**Database design of**

**charity running competition**

**Name: Ruipeng Jiao**

# 1. Software and Version

**Visio**: Design and draw E-R diagram.

**Powerdesigner 16.6**: Design the database and complete the SQL statement.

**Navicat 15.0.9**: It is convenient to control the database for relevant testing and operation.

**MySQL-installer-community-5.7.13.0.msi**

**MySQL Workbrench 8.0**: Database operation with Navicat.

**DBeaver**: Control the database for relevant testing and operation.

# 2. Overall Demand

Venue, Team’s agent, Team, Athlete, Customer, Race event, Ticket.

# 3. Specific demand analysis

## 3.1 Venue table

Storage venue number, number of staff and venue name. As the venue needs to provide a starting point and an ending point. Therefore, in addition to the venue address, the attribute also includes the start address and end address. The venue contacts information, the maximum number of visitors and the start date of the event should also be provided. As it is a charity running competition, it may include long-distance running sports such as marathon. Therefore, it is necessary to have the start time and end time of the activity. The number of tickets is also required. In addition, suppose that a charity running competition is held in only one venue.

## 3.2 Team table

The primary key is the team number. The table contains the agent label and unique team name. There are team description and team number. However, because the team may be less than 15 people, but not necessarily less than 15 people, there is no limit on the number of people. Others also contain e-mail, phone numbers and other information.

## 3.3 Race event table

The primary key is the competition item number. The table contains the competition item name, competition item description and venue number. The field number is used as the foreign key to link the relationship with the competition items.

## 3.4 Ticket table

The ticket should not have customer information, because a customer can buy multiple tickets. So, the ticket number is used as the primary key. The venue number is a foreign key.

## 3.5 Customer table

In consideration of safety and responsibility identification, managers need to know which venue a customer went to. Customer number, ticket number and venue number are used as joint primary keys. Other attributes include customer name, customer address, customer telephone, and others.

## 3.6 Staff table

Staff ID and venue ID are used as joint primary keys. Other attributes include staffname, stafftelephone and jobtitle. Moreover, an employee must work in a venue, that is, there is no case that the employee does not work.

## 3.7 Athlete table

Athleteid is the primary key. Other attributes include venueid, eventid, AgentID, teamid, athletename, and biology. Athletes participate in competitions and compete in venues. So the team doesn't go directly to the venue. In other words, the team has nothing to do with the venue and the game.

## 3.8 Team agent table

Because an agent can manage multiple teams, the administrator's information should appear in the team table, so it does not appear in the agent table. AgentID is used as the primary key. Other attributes include agentname and agenttelephone.

# 4. Data dictionary

## 4.1 Data structure（#Represents the primary key and some tables have federated primary keys）

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Data structure name** | **Meaning description** | **Constituent** |
| 1 | Venue | Venue table | #VenueID, EventID, VenueName, VenueTelephone, VenuseAddress, TicketSellNumber, Date, StratTime, FinishTime, MaxCustomerNumber, StaffNumber, StratlineAddress, FinishlineAddress, |
| 2 | Team | Team table | #TeamID, #AgentID, TeamName, TeamTelephone, TeamDescription, TeamPeopleNumber, TeamOther, TeamEmail |
| 3 | RaceEvent | RaceEvent table | #EventID, VenueID, EventName, EventDescription |
| 4 | Ticket | Ticket table | #TicketID, #VenueID |
| 5 | Customer | Customer table | #CustomerID, #VenueID, #TicketID, CustomerName, CustomerAddress, CustomerTelephone, CustomerOther |
| 6 | Staff | Staff table | #StaffID, #VenueID, StaffName, StaffTelephone, JobTitle |
| 7 | Athlete | Athlete table | #AthleteID, VenueID, EventID, AgentID, TeamID, AthleteName, Biography |
| 8 | TeamAgent | TeamAgent table | #AgentID, AgentName, AgentTelephone |

## 4.2 Relationship between data structures

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Data structure1** | **Data structure2** | **Relationship** |
| 1 | TeamAgent | Team | 1 to many |
| 2 | Team | Athlete | 1 to many |
| 3 | Athlete | RaceEvent | 1 to many |
| 4 | Athlete | Venue | 1 to many |
| 5 | RaceEvent | Venue | 1 to 1 |
| 6 | Venue | Ticket | 1 to many |
| 7 | Ticket | Customer | 1 to many |
| 8 | Staff | Venue | 1 to many |

