CRICCBOOK PROJECT DOCUMENTATION

Table ofContents

CRICCBOOK

A PROJECT Done By

Name	Roll number
Anurag	1602-19-737-127
Harshit	1602-19-737-132
Shiek Amaan	1602-19-737-166

ABSTRACT

The project objective is to develop a platform where a cricket enthusiast can book Match tickets online in a very clean and user-friendly UI. **CriccBook** is a **Desktop-based application** that can be accessed throughout the Net and can be accessed by anyone who has an internet connection. This application will automate the booking of tickets and inquiries about the availability of the tickets from the comfort of their home.

Watching cricket in the stadium with family and friends is a way different experience and one of the best mediums of entertainment in this present world scenario. As everything comes at a cost, all this excitement vanishes after standing for hours in long queues to get tickets booked.

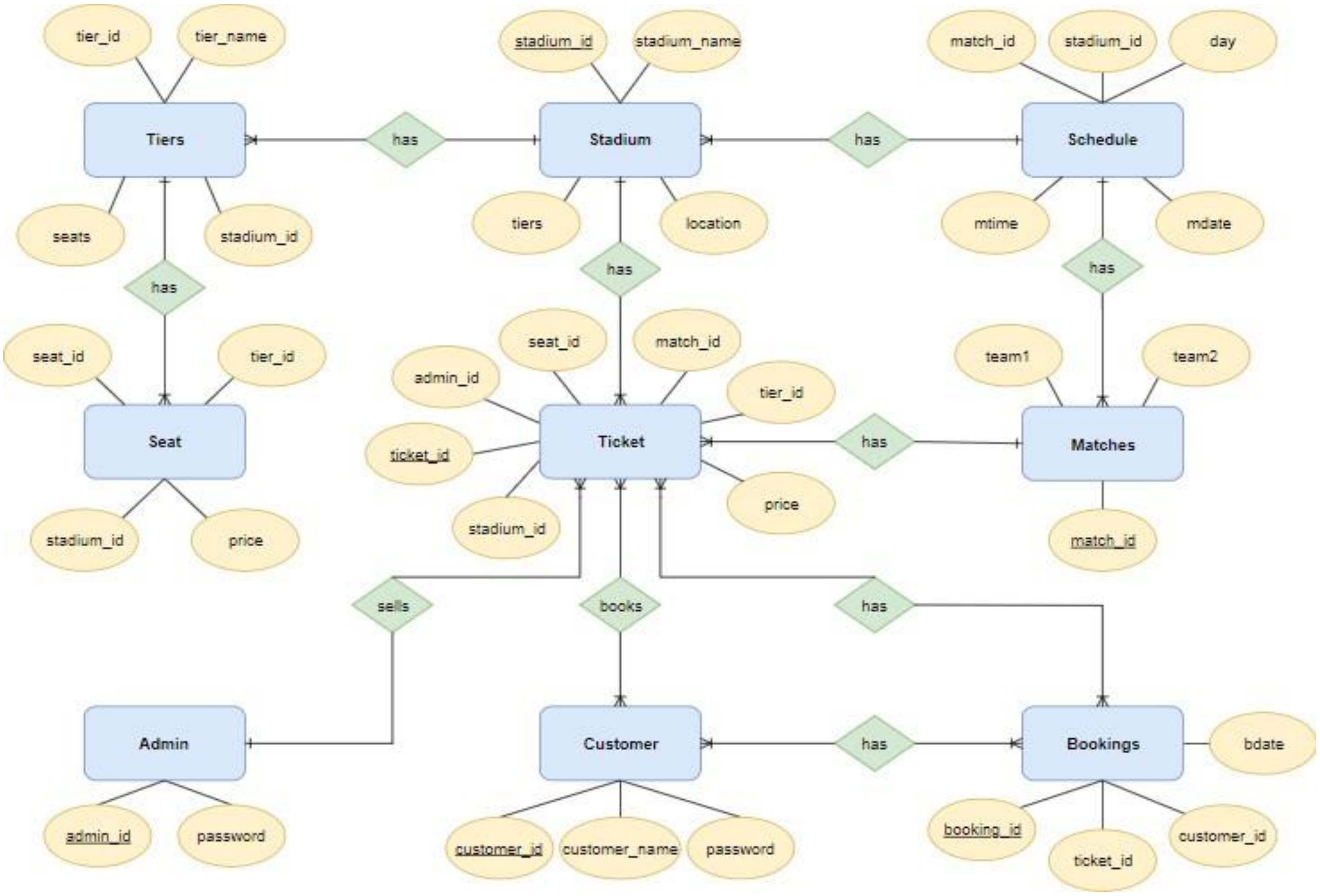
This application provides complete information regarding ongoing and coming up matches with full details of match timings, seat availability, ticket cost with respective stadium stands. User can book tickets for individual and also for groups, where as admin can regularly maintain and manage the information.

Thus our application, **CriccBook** is one of the best application for every cricket fan who don’t want to stand in long queues and spend a lot of time. Fans can book tickets online at any time of day and any day of the week. It also provides the option to cancel the tickets which are reserved previously

PROJECT DESCRIPTION:

CriccBook is a Desktop application for booking cricket match tickets online. Customers can use our application to know about current match locations and their schedules and also provides all other additional information. The objective of **CriccBook** is to create an application for match ticket booking that allows users to know about upcoming matches, match locations, available seats, their schedules, ticket price and many more. In the booking process when a customer selects his city then cinema theatres of that location are filtered. In the next step, he/she selects the theatre where he/she wishes to see a movie, then selects the movie and other details like show date, show time, screen, class, price, and no. of tickets. Based on given parameters a layout of seat status is visible to the customer. Now the customer can select desired seat locations as per the number of seats required and then tickets get booked. The Administrator will be able to see booked and canceled tickets

ER DIAGRAM:


tier_id
tier_name
Tiers
has
Stadium
has
Schedule
stadium_id
stadium_name
match_id
stadium_id
day
seats
stadium_id
tiers
location
mtime
mdate
has
has
has
seat_id
tier_id
Seat
stadium_id
price
admin_id
seat_id
match_id
tier_id
ticket_id
Ticket
stadium_id
price
has
Matches
team1
team2
match_id
sells
books
has
Admin
admin_id
password
Customer
customer_id
customer_name
password
has
Bookings
bdate
booking_id
ticket_id
customer_id

SOFTWARE REQUIREMENTS:

) Operating System – Windows 7 and above

) JAVA Development Kit (JDK 8)

) MySQL Database version 5.7 and above with appropriate Driver Connectors

LIST OF TABLES:

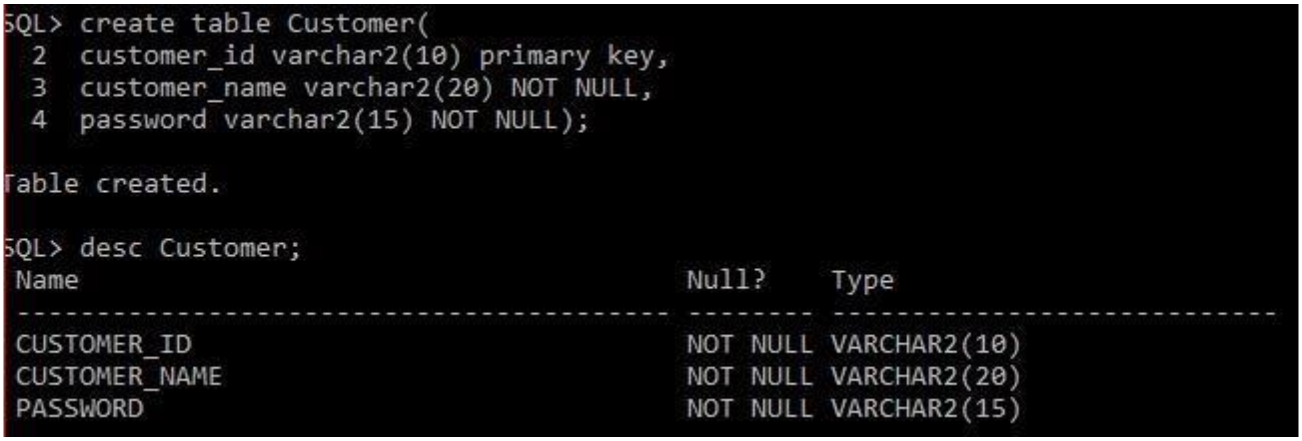
- ADMIN
- CUSTOMER
- MATCHES
- STADIUM
- SCHEDULE
- TIER
- SEAT
- TICKET
- BOOKINGS

CUSTOMER TABLE:

