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: 32

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

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/

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.

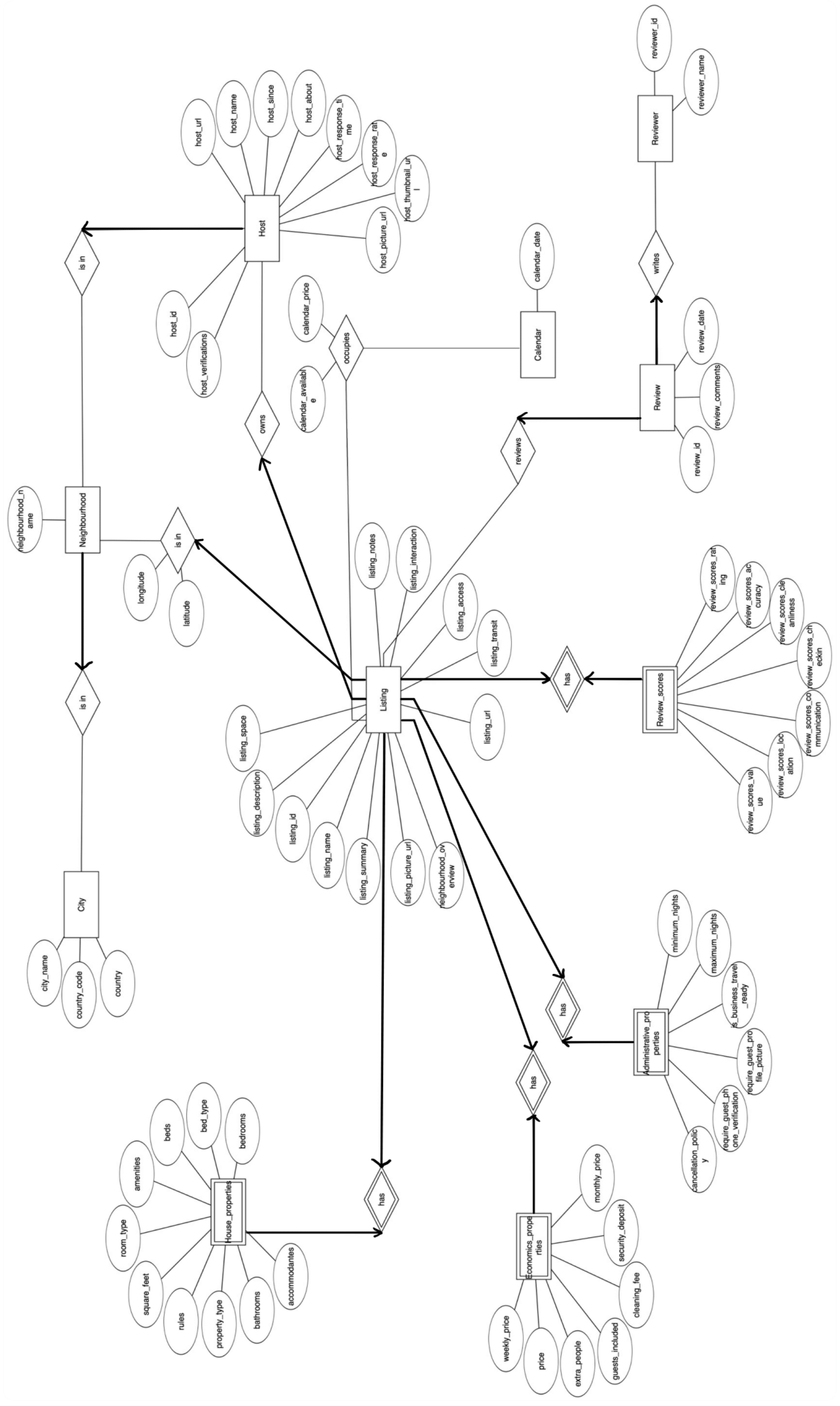
Entity Relationship Schema

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.
Calender	Availabilites of a listing.
Location	Location on a map of a listing.

