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/

Databases Project – Spring 2019

Team No: 26

Names: Camilla Giaccari, Hédi Sassi, Simon Perriard

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/

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/

Deliverable 1

Assumptions

We made no assumption concerning the correctness of the data, we checked every field of every CSV file. The type of each field has been checked and each line containing a wrong input (i.e. missing mandatory field, negative price,...) has been kicked out of the dataset.

We defined some mandatory fields, listed below.

Listings: listing_id, listing_url, listing_name, host_id, host_url, host_name

Reviews: all fields are mandatory

Calendar: all fields except price are mandatory

Entity Relationship Schema

Schema

The schema can be found here :
https://github.com/hedi-sassi/rbnb_db_project/tree/master/ER

Description

First, we chose to separate the main listing attributes and the listing's details. This way, when we want display a lot of listings, we only query the table with the main information

(small description, name, thumbnail etc..) and if someone select the listing, we also query the tables containing the details.

We did the same for the host and host details tables.

The listing table is connected (one to one mapping) with the calendar, review scores, material description and cost details tables. This implies they are all weak entities with respect to the listing table.

We decided to create special tables to hold the amenities and the host verifications as those are list attributes. We link them to the listing using intermediate tables containing the listing id and the amenities/host verifications id.

Relational Schema

ER schema to Relational schema

The schema can be found here :
https://github.com/hedi-sassi/rbnb_db_project/tree/master/relational_model

Weak entities are accounted for with the help of foreign keys. If the foreign key is not present, it will trigger a “Cascade” deletion policy.

DDL

The DDL can be found here:
https://github.com/hedi-sassi/rbnb_db_project/tree/master/relational_model

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

We split the work as followed:

- ER model: Camilla

- Relational Model : Simon

- Data verification (scala program on the repo) : Hédi

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/

Deliverable 2

Assumptions

We made no assumption about the data.

The ER schema and the Relational schema have been updated following the directions given by the TAs for Milestone1.

Data Loading

We organized the data as described in the Relational schema and imported it using SQLDeveloper.

Query Implementation

Query 1:

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

Description of logic:

We take the average value of the price attribute, considering only the listings with 8 bedrooms in their material description.

SQL statement

```
select AVG(CD.PRICE)

from COSTS_DETAILS CD , LISTING L

where L.LISTING_ID = CD.LISTING_ID AND L.LISTING_ID IN (Select M.LISTING_ID

                                        from MATERIAL_DESCRIPTION M

                                        where M.BEDROOMS = 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/

Result
Only one row with value 313,153846153846153846153846153846153846

Query 2:
What is the average cleaning review score for listings with TV?

Description of logic:
We take the average value of the REVIEW_SCORES_CLEANLINESS attribute, considering only the listings with 'TV' in their amenities.

SQL statement
```
select AVG(RS.REVIEW_SCORES_CLEANLINESS)

from REVIEWS_SCORES RS, LISTING L

where RS.LISTING_ID = L.LISTING_ID AND L.LISTING_ID IN (select LA.LISTING_ID
                                    from AMENITIES AM, LISTING_AMENITIES LA
                                    where AM.AMENITY_ID = LA.AMENITY_ID and
AM.AMENITY_NAME = 'TV' );
```

Result
Only one row with value 9,39864565813932902540497477206337965832

Query 3:
Print all the hosts who have an available property between date 03.2019 and 09.2019.

Description of logic:
We take all the informations about the hosts who have a listing with available date following (>=) 03.2019 and prior (<=) 09.2019.

SQL statement
```
select *

