

# 非关系型数据库课程项目

金河市公交线路系统

# 项目初衷

- 不仅仅是简单的CRUD，调用框架。
- 适量的工程设计与开发能力。
- 与时俱进、结合具体生活场景。
- 为大四下软件开发实践做铺垫。
- 积累项目经验，简历or毕业设计。

# 项目要求

- 必须使用至少一种非关系型数据库。（Redis,Mongodb,Neo4j,HBase）
- 仔细阅读需求文档，结合数据特点，选择合适的非关系型数据库。
- 需要有前端界面进行人机交互与可视化。
- 注意设计的合理性，是否符合真实场景的需求。
- 项目简要报告：数据库选型及设计、需求实现方法（SQL或代码）。

# 需求

- 提供公交线路的基本信息查询：当搜索某条公交线路时，返回线路的基本信息，包括线路是否有向、里程、单程运行时间、线路走向、线路运营时间、线路类型、班次间隔等。
- 提供公交线路的站点查询：
  - 当搜索某条公交线路时，返回线路的全部站点。因为部分线路存在单行站，所以需要考虑线路的上下行。对于特殊站点，例如起点站、终点站、地铁站、单行站，需要在前端界面有额外的标识。
  - 当搜索某个站点时，返回停靠的公交线路，也需要考虑到单行站和方向的问题。
  - 搜索任意两个站点，查询是否存在直达线路。
  - 搜索某条线路的两个站点，返回线路的运行方向、沿路全部站点和行驶时长。
  - 搜索任意两个站点，返回最短路径（可假设每两个相邻站点间等距），并提供线路换乘。
- 提供公交线路的班次查询：
  - 当搜索某条公交线路时，返回线路的所有班次信息。
  - 给定某时刻和某站点，返回该时刻10分钟内的班次到点信息。
  - 给定某时刻和某线路，返回该线路能乘坐的班次信息。
- 提供公交线路的分析查询：
  - 统计站点信息：例如统计地铁站数量、终点站数量、停靠线路较多的站点、停靠线路较少的站点等。
  - 统计线路信息：例如发车最早的路线、运行时间最长的路线、站点最多的路线、线路类型统计等。
- 提供公交线路和站台的增删改：需要支持线路新增和站台新增，需要考虑到增删站点是否会影响其他线路。

# 需求给分点

## 基本信息查询

- 查询30路公交的基本信息。（2分）

## 站点查询

- 查询2路公交的全部站点信息，以链状形式返回，即站点间具有方向性。由于2路不是环线，所以需要区分上下行。（2分）
- 查询全部公交的全部站点信息，按照字典序排序。（2分）
- 查询锦城广场站停靠的所有线路。站点由id唯一标识，但存在存在同名不同id的站点，可能是同名不同方向的站点，例如17路上下行均经过锦城广场站，此时需要提示方向；也可能是为了分流而设置的同名不同位置的站点；也可能是快速公交、高峰公交特有的站点。若查询到多个同名站点，需要按照id进行分组。（2分）
- 查询乘坐10路从大悦城到小吃街，线路的运行方向（上行或下行）、沿路站点和运行时长。（运行时长由班次信息计算得出）（2分）
- 查询从id为16115的站台(红瓦寺)到id为14768的站台(动物园)的最短路径，该需求对应的场景是用户未知乘坐的线路，需要查询最优的换乘方式。（4分）
- 查询从荷花池到环球中心(始发站)是否存在直达线路。（2分）

# 需求给分点

## 班次查询

- 查询N8路（环线）的全部班次信息。（2分）
- 查询上午08:37分新华书店站10分钟内即将停靠的线路，并显示几分钟后某线路即将到站，若查询到多个同名站点，需要按照id进行分组。（2分）
- 查询上午10:32分地铁万盛站82路的最近的3趟班次信息。可能当前时间下暂无班次。（2分）

## 分析查询

- 统计停靠路线最多的站点（按照id统计）并排序，显示前15个和后15个。（2分）
- 统计地铁站数量（以地铁开头）、起点站（末尾标识始发站）数量、终点站（末尾标识终点站）数量、单行站（比较上下行确定单行站）数量。（2分）
- 分组统计常规公交（包括干线、支线、城乡线、驳接线、社区线）、快速公交（K字开头）、高峰公交（G字开头）、夜班公交（N字开头）的数量。（2分）
- 查询15路和30路重复的站点名，并统计站点数。（2分）
- 查询261路上行一共有多少条可以换乘的线路，注意去重。换乘线路数即261路上行停靠的所有站台停靠其他线路的数量的总和。（2分）
- 根据线路可换乘数量降序排序，显示前15条。（2分）

# 需求给分点

- 查询连接两个站台之间线路最多的两个站台并且按照降序排列，显示前15个。（2分）
- 根据站点数量对线路进行排序，显示前15条和后15条。（2分）
- 根据运行时间对线路进行排序，运行时间由班次数据计算而得，显示前15条和后15条。（2分）
- 计算两条线路之间的重复系数。对于一条线路，有若干个站台A、B、C…。假如A与B之间有4条线路，则A与B的非重复系数为 $\frac{1}{4} = 0.25$ 。一条线路的非重复系数则是线路沿线所有站台间的非重复系数的平均。例如，计算N12路上行的非重复系数为0.47261904761904755。再提供两组测试用例：G95非重复系数为1（仅两站且一条线路相连）；208路上行非重复系数为0.8833333333333332。（4分）