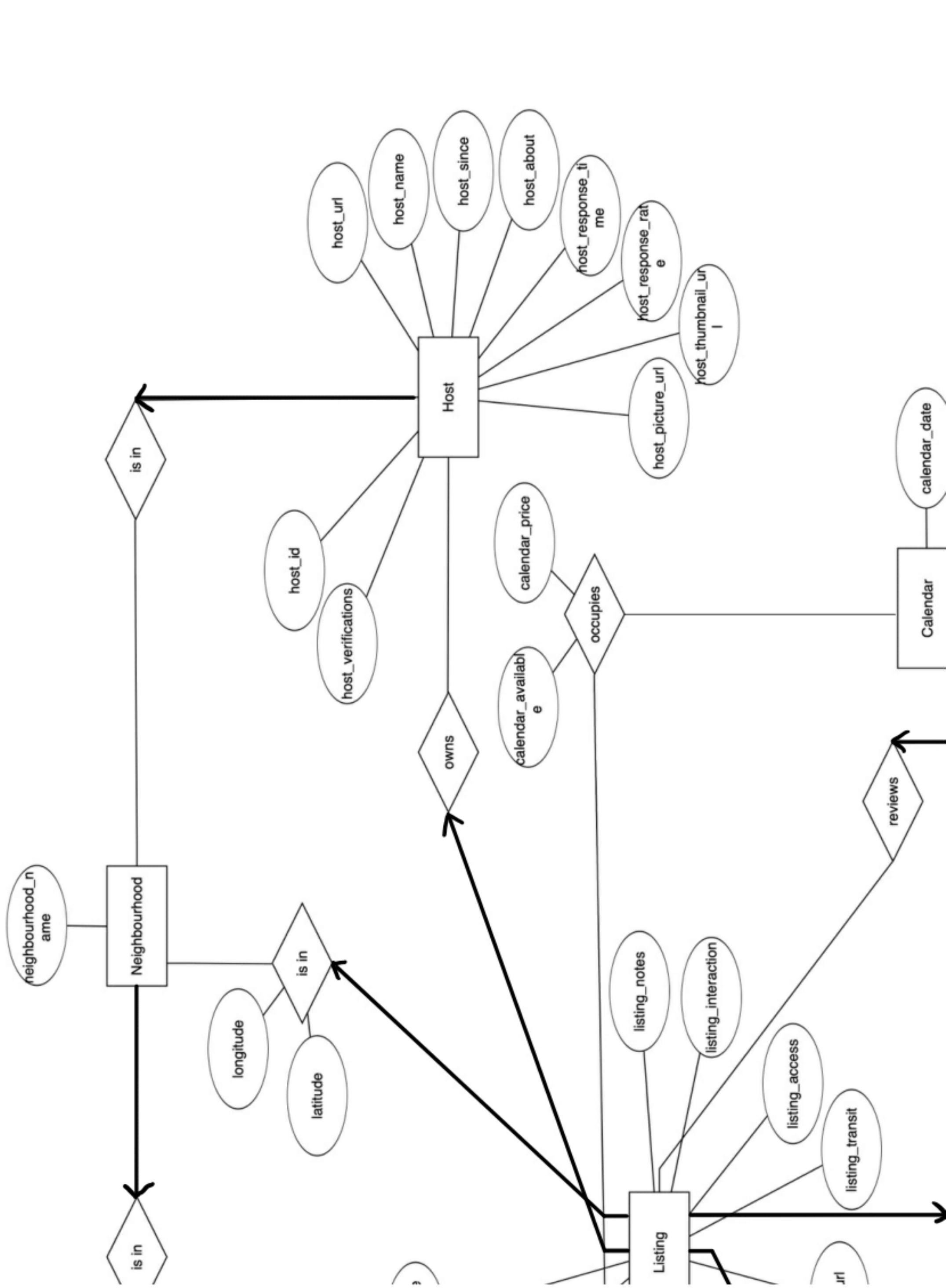
Entity	**Relation**	**Entity**	**Constraints**
a Listing	has	House_properties	one-to-one relationship (Listing‘s weak entity)
a Listing	has	Economic_properties	one-to-one relationship (Listing‘s weak entity)
a Listing	has	Administrative_properties	one-to-one relationship (Listing‘s weak entity)
a Listing	has	Review_scores	one-to-one relationship (Listing‘s weak entity)
a Host	owns	a Listing	each listing has exatcly one host
a Listing	occupies	a Calender	a listing has availabilities in time. Each date has the corresponding listing's availability
a Review	reviews	a Listing	a review reviews exaxtly one listing
a Reviewer	writes	a Review	a review has exactly one reviewer
a Listing	is in	a Neighbourdhood	a listing is in exactly one neigbourhood
a Listing	is in	a City	a listing is in exactly one city
a Neigbourhood	is in	a City	a neigborhood is in exactly one city

