

Databases Project – Spring 2019

Team No: 10

Names: Andrea Veneziano, Rafael Bischof, Ludo Hoffstetter

Contents

Contents	1
Deliverable 1	2
Assumptions	2
Entity Relationship Schema	2
Schema	2
Description	2
Relational Schema	2
ER schema to Relational schema	2
DDL 3	
General Comments	3
Deliverable 2	4
Assumptions	4
Data Loading	4
Query Implementation	4
Query a:	4
Description of logic:	4
SQL statement	4
Interface	4
Design logic Description	4
Screenshots	4
General Comments	4

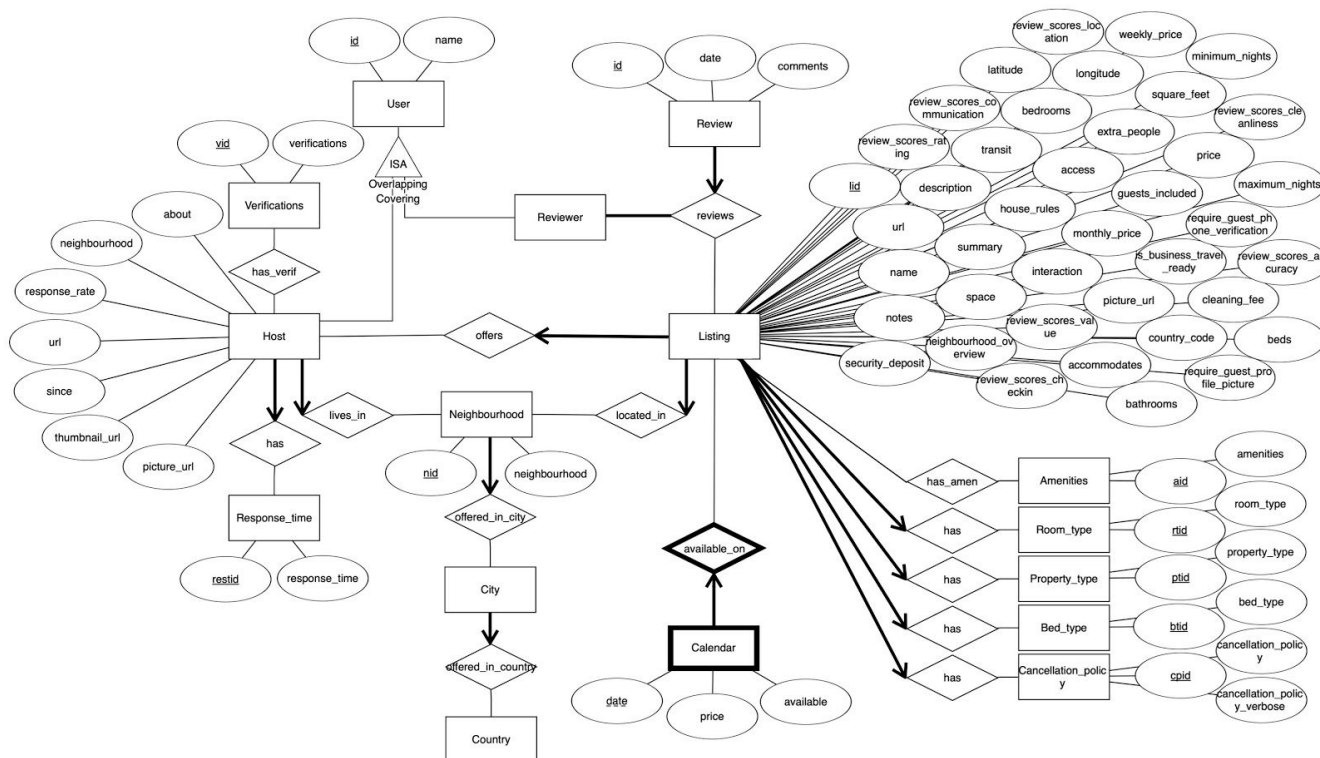
Deliverable 3	5
Assumptions	5
Query Implementation	5
Query a:	5
Description of logic:	5
SQL statement	5
Query Analysis	5
Selected Queries (and why)	5
Query 1	5
Query 2	5
Query 3	5
Interface	6
Design logic Description	6
Screenshots	6
General Comments	6

Deliverable 1

Assumptions

- Most of the accesses to *Listings* require all the attributes to be loaded.
- Listing, host, review and reviewer id's are unique across all cities.
- Reviews stay in the database even when the associated listing is deleted.
- Hosts stay in the database even when the associated listing is deleted or removed from the host.
- A Listing might not have an associated calendar entity.
- Since in our data all users are either a *Host* or a *Reviewer*, we decided not to implement a relation *Users*. The amount of stored data would stay the same, but the computational cost to retrieve the host's or the reviewer's names would be unnecessarily higher in a hierarchical model.
We are aware of the fact that, as soon as f.ex. *Clients* are added to the system, our model would have to be modified. But we assume that this will not be the case here.
- Neighbourhood information is the same for both Hosts and Listings.