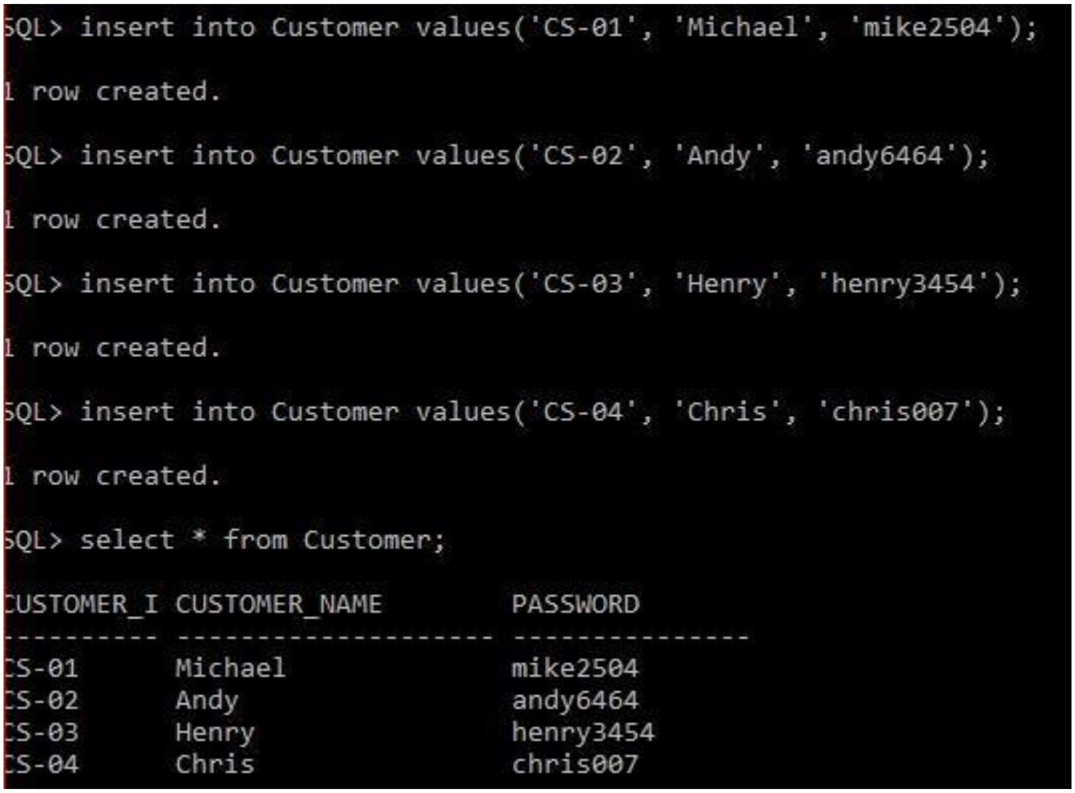
LIST OF ATTRIBUTES:

- ✓ customer_id VARCHAR2(10)
- ✓ customer_name VARCHAR2(10)
- ✓ password VARCHAR2(20)

PRIMAY KEY: customer_id



```
SQL> create table Customer(
  2  customer_id varchar2(10) primary key,
  3  customer_name varchar2(20) NOT NULL,
  4  password varchar2(15) NOT NULL);

Table created.

SQL> desc Customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(10)
 CUSTOMER_NAME                             NOT NULL VARCHAR2(20)
 PASSWORD                                  NOT NULL VARCHAR2(15)
```



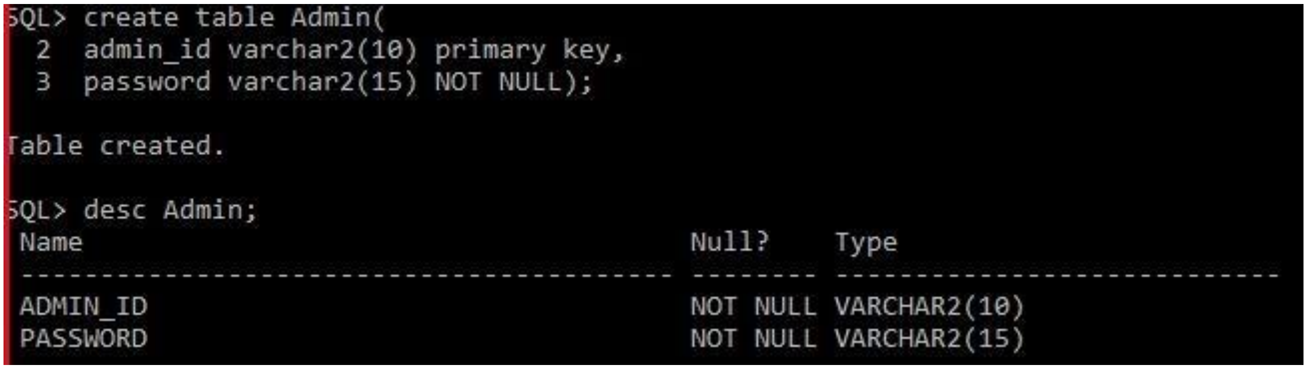
```
SQL> insert into Customer values('CS-01', 'Michael', 'mike2504');

1 row created.

SQL> insert into Customer values('CS-02', 'Andy', 'andy6464');

1 row created.

SQL> insert into Customer values('CS-03', 'Henry', 'henry3454');

1 row created.

SQL> insert into Customer values('CS-04', 'Chris', 'chris007');

1 row created.

SQL> select * from Customer;

CUSTOMER_I CUSTOMER_NAME        PASSWORD
---------- -------------------- ---------------
CS-01      Michael              mike2504
CS-02      Andy                 andy6464
CS-03      Henry                henry3454
CS-04      Chris                chris007
```

ADMIN TABLE:

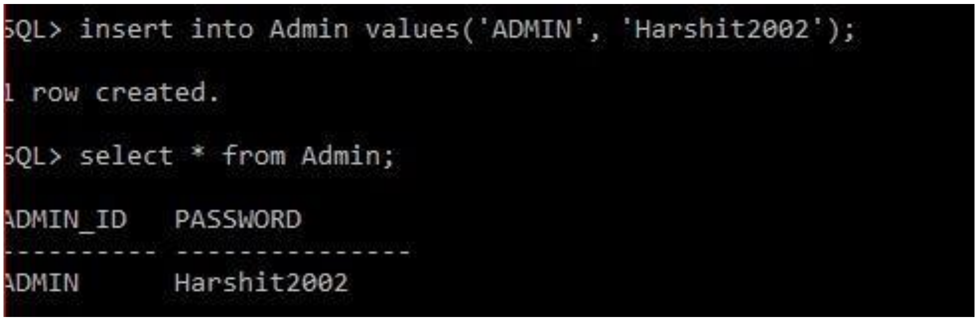
LIST OF ATTRIBUTES:

- ✓ admin_id VARCHAR2(10)
- ✓ password VARCHAR2(15)

PRIMARY KEY: admin_id



```
SQL> create table Admin(
  2  admin_id varchar2(10) primary key,
  3  password varchar2(15) NOT NULL);

Table created.

SQL> desc Admin;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADMIN_ID                                  NOT NULL VARCHAR2(10)
 PASSWORD                                  NOT NULL VARCHAR2(15)
```



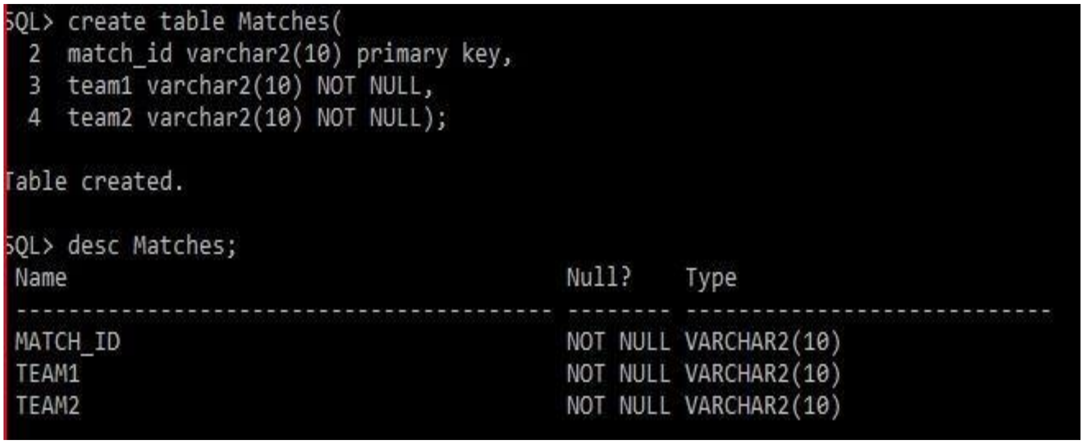
```
SQL> insert into Admin values('ADMIN', 'Harshit2002');

1 row created.

SQL> select * from Admin;

ADMIN_ID   PASSWORD
---------- ---------------
ADMIN      Harshit2002
```

MATCHES TABLE:

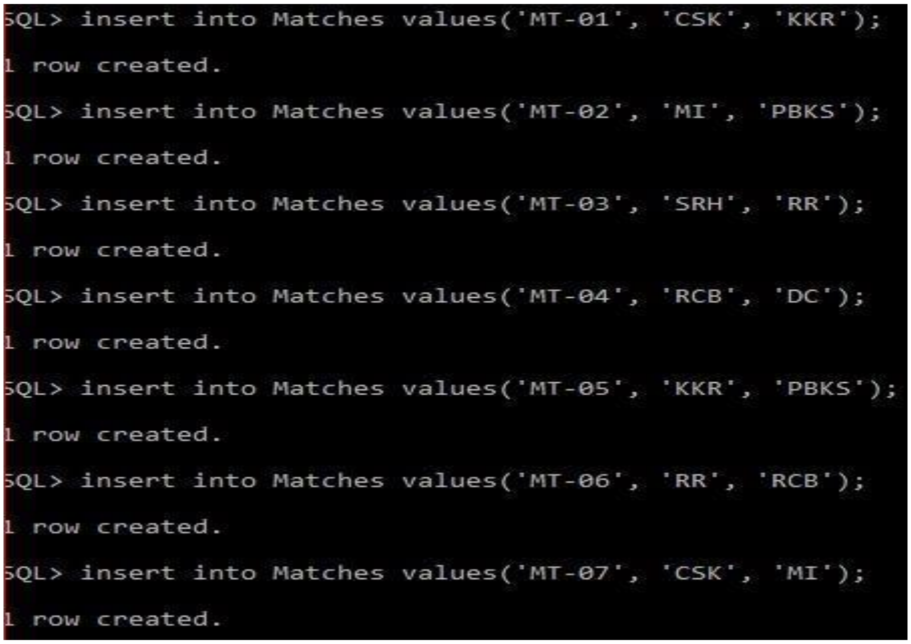
LIST OF ATTRIBUTES:

- ✓ match_id VARCHAR2(10)
- ✓ team1 VARCHAR2(10)
- ✓ team2 VARCHAR2(10)