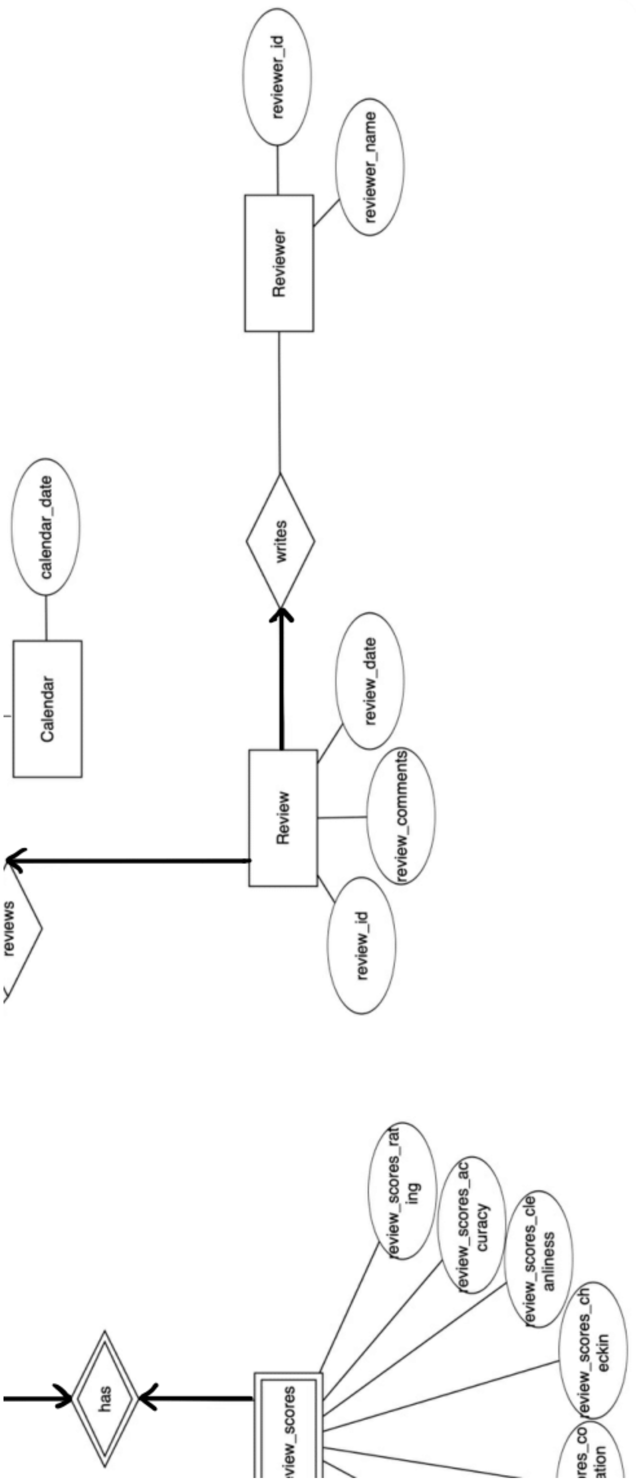
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