Schema



- A *Listing* has exactly one *Host*.
- A *Reviewer*, by definition, reviews at least a *Listing* by writing at least a *Review*.
- A *Review* is written by exactly one *Reviewer* and concerns exactly one *Listing*.
- Entries in *Calendar* can only uniquely be identified with the help of a listing id. Therefore it is a weak entity.
- *Host* and *Reviewer* are subentities of *User*. They inherit the attributes *id* and *name*.
- The ISA relationship is covering because in our database a user has to be either a *Reviewer* or a *Host* or both.
- The ISA relationship allows overlapping, because a user can be at the same time a *Reviewer* and a *Host*.
- We normalized neighbourhood, country, city, property_type, room_type, bed_type and cancellation_policy attributes in separate entities and referenced them through an id. They all form a one-to-many relationship with *Listing*.
- Neighbourhood also forms a one-to-many relationship with *Host*.

- We also normalized in separate entities verifications and amenities attributes for Host and Listing, respectively. Each of them forms a many-to-many relationship.
- For simplicity, the fields (id, extracted_field) of every normalized entity were omitted on the schema.

Relational Schema

ER schema to Relational schema

- Host(hid: int, url: string, name: string, since: date, about: string, response_time: string, response_rate: float, thumbnail_url: string, picture_url: string, nid: int)
- Verifications(vid: int, verifications: string)
- Neighbourhood(nid: int, neighbourhood: string, ciid: int)
- Has_verif(vid: int, nid: int)
- Reviews(id: int, lid: int, rid: int, rdate: date, comments: string)
- Reviewer(rid: int, name: string)
- Calendar(date: cdate, lid: int, available: boolean, price: float)
- Country(coid: int, country: string, country_code: string)
- City(ciid: int, city: string, coid: int)
- Amenities(aid: int, amenities: string)
- Has_amen(aid: int, lid: int)
- Listing(lid: int, url: string, name: string, summary: string, space: string, description: string, neighbourhood_overview: string, notes: string, transit: string, laccess: string, interaction: string, house_rules: string, picture_url: string, nid: int, latitude: float, longitude: float, ptid: int, rtid: int, accommodates: int, bathrooms: float, bedrooms: int, beds: int, btid: int, square_feet: int, price: float, weekly_price: float, monthly_price: float, security_deposit: float, cleaning_fee: float, guests_included: int, extra_people: float, minimum_nights: int, maximum_nights: int, is_business_travel_ready: boolean, cpid: int, require_guest_profile_picture: boolean, require_guest_phone_verification: boolean, review_scores_rating: int, review_scores_accuracy: int, review_scores_cleanliness: int, review_scores_checkin: int, review_scores_communication: int, review_scores_location: int, review_scores_value: int, hid: int)
- Property_type(ptid: int, property_type: string)
- Room_type(rtid: int, room_type: string)
- Bed_type(btid: int, bed_type: string)
- Cancellation_policy(cpid: int, cancellation_policy: string)

DDL

CREATE TABLE Host

(
hid INT,

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/>

```
url VARCHAR(100) NOT NULL,  
name VARCHAR(100) NOT NULL,  
since DATE NOT NULL,  
about VARCHAR(1000) NOT NULL,  
response_time VARCHAR(30),  
response_rate FLOAT,  
thumbnail_url VARCHAR(100) NOT NULL,  
picture_url VARCHAR(100) NOT NULL,  
nid INT NOT NULL,  
PRIMARY KEY (hid),  
FOREIGN KEY(nid) REFERENCES Neighbourhood(nid)  
);
```

```
CREATE TABLE Verifications  
(  
  vid INT,  
  verifications VARCHAR(100) NOT NULL,  
  PRIMARY KEY (vid)  
);
```

```
CREATE TABLE Neighbourhood  
(  
  nid INT,  
  neighbourhood VARCHAR(200) NOT NULL,  
  ciid INT NOT NULL,  
  PRIMARY KEY (nid),  
  FOREIGN KEY (ciid) REFERENCES City(ciid)  
);
```

```
CREATE TABLE Has_verif  
(  
  vid INT,  
  hid INT,  
  PRIMARY KEY (vid, hid),  
  FOREIGN KEY (vid) REFERENCES Verifications(vid) ON DELETE CASCADE,  
  FOREIGN KEY (hid) REFERENCES Host(hid) ON DELETE CASCADE  
);
```

```
CREATE TABLE Response_time  
(  
  restid INT,
```

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/>

```
    response_time VARCHAR(100) NOT NULL,  
    PRIMARY KEY (restid)  
);
```

```
CREATE TABLE Reviews  
(  
    lid INT NOT NULL,  
    id INT,  
    rdate DATE NOT NULL,  
    rid INT NOT NULL,  
    comments TEXT NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (lid) REFERENCES Listing(lid),  
    FOREIGN KEY (rid) REFERENCES Reviewer(rid)  
);
```

```
CREATE TABLE Reviewer  
(  
    rid INT,  
    name VARCHAR(100) NOT NULL,  
    PRIMARY KEY (rid)  
);
```

```
CREATE TABLE Calendar  
(  
    lid INT,  
    cdate DATE,  
    available CHAR(1) NOT NULL,  
    price FLOAT NOT NULL,  
    PRIMARY KEY (cdate, lid),  
    FOREIGN KEY (lid) REFERENCES Listing ON DELETE CASCADE  
);
```

```
CREATE TABLE Country  
(  
    coid INT,  
    country VARCHAR(100) NOT NULL,  
    country_code VARCHAR(2) NOT NULL,  
    PRIMARY KEY (coid)  
);
```