## 4.3 Data items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Belonging table** | **Data item name** | **storage structure** | **Data item meaning** |
| 1 | TeamAgent | AgentID | char(10) | Team Agent ID |
| 2 | TeamAgent | AgentName | char(50) | Team agent name |
| 3 | TeamAgent | AgentTelephone | Int | Team agent contact information |
| 4 | Team | TeamID | char(10) | Team ID |
| 5 | Team | TeamName | char(20) | Team name |
| 6 | Team | TeamTelephone | Int | Team contact information |
| 7 | Team | TeamDescription | char(100) | Team description information |
| 8 | Team | TeamPeopleNumber | Int | Number of people included in the team |
| 9 | Team | TeamOther | char(100) | Additional content to be added by the team |
| 10 | Team | TeamEmail | char(50) | Team e-mail information |
| 11 | Athlete | AthleteID | char(10) | Athlete ID |
| 12 | Athlete | AthleteName | char(50) | Athlete Name |
| 13 | Athlete | Biography | char(200) | Athlete biography |
| 14 | RaceEvent | EventID | char(10) | Event ID |
| 15 | RaceEvent | EventDescription | char(100) | Event other description |
| 16 | Venue | VenueID | char(10) | Venue ID |
| 17 | Venue | VenueName | char(50) | Venue Name |
| 18 | Venue | VenueTelephone | Int | Venue contact information |
| 19 | Venue | VenuseAddress | char(100) | Venue address |
| 20 | Venue | TicketSellNumber | Int | Number of tickets sold at the venue |
| 21 | Venue | Date | date | The date on which a competition is held at the venue |
| 22 | Venue | StratTime | datatime | The starting time of the competition held at the venue |
| 23 | Venue | FinishTime | datatime | The ending time of the competition held at the venue |
| 24 | Venue | MaxCustomerNumber | Int | Maximum capacity of the people |
| 25 | Venue | StaffNumber | Int | Number of venue staff |
| 26 | Venue | StratlineAddress | char(50) | The starting address of the competition held at the venue |
| 27 | Venue | FinishlineAddress | char(50) | The ending address of the competition held at the venue |
| 28 | Staff | StaffID | char(10) | Staff ID |
| 29 | Staff | StaffName | char(50) | Staff name |
| 30 | Staff | StaffTelephone | int | Staff telephone number |
| 31 | Staff | JobTitle | char(50) | The staff of job title |
| 32 | Ticket | TicketID | char(10) | Ticket ID |
| 33 | Customer | CustomerID | Int | Customer ID |
| 34 | Customer | CustomerName | char(20) | Customer name |
| 35 | Customer | CustomerAddress | char(50) | The address of customer |
| 36 | Customer | CustomerTelephone | int | The telephone of customer |
| 37 | Customer | CustomerOther | char(100) | Other information about customer |

## 4.4 Entity-Relationship diagram

According to the above information, the Entity-Relationship diagram can be obtained.

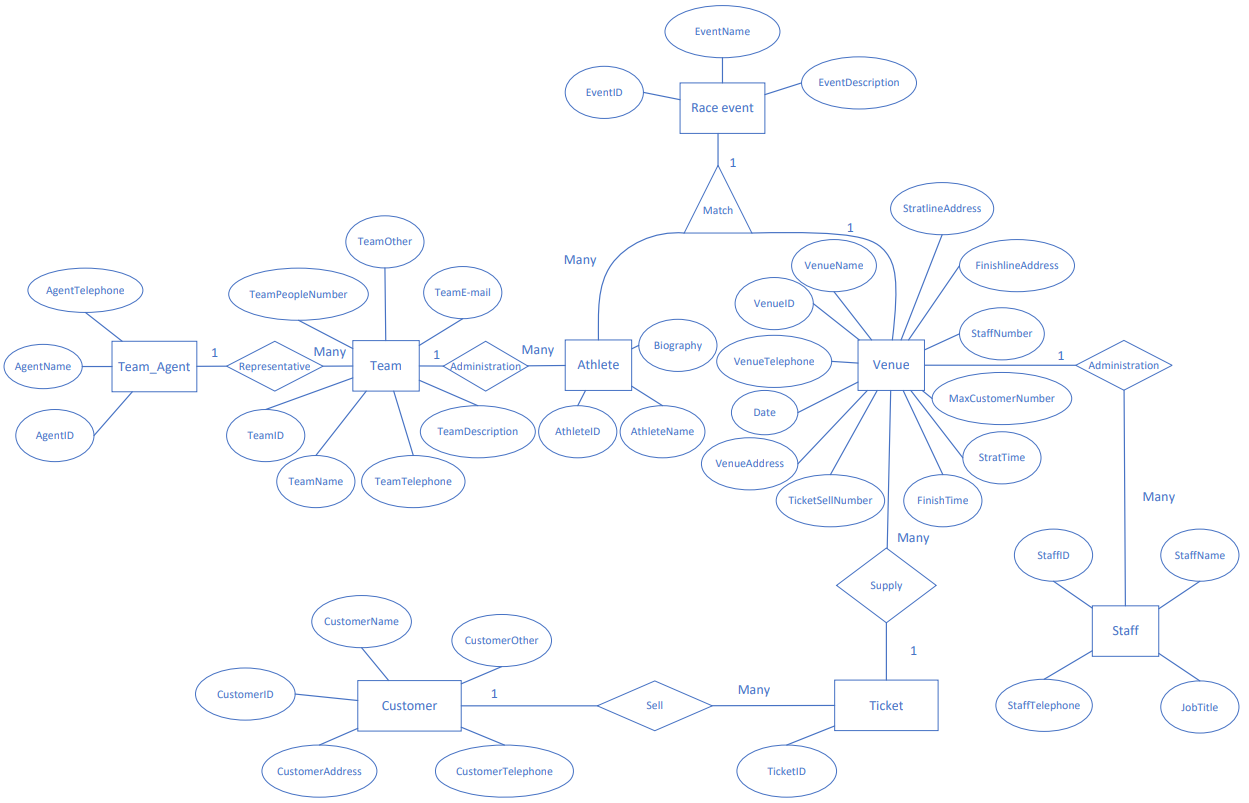


Figure 4.1 Entity-Relationship diagram

# 5. SQL statement

/\*==============================================================\*/

/\* DBMS name: MySQL 5.7 \*/

/\* Created on: 2021/10/6 22:58:22 \*/

/\*==============================================================\*/

/\*==============================================================\*/

/\*Create Database\*/

/\*==============================================================\*/

drop database if exists race;

create database race;

use race;