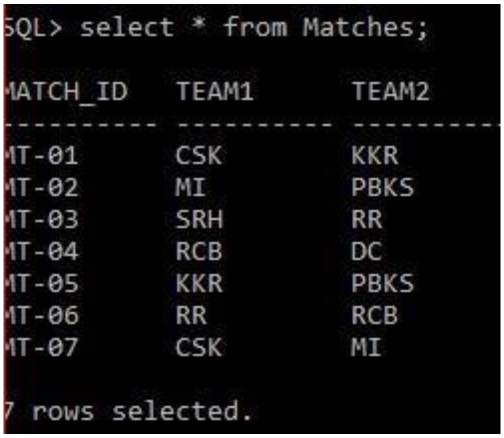
PRIMARY KEY: match_id



```
SQL> create table Matches(
  match_id varchar2(10) primary key,
  team1 varchar2(10) NOT NULL,
  team2 varchar2(10) NOT NULL);

Table created.

SQL> desc Matches;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MATCH_ID                                  NOT NULL VARCHAR2(10)
 TEAM1                                     NOT NULL VARCHAR2(10)
 TEAM2                                     NOT NULL VARCHAR2(10)
```



```
SQL> insert into Matches values('MT-01', 'CSK', 'KKR');

1 row created.

SQL> insert into Matches values('MT-02', 'MI', 'PBKS');

1 row created.

SQL> insert into Matches values('MT-03', 'SRH', 'RR');

1 row created.

SQL> insert into Matches values('MT-04', 'RCB', 'DC');

1 row created.

SQL> insert into Matches values('MT-05', 'KKR', 'PBKS');

1 row created.

SQL> insert into Matches values('MT-06', 'RR', 'RCB');

1 row created.

SQL> insert into Matches values('MT-07', 'CSK', 'MI');

1 row created.
```



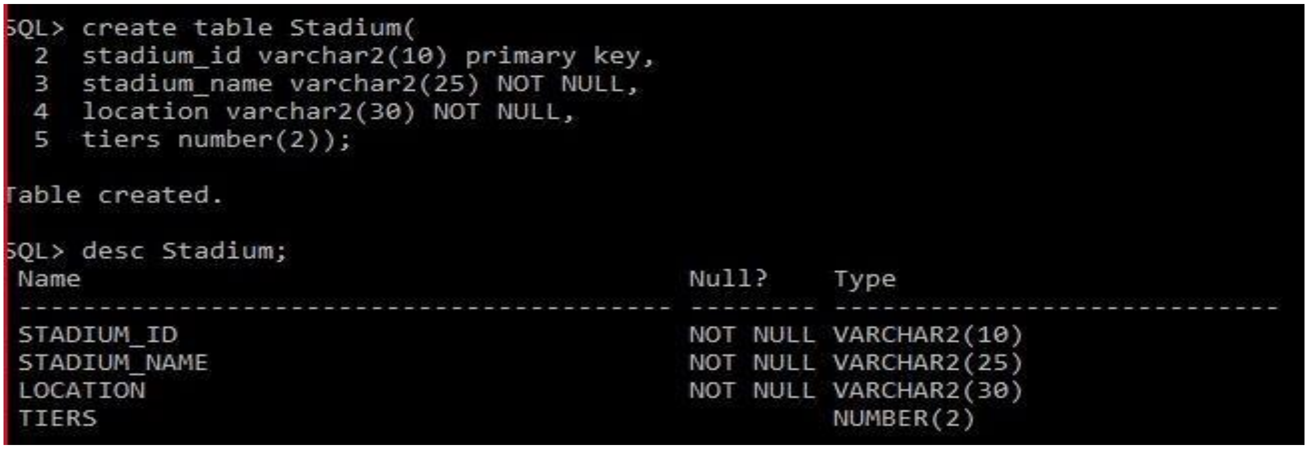
```
SQL> select * from Matches;

MATCH_ID   TEAM1      TEAM2
---------- ---------- ----------
MT-01      CSK        KKR
MT-02      MI         PBKS
MT-03      SRH        RR
MT-04      RCB        DC
MT-05      KKR        PBKS
MT-06      RR         RCB
MT-07      CSK        MI

7 rows selected.
```

STADIUM TABLE:

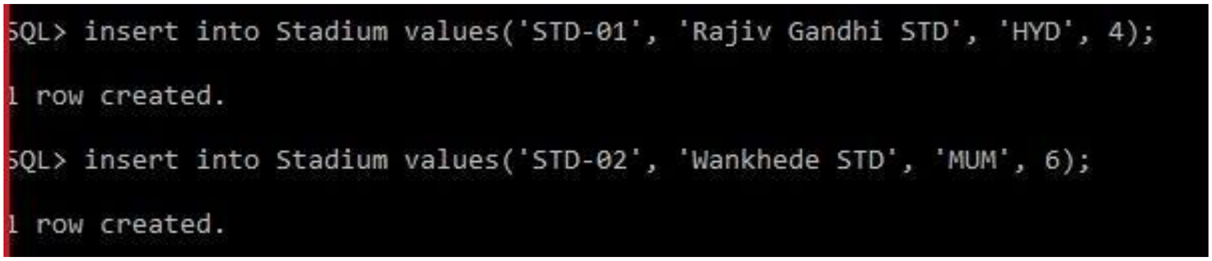
LIST OF ATTRIBUTES:

- ✓ stadium_id VARCHAR2(10)
- ✓ stadium_name VARCHAR2(25)
- ✓ location VARCHAR2(30)
- ✓ tiers NUMBER(2)

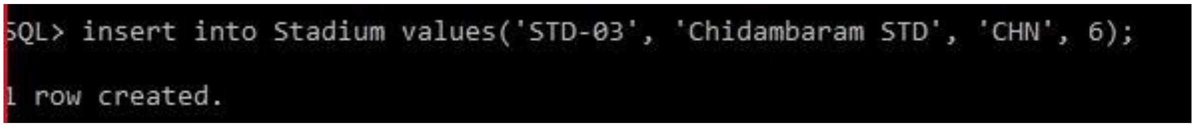
PRIMARY KEY: stadium_id



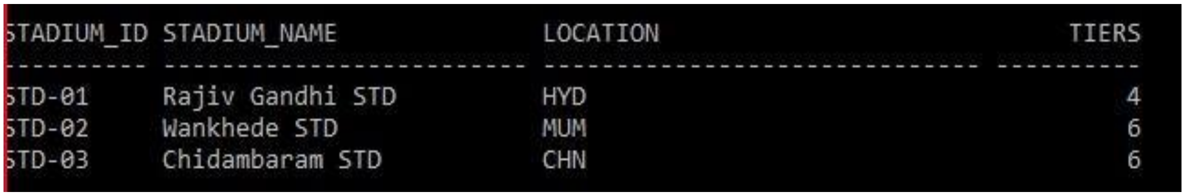
```
SQL> create table Stadium(
  2  stadium_id varchar2(10) primary key,
  3  stadium_name varchar2(25) NOT NULL,
  4  location varchar2(30) NOT NULL,
  5  tiers number(2));

Table created.

SQL> desc Stadium;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STADIUM_ID                                NOT NULL VARCHAR2(10)
 STADIUM_NAME                              NOT NULL VARCHAR2(25)
 LOCATION                                  NOT NULL VARCHAR2(30)
 TIERS                                              NUMBER(2)
```



```
SQL> insert into Stadium values('STD-01', 'Rajiv Gandhi STD', 'HYD', 4);

1 row created.

SQL> insert into Stadium values('STD-02', 'Wankhede STD', 'MUM', 6);

1 row created.
```



```
SQL> insert into Stadium values('STD-03', 'Chidambaram STD', 'CHN', 6);

1 row created.
```



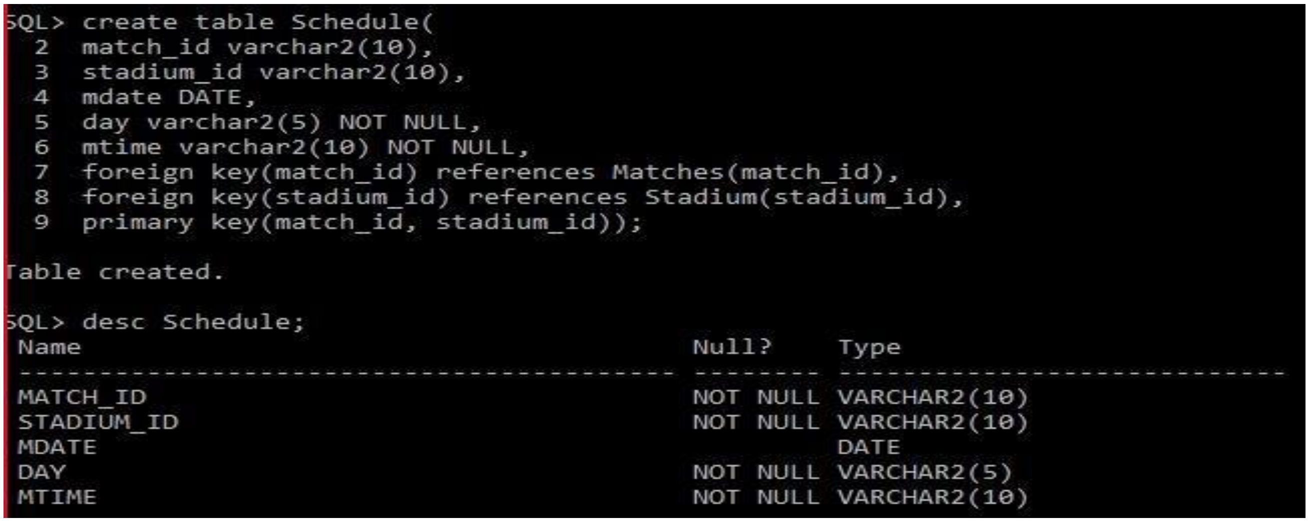
```
STADIUM_ID STADIUM_NAME              LOCATION                            TIERS
---------- ------------------------- ------------------------------ ----------
STD-01     Rajiv Gandhi STD          HYD                                     4
STD-02     Wankhede STD              MUM                                     6
STD-03     Chidambaram STD           CHN                                     6
```

