

Databases Project – Spring 2019

Team No: 32

Names: Sophie Ammann, Samuel Chassot, Daniel Filipe Nunes Silva

Contents

Contents.....	1
Deliverable 1.....	2
Assumptions.....	2
Entity Relationship Schema.....	3
Schema Overview.....	5
Zoomed Schema.....	6
Relational Schema.....	10
DDL.....	11
General Comments.....	15

Deliverable 1

Assumptions

We use MySQL syntax for this project.

The weak entities (House_properties, Economic_properties, Administrative_properties and Review_scores) are linked to Listing and exist if and only if the listing exists: they have to be created/deleted when a Listing is created/deleted (ON DELETE CASCADE). The attributes of Review_scores can be *null* if the Listing is new and was not yet evaluated. Storing information of the listings grouped by theme in different tables should provide us some efficiency advantage for further queries.

Both Listing and Neighbourhood are linked to a city, therefore we have decided to relate a Listing to a Neighbourhood, that is itself linked to a City. We assumed it would avoid having the same information twice (*is in a City - relation*).

The Review is written by exactly one Reviewer and for exactly one Listing. If a Reviewer unsubscribes of the Airbnb platform, we have decided that the Review shall still exist. Also, if a Listing disappears, we suppose that the Review still exists.

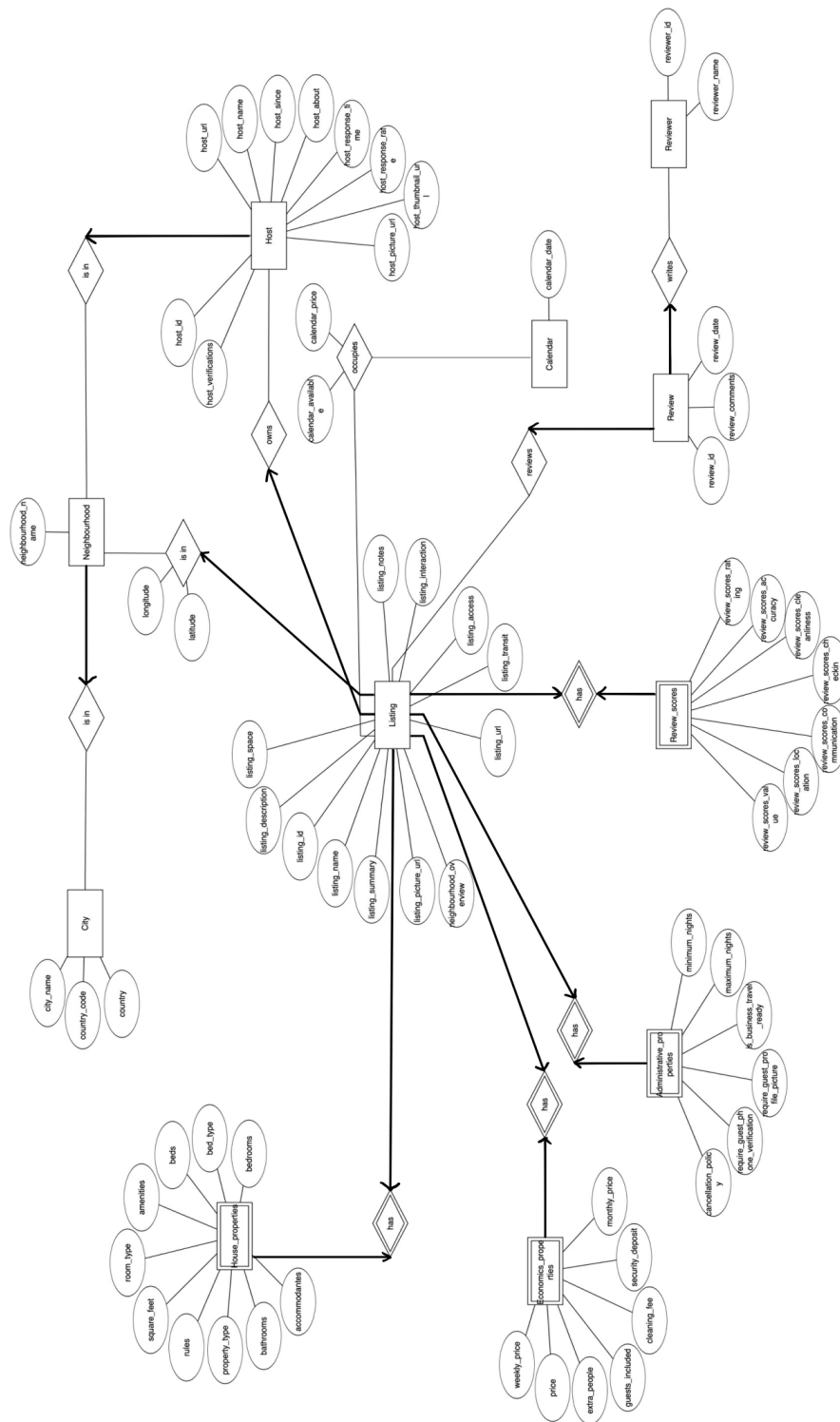
We suppose that City in this database has a unique name inside their country. Hence we can use city_name and country_code as primary key.