from HOST H

where H.HOST_ID IN (select L.HOST_ID
            from LISTING L, CALENDAR CA
```

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

where L.LISTING_ID = CA.LISTING_ID and CA.AVAILABLE = 't' and CA.CALENDAR_DATE >= '01-MAR-19' and CA.CALENDAR_DATE <= '30-SEP-19');

Result

(HOST_ID; HOST_URL; HOST_NAME; HOST_SINCE; HOST_THUMBNAIL_URL)

71615 https://www.airbnb.com/users/show/71615 Mireia And Maria 19-GEN-10 https://a0.muscache.com/im/users/71615/profile_pic/1426612511/original.jpg?aki_policy=profile_small

82522 https://www.airbnb.com/users/show/82522 Meritxell 18-FEB-10 https://a0.muscache.com/im/pictures/ece65ffd-a798-4209-b1b0-a51060412b29.jpg?aki_policy=profile_small

108310 https://www.airbnb.com/users/show/108310 Pedro 14-APR-10 https://a0.muscache.com/im/pictures/user/7f7e9c1a-7274-4e90-a797-f079ffd9a9a3.jpg?aki_policy=profile_small

134698 https://www.airbnb.com/users/show/134698 Svetlana 29-MAG-10 https://a0.muscache.com/im/users/134698/profile_pic/1334849467/original.jpg?aki_policy=profile_small

136853 https://www.airbnb.com/users/show/136853 Fidelio 02-GIU-10 https://a0.muscache.com/im/users/136853/profile_pic/1312382561/original.jpg?aki_policy=profile_small

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:

We use COUNT to determine how many different listing with (IN) different host_id having the same host_name exist.

SQL statement

```
select COUNT(*)

from LISTING L, HOST H
```

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


```
where L.HOST_ID = H.HOST_ID and H.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
                );
```

Result
Only one row with value 30343

Query 5:

Print all the dates that 'Viajes Eco' has available accommodations for rent.

Description of logic:
We take the calendar_date of all the listings with availability value ’t’ (true) and host_name ‘Viajes Eco’.

SQL statement

```
select CA.CALENDAR_DATE
from CALENDAR CA, LISTING L
where CA.LISTING_ID = L.LISTING_ID and CA.AVAILABLE = 't' and L.LISTING_ID IN (select L1.LISTING_ID
                                                                               from LISTING L1, HOST H
                                                                               where L1.HOST_ID = H.HOST_ID and H.HOST_NAME = 'Viajes Eco'
                                                                               );
```

Result
10-NOV-18

11-NOV-18

12-NOV-18

13-NOV-18

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

14-NOV-18

Query 6:

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

Description of logic:

We take only the ones that are in the group of hosts with only one different host_id per listing.

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

                   );
```

Result

(HOST_ID; HOST_NAME)

108310	Pedro
73163	Andres
158596	Ester
90417	Etain
280070	Cristina

Query 7:

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

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 of logic:
We created a view named “wifi” (with all the listings with ‘WiFi’ in their amenities) and then used it in the SQL statement:

```
create view wifi as
  select LA.LISTING_ID
  from AMENITIES AM, LISTING_AMENITIES LA
  where AM.AMENITY_ID = LA.AMENITY_ID and AM.AMENITY_NAME = 'Wifi';
```

SQL statement

```
select AVG(CD1.PRICE) - AVG(CD2.PRICE)
from COSTS_DETAILS CD1, COSTS_DETAILS CD2
where CD1.LISTING_ID in (select * from wifi)
and CD2.LISTING_ID not in (select * from wifi);
```

Result
Only one row with value 6,661741644966838826626767756693377835 97

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:
We take the subtraction of two average prices: the first one is from the listings with 8 beds in their material_description and Berlin as their city; the second one is from the listings with 8 beds in their material_description and Madrid as their city.

SQL statement

```
select AVG(CD1.PRICE) - AVG(CD2.PRICE)
from COSTS_DETAILS CD1, COSTS_DETAILS CD2
where CD1.LISTING_ID IN (select MD.LISTING_ID
                         from MATERIAL_DESCRIPTION MD
```

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


```
                where MD.BEDS = 8)

and CD1.LISTING_ID IN (select L.LISTING_ID
                from LISTING L
                where L.CITY = 'Berlin')

and CD2.LISTING_ID not in (select MD.LISTING_ID
                  from MATERIAL_DESCRIPTION MD
                  where MD.BEDS = 8)

and CD2.LISTING_ID IN (select L.LISTING_ID
                from LISTING L
                where L.CITY = 'Madrid');
```

Result

Only one row with value 44,4658049044409025107111000628774474676 3

Query 9:

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

Description of logic:

We use “order by COUNT(*) DESC” to determine the hosts with the more listings in a descending order. We take only the listings with Spain as their country. We use “where rownum <= 10” to take only the first 10 rows of the result.

