

**Database Application and Design (SOC-3060) -- Spring 2019**

**Database System Report**

**for**

**<Project Title: *MountHelp*>**

**Prepared by <u1610012, Abu-Bakr Jabbarov, 002>**

**<u1610059\*, Dilmurod Ubaydullaev, 002>**

**<u1610250, Usmonbek Ravshanov, 003>**

**<Team-name: Dream-Chasers>**

**<Date: 06/05/2019>**

Table of Contents

[1. Introduction and Scope 5](#_Toc8073890)

[2. Product Perspective 6](#_Toc8073891)

[3. Specific Requirements 9](#_Toc8073892)

[3.1 Logical Database Requirements 9](#_Toc8073893)

[3.2 Functional Requirements 12](#_Toc8073894)

[4. Design 12](#_Toc8073895)

[4.1 Conceptual Database Design 13](#_Toc8073896)

[4.3 Physical Database Design by ERWin 15](#_Toc8073899)

[5. Dependency Design 16](#_Toc8073901)

[5.1 Normalization 18](#_Toc8073902)

[6. Interface 19](#_Toc8073903)

[7. Testing of Database 21](#_Toc8073904)

[8. Bibliography 24](#_Toc8073905)

# 1. Introduction and Scope

*Introduce the DB application of your choice (approved by Project Proposal) and describe it in plain English. The length of this “narrative” should be more or less one page written in a classic, regular font type (e.g., Arial or Times), font-size 12pt, single spacing, about 1.5cm margins. Introduce the basic functionalities, goals, entities and relationships among the data concerned. In particular, this section should: (a) Identify the software product to be produced by name (e.g., Host DBMS, Report Generator, etc.); (b) Explain what the software product will, and, if necessary, will not do; (c) Describe the application of the software being specified, including relevant benefits, objectives, and goals.*

Tourism industry in Uzbekistan is growing in a fast way. Most tourists visit the oldest and greatest places in Uzbekistan such as Rome of the East (Samarkand), Town Museum (Bukhara), Caravans city (Khiva), Karakalpakstan and other places. [1] The area of mountains of Uzbekistan make up 96 thousand kilometers, which in total 21,3% of the territory of Uzbekistan. [2] Mountains are important assets for the tourism industry. They take up an estimated share of 15-20% of the global tourism market, generating between 100 and 140 billion US$ per year. [3] Observing these statistics and knowing the fact that Uzbekistan does not concede a ranking in global tourism, we have decided to design software product called “*MountHelp*” that will attract attention of all tourists and make tours much easier.

The main problems in Mountain tourism of Uzbekistan are: lack of information about the mountains, their locations and routes; very few and expensive hotels; slow and uncomfortable transportation (public transport); Our software product provides solution to all of these problems. Our application consists of three main branches: finding house to stay, joining the tour, booking a place in a car. Our application is very helpful for international tourists as most of Uzbek people have their own cars and cottages in mountains.

Tourists will be registered as TOURISTs. They can BOOK and REVIEW the HOUSE that is owned by HOST. Moreover, TOURIST can JOIN and REVIEW CAR\_DRIVERs who own CARS. As was mentioned above, TOURISTs can join to the TOURs that can be PUBLIC or INDIVIDUAL. The additional features of software product is EMERGENCY SERVICES and RESTROOMS. International tourists can have difficulties finding hospitals or police departments. Therefore, our application will store the information related to POLICE, FIRE STATION, HOSPITALS (first aid) and GENERAL SERVISES (car repair and etc.) The other additional feature is RESTROOMS. One of the biggest problems is finding restrooms in the mountains. Therefore, with our application it is easy to find it.

# 

# 2. Product Perspective

*This section should: (a) Compare the proposed software product with other similar products existing in the market; provide precise references (e.g., names and web links) of those similar products; highlight why or in which features and functionalities the proposed product should be distinguished from other comparable products. (b) Put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the product is a component of a larger system, as frequently occurs, then this section should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. Note: A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful; it will be considered a plus.*

As it was written above, the software product that we have generated is devoted to the development of mountain tourism. It can also be used in the big cities if needed. The three main components of our application are hosting, ride sharing, and joining to the tours assembled by organizers. Host families of mountain areas accommodate their houses, tourists can find and join the ride shared by car drivers, people (organizers) publish their tours (events) so that anyone can join them (There are public and individual tours that can be distinguished by type). There are no alternative applications for our product, however, there some applications that has almost similar functionality of the only part of our software product that is still used not in the mountains. They are: Airbnb (for housing), Uber (for ride share) and Travello (for tours).

Our software product is combination of those three programs plus additional features that has the main goal of attracting tourists to the mountains of Uzbekistan and make their trip comfortable.

Similarities:

Airbnb [4]: <http://www.airbnb.com/>

Functionalities: host a home or become a host. User inserts the location, check-in and checkout dates with number of guests. Selects the needed one and pays for that. User can see the ranking and the reviews of host. User can become a host and earn money.

Uber [5]: <https://www.uber.com/>

Functionalities: Uber has a lot of functions like delivery, carrier, shipper, business travel, but the main function is ride share. User can be a driver or join a ride. User signs up by entering name, email and other necessary information to drive or taps phone number to ride. User can check the ratings for drivers and riders as well.

Travello [6]: <https://www.travelloapp.com/>

Functionalities: Travello is social networking application for travelers. It doesn’t offer the tours directly, but searches for people nearby to join the travel. User can discover the tour, post travel updates, join groups and match travel plans.

Differences:

Our software product merges modified versions of all these three applications and adds additional features in it:

* User can be registered as tourist and use the services: hosting, ride sharing and tour organizing.
* User can be registered as car driver, offer the ride and earn money.
* User can be registered as organizer, offer individual or public tours for money or free as well.
* User can be registered as host, offer the house and earn money.
* User can see the photos and reviews of houses and the ranks of hosts. Airbnb doesn’t offer house photos
* User can earn money by offering tours indicating all the schedule and activities that is not supported in Travello.
* User can access the additional features: restrooms and emergency services that are crucial in the mountains and get all the information needed.