Entity Relationship Schema

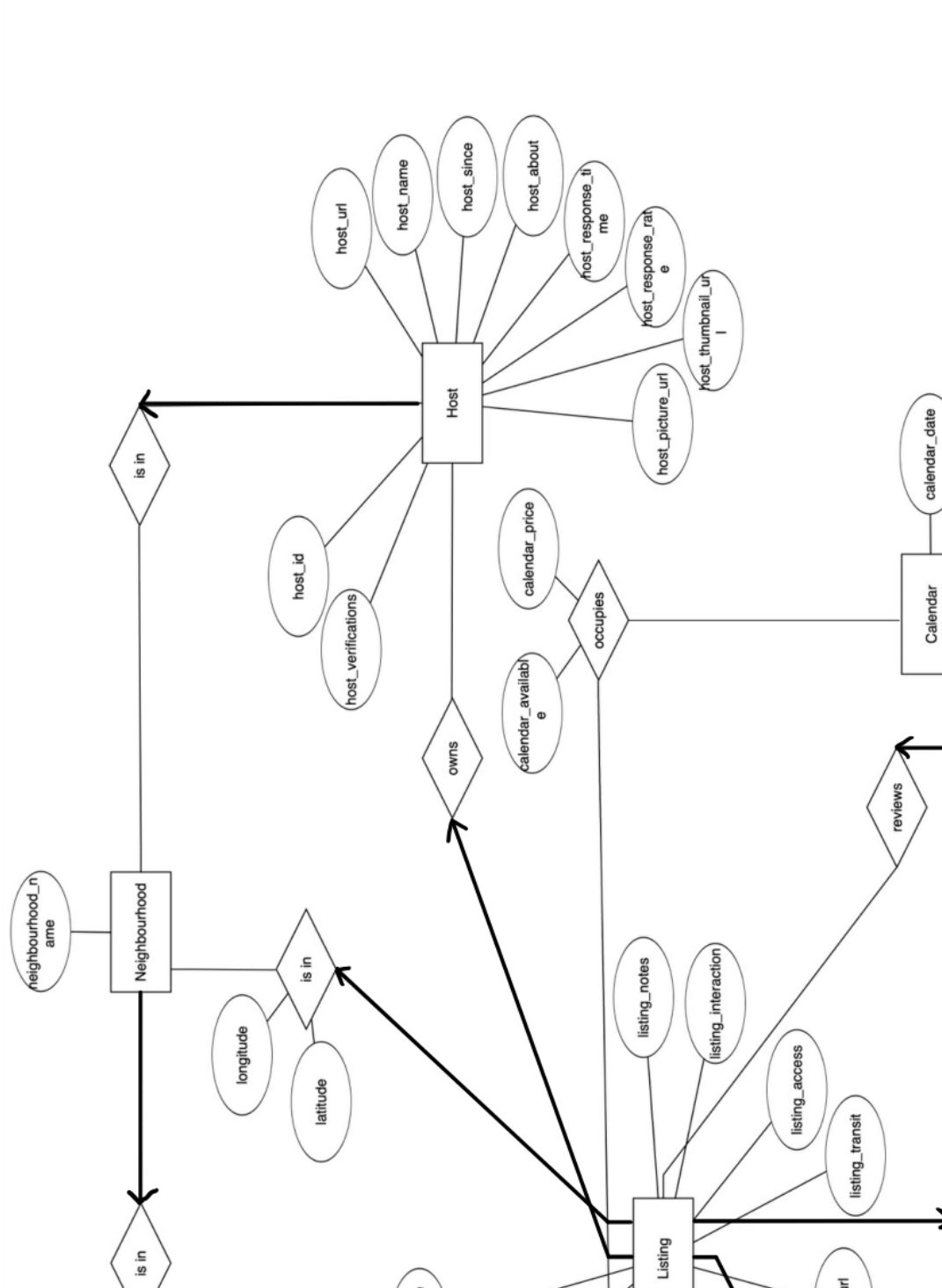
Entities	Description
Listing	Represents a listing in an AirBnb service.
Host	Person that hosts a listing.
Neighbourhood	Part of a city.
City	City.
House_properties	Properties of the accommodation.
Economic_properties	Costs related to the rent of the accommodation.
Administrative_properties	Rules related to the rent of the accommodation.
Review	Review in the Airbnb system of a listing.
Reviewer	Person who writes a review about a listing.
Review_scores	Scores related to a listing in different domains of reviewing.
Calendar	Availabilites of a listing.
Location	Location on a map of a listing.