/\*==============================================================\*/

/\* Table: athlete \*/

/\*==============================================================\*/

create table athlete

(

VenueID char(10) not null comment '',

EventID char(10) not null comment '',

AgentID char(10) not null comment '',

TeamID char(10) not null comment '',

AthleteID char(10) not null comment '',

AthleteName char(50) not null comment '',

Biography char(200) not null comment '',

primary key (AthleteID)

);

INSERT INTO `athlete` VALUES ('V1', 'E1', '1', 'N1', 'Ac001', 'JIAORUIPENG', 'Mr.nobody');

INSERT INTO `athlete` VALUES ('V1', 'E1', '2', 'N3', 'Ac052', 'Dennisjiao', 'Mr.right');

INSERT INTO `athlete` VALUES ('V2', 'E4', '3', 'N5', 'Am100', 'SuBingtian', 'Greatman');

INSERT INTO `athlete` VALUES ('V2', 'E4', '4', 'N4', 'Am105', 'Jackson', 'UUS');

/\*==============================================================\*/

/\* Table: customer \*/

/\*==============================================================\*/

create table customer

(

VenueID char(10) not null comment '',

TicketID char(20) not null comment '',

CustomerID int not null comment '',

CustomerName char(20) not null comment '',

CustomerAddress char(50) not null comment '',

CustomerTelephone int not null comment '',

CustomerOther char(100) comment '',

primary key (VenueID, TicketID, CustomerID)

);

INSERT INTO `customer` VALUES ('V1', 'TV001', 546, 'jackson', 'NE1 1TT', 451313513, '');

INSERT INTO `customer` VALUES ('V1', 'TV002', 13, 'DAJIDA', 'NE1 1TT', 435131, NULL);

INSERT INTO `customer` VALUES ('V2', 'TN001', 20, 'dawd', 'NE1 1AD', 4863, NULL);

INSERT INTO `customer` VALUES ('V2', 'TN002', 123, 'ADAWD', 'NE1 1TT', 54684, 'VIP');

INSERT INTO `customer` VALUES ('V2', 'TV005', 7, 'QWDA', 'NE1 11M', 515543, NULL);

/\*==============================================================\*/

/\* Table: raceEvent \*/

/\*==============================================================\*/

create table raceEvent

(

EventID char(10) not null comment '',

VenueID char(10) not null comment '',

EventName char(100) not null comment '',

EventDescription char(100) not null comment '',

primary key (EventID)

);

INSERT INTO `raceevent` VALUES ('E1', 'V1', 'Men\'s Marathon','This is MEN race event');

INSERT INTO `raceevent` VALUES ('E2', 'V2', 'Women\'s Marathon','This is WOMEN race event');

INSERT INTO `raceevent` VALUES ('E4', 'V4', '100 Meters','This is 100 Meter race event');

/\*==============================================================\*/

/\* Table: staff \*/

/\*==============================================================\*/

create table staff

(

VenueID char(10) not null comment '',

StaffID char(10) not null comment '',

StaffName char(50) not null comment '',

StaffTelephone int not null comment '',

JobTitle char(50) not null comment '',

primary key (StaffID)

);

INSERT INTO `staff` VALUES ('V1', 'S12', 'StaffNO1', 1135431, 'Highlevel');

INSERT INTO `staff` VALUES ('V2', 'S123', 'john', 16549, 'Mid');

INSERT INTO `staff` VALUES ('V1', 'S32', 'sno2', 13532, 'low');

/\*==============================================================\*/

/\* Table: team \*/

/\*==============================================================\*/

create table team

(

AgentID char(10) not null comment '',

TeamID char(10) not null comment '',

TeamName char(20) not null comment '',

TeamTelephone int not null comment '',

TeamDescription char(100) not null comment '',

TeamPeopleNumber int not null comment '',

TeamOther char(100) not null comment '',

TeamEmail char(50) not null comment '',

primary key (AgentID, TeamID)

);

INSERT INTO `team` VALUES ('1', 'N1', 'RNG', 123123, 'OK', 32113, '123DAD', '123DA');

INSERT INTO `team` VALUES ('1', 'N2', 'OMG', 32151, 'Nice', 123123, 'asd123', 'dawd1');

INSERT INTO `team` VALUES ('2', 'N3', 'IG', 123546, 'GREAT', 131215, 'AWD1', '12EDA');

INSERT INTO `team` VALUES ('3', 'N5', 'FPX', 456123, 'TEST', 15656, 'DAW12', '12DAWD');

INSERT INTO `team` VALUES ('4', 'N4', 'EDG', 654123, 'FINE', 54654, 'DASD123', '124A');

/\*==============================================================\*/

/\* Table: teamagent \*/

/\*==============================================================\*/

create table teamagent

(

AgentID char(10) not null comment '',

AgentName char(50) not null comment '',

AgentTelephone int not null comment '',

primary key (AgentID)

);

INSERT INTO `teamagent` VALUES ('1', 'Kiven', 123123);

INSERT INTO `teamagent` VALUES ('2', 'John', 123232);

INSERT INTO `teamagent` VALUES ('3', 'Mike', 323121);

INSERT INTO `teamagent` VALUES ('4', 'Jack', 123123);

/\*==============================================================\*/

/\* Table: ticket \*/

/\*==============================================================\*/

create table ticket

(

VenueID char(10) not null comment '',

TicketID char(20) not null comment '',

primary key (VenueID, TicketID)

