

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 boarding 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

基础部分只考察效果

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。中途添加，后需要后移之类的
- 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。 乘车数据：记录乘车时间。。下车时间。。 price。。

- 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。

可以看在车上的乘客数量

1.2 Functional Requirements:

效率和准确性。。？可以不用关系型数据库以要求数量

- 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. 什么swagger。。什么可以拿来测试
 - GUI-based desktop application. 分类：地铁线、地铁站xxx管理、xxx
 - 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.

需要考虑准确性：在地铁站price等查询之类的时候不能出大错

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> 可以用MyBatis简化
 - <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%) 做出效果才有高分
- 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。 鼓励熟悉新数据库

一些老师
ta想法

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 存库里？否则直接查的时候比较慢？
- 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 在station列加状态并应用，(ex 新加地铁站，并且要有相应的限制功能设计)
- 商务车厢
- 建设完整的公交、地铁相结合的体系。 公交地铁多对多的能力？可以找新数据
- 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。 查询事件记录，根据n个参数进行查询。点进搜索都能搜索出相关记录 但是要设计很多select语句在底层。。要设计一条完美的比较难
- 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。 部署一次，直接使用，不是很多脚本 用package socket通信？ 这个更好，建立请求借口 后端有框架自带连接池？

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。 不能在主键上加index

7. 支持高并发 管理很多乘车记录，添加乘车记录和查询乘车记录不一样。查询？建立索引很快？ 定时把数据迁移到另测试。一个库里，把约束调整一下，可以查询三天以内的数据

8. 良好的展示与表达能力。 用连接池好一点，要写出测试流程。

3 Report (5%)

写得好可以直接给满，写的很糟糕可能会扣

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.