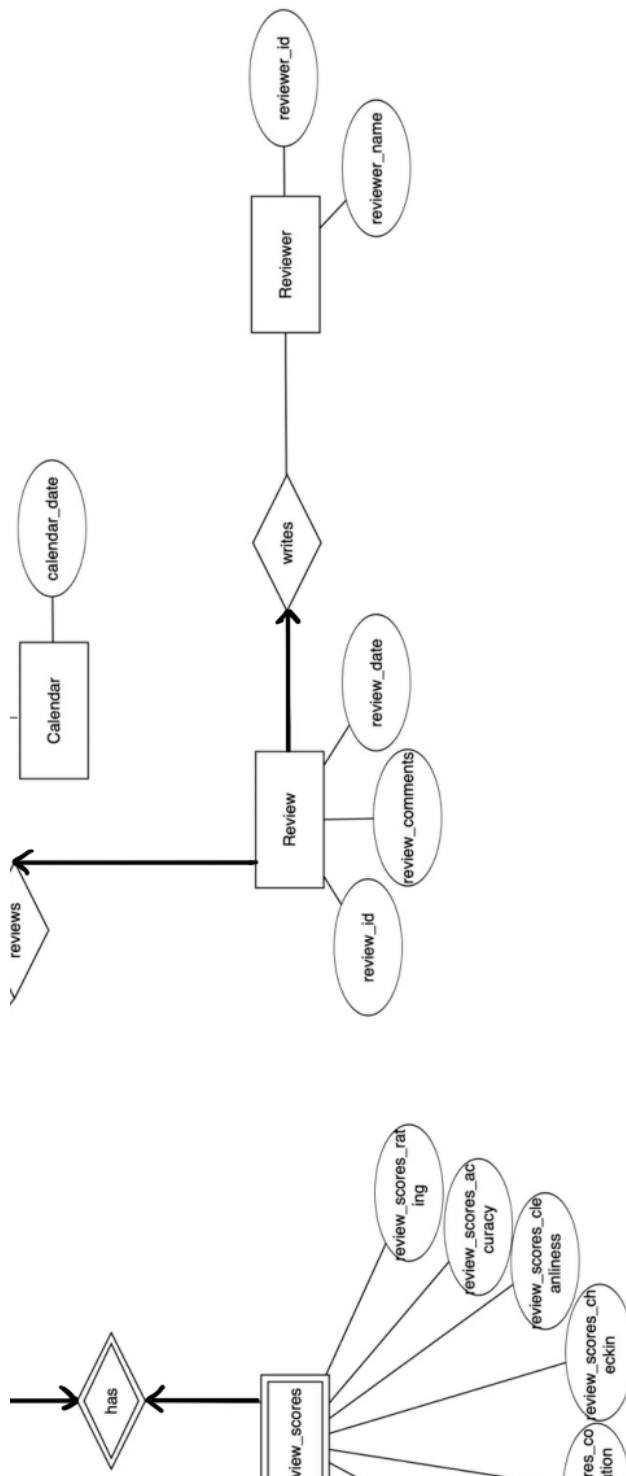
Entity	Relation	Entity	Constraints
a Listing	has	House_properties	one-to-one relationship (Listing's weak entity)
a Listing	has	Economic_properties	one-to-one relationship (Listing's weak entity)
a Listing	has	Administrative_properties	one-to-one relationship (Listing's weak entity)
a Listing	has	Review_scores	one-to-one relationship (Listing's weak entity)
a Host	owns	a Listing	each listing has exactly one host
a Listing	occupies	a Calendar	a listing has availabilities in time. Each date has the corresponding listing's availability
a Review	reviews	a Listing	a review reviews exactly one listing
a Reviewer	writes	a Review	a review has exactly one reviewer
a Listing	is in	a Neighbourhood	a listing is in exactly one neighbourhood
a Listing	is in	a City	a listing is in exactly one city
a Neighbourhood	is in	a City	a neighborhood is in exactly one city