SQL statement

```
select * from

  (select H.HOST_ID , H.HOST_NAME
```

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


```
from LISTING L, HOST H

where L.COUNTRY = 'Spain' 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;
```

Result

(HOST_ID; HOST_NAME)

4459553	Eva&Jacques
99018982	Apartamentos
32046323	Juan
28038703	Luxury Rentals Madrid
1391607	Aline

Query 10:

Find the top-10 rated apartments in Barcelona.

Description of logic:

As for the previous query, we use “order by RS.REVIEW_SCORES_RATING DESC” to determine the rating scores in a descending order. We take only the listings with Barcelona as their city and ‘Apartment’ as their type. We use “where rownum <= 10” to take only the first 10 rows of the result.

SQL statement

```
select * from

(select L.LISTING_ID, L.LISTING_NAME

from LISTING L, REVIEWS_SCORES RS

where L.LISTING_ID = RS.LISTING_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/

```
  and L.LISTING_ID IN (select L1.LISTING_ID

                from LISTING L1, MATERIAL_DESCRIPTION MD

                where L.CITY = 'Barcelona' and MD.LISTING_ID = L.LISTING_ID and
MD.PROPERTY_TYPE = 'Apartment')

order by RS.REVIEW_SCORES_RATING DESC)

where rownum <=10;
```

Result

(LISTING_ID; LISTING_NAME)

475786	Room to rent in beautiful apartment
763465	Very Nice Room to rent in Raval
783032	room for rent 10 minutes from center
740113	Sunny, authentic Sant Antoni Apartment
721510	32 Valencia Apartment 2 bedrooms

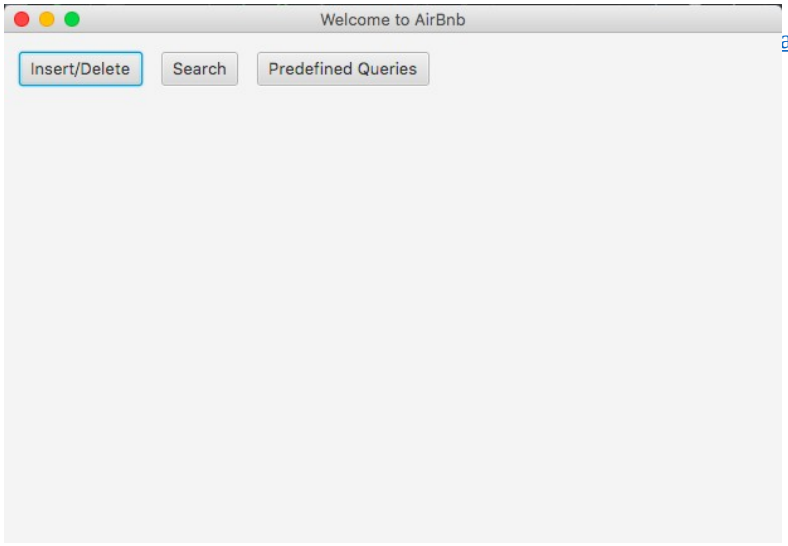
Interface

Design logic Description

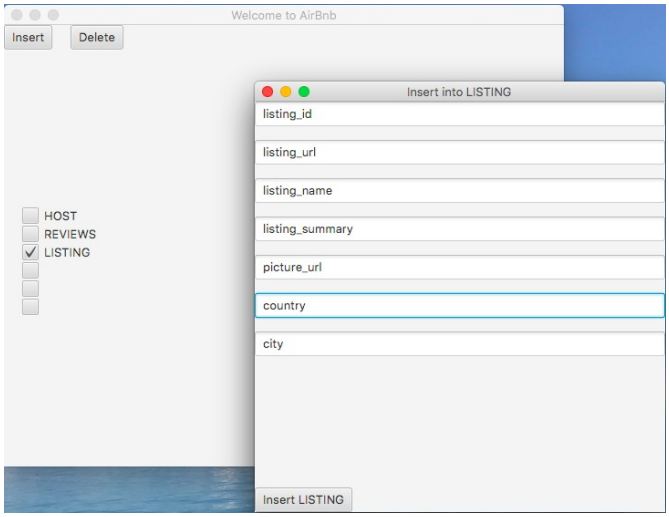
We decided to use JavaFX.

