CS307 Project2 Database Application

> Main Contributors:
>
> ZHU Yueming
>
> Review: MA Yuxin

1. Basic Requirements: (65%)

1.1 API Specification：

To provide basic functionality of accessing a database system, you are required to build a backend library which exposes a set of application programming interfaces (APIs). The general descriptions for each API are listed below.

1. Add, modify, delete a station.
2. Add, modify, delete a new line.
3. Station and line:
    - Place one **or more** stations at a **specified** location on a line.
    - Remove a station from a line.
4. Search the name of the stations that are the n-th station ahead and the n-th station behind a specific station on a line.
5. Boarding functionality for a passenger or card
    - In addition to passenger or card information, it is necessary to record the starting station and boarding time(start_time).
6. Exit functionality for a passenger or card.
    - In addition to its bording information, it is necessary to record the destination station, exit time(end_time), and calculate the price based on **Price.xlsx**.
7. Can view all information about passengers or cards who have boarded but have not yet exited at the current time.

1. 添加、修改、删除一个地铁站
2. 添加、修改、删除一个地铁线
3. 地铁站与地铁线的关系：
    - 将一个或多个地铁站在放入一个地铁线中指定的位置上。
    - 将一个地铁站从一个地铁线中移除
4. 查询某一个地铁站在一个地铁线路上向前数第n站与向后数第n站的地铁站名字
5. 实现乘客或公交卡上车功能。
    - 这里除乘客与公交卡信息之外，需要记录始发站与上车时间
6. 实现乘客或公交卡下车功能。

    - 这里除乘客与公交卡信息之外，需要记录终点站与下车时间，根据Price.xlsx信息计算票价
7. 查询当前时间已经上车但是还未出站的乘客与公交卡信息。

1.2 Functional Requirements:

- It is required to use a **general-purpose programming language** which can interact with the **database** to fulfill all the requirements mentioned in Section 1.1.
- To test all the APIs with necessary input data and display the result set, you should provide a type of **interface to interact with your program**. The interface can be:
    - Command-line-based application for input and output.
    - HTTP/RESTful interface services.
    - GUI-based desktop application.
    - Webpage-based.
- **Prepare testing data**: The json files in project 1 with the Price.xlsx as the testing data and store them in the database for project 2.

2. Advanced Requirements: (30%)

> If you would like to get the full mark for any advanced requirement, please try to demonstrate yourself by providing a high-quality solution.

- Complete the project using Open Gauss database or Mysql database. (up to 3%)
    - https://edu.hicomputing.huawei.com/teaching
    - https://www.modb.pro/db/611481
- Based on the basic requirements in Section 1, further enhance the usability of the APIs to accept more flexible types of requests. You may think about more requirements than those proposed in Section 1 and implement the new requirements. Such as: (up to 15%)
    - Support path queries between two stations.
    - Add and appropriately utilize the status of stations, such as under construction, operational, closed, etc.
    - Business carriage in the subway.
    - Establish a comprehensive system to integrate buses and subways.
    - Implement a multi-parameter search for ride records. For example, enable searching ride records based on 1-n parameters like subway stations, passengers, time periods, etc.
    - Other functionalities.
- Encapsulate the features and implement a real back-end server instead of several independent scripts or programs. In the server design, consider the following technologies: (up to 12%)
    - Package management
    - Using sockets or HTTP/RESTful Web
    - Using connection pools
    - Using backend frameworks or ORM mapping, etc.
- Page display design (up to 4%)
    - A usable and beautiful GUI design or webpage design for data presentation. (up to 4%)
    - Giving a wonderful input and output format display based on the command line. (up to

2%)

- Appropriately utilize database user permissions, procedures, indexes, views, triggers, and other functionalities. (up to 5%)
- Big data management. (Up to 5%)
- Support high-concurrency with proper pressure tests. (up to 8%)
- Effective presentation and communication. (up to 3%)

1. 使用到了open gauss或者MySql数据库完成项目。
2. 在保证Section2需求的基础上，进一步完善API设计，设计出更多的系统功能性需求。
   - 支持两个地铁站之间的路径查询
   - 增加并合理的使用地铁站状态，例如：建设中、运营中、关闭中等
   - 商务车厢
   - 建设完整的公交、地铁相结合的体系。
   - 多参数搜索乘车记录功能。例如： 通过地铁站、乘车人、时间段等实现1-n个参数搜索乘车记录功能。
   - 其他功能
3. 封装并实现一个真正的后端服务器，而不是几个独立的脚本程序。服务器设计中，可以考虑下面技术：代码包管理、使用套接字的通信 或HTTP/RESTful Web、使用连接池、使用后端框架、使用ORM映射等。
4. 页面显示设计。
   - 展示数据时，提供实用性强有好看的GUI页面或网页。
   - 展示数据时，提供一个设计好看的基于命令行输入输出交互的页面
5. 合理使用数据库用户权限、过程、索引、视图、触发器等功能
6. 大数据管理。
7. 支持高并发，并要通过适当的压力测试。
8. 良好的展示与表达能力。

3 Report (5%)

A report no longer than 12 pages. The following content should be include:

- Basic information of your group: Please follow the same requirement as described in Project 1.
- API specification of your code: Please describe the purpose and use of interfaces (you may only use 1-2 sentences for each API in case the report becomes too long). Also, you need to illustrate the types and meanings of the parameters and return values. You can take any API documents of mature open source projects as references of how to organize the specification of your interfaces.
- If you have finished any advanced requirements, describe what you have done and how you did it.

3 What to Submit

Please submit your report and attachments before **23:55 on June 7th 2024**, including:

- All scripts you have written.
- A report with pdf type.