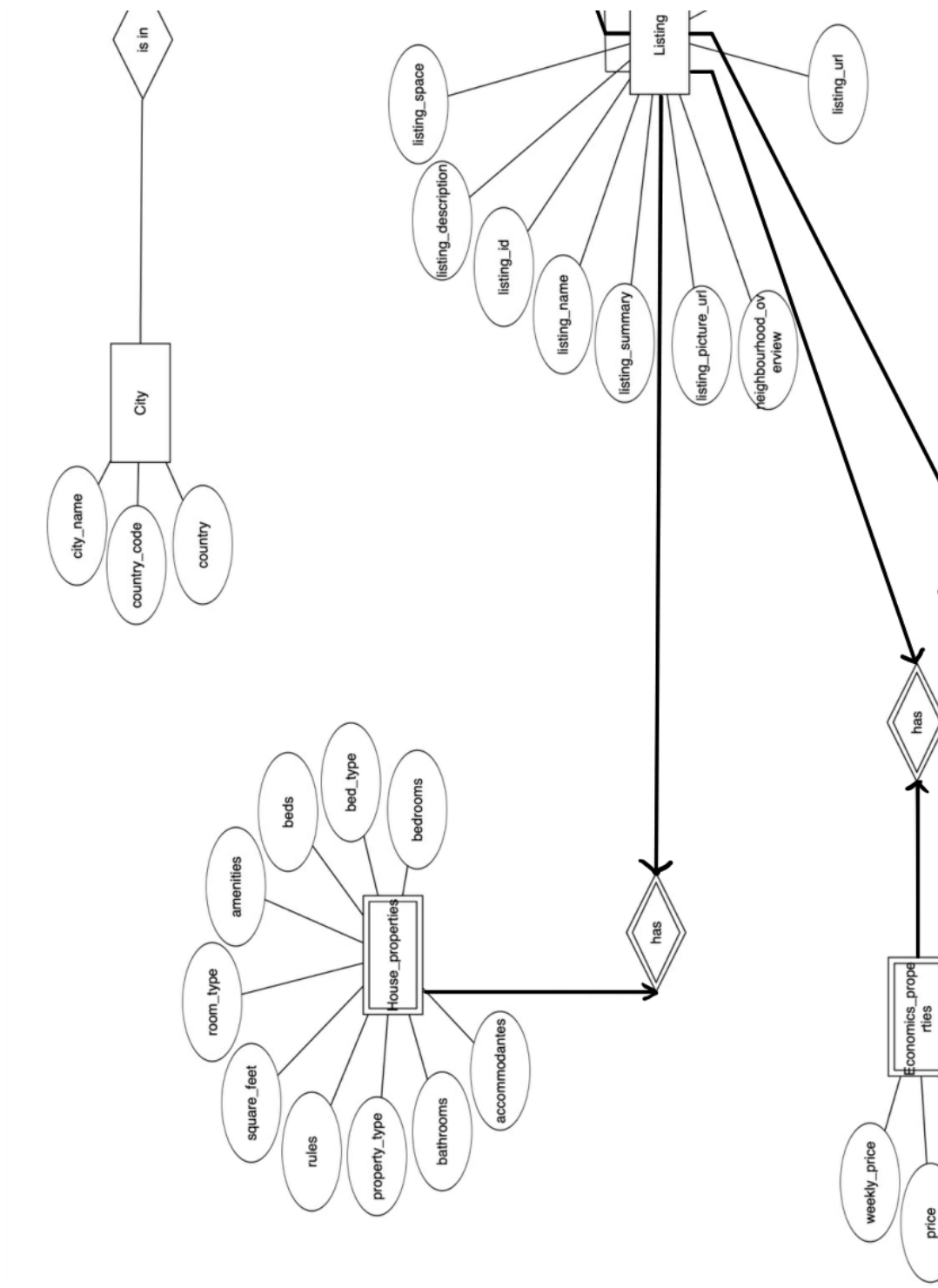
Schema Overview



Zoomed Schema







DIAS: Data-Intensive Applications