



数据库系统原理课程

第三次作业

学院：	计算机与信息技术学院
专业：	计算机科学与技术
班级：	计科 1804
姓名：	王子龙
学号：	18281218
教师：	徐薇

2020 年 10 月 10 日

1. 按如下格式要求在实验报告中描述所有涉及到的表的结构。

表名	Remaining_Tickets				
数据库用户	root				
主键	TrainNumber、DepartureStation、ArrivalStation				
外键	TrainNumber				
排序字段	TrainNumber				
索引字段	TrainNumber				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
TrainNumber	Varchar(7)	N	N		部分主键、外键
DepartureStation	Varchar(15)	N	N		部分主键
ArrivalStation	Varchar(15)	N	N		部分主键
SpecialClassRemaining	int	N	N		
FirstClassRemaining	int	N	N		
SecondClassRemaining	int	N	N		
SpecialClassFare	double	N	N		
FirstClassFare	double	N	N		
SecondClassFare	double	N	N		

*Remaining: 为*票余量

*Fare: 为*票价格

表名	Train_Info				
数据库用户	root				
主键	TrainNumber				
外键	N/A				
排序字段	TrainNumber				
索引字段	TrainNumber				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
TrainNumber	Varchar(7)	N	Y		主键
DepartureStation	Varchar(15)	N	N		
ArrivalStation	Varchar(15)	N	N		
Duration	int	N	N		
isGaotie	boolean	N	N		

DepartureStation: 出发站

ArrivalStation: 抵达站

表名	User_Info				
数据库用户	root				
主键	ID				
外键	N/A				
排序字段	username				
索引字段	username				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
username	Varchar(100)	N	Y		
password	Varchar(100)	N	N		
name	Varchar(100)	N	N		

phoneNumber	Varchar(20)	N	N		
emailAddress	Varchar(100)	N	N		
ID	Varchar(25)	N	Y		主键
Type	Int	N	N		
IDType	Int	N	N		

Type: 普通、学生、军人、老人, etc.

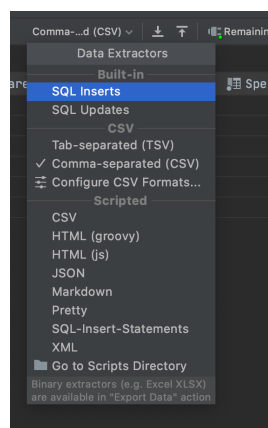
IDType: 身份证、护照、永久居留证, etc.

表名	Way_Station				
数据库用户	root				
主键	TrainNumber				
外键	TrainNumber				
排序字段	TrainNumber				
索引字段	TrainNumber				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
TrainNumber	Varchar(7)	N	Y		主键
Station1	Varchar(15)	N	N		
ArrivalTime1	datetime	N	N		
DepartureTime1	datetime	N	N		
Station2	Varchar(15)	N	N		
ArrivalTime2	datetime	N	N		
DepartureTime2	datetime	N	N		
Station3	Varchar(15)	Y	N		
ArrivalTime3	datetime	Y	N		
DepartureTime3	datetime	Y	N		
...
Station38	Varchar(15)	Y	N		
ArrivalTime38	datetime	Y	N		
DepartureTime38	datetime	Y	N		

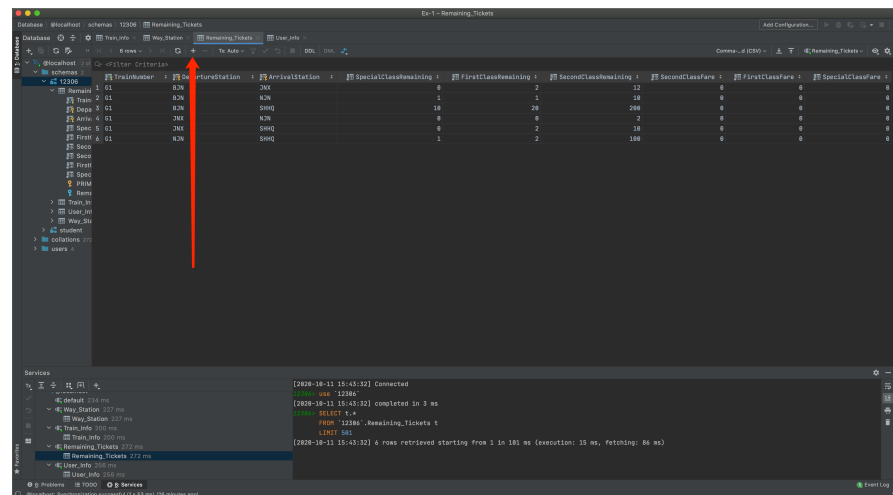
- 根据以上定义，写出各表的建表语句，并在你选的关系型数据库平台上建立各个表，请将建 表语句统一写在扩展名为 `sql` 的文件中，构建一个建库脚本文本，命名要求为：

DBLab3_18281218.sql

- 在 DataGrip 中可以直接以 `csv` 文件的形式导入数据插入数据：



此外可以在 DataGrip 中使用 “+” 加入数据:



INSERT

INTO Trian_Info values(G2,SHHQ,BJN,268,1);以手动插入数据。

4. 单表查询

```
mysql> SELECT * FROM `12306`.Remaining_Tickets;
```

TrainNumber	DepartureStation	ArrivalStation	SpecialClassRemaining	FirstClassRemaining	SecondClassRemaining	SecondClassFare	FirstClassFare	SpecialClassFare
G1	BJN	JNX	0	2	12	0	0	0
G1	BJN	NJN	1	1	10	0	0	0
G1	BJN	SHHQ	10	20	200	0	0	0
G1	JNX	NJN	0	0	2	0	0	0
G1	JNX	SHHQ	0	2	10	0	0	0
G1	NJN	SHHQ	1	2	100	0	0	0

6 rows in set (0.00 sec)

多表连接查询:

```
mysql> SELECT SecondClassFare FROM `12306`.Train_Info, `12306`.Remaining_Tickets where Train_Info.DepartureStation='BJN' and SecondClassRemaining > 10;
```

SecondClassFare
0
0
0
0
0
0

6 rows in set (0.00 sec)

唯一索引: *create unique index st on `12306`.Train_Info(TrainNumber);*

聚簇索引: *create cluster index aa on `12306`.User_Info (ID);*

7. 若某个表中涉及百万甚至千万级以上的数据,请提出仿真这些数据
的方案,并在实验报告加以叙述。

使用优化 SQL 语句:将 SQL 语句进行拼接,使用 insert into table
() values (), (), (), () 然后再一次性插入, 如果字符串太长, 则需

要配置下 MYSQL，在 mysql 命令行中运行：set global
max_allowed_packet = 2*1024*1024*10; 消耗时间为：11:24:06
11:25:06；插入 200W 条测试数据仅仅用了 1 分钟！代码如下：

```
$sql="insert into twenty_million (value) values";  
for($i=0;$i<2000000;$i++){  
    $sql.="('50'),";  
};  
$sql = substr($sql,0,strlen($sql)-1);  
$connect_mysql->query($sql);
```