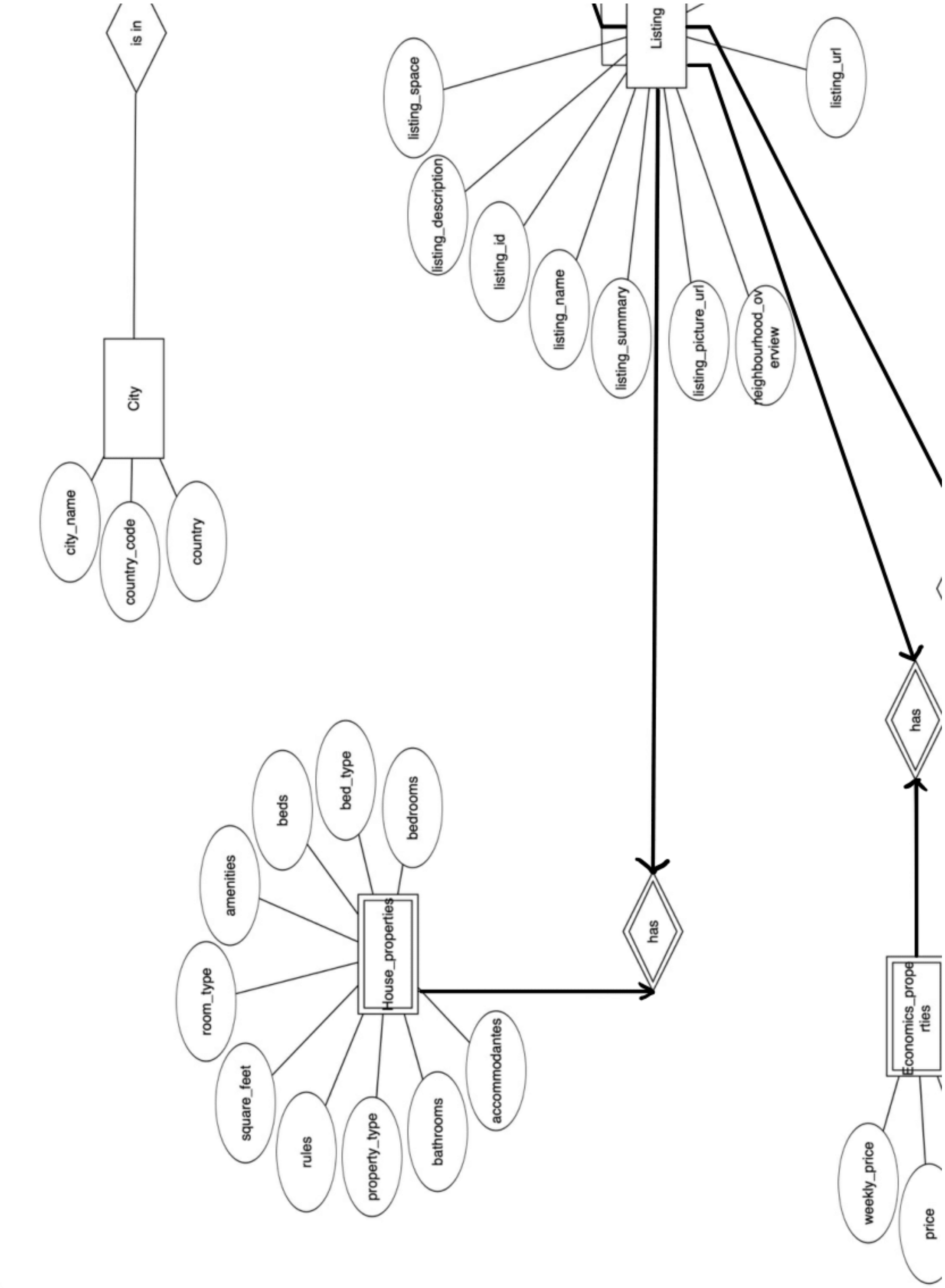
Schema Overview

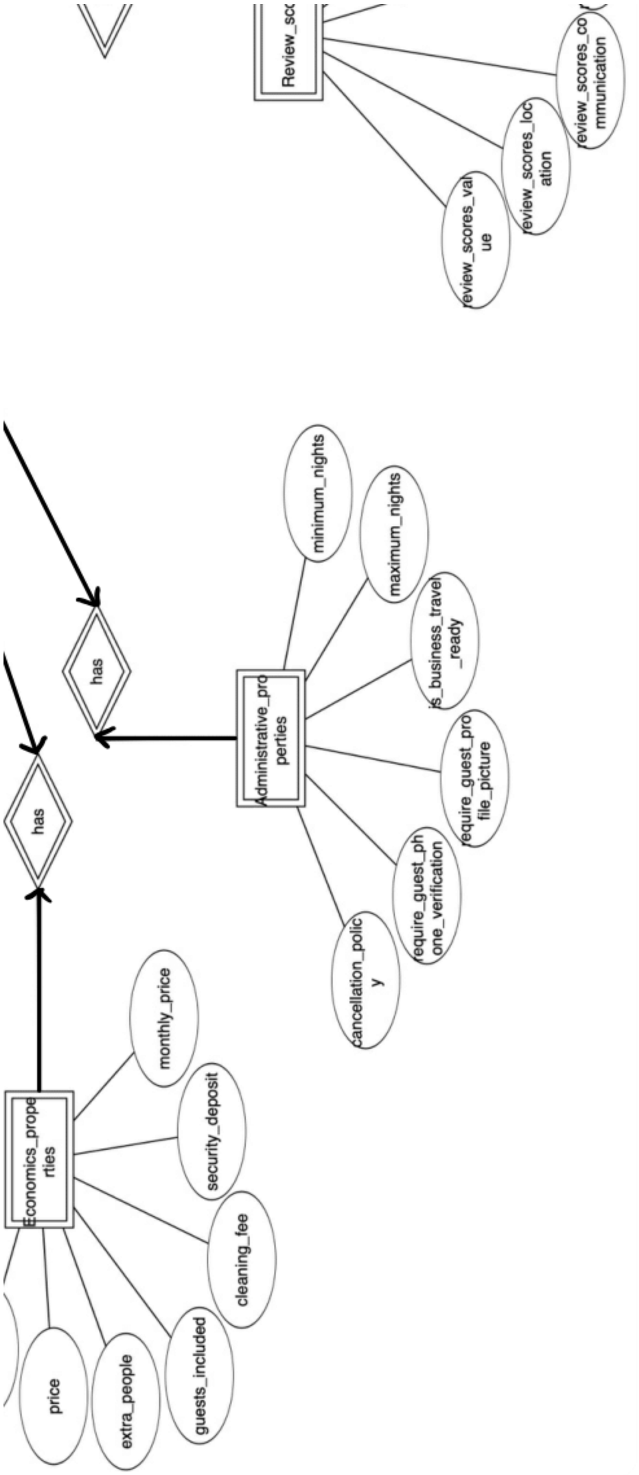


Zoomed Schema




reviewer_id
reviewer_name
Reviewer
writes
calendar_date
Calendar
review_date
Review
review_comments
review_id
reviews
has
view_scores
review_scores_rat ing
review_scores_ac curacy
review_scores_cle anliness
review_scores_ch eckin
res_co ation