);

INSERT INTO `ticket` VALUES ('V1', 'TV001');

INSERT INTO `ticket` VALUES ('V1', 'TV002');

INSERT INTO `ticket` VALUES ('V1', 'TV003');

INSERT INTO `ticket` VALUES ('V1', 'TV004');

INSERT INTO `ticket` VALUES ('V2', 'TN001');

INSERT INTO `ticket` VALUES ('V2', 'TN002');

INSERT INTO `ticket` VALUES ('V2', 'TV005');

/\*==============================================================\*/

/\* Table: venue \*/

/\*==============================================================\*/

create table venue

(

EventID char(10) not null comment '',

VenueID char(10) not null comment '',

VenueName char(50) not null comment '',

VenueTelephone int not null comment '',

VenueAddress char(40) not null comment '',

TicketSellNumber int not null comment '',

Date date not null comment '',

StratTime datetime not null comment '',

FinishTime datetime not null comment '',

MaxCustomerNumber int not null comment '',

StaffNumber int not null comment '',

StratlineAddress char(50) not null comment '',

FinishlineAddress char(50) not null comment '',

CHECK (staffNumber<=5),

primary key (VenueID)

);

INSERT INTO `venue` VALUES ('E1', 'V1', 'NCL', 1512315618, 'address1', 500, '2021-10-06', '2021-10-07 23:26:32', '2021-10-08 23:26:37', 1000, 2, 'strataddress', 'finishaddress');

INSERT INTO `venue` VALUES ('E2', 'V2', 'NCL001', 153153, 'address2', 530, '2021-10-04', '2021-10-04 09:00:00', '2021-10-04 12:00:00', 2000, 1, 'strataddress1', 'finishaddress1');

alter table Athlete add constraint FK\_ATHLETE\_ADMINISTR\_TEAM foreign key (AgentID, TeamID)

references Team (AgentID, TeamID) on delete restrict on update restrict;

alter table Athlete add constraint FK\_ATHLETE\_MATCH\_RACEEVEN foreign key (EventID)

references RaceEvent (EventID) on delete restrict on update restrict;

alter table Athlete add constraint FK\_ATHLETE\_MATCH2\_VENUE foreign key (VenueID)

references Venue (VenueID) on delete restrict on update restrict;

alter table Customer add constraint FK\_CUSTOMER\_SELL\_TICKET foreign key (VenueID, TicketID)

references Ticket (VenueID, TicketID) on delete restrict on update restrict;

alter table Staff add constraint FK\_STAFF\_ADMINISTR\_VENUE foreign key (VenueID)

references Venue (VenueID) on delete restrict on update restrict;

alter table Team add constraint FK\_TEAM\_REPRESENT\_TEAMAGEN foreign key (AgentID)

references TeamAgent (AgentID) on delete restrict on update restrict;

alter table Ticket add constraint FK\_TICKET\_SUPPLY\_VENUE foreign key (VenueID)

references Venue (VenueID) on delete restrict on update restrict;

alter table Venue add constraint FK\_VENUE\_MATCH3\_RACEEVEN foreign key (EventID)

references RaceEvent (EventID) on delete restrict on update restrict;

SELECT \* from athlete;

SELECT \* from customer;

SELECT \* from raceevent;

SELECT \* from staff;

SELECT \* from team;

SELECT \* from teamagent;

SELECT \* from ticket;

SELECT \* from venue;

