= '01-MAR-19' AND
       cal.CALENDAR_DATE <=  '30-APR-19' AND
        accepted_listing.listing_id = cal.listing_id AND
        cal.AVAILABILITY = 't'
 GROUP BY cal.LISTING_ID
 ORDER BY avgPrice ASC)
WHERE rownum <= 5;
```

	AVGPRICE	LISTING_ID
1	20	1490274
2	21.065573770491803278688524590163934426 23	24043706
3	21.290322580645161290322580645161290322 58	1368460
4	22	7071541
5	22	6691656

Results for Query 4

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 5:

Each property can accommodate different number of people (1 to 16). Find the top-5 rated (review_score_rating) listings for each distinct category based on number of accommodated guests with at least two of these facilities: Wifi, Internet, TV, and Free street parking.

Description of logic:

First, we got filtered LISTING_ID by summing all kinds of amenities for every LISTING because we have a constraints on the number of facilities(at least 2 of mentioned facilities), then we join these tables and also constrain on the number of accommodates, number of target facilities and also ensure that the review scores rating is not null, then we sort the accommodates attribute with ascending order, finally we could get top-5 rated listings for different number of accommodates(range 1 to 16)

SQL statement

```
SELECT * FROM
  (SELECT accepted_listing.listing_id, l.accommodates,l.review_scores_rating,
         row_number() over(partition by l.accommodates ORDER BY
         l.review_scores_rating DESC) AS row_number
   FROM (SELECT  hA.listing_id, count(*) AS num FROM hasAmenities hA, Amenities A
          WHERE hA.amenities_id = A.amenities_id AND(
                A.amenities_type = 'WIFI' OR
                A.amenities_type = 'Internet' OR
                A.amenities_type = 'TV' OR
                A.amenities_type = 'Free street parking')
           GROUP BY hA.listing_id) accepted_listing,
           LISTING l
  WHERE l.listing_id = accepted_listing.listing_id AND
        l.accommodates <=16 AND
        accepted_listing.num >= 2 AND
        l.review_scores_rating is not null
  ORDER BY l.accommodates ASC)
WHERE row_number <= 5;
```

	LISTING_ID	ACCOMMODATES	REVIEW_SCORES_RATING	ROW_NUMBER
1	16485441	1	100	1
2	12442906	1	100	2
3	27646863	1	100	3
4	28777906	1	100	4
5	28238152	1	100	5
6	18508649	2	100	1
7	6787760	2	100	2
8	4403695	2	100	3
9	15668177	2	100	4
10	15083699	2	100	5
11	21795596	3	100	1
12	665882	3	100	2
13	28482023	3	100	3
14	28016965	3	100	4
15	27824958	3	100	5

Results for Query 5

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 6:

What are top three busiest listings per host? The more reviews a listing has, the busier the listing is.

Description of logic:

Firstly, join listing table and reviewed table and count the number of reviewed each listing has. Then use row_number() to returns the sequential number of a row and filter the row that is less than 3

SQL statement

```
SELECT HOST_ID, LISTING_ID FROM
  (SELECT HOST_ID, LISTING_ID, ROW_NUMBER() over(partition by HOST_ID
          ORDER BY NumberOfReviewPerList DESC) AS RANK FROM
          (SELECT distinct(l.LISTING_ID), l.HOST_ID,
           count(*) over(partition by l.LISTING_ID) AS NumberOfReviewPerList
           FROM LISTING l, REVIEWED r
           WHERE l.LISTING_ID = r.LISTING_ID))
WHERE RANK <= 3;
```

	HOST_ID	LISTING_ID
1	2217	2015
2	2217	21315310
3	2217	18773184
4	3073	6287375
5	3718	3176
6	4108	3309
7	5154	18132872
8	10704	8217664
9	10704	733941
10	10704	9572534

Results for Query 6

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 7:

What are the three most frequently used amenities at each neighborhood in Berlin for the listings with “Private Room” room type?

Description of logic:

First we join AMENITIES, LISTING, HASAMENITIES, NEIGHBOURHOOD, CITY and ROOM_TYPE these tables and filter them with the constraints on room type, city name , and we calculate the statistics of the number of used amenities for every neighbourhood and amenity. Then we use row_number() function on neighbourhood name and the summary statistics we calculated before, by sorting this statistics with descending order and select the first 3 rows could provide us with the 3 most frequently used amenties.

SQL statement

