Projet : Mission Management System for a Company

Lpro CNMS — SGBD — 2018/2019

Objectives

Modelizatin and creation of a database for managing a company's employees missions. Development of scripts allowing maintenance, edition and queries on the database.

1. Description of the project

1.1. Objectives

The purpose of the project is to create a system to manage a company's employees missions. This project requires the modelization of the database, its implementation and its edition. Scripts allowing modification, visualization and maintenance will be required.

This document describes the company's need, the steps required to achieve the project, and the evaluation.

Each employee works and is associated to a service, and each service has a head. Employees missions must be validated by the head of the service, who's the only person able to validate these missions.

1.2. Company's needs

In the company all employees work in services. Each service is managed by a head, helped by an assistant. The company has to store for each employee some personal information: address, phone, social security number, date of birth, emergency contact, etc. There are several staff categories: employees, service heads, director, system administrators. Employees have as a direct supervisor their service head, and service heads have the director as a direct supervisor. System administrators are a separate category, belonging to the *informatics* service. They have specific rights on the company's data, and have as a direct supervisor the informatics service head. Services are defined with a name, an email and an address. Each service has a head and an

assistant. The company has a certain amount of service cars, for each has to be stored: the car registration, the brand, insurance contract number, date of the last technical control, amount of kilometers.

Employees regularly have to leave for missions. The missions have variable length (from a couple of hours to several days), variable places (in France or abroad). Each mission is defined by:

- Beginning and end of the mission
- Purpose of the mission
- Destination address

On top of these information, for each mission has to be known: the transport, accommodation, and other costs. Transport can be: plane, train, bus, car rental, the company's service cars. Transport can be a combination of those (e.g. plane and car rental). In each case, the following information will have to be stored:

- Plane : ticket reference, flight number, departure date, arrival date, company, seats, options
- Train : ticket reference, train number, departure date, arrival date, company, seats, options
- Car rental : rental reference, company, car registration, rental dates, cost
- Service car : reference of the car, traveling dates, kilometers driven, comments

For each mission, the employee will have to make sure it is approved before departure. When returned, they need to provide invoices justifying their expenses, that will also have to be validated. If all is validated, the employee will be refunded the costs. A mission can have several statuses:

- Approbation : approved or non approved - will indicate if the mission has been approved by the n+1 before departure
- Status : on hold, validated, rejected

Once the mission validated, information about the mission costs will have to be stored, with the invoices. Each invoice will contain an object, a date, a cost and an issuer. The expense report contains all the invoices, corresponding to the total amount of expenses and a status (*on hold* or *paid*).

Administrators and service heads are also employees: they can enter their own missions on the system and they need someone to validate them.

1.3. Required scripts

You will write scripts containing queries allowing to perform actions described in the following table:

Table 1. Scripts attendus

User category	Actions
Employee	View his/her missions and their status
	Create new missions
	View, modify, delete ongoing missions
	View, modify, delete personal information
Head of service	Same scripts as employees
	View all missions of the service's employees and their status
	View, modify and delete missions of an employee of the service
	Validate or reject a mission of an employee of the service
	View, modify, delete personal information of an employee of the service
Administrator	Same scripts as head of services (for all services)
	View, create, modify, delete employees personal information

User category	Actions
	View, create, modify, delete head of service personal information Also, provide queries to: * Display the list of missions needing validation * Display the list of missions on hold * Show cars which technical control has been done more than a year ago * show cars with more than 50000km done * For each service: how many missions have been validated, how many are ongoing, how many have been rejected * Show all the employees who got more than 2 missions rejected Also, write transaction scripts to: * Create expense reports * Add a new mission

1.4. Optional functionalities

Following functionalities will give you extra points. Difficulty is given with stars (*, easy, * * * *, difficult)

Table 2. Optional functionalities (pick one or more)

Name	Difficulty	Description
Static web interface	* *	HTML pages allowing to view tables
Dynamic web interface	* * *	Pages allowing to add dynamically data in the tables
Constrained update of the database	* *	Add constraints to data given by users via the HTML interface

Name	Difficulty	Description
Holidays	* * *	Employees can store their holidays on the system. Missions can't happen durng holidays.

1.5. Documentation

You will document your database: describe the modelization and creation of your databas. Also describe and comment your scripts.

2. Implementation

You must use the implementation framework provided during the labs. Your database must be PostgreSQL.

2.1. Modelization of the database

The purpose of this step is to modelize and implement your database. To do so, you need to follow the steps given in the lectures:

- Entity-association model
- Logical model
- Implementation
- Insertion of data in the tables

The evaluation will be based on these elements. Make sure you define everything carefully, paying attention to constraints, primary and foreign keys. Describe these steps in the documentation.

2.2. Add data in the tables

The file donnees-missions.csv in chamilo's documents gives you an example of data you can store in your database. These data are not necessarily in the same format your tables are, you will need to edit them before insertion. Scripts allowing that will be evaluated.

2.3. Scripts SQL

Give SQL scripts allowing to perform Required scripts.

2.4. Optional functionalities

If you pick some of the optional functionalities, you can use frameworks such as bootstrap (http://getbootstrap.com/).

Last updated 2019-02-10 21:46:52 CET