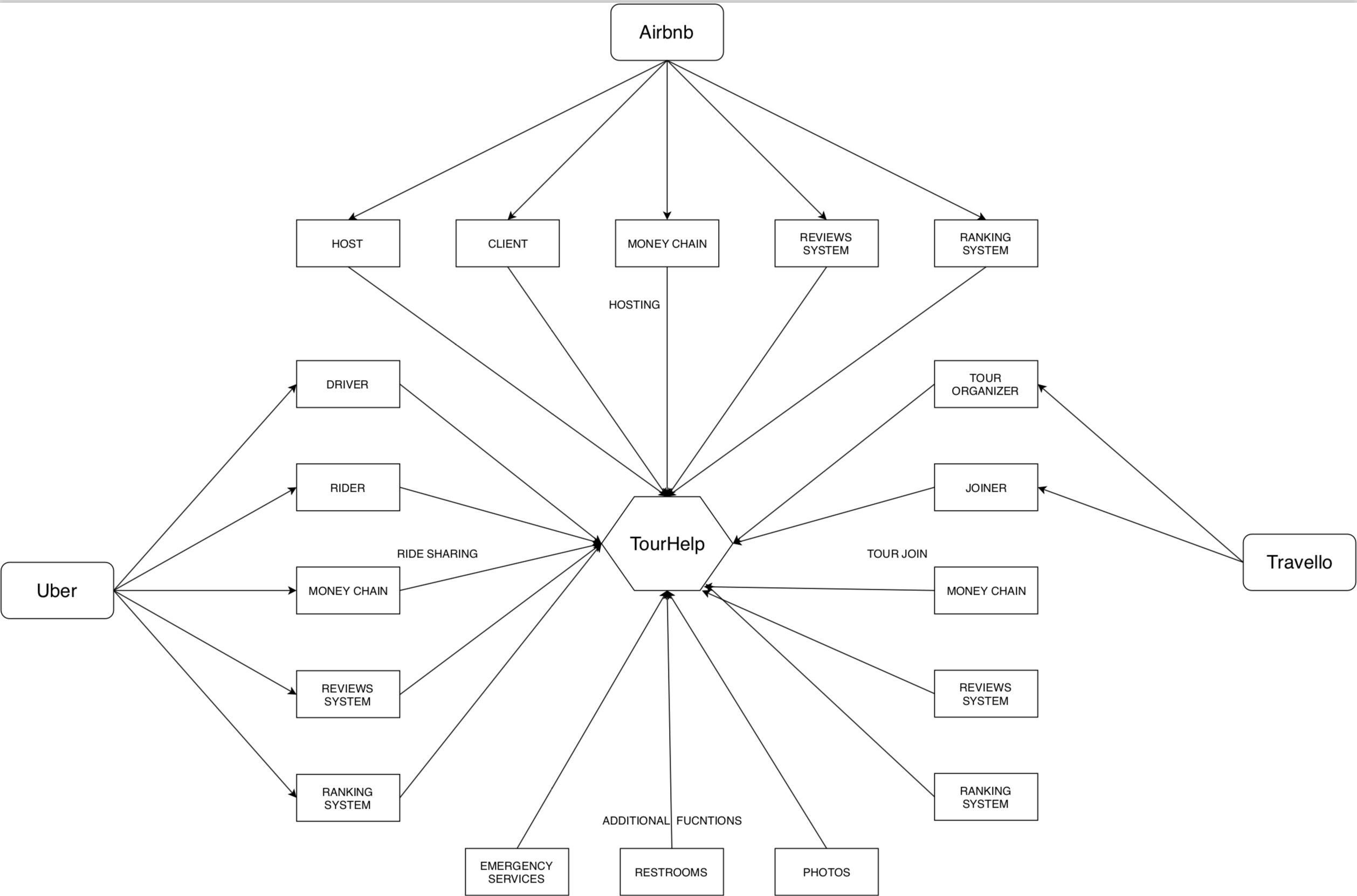


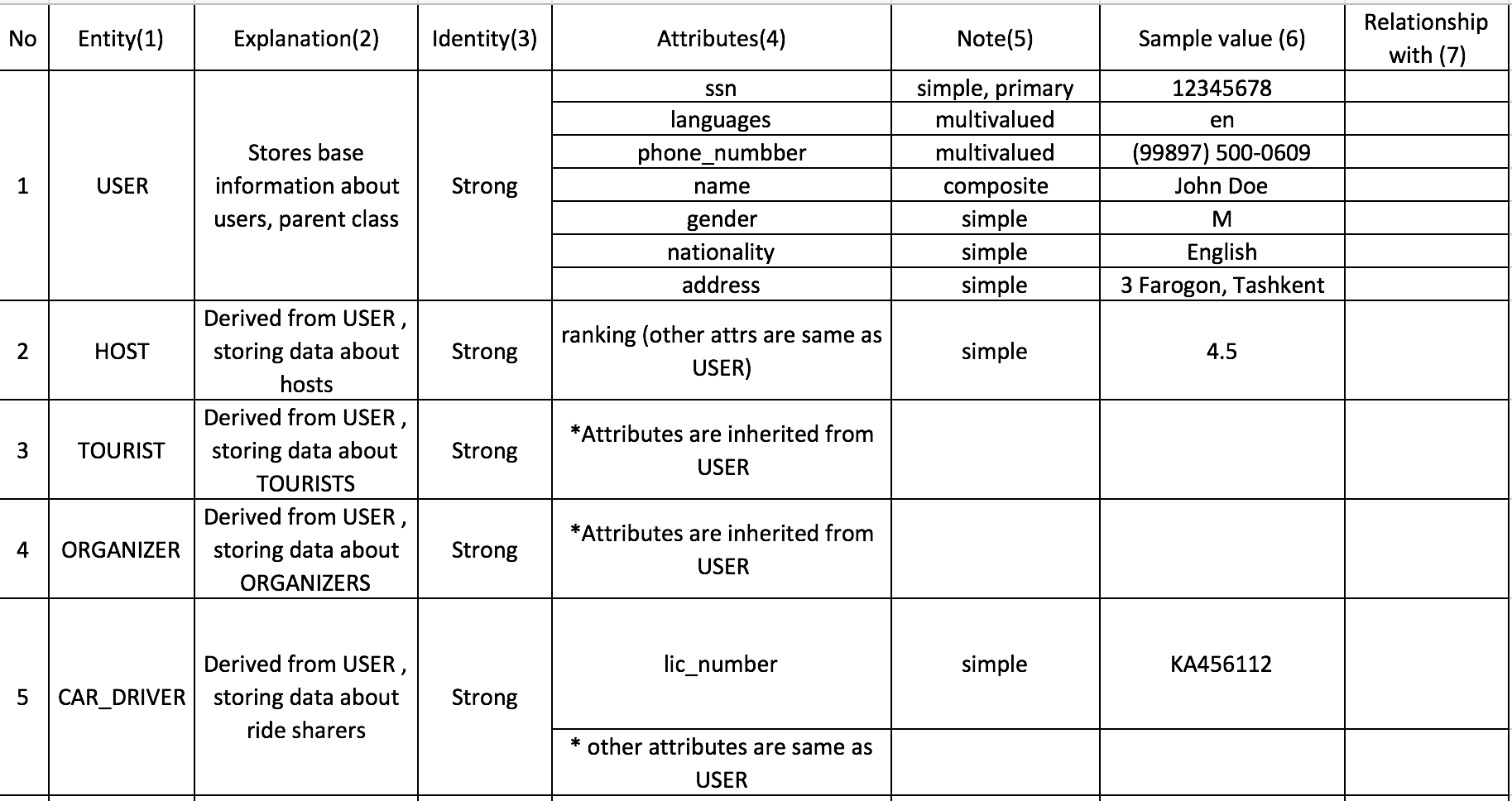
Figure 1: Product Perspective Diagram

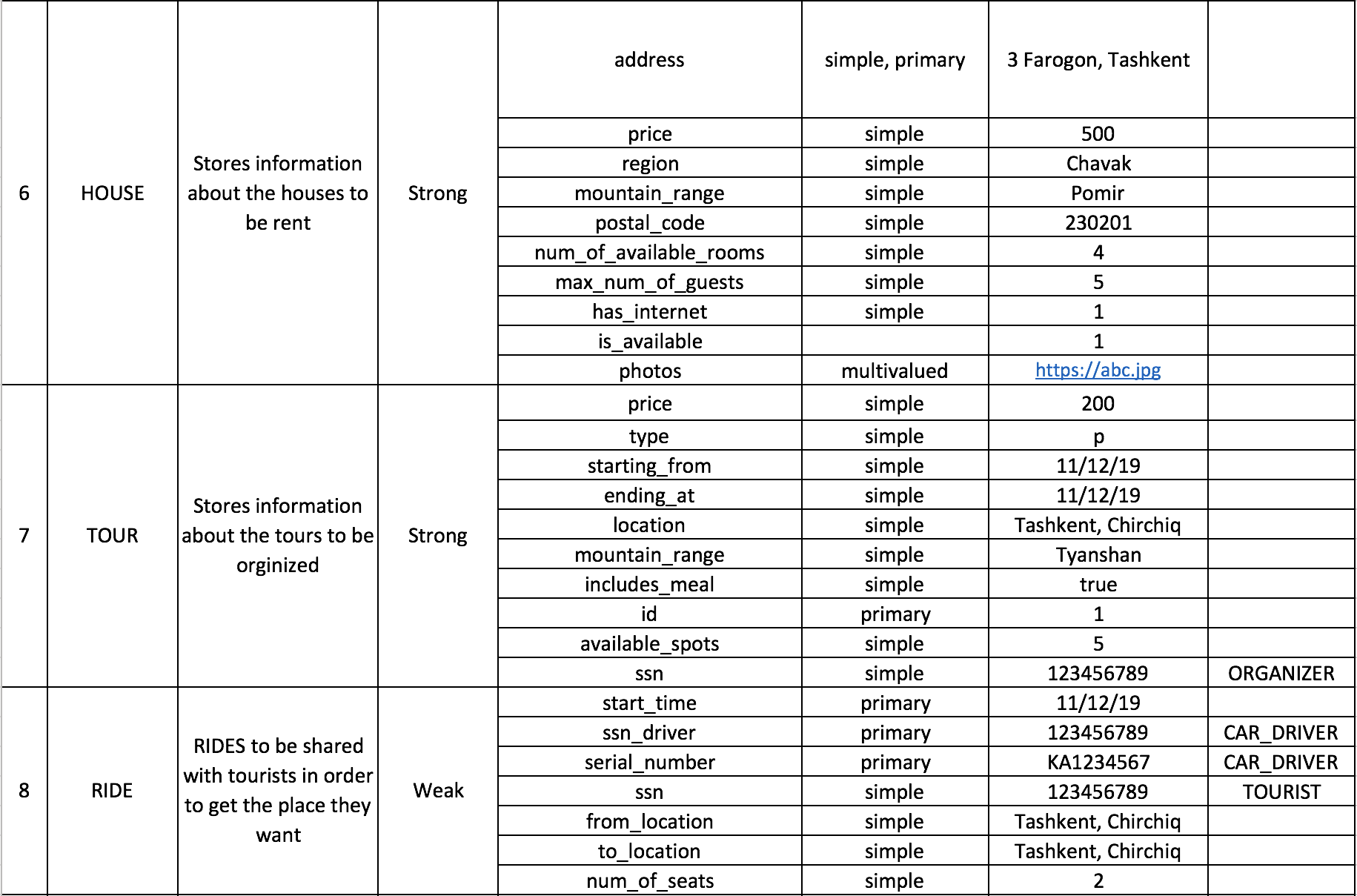
# 3. Specific Requirements

*This section should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. For a (simplified) database application, this section should include at the very least the following:*