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/>

```
CREATE TABLE City
(
  ciid INT,
  city VARCHAR(100) NOT NULL,
  coid INT NOT NULL,
  PRIMARY KEY (ciid),
  FOREIGN KEY (coid) REFERENCES Country(coid)
);
```

```
CREATE TABLE Amenities
(
  aid INT,
  amenities VARCHAR(200) NOT NULL,
  PRIMARY KEY (aid)
);
```

```
CREATE TABLE Has_amen
(
  aid INT,
  lid INT,
  PRIMARY KEY (aid, lid),
  FOREIGN KEY (aid) REFERENCES Amenities(aid) ON DELETE CASCADE,
  FOREIGN KEY (lid) REFERENCES Listing(lid) ON DELETE CASCADE
);
```

```
CREATE TABLE Listing
(
  lid INT,
  url VARCHAR(100) NOT NULL,
  name VARCHAR(300) NOT NULL,
  summary TEXT,
  space TEXT,
  description TEXT,
  neighbourhood_overview TEXT,
  notes TEXT,
  transit TEXT,
  laccess TEXT,
  interaction TEXT,
  house_rules TEXT,
  picture_url VARCHAR(100) NOT NULL,
  nid INT NOT NULL,
```

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/>

```
latitude FLOAT NOT NULL,
longitude FLOAT NOT NULL,
property_type_id INT NOT NULL,
room_type_id INT NOT NULL,
accommodates INT NOT NULL,
bathrooms FLOAT,
bedrooms INT,
beds INT,
bed_type_id INT NOT NULL,
square_feet INT,
price FLOAT NOT NULL,
weekly_price FLOAT,
monthly_price FLOAT,
security_deposit FLOAT,
cleaning_fee FLOAT,
guests_included INT NOT NULL,
extra_people FLOAT NOT NULL,
minimum_nights INT NOT NULL,
maximum_nights INT NOT NULL,
review_scores_rating INT,
review_scores_accuracy INT,
review_scores_cleanliness INT,
review_scores_checkin INT,
review_scores_communication INT,
review_scores_location INT,
review_scores_value INT,
is_business_travel_ready CHAR(1) NOT NULL,
cancellation_policy_id INT NOT NULL,
require_guest_profile_picture CHAR(1) NOT NULL,
require_guest_phone_verification CHAR(1) NOT NULL,
hid INT NOT NULL,
PRIMARY KEY (lid),
FOREIGN KEY (hid) REFERENCES Host ON DELETE CASCADE,
FOREIGN KEY (nid) REFERENCES Neighbourhood(nid),
FOREIGN KEY (property_type_id) REFERENCES Property_type(ptid),
FOREIGN KEY (room_type_id) REFERENCES Room_type(rtid),
FOREIGN KEY (bed_type_id) REFERENCES Bed_type(btid),
FOREIGN KEY (cancellation_policy_id) REFERENCES Cancellation_policy(cpid)
);
```


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/>

```
CREATE TABLE Property_type
```

```
(  
  ptid INT,  
  property_type VARCHAR(50) NOT NULL,  
  PRIMARY KEY (ptid)  
);
```

```
CREATE TABLE Room_type
```

```
(  
  rtid INT,  
  room_type VARCHAR(50) NOT NULL,  
  PRIMARY KEY (rtid)  
);
```

```
CREATE TABLE Bed_type
```

```
(  
  btid INT,  
  bed_type VARCHAR(50) NOT NULL,  
  PRIMARY KEY (btid)  
);
```

```
CREATE TABLE Cancellation_policy
```

```
(  
  cpid INT,  
  cancellation_policy VARCHAR(50) NOT NULL,  
  PRIMARY KEY (cpid)  
);
```

General Comments

Every team member worked equally.

Deliverable 2

Assumptions

Data Loading

We made several modifications to the .csv files in order to load and clean the data in the DBMS:

- We extracted several columns and used them to form new relations, such as Host.
- We parsed all the "repeating" string type attributes in the data in order to normalize.
- We changed the format of the dates.
- We changed the type of several attributes, such as percentages to floats.
- We standardized different string type attributes, such as city names.
- When normalizing Listing.city into the separate entity City (in relation with Neighbourhood), we filtered the different user typed cities according to their corresponding neighbourhoods.

Query Implementation

Query a:

Description of logic:

The query computes the average price for listings with 8 bedrooms.

SQL statement

```
SELECT AVG(L.price)
FROM Listing L
WHERE L.bedrooms = 8
```

Result

Average price = 313.15384615384613

Query b:

Description of logic:

The query computes the average cleaning review score for listings with TV.

SQL statement

```
SELECT AVG(L.review_scores_cleanliness)
FROM Listing L, Amenities A, Has_amen H
WHERE L.lid = H.lid AND H.aid = A.aid AND A.amenities = 'TV'
```

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*

Average score = 9.3981

Query c:

Description of logic:

The query prints all the hosts who have an available property between date 03.2019 and 09.2019.

SQL statement

SELECT DISTINCT H.hid

FROM Host H, Listing L, Calendar C

WHERE L.lid = C.lid AND L.hid = H.hid AND C.available = 't' AND C.cdate < "20190901" AND C.cdate >= "20190301"

Result

hid
382031
4369049
1227290
82522
108310

Query d:

Description of logic:

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

SQL statement

SELECT COUNT(DISTINCT H1.hid)

FROM Listing L1, Listing L2, Host H1, Host H2

WHERE L1.hid = H1.hid AND L2.hid = H2.hid AND H1.hid < H2.hid AND H1.name = H2.name

Result

Count = 18584

Query e:

Description of logic:

The query prints all the dates that host 'Viajes Eco' has available accommodations for rent.