```
SELECT AMENITIES_TYPE, NEIGHBOURHOOD_NAME FROM
  (SELECT AMENITIES_TYPE, AMEN_CNT, NEIGHBOURHOOD_NAME,
          row_number() over(partition by SORTED_AMEN.NEIGHBOURHOOD_NAME
                            ORDER BY AMEN_CNT DESC) AS SORT
   FROM (SELECT distinct AMENITIES_TYPE, AMEN_CNT, NEIGHBOURHOOD_NAME FROM
          (SELECT A.AMENITIES_TYPE, count(AMENITIES_TYPE)over(partition by
            NEIGH.NEIGHBOURHOOD_NAME, A.AMENITIES_TYPE) AS AMEN_CNT,
            NEIGH.NEIGHBOURHOOD_NAME FROM AMENITIES A, LISTING L, hasAmenities hA,
            NEIGHBOURHOOD NEIGH, CITY C, ROOM_TYPE RT
           WHERE     L.LISTING_ID = hA.LISTING_ID AND
                     hA.AMENITIES_ID = A.AMENITIES_ID AND
                     L.NEIGHBOURHOOD_ID = NEIGH.NEIGHBOURHOOD_ID AND
                      NEIGH.city_id = C.city_id AND
                     C.city_name = 'Berlin' AND
                      L.ROOM_TYPE_ID = RT.ROOM_TYPE_ID AND
                     RT.ROOM_TYPE = 'Private room') AMEN
        ORDER BY AMEN.NEIGHBOURHOOD_NAME, AMEN.AMEN_CNT DESC)SORTED_AMEN)final_data
WHERE final_data.SORT <= 3
```

	AMENITIES_TYPE		NEIGHBOURHOOD_NAME
1	Essentials	...	Adlershof
2	Heating	...	Adlershof
3	Wifi	...	Adlershof
4	Heating	...	Alt-Hohenschönhausen
5	Essentials	...	Alt-Hohenschönhausen
6	Wifi	...	Alt-Hohenschönhausen
7	Wifi	...	Alt-Treptow
8	Essentials	...	Alt-Treptow
9	Heating	...	Alt-Treptow
10	Heating	...	Altglienicke
11	Kitchen	...	Altglienicke
12	Wifi	...	Altglienicke
13	Wifi	...	Baumschulenweg
14	Kitchen	...	Baumschulenweg

Results for Query 7

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Query 8:

What is the difference in the average communication review score of the host who has the most diverse way of verifications and of the host who has the least diverse way of communication. In case of a multiple number of the most or the least diverse communicating hosts, pick a host one FROM the most and one FROM the least communicating hosts.

Description of logic:

To make things easier, we create a auxiliary table NUM_HOST_VERIFICATION with host_id and the total number of verification types of every host. The calculation could be divided into 2 parts: we calculate the average communication review score for host with most diverse way of verifications and for host with least diverse way of verifications respectively. Therefore, for the host with most diverse way of verifications, by sorting the number of verifications with descending order and select the first row we could get the host with most diverse way, then we join the filtered table with LISTING and get review communication scores corresponding to the host with most diverse verifications. The logics for the host with least diverse way of verifications are the same.
After that we just need to average the review communication scores for these 2 hosts respectively and make a subtraction.

SQL statement

```
CREATE VIEW NUM_HOST_VERIFICATIONS AS
  SELECT H.HOST_ID, COUNT(HASV.VERIFICATION_ID) over(partition by HASV.HOST_ID) AS
  VERI_NUM FROM HOST H, HASVERIFICATION HASV, HOST_VERIFICATION HV
  WHERE H.HOST_ID = HASV.HOST_ID AND HASV.VERIFICATION_ID = HV.VERIFICATION_ID;

SELECT AVG(AVG_MOST.AVG_MOST_SCORE-AVG_MIN.AVG_LEAST_SCORE) AS GAP FROM
  (SELECT coalesce(AVG(L1.REVIEW_SCORES_COMMUNICATION), 0) AS AVG_MOST_SCORE FROM
    (SELECT filter1.HOST_ID FROM
       (SELECT NV.HOST_ID FROM NUM_HOST_VERIFICATIONS NV
        ORDER BY NV.VERI_NUM DESC)filter1
        WHERE rownum = 1) MAX_HOST,
   LISTING L1
   WHERE L1.HOST_ID = MAX_HOST.HOST_ID AND
         L1.REVIEW_SCORES_COMMUNICATION is not null)AVG_MOST,

  (SELECT coalesce(AVG(L2.REVIEW_SCORES_COMMUNICATION), 0) AS AVG_LEAST_SCORE FROM
    (SELECT filter2.HOST_ID FROM
       (SELECT NV2.HOST_ID FROM NUM_HOST_VERIFICATIONS NV2
        ORDER BY NV2.VERI_NUM ASC)filter2
        WHERE rownum = 1) MIN_HOST,
     LISTING L2
     WHERE L2.HOST_ID = MIN_HOST.HOST_ID AND
           L2.REVIEW_SCORES_COMMUNICATION is not null)AVG_MIN;

drop view NUM_HOST_VERIFICATIONS;
```

	GAP
1	10

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Results for Query 8

Query 9:

What is the city who has the highest number of reviews for the room types whose average number of accommodates are greater than 3.

Description of logic:

First we have to join LISTING, ROOM_TYPE and calculate the average number of accommodates for every kind of room type, then we summarize the received total number of reviews of every listing from REVIEWED table.

Since we care about the city and room types, so we take partition by city and room types and summaries the total number of reviews for the listing located in the same city and is of the same room type. Then we take partition by room type and order by the number of total received reviews, select the records with rank equals to 1 and we could get the corresponding city name which has the highest number of reviews for different room types.

We use rank() here because we have considered the condition when there comes two or more same highest number of reviews for the same city.

SQL statement

```
SELECT ROOM_TYPE, CITY_NAME FROM
  (SELECT Filter_Listing.ROOM_TYPE, Filter_Listing.CITY_NAME, rank() over(partition
          by Filter_Listing.ROOM_TYPE ORDER BY total) AS rank FROM
    (SELECT distinct(AVG_ACC.ROOM_TYPE),average_per_room_type,sum(REVIEW_LISTING)
      over(partition by CITY_NAME, RT.ROOM_TYPE) AS total, CITY_NAME FROM
       LISTING L, CITY C, NEIGHBOURHOOD NEIGH, ROOM_TYPE RT,
        (SELECT distinct(RVD.LISTING_ID), count(*)
        over(partition by RVD.LISTING_ID) AS REVIEW_LISTING FROM REVIEWED RVD) R_L,
          (SELECT L.LISTING_ID, RT.ROOM_TYPE, AVG(L.ACCOMMODATES)
           over(partition by RT.ROOM_TYPE) AS average_per_room_type
             FROM LISTING L, ROOM_TYPE RT
             WHERE L.ROOM_TYPE_ID = RT.ROOM_TYPE_ID)AVG_ACC