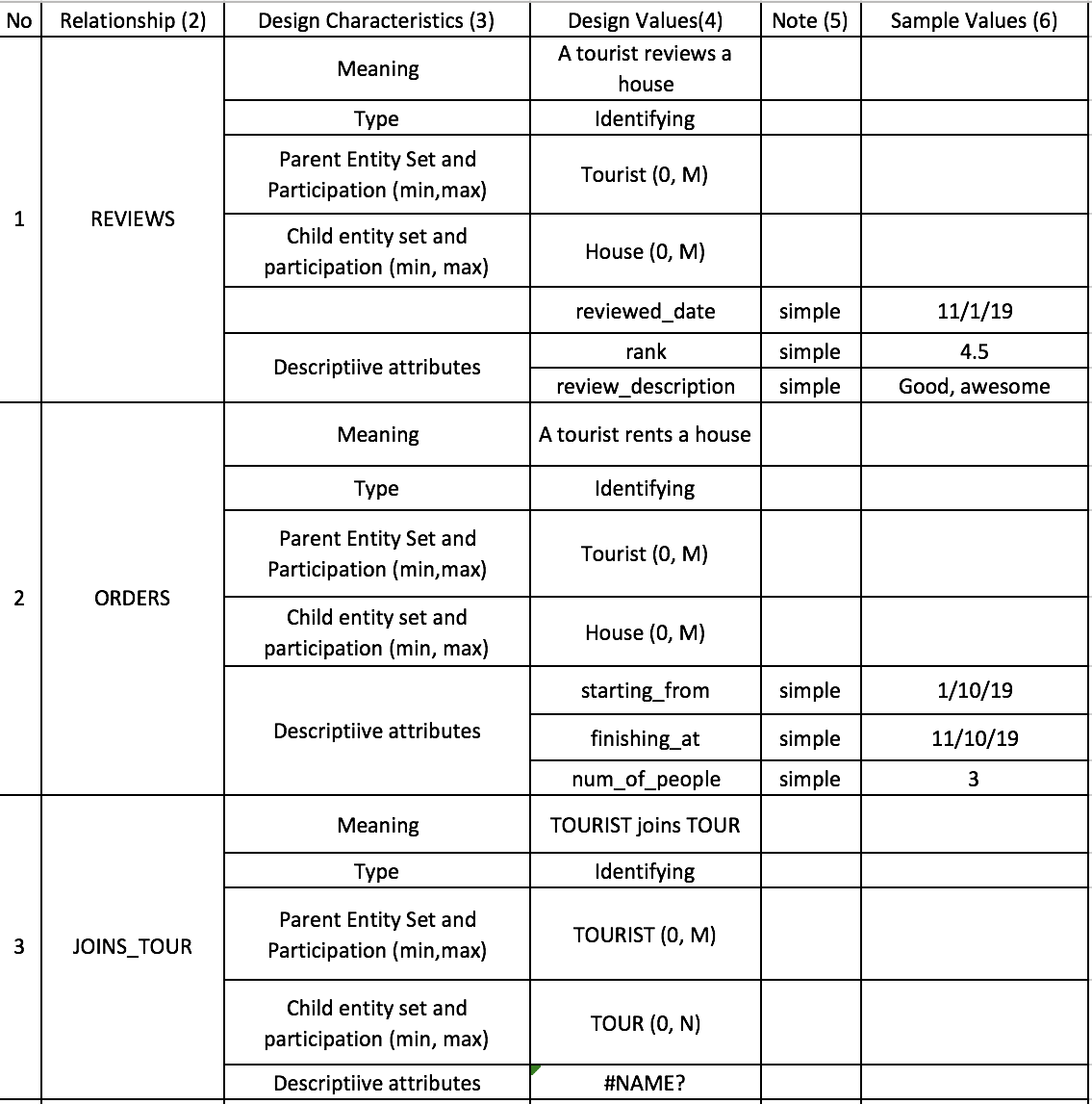
## 3.1 Logical Database Requirements

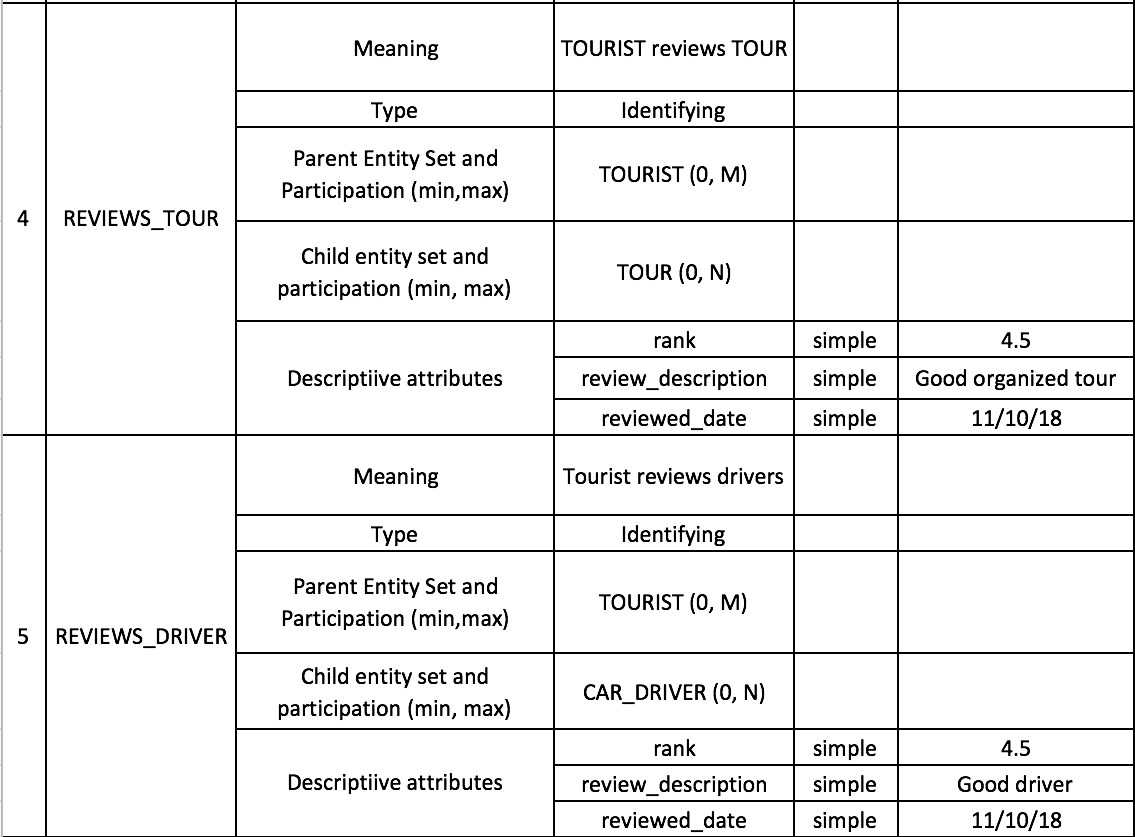
*(a)  Logical database requirements (3.1). Define (“specify’) the logical requirements for any data (information) that is to be placed into a database. For a simple database application as that considered here, this may include the following: a) Data entities and their relationships; (\*) b) Integrity constraints.  (\*) IMPORTANT: Data entities and relationships must be specified by using a kind of tables of the form discussed in class for entity set and relationship sets, respectively. The tables can possibly be modified to deal with relationships of degree higher than two, classes and aggregations. Simplified tables are also accepted, provided that they containing the fundamental elements of original tables. In doubt, ask.*











## 3.2 Functional Requirements

*(b)  Functional requirements (3.2). Define (“specify”) the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. The specification format is an enumerated list of functions expressed by the English statement: “The system shall...”.*

The system will:

* Get location as input and show all the available houses accommodated hosts as output
* Get location as input and show all the available tours assembled by organizers as output
* Get location as input and show all the available rides shared by car drivers as output
* Get the house address as input and show reviews and photos related to that house as outputs
* Get the ride id as input and show reviews related to the car driver as output
* Get tour id as input and show reviews related to tours as output
* Filter drivers, hosts and organizers by their spoken languages, nationality and gender
* Show and compare the prices and ranking of houses, tours and rides
* Search and show restrooms near your location
* Search and show the information about emergency services near your location

# 4. Design

*This section should contain the design elements for a complete software design description (SDD; cf. IEEE standard n.1016-2009). For a simple database application as that considered here, this section should include at the very least the following:*