SCHEDULE TABLE:

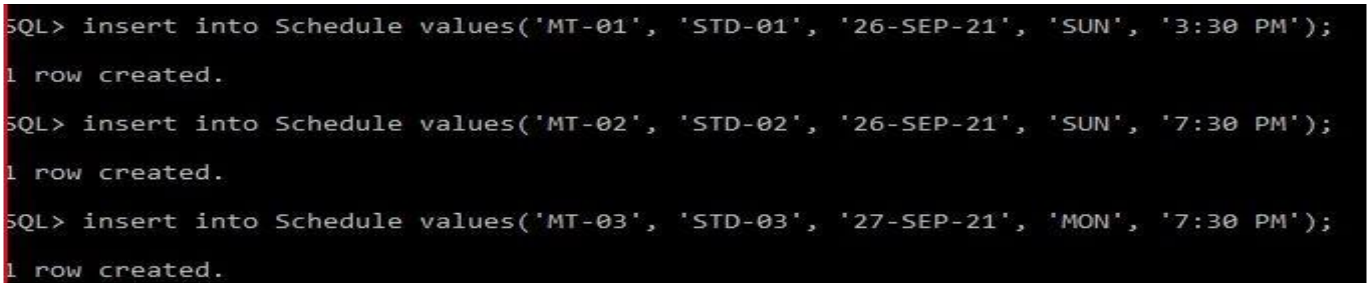
LIST OF ATTRIBUTES:

- ✓ match_id VARCHAR2(10)
- ✓ stadium_id VARCHAR2(10)
- ✓ mdate DATE
- ✓ day VARCHAR2(5)
- ✓ mtime VARCHAR2(10)

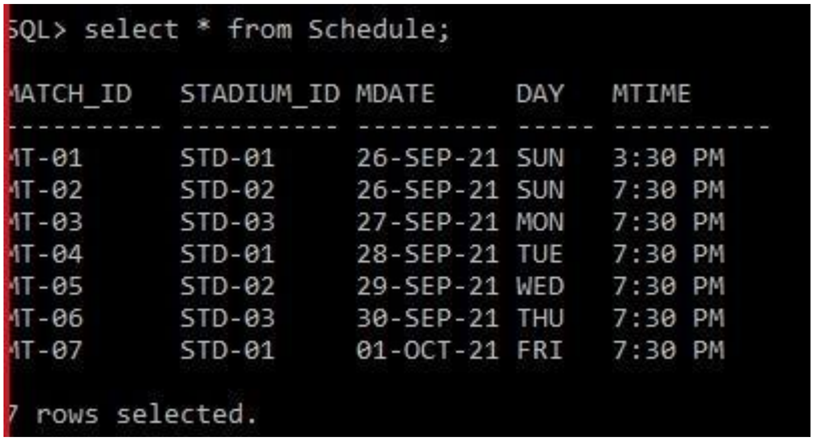
PRIMARY KEY: (match_id, stadium_id)
FOREIGN KEY: stadium_id, match_id



```
SQL> create table Schedule(
  2  match_id varchar2(10),
  3  stadium_id varchar2(10),
  4  mdate DATE,
  5  day varchar2(5) NOT NULL,
  6  mtime varchar2(10) NOT NULL,
  7  foreign key(match_id) references Matches(match_id),
  8  foreign key(stadium_id) references Stadium(stadium_id),
  9  primary key(match_id, stadium_id));

Table created.

SQL> desc Schedule;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MATCH_ID                                  NOT NULL VARCHAR2(10)
 STADIUM_ID                                NOT NULL VARCHAR2(10)
 MDATE                                              DATE
 DAY                                       NOT NULL VARCHAR2(5)
 MTIME                                     NOT NULL VARCHAR2(10)
```



```
SQL> insert into Schedule values('MT-01', 'STD-01', '26-SEP-21', 'SUN', '3:30 PM');

1 row created.

SQL> insert into Schedule values('MT-02', 'STD-02', '26-SEP-21', 'SUN', '7:30 PM');

1 row created.

SQL> insert into Schedule values('MT-03', 'STD-03', '27-SEP-21', 'MON', '7:30 PM');

1 row created.
```



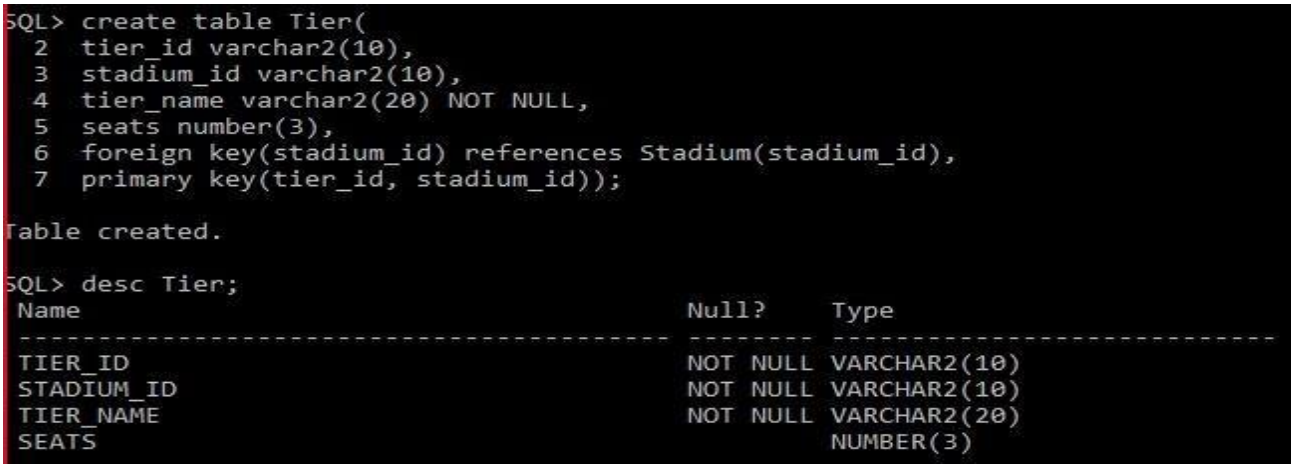
```
SQL> select * from Schedule;

MATCH_ID   STADIUM_ID MDATE     DAY   MTIME
---------- ---------- --------- ----- ----------
MT-01      STD-01     26-SEP-21 SUN   3:30 PM
MT-02      STD-02     26-SEP-21 SUN   7:30 PM
MT-03      STD-03     27-SEP-21 MON   7:30 PM
MT-04      STD-01     28-SEP-21 TUE   7:30 PM
MT-05      STD-02     29-SEP-21 WED   7:30 PM
MT-06      STD-03     30-SEP-21 THU   7:30 PM
MT-07      STD-01     01-OCT-21 FRI   7:30 PM

7 rows selected.
```

TIER TABLE:

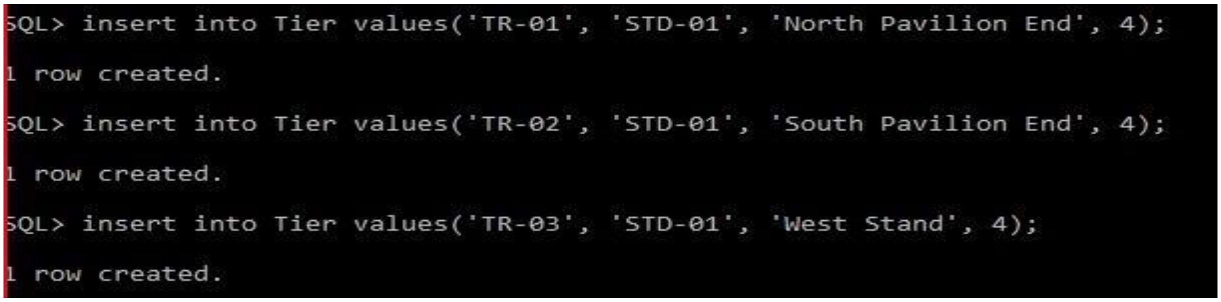
LIST OF ATTRIBUTES:

- ✓ tier_id VARCHAR2(10)
- ✓ stadium_id VARCHAR2(10)
- ✓ tier _name VARCHAR2(20)
- ✓ seats NUMBER(3)