The interface is simple with intuitive buttons: it’s possible to insert/delete data by modifying the attributes of the item; there is a ‘Search’ button to search for a key-word in the database; the 10 queries described above are added as ‘Predefined queries’, so that the program can give the results to the user without showing any SQL language.

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

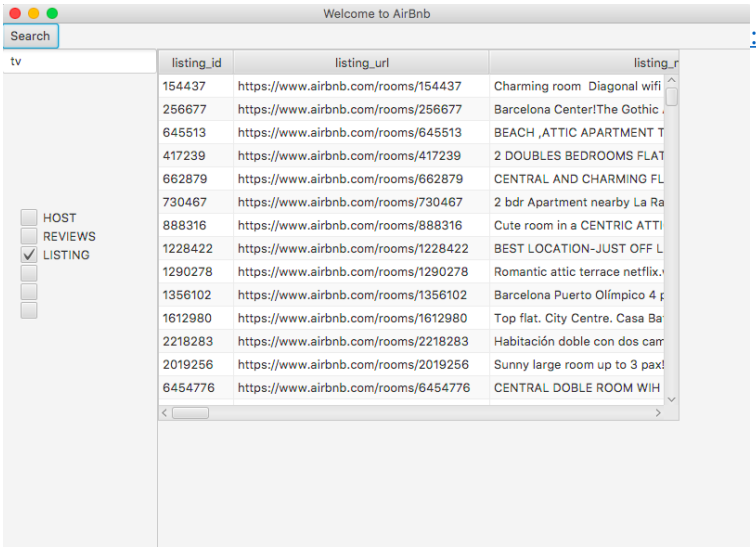


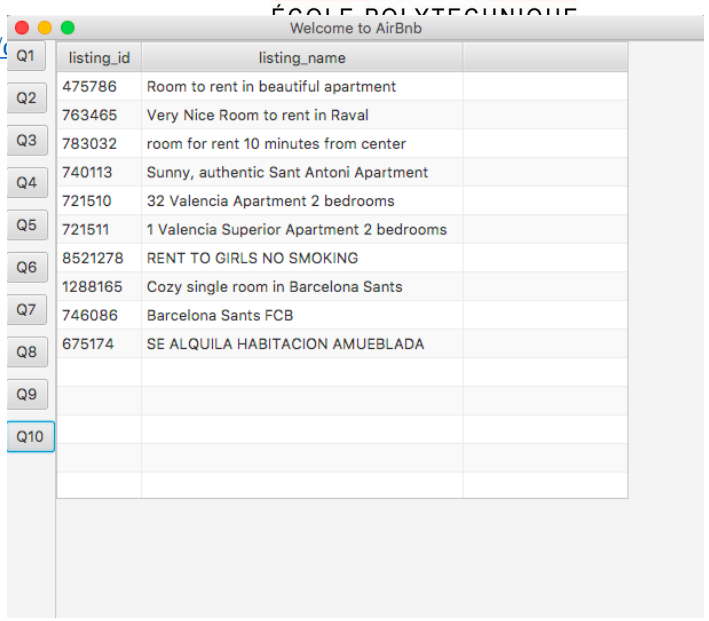
as.epfl.ch/



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





General Comments

We split the work as followed:

- Data insertion in the DB: Simon and Hédi
- Queries writing: Simon and Hédi
- Queries test and corrections: Camilla
- User interface: Hédi
- Report: Camilla

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/

Deliverable 3

Assumptions

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

Query Implementation

<For each query>

Query a:

Description of logic:

<What does the query do and how do I decide to solve it>

SQL statement

<The SQL statement>

Query Analysis

Selected Queries (and why)

Query 1

<Initial Running time:

Optimized Running time:

Explain the improvement:

Initial plan

Improved plan>

Query 2

<Initial Running time:

Optimized Running time:

Explain the improvement:

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/

Initial plan

Improved plan>

Query 3

<Initial Running time:

Optimized Running time:

Explain the improvement:

Initial plan

Improved plan>

Interface

Design logic Description

<Describe the general logic of your design as well as the technology you decided to use>

Screenshots

<Provide some initial screen shots of your interface>

General Comments

<In this section write general comments about your deliverable (comments and work allocation between team members>