## 4.1 Conceptual Database Design

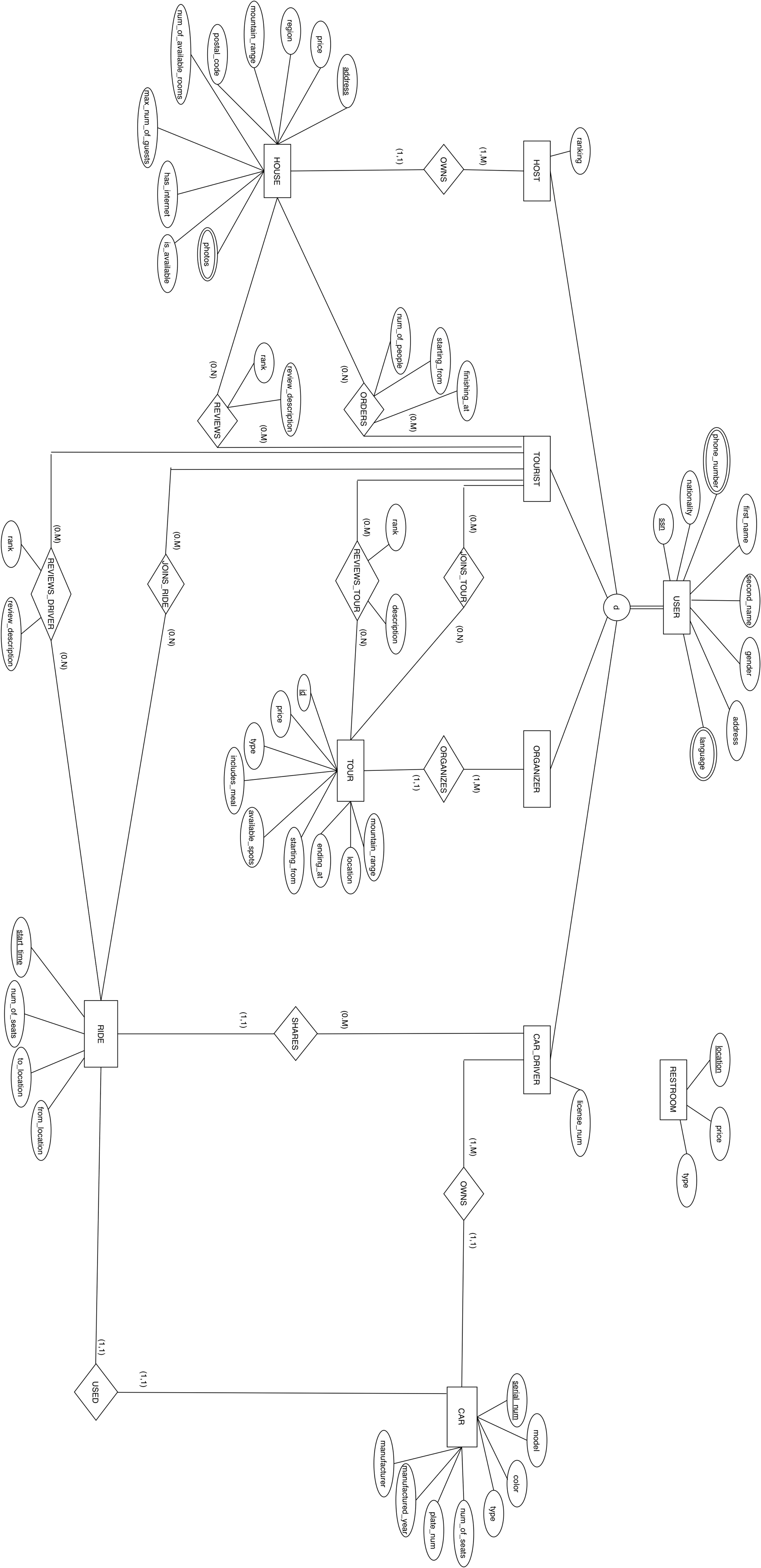


Figure 2: EER Conceptual database design

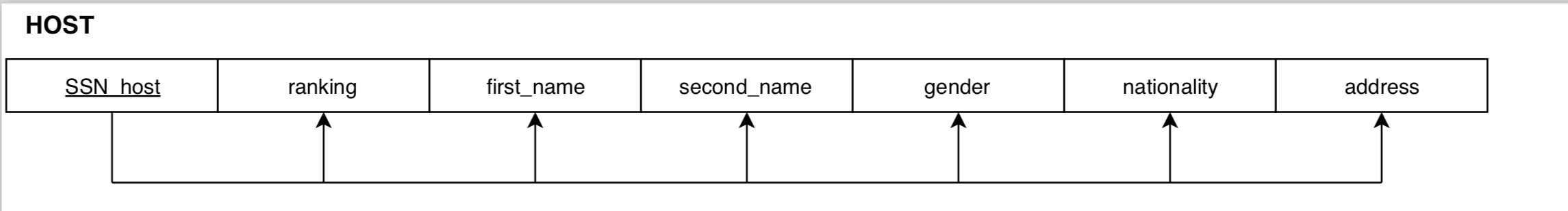
## 4.2 Logical Database Design by ERWin../Downloads/team10_logical.png

## 4.3 Physical Database Design by ERWin

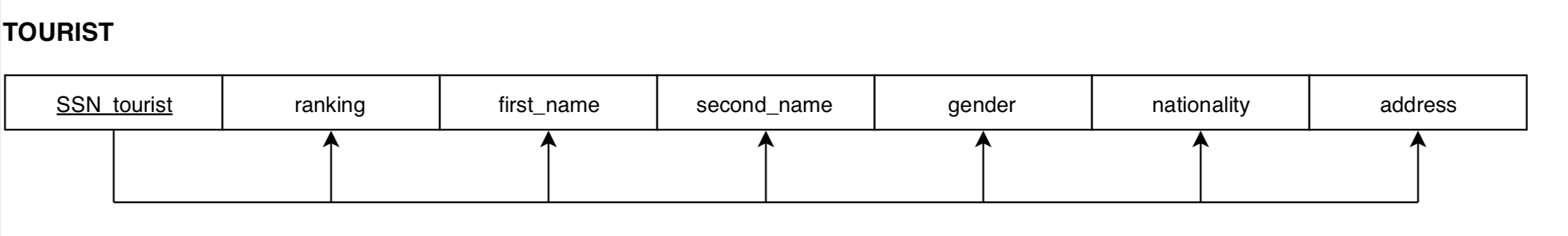
## ../Downloads/team10_physical.png

# 5. Dependency Design

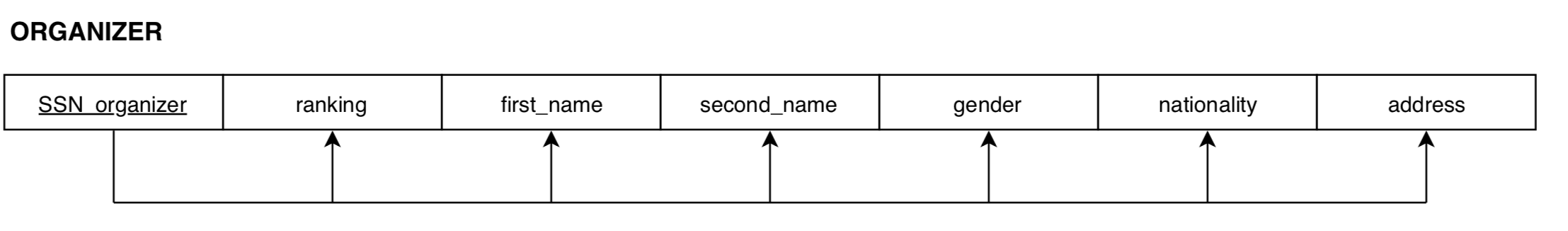
*Define (“design”, “validate”) the relationships of interconnection and access (‘attribute dependencies’) among entities. Validation of the logical design shall be conducted by using functional dependency theory and normal forms.7 In particular, for each logical schema, i.e., SQL table, produced by Physical design in ERWin, follow these requirements: (a) Rewrite it with its proper unique name (“EMP PROJ”) and primary key (underlined at- tribute(s)) by using the standard notation as indicated here as a sample; (b) Indicate all functional dependencies defined on the schema; (c) Indicate the highest normal form (among those discussed in class) that the schema satisfies. (d) For the five most important schemas, write (below the schema) the SQL table generated from Physical level design (Physical Diagram) by ERWin’s forwarding engineering.*



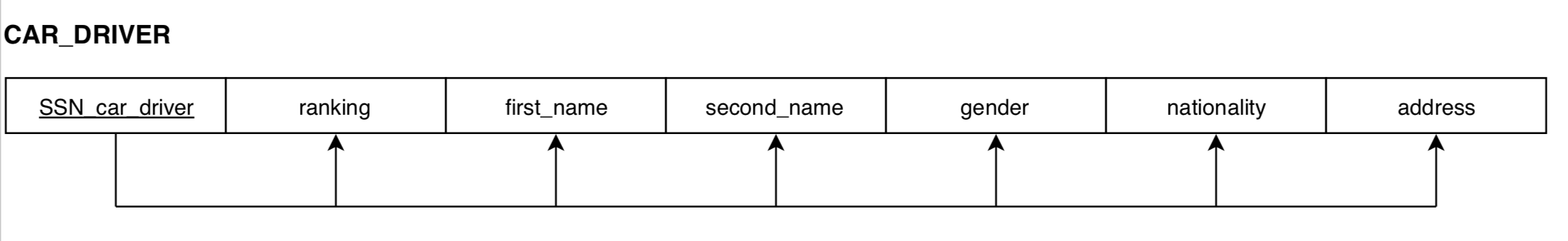
FD1: {SSN\_host}->{ranking, first\_name, second\_name, gender, nationality, address}