         WHERE R_L.LISTING_ID = AVG_ACC.LISTING_ID AND
               average_per_room_type > 3 AND
               L.LISTING_ID = AVG_ACC.LISTING_ID AND
               L.NEIGHBOURHOOD_ID = NEIGH.NEIGHBOURHOOD_ID AND
               NEIGH.CITY_ID = C.CITY_ID AND
              RT.ROOM_TYPE_ID = L.ROOM_TYPE_ID)Filter_Listing)
WHERE RANK = 1;
```

	ROOM_TYPE	CITY_NAME
1	Entire home/apt	Berlin
2	Shared room	Madrid

Results for Query 9

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Query 10:

Print all the neighborhoods in Madrid which have at least 50 percent of their listings occupied at some date in year 2019 and their host has joined airbnb before 01.06.2017

Description of logic:

<What does the query do and how do I decide to solve it>
The query requires to find correct neighbourhoods in Madrid according to the specifications.
Firstly we compute the number of required listings in each neighbourhoods, which is in Madrid, occupied in year 2019 and their host joined airbnb before 2017-06-01. Secondly, we compute the total number of listings in each neighbourhood. Finally, these two tables are joined on attribute neighbourhood_id to get the occupancy ratio of each neighbourhood. The neighbourhoods that have a ratio larger than 50% are then printed.

SQL statement

<The SQL statement>

```
SELECT nb.NEIGHBOURHOOD_NAME
FROM (SELECT l.neighbourhood_id, count(distinct l.listing_id) filtered
      FROM listing l, neighbourhood nb, city, calendaravailable cal, host h
       WHERE l.neighbourhood_id = nb.neighbourhood_id AND
             nb.city_id = city.city_id AND
             city.city_name ='Madrid' AND
             cal.listing_id = l.listing_id AND
              cal.availability = 'f' AND
              CAL.CALENDAR_DATE >= date '2019-01-01' AND
              l.host_id = h.host_id AND
              h.host_since <= date '2017-06-01'
       GROUP BY l.neighbourhood_id) filtered,
       (SELECT l.neighbourhood_id, count(distinct l.listing_id) total
       FROM listing l, neighbourhood nb, city
       WHERE l.neighbourhood_id = nb.neighbourhood_id AND
             nb.city_id = city.city_id AND
             city.city_name ='Madrid'
       GROUP BY l.neighbourhood_id) total,
       neighbourhood nb
WHERE (filtered.filtered/total.total)>=0.5 AND
       filtered.neighbourhood_id = total.neighbourhood_id AND
      nb.NEIGHBOURHOOD_ID = filtered.neighbourhood_id;
```

	NEIGHBOURHOOD_NAME
1	Embajadores
2	La Latina
3	Recoletos
4	Almagro
5	Sol
6	Cortes
7	Justicia
8	Carabanchel
9	Malasaña
10	Palacio

Results for Query 10

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 11:

Print all the countries that at least 20% of their listings were available in 2018.

Description of logic:

<What does the query do and how do I decide to solve it>
The query aims to find a ratio between the number of adequate listings and the total listing amount.
Firstly, we find out the required number of listings, which are filtered by setting the calendar date between 2018-01-01 and 2018-12-31 with availability status as true, and then joined with country in chain. Then the total number of listings in each country is also computed in this way. Finally, the two results are joined according to their country and the ratio between them is calculated. The countries with a ratio larger than 0.2 is printed.

SQL statement

<The SQL statement>

```
SELECT country.country_name
FROM (SELECT count(distinct l.listing_id) AS available_amount,
             country.country_id AS country_id
      FROM listing l, calendaravailable cal, neighbourhood nbhd, city, country
      WHERE l.listing_id = cal.listing_id AND
            CAL.CALENDAR_DATE >= date '2018-01-01' AND
            CAL.CALENDAR_DATE < date '2019-01-01' AND
            cal.availability = 't' AND
            l.neighbourhood_id = nbhd.neighbourhood_id AND
            nbhd.city_id = city.city_id AND
            city.country_id = country.country_id
      GROUP BY country.country_id) filtered,
      (SELECT count(*) AS total_amount, country.country_id AS country_id
       FROM listing l, neighbourhood nbhd, city, country
       WHERE l.neighbourhood_id = nbhd.neighbourhood_id AND
       nbhd.city_id = city.city_id AND
       city.country_id = country.country_id
       GROUP BY country.country_id) total,
       country
WHERE total.country_id = filtered.country_id AND
      (filtered.available_amount/ total.total_amount) >= 0.2 AND
      country.country_id = filtered.country_id;
```

	COUNTRY_NAME
1	Germany
2	Spain

Results for Query 12

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 12:

Print all the neighborhoods in Barcelona WHERE more than 5 percent of their accommodation's cancelation policy is strict with grace period.

Description of logic:

<What does the query do and how do I decide to solve it>
The query aims to find a subset of neighbourhood names in Barcelona.
Firstly I filter the listings to find the listings in Barcelona, with specified cancellation policy, and then aggregate to get the count by neighbourhood. I also computed the total number of listings in each neighbourhood in Barcelona. By taking the ratio of filtered and the total in each neighbourhood, I can get the percentage in the question. Then, filter out the neighbourhoods that are more than 5% and print their names.

SQL statement

<The SQL statement>