SQL statement

SELECT DISTINCT C.cdate

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/>

FROM Calendar C, Host H, Listing L

WHERE L.hid = H.hid AND H.name = "Viajes Eco" AND L.lid = C.lid AND C.available = 't'

Result

cdate
2018-11-08
2018-11-09
2018-11-10
2018-11-11
2018-11-12

Query f:

Description of logic:

The query finds all the hosts (host_ids, host_names) that have only one listing.

SQL statement

SELECT H.hid, H.name

FROM Host H

WHERE 1 = (SELECT COUNT(*) FROM Listing L WHERE H.hid = L.hid)

Result

hid	name
3073	Ricard
3718	Britta
4108	Jana
5154	Raúl
11015	Josaiah

Query g:

Description of logic:

The query computes the difference in the average price of listings with and without Wifi.

SQL statement

SELECT AVG(T1.price) - AVG(L.price) AS DIFFERENCE

FROM (SELECT L.lid, L.price

FROM Listing L, Amenities A, Has_amen H

WHERE L.lid = H.lid AND H.aid = A.aid AND A.amenities = 'Wifi') T1, Listing L

WHERE L.lid NOT IN (SELECT T.lid FROM (SELECT L.lid, L.price

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/>

FROM Listing L, Amenities A, Has_amen H
WHERE L.lid = H.lid AND H.aid = A.aid AND A.amenities = 'Wifi') T)

Result

Difference = 3.213881387155041

Query h:

Description of logic:

The query computes how much more (or less) costly is to rent a room with 8 beds in Berlin compared to Madrid on average. The result is positive if Berlin is more expensive, negative otherwise.

SQL statement

```
SELECT AVG(L1.price) - AVG(L2.price) AS BerlinMinusMadrid
FROM Listing L1, Listing L2, City C1, City C2, Neighbourhood N1, Neighbourhood N2
WHERE L1.beds = 8 AND L2.beds = 8 AND L1.nid = N1.nid AND N1.ciid = C1.ciid AND L2.nid = N2.nid AND N2.ciid = C2.ciid AND C1.city = "Berlin" AND C2.city = "Madrid"
```

Result

Difference = -101.59261501210656

Query i:

Description of logic:

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

SQL statement

```
SELECT H.hid, H.name
FROM Host H, Listing L, Country Co, Neighbourhood N, City Ci
WHERE H.hid = L.hid AND L.nid = N.nid AND N.ciid = Ci.ciid AND Ci.coid = Co.coid AND Co.country = 'Spain'
GROUP BY H.hid
ORDER BY COUNT(*) DESC LIMIT 10
```

Result

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

Query I:

Description of logic:

The query finds the top-10 rated (review_score_rating) apartments (id,name) in Barcelona.

SQL statement

```
SELECT L.id, L.name
FROM Listing L, City C, Property_type T, Neighbourhood N
WHERE L.nid = N.nid AND N.ciid = C.ciid AND C.city = "Barcelona" AND L.ptid = T.ptid AND T.property_type =
'apartment'
ORDER BY L.review_scores_rating DESC LIMIT 10
```

Result

id	name
71520	Charming apartment with fantastic views!
179488	Room for rent in BCN/ non smoker
190348	Apartment with large terrace
250016	Excellent - 3 bedrooms, 2 bathrooms - Gothic
282679	Charming Penthouse, two amazing terraces

Interface

Design logic Description

(We are currently experiencing issues with our hosting provider, see general comments) We are creating a website which uses a mysql database. We created 3 pages (listings, hosts, reviews) and display the most relevant data to the user. Once the user clicks on a listing/host/review, a new page opens with all the data associated with the element.

The search function implements a full text search on everything that is text inside a table and then sorts the results by relevance and highlights all the searched words. We are currently not able to search in multiple tables at the same time. This still needs to be implemented.

Screenshots

http://web13.login-107.hoststar.ch/IDBS_project/listings/

General Comments

We are experiencing major performance issues on the host's phpMyAdmin site. Not because our queries run for too long, but because there must be an issue on the host's side. We are currently not able to display the content of certain tables and operations like INSERT or DELETE (even single instructions, not imports or deletions of millions of entries) won't execute. This prevented us from filling all the tables necessary to get the results to our queries. We are in touch with the support to figure out a solution and will catch up the lost time once we get the site running properly. (UPDATE the host set up a new server with SSD storage for us and we are refilling the new database at the moment)

After reading the review of the second deliverable, we changed a bit the ER model by normalizing City and Country from Listing to Neighbourhood and also Response_time from Host.

Every team member worked equally.

Deliverable 3

Assumptions

We did not make any additional assumption this time.

Query Implementation

Query a:

Description of logic:

The query prints how many hosts in each city have declared the area of their property in square meters, sorting the output based on the city name in ascending order.

SQL statement

```
SELECT COUNT(L.hid) AS counts, C.city
FROM Listing L, City C, Neighbourhood N
WHERE L.square_feet IS NOT NULL AND L.nid = N.nid AND N.ciid = C.ciid
GROUP BY N.ciid
ORDER BY counts ASC
```

Result

counts	city
5	Hospitalet de Llobregat
335	Madrid
405	Berlin
463	Barcelona

Query b:

Description of logic:

The query finds the top-5 neighborhoods using median review scores (review_scores_rating) of listings in Madrid.

