

《数据管理技术》期中大作业

大作业要求

描述

- 本次期中大作业要求同学们从后端数据库的角度，对数据库进行设计，以及对数据库进行优化。本次大作业的目的是让同学们对数据库的基本原理有一个更加深刻的理解，同时也是为了让同学们对数据库的基本操作更加熟悉。
- 本次实验需要针对数据库的CURD操作以及索引进行后端开发，并且向应用层提供RESTful接口服务。（具体的接口要求下面详细说明）
- 作为一个可选项，同学们可以尝试将全部服务部署到docker镜像中。

详细说明

1. 数据库:MySQL以及连接方式

本次大作业的数据库选择MySQL 8.0+版本，同学们需要基于此数据库进行代码开发。

- 我们会为所有同学提供测试的数据以及数据库链接，我们要求大家把数据正确的导入到数据库中
- 开发阶段建议大家使用本地数据库。确保代码没有问题后再连接提供的数据库进行数据导入。
 - 数据库链接以及用户密码：mysql.buaa.seekthought.com:3306 用户名：自己的学号 密码：自己的学号
 - 连接方式：以mycli为例：`mycli -h mysql.buaa.seekthought.com -P 3306 -D db_{学号} -u {学号} -p {学号}`，比如：`mycli -h mysql.buaa.seekthought.com -P 3306 -D db_22061234 -u 22061234 -p 22061234`。

mycli安装方式：pip install mycli

△ 请大家不要使用其他同学的账号密码，仅使用自己的账号进行开发。

- 选择docker镜像开发的同学也只需要提供要求的服务(数据库CURD、索引、RESTful接口服务)，但是要确保自己的镜像可以连接本地数据库以及我们提供的数据库

2. ORM框架

- 是否选择使用ORM框架由同学们自己决定。
- 后端开发语言和框架都由同学们自己决定。

3. 数据集以及索引和触发器介绍

我们本次实验用到的数据集有5个表，分别是employees, departments, dept_emp, dept_manager, titles。具体的表创建定义：

departments

```
create table departments
(
    dept_no    char(4)        not null
               primary key,
    dept_name  varchar(40) not null,
    constraint dept_name
               unique (dept_name)
);
```

employees

```
create table employees
(
    emp_no      int            not null
                 primary key,
    birth_date  date           not null,
    first_name  varchar(14)    not null,
    last_name   varchar(16)    not null,
    gender      enum ('M', 'F') not null,
    hire_date   date           not null
);
```

dept_emp

```
create table dept_emp
(
    emp_no      int            not null,
    dept_no     char(4) not null,
    from_date   date           not null,
    to_date     date           not null,
    primary key (emp_no, dept_no),
    constraint dept_emp_ibfk_1
               foreign key (emp_no) references employees (emp_no)
               on delete cascade,
    constraint dept_emp_ibfk_2
               foreign key (dept_no) references departments (dept_no)
               on delete cascade
);
create index dept_no
on dept_emp (dept_no);
```

dept_manager

```
create table dept_manager
(
    emp_no      int            not null,
```

```

dept_no    char(4) not null,
from_date  date     not null,
to_date    date     not null,
primary key (emp_no, dept_no),
constraint dept_manager_ibfk_1
    foreign key (emp_no) references employees (emp_no)
        on delete cascade,
constraint dept_manager_ibfk_2
    foreign key (dept_no) references departments (dept_no)
        on delete cascade
);
create index dept_no
    on dept_manager (dept_no);

```

titles

```

create table titles
(
    emp_no    int         not null,
    title     varchar(50) not null,
    from_date date        not null,
    to_date   date        null,
    primary key (emp_no, title, from_date),
    constraint titles_ibfk_1
        foreign key (emp_no) references employees (emp_no)
            on delete cascade
);

```

索引和触发器

1. 除了上述的DDL，我们还需要大家在employees表上添加一个额外的索引，添加在first_name字段上。使得通过该字段查询时可以使用索引
2. 关于触发器，首先我们需要新建一个表，用于维护当前部门经理的title信息：

```

create table dept_manager_title
(
    emp_no    int         not null
        primary key,
    from_date  date        not null,
    to_date    date        null
);