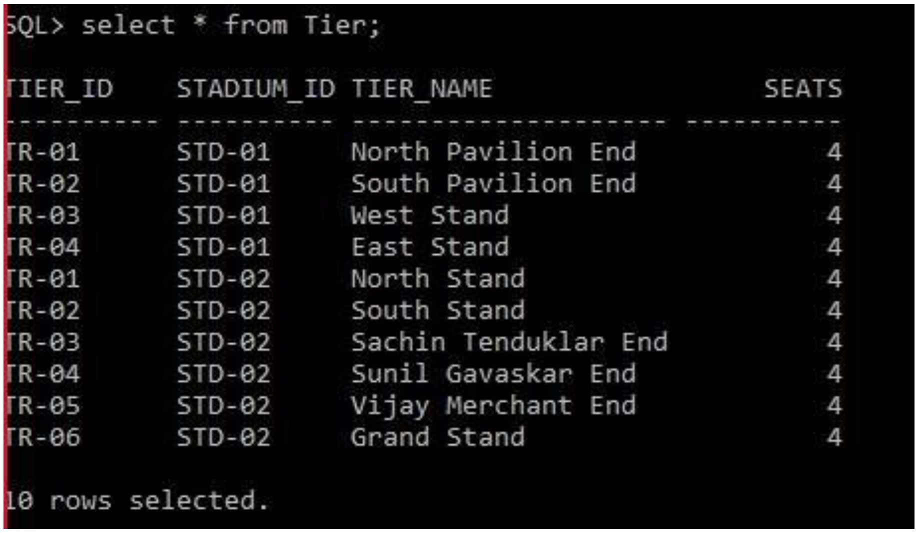
PRIMARY KEY: (tier_id, stadium_id)
FOREIGN KEY: stadium_id



```
SQL> create table Tier(
  2  tier_id varchar2(10),
  3  stadium_id varchar2(10),
  4  tier_name varchar2(20) NOT NULL,
  5  seats number(3),
  6  foreign key(stadium_id) references Stadium(stadium_id),
  7  primary key(tier_id, stadium_id));

Table created.

SQL> desc Tier;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TIER_ID                                   NOT NULL VARCHAR2(10)
 STADIUM_ID                                NOT NULL VARCHAR2(10)
 TIER_NAME                                 NOT NULL VARCHAR2(20)
 SEATS                                              NUMBER(3)
```



```
SQL> insert into Tier values('TR-01', 'STD-01', 'North Pavilion End', 4);

1 row created.

SQL> insert into Tier values('TR-02', 'STD-01', 'South Pavilion End', 4);

1 row created.

SQL> insert into Tier values('TR-03', 'STD-01', 'West Stand', 4);

1 row created.
```



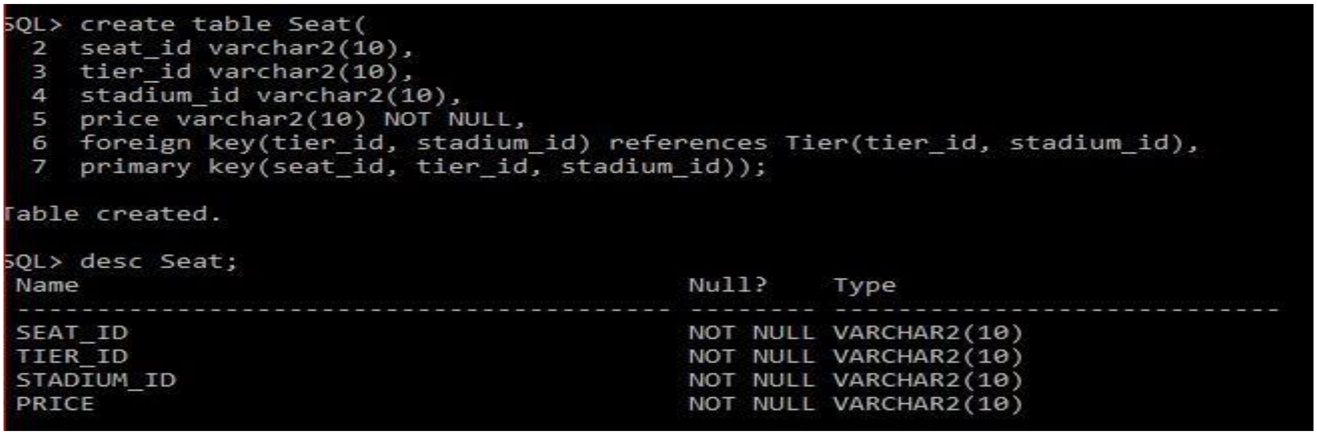
```
SQL> select * from Tier;

TIER_ID    STADIUM_ID TIER_NAME                 SEATS
---------- ---------- -------------------- ----------
TR-01      STD-01     North Pavilion End            4
TR-02      STD-01     South Pavilion End            4
TR-03      STD-01     West Stand                    4
TR-04      STD-01     East Stand                    4
TR-01      STD-02     North Stand                   4
TR-02      STD-02     South Stand                   4
TR-03      STD-02     Sachin Tenduklar End          4
TR-04      STD-02     Sunil Gavaskar End            4
TR-05      STD-02     Vijay Merchant End            4
TR-06      STD-02     Grand Stand                   4

10 rows selected.
```

SEAT TABLE:

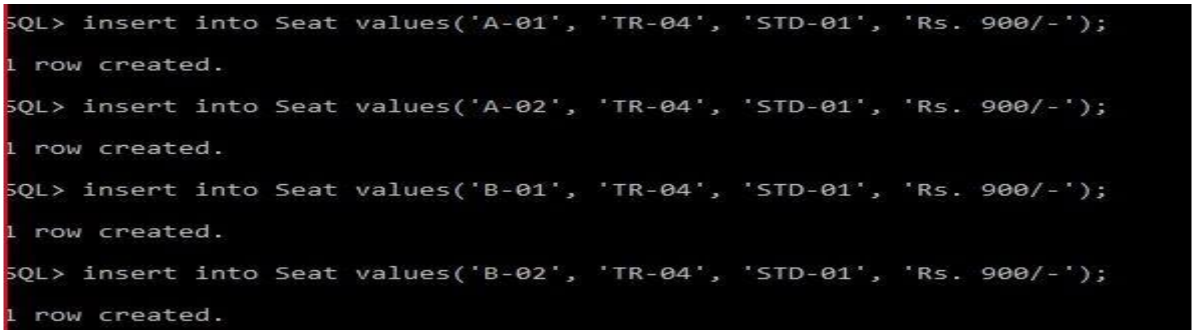
LIST OF ATTRIBUTES:

- ✓ seat_id VARCHAR2(10)
- ✓ tier_id VARCHAR2(10)
- ✓ stadium _id VARCHAR2(10)
- ✓ price VARCHAR2(10)

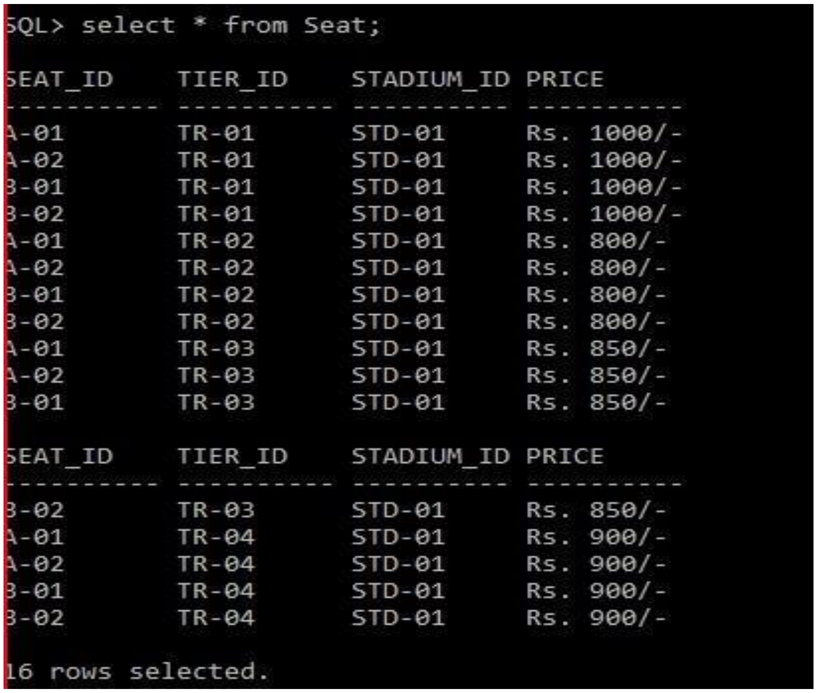
PRIMARY KEY: (seat_id**,** tier_id, stadium_id)
FOREIGN KEY: tier_id**,** stadium_id



```
SQL> create table Seat(
  2  seat_id varchar2(10),
  3  tier_id varchar2(10),
  4  stadium_id varchar2(10),
  5  price varchar2(10) NOT NULL,
  6  foreign key(tier_id, stadium_id) references Tier(tier_id, stadium_id),
  7  primary key(seat_id, tier_id, stadium_id));

Table created.

SQL> desc Seat;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SEAT_ID                                   NOT NULL VARCHAR2(10)
 TIER_ID                                   NOT NULL VARCHAR2(10)
 STADIUM_ID                                NOT NULL VARCHAR2(10)
 PRICE                                     NOT NULL VARCHAR2(10)
```



```
SQL> insert into Seat values('A-01', 'TR-04', 'STD-01', 'Rs. 900/-');

1 row created.

SQL> insert into Seat values('A-02', 'TR-04', 'STD-01', 'Rs. 900/-');

1 row created.

SQL> insert into Seat values('B-01', 'TR-04', 'STD-01', 'Rs. 900/-');

1 row created.

SQL> insert into Seat values('B-02', 'TR-04', 'STD-01', 'Rs. 900/-');

1 row created.
```