and Systems Laboratory

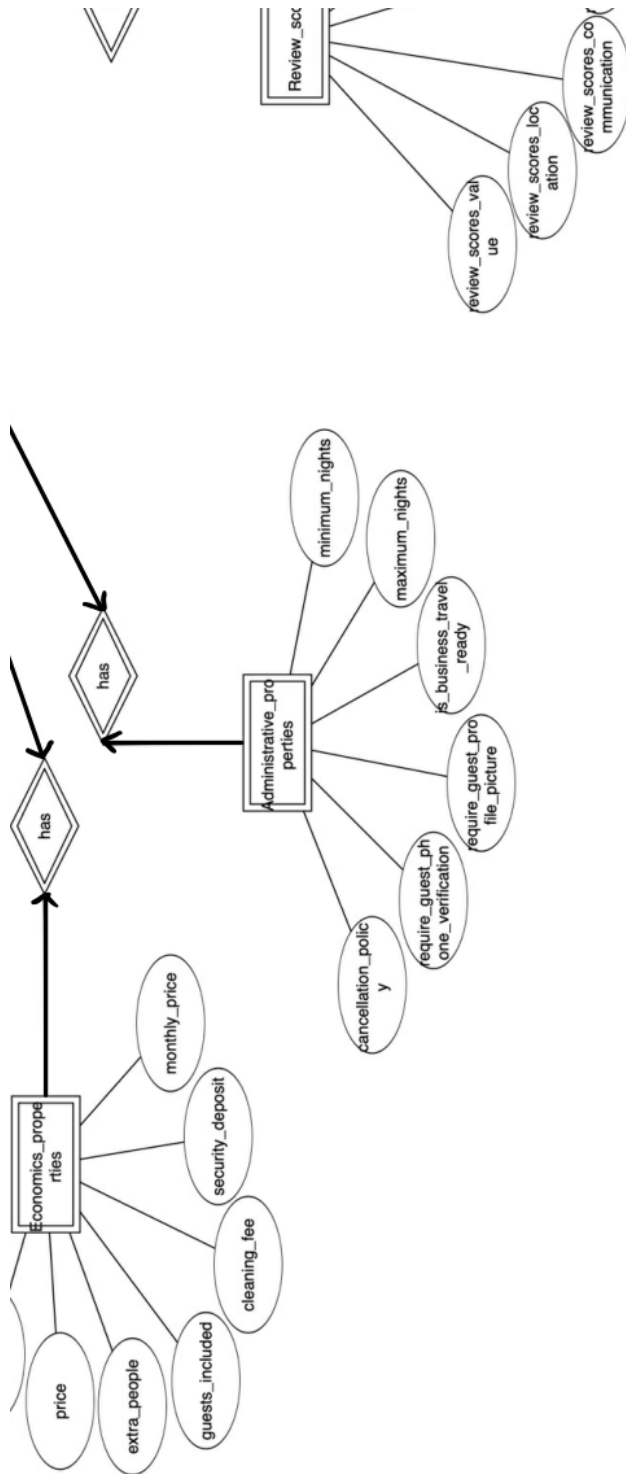
School of Computer and Communication Sciences