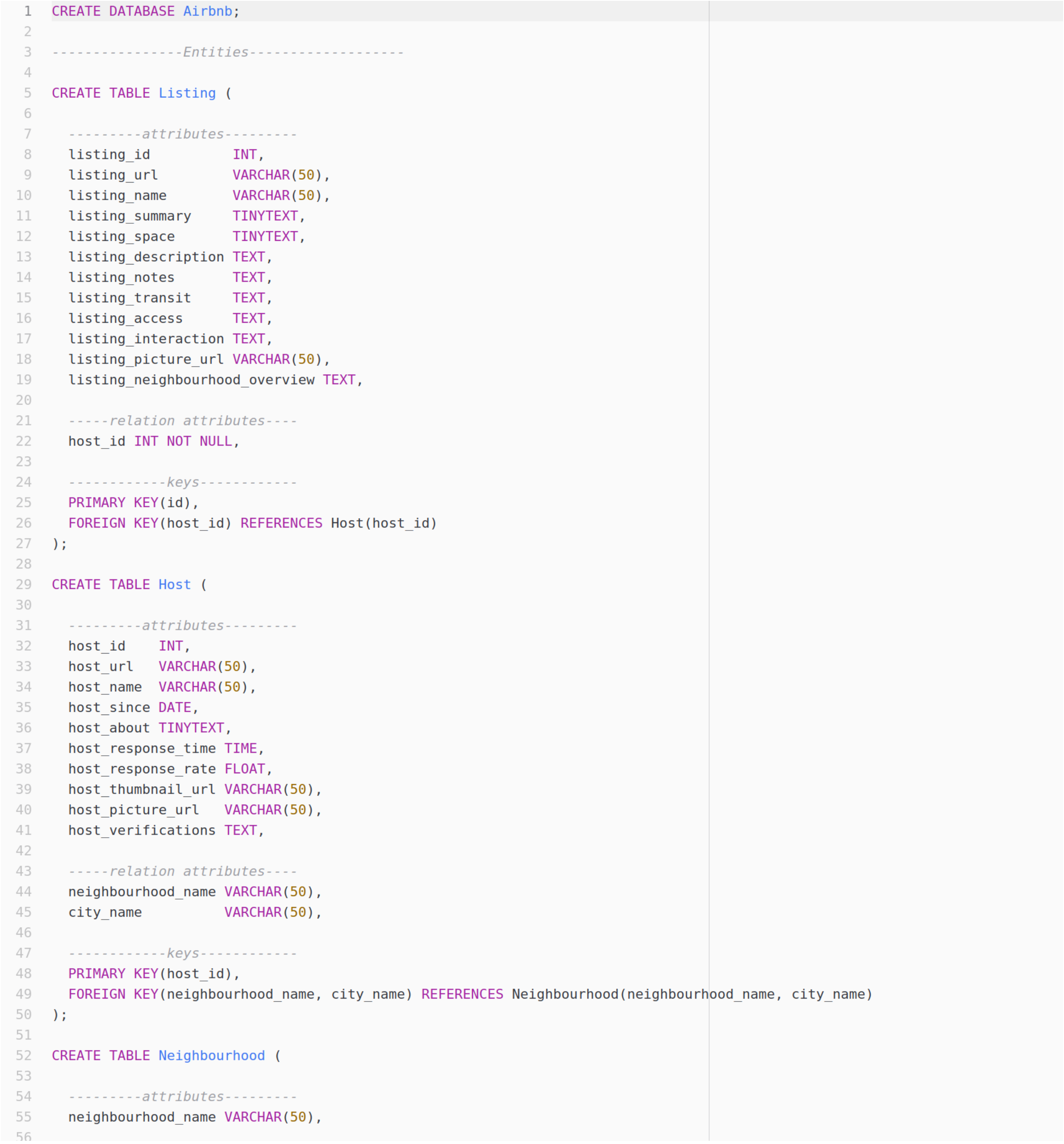

is in
City
city_name
country_code
country
Listing
listing_space
listing_description
listing_id
listing_name
listing_summary
listing_picture_url
neighbourhood_ov erview
listing_url
has
has
House_properties
beds
bed_type
bedrooms
amenities
room_type
square_feet
rules
property_type
bathrooms
accommodantes
Economics_prope rties
weekly_price
price


Review_sco
review_scores_communication
review_scores_location
review_scores_value
has
has
minimum_nights
maximum_nights
is_business_travel_ready
Administrative_properties
require_guest_profile_picture
require_guest_phone_verification
cancellation_policy
Economics_properties
monthly_price
security_deposit
cleaning_fee
guests_included
extra_people
price