/\*==============================Finish=============================\*/

# 6. Project screenshot

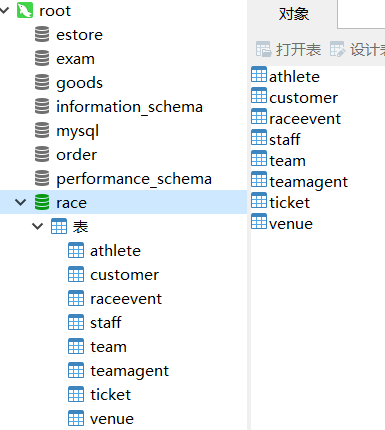


Figure 6.1 Database

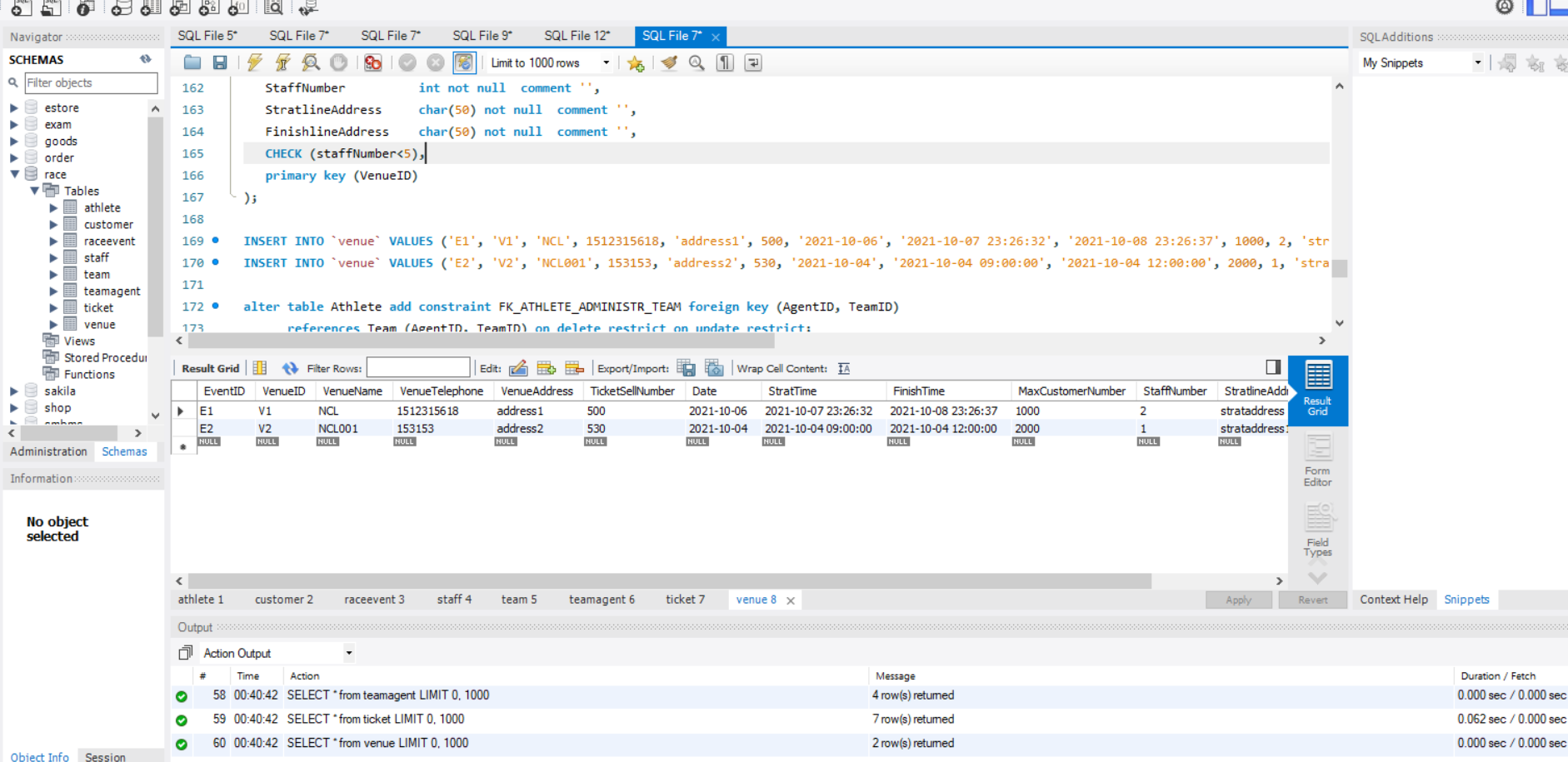


