

课程作业

课程名称：数据库系统原理
作业次数：作业#3
学号：21281280
姓名：柯劲帆
班级：物联网2101班
指导老师：郝爽
修改日期：2024年3月30日

1. 题目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。

(格式说明略)

Passengers

表名	数据库用户	主键	外键	排序字段	索引字段
Passengers	kejingfan	ID		ID	ID; Phone_number

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	BIGINT	N	Y		主键, 符合正则'^\d{18}\$'
Name	VARCHAR (255)	N	N		
Phone_number	BIGINT	N	Y		唯一, 符合正则'^\d{11}\$'

Users

表名	数据库用户	主键	外键	排序字段	索引字段
Users	kejingfan	ID	Phone_number; CitizenID	ID	ID; Phone_number; CitizenID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	INT	N	Y		主键
Password	VARCHAR (255)	N	N		
Phone_number	BIGINT	N	Y		外键
CitizenID	BIGINT	N	Y		外键

Stations

表名	数据库用户	主键	外键	排序字段	索引字段
Stations	kejingfan	ID		ID	ID; Name

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	INT	N	Y		主键
Name	VARCHAR (255)	N	Y		唯一
Province	VARCHAR (255)	N	N		
City	VARCHAR (255)	N	N		

Trains

表名	数据库用户	主键	外键	排序字段	索引字段
Trains	kejingfan	ID	StationID_start; StationID_arrive	ID	ID; StationID_start; StationID_arrive

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	VARCHAR (255)	N	Y		主键
Property	SMALLINT	N	N		
StationID_start	INT	N	N		外键
StationID_arrive	INT	N	N		外键

说明：

Property字段，值的二进制的每一位表示：

位数	3	2	1	0
值为0/1的含义	无/有 静音车厢	不是/是 智能车厢	无/有 卧铺	不是/是 复兴号

如 $\text{Property} = (9)_{10} = (1001)_2$ ，说明有静音车厢、是复兴号。

Seats

表名	数据库用户	主键	外键	排序字段	索引字段
Seats	kejingfan	ID	TrainID	ID	ID; TrainID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	VARCHAR (255)	N	Y		主键

字段名称	数据类型	允许为空	唯一	默认值	约束条件
Type	VARCHAR (255)	N	N		
TrainID	VARCHAR (255)	N	N		外键

Tickets

表名	数据库用户	主键	外键	排序字段	索引字段
Tickets	kejingfan	ID	StationID_start; StationID_arrive; SeatID	ID	ID; StationID_start; StationID_arrive; SeatID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	INT	N	Y		主键
Price	DECIMAL (7, 2)	N	N		
Time_start	DATETIME	N	N		
Time_arrive	DATETIME	N	N		检查 (Time_start <= Time_arrive)
StationID_start	INT	N	N		外键
StationID_arrive	INT	N	N		外键
SeatID	VARCHAR (255)	N	N		外键

Transactions

表名	数据库用户	主键	外键	排序字段	索引字段
Transactions	kejingfan	ID	User; PassengerID; TrainID; StationID_start; StationID_arrive; SeatID	ID	ID; User; PassengerID; TrainID; StationID_start; StationID_arrive; SeatID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
ID	BIGINT	N	Y		主键

字段名称	数据类型	允许为空	唯一	默认值	约束条件
Time	DATETIME	N	N		
Paid	TINYINT	N	N		
User	INT	N	N		外键
PassengerID	BIGINT	N	N		外键
TrainID	VARCHAR (255)	N	N		外键
StationID_start	INT	N	N		外键
StationID_arrive	INT	N	N		外键
Time_start	DATETIME	N	N		
Time_arrive	DATETIME	N	N		检查 (Time_start < Time_arrive)
SeatID	VARCHAR (255)	N	N		外键

Sold

表名	数据库用户	主键	外键	排序字段	索引字段
Sold	kejingfan		TransactionID; TicketID		TransactionID; TicketID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
TransactionID	BIGINT	N	N		外键
TicketID	INT	N	N		外键

Passbys

表名	数据库用户	主键	外键	排序字段	索引字段
Passbys	kejingfan		StationID; TrainID		StationID; TrainID

字段名称	数据类型	允许为空	唯一	默认值	约束条件
StationID	INT	N	N		外键
TrainID	VARCHAR (255)	N	N		外键
arrive_time	DATETIME	N	N		