```
SELECT nb.neighbourhood_name
FROM (SELECT l.neighbourhood_id, count(*) filtered
       FROM listing l, neighbourhood nb, city, cancellation cc
       WHERE l.cancellation_policy_id = cc.cancellation_id AND
             cc.cancellation_policy = 'strict_14_with_grace_period' AND
             l.neighbourhood_id = nb.neighbourhood_id AND
             nb.city_id = city.city_id AND
              city.city_name = 'Barcelona'
       GROUP BY l.neighbourhood_id) filtered,
       (SELECT l.neighbourhood_id, count(*) total
        FROM listing l, neighbourhood nb, city
        WHERE l.neighbourhood_id = nb.neighbourhood_id AND
              nb.city_id = city.city_id AND
               city.city_name = 'Barcelona'
        GROUP BY l.neighbourhood_id) total,
        neighbourhood nb
WHERE (filtered.filtered/total.total)>0.05 AND
       filtered.neighbourhood_id = total.neighbourhood_id AND
       nb.NEIGHBOURHOOD_ID = filtered.neighbourhood_id;
```

	NEIGHBOURHOOD_NAME
1	El Camp de l'Arpa del Clot
2	La Nova Esquerra de l'Eixample
3	Camp d'en Grassot i Gràcia Nova
4	el Fort Pienc
5	Diagonal Mar – La Mar Bella
6	Vila de Gràcia
7	El Gòtic
8	El Poble-sec
9	Sants-Montjuïc
10	Dreta de l'Eixample

Results for Query 12

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query Analysis

Selected Queries (and why)

Query 2

<Initial Running time: 70ms
Optimized Running time: 25ms
Explain the improvement:

The attribute neighbourhood_id in table is firstly a foreign key referencing table neighbourhood and acts as the join attribute. Secondly, neighbourhood_id is also used to count the number of records that are grouped by neighbourhood. On the other hand, in table listing, attribute review_scores_rating requires to be sorted when counting the row number when partitioned by neighbourhood. Therefore, it is beneficial to combine the the two attributes as a compound index.

Initial plan

```
SELECT neighbourhood_name, review_scores_rating FROM
 (SELECT nb.NEIGHBOURHOOD_NAME, l.REVIEW_SCORES_RATING,
         floor((count(all *) over(partition by (l.NEIGHBOURHOOD_ID))+1)/2)
                                                              AS median_index,
         row_number() over(partition by l.neighbourhood_id ORDER BY
                                l.review_scores_rating DESC) AS rank
  FROM listing l, neighbourhood nb, city
  WHERE l.neighbourhood_id = nb.neighbourhood_id AND
        city.CITY_ID=nb.CITY_ID AND
        city.CITY_NAME = 'Madrid' AND
        l.REVIEW_SCORES_RATING is not null) medians
WHERE medians.median_index = medians.rank
ORDER BY medians.REVIEW_SCORES_RATING DESC
fetch first 5 rows only;
```

Improved plan>

```
CREATE INDEX index_q2 on listing( REVIEW_SCORES_RATING,neighbourhood_id );

SELECT neighbourhood_name, review_scores_rating FROM
 (SELECT nb.NEIGHBOURHOOD_NAME, l.REVIEW_SCORES_RATING,
         floor((count(all *) over(partition by (l.NEIGHBOURHOOD_ID))+1)/2)
                                                              AS median_index,
         row_number() over(partition by l.neighbourhood_id ORDER BY
                                l.review_scores_rating DESC) AS rank
  FROM listing l, neighbourhood nb, city
  WHERE l.neighbourhood_id = nb.neighbourhood_id AND
        city.CITY_ID=nb.CITY_ID AND
        city.CITY_NAME = 'Madrid' AND
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

```
        l.REVIEW_SCORES_RATING is not null) medians
WHERE medians.median_index = medians.rank
ORDER BY medians.REVIEW_SCORES_RATING DESC
fetch first 5 rows only;

DROP INDEX index_q2;
```

Query 3

<Initial Running time: 52ms
Optimized Running time: 17ms
Explain the improvement:

The attribute of host_id in Table LISTING is a foreign key referencing Table HOST and here it also acts as the join attribute since we have to join 2 tables with host_id.

Initial plan

```
SELECT h.host_id, h.host_name FROM
      (SELECT host_id, rank() over(ORDER BY numListingPerHost DESC) AS rank
       FROM (SELECT l.host_id, count(*) AS numListingPerHost FROM listing l
             GROUP BY l.host_id)) rankingTable,
             host h
       WHERE h.host_id = rankingTable.host_id AND
             rankingTable.rank = 1;
```

Improved plan>

```
CREATE INDEX index_q2_host_id on LISTING (HOST_ID);
SELECT h.host_id, h.host_name FROM
      (SELECT host_id, rank() over(ORDER BY numListingPerHost DESC) AS rank
       FROM (SELECT l.host_id, count(*) AS numListingPerHost FROM listing l
             GROUP BY l.host_id)) rankingTable,
             host h
       WHERE h.host_id = rankingTable.host_id AND
             rankingTable.rank = 1;
DROP INDEX index_q2_host_id;
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query 12

<Initial Running time: 101 ms
Optimized Running time: 22 ms
Explain the improvement:
In table listing, both attributes cancellation_policy_id and neighbourhood_id are foreign keys referencing the tables to join. The compound index on (cancellation_policy_id, neighbourhood_id) helps to perform join more efficiently and faster.

Initial plan

