

Homework

SAB – 2017/2018

System description

Student's goal is to build a system for package transport. The system keeps records about cities (name, postal code), district (name, city, x-coordinate (in km), y-coordinate (in km)). For users system records their first name, last name, username, password and number of sent packages. Some users are administrators. User can become courier by sending a request to administrator (request, apart from user information, should also contain the information of a vehicle that user intends to drive). Administrator may accept or refuse those requests (requests are not permanently persisted). Every vehicle is persisted with its licence plate number, its fuel type (0 - "propane", 1 - "diesel", 2 - "natural gas") and its fuel consumption (liters per kilometers). Once administrator has accepted the request, so user becomes courier (only one person at the time might drive a single vehicle, a person might only drive one vehicle at the time). Previous action should be done inside stored procedure(s). For every courier system keeps track of their vehicle, number of delivered packages, their profit, and a status (0 - "not driving", 1 - "driving").

Transport of a package is done by sending a request including sender information, district of departure and district of arrival, type of package (0 - "letter", 1 - "standard", 2 - "fragile") and its weight in kilograms (might be decimal value). Afterwards courier, with "not driving" status, give their transport offers. Offer is represented as a percentage of base price of delivery (e.g. 10 - represent 10% of base price, base price is calculate with function F given below), information about courier and information about the package that offer is for. User who has sent a request for transport, chooses one offer. After user choice, all offers for that request (trigger(s) with prefix $TR_TransportOffer_)$. Every package also contains courier's information, status of delivery (0 - "created", 1 - "offer accepted", 2 - "delivery in progress", 3 - "delivered"), price and offer accepted time. Each drive consist of packages for which an offer has been accepted. Before starting a new drive, couriers takes packages and creates deliveries from them. Courier is delivering packages in FCFS order. Once the drive has finished, the courier profit has to be calculated. Profit is difference between the sum of prices of all deliveries and amount of money spent on fuel for that drive (fuel prices: "propane" - 15/litre, "natural gas" - 36/litre, "diesel" - 32/litre). Drive starts on picking up first package and end after delivery last package.

$$F : (INITIAL_PRICE[i] + (WEIGHT_FACTOR[i] * weight) * PRICE_PER_KG[i]) * euclidean_distance$$

	Letter	Standard	Fragile
Initial Price	10	25	75
Weight Factor	0	1	2
Price Per KG	/	100	300

It is desirable that there when its possible referential integrities be: ON UPDATE CASCADE, ON DELETE NO ACTION. For that reason, method that deletes certain row from table should not delete other rows from other tables which references row for deletion. Other rows should first explicitly delete via other methods (if that method doesn't exist use referential integrity ON DELETE CASCADE).

Every Id column which primary key is not foreign key should be IDENTITY column.

Use type DECIMAL(10,3) for real numbers. Default maximal length of all textual columns in tables is 100 characters, unless otherwise stated.

Requirements and testing

It is necessary to build database that is defined under System description section.

It is necessary to write classes with prefix **lfynnnn_** which implements all interfaces and inside it you need to implement all interfaces methods. Classes can contain other methods and fields but it is necessary to implement interface methods because they will be tested.

There is few public tests and couple of private tests. Achieved points on homework are formed using next formula:

$$AchievedPoints = (PointsFromPublicTests + PointsFromPrivateTests) * modificationFactor$$

Homework is worth 20 points (public tests – 10 points, private tests – 10 points, modificationFactor between 0 and 1).

Structure of solution

You need to upload zip archive with name **lfynnnn.java** on link https://rti.etf.bg.ac.rs/domaci/index.php?servis=SAB_domaci_1718. Inside zip archive need to be:

1. Classes with prefix **lfynnnn_** – implementation of interface (need to be in packet **student**)
2. modify class **studentMain.java**
3. Other java classes and packets, if they exist (need to be in packet **student**)
4. **lfynnnn.erwin** i **lfynnnn.png** – database model made in tool ERwin Data Modeler r9.7
5. **lfynnnn.sql** – SQL file made by ERwin Data Modeler + all other SQL code (triggers, stored procedures, functions and other SQL code) including triggers from **lfynnnn-triggers.sql** and stored procedures from **lfynnnn-stored_procedure.sql**
6. **lfynnnn-triggers.sql** –SQL file with triggers with prefixes **TR_TransportOffer_** and all the others.
7. **lfynnnn-stored_procedure.sql** –SQL file with stored procedures
8. **lfynnnn.bak** – database backup file

Deadline and other information

1. Above mentioned naming convention for files has the following definition: l – first letter of last name, f – first letter of first name, yy – last two digits of registration year, nnnn – index number
2. Deadline to upload homework is friday 22.06.2018. at 8AM
3. Testing of homework will be organized 22. i 23.06.2018,
4. Use JPA or JDBC to communicate with database.
5. Usage of MS SQL database is required