字段名称	数据类型	允许为空	唯一	默认值	约束条件
leave_time	DATETIME	N	N		检查 (arrive_time <= leave_time)

2. 题目2

根据以上定义，写出各表的建表语句，并在你选定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文件中，构建一个建库脚本文本，命名要求为：DBLabScript_学号.sql

```

1 CREATE TABLE Passengers (
2     ID BIGINT PRIMARY KEY,
3     `Name` VARCHAR ( 255 ) NOT NULL,
4     Phone_number BIGINT UNIQUE NOT NULL,
5     CHECK ( REGEXP_LIKE ( ID, '^\\d{18}$' ) ),
6     CHECK ( REGEXP_LIKE ( Phone_number, '^\\d{11}$' ) )
7 );
8
9 CREATE TABLE Users (
10     ID INT AUTO_INCREMENT PRIMARY KEY,
11     `Password` VARCHAR ( 255 ) NOT NULL,
12     Phone_number BIGINT UNIQUE NOT NULL,
13     CitizenID BIGINT UNIQUE NOT NULL,
14     FOREIGN KEY ( Phone_number ) REFERENCES Passengers ( Phone_number ) ON
DELETE CASCADE,
15     FOREIGN KEY ( CitizenID ) REFERENCES Passengers ( ID ) ON DELETE CASCADE
16 );
17
18 CREATE TABLE Stations (
19     ID INT AUTO_INCREMENT PRIMARY KEY,
20     `Name` VARCHAR ( 255 ) UNIQUE NOT NULL,
21     Province VARCHAR ( 255 ) NOT NULL,
22     City VARCHAR ( 255 ) NOT NULL
23 );
24
25 CREATE TABLE Trains (
26     ID VARCHAR ( 255 ) PRIMARY KEY,
27     Property SMALLINT NOT NULL,
28     StationID_start INT NOT NULL,
29     StationID_arrive INT NOT NULL,
30     FOREIGN KEY ( StationID_start ) REFERENCES Stations ( ID ) ON DELETE
CASCADE,
31     FOREIGN KEY ( StationID_arrive ) REFERENCES Stations ( ID ) ON DELETE
CASCADE
32 );
33
34 CREATE TABLE Seats (
35     ID VARCHAR ( 255 ) PRIMARY KEY,
36     Type VARCHAR ( 255 ) NOT NULL,
37     TrainID VARCHAR ( 255 ) NOT NULL,
38     FOREIGN KEY ( TrainID ) REFERENCES Trains ( ID ) ON DELETE CASCADE

```

```

39 );
40
41 CREATE TABLE Tickets (
42     ID INT AUTO_INCREMENT PRIMARY KEY,
43     Price DECIMAL ( 7, 2 ) NOT NULL,
44     Time_start DATETIME NOT NULL,
45     Time_arrive DATETIME NOT NULL,
46     StationID_start INT NOT NULL,
47     StationID_arrive INT NOT NULL,
48     SeatID VARCHAR ( 255 ) NOT NULL,
49     FOREIGN KEY ( StationID_start ) REFERENCES Stations ( ID ) ON DELETE
CASCADE,
50     FOREIGN KEY ( StationID_arrive ) REFERENCES Stations ( ID ) ON DELETE
CASCADE,
51     FOREIGN KEY ( SeatID ) REFERENCES Seats ( ID ) ON DELETE CASCADE,
52     CHECK (Time_start <= Time_arrive)
53 );
54
55 CREATE TABLE Transactions (
56     ID BIGINT AUTO_INCREMENT PRIMARY KEY,
57     `Time` DATETIME NOT NULL,
58     Paid TINYINT NOT NULL,
59     `User` INT NOT NULL,
60     PassengerID BIGINT NOT NULL,
61     TrainID VARCHAR ( 255 ) NOT NULL,
62     StationID_start INT NOT NULL,
63     StationID_arrive INT NOT NULL,
64     Time_start DATETIME NOT NULL,
65     Time_arrive DATETIME NOT NULL,
66     SeatID VARCHAR ( 255 ) NOT NULL,
67     FOREIGN KEY ( `User` ) REFERENCES Users ( ID ) ON DELETE CASCADE,
68     FOREIGN KEY ( PassengerID ) REFERENCES Passengers ( ID ) ON DELETE
CASCADE,
69     FOREIGN KEY ( TrainID ) REFERENCES Trains ( ID ) ON DELETE CASCADE,
70     FOREIGN KEY ( StationID_start ) REFERENCES Stations ( ID ) ON DELETE
CASCADE,
71     FOREIGN KEY ( StationID_arrive ) REFERENCES Stations ( ID ) ON DELETE
CASCADE,
72     FOREIGN KEY ( SeatID ) REFERENCES Seats ( ID ) ON DELETE CASCADE,
73     CHECK (Time_start < Time_arrive)
74 );
75
76 CREATE TABLE Sold (
77     TransactionID BIGINT NOT NULL,
78     TicketID INT NOT NULL,
79     FOREIGN KEY ( TransactionID ) REFERENCES Transactions ( ID ) ON DELETE
CASCADE,
80     FOREIGN KEY ( TicketID ) REFERENCES Tickets ( ID ) ON DELETE CASCADE
81 );
82
83 CREATE TABLE Passbys (
84     StationID INT NOT NULL,
85     TrainID VARCHAR ( 255 ) NOT NULL,
86     arrive_time DATETIME NOT NULL,
87     leave_time DATETIME NOT NULL,
88     FOREIGN KEY ( StationID ) REFERENCES Stations ( ID ) ON DELETE CASCADE,