```

同时我们需要大家在dept_manager表和dept_manager_title表之间添加一个触发器，该触发器的逻辑是

- 在dept_manager表中添加一条记录的时候，同步在dept_manager_title表中添加一个记：
(emp_no, from_date, to_date);

- 若从dept_manager表中删除一条记录，则从dept_manager_title表中删除对应的记录；

4. RESTful接口

同学们需要提供RESTful接口供前端调用，具体而言，我们要求提供以下4个接口。

- 数据插入接口
- 数据更新接口
- 数据删除接口
- 数据查询接口

假设我们现在操作的表是User，其有两个字段，分别是id:int和name:varchar(10)。

每个接口的具体要求如下：

1. 数据插入接口

- 接口地址：/api/v1/{table_name}
- 接口方法：POST
- body：json格式的数据

Example

```
{
  "rows": [
    {"id": "1", "name": "test1"},
    {"id": "2", "name": "test2"},
    ...
  ]
}
```

该接口必须能够识别json中的数据格式，提取出正确的数据并将其插入到数据库中。该接口将数据插入到table_name指定的表中。

2. 数据更新接口

- 接口地址：/api/v1/{table_name}
- 接口方法：PUT
- body：json格式的数据

Example，为了简单我们默认更新的时候会对一行数据全部更新，即不更新的字段也会被传入

```
{
  "id": "1",
  "name": "test1_new"
}
```

3. 数据删除接口

- 接口地址: `/api/v1/{table_name}/{id}`
- 接口方法: DELETE
- body: none

对于dept_emp表, 其id可以看作是两个字段的组合, 即emp_no和dept_no, 因此我们使用`/api/v1/dept_emp/10001/10001`来表示删除dept_emp表中emp_no为10001, dept_no为10001的行。

4. 数据查询接口

- 接口地址: `/api/v1/{table_name}/{id}`
- 接口方法: GET
- body: none

i) 对于查询请求, `{id}`部分和DELETE类似, 对于dept_emp表需要两个id指定; 这里默认这种查询返回该id对应的全部行数据。

ii) 这里指定一种过滤条件, 对应于SQL中的`where a = b`。即如果我们想查询employees表中id为10001的行, 我们可以使用`/api/v1/employees?id=10001`, 如果我们想查询employees表中first_name为Georgi的行, 我们可以使用`/api/v1/employees?first_name=Georgi`。

对于查询接口只需要提供这两种情况, 且? 指定的条件只有一个, 即我们不考虑`where a = b and c = d`的情况, 等价于访问`/api/v1/employees?id=10001&first_name=Georgi`。

关于RESTful服务的进一步说明

- 上述RESTful接口说明是推荐设置, 实际上, 我们鼓励同学们自己设计符合规范的RESTful接口, 只要把相关接口信息详细解释清楚。
- 这里需要注意, 本次作业的数据集是csv文件, 需要同学们通过RESTful接口注入到数据库中, 因此从csv到调用RESTful接口注入数据的脚本文件也应该实现, 并且需要提供启动脚本。
- 因此设计自己的RESTful接口是允许的, 前提是对设计的接口提供明确的说明。

△ 为了在地址中直接添加参数, 可以使用Flask_restful。安装: `pip install flask_restful`。当然其他框架应该也提供类似的功能, 请大家自行学习。

5. 测试

首先说明一点, 对于我们提供的数据, 需要大家全部注入到数据库对应的表中, 这算是作业的一部分。

附: 关于触发器, 在注入dept_manager数据时请保证触发器正常工作, 维护一个dept_manager_title表。此时不需要考虑删除的情况。

SQL相关测试

这部分的测试分为两类:

1. 基本查询, MySQL语句查询。
2. 索引查询, 针对employees表添加的索引进行查询
3. 触发器查询, 针对触发器逻辑进行mysql查询。

RESTful服务测试

- 我们会对大家搭建的RESTful服务进行测试，为了启动同学们撰写的代码，需要大家提供详细的启动脚本。因此，请大家将后端服务启动脚本保存在`start.sh`文件中，启动脚本需要在项目根目录下。
- 具体测试是针对接口的调用，但是不排除会调用RESTful服务对全部数据进行注入。