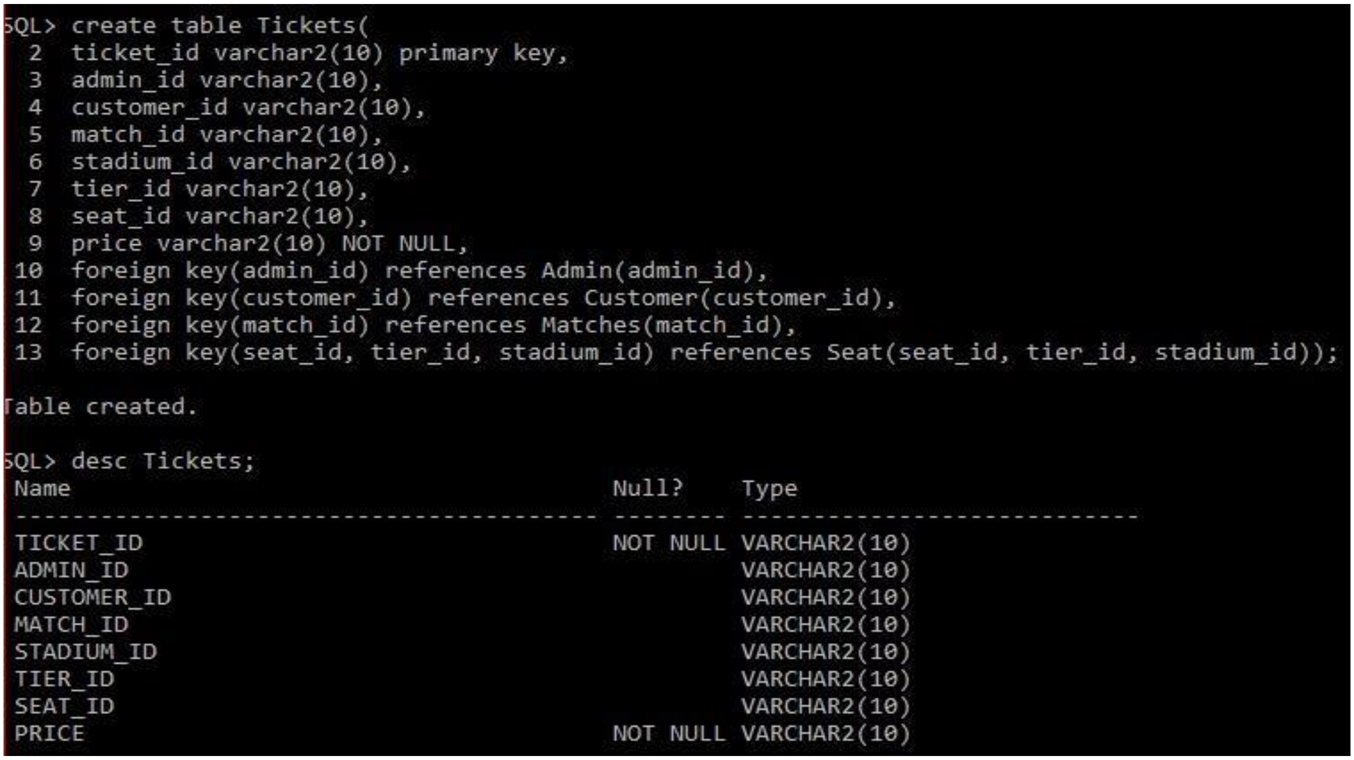
```
SQL> select * from Seat;

SEAT_ID    TIER_ID    STADIUM_ID PRICE
---------- ---------- ---------- ----------
A-01       TR-01      STD-01     Rs. 1000/-
A-02       TR-01      STD-01     Rs. 1000/-
B-01       TR-01      STD-01     Rs. 1000/-
B-02       TR-01      STD-01     Rs. 1000/-
A-01       TR-02      STD-01     Rs. 800/-
A-02       TR-02      STD-01     Rs. 800/-
B-01       TR-02      STD-01     Rs. 800/-
B-02       TR-02      STD-01     Rs. 800/-
A-01       TR-03      STD-01     Rs. 850/-
A-02       TR-03      STD-01     Rs. 850/-
B-01       TR-03      STD-01     Rs. 850/-

SEAT_ID    TIER_ID    STADIUM_ID PRICE
---------- ---------- ---------- ----------
B-02       TR-03      STD-01     Rs. 850/-
A-01       TR-04      STD-01     Rs. 900/-
A-02       TR-04      STD-01     Rs. 900/-
B-01       TR-04      STD-01     Rs. 900/-
B-02       TR-04      STD-01     Rs. 900/-

16 rows selected.
```

TICKET TABLE:

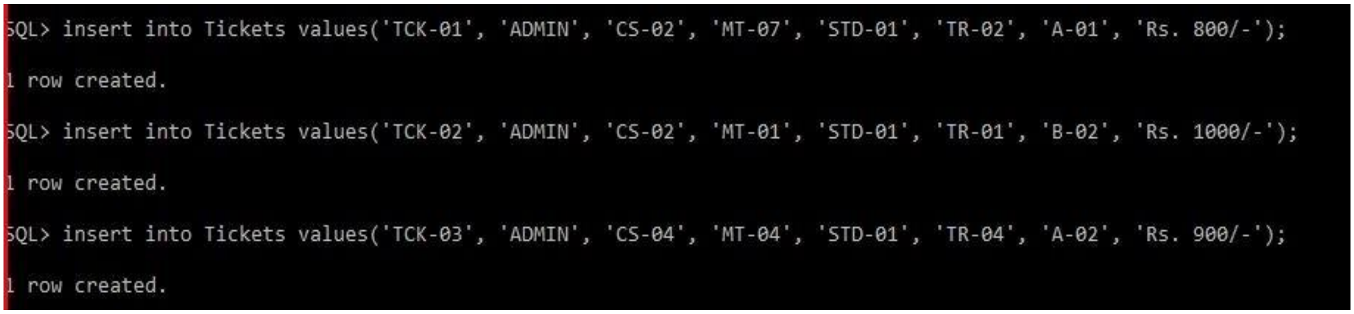
LIST OF ATTRIBUTES:

- ✓ ticket_id VARCHAR2(10)
- ✓ admin_id VARCHAR2(10)
- ✓ customer_id VARCHAR2(10)
- ✓ match_id VARCHAR2(10)
- ✓ stadium_id VARCHAR2(10)
- ✓ tier_id VARCHAR2(10)
- ✓ seat_id VARCHAR2(10)
- ✓ price VARCHAR2(10)

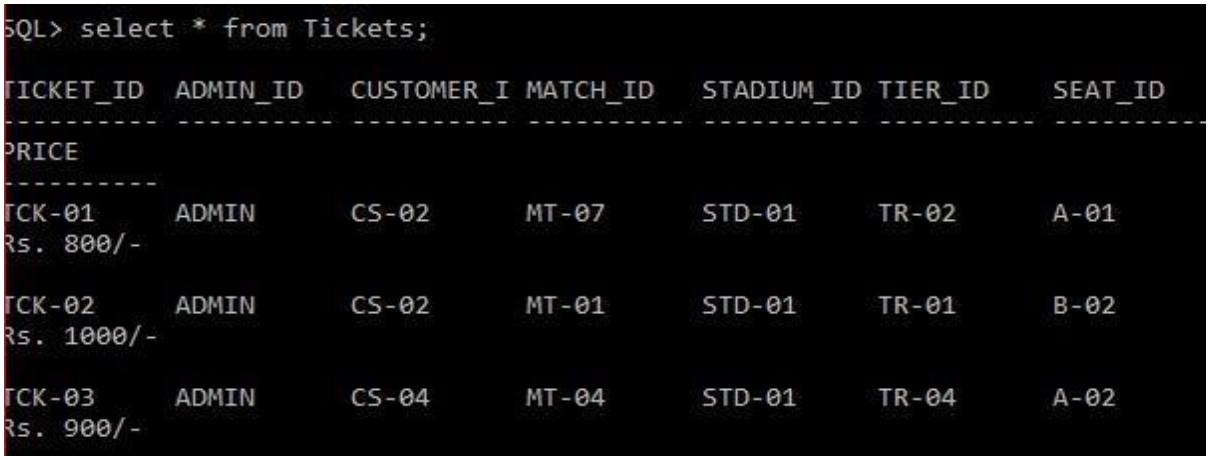
PRIMARY KEY: ticket_id
FOREIGN KEY: admin_id, customer_id, match_id, stadium_id, tier_id, seat_id



```
SQL> create table Tickets(
     ticket_id varchar2(10) primary key,
     admin_id varchar2(10),
     customer_id varchar2(10),
     match_id varchar2(10),
     stadium_id varchar2(10),
     tier_id varchar2(10),
     seat_id varchar2(10),
     price varchar2(10) NOT NULL,
     foreign key(admin_id) references Admin(admin_id),
     foreign key(customer_id) references Customer(customer_id),
     foreign key(match_id) references Matches(match_id),
     foreign key(seat_id, tier_id, stadium_id) references Seat(seat_id, tier_id, stadium_id));

Table created.

SQL> desc Tickets;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TICKET_ID                                 NOT NULL VARCHAR2(10)
 ADMIN_ID                                           VARCHAR2(10)
 CUSTOMER_ID                                        VARCHAR2(10)
 MATCH_ID                                           VARCHAR2(10)
 STADIUM_ID                                         VARCHAR2(10)
 TIER_ID                                            VARCHAR2(10)
 SEAT_ID                                            VARCHAR2(10)
 PRICE                                     NOT NULL VARCHAR2(10)
```



```
SQL> insert into Tickets values('TCK-01', 'ADMIN', 'CS-02', 'MT-07', 'STD-01', 'TR-02', 'A-01', 'Rs. 800/-');

1 row created.

SQL> insert into Tickets values('TCK-02', 'ADMIN', 'CS-02', 'MT-01', 'STD-01', 'TR-01', 'B-02', 'Rs. 1000/-');

1 row created.

SQL> insert into Tickets values('TCK-03', 'ADMIN', 'CS-04', 'MT-04', 'STD-01', 'TR-04', 'A-02', 'Rs. 900/-');

1 row created.
```



```
SQL> select * from Tickets;

TICKET_ID  ADMIN_ID   CUSTOMER_I MATCH_ID   STADIUM_ID TIER_ID    SEAT_ID
---------- ---------- ---------- ---------- ---------- ---------- ---------
PRICE
----------
TCK-01     ADMIN      CS-02      MT-07      STD-01     TR-02      A-01
Rs. 800/-

TCK-02     ADMIN      CS-02      MT-01      STD-01     TR-01      B-02
Rs. 1000/-

TCK-03     ADMIN      CS-04      MT-04      STD-01     TR-04      A-02
Rs. 900/-
```