```

```

89 FOREIGN KEY ( TrainID ) REFERENCES Trains ( ID ) ON DELETE CASCADE,
90 CHECK (arrive_time <= leave_time)
91 );

```

3. 题目3

掌握使用各自选用的关系型数据库的控制台插入数据的不同方法（执行数据批量插入脚本、窗口界面表格式手工录入、命令行交互式录入），实际填入测试数据，以验证你所设计的数据模型的合理性和完整性，注意验证三种完整性约束。

执行数据批量插入脚本

```

1  INSERT INTO Stations ( `Name`, Province, City )
2  VALUES
3      ( '北京南', '北京', '北京' ),
4      ( '沧州西', '河北', '沧州' ),
5      ( '济南西', '山东', '济南' ),
6      ( '徐州东', '江苏', '徐州' ),
7      ( '宿州东', '安徽', '宿州' ),
8      ( '蚌埠南', '安徽', '蚌埠' ),
9      ( '南京南', '江苏', '南京' ),
10     ( '无锡南', '江苏', '无锡' ),
11     ( '上海虹桥', '上海', '上海' );
12
13  INSERT INTO Trains ( ID, StationID_start, StationID_arrive, Property )
14  VALUES
15     ( 'G103', 1, 9, 1 );
16
17  INSERT INTO Passbys ( StationID, TrainID, arrive_time, leave_time )
18  VALUES
19     ( 1, 'G103', '2024-03-28 06:20:00', '2024-03-28 06:20:00' ),
20     ( 2, 'G103', '2024-03-28 07:13:00', '2024-03-28 07:15:00' ),
21     ( 3, 'G103', '2024-03-28 08:01:00', '2024-03-28 08:04:00' ),
22     ( 4, 'G103', '2024-03-28 09:07:00', '2024-03-28 09:09:00' ),
23     ( 5, 'G103', '2024-03-28 09:28:00', '2024-03-28 09:33:00' ),
24     ( 6, 'G103', '2024-03-28 09:56:00', '2024-03-28 09:58:00' ),
25     ( 7, 'G103', '2024-03-28 10:40:00', '2024-03-28 10:43:00' ),
26     ( 8, 'G103', '2024-03-28 11:27:00', '2024-03-28 11:29:00' ),
27     ( 9, 'G103', '2024-03-28 11:58:00', '2024-03-28 11:58:00' );

```

窗口界面表格式手工录入

使用Navicat软件进行窗口界面表格式手工录入。

Objects * H:\Courseware\Principles_of_Datab... Trains @DBLab (DELL_MySQL) - Table				
Begin Transaction Text Filter Sort Columns Import Export				
ID	Property	StationID_start	StationID_arrive	
G103	1	1	9	
* G137	0	1	9	...

