Smart Traffic Management Database

Chrysikos Christos, 9432 | Mavridis Antonis, 9563 | Gitopoulos Giorgos, 9344

Databases @ ECE AUTh, November 2021

1. Introduction

This project deals with the theoretical design of a database of a proposed application. After the description of the application and its data requirements, we will present the structure of the database and its user types. Moreover, some examples will be used in order to explain the database functionality.

1.1. Application Description

The proposed application concerns *traffic management* in a *smart city*. A *smart city* is a technologically modern urban area that uses different types of electronic methods, voice activation methods and sensors to collect specific data. That data is used to improve the operations across the city, such as traffic management.
The purpose of the application is to match each vehicle with the fastest feasible route in order to reach the desirable destination and help the drivers to locate available parking slots easier. Also, it aims to detect traffic violations and store them. First of all, we assume a computing cloud that will perform all the necessary computations for the application. In addition, a database will be needed to store all the collected data.
Concerning the application, the exact position of each vehicle will be collected by GPS, it will be stored in the database and a traffic metric will be computed for each area of the city according the number of vehicles in the area. Every driver will select its destination and the cloud will export the optimal route for this vehicle, using a *Shortest Path Algorithm* (i.e. *Dijkstra's Algorithm*) and taking into account the neighboring areas, the traffic metric of each area, the traffic lights and the tolls. Also, vision sensors will be placed in every parking slot and the data of their status will be stored to the database. So, the drivers will be able to find the empty parking slots in a specific area using the application. About the traffic violations, cameras will be placed through the city streets and in the traffic lights too, in order to detect speed limit or road signs violations and violations of red traffic lights respectively.

1.2. Data Requirements

Some of the data that the database should store, with the corresponding estimations of storage size per day, are:

- the users with their id, name, age and gender (2,000,000 $\sim$ 10,000,000 users),
- the vehicles with their licence plate, type, driver id, location, destination and route ($\sim$ 1 vehicle per user),
- the areas of the city with their id, name and traffic metric (20 $\sim$ 50 areas),
- the couples of neighbouring areas in order to declare the routes ($\sim$ 3 neighbouring areas per area),
- the routes that are exported by the cloud (as a sequence of areas *) with their id and duration ($\sim$ 3 routes per vehicle),
- the traffic lights with their details (5,000 $\sim$ 20,000 traffic lights),
- the tolls with their details (3 $\sim$ 10 tolls),
- the parking slots with their details (5,000,000 $\sim$ 20,000,000 parking slots)
- the traffic violations and their details (5,000 $\sim$ 20,000 violations)

* Note that the smaller each area is, the more precise the route is. For simplicity we assume larger areas of a city.

2. User Categories & Requirements

In this section, we will present the types of users of the database and their corresponding data access requirements.

- **End user:** The end users of the application are the drivers of the vehicles. They can access all the data of their own vehicle, their own route, the data of every area of the city, the tolls, the traffic lights and the data of every parking slot. They are able to write in the database just their own destination, while all the other data are read - only for them.
- **Administrator:** The administrators can read and modify all the data of the database. They are responsible for the proper operation of the system and they should take action if they detect any type of problem.
- **Sensor - Embedded system:** The several sensors with their embedded systems will be considered as users of the database, as they are able to write and modify data. For example, a system of a parking slot sensor updates the status of the slot in the database, every time that it changes and the embedded system of a camera writes down the plate number of a vehicle and its fee in the violations table, when it detects that it violated the speed limit. The embedded systems of the sensors have no access to other data and cannot read data at all. They can just write in or modify some specific memory addresses.

- **Cloud:** The computing cloud has access to all the data concerned to the route computation. It can read the current location of every vehicle to compute the traffic metric of every area as a fuzzy variable and writes it in the table of the areas. Also, it reads the location of the traffic lights, the tolls, the destination of every vehicle, the neighbouring areas and the computed traffic metric of each area in order to compute the fastest route for each vehicle. When it does that, it writes the route id in the corresponding line of the vehicle table, the route id and its duration in the route table and the areas that the route passes through in the route - area table (Note that the tables will be presented in depth in section 4).

3. Entity – Relationship Model

3.1. Description

TEXT HERE

3.2. Entities

TEXT HERE

3.3. Relationships

TEXT HERE

3.4. Entity – Relationship Diagram

TEXT HERE

4. Relational Model

4.1. Data Domains

4.2. Relations

4.3. Relational Diagram

4.4. Views

5. Examples

5.1. Table Examples

5.2. Relational Algebra Examples