SQL statement

```
SET @row_number:=0;
SET @neighbourhood_id:="";

SELECT
    neighbourhood_id, AVG(b.review_scores_rating) AS median
```

```
FROM
  (SELECT
    @row_number:=CASE WHEN @neighbourhood_id = a.nid THEN @row_number + 1 ELSE 1 END AS
count_of_group,
    @neighbourhood_id:=a.nid AS neighbourhood_id,
    a.nid,
    a.review_scores_rating,
    (SELECT COUNT(*) FROM Neighbourhood N, City C, Listing L
      WHERE L.nid = N.nid AND N.ciid = C.ciid AND C.city = 'Madrid' AND N.nid = a.nid) AS total_of_group
FROM
  (SELECT
    N.nid, L.review_scores_rating
      FROM Neighbourhood N, City C, Listing L
      WHERE L.nid = N.nid AND N.ciid = C.ciid AND C.city = 'Madrid'
    ORDER BY N.nid , L.review_scores_rating) as a) AS b
WHERE
  count_of_group BETWEEN total_of_group / 2.0 AND total_of_group / 2.0 + 1
GROUP BY neighbourhood_id
ORDER BY AVG(b.review_scores_rating) DESC LIMIT 5
```

Result

neighbourhood_id	median
213	100.0000
188	98.0000
215	97.0000
196	96.0000
217	96.0000

Query c:

Description of logic:

The query find all the hosts (host_ids, host_names) with the highest number of listings.

SQL statement

```
SELECT H.hid, H.name
FROM Host H, Listing L
WHERE H.hid = L.hid
```


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 H.hid

ORDER BY COUNT(*) DESC

Result

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

Query d:

Description of logic:

The query finds the 5 most cheapest Apartments (based on average price within the available dates) in Berlin available for at least one day 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.

SQL statement

SELECT L.lid

FROM Listing L, Calendar Ca, Host H, City Ci, Cancellation_policy CP, Has_verif Hv, Verifications V,
Property_type T, Neighbourhood N

WHERE L.nid = N.nid AND N.ciid = Ci.ciid AND Ci.city = 'Berlin' AND L.beds >= 2

AND L.review_scores_location >= 8 AND L.cpid = CP.cpid

AND CP.cancellation_policy = 'flexible' AND Ca.lid = L.lid AND Ca.available = 't'

AND Ca.cdate >= "20190301" AND Ca.cdate <= "20190430"

AND L.hid = H.hid AND Hv.hid = H.hid AND Hv.vid = V.vid AND V.verifications = 'government_id' AND L.ptid =

T.ptid AND T.property_type = 'apartment'

GROUP BY L.lid

ORDER BY AVG(Ca.price) ASC LIMIT 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/>*Result*

lid
1490274
24043706
1368460
6691656
7071541

Query e:

Description of logic:

The query finds 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.

This query was made by chaining the follow SQL statement 16 times, i.e with all the possible values of L.accommodates. Since the performance of this query is surprisingly good, we assume that SQL is smart enough to optimize it.

SQL statement

```
(SELECT L.accommodates, L.lid, L.review_scores_rating
FROM Listing L
WHERE L.accommodates = 1 AND 2 <=
      (SELECT COUNT(A.aid)
      FROM Amenities A, Has_amen Ha
      WHERE Ha.lid = L.lid AND Ha.aid = A.aid
      AND (A.amenities = "Wifi" OR A.amenities = "Internet" OR A.amenities = "TV" OR A.amenities = "Free
street parking"))
ORDER BY L.accommodates DESC, L.review_scores_rating DESC, L.lid ASC LIMIT 5)
UNION ALL
...
```

Result

accommodates	lid	review_scores_rating
1	109369	100
1	179488	100
1	240735	100
1	250121	100
1	287660	100
2	55455	100
2	116832	100
2	158890	100
2	190348	100
2	204617	100
3	24805	100
3	168615	100
3	248375	100
3	377304	100
3	429604	100
4	64590	100
4	74352	100
4	115576	100
4	186908	100
4	190448	100
5	207740	100
5	452164	100
5	494434	100
5	525162	100
5	573259	100
6	47938	100
6	250016	100
6	282679	100
6	674068	100
6	721378	100
7	9991	100
7	832755	100
7	2952788	100
7	3495408	100
7	3654742	100
8	230611	100
8	3108648	100
8	3108830	100
8	3598873	100
8	3740368	100
9	71520	100
9	751527	100
9	6970615	100
9	13651387	100
9	15472852	100
10	60161	100
10	1005399	100
10	9584066	100
10	16599521	100
10	17973836	100
11	11343804	99
11	16237757	99
11	16252509	99
11	13868134	98
11	16183489	98
12	9488554	100
12	9880406	100
12	10133140	100
12	10784053	100
12	12567906	100
13	23694336	100
13	22694811	99
13	25437138	99
13	6431465	95
13	22449147	95
14	19147659	100
14	21601526	100
14	23781693	100
14	9536912	99
14	1098894	98
15	22673624	98
15	7560506	96
15	9194127	95
15	14588206	95
15	354033	92
16	478616	100
16	8734083	100
16	11384240	100
16	16799799	100
16	22966047	100

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 f:

Description of logic:

The query finds the top three busiest listings per host. The more reviews a listing has, the busier the listing is.

