

Databases Project – Spring 2019

Team No: 26

Names: Camilla Giaccari, Hédi Sassi, Simon Perriard

Contents

Contents	1
Deliverable 1	4
Assumptions	4
Entity Relationship Schema	4
Schema	4
Description	4
Relational Schema	5
ER schema to Relational schema	5
DDL	5
General Comments	6
Deliverable 2	7
Assumptions	7
Data Loading	7
Query Implementation	7
Query 1:	7
Description of logic:	7
SQL statement	7
Result	7
Query 2:	8
Description of logic:	8
SQL statement	8
Result	8

Query 3: 8

Description of logic: 8

SQL statement 8

Result 9

Query 4: 9

Description of logic: 9

SQL statement 9

Result 10

Query 5: 10

Description of logic: 10

SQL statement 10

Result 10

Query 6: 11

Description of logic: 11

SQL statement 11

Result 11

Query 7: 11

Description of logic: 11

SQL statement 12

Result 12

Query 8: 12

Description of logic: 12

SQL statement 12

Result 13

Query 9: 13

Description of logic: 13

SQL statement 13

Result 14

Query 10:	14
Description of logic:	14
SQL statement	14
Result	15
Interface	15
Design logic Description	15
Screenshots	15
General Comments	16
Deliverable 3	17
Assumptions	17
Query Implementation	17
Query a:	17
Description of logic:	17
SQL statement	17
Query Analysis	17
Selected Queries (and why)	17
Query 1	17
Query 2	17
Query 3	18
Interface	18
Design logic Description	18
Screenshots	18
General Comments	18

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

General Comments

We split the work as followed:

- ER model: Camilla
- Relational Model : Simon
- Data verification (scala program on the repo) : Hédi

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);
```

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 (\geq) 03.2019 and prior (\leq) 09.2019.

SQL statement

```
select *
from HOST H
where H.HOST_ID IN (select L.HOST_ID
                    from LISTING L, CALENDAR CA
                    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' );
```


URL: <http://dias.epfl.ch/>

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?](https://a0.muscache.com/im/users/71615/profile_pic/1426612511/original.jpg?aki_policy=profile_small)
aki_policy=profile_small

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

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

134698 <https://www.airbnb.com/users/show/134698> Svetlana 29-MAG-10 [https://](https://a0.muscache.com/im/users/134698/profile_pic/1334849467/original.jpg?aki_policy=profile_small)
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-GLU-10 [https://](https://a0.muscache.com/im/users/136853/profile_pic/1312382561/original.jpg?aki_policy=profile_small)
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
```

```
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
```

URL: <http://dias.epfl.ch/>

);

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

14-NOV-18

URL: <http://dias.epfl.ch/>

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?

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

URL: <http://dias.epfl.ch/>

```
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 6,66174164496683882662676775669337783597

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  
                        where MD.BEDS = 8)  
  
and CD1.LISTING_ID IN (select L.LISTING_ID
```

URL: <http://dias.epfl.ch/>

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 44,46580490444090251071110006287744746763

