

# 非关系型数据库项目——金河市公交线路系统

## 项目介绍

金河市（虚拟城市）是一座风景优美的城市，包括信息技术产业专区高新区、历史悠久的老城区、人口流动较多的天河区和有高速公路联通的合江县。有两座火车站（火车西站和金河南站）和一座快铁站（普光快铁），有两座客运站（金河客运站和北客站）。其公交网络也比较发达，一共有93条公交线路，包括常规线路、快速公交、高峰公交、夜班公交等多种类型。现需要你利用所学到的非关系型数据库理论，为金河市设计一个公交线路系统。

## 需求介绍

运用非关系型数据库理论，将提供的数据入库，开发前后端界面，为系统提供基础的增删改查需求和可视化功能。使用的前后端技术不限，数据库选型需要至少包含一种非关系型数据库。根据数据特点，可能需要多种数据库相结合。

- 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。

## 界面参考

- 5个Tab，分别是基本信息查询、站点查询、班次查询、分析查询和线路管理，具体内容见下一节。
- 基本信息查询，提供1个按照线路名查询的搜索框，以表格、卡片或其他组件返回相关信息。
- 站点查询，一共有5个需求，可以将界面分成5个区域。
  - 全部站点查询，包括单条线路的全部站点信息和所有线路的全部站点信息。
  - 站点停靠线路查询，查询某站点停靠的所有线路信息。
  - 沿途站点查询，查询某条具体线路两个站台之间的信息。
  - 最短路径查询，查询两个站点的最短路径。
  - 直达站点查询，查询两个站点是否可以直达。
- 班次查询

- 某条线路全部班次查询，可以以表格或timeline等组件呈现。
- 某一时刻范围具体站点班次信息，使用时间选择器、timeline、下拉框、表格等组件呈现。
- 某一时刻范围具体站点具体线路最近班次信息，使用时间选择器、timeline、下拉框、表格等组件呈现。
- 分析查询，具体内容参考下一节，可以使用echart等加入一些统计图表，进行排序、统计、计算等一系列可视化功能。
- 线路关系，实现3个增删改的需求，形式不限。

## 具体应用场景

为了让学生更好体会项目需求，现给出几个具体应用场景，学生需要考虑采用怎样的查询方式和数据库设计实现以下的功能。在实验报告和项目中体现出，可以以SQL语句或伪代码的形式体现。

### 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）

### 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）

- 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）

## 数据介绍

stations.json 站点信息，以id唯一标识。

```
1 { "english": "YongTongLu", "id": "41394", "name": "永通路" }
```

routes.json 路线站点信息，alongStation为数组，记录了沿线所有站点的id。

```
1 {
2   "alongStation": [
3     "16560",
4     "803",
5     "98730",
6     "14214",
7     "761",
8     "750",
9     "744",
10    "730",
11    "100201",
12    "1104",
13    "1148",
14    "3654",
15    "15343",
16    "22007",
17    "22011",
18    "23048",
19    "23058",
20    "23084",
21    "23114",
22    "27327",
23    "23133",
24    "27760",
25    "27810",
```

```

26         "27789",
27         "27711",
28         "27732",
29         "27698",
30         "23351",
31         "27676"
32     ],
33     "name": "1路上行"
34 }

```

lines.json 路线基本信息

```

1  {
2      "directional": true, // 无向为false, 不仅仅是环线, 还有G62路等单向线路
3      "interval": 5, // 班次间隔 (分钟)
4      "kilometer": 15.0, // 单向里程 (公里)
5      "name": "1", // 线路名
6      "onewayTime": "约52分", // 运行时长
7      "route": "金河客运站-花明公交站", // 线路走向
8      "runtime": "6:00-23:59", // 运营时间, 和班次表有略微不一致
9      "type": "干线" // 线路类型
10 }

```

timetables.json 班次信息

```

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。

```

## 评分细则

- 数据入库：（10分）
  - 数据入库即得10分
- 数据库设计：（20分）
  - 给出数据库设计图（15分）
  - 选用该数据库的理由（5分）
- 前端界面：（15分）
  - 基础分10分，依据美观程度加分。
- 功能点：（50分）
  - 基本信息查询：2分
  - 站点查询：14分
  - 班次查询：6分
  - 分析查询：22分

- 增删改：6分
- 项目演示：（5分）
  - 基础分5分，每有一个不符合实际应用场景的点扣1分。
- 加分点：（额外5-10分）
  - 对于最短路径算法有更仔细的考虑，比如换乘次数最少、站点最少、带约束条件的最短路径、时间最短等。
  - 对性能有额外考虑。
  - 实现了额外需求。
  - 其他创新点。