Ecole Polytechnique Fédérale de Lausanne

Building BC, Station 14

CH-1015 Lausanne

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



Relational Schema

Table	Refers to	Relates with
Listing	Listing (entity)	Host_id (owns)
Host	Host (entity)	
Neighbourhood	Neighbourhood (entity)	City (is in)
House_properties	House_properties (entity)	Listing (has)
Economic_properties	Economic_properties (entity)	Listing (has)
Administrative_properties	Administrative_properties (entity)	Listing (has)
Review_scores	Review_scores (entity)	Listing (has)
Review	Review (entity)	Reviewer (writes), Listing (reviews)
Reviewer	Reviewer (entity)	
Calender	Calender (entity)	Listing (occupies)
City	City (entity)	
Location	(relation)	Listing, Neighbourhood, City (is in)

DDL

```
1 CREATE DATABASE Airbnb;
2
3 -----Entities-----
4
5 CREATE TABLE Listing (
6
7     -----attributes-----
8     listing_id INT,
9     listing_url VARCHAR(50),
10    listing_name VARCHAR(50),
11    listing_summary TINYTEXT,
12    listing_space TINYTEXT,
13    listing_description TEXT,
14    listing_notes TEXT,
15    listing_transit TEXT,
16    listing_access TEXT,
17    listing_interaction TEXT,
18    listing_picture_url VARCHAR(50),
19    listing_neighbourhood_overview TEXT,
20
21    -----relation attributes----
22    host_id INT NOT NULL,
23
24    -----keys-----
25    PRIMARY KEY(id),
26    FOREIGN KEY(host_id) REFERENCES Host(host_id)
27 );
28
29 CREATE TABLE Host (
30
31     -----attributes-----
32     host_id INT,
33     host_url VARCHAR(50),
34     host_name VARCHAR(50),
35     host_since DATE,
36     host_about TINYTEXT,
37     host_response_time TIME,
38     host_response_rate FLOAT,
39     host_thumbnail_url VARCHAR(50),
40     host_picture_url VARCHAR(50),
41     host_verifications TEXT,
42
43     -----relation attributes----
44     neighbourhood_name VARCHAR(50),
45     city_name VARCHAR(50),
46
47     -----keys-----
48     PRIMARY KEY(host_id),
49     FOREIGN KEY(neighbourhood_name, city_name) REFERENCES Neighbourhood(neighbourhood_name, city_name)
50 );
51
52 CREATE TABLE Neighbourhood (
53
54     -----attributes-----
55     neighbourhood_name VARCHAR(50),
56
```

```
55     neighbourhood_name VARCHAR(50),
56
57     -----relation attributes-----
58     city_name    VARCHAR(50),
59     country_code INT,
60
61     -----keys-----
62     PRIMARY KEY(neighbourhood_name, city_name)
63     FOREIGN KEY(city_name, country_code) REFERENCES City(city_name, country_code) ON DELETE CASCADE
64 );
65
66 CREATE TABLE House_properties (
67
68     -----attributes-----
69     property_type VARCHAR(50),
70     room_type     VARCHAR(50),
71     accommodates  TINYINT,
72     bathrooms    TINYINT,
73     bedrooms     TINYINT,
74     beds         TINYINT,
75     bed_type     VARCHAR(50),
76     amenities    TEXT,
77     square_feet  SMALLINT,
78
79     -----relation attributes-----
80     listing_id INT,
81
82     -----keys-----
83     PRIMARY KEY(listing_id),
84     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
85 );
86
87 CREATE TABLE Economic_properties (
88
89     -----attributes-----
90     price          FLOAT,
91     weekly_price   FLOAT,
92     monthly_price  FLOAT,
93     security_deposit FLOAT,
94     cleaning_fee   FLOAT,
95     guests_included TINYINT,
96     extra_people   FLOAT,
97
98     -----relation attributes-----
99     listing_id INT,
100
101     -----keys-----
102     PRIMARY KEY(listing_id),