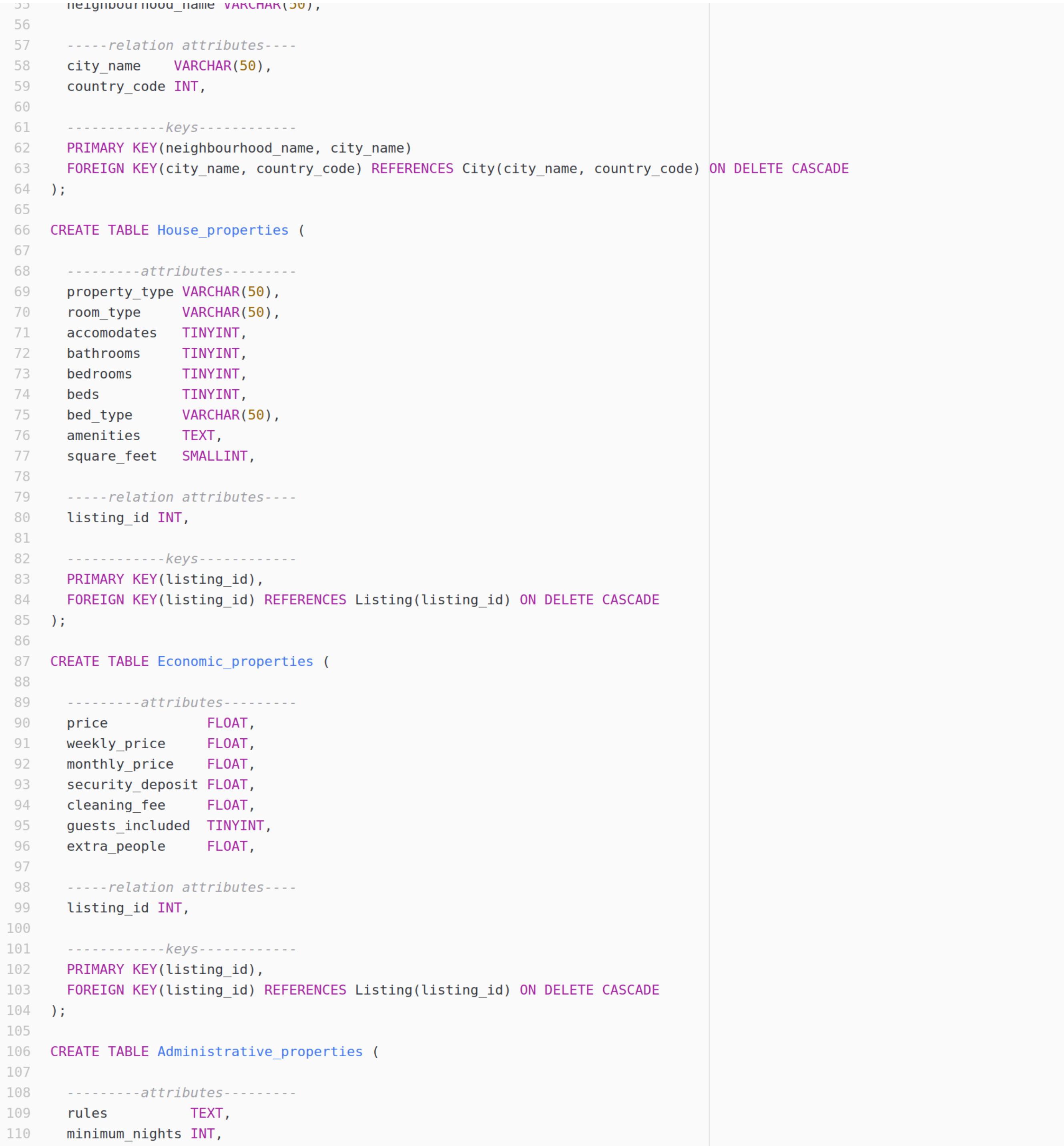
Relational Schema

Table	Referes to	Relates with
Listing	Listing (entity)	Host_id (owns)
Host	Host (entity)	
Neigbourhood	Neigbourhood (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)

