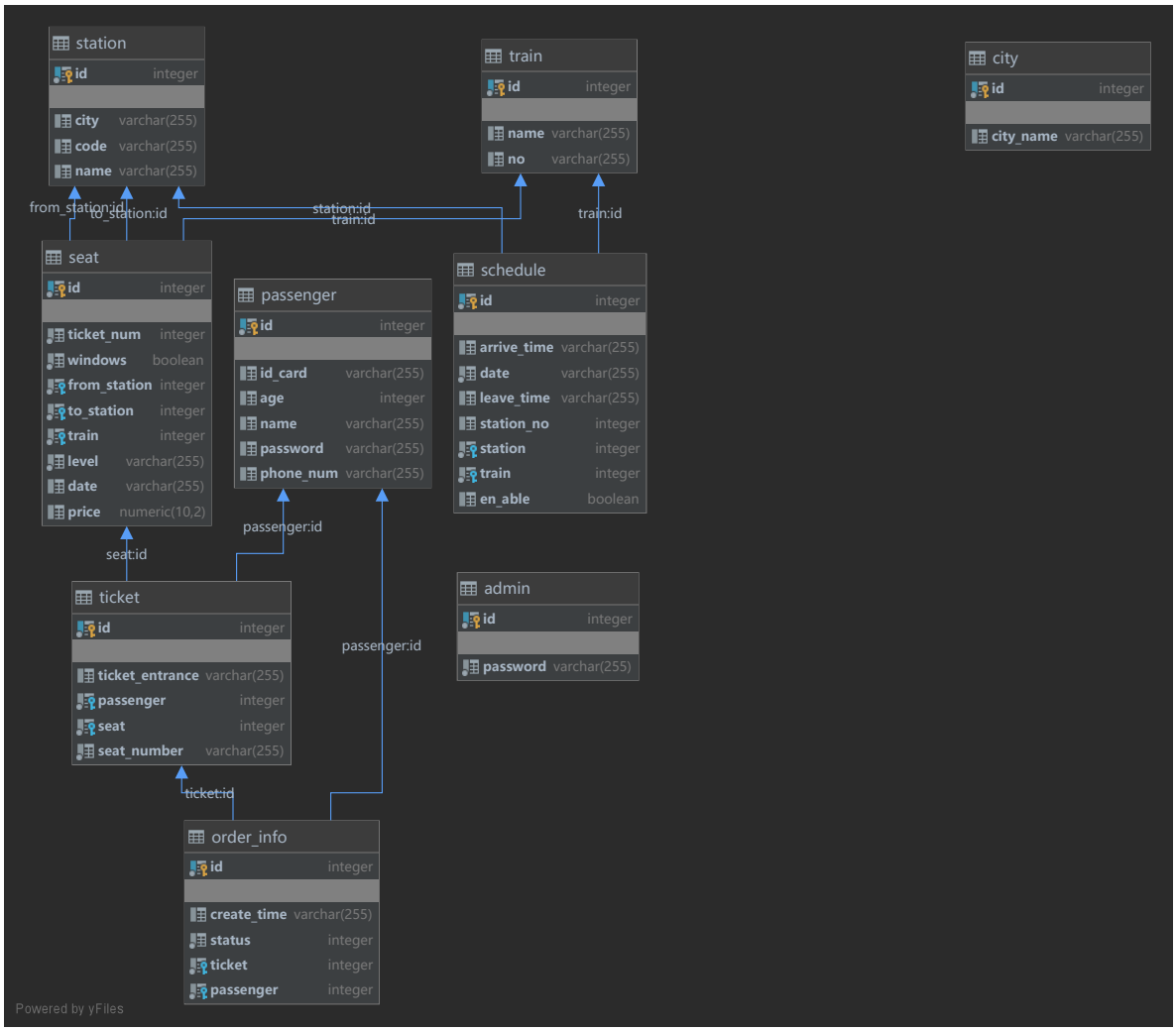
DB project 32106购票系统 答辩

数据库设计

数据库整体设计示意图



注：city表是之前测试功能留下的，实际项目中没有用到这个表

数据库相关设计

Trigger：身份证/手机号合法性

Index：较大的表（schedule、seat）加入了一些index加快查询

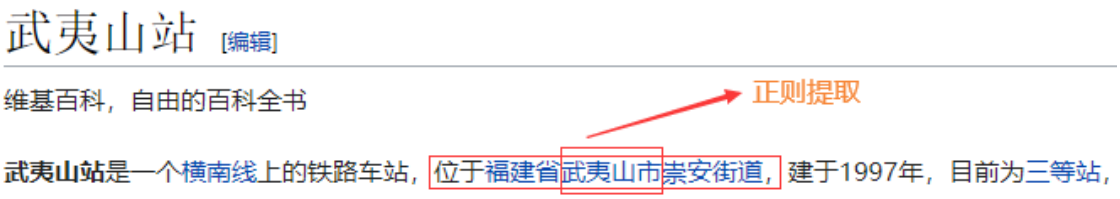
主键外键： 有关系的表都有外键相连

完成project后的体会：一些约束应该放在后端用代码实现而非使用数据库的约束，这样可以有效加快数据库的访问效率，但这毕竟是一个数据库Pro.....

数据来源

- **车程表**：12306旧版API，全国五月份（有一些车次一个月只开几次，五月初爬取，可能有部分车次有停运等变动）
- **座位表**：根据车的类型自动生成座位等级/默认余票五张

- 车站数据来自12306的车站电报码表，其对应的城市数据通过爬取维基百科用正则表达式提取,可能存在少量错误，eg.