```
SELECT nb.neighbourhood_name
FROM (SELECT l.neighbourhood_id, count(*) filtered
      FROM listing l, neighbourhood nb, city, cancellation cc
      WHERE l.cancellation_policy_id = cc.cancellation_id AND
            cc.cancellation_policy = 'strict_14_with_grace_period' AND
            l.neighbourhood_id = nb.neighbourhood_id AND
            nb.city_id = city.city_id AND
            city.city_name = 'Barcelona'
      GROUP BY l.neighbourhood_id) filtered,
      (SELECT l.neighbourhood_id, count(*) total
       FROM listing l, neighbourhood nb, city
       WHERE l.neighbourhood_id = nb.neighbourhood_id AND
             nb.city_id = city.city_id AND
             city.city_name = 'Barcelona'
       GROUP BY l.neighbourhood_id) total,
       neighbourhood nb
WHERE (filtered.filtered/total.total)>0.05 AND
      filtered.neighbourhood_id = total.neighbourhood_id AND
      nb.NEIGHBOURHOOD_ID = filtered.neighbourhood_id;
```

Improved plan>

```
CREATE INDEX index_q12 on listing(cancellation_policy_id, neighbourhood_id);

SELECT nb.neighbourhood_name
FROM (SELECT l.neighbourhood_id, count(*) filtered
      FROM listing l, neighbourhood nb, city, cancellation cc
      WHERE l.cancellation_policy_id = cc.cancellation_id AND
            cc.cancellation_policy = 'strict_14_with_grace_period' AND
            l.neighbourhood_id = nb.neighbourhood_id AND
            nb.city_id = city.city_id AND
            city.city_name = 'Barcelona'
      GROUP BY l.neighbourhood_id) filtered,
      (SELECT l.neighbourhood_id, count(*) total
       FROM listing l, neighbourhood nb, city
       WHERE l.neighbourhood_id = nb.neighbourhood_id AND
             nb.city_id = city.city_id AND
             city.city_name = 'Barcelona'
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/


```
       GROUP BY l.neighbourhood_id) total,