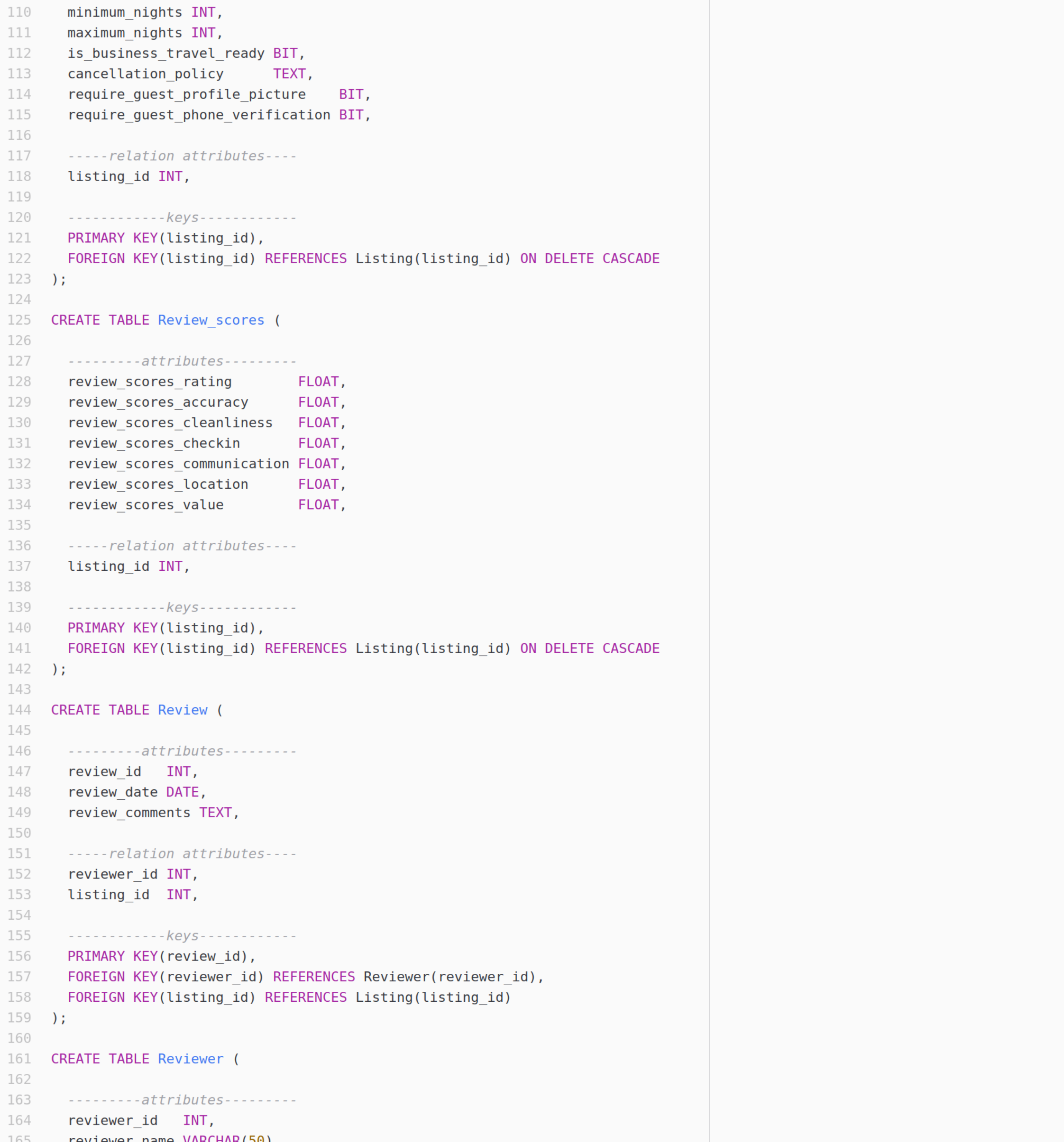
DDL



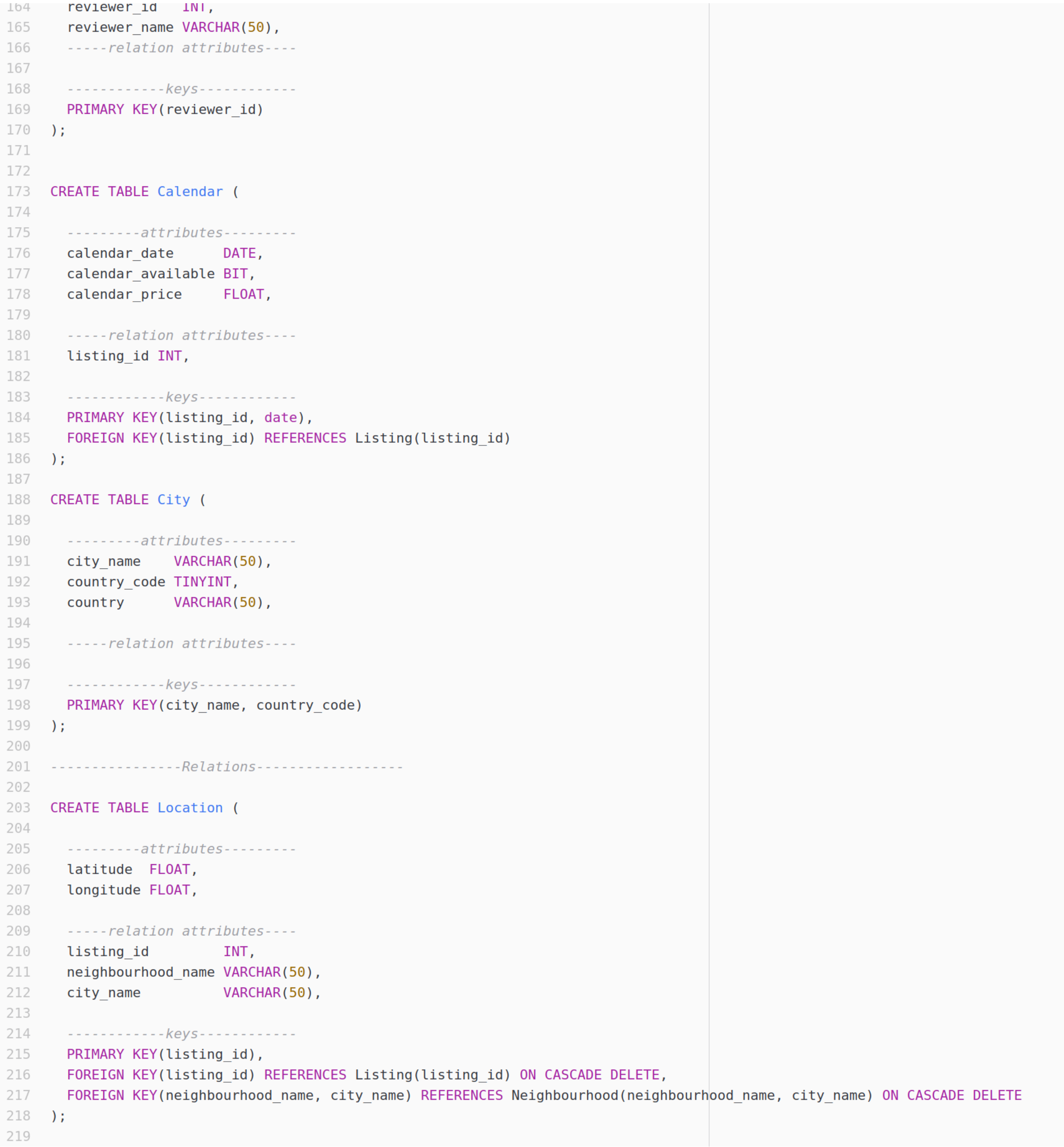
```
CREATE DATABASE Airbnb;

----------------Entities------------------

CREATE TABLE Listing (

  ---------attributes---------
  listing_id          INT,
  listing_url         VARCHAR(50),
  listing_name        VARCHAR(50),
  listing_summary     TINYTEXT,
  listing_space       TINYTEXT,
  listing_description TEXT,
  listing_notes       TEXT,
  listing_transit     TEXT,
  listing_access      TEXT,
  listing_interaction TEXT,
  listing_picture_url VARCHAR(50),
  listing_neighbourhood_overview TEXT,

  -----relation attributes----
  host_id INT NOT NULL,

  ------------keys------------
  PRIMARY KEY(id),
  FOREIGN KEY(host_id) REFERENCES Host(host_id)
);

CREATE TABLE Host (

  ---------attributes---------
  host_id    INT,
  host_url   VARCHAR(50),
  host_name  VARCHAR(50),
  host_since DATE,
  host_about TINYTEXT,
  host_response_time TIME,
  host_response_rate FLOAT,
  host_thumbnail_url VARCHAR(50),
  host_picture_url   VARCHAR(50),
  host_verifications TEXT,

  -----relation attributes----
  neighbourhood_name VARCHAR(50),
  city_name          VARCHAR(50),

  ------------keys------------
  PRIMARY KEY(host_id),
  FOREIGN KEY(neighbourhood_name, city_name) REFERENCES Neighbourhood(neighbourhood_name, city_name)
);

CREATE TABLE Neighbourhood (

  ---------attributes---------
  neighbourhood_name VARCHAR(50),