Figure 6.2 Successfully run SQL script

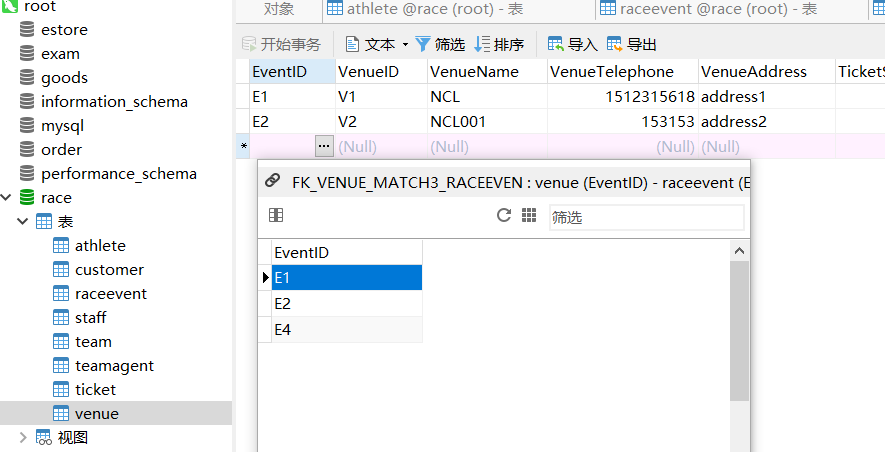


Figure 6.3 Foreign keys work when adding data

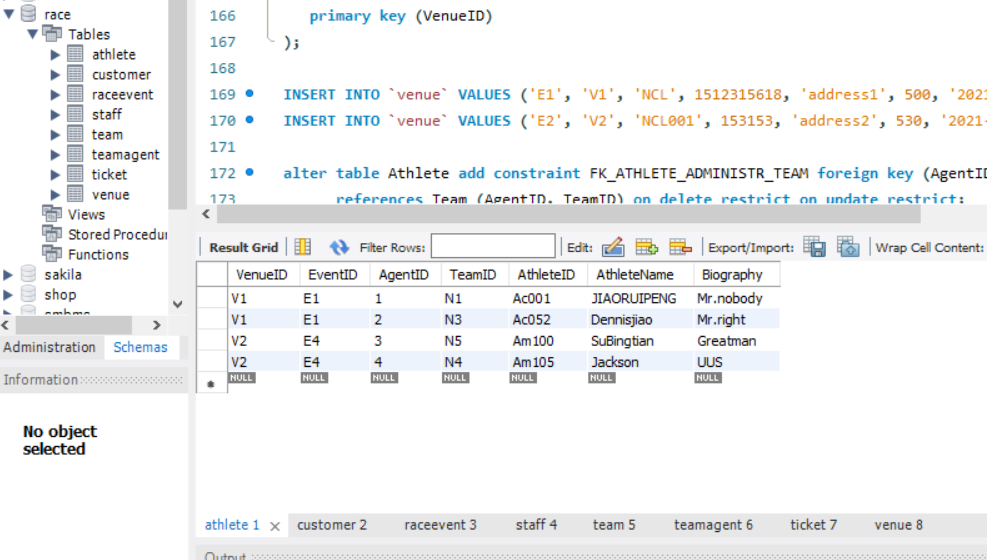


Figure 6.4 Athlete table

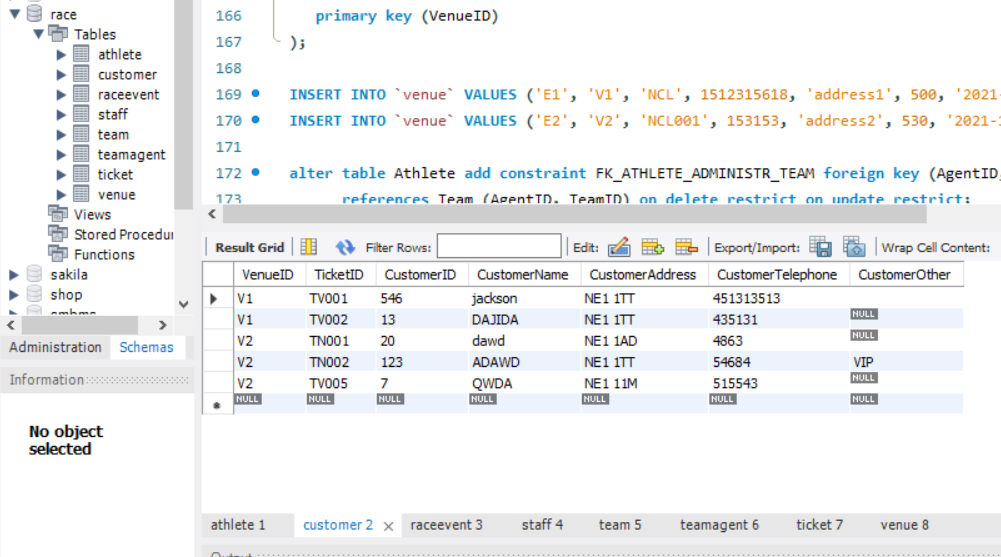