       neighbourhood nb
WHERE (filtered.filtered/total.total)>0.05 AND
      filtered.neighbourhood_id = total.neighbourhood_id AND
      nb.NEIGHBOURHOOD_ID = filtered.neighbourhood_id;


DROP INDEX index_q12;
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

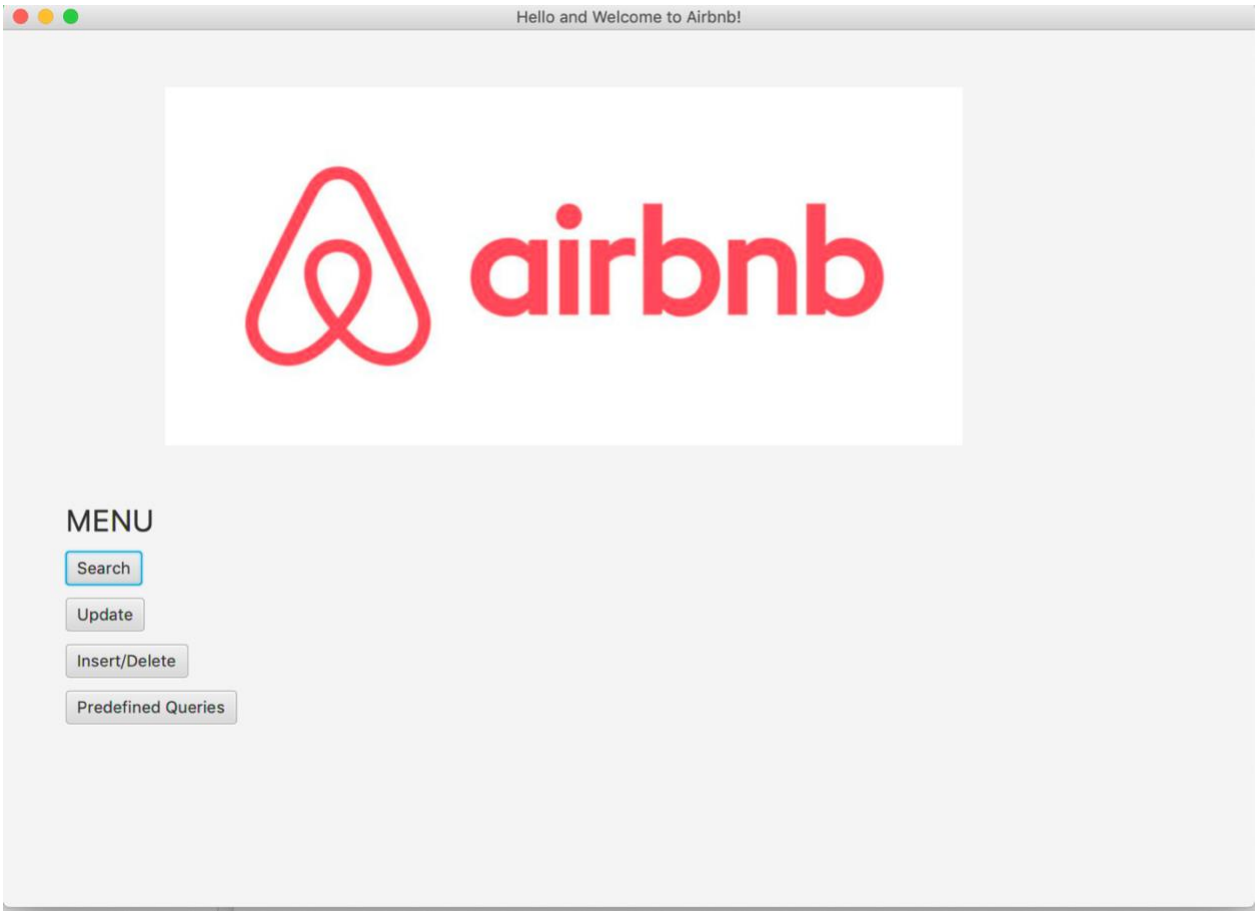
Interface

Design logic Description

JavaFX is used to design the interface. The front window includes 4 buttons which are Search button, Insert/Delete button, Update button and Predefined Queries button. The search button allows user to go to search page which enable them to do basic keyword-based search. The search page guides the use to type a keyword-based and to select a table to do the search. Furthermore, the search page allows to select specific column to do the query. The Insert/Delete button allows user to insert and delete the record. The update page allows the user to