```



```
  neighbourhood_name VARCHAR(50),

  -----relation attributes----
  city_name    VARCHAR(50),
  country_code INT,

  ------------keys------------
  PRIMARY KEY(neighbourhood_name, city_name)
  FOREIGN KEY(city_name, country_code) REFERENCES City(city_name, country_code) ON DELETE CASCADE
);

CREATE TABLE House_properties (

  ---------attributes---------
  property_type VARCHAR(50),
  room_type     VARCHAR(50),
  accomodates   TINYINT,
  bathrooms     TINYINT,
  bedrooms      TINYINT,
  beds          TINYINT,
  bed_type      VARCHAR(50),
  amenities     TEXT,
  square_feet   SMALLINT,

  -----relation attributes----
  listing_id INT,

  ------------keys------------
  PRIMARY KEY(listing_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
);

CREATE TABLE Economic_properties (

  ---------attributes---------
  price            FLOAT,
  weekly_price     FLOAT,
  monthly_price    FLOAT,
  security_deposit FLOAT,
  cleaning_fee     FLOAT,
  guests_included  TINYINT,
  extra_people     FLOAT,

  -----relation attributes----
  listing_id INT,

  ------------keys------------
  PRIMARY KEY(listing_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
);

CREATE TABLE Administrative_properties (

  ---------attributes---------
  rules          TEXT,
  minimum_nights INT,
```



```
  minimum_nights INT,
  maximum_nights INT,
  is_business_travel_ready BIT,
  cancellation_policy      TEXT,
  require_guest_profile_picture    BIT,
  require_guest_phone_verification BIT,

  -----relation attributes----
  listing_id INT,

  ------------keys------------
  PRIMARY KEY(listing_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
);

CREATE TABLE Review_scores (

  ---------attributes---------
  review_scores_rating        FLOAT,
  review_scores_accuracy      FLOAT,
  review_scores_cleanliness   FLOAT,
  review_scores_checkin       FLOAT,
  review_scores_communication FLOAT,
  review_scores_location      FLOAT,
  review_scores_value         FLOAT,

  -----relation attributes----
  listing_id INT,

  ------------keys------------
  PRIMARY KEY(listing_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON DELETE CASCADE
);

CREATE TABLE Review (

  ---------attributes---------
  review_id   INT,
  review_date DATE,
  review_comments TEXT,

  -----relation attributes----
  reviewer_id INT,
  listing_id  INT,

  ------------keys------------
  PRIMARY KEY(review_id),
  FOREIGN KEY(reviewer_id) REFERENCES Reviewer(reviewer_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id)
);

CREATE TABLE Reviewer (

  ---------attributes---------
  reviewer_id   INT,
  reviewer_name VARCHAR(50),
```



```
  reviewer_id   INT,
  reviewer_name VARCHAR(50),
  -----relation attributes----

  ------------keys------------
  PRIMARY KEY(reviewer_id)
);


CREATE TABLE Calendar (

  ---------attributes---------
  calendar_date      DATE,
  calendar_available BIT,
  calendar_price     FLOAT,

  -----relation attributes----
  listing_id INT,

  ------------keys------------
  PRIMARY KEY(listing_id, date),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id)
);

CREATE TABLE City (

  ---------attributes---------
  city_name    VARCHAR(50),
  country_code TINYINT,
  country      VARCHAR(50),

  -----relation attributes----

  ------------keys------------
  PRIMARY KEY(city_name, country_code)
);

----------------Relations-----------------

CREATE TABLE Location (

  ---------attributes---------
  latitude  FLOAT,
  longitude FLOAT,

  -----relation attributes----
  listing_id         INT,
  neighbourhood_name VARCHAR(50),
  city_name          VARCHAR(50),

  ------------keys------------
  PRIMARY KEY(listing_id),
  FOREIGN KEY(listing_id) REFERENCES Listing(listing_id) ON CASCADE DELETE,
  FOREIGN KEY(neighbourhood_name, city_name) REFERENCES Neighbourhood(neighbourhood_name, city_name) ON CASCADE DELETE
);

```

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).