Figure 6.5 Customer table

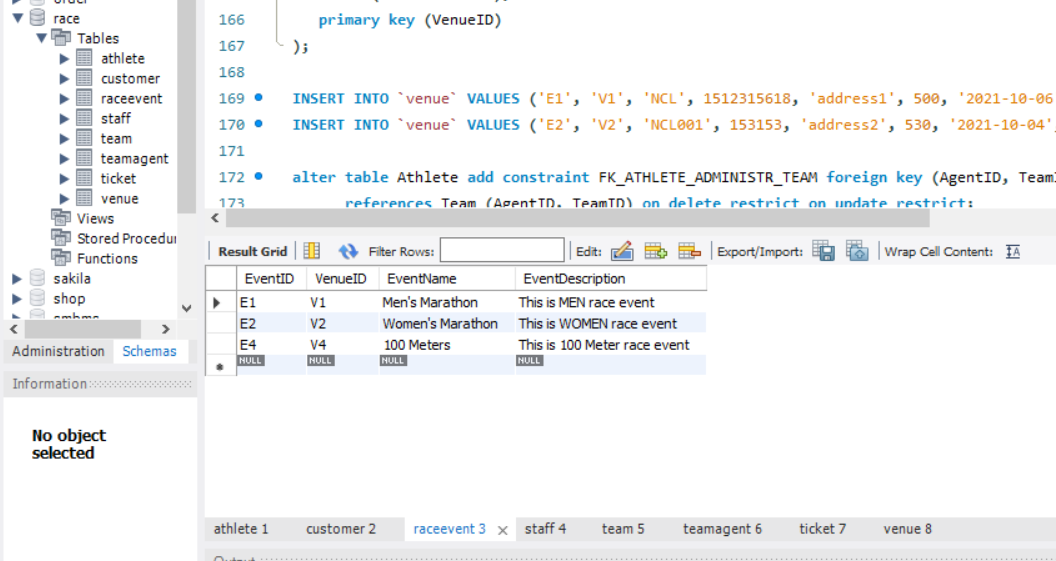


Figure 6.6 Raceevent table

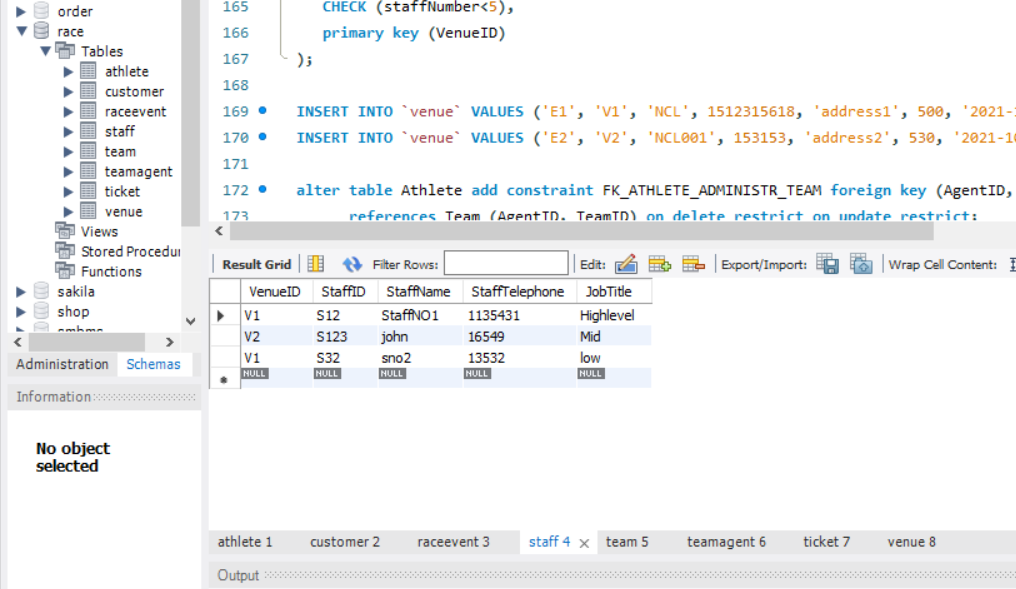


Figure 6.7 Staff table

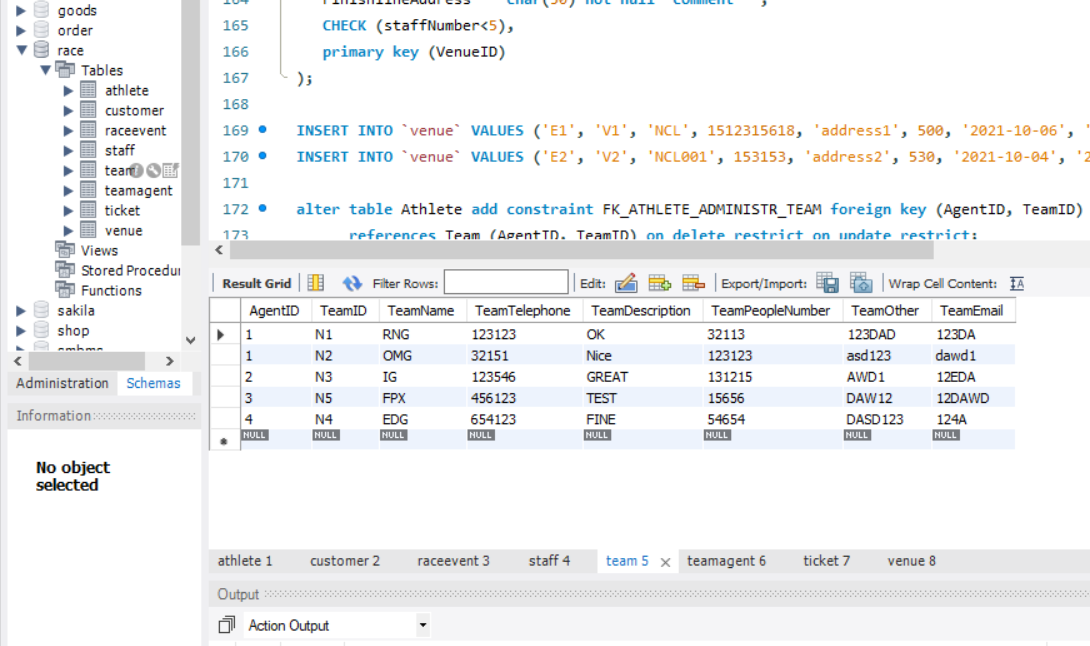