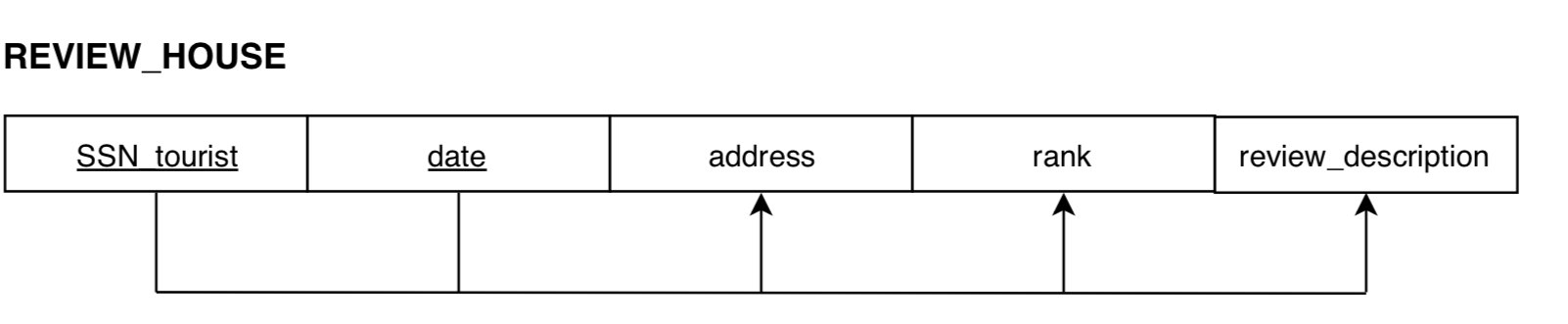
FD1:{SSN\_tourist} ->{ ranking, first\_name, second\_name, gender, nationality, address }



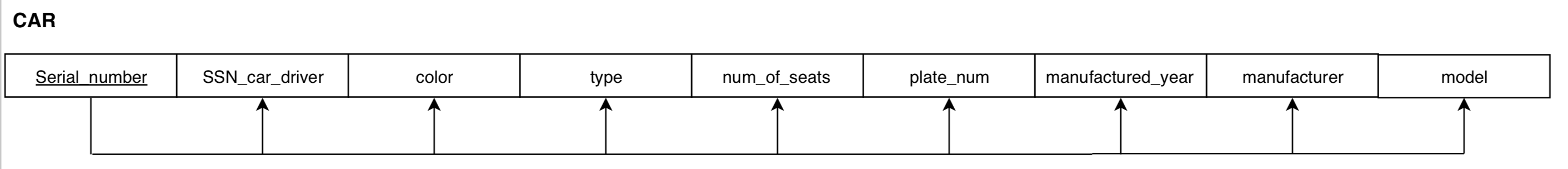
FD1:{SSN\_organizer} ->{ ranking, first\_name, second\_name, gender, nationality, address }



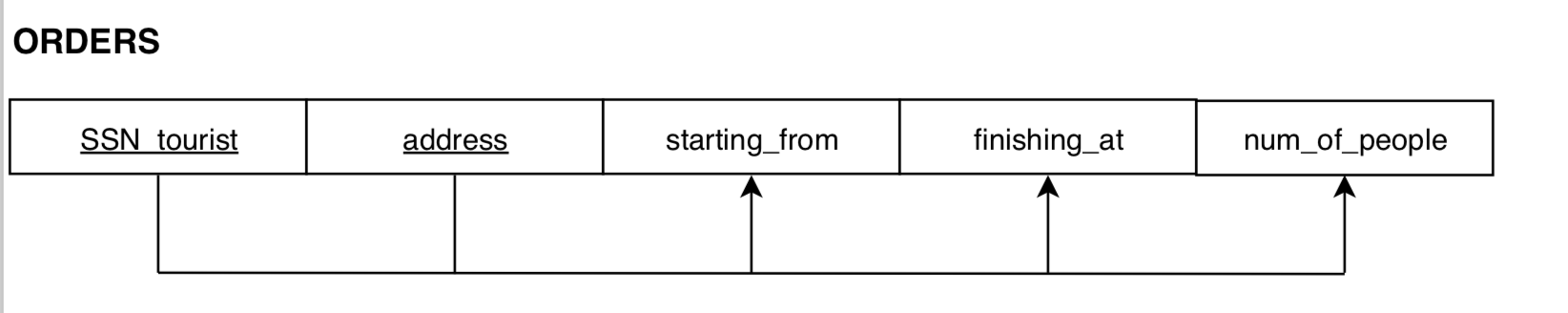
FD1:{SSN\_car\_driver} ->{ ranking, first\_name, second\_name, gender, nationality, address, lic\_number }



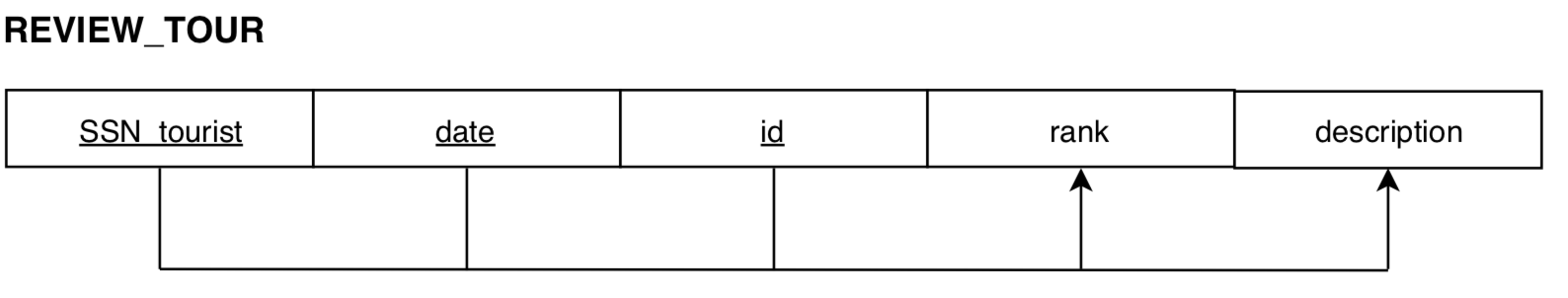
FD1:{SSN\_tourist,date}->{address, rank, review\_description}



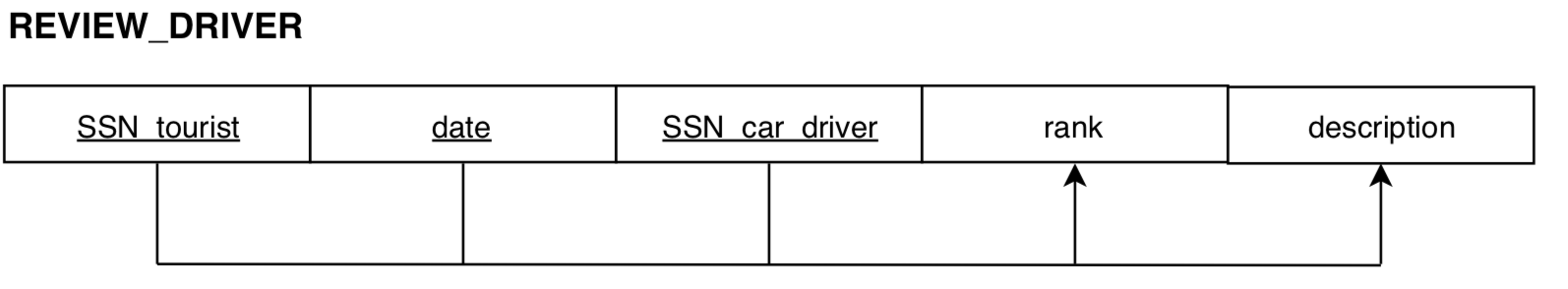
FD1:{serial\_number}->{SSN\_car\_driver,color,type,num\_of\_seats,plate\_num,manufactured\_year,manufacturer,model};