Project整体实现及相关bonus

数据库：postgresql　　数据库位于阿里云学生机

后端与数据库的交互：Spring-Jpa（ORM）　　有效防止sql注入

数据库连接池：Hikaricp （配置信息详见提交的代码中**dbproj**项目下的配置文件）

后端：Springboot框架 很多实用的注解帮助我们极大减少了代码量

前端：JavaFx 前后端使用Web通讯，restful风格API

事务：对数据库进行增删改的连续操作都会放进一个事务里。eg. 购票中把余票锁定(余票数量-1)和订单确认放入一个事务中，可以防止车票超售

并发性：Springboot本身有一些对并发的处理，可以应付一般的高并发场景。

并发性测试

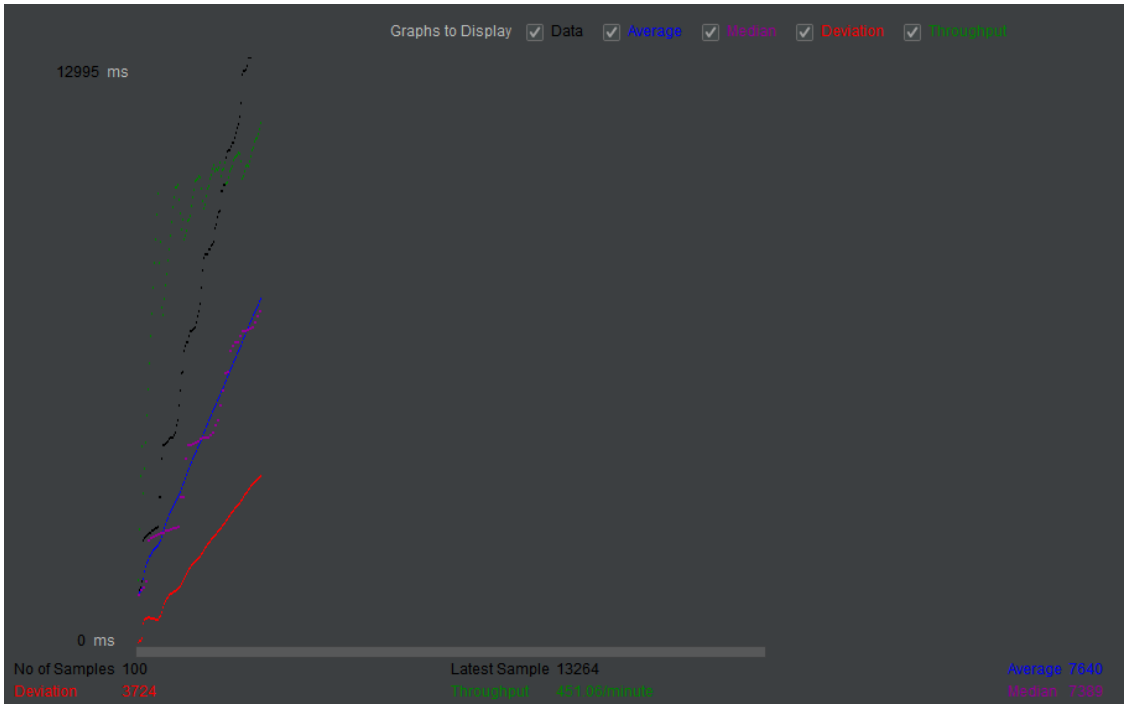
测试工具： JMeter

测试参数：并发量分别在100/1000/10000时的后端响应情况

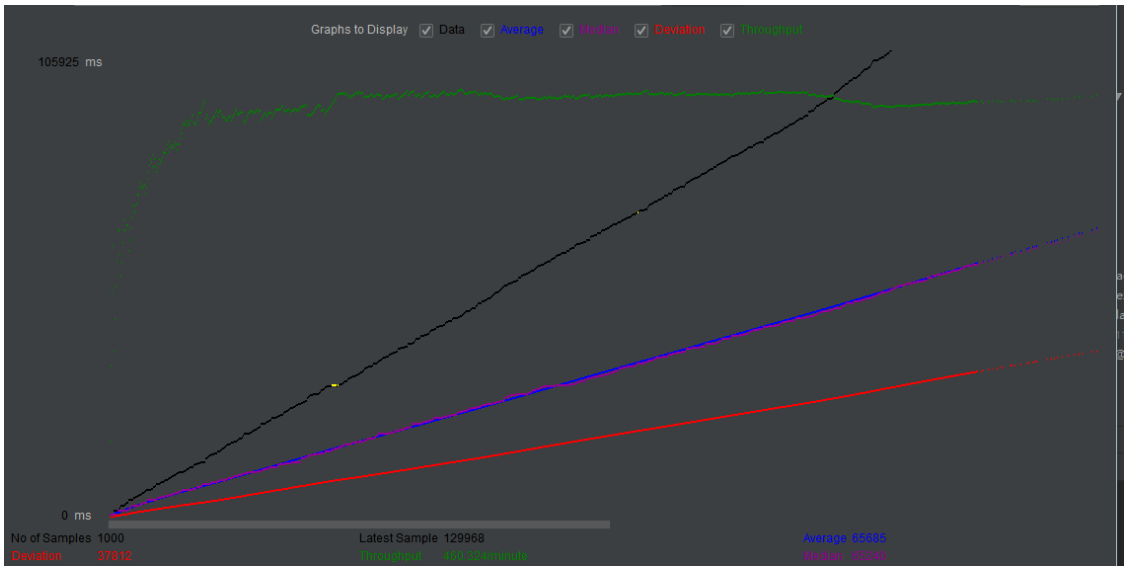
受测测API接口：较常用的城市间查询接口**/32106/Schedule/query/bycity**