## 增删改

- 添加一条站点数不少于10的线路，要求线路合理。（2分）
- 删除某条线路，若沿途站点为该线路独有，也需要删除该站点。（2分）
- 为某条线路增加、删除、替换站点（班次信息和路线信息也需要同步更新）。（2分）

# 数据

- stations.json 站点数据，由id区分，存在一定量的同名站点。
- lines.json 线路基本信息。
- routes.json 线路站点信息，注意上行、下行方向，有的线路无方向。
- timetables.json 线路班次信息，和线路站点一一对应。

```
1 { "name": "G63路上行", "timetable": [ [ "07:30", "07:34", "07:38", "07:42", "07:46", "07:50", "07:54" ],  
  [ "07:45", "07:49", "07:53", "07:57", "08:01", "08:05", "08:09" ],  
  [ "08:00", "08:04", "08:08", "08:12", "08:16", "08:20", "08:24" ] ] } // timetable为二维数组，G63路上行有3趟班次，7个站点，数  
  组第一个维度为某个班次，可见是15分钟一班；数组第二个维度长度与站点数一一对应，表示到达某站的时间，每个站大约运行4分钟。
```

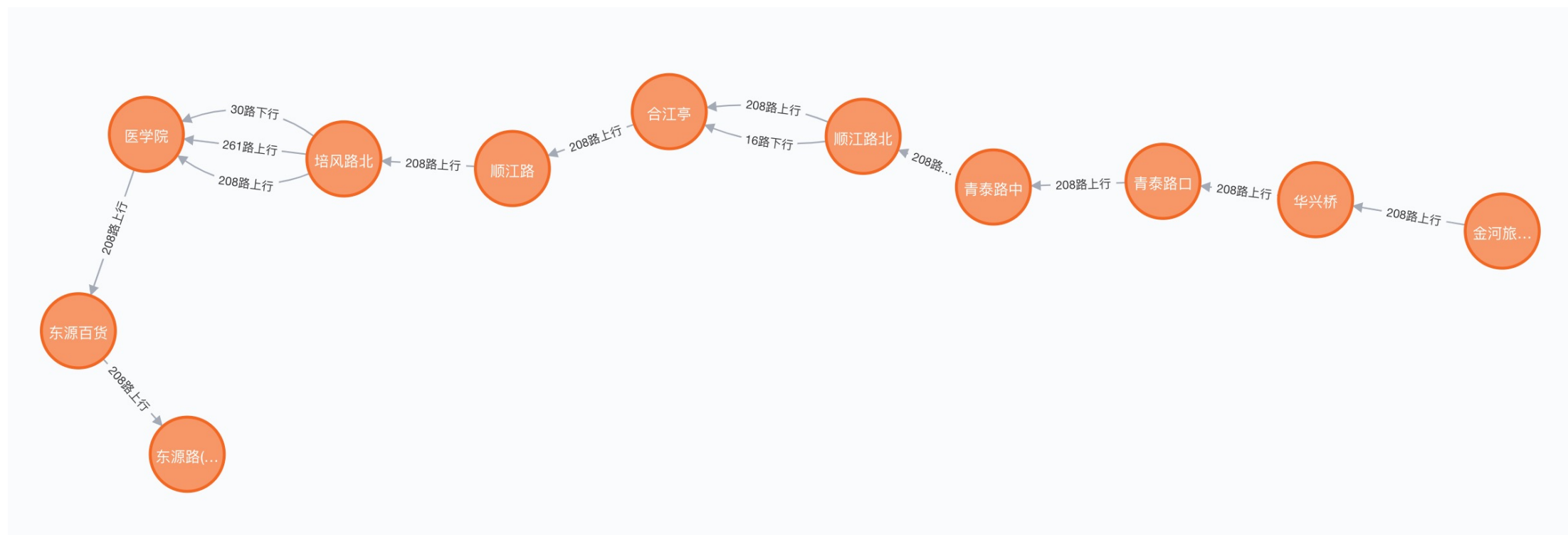


# 难点

- 同名站台在逻辑上如何处理？（结合实际生活场景考虑）
- 线路方向如何区分？（可能存在单行站）
- 沿线站点查询、最短路径查询如何实现？
- 班次查询、统计查询的算法与可视化。。。

# 项目效果举例

查询某条线路站点：



# 评分细则（更新）

- 数据入库：10分
- 数据库设计：20分（数据库设计15分，数据库选型5分）
- 功能需求：60分（原50分需求拆分为25个功能点，4分算两个功能点，自由选择一部分功能点实现）

得分上限	36分	42分	48分	54分	60分
2-3人组	10个功能点	11个功能点	12个功能点	14个功能点	16个功能点
4人组	14个功能点	15个功能点	16个功能点	18个功能点	20个功能点

- 项目演示与报告：10分