Figure 6.8 Team table

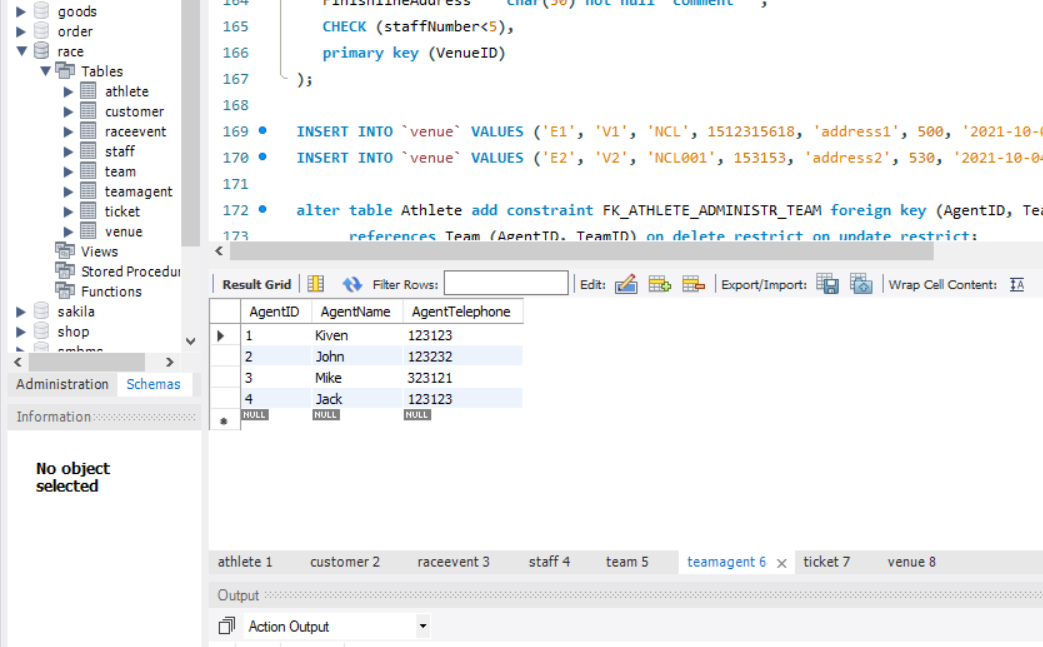


Figure 6.9 Teamagent table

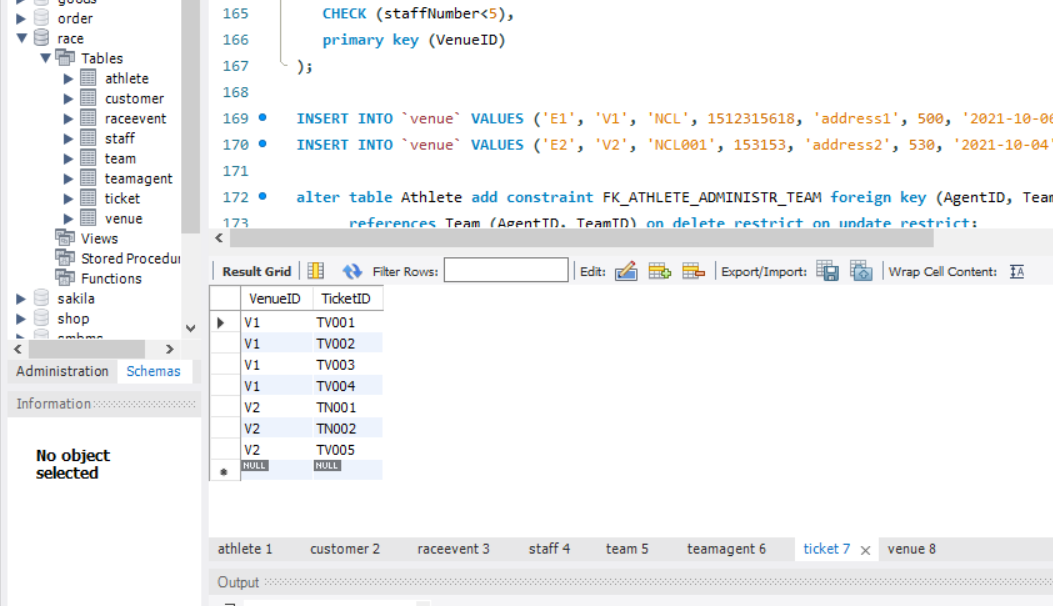


Figure 6.10 Ticket table

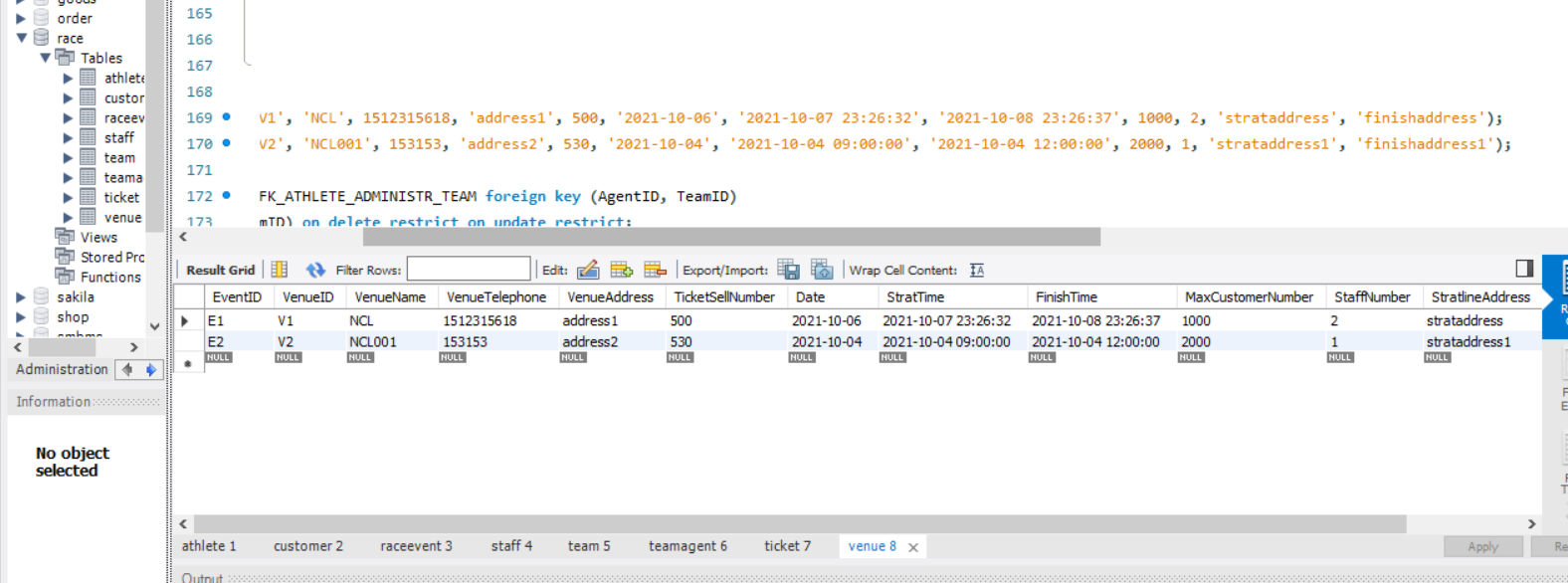


Figure 6.11 Venue table