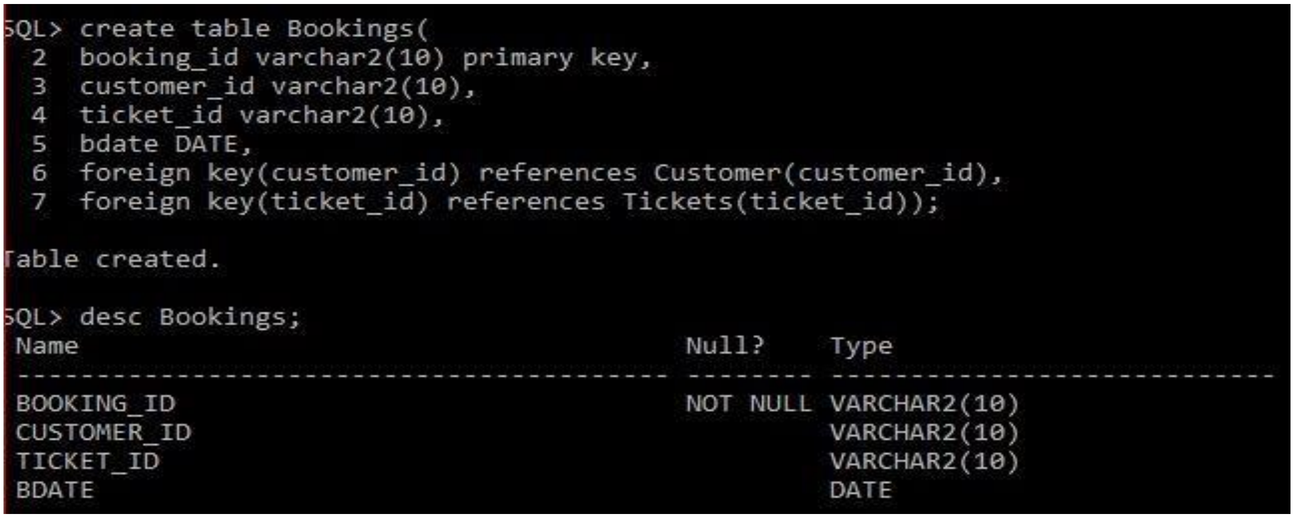
BOOKINGS TABLE:

LIST OF ATTRIBUTES:

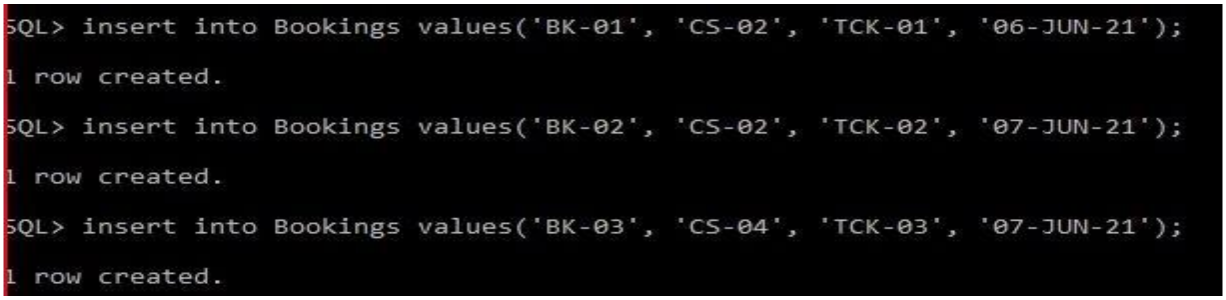
- ✓ booking_id VARCHAR2(10)
- ✓ customer_id VARCHAR2(10)
- ✓ ticket_id VARCHAR2(10)
- ✓ bdate DATE

PRIMARY KEY: booking_id

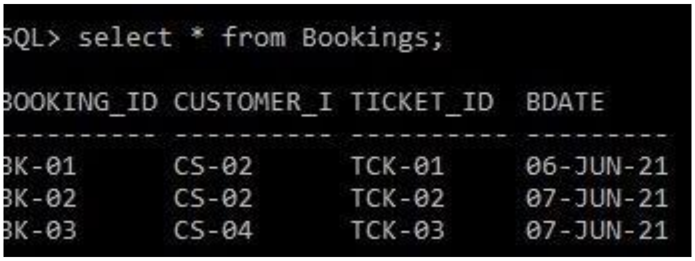
FOREIGN KEY: customer_id, ticket_id



```
SQL> create table Bookings(
  2  booking_id varchar2(10) primary key,
  3  customer_id varchar2(10),
  4  ticket_id varchar2(10),
  5  bdate DATE,
  6  foreign key(customer_id) references Customer(customer_id),
  7  foreign key(ticket_id) references Tickets(ticket_id));

Table created.

SQL> desc Bookings;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKING_ID                                NOT NULL VARCHAR2(10)
 CUSTOMER_ID                                        VARCHAR2(10)
 TICKET_ID                                          VARCHAR2(10)
 BDATE                                              DATE
```



```
SQL> insert into Bookings values('BK-01', 'CS-02', 'TCK-01', '06-JUN-21');

1 row created.

SQL> insert into Bookings values('BK-02', 'CS-02', 'TCK-02', '07-JUN-21');

1 row created.

SQL> insert into Bookings values('BK-03', 'CS-04', 'TCK-03', '07-JUN-21');

1 row created.
```



```
SQL> select * from Bookings;

BOOKING_ID CUSTOMER_I TICKET_ID  BDATE
---------- ---------- ---------- ---------
BK-01      CS-02      TCK-01     06-JUN-21
BK-02      CS-02      TCK-02     07-JUN-21
BK-03      CS-04      TCK-03     07-JUN-21
```

CODE:

CriccBook.java

```
 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.*;

public class CriccBook extends JFrame{
	private static final long serialVersionUID = 1L;

	private ImageIcon Bimg;
	private JLabel bg,title;
	private JButton register,login;
	private JMenuBar mb;
	private JMenu Menu;
	private JMenuItem About,Admin,Exit;

	CriccBook()
 {
	initializeComponents();
	settingBounds();
	settingFont();
	addComponents();
	registerListeners();
	setLookandFeel();

	setTitle("CriccBook");
	setSize(650,650);
	setLayout(null);
	setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	setVisible(true);
 }

 public void initializeComponents()
 {
	Bimg = new ImageIcon("F:/CriccBook/src/Stadium.jpg");

	bg = new JLabel(Bimg);
	title = new JLabel("Welcome To CriccBook");

	mb = new JMenuBar();
	Menu= new JMenu("Menu");
	About = new JMenuItem("About");
	Admin = new JMenuItem("Admin");
	Exit = new JMenuItem("Exit");

	Menu.add(About);Menu.add(Admin);Menu.add(Exit);
	mb.add(Menu);
```

```
        register = new JButton("Register");
        login = new JButton("Login");
}

public void settingBounds()
{
        bg.setSize(650,650);
        title.setBounds(220,150,300,30);
        register.setBounds(250,200,150,50);
        login.setBounds(250,300,150,50);
}

public void settingFont()
{
        title.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 22));
        title.setForeground(new Color(255,0,0));
}

public void addComponents()
{
        setJMenuBar(mb);

        add(bg);
        bg.add(title);
        bg.add(register);
        bg.add(login);
}

public void registerListeners()
        {

        Admin.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                new Login();
                                dispose();
            }
          });

        About.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                JOptionPane.showMessageDialog(CriccBook.this,"CriccBook is an Desktop
        application.\n"+ "It helps Customers to book Cricket match tickets in advance and enjoy the match.\n"+
        "Customer must be a registered user to do so.\nTHANK YOU!!","CRICCBOOK",
        JOptionPane.INFORMATION_MESSAGE);
            }
          });

        Exit.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                System.exit(1);
```

```
                }
            });

                register.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                new Register();
                dispose();
                }
            });

                login.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                new Login();
                                dispose();
                }
            });
          }

    public void setLookandFeel()
    {
          try
       {
          UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
          SwingUtilities.updateComponentTreeUI(CriccBook.this);
       }
       catch(Exception e2)
       {
          System.out.println("Look and Feel not set");
       }
    }

    public static void main(String args[]) {
       new CriccBook();
    }
}
```

Admin.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class Admin extends JFrame {
        private static final long serialVersionUID = 1L;

        private ImageIcon Bimg;
        private JLabel bg,title;
        private JMenuBar mb;
        private JMenu Account;
        private JMenuItem Schedule,Stadium,Logout;
        private JButton customers,bookings;

        Admin()
        {
                initializeComponents();
                settingBounds();
                settingFont();
                addComponents();
                registerListeners();


                setTitle("Admin");
                setSize(600,575);
                setLayout(null);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setVisible(true);
        }

        public void initializeComponents(){
                Bimg = new ImageIcon("F:/CriccBook/src/Admin.png");
                bg = new JLabel(Bimg);

                mb = new JMenuBar();
                Account = new JMenu("My Account");
                Schedule = new JMenuItem("Schedule");
                Stadium = new JMenuItem("Stadium");
                Logout = new JMenuItem("Logout");

                Account.add(Schedule);Account.add(Stadium);Account.add(Logout);
                mb.add(Account);

                title = new JLabel("Admin Panel");
        customers = new JButton("Customers");
        bookings = new JButton("Bookings");
    }

    public void settingBounds()
    {
        bg.setSize(600,575);
        title.setBounds(230,120,120,20);

        customers.setBounds(240,180,100,50);
        bookings.setBounds(240,260,100,50);
    }

    public void settingFont(){
        title.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 22));
        title.setForeground(new Color(255,0,0));
    }

    public void addComponents(){
        setJMenuBar(mb);

        add(bg);
        bg.add(title);
        bg.add(customers);
        bg.add(bookings);
    }
    public void registerListeners(){
        Schedule.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new Schedule();
       }  });
        Stadium.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new Stadium();
       }});

        customers.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new CTable();
       }});