命令行交互式录入

登录MySQL:

```
1 | $ mysql -u kejingfan -p -h 192.168.31.197
```

使用数据库:

```
1 | mysql> USE DBLab;
```

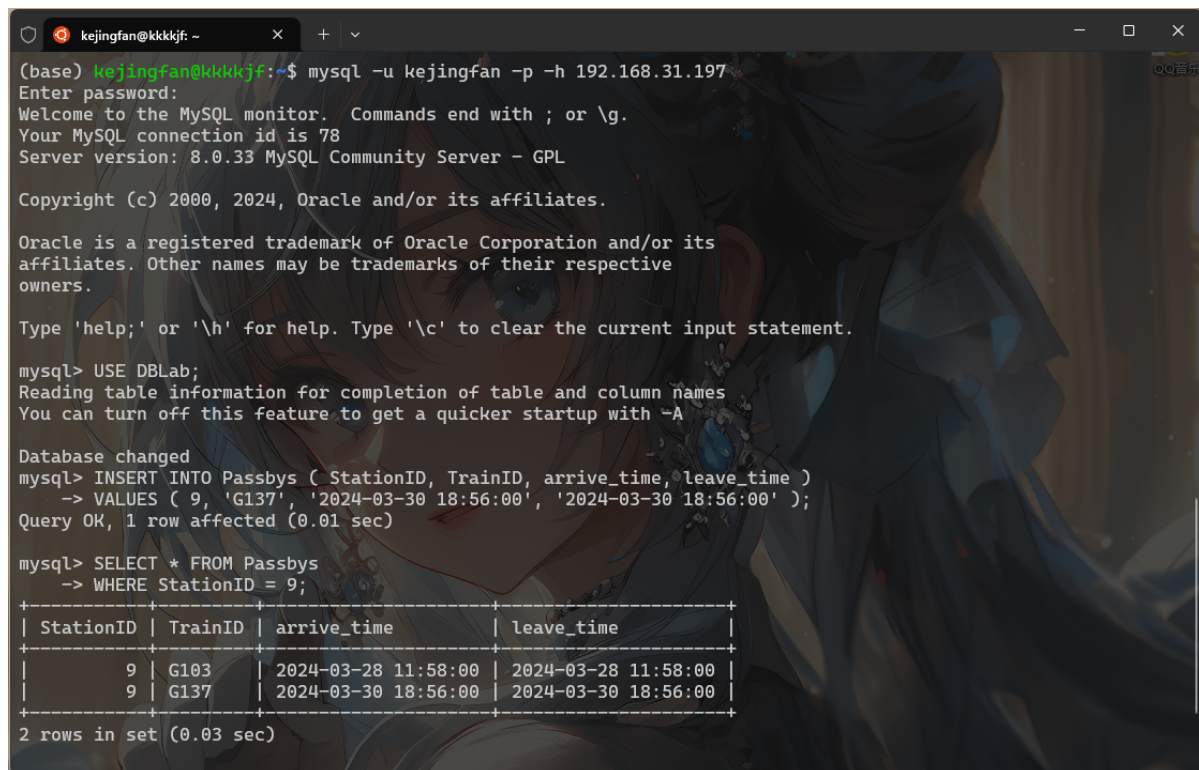
输入插入数据命令:

```
1 | mysql> INSERT INTO Passbys ( StationID, TrainID, arrive_time, leave_time )
2 |     -> VALUES ( 9, 'G137', '2024-03-30 18:56:00', '2024-03-30 18:56:00' );
```

查看插入结果:

```
1 | mysql> SELECT * FROM Passbys
2 |     -> WHERE StationID = 9;
3 |
4 | +-----+-----+-----+-----+
5 | | StationID | TrainID | arrive_time          | leave_time          |
6 | +-----+-----+-----+-----+
7 | |          9 | G103    | 2024-03-28 11:58:00 | 2024-03-28 11:58:00 |
8 | |          9 | G137    | 2024-03-30 18:56:00 | 2024-03-30 18:56:00 |
9 | +-----+-----+-----+-----+
10 | 2 rows in set (0.03 sec)
```

截图如下:



```
(base) kejingfan@kkkkjf:~$ mysql -u kejingfan -p -h 192.168.31.197
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 78
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE DBLab;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO Passbys ( StationID, TrainID, arrive_time, leave_time )
-> VALUES ( 9, 'G137', '2024-03-30 18:56:00', '2024-03-30 18:56:00' );
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM Passbys
-> WHERE StationID = 9;
+-----+-----+-----+-----+
| StationID | TrainID | arrive_time          | leave_time          |
+-----+-----+-----+-----+
|          9 | G103    | 2024-03-28 11:58:00 | 2024-03-28 11:58:00 |
|          9 | G137    | 2024-03-30 18:56:00 | 2024-03-30 18:56:00 |
+-----+-----+-----+-----+
2 rows in set (0.03 sec)
```


