# 2021 大数据管理课程实验

## 实验的需要的软件以及数据。

#### **实验需要用到的软件有**：

1. 数据库：图数据库Neo4j，关系数据库MySQL，文档数据库MongoDB。
2. 编程语言：java
3. 数据集：Yelp Dataset(<https://www.yelp.com/dataset/documentation/main>)

#### 软件的安装：

1. Neo4j安装，使用zip版本，解压后进行配置即可使用。
2. 使用安装包，安装在想要安装的位置。

#### 任务如下：

* 1. MySQL体验MySQL在InnoDB存储引擎下的MVCC多版本并发控制，实现的事务ACID特性。
     1. 创建一个数据库test，在此数据库中创建一个表student，属性只有一个name类型是varchar,然后启动2个(win+r，再输入cmd)窗口，登陆MySQL。两个窗口可以都使用root账号登陆（mysql -uroot -pxxxx）xxx是密码。

准备如下：

create database test;

use test;

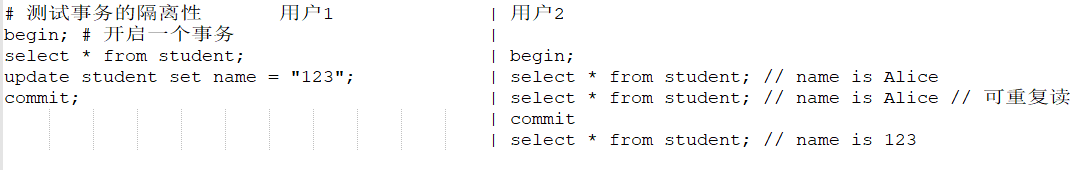
create table `student` (

`name` varchar(64) not null

) engine=INNODB;

insert into `student` values("Alice");

开始实验：



* 1. Neo4j的实验
     1. 使用import.bat命令把数据导入到neo4j中
     2. 查询标签是UserNode的节点，限制10个
     3. 查询名字是 Dwi 的节点。
     4. 查询userid是q-0Um-wv45DcHdNIyhTUOw 发表的评价。
     5. 查询被userid是q-0Um-wv45DcHdNIyhTUOw 评价的商家的名称。
     6. 查询评论过businessid为\_9kpamfhbsG5NVNpP-ED3w的user。
     7. 查询用户，按照粉丝数降序排序限制20条即可。
     8. 使用where查询城市在North Las Vegas的商家名字。
     9. 查询userid是q-0Um-wv45DcHdNIyhTUOw 的朋友数量。
     10. 查询userid是q-0Um-wv45DcHdNIyhTUOw 的朋友的名字，以一個list的形式返回。
     11. 使用with传递查询结果到后续的处理。查询出Dwi的朋友（直接相邻）分别都有多少位朋友（直接相邻）（Dwi的userid为q-0Um-wv45DcHdNIyhTUOw）。
     12. 统计商家在每个城市的分布情况，返回城市中商家的数量和城市名称。
     13. 统计包含类别是Asian Fusion的商家的城市，按照每个城市中商家的数量降序排序，返回城市和相应城市中的商家的数量。
     14. 统计用户名重名的次数，返回姓名重复的次数和对应的姓名，按照次数降序排列。
     15. 统计每个用户名热度（名字的重复的次数在所有的不同的用户名中的占比），返回热度和用户名。
     16. 统计每个用户做出的评价的数量，按照用户评价的数量降序排列，返回用户名和评价的数量。
     17. 实验建立索引对查询带来的提升，但会导致插入，删除等操作变慢（需要额外维护索引代价）。
     18. 查询与用户user1（userid: DXDeGUM\_L7tt3u\_5IkFRLw) 不是朋友关系的用户中和user1评价过相同的商家的用户，返回用户名，共同评价的商家的数量，按照评价数量降序排序[查看该查询计划，并尝试根据查询计划优化]。
  2. MongoDB实验任务
     1. 查询user集合从第二条记录开始的三条记录。
     2. 查询user\_id是q-0Um-wv45DcHdNIyhTUOw的user。
     3. 查询fans不小于50且review\_count 大于 10的user限制10条
     4. 查询所有名字为rashmi的用户名，不区分大小写。
     5. 查询同时是user\_id为xjrUcid6Ymq0DoTJELkYyw和6yCWjFPtp\_AD4x93WAwmnw的朋友的用户，限制20条。
     6. 统计user一共有多少条数据。
     7. 计用户名重名的次数，返回姓名重复的次数和对应的姓名，按照次数降序排列。
     8. 查询评价对应的user的信息【提示2个表关联查询$lookup】
     9. 使用explain看db.user.find({user\_id: "z4xZolh3zzBmPEW6RuGuJQ"});执行计划，进行查询优化
     10. 创建一个review的子集合reviewSub(取review的前一百万条数据)，对评论的内容建立全文索引，进行查询评价的内容中包含关键词ambience good lively的评价
     11. 体验文档的结构是自由的【同一个集合中可以存在异构的文档】
     12. 查询距离店铺6016c6b4af81085b0f2183d5(object id) 10 米以内的商家
     13. 实时统计一个网站的访问次数，可以这样设计db.page\_counter.insertOne({cnt: 0})，如何更新这个字段，如果使用MySQL怎么实现，性能怎么样，如何优化，和mongoDB对比的情况如何
     14. 使用map reduce计算每位用户做出评价的平均分（建议在reviewSub集合上做，review过于大）。

#### 4．应用系统（可选）

尝试实现一个功能用来在MySQL和neo4j中添加一个新的用户，需要满足事务的一致性，保证在添加的用户在上述2个数据库中状态一致（同时添加成功，或者都不能添加，不能出现只在某一个数据库添加成功，提示：可以自定义springBoot的ChainedTransactionManager）