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

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

URL: <http://dias.epfl.ch/>

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

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')
```

URL: <http://dias.epfl.ch/>

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

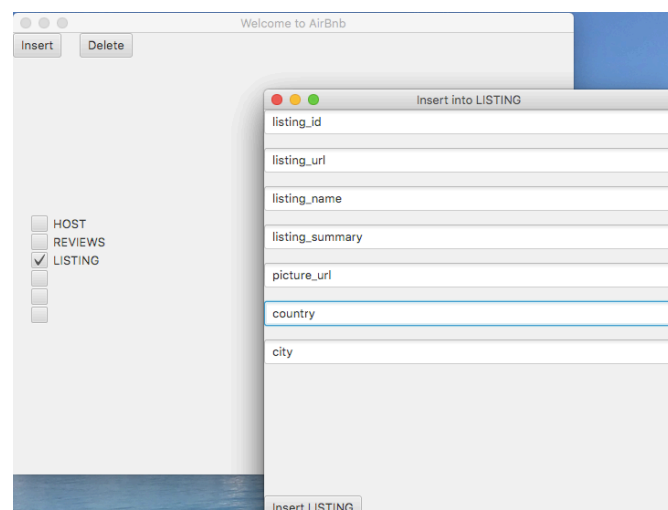
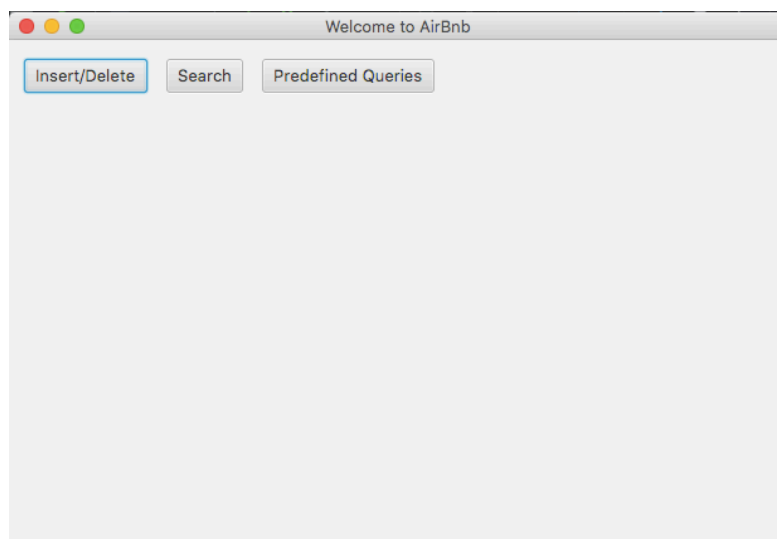
Interface

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.

Screenshots



URL: <http://dias.epfl.ch/>

Welcome to AirBnb

Search

tv

HOST
REVIEWS
☒ LISTING

listing_id	listing_url	listing_r
154437	https://www.airbnb.com/rooms/154437	Charming room Diagonal wifi
256677	https://www.airbnb.com/rooms/256677	Barcelona Center!The Gothic
645513	https://www.airbnb.com/rooms/645513	BEACH ,ATTIC APARTMENT T
417239	https://www.airbnb.com/rooms/417239	2 DOUBLES BEDROOMS FLAT
662879	https://www.airbnb.com/rooms/662879	CENTRAL AND CHARMING FL
730467	https://www.airbnb.com/rooms/730467	2 bdr Apartment nearby La Ra
888316	https://www.airbnb.com/rooms/888316	Cute room in a CENTRIC ATT
1228422	https://www.airbnb.com/rooms/1228422	BEST LOCATION-JUST OFF L
1290278	https://www.airbnb.com/rooms/1290278	Romantic attic terrace netflix
1356102	https://www.airbnb.com/rooms/1356102	Barcelona Puerto Olímpico 4 p
1612980	https://www.airbnb.com/rooms/1612980	Top flat. City Centre. Casa Ba
2218283	https://www.airbnb.com/rooms/2218283	Habitación doble con dos carr
2019256	https://www.airbnb.com/rooms/2019256	Sunny large room up to 3 paxl
6454776	https://www.airbnb.com/rooms/6454776	CENTRAL DOBLE ROOM WIH

Welcome to AirBnb

	listing_id	listing_name
Q1		
Q2	475786	Room to rent in beautiful apartment
	763465	Very Nice Room to rent in Raval
Q3	783032	room for rent 10 minutes from center
Q4	740113	Sunny, authentic Sant Antoni Apartment
	721510	32 Valencia Apartment 2 bedrooms
Q5	721511	1 Valencia Superior Apartment 2 bedrooms
Q6	8521278	RENT TO GIRLS NO SMOKING
	1288165	Cozy single room in Barcelona Sants
Q7	746086	Barcelona Sants FCB
Q8	675174	SE ALQUILA HABITACION AMUEBLADA
Q9		
Q10		

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

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:

Initial plan

URL: <http://dias.epfl.ch/>

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>