SQL statement

```
SELECT L.lid
FROM Listing L, Reviews R
WHERE L.lid = R.lid
GROUP BY L.lid, L.hid
ORDER BY COUNT(L.lid) DESC LIMIT 3
```

Result

lid
186221
297066
674461

Query g:

Description of logic:

The query finds the three most frequently used amenities at each neighborhood in Berlin for the listings with "Private Room" room type.

SQL statement

```
SELECT DISTINCT N.neighbourhood,
(SELECT A.amenities FROM Listing L2, Has_amen H, Amenities A WHERE L2.nid = N.nid AND L2.lid = H.lid AND
A.aid = H.aid GROUP BY H.aid ORDER BY COUNT(H.aid) DESC LIMIT 1) AS Amenity1,
(SELECT A.amenities FROM Listing L2, Has_amen H, Amenities A WHERE L2.nid = N.nid AND L2.lid = H.lid AND
A.aid = H.aid GROUP BY H.aid ORDER BY COUNT(H.aid) DESC LIMIT 1 OFFSET 1) AS Amenity2,
(SELECT A.amenities FROM Listing L2, Has_amen H, Amenities A WHERE L2.nid = N.nid AND L2.lid = H.lid AND
A.aid = H.aid GROUP BY H.aid ORDER BY COUNT(H.aid) DESC LIMIT 1 OFFSET 2) AS Amenity3
FROM Neighbourhood N, Listing L, Room_type R, City C
WHERE N.nid = L.nid AND L.rtid = R.rtid AND R.room_type = "Private Room" AND C.ciid = N.ciid AND C.city =
"Berlin"
```

Result

neighbourhood	Amenity1	Amenity2	Amenity3
Friedrichshain	Wifi	Kitchen	Heating
Charlottenburg	Wifi	Heating	Kitchen
Kreuzberg	Wifi	Kitchen	Heating
Prenzlauer Berg	Kitchen	Wifi	Heating
Alt-Hohenschönhausen	Heating	Wifi	Kitchen
Moabit	Wifi	Kitchen	Heating
Schöneberg	Wifi	Heating	Kitchen
Wedding	Kitchen	Wifi	Heating
Mitte	Wifi	Heating	Essentials
Neukölln	Kitchen	Wifi	Heating
Tempelhof	Kitchen	Heating	Wifi
Pankow	Heating	Kitchen	Wifi
Wilmerdorf	Heating	Wifi	Kitchen
Alt-Treptow	Wifi	Kitchen	Heating
Steglitz	Heating	Heating	Kitchen
Potsdamer Platz	Essentials	Heating	Kitchen
Schmargendorf	Kitchen	Wifi	Essentials
Reinickendorf	Heating	Kitchen	Wifi
Karlshorst	Heating	Kitchen	Wifi
Köpenick	Heating	Wifi	Essentials
Westend	Wifi	Heating	Essentials
Altglienicke	Heating	Heating	Kitchen
Halensee	Wifi	Heating	Kitchen
Lichterfelde	Wifi	Heating	Kitchen
Zehlendorf	Wifi	Heating	Kitchen
Mariendorf	Kitchen	Wifi	Essentials
Lichtenberg	Heating	Heating	Wifi
Wittenau	Heating	Heating	Heating
Friedrichsfelde	Wifi	Wifi	Wifi
Marzahn	Heating	Essentials	Kitchen

Query h:

Description of logic:

The query computes 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 verifications.

SQL statement

```
SELECT AVG(L1.review_scores_communication) - AVG(L2.review_scores_communication)
FROM Listing L1, Listing L2, Host H1, Host H2
```

```
WHERE L1.hid = H1.hid AND L2.hid = H2.hid AND H1.hid =  
(SELECT H.hid FROM Host H, Has_verif V WHERE H.hid = V.hid GROUP BY H.hid ORDER BY COUNT(*) DESC LIMIT  
1) AND H2.hid =  
(SELECT H.hid FROM Host H, Has_verif V WHERE H.hid = V.hid GROUP BY H.hid ORDER BY COUNT(*) ASC LIMIT  
1)
```

Result

Difference = 1.0000

Query i:

Description of logic:

The query finds the city who has the highest number of reviews for the room types whose average number of accommodates are greater than 3.

SQL statement

```
SELECT C.city  
FROM Listing L, City C, Reviews R, Neighbourhood N  
WHERE L.nid = N.nid AND C.ciid = N.ciid AND L.lid = R.lid AND L.rtid IN  
(SELECT T.rtid FROM (SELECT L1.rtid, AVG(L1.accommodates) AS average FROM Listing L1 GROUP BY L1.rtid) T  
WHERE T.average > 3)  
GROUP BY L.lid  
ORDER BY COUNT(R.lid) DESC LIMIT 1
```

Result

City = Barcelona

Query l:

Description of logic:

The query finds all the neighborhoods in Madrid which have at least 50 percent of their listings occupied in year 2019 and their host has joined airbnb before 01.06.2017.

