

Description of Data Base

Krzysztof Piotrowski

22nd of November

1 Textual description of a database

1.1 Aim

The main aim of this database is to manage travel agency's system. First of all, it keeps data about transaction between client and company. Secondly, it covers information about trips in every detail (including destination, time, place, transport issue and trip's hotel). Lastly, it is designed to administrate employees' data.

1.2 Description

Travel agency offers various types of trip to different countries. For each trip we store the following: destination, starting and ending dates, price, and maximum number of people that can take part in trip. Additionally, each trip is associated with information about hotels. Trip can have several hotels and hotels can be associated with different trips, each of that hotel contains information about name, number of stars and exact address location. Also, trip is associated with transport details, in particular: how client can get to its destination (airplane, bus, etc.) and exact place and time of departure. Lastly, trip has assigned guide, who is a supervisor of each trip, that take care for each client during their residence. Only one guide is assigned to trip, and each guide can carry on only one trip. There are two more group of people that database is aware of: clients and agents. Agents is special type of worker that sell trips to various clients. Company note each transaction and check if client already paid for the trip. System stores personal details of each person involved in system: they name, surname, mail, telephone number, and address. Each address consists of a street, street number, zip-code, city, and country. Optionaly, It can have apartment number (if occurs). Each trip, person, hotel, transport detail can be located in system using special numeric ID assigned to each one of them.

1.3 Objects - entities and attributes

Travel Agent

- name
- surname
- mail
- telephone
- agent ID

Guide

- name
- surname
- mail
- telephone
- guide ID

Customer

- name
- surname
- mail
- telephone
- customer ID

Trip

- destination
- price
- starting date
- ending date
- max. num. of participants
- trip ID

Transport

- mean of transport
- place
- date
- transport ID

Hotel

- name
- stars
- hotel ID

Transaction

- payment status
- transaction ID

Address

- street
- street number
- apartment number (optional)
- zip code
- address ID

City

- country
- city
- city ID

1.4 Requirements concerning data

1.4.1 Data types

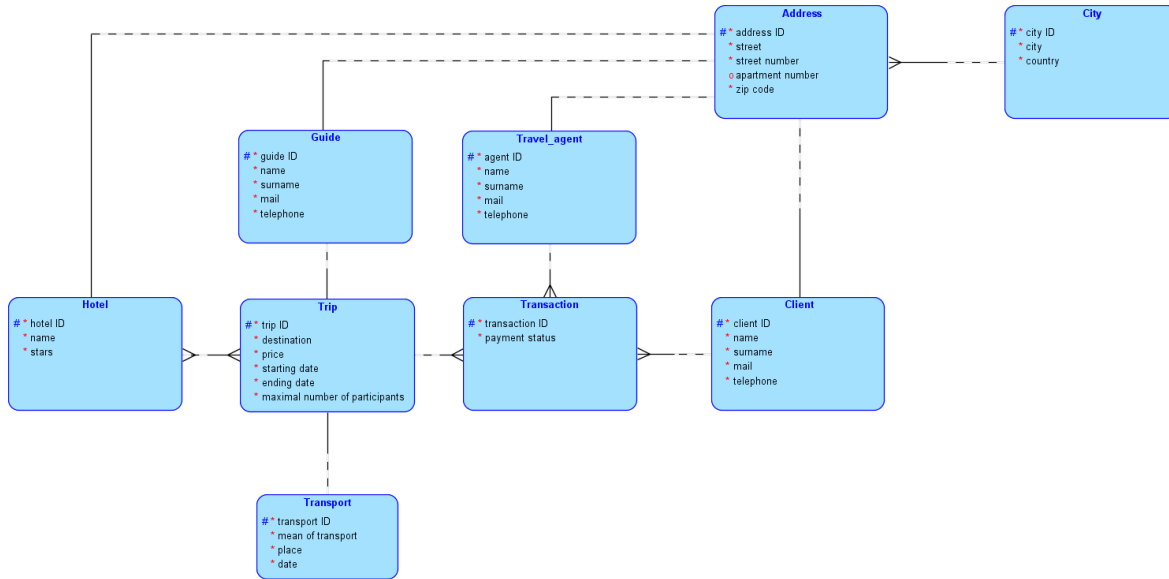
For most of attributes the **VARCHAR2** will be used sizing from 8 (for zip-code) to 20 (for the rest of attributes). Numerical values such as telephone, number of stars or IDs will use **NUMBER** data type. Size will be 8 for IDs, 13 for telephone or 1 for number of stars. Additionally, there is price which also uses **NUMBER** but with precision point. Its size is set to 7 digits, where 2 are marked as precision point. For any kind of dates, the **DATE** data type will be used. Also, in database there is one boolean value - it is represented by **CHAR** data type.