4. 题目4

请根据以上设计结果，重新完善与整理实验2中所设计的模式，可重新修改并提交作业2。
已在作业2中完善模式。

5. 题目5

请设计一些简单的单表查询、多表连接查询语句，查询表中的内容，并截图证明。
单表查询

```
1 mysql> SELECT * FROM Passbys
2     -> WHERE StationID = 9;
3
4 +-----+-----+-----+-----+
5 | StationID | TrainID | arrive_time          | leave_time          |
6 +-----+-----+-----+-----+
7 |          9 | G103    | 2024-03-28 11:58:00 | 2024-03-28 11:58:00 |
8 |          9 | G137    | 2024-03-30 18:56:00 | 2024-03-30 18:56:00 |
9 +-----+-----+-----+-----+
10 2 rows in set (0.03 sec)
11
12 mysql> SELECT * FROM Trains;
13
14 +-----+-----+-----+-----+
15 | ID   | Property | StationID_start | StationID_arrive |
16 +-----+-----+-----+-----+
17 | G103 | 1        | 1               | 9               |
18 | G137 | 0        | 1               | 9               |
19 +-----+-----+-----+-----+
20 2 rows in set (0.01 sec)
```

多表连接查询

```
1 SELECT
2     Passbys.TrainID,
3     Stations.`Name`,
4     Passbys.arrive_time,
5     Passbys.leave_time,
6     ( Passbys.leave_time - Passbys.arrive_time ) / 100 AS Stay_time
7 FROM
8     Passbys
9     JOIN Stations ON Passbys.StationID = Stations.ID;
```

```
mysql> SELECT
-> Passbys.TrainID,
-> Stations.`Name`,
-> Passbys.arrive_time,
-> Passbys.leave_time,
-> ( Passbys.leave_time - Passbys.arrive_time ) / 100 AS Stay_time
-> FROM
-> Passbys
-> JOIN Stations ON Passbys.StationID = Stations.ID;
```

TrainID	Name	arrive_time	leave_time	Stay_time
G103	上海虹桥	2024-03-28 11:58:00	2024-03-28 11:58:00	0.0000
G137	上海虹桥	2024-03-30 18:56:00	2024-03-30 18:56:00	0.0000
G103	北京南	2024-03-28 06:20:00	2024-03-28 06:20:00	0.0000
G103	南京南	2024-03-28 10:40:00	2024-03-28 10:43:00	3.0000
G103	宿州东	2024-03-28 09:28:00	2024-03-28 09:33:00	5.0000
G103	徐州东	2024-03-28 09:07:00	2024-03-28 09:09:00	2.0000
G103	无锡南	2024-03-28 11:27:00	2024-03-28 11:29:00	2.0000
G103	沧州西	2024-03-28 07:13:00	2024-03-28 07:15:00	2.0000
G103	济南西	2024-03-28 08:01:00	2024-03-28 08:04:00	3.0000
G103	蚌埠南	2024-03-28 09:56:00	2024-03-28 09:58:00	2.0000

10 rows in set (0.02 sec)

6. 题目6

请尝试练习在某些表上建立唯一索引和聚集索引的方法，并将建索引的语句写入建库脚本中。

```
1 CREATE TABLE Passengers (
2     ID BIGINT PRIMARY KEY,
3     `Name` VARCHAR ( 255 ) NOT NULL,
4     Phone_number BIGINT UNIQUE NOT NULL,
5     CHECK ( REGEXP_LIKE ( ID, '^\\d{18}$' ) ),
6     CHECK ( REGEXP_LIKE ( Phone_number, '^\\d{11}$' ) )
7 );
8
9 CREATE TABLE Users (
10     ID INT AUTO_INCREMENT PRIMARY KEY,
11     `Password` VARCHAR ( 255 ) NOT NULL,
12     Phone_number BIGINT UNIQUE NOT NULL,
13     CitizenID BIGINT UNIQUE NOT NULL,
14     FOREIGN KEY ( Phone_number ) REFERENCES Passengers ( Phone_number ) ON
15     DELETE CASCADE,
16     FOREIGN KEY ( CitizenID ) REFERENCES Passengers ( ID ) ON DELETE CASCADE
17 );
```

已在建库脚本中加入。

其中，含 PRIMARY KEY 的语句是聚集索引，含 UNIQUE 的语句是唯一索引。

7. 题目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方案，并在实验报告加以叙述。

目前的方案是使用爬虫爬取12306半个月内火车票的真实数据；对于购票的数据，使用Python随机生成。