6. Docker镜像开发

- 后端开发对于环境是比较敏感的，遇到开发环境改变甚至操作系统的改变都是有可能的，docker作为一个跨平台的容器技术，可以很好的解决这个问题。因此，我们给出一个选作项：同学们可以使用docker进行开发，将上述完整需求部署在docker镜像中
- 具体样例在第五次实验课有展示，同学们提供Dockerfile文件以及相应的启动说明，同样放在根目录下。

作业提交

- 开发的代码源文件及相应文档
- 脚本1: RESTful服务启动脚本`start.sh`，请确保本地可以正常启动。
- 脚本2: 数据注入脚本`insert.sh`，该脚本使得数据从CSV文件调用CURD服务注入到数据库中，**不需要使用RESTful服务**。
- 实验报告：叙述自己的开发思路以及使用的技术。报告格式不限，但是需要包含上述的作业内容过程说明。
- (选做)Dockerfile文件以及相应的启动说明

说明：

- 为什么需要大家提供`.sh`脚本文件，因为考虑到不同同学可能使用不同的框架，因此启动服务的指令是多样的，为了方便测试，统一将启动指令放在`start.sh`文件中。比如你使用python框架，文件为`main.py`，那么在`.sh`文件中放`python main.py args...`即可。

提交文件根目录参考：

```
- src
  - main.py
  - other_source_file
- insert.sh
- start.sh
- readme.md (相关的脚本执行说明，可以使用其他文档格式，推荐使用.md文件)
- 实验报告
```

作业提交方式：上传到云平台，命名为"`学号-姓名-数据管理技术期中大作业`"。请大家用tar.gz压缩格式。

分数构成

本次期中作业总分数为10分。具体的：

- 完成基本的CURD功能以及通过MySQL语句测试可以获得5分
- 通过索引测试和触发器可以获得2分
- 完成RESTful接口服务获得3分

- 选择Docker镜像开发并且成功通过测试**额外+2分**

Q & A

1. 数据集获取方式 请登录bhpan查看作业的数据集。
<https://bhpan.buaa.edu.cn/#/link/19801618C4742AAAC9FBA3E2FA28428A>
2. 关于Docker
 - Docker作为一个选做项，不强制要求，有兴趣有余力的同学可以尝试，我们会给一些额外的分数作为激励。(课程总分+2分)
 - 选择Docker同样需要实现两个脚本，只需要将RESTful服务部署在Docker即可，不需要将insert服务部署在docker。（这个是因为使用RESTful服务访问数据库才是作业的整体流程，**insert.sh**是为了方便部分作业的提交）
3. 作业截止时间 11周周三晚上12:00（2023/05/03:24:00）
4. 关于提交的两个脚本文件**start.sh**和**insert.sh**，考虑到大家很多同学使用的是windows系统，所以为了简化这个一过程，请使用windows的同学都使用python开发，**注意这是强制要求**。本次大作业的任务通过python启动是足够的，只需要在.sh文件中写python启动脚本即可！并且需要大家自己测试.sh可以正常启动。建议使用wsl或者git bash测试。
5. 关于测试流程：
 - 数据库的CURD以及索引触发器等我们会直接对线上数据库(mysql.buaa.seekthought.com)进行查询
 - 关于RESTful服务，我们会启动大家的服务进行测试。
 - 我们会对代码进行抽查，执行insert.sh脚本检验代码的真实性。
6. 关于作业 本次作业可以理解为两个部分，第一部分数据库CURE以及索引和触发器的实现，这部分要求通过代码实现，而不是通过GUI实现。**insert.sh**脚本就是可以使得我们提供的数据集通过第一部分的代码注入到数据库中（localhost或者我们提供的线上数据库）第二部分是提供RESTful服务，使得数据可以通过http请求注入到数据库并且执行一些查询。
7. 关于数据集位置 请大家把数据集放在项目根目录，并且从根目录读取数据。上传的文件中**请不要包含数据集**，只提交代码文件、脚本文件、说明文档以及实验报告。
8. 关于本地数据库和我们提供的线上数据库 建议大家在本本地数据库实现全部开发并且确认后再连接线上数据库进行数据注入，并且确保最后提交的代码是线上数据库的版本。