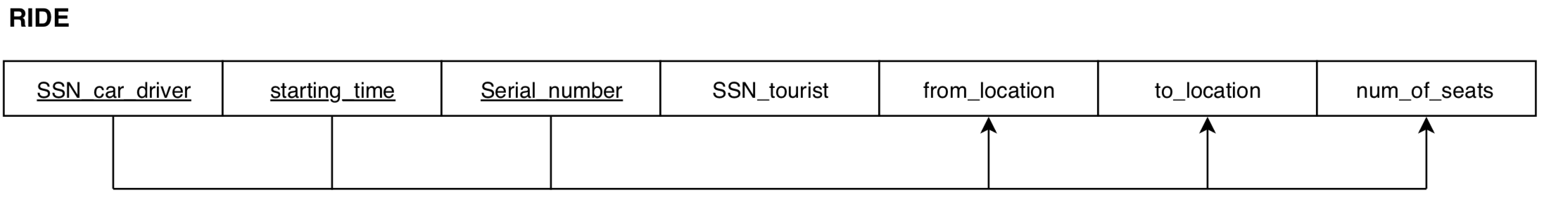
FD1:{SSN\_tourist, address}->{starting\_from, finishing\_at,num\_of\_people};



FD1:{SSN\_tourist,date,id}->{rank, description};

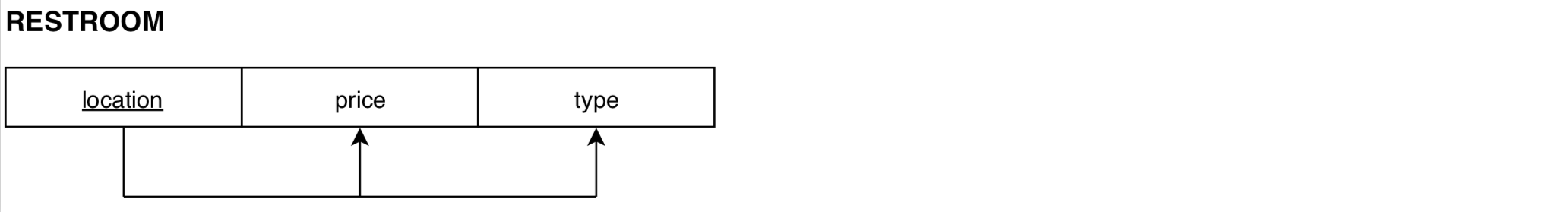


FD1:{SSN\_tourist, date, SSN\_car\_driver}->{rank, description};

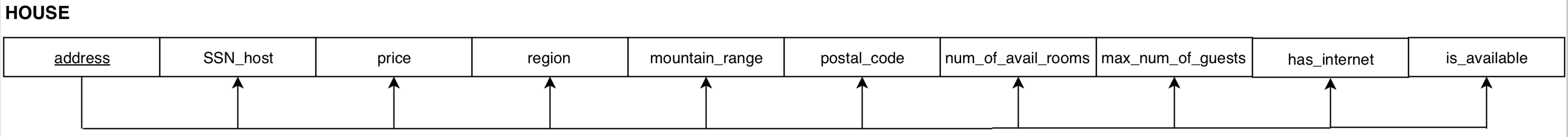


FD1:{starting\_time,SSN\_car\_driver,serial\_number}->{SSN\_tourist,from\_location,to\_location,num\_of\_seats}

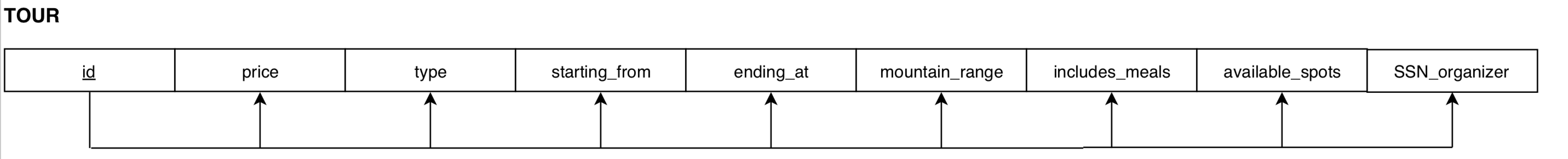
**Note:** here primary key of the table does not determine SSN\_tourist, so that SSN\_tourist can have many different values for one primary key (or for one ride).



FD1:{location}->{price, type};



FD1:{address}->{SSN\_host, price, region, mountain\_range,postal\_code,num\_of\_available\_rooms,max\_num\_of\_guests,has\_internet,is\_available}



FD1:{id}->{price,type,starting\_from,ending\_at,mountain\_range,includes\_meals,available\_spots,SSN\_organizer}

## 5.1 Normalization

**USER**

|  |  |  |
| --- | --- | --- |
| SSN | {Phone\_number} | {Language} |

Since this table holds multivalued attribute, we should use 1NF technique.

**USER**

|  |
| --- |
| SSN |

**USER\_PHONE**

|  |  |
| --- | --- |
| SSN | Phone\_number |

**USER\_LANGUAGE**

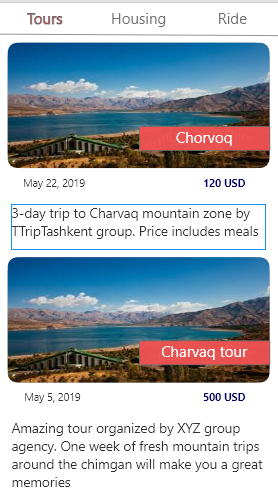
|  |  |
| --- | --- |
| SSN | languages |

Having normalized USER table, we got these tables above.

Other tables are correct and already meet the requirements of NF1, NF2, NF3, BCNF

# 6. Interface

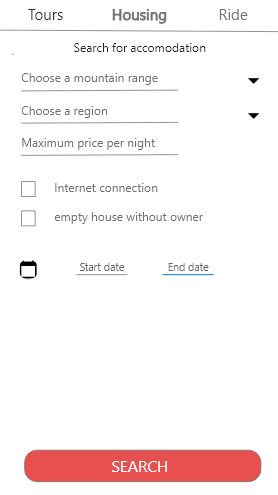
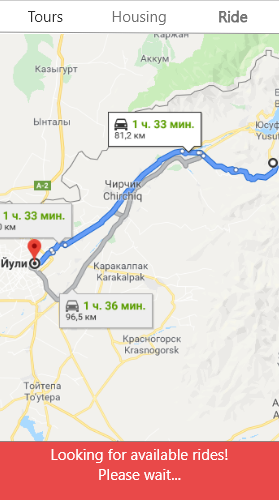
*Introduce and describe the (design of the) basic elements of the application developed on top of the database designed above. Briefly explain the technical elements of the application’s function- alities and user interface. Embed some screen-shots representing the main functionalities listed in 3(b) above. In particular, (a) Embed a screen-shot showing an “Insert” of some data. (b) Embed a screen-shot showing an “Deletion” of some data. (c)  Embed a screen-shot showing a “Search” (query) over some data. (d)  Embed a screen-shot showing the specific functionalities of the system (if any). “Specific” here means functionalities that go far beyond the typical transactions of dabatabase systems like add/delete/update/select. For example, data analysis, learning, interactive/game-based interface, etc.) Note: The definition (“design”) of the UML component diagram for the proposed software in- terface is considered a plus.*



The first screenshot is the flag page. In the second window, it can be seen that there are two three sections: Tours, Housing and Ride. All these three sections are buttons that will navigate through their window. So, Tours window lists the available tours with their dates, price and tour description. User will slide down the window and choose the one that he/she likes.