测试结果：

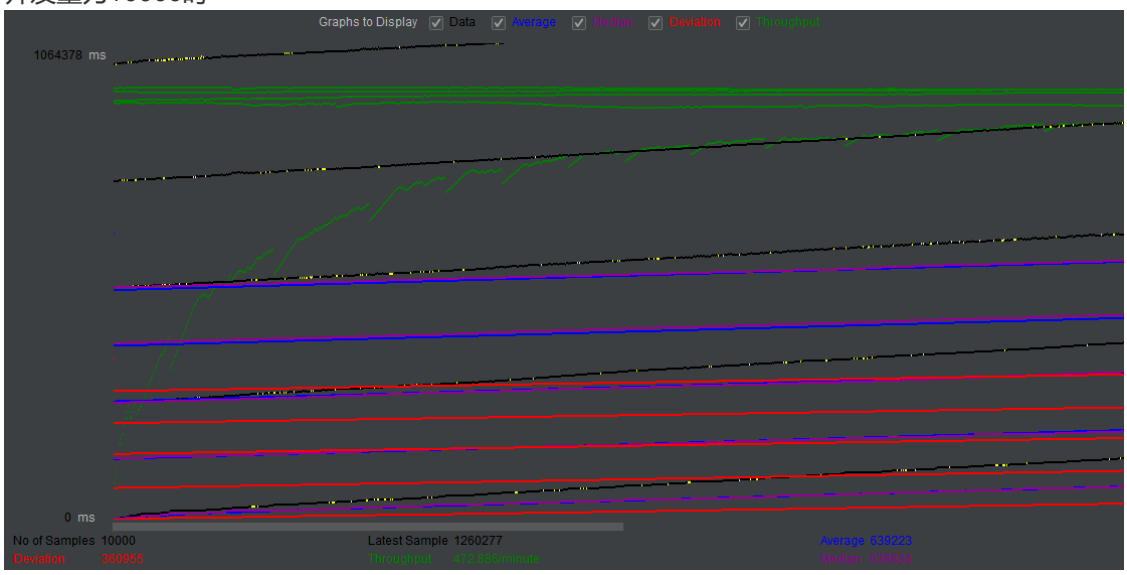
- 并发量为100时



- 并发量为1000时



- 并发量为10000时



结论：

- 较高的并发量不会使后端程序崩溃，可以看到在三种不同程度的压力下后端的数据吞吐量（**thoughput**）基本是保持不变的
- 看到并发量在100时不会发生数据丢失，响应时间在可忍耐范围，1000/10000并发量下会出现不同程度的数据丢失，但是绝大部分请求能够正确响应，但响应时间较长
- 通过查看后端抛出的异常信息

```
2020-05-22 10:56:43.072  WARN 7852 --- [o-8081-exec-729] o.h.engine.jdbc.spi.SqlExceptionHelper   : SQL Error: 0, SQLState: null
2020-05-22 10:56:43.072 ERROR 7852 --- [o-8081-exec-729] o.h.engine.jdbc.spi.SqlExceptionHelper   : DatebookHikariCP - Connection is not available, request timed out after 30001ms.
2020-05-22 10:56:43.073 ERROR 7852 --- [o-8081-exec-729] o.a.c.c.C.[.[.[.[dispatcherServlet]      : Servlet.service() for servlet [dispatcherServlet] in context with path [/32106] thre

java.sql.SQLTransientConnectionException: DatebookHikariCP - Connection is not available, request timed out after 30001ms.
    at com.zaxxer.hikari.pool.HikariPool.createTimeoutException(HikariPool.java:676) ~[HikariCP-3.2.0.jar:na]
    at com.zaxxer.hikari.pool.HikariPool.getConnection(HikariPool.java:198) ~[HikariCP-3.2.0.jar:na]
    at com.zaxxer.hikari.pool.HikariPool.getConnection(HikariPool.java:155) ~[HikariCP-3.2.0.jar:na]
    at com.zaxxer.hikari.HikariDataSource.getConnection(HikariDataSource.java:128) ~[HikariCP-3.2.0.jar:na] <26 internal calls>
    at org.springframework.orm.jpa.SharedEntityManagerCreator$DeferredQueryInvocationHandler.invoke(SharedEntityManagerCreator.java:402) ~[spring-orm-5.1.5.RELEASE.jar:5.1.5.RELEASE] <
    at org.springframework.data.jpa.repository.query.JpaQueryExecution$CollectionExecution.doExecute(JpaQueryExecution.java:129) ~[spring-data-jpa-2.1.5.RELEASE.jar:2.1.5.RELEASE]
```

  推测是因为数据库在阿里云上，大量数据传输带宽不够导致的高延迟使连接超时关闭
- 这只是一个**粗略**的并发压力测试，造成的压力和实际情况中多服务器同时访问还是有一定差距。不过相较于我们对它的期望来说还是令人满意的。

具体的功能演示

管理员：增删站点、车次 掌握普通用户信息

普通用户：查询（按城市、车站、转乘推荐），购票， 取消订单，对自己的个人信息更改等

播放来自于我队友王学豪同学的前端功能演示视频