103     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
104 );
105
106 CREATE TABLE Administrative_properties (
107
108     -----attributes-----
109     rules          TEXT,
110     minimum_nights INT,
```

```
110     minimum_nights INT,
111     maximum_nights INT,
112     is_business_travel_ready BIT,
113     cancellation_policy TEXT,
114     require_guest_profile_picture BIT,
115     require_guest_phone_verification BIT,
116
117     -----relation attributes-----
118     listing_id INT,
119
120     -----keys-----
121     PRIMARY KEY(listing_id),
122     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
123 );
124
125 CREATE TABLE Review_scores (
126
127     -----attributes-----
128     review_scores_rating FLOAT,
129     review_scores_accuracy FLOAT,
130     review_scores_cleanliness FLOAT,
131     review_scores_checkin FLOAT,
132     review_scores_communication FLOAT,
133     review_scores_location FLOAT,
134     review_scores_value FLOAT,
135
136     -----relation attributes-----
137     listing_id INT,
138
139     -----keys-----
140     PRIMARY KEY(listing_id),
141     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
142 );
143
144 CREATE TABLE Review (
145
146     -----attributes-----
147     review_id INT,
148     review_date DATE,
149     review_comments TEXT,
150
151     -----relation attributes-----
152     reviewer_id INT,
153     listing_id INT,
154
155     -----keys-----
156     PRIMARY KEY(review_id),
157     FOREIGN KEY(reviewer_id) REFERENCES Reviewer(reviewer_id),
158     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id)
159 );
160
161 CREATE TABLE Reviewer (
162
163     -----attributes-----
164     reviewer_id INT,
165     reviewer_name VARCHAR(50)
```

```
164     reviewer_id    INT,
165     reviewer_name  VARCHAR(50),
166     -----relation attributes-----
167
168     -----keys-----
169     PRIMARY KEY(reviewer_id)
170 );
171
172
173 CREATE TABLE Calendar (
174
175     -----attributes-----
176     calendar_date    DATE,
177     calendar_available BIT,
178     calendar_price    FLOAT,
179
180     -----relation attributes-----
181     listing_id INT,
182
183     -----keys-----
184     PRIMARY KEY(listing_id, date),
185     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id)
186 );
187
188 CREATE TABLE City (
189
190     -----attributes-----
191     city_name    VARCHAR(50),
192     country_code TINYINT,
193     country      VARCHAR(50),
194
195     -----relation attributes-----
196
197     -----keys-----
198     PRIMARY KEY(city_name, country_code)
199 );
200
201 -----Relations-----
202
203 CREATE TABLE Location (
204
205     -----attributes-----
206     latitude  FLOAT,
207     longitude FLOAT,
208
209     -----relation attributes-----
210     listing_id      INT,
211     neighbourhood_name VARCHAR(50),
212     city_name        VARCHAR(50),
213
214     -----keys-----
215     PRIMARY KEY(listing_id),
216     FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON CASCADE DELETE,
217     FOREIGN KEY(neighbourhood_name, city_name) REFERENCES Neighbourhood(neighbourhood_name, city_name) ON CASCADE DELETE
218 );
219
```

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

For this first work, we thought it was important to work the three together to understand the database correctly. We designed the basis of the ER model, and modified it until the three of us were satisfied. Then we split the work (SQL commands, report, creation of ER model).