SQL statement

```
SELECT DISTINCT N.nid, N.neighbourhood  
FROM Neighbourhood N, City Ci  
WHERE N.ciid = Ci.ciid AND Ci.city = 'Madrid' AND  
(SELECT COUNT(DISTINCT L.lid)  
FROM Calendar Ca, Host H, Listing L  
WHERE L.nid = N.nid AND Ca.lid = L.lid AND L.hid = H.hid AND H.since < '20170601' AND Ca.available = 'f' AND  
Ca.cdate >= '20190101' AND Ca.cdate <= '20191231') >=  
(SELECT COUNT(DISTINCT L.lid) * 0.5  
FROM City Ci, Listing L
```

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/>



WHERE L.nid = N.nid)

Result

nid	neighbourhood
42	La Latina
67	Sol
71	Aluche
72	Palos do Moguer
74	Castellana
75	Almagro
76	Embajadores
77	Trafalgar
165	Jerónimos
166	Delicias
169	Malasaña
170	Justicia
171	Legazpi
172	San Blas
173	Palacio
174	Cortes
175	Recoletos
176	Cuatro Caminos
177	Ciudad Lineal
178	Arapiles
179	Fuencarral-el Pardo
180	Villaverde
181	Carabanchel
182	Hortaleza
183	Pacífico
184	Ciudad Jardin
185	El Tréntaiseis
186	Prosperidad
187	Acacias
188	Estrella

Query m:

Description of logic:

The query finds all the countries that in 2018 had at least 20% of their listings available.

SQL statement

```
SELECT DISTINCT Co.coid, Co.country
FROM Country Co, Listing L, Neighbourhood N, City C
WHERE
```

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/>

```
(SELECT COUNT(L.lid)
FROM Calendar Ca
WHERE L.nid = N.nid AND N.ciid = C.ciid AND Co.coid = C.coid AND Ca.lid = L.lid AND Ca.available = 't' AND
Ca.cdate >= '20180101' AND Ca.cdate <= '20181231') >=
(SELECT COUNT(L.lid) * 20 / 100
WHERE L.nid = N.nid AND N.ciid = C.ciid AND Co.coid = C.coid)
```

Result

coid	country
2	Germany
1	Spain

Query n:

Description of logic:

The query finds all the neighbourhoods in Barcelona where more than 5 percent of their accommodation's cancellation policy is strict with grace period.

SQL statement

```
SELECT N.nid, N.neighbourhood
FROM Neighbourhood N
WHERE
  (SELECT COUNT(N.nid)
   FROM Listing L, City C, Cancellation_policy CP
   WHERE L.nid = N.nid AND N.ciid = C.ciid AND C.city = 'Barcelona' AND L.cpid = CP.cpid AND
   CP.cancellation_policy = 'strict_14_with_grace_period') > (SELECT COUNT(N.nid) * 5 / 100 FROM Listing L, City
C
WHERE L.nid = N.nid AND N.ciid = C.ciid AND C.city = 'Barcelona')
```


Result

nid	neighbourhood
1	El Camp de l'Arpa del Clot
2	El Poblenou
3	Vila de Gràcia
4	El Baix Guinardó
5	La Nova Esquerra de l'Eixample
6	El Raval
7	Camp d'en Grassot i Gràcia Nova
8	El Gòtic
9	L'Antiga Esquerra de l'Eixample
10	La Barceloneta
11	Dreta de l'Eixample
12	El Besòs i el Maresme
13	el Fort Pienc
14	El Born
15	Diagonal Mar - La Mar Bella
16	Vallcarca i els Penitents
17	El Poble-sec
18	Sant Antoni
19	El Clot
20	Sants-Montjuïc
21	Pedralbes
22	Guinardó
23	Sant Andreu de Palomar
24	Provençals del Poblenou
25	Horta
26	La Salut
27	Sant Pere/Santa Caterina
28	Sant Gervasi - Galvany
29	Sant Martí de Provençals
30	La Sagrada Família
31	Carmel
32	El Coll
33	Glòries - El Parc
34	La Vila Olímpica
35	Vilapicina i la Torre Llobeta
37	El Putget i Farró
38	Eixample
39	Sarrià
40	El Congrés i els Indians
41	Porta
43	Torre Baró
44	Les Tres Torres
45	Sant Genís dels Agudells
46	La Font d'en Fargues
47	Navas
48	Les Corts
50	La Sagrera
52	La Teixonera
53	Gràcia
54	La Verneda i La Pau
55	La Vall d'Hebron
56	Sant Gervasi - la Bonanova
57	La Prosperitat
58	Nou Barris
59	Can Baró
60	Turó de la Peira - Can Peguera
62	Verdum - Los Roquetes
63	La Trinitat Vella
64	El Bon Pastor
66	Trinitat Nova
70	Sant Martí
73	Ciutat Vella
223	Horta-Guinardó

Query Analysis

Selected Queries (and why)

We chose queries b, l and m because they did not use only primary keys in their execution plans and they seemed to have room for 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/>



Query 1 (query b)

Initial Running time: 1.7464 s

Optimized Running time: 1.6486 s

Explain the improvement: we added 2 indices.

Initial plan: uses scan on C.city, block nested loop on N.ciid.

Improved plan: uses index on C.city, index on N.ciid.

Query 2 (query l)

Initial Running time: 13.7991 s

Optimized Running time: 2.6243 s

Explain the improvement: we added 3 indices.

Initial plan: uses scan on C.ciid, block nested loop on C.coid, scan on N.ciid.

Improved plan: uses index on C.ciid, index on C.coid, index on N.ciid

Query 3 (query m)

Initial Running time: 0.0162 s

Optimized Running time: 0.0116 s

Explain the improvement: We added 3 indices.

Initial plan: uses scan on Co.country, block nested loop on C.coid, block nested loop on N.ciid.




Improved plan: uses index on Co.country, index on C.coid, index on N.ciid.




Interface



Design logic Description



Our interface consists of a website based on the CSM Wordpress, while the database is based on MySQL.