update with specific table, column and values. Predefined Queries button is used to show the 22 predefined queries required by the project.

Screenshots

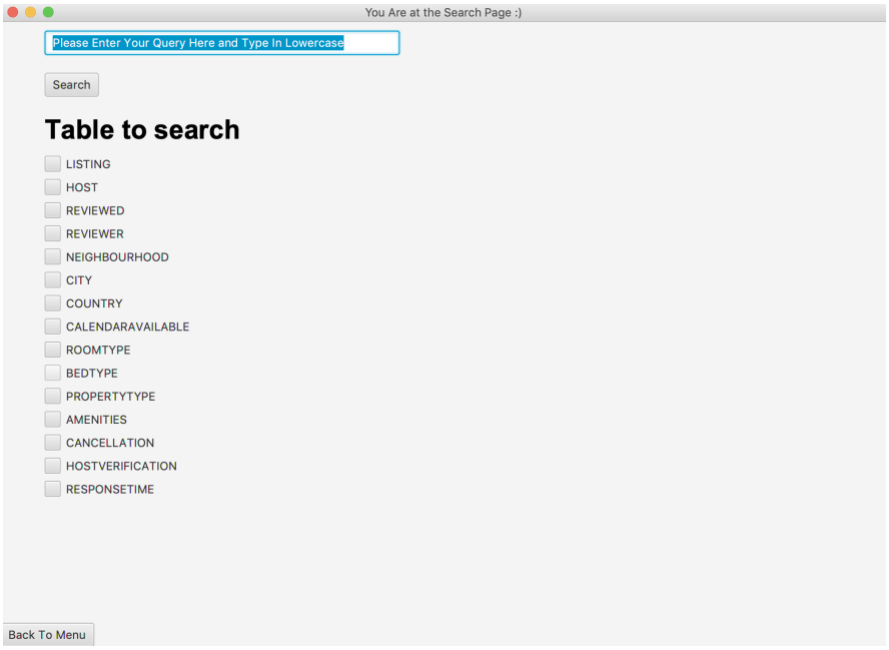
Front Page

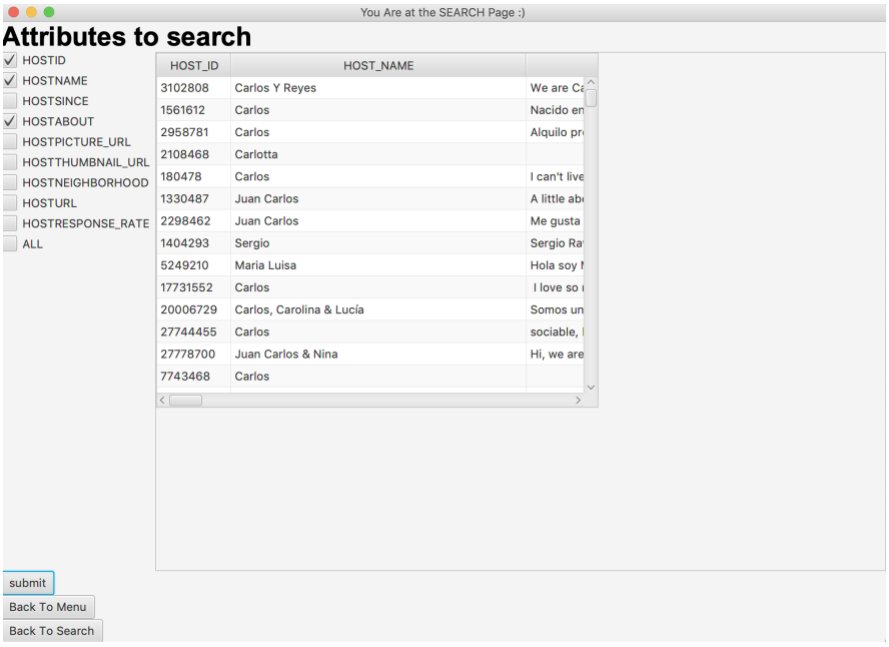


DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Page for searching

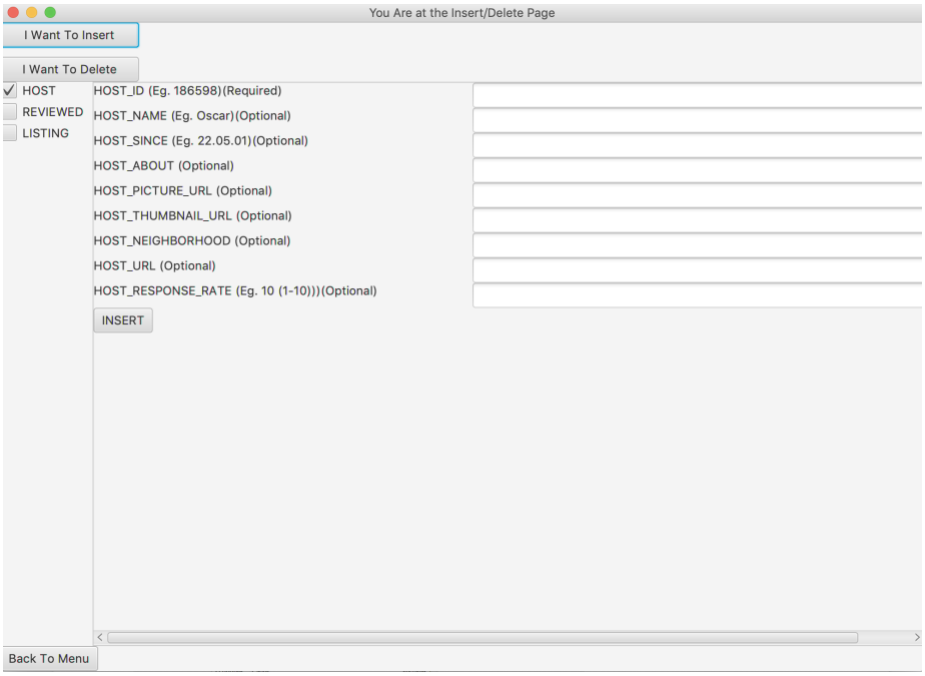


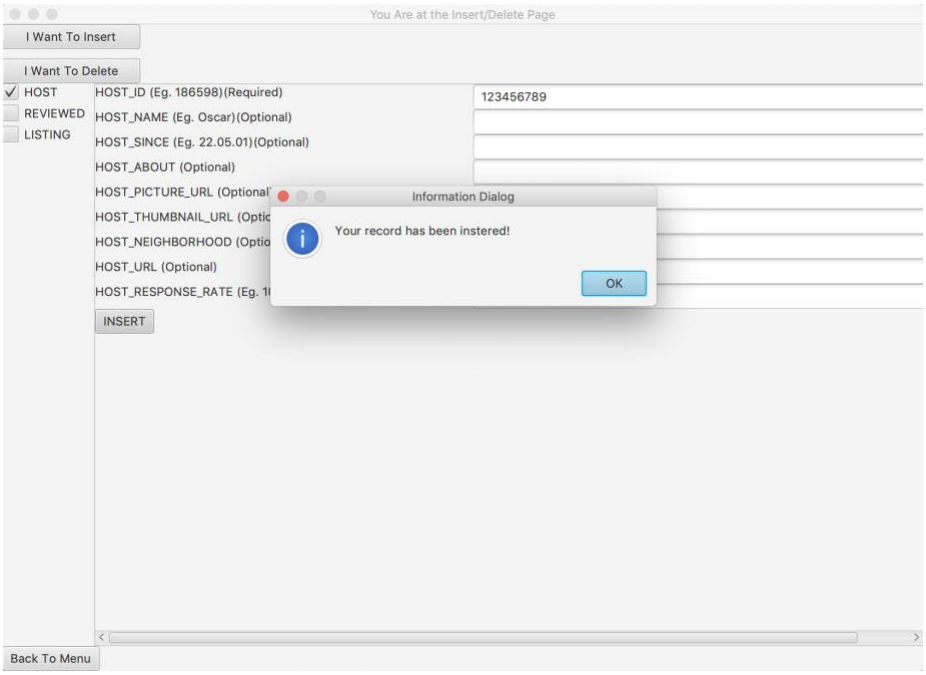


DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Page for inserting and deleting

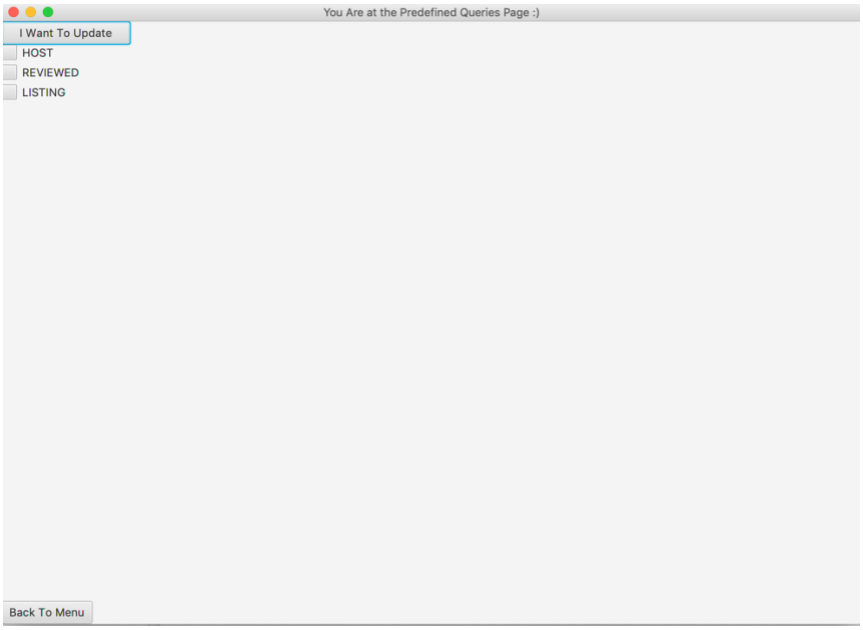


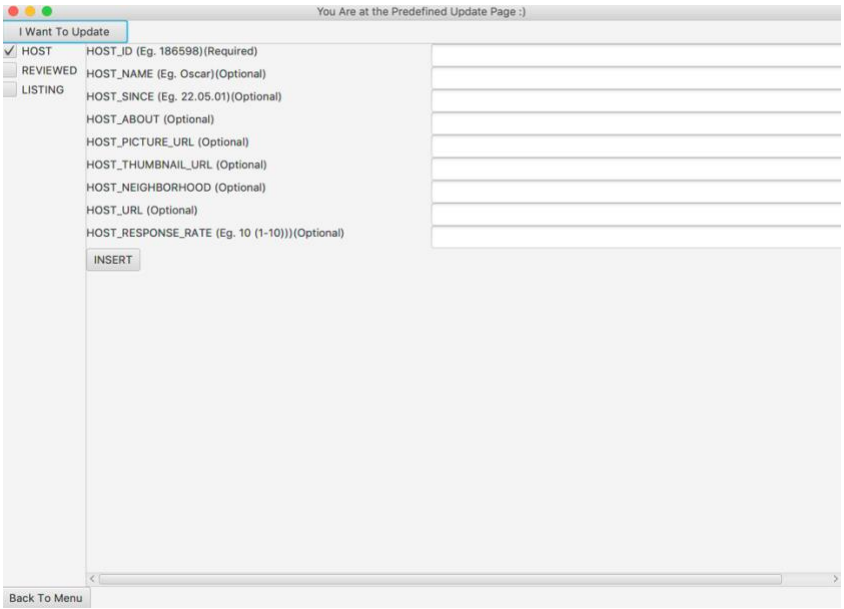


DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

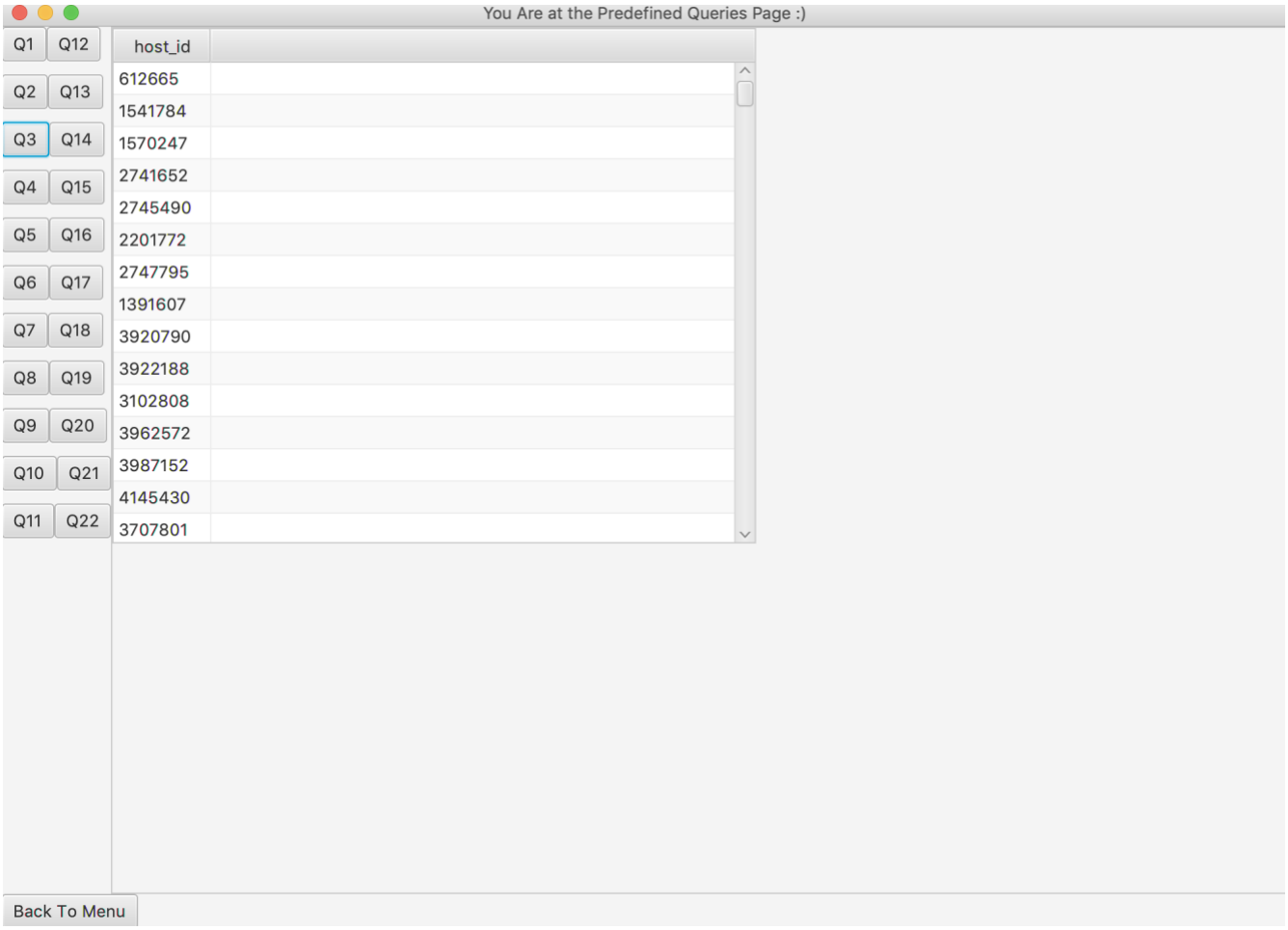
Page for update





DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Page for predefined queries



General Comments

Interface by Wan-Tzu

Query implement by Mingbo, Futong and Wan-Tzu

Query optimization by Mingbo, Futong