1.4.2 Mandatory of attributes

In this database, almost all of the attributes are mandatory. However, there is one exception - apartment number in address entity is optional.

2 ER Diagram

2.1 Diagram

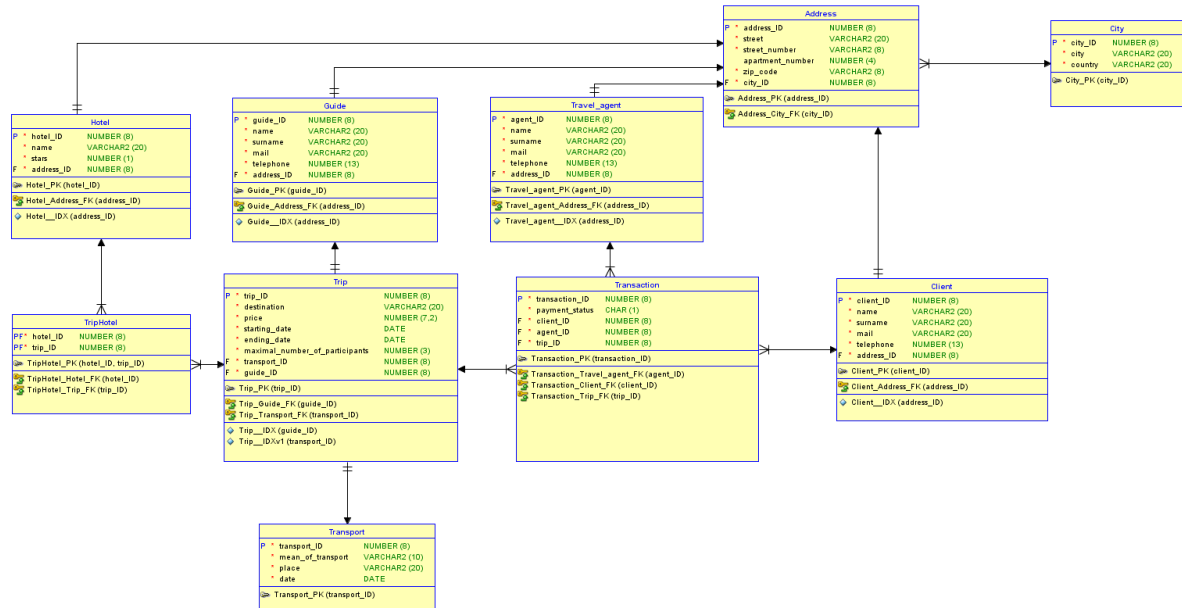


2.2 Description of entities and their relations

Each entity will have unique ID to make management easier. Database has three person entities: travel agent, guide and customer. For each of them we store basic information: name, surname and contact details (mail and telephone). Because address is more complex, it was created as separate entity. Each person entity is related to address entity. Address entity stores information about: street, building number and zip code. It is also related to city entity, which consists of country - city combination, to make the storage easier. Database connects 3 entities: client, travel agent and trip via transaction entity. Transaction entity stores information about payment status and thanks to that it allows client to have more than one trip, trip to have many participants, and travel agent to sell many trips to many clients. For each trip we store information about price, destination, duration of the trip, maximal number of participants. Additionally, it is connected with hotel entity. Hotel entity gathers information about the number of stars and also is related with address entity to know location of a hotel. Trip can have many hotels (e.g. if there is a trip around one country) and hotels can have many trips (e.g. if there are two different trips to the same location). Despite connection with hotel or transaction entities, it has obligatory connection with guide entity. Each trip must have one guide and guide can have zero or one trip. Additionally, trip must have mandatory relation with transport entity. Each transport entity covers information about mean of transport, place and date (including time) of departure.

3 Relational Schema

3.1 Schema



3.2 Description of a way of mapping an ER-diagram into a relational schema

When all relationships between entities are declared, they must inherit proper keys from other entities as foreign keys. In my example, each person entity and hotel entity has address ID as foreign key which direct them to their address. Trip knows his guide ID and transport ID. In transaction entity we have 3 IDs to know who sold, who bought trip and for which trip. Lastly, address has also city ID to know in which city and country this address occurs. Because we had special type of relational (M to N) between trip and hotel, new table (sometimes called proxy) appeared which takes both related entity keys and change M:N relation into two 1:N relations. It enables trips to have many hotels and hotels to have many trips.