It has a page “Browse” which is built up similarly to Airbnb in the way that it has a search function and a filter and then displays all listings that match the requirements. When clicking on a listing, a detailed page of this listing opens up with nicely formatted data.









Berlin-Mitte Valuel Quiet courtyard/very central

 lan


60 \$ per day

Rated 93 out of 100

Great location! 30 of 75 sq meters. This wood floored/high ceiling typical Berlin "Altbau" section of an apartment consists of 1 simple large room, a small kitchen and a bathroom + shower. The ...




Mitte-Wedding big 3 Room Apt, U8!

 lan


83 \$ per day

Rated 96 out of 100

Located in Mitte-Wedding, cool Berlin feeling near Mitte-Wedding. Minutes from Alexanderplatz. A genuine Berlin area, 50 meters from "Uferhallen" dance and theatre center, Rosa Park's relocated house, Humboldthain Park, shopping, International ...




Mauerpark1 P-Berg under Roof = Quiet, Lively area

 lan

34 \$ per day

Rated 95 out of 100

Little („Einlieger“-)apartment under the roof! Sunny, quiet, facing a peaceful courtyard, top floor Berlin. Luxurious bathroom + a huge bathtub under a beautiful skylight! You are in the middle of ...




BerlinSpot Schöneberg near KaDeWe

Apartment, Private room

Summary

First of all: I prefer short-notice bookings. Requests 1-2 weeks before arrival are perfect. Longer plans in advance are sometimes a bit difficult for me. But please just try! My flat is in the middle of West-Berlin, directly in the mostly gay and very safe area of Nollendorplatz/"Kiez" with a lot of good restaurants, outdoor cafés, small shops, famous store KaDeWe and Gadschewski's, and a beautiful fountain set on a square.




Availability

November 2019

M	T	W	T	F	S	S
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

< previous next >

Book this listing

 Jana

Host on Airbnb since 2008




ENJOY EVERY DAY AS IF IT'S YOUR LAST!!!



Living in the middle ...



Location




Schöneberg, Berlin, Germany (DE)



Neighborhood overview



























































































































































Meet Ian on airbnb

Ian


Host since 18th of August 2008

Response rate: 96%, response time: within an hour


Believe in sharing economy.

Verifications: email, phone, reviews, jumio, offline_government_id, government_id

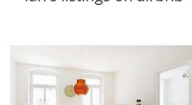
Ian's listings on airbnb




Berlin-Mitte Value! Quiet courtyard/very central



Ian




Mitte-Wedding big 3 Room Apt, U8!




Ian

83 \$ per day



Mauerpark! P-Berg under Roof = Quiet. Lively area



Ian

Host-page

The website also has a separate page for each table in our database, where all data in the table is shown and values can be inserted, updated or deleted.

Browse Preset Queries ▾ Tables ▾

Hosts

Host id

Fetch data of this host

Host data

Show 10 ▾ entries

Search:

hid ▴	url ▴	name ▴	since ▴	about ▴
2217	https://www.airbnb.com/users/show/2217	Ian	18 Aug 2008	Believe in sharing economy.
3073	https://www.airbnb.com/users/show/3073	Ricard	19 Sep 2008	I am a freelance professional and university professor about
3718	https://www.airbnb.com/users/show/3718	Britta	19 Oct 2008	We love to travel ourselves a lot and prefer to stay in apartme
4108	https://www.airbnb.com/users/show/4108	Jana	07 Nov 2008	ENJOY EVERY DAY AS IF IT'S YOUR LAST!!! Living in the middle
5154	https://www.airbnb.com/users/show/5154	Raül	14 Dec 2008	I'm a photographer based in Madrid. I love travel, art and leis
10704	https://www.airbnb.com/users/show/10704	Weflating	19 Mar 2009	Hola! Somos Weflating Apartments un equipo de gente joven
10966	https://www.airbnb.com/users/show/10966	Claudia	22 Mar 2009	Herzlich Willkommen in Berlin ! Ich freue mich darauf, auch ir
11015	https://www.airbnb.com/users/show/11015	Josaiah	22 Mar 2009	Simple guy with simple needs. Very friendly and no hangups

Furthermore, there is a separate page for almost all preset queries, where the results are output in a table. If the query contained variable requirements like “all listings with 8 bedrooms”, then these requirements can be

modified and the table is updated accordingly.

Browse Preset Queries Tables

The query finds all the neighbourhoods in Barcelona where more than 5 percent of their accommodation's cancellation policy is strict with grace period.

-- select city --

Percentage

-- select cancellation_policy --

Submit

Show 10 entries Search:

nid	neighbourhood
1	El Camp de l'Arpa del Clot
2	El Poblenou
3	Vila de Gràcia
4	El Baix Guinardó
5	La Nova Esquerra de l'Eixample
6	El Raval
7	Camp d'en Grassot i Gràcia Nova
8	El Gòtic
9	L'Antiga Esquerra de l'Eixample
10	La Barceloneta

nid neighbourhood

Showing 1 to 10 of 63 entries (filtered from 224 total entries)

<< < 1 2 3 4 5 6 7 > >>

The website can be found under <http://zwfzhoay.lx3.hoststar.website/>

Login: <http://zwfzhoay.lx3.hoststar.website/wp-login.php>

User: ElPrimo

PW: Vamonos1!

General Comments

Some queries were trickier than other to implement and required more SQL knowledge than what we had. For example, we used variables in some SQL statements to solve our issues.

Every team member worked equally.