As the user chooses the tour, he/she will be offered the available houses and shared rides from that location. As the user presses those buttons, application will navigate next windows.



To search housing, user taps housing button, where will be asked to enter the mountain range, region in dropdown box and maximum price per night into input fields. Also, internet and house without owner (owner will live in a separate house) can be chosen using a tick. Dates are indicated in the input fields to check the availability of the house.

The ride choice has the same principle as Google Maps. The location from and to are indicated in the map. After, all the available rides will be shown with the detailed information.

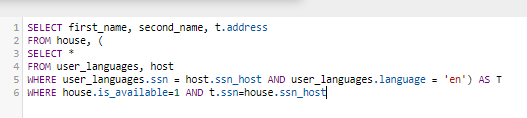
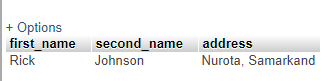
# 7. Testing of Database

*This section should contain an enumerated list in increasing order of complexity of the five most representative queries that a user might run on the database by using the proposed application. For each item of the list, i.e., for each query, do: (a) Write it in plain English. (b) Write it by SQL standard (Silberschatz’SQL; preferred) or by SQL you executed within phpMyAdmin of your DBMS. (Note: you are free to use any relational DBMS, provided you can run phpMyAdmin on top of it as interface.) (c)  Embed phpMyAdmin’ screen-shot of the query and its result after execution. (\*) (d) Embed Interface’ screen-shot of the query and its result after execution (if any). (\*) The DB have to be “populated” with data in order to test the queries. The amount of sample data necessary depends on the type and complexity of the queries.*

1. Show all hosts of available houses who can speak English

**SQL**:

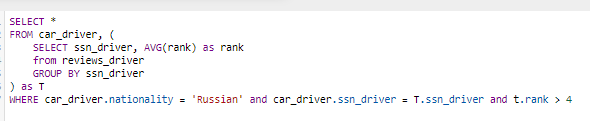
SELECT first\_name, second\_name, t.address  
FROM house, (  
 SELECT \*   
 FROM user\_languages, host  
 WHERE user\_languages.ssn = host.ssn\_host AND user\_languages.language = 'en') AS T  
WHERE house.is\_available=1 AND t.ssn=house.ssn\_host

  
  
**Output**:  


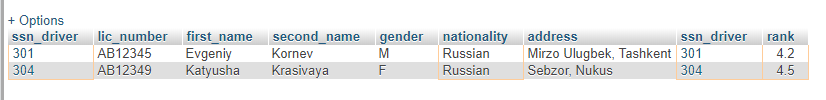
2. Show all the rides whose shared by Russian speaking drivers with at least 4.0 ranking

**SQL:**

SELECT \*   
FROM car\_driver, (  
 SELECT ssn\_driver, AVG(rank) as rank  
 from reviews\_driver  
 GROUP BY ssn\_driver  
) as T  
WHERE car\_driver.nationality = 'Russian' and car\_driver.ssn\_driver = T.ssn\_driver and t.rank>4



**OUTPUT:**



3. Show all available houses that have internet and can accommodate at least 5 people

**SQL:**

SELECT \*   
FROM house  
WHERE house.has\_internet = 1 AND house.is\_available = 1 AND house.max\_num\_of\_guests>4



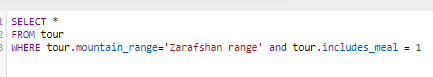
**OUTPUT:**



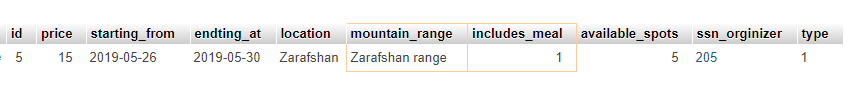
4. Show tours in Zarafshan range that includes meal

**SQL:**

SELECT \*   
FROM tour  
WHERE tour.mountain\_range='Zarafshan range' and tour.includes\_meal = 1



**OUTPUT:**



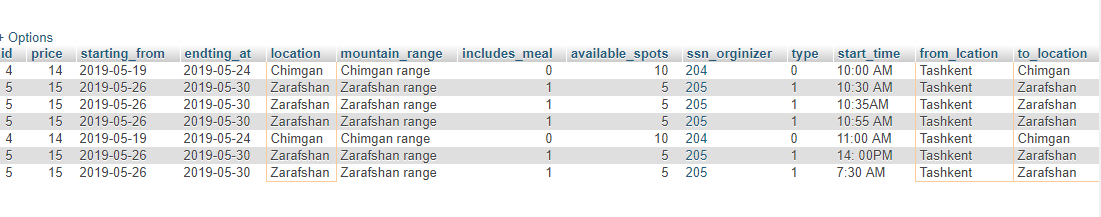
5. Show tours that has corresponding rides leaves from Tashkent

**SQL**:

SELECT \*   
FROM tour, ride  
WHERE ride.from\_lcation = 'Tashkent' AND tour.location = ride.to\_location



**OUTPUT**:



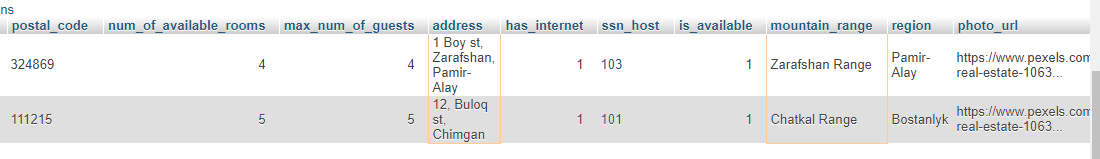
6. Show houses at Chatkal or Zarafshan range that are with photos

**SQL:**

SELECT \*   
FROM house, photos   
WHERE house.address = photos.address AND (house.mountain\_range = 'Zarafshan Range' OR house.mountain\_range = 'Chatkal Range')



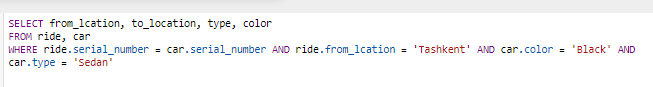
**OUTPUT**



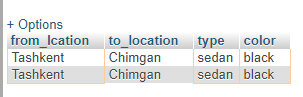
7. Show rides from Tashkent with black sedans

**SQL:**

SELECT from\_lcation, to\_location, type, color  
FROM ride, car  
WHERE ride.serial\_number = car.serial\_number AND ride.from\_lcation = 'Tashkent' AND car.color = 'Black' AND car.type = 'Sedan'



**OUTPUT:**



# 8. Bibliography

|  |  |
| --- | --- |
| [1] | [Online]. Available: https://www.people-travels.com/about-uzbekistan/uzbekistan-tourism.html. |
| [2] | [Online]. Available: https://www.people-travels.com/about-uzbekistan/uzbekistan-mountains/. |
| [3] | [Online]. Available: https://slideplayer.com/slide/6021012/. |
| [4] | "Airbnb," [Online]. Available: https://www.airbnb.com. |
| [5] | "Uber," Uber Technologies Inc., [Online]. Available: https://www.uber.com. |
| [6] | "Travello," [Online]. Available: https://www.travelloapp.com. |
| [7] | [Online]. Available: https://slideplayer.com/slide/6021012/. |