        Logout.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new CriccBook();
                dispose();
       }
     });
    }
}
```

Customer.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Customer extends JFrame implements ActionListener {
        private static final long serialVersionUID = 1L;

        private ImageIcon Bimg;
        private JLabel bg,select,title;
        private JComboBox CB;
        private JMenuBar mb;
        private JMenu Account;
        private JMenuItem Details,History,Logout;

        private String ID,Pass;

        Customer(String ID,String Pass)
        {
                this.ID=ID;
                this.Pass=Pass;

                initializeComponents();
                settingBounds();
                settingFont();
                addComponents();
                registerListeners();


                setTitle("Customer");
                setSize(600,575);
                setLayout(null);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setVisible(true);
        }

        public void initializeComponents(){
                Bimg = new ImageIcon("F:/CriccBook/src/IPL.png");
                bg = new JLabel(Bimg);

                mb = new JMenuBar();
                Account = new JMenu("My Account");
                Details = new JMenuItem("Details");
                History = new JMenuItem("History");
                Logout = new JMenuItem("Logout");

                Account.add(Details);Account.add(History);Account.add(Logout);
                mb.add(Account);
                title = new JLabel("Start Your Bookings");
                select = new JLabel("Select Team:");

                String[] teams = {"CSK", "PBKS", "MI", "RCB", "DC", "KKR", "RR", "SRH"};
                CB = new JComboBox(teams);
        }

        public void settingBounds(){
                bg.setSize(600,575);

                title.setBounds(210,10,200,20);

                select.setBounds(200,180,120,20);
                CB.setBounds(320,180,70,25);
        }
        public void settingFont(){

                select.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 20));
                select.setForeground(new Color(255,0,0));

                title.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 22));
                title.setForeground(new Color(255,0,0));
        }
        public void addComponents(){
                add(bg);
                setJMenuBar(mb);
                bg.add(title);
                bg.add(select);
                bg.add(CB);
        }
        public void registerListeners(){
                Details.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent e){
                        String Details = "\nUsername: " + ID + "Password: " + Pass;
                            JOptionPane.showMessageDialog(Customer.this,Details,"Customer
Details", JOptionPane.INFORMATION_MESSAGE);
       }
     });
                Logout.addActionListener(new ActionListener(){
                    public void actionPerformed(ActionEvent e){
                        new CriccBook();
                        dispose();
       }});
                CB.addActionListener(this);
        }
        public void actionPerformed(ActionEvent AE){
                JComboBox comboBox = (JComboBox)AE.getSource();
           String Team = (String)comboBox.getSelectedItem();
                new TMatches(ID,Team);}
}
```

Register.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Register extends JFrame implements ActionListener
{
        private static final long serialVersionUID = 1L;

        private ImageIcon Bimg;
        private JLabel bg,title,c_id,c_name,c_pass;
        private JTextField cid,cname;
        private JPasswordField cpass;
        private JButton submit,reset;
        private JMenuBar mb;
        private JMenu Menu;
        private JMenuItem Home,Exit;

        private SQL_Connection SQL;

        Register()
        {
                initializeComponents();
                settingBounds();
                settingFont();
                addComponents();
                registerListeners();

                setTitle("Registration");
                setSize(600,400);
                setLayout(null);
                setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                setVisible(true);
        }

        public void initializeComponents()
        {
                Bimg=new ImageIcon("F:/CriccBook/src/Reg.jpg");
                bg=new JLabel(Bimg);

                mb = new JMenuBar();
                Menu= new JMenu("Menu");
                Home = new JMenuItem("Home");
                Exit = new JMenuItem("Exit");

                Menu.add(Home);Menu.add(Exit);
                mb.add(Menu);

                title =new JLabel("Customer Registration");
                c_id = new JLabel("Customer ID:");
                c_name = new JLabel("Full Name:");
                c_pass = new JLabel("Password:");

                cid=new JTextField();
                cname=new JTextField();

                cpass=new JPasswordField();
                cpass.setEchoChar('*');

                submit = new JButton("SUBMIT");
                reset = new JButton("RESET");
        }
        public void settingBounds(){
                bg.setSize(600,400);

                title.setBounds(165,30,300,30);
                c_id.setBounds(160,80,110,30);
                c_name.setBounds(180,130,100,30);
                c_pass.setBounds(180,180,100,30);

                cid.setBounds(280,80,100,30);
                cname.setBounds(280,130,100,30);
                cpass.setBounds(280,180,100,30);

                submit.setBounds(170,230,100,30);
                reset.setBounds(290,230,100,30);
        }
        public void settingFont(){
                title.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 22));
                title.setForeground(new Color(255,0,0));
                c_id.setFont(new Font("", Font.BOLD, 16));
                c_id.setForeground(new Color(0,255,0));
                c_name.setFont(new Font("", Font.BOLD, 16));
                c_name.setForeground(new Color(0,255,0));
                c_pass.setFont(new Font("", Font.BOLD, 16));
                c_pass.setForeground(new Color(0,255,0));
        }
        public void addComponents(){
                setJMenuBar(mb);
                add(bg);
                bg.add(title);
                bg.add(c_id);
                bg.add(c_name);
                bg.add(c_pass);
                bg.add(cid);
                bg.add(cname);
                bg.add(cpass);
                bg.add(submit);
                bg.add(reset);
```

```
	}

	public void registerListeners(){
		Home.addActionListener(new ActionListener(){
			public void actionPerformed(ActionEvent e){
				new CriccBook();
				dispose();
	    }
	   });
		Exit.addActionListener(new ActionListener(){
			public void actionPerformed(ActionEvent e){
				System.exit(1);
	    }
	   });
		submit.addActionListener(this);
		reset.addActionListener(new ActionListener(){
			public void actionPerformed(ActionEvent e){
		cid.setText("");
		cname.setText("");
		cpass.setText("");
	    }
	   });
	}
	public void actionPerformed(ActionEvent AE){
		SQL = new SQL_Connection();
		String ID = cid.getText(),Name = cname.getText();
		char[] pw = cpass.getPassword();
		String Password = String.valueOf(pw);
		String Details = "Customer ID: " + ID +"\nCustomer Name: "+ Name;
		if(SQL.CID_Validation(ID) == 1){
			if(SQL.ValidPassword(Password) == 1){
				SQL.registration(ID, Name, Password);
				JOptionPane.showMessageDialog(this, Details, "SUCCESSFUL
REGISTRATION", JOptionPane.INFORMATION_MESSAGE);

				new Customer(ID,Password);
				dispose();
			}
			else{
		JOptionPane.showMessageDialog(this, "Password should be atleast 8 characters!!",
"WARNING", JOptionPane.WARNING_MESSAGE);
	    }
		}
		else{
			JOptionPane.showMessageDialog(this, "Customer ID already exists!!", "WARNING",
JOptionPane.WARNING_MESSAGE);
		}}

}
```

Login.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Login extends JFrame implements ActionListener
{
        private static final long serialVersionUID = 1L;

        private ImageIcon Bimg;
        private JLabel bg,title,c_id,c_pass;
        private JTextField cid;
        private JPasswordField cpass;
        private JButton submit,reset;
        private JMenuBar mb;
        private JMenu Menu;
        private JMenuItem Home,Exit;

        private SQL_Connection SQL;

        Login()
        {
                initializeComponents();
                settingBounds();
                settingFont();
                addComponents();
                registerListeners();

                setTitle("Login");
                setSize(600,400);
                setLayout(null);
                setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                setVisible(true);

        }

        public void initializeComponents()
        {
                Bimg=new ImageIcon("F:/CriccBook/src/Reg.jpg");
                bg=new JLabel(Bimg);

                mb = new JMenuBar();
                Menu= new JMenu("Menu");
                Home = new JMenuItem("Home");
                Exit = new JMenuItem("Exit");

                Menu.add(Home);Menu.add(Exit);
                mb.add(Menu);
                title =new JLabel("Sign-In");
                c_id = new JLabel("Username:");
                c_pass = new JLabel("Password:");

                cid=new JTextField();

                cpass=new JPasswordField();
                cpass.setEchoChar('*');

                submit = new JButton("SUBMIT");
                reset = new JButton("RESET");
        }

        public void settingBounds()
        {
                bg.setSize(600,400);

                title.setBounds(230,30,300,30);
                c_id.setBounds(180,80,110,30);
                c_pass.setBounds(180,150,100,30);

                cid.setBounds(280,80,100,30);
                cpass.setBounds(280,150,100,30);

                submit.setBounds(170,230,100,30);
                reset.setBounds(290,230,100,30);
        }

        public void settingFont()
        {
                title.setFont(new Font("Times New Roman", Font.BOLD|Font.ITALIC, 22));
                title.setForeground(new Color(255,0,0));

                c_id.setFont(new Font("Times New Roman", Font.BOLD, 18));
                c_id.setForeground(new Color(0,255,0));

                c_pass.setFont(new Font("Times New Roman", Font.BOLD, 18));
                c_pass.setForeground(new Color(0,255,0));
        }

        public void registerListeners()
        {
                Home.addActionListener(new ActionListener(){
                        public void actionPerformed(ActionEvent e){
                                new CriccBook();
                                dispose();
              }
           });
```

```
        Exit.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                System.exit(1);
       }
     });

        submit.addActionListener(this);
        reset.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
        cid.setText("");
        cpass.setText("");
       }
     });
    }
    public void addComponents(){
        setJMenuBar(mb);
        add(bg);
        bg.add(title);
        bg.add(c_id);
        bg.add(c_pass);
        bg.add(cid);
        bg.add(cpass);
        bg.add(submit);
        bg.add(reset);
    }
    public void actionPerformed(ActionEvent AE){
        SQL=new SQL_Connection();
        String ID=cid.getText();
        char[] pw=cpass.getPassword();
        String Password=String.valueOf(pw);

        if(SQL.login(ID,Password)){
            JOptionPane.showMessageDialog(this,"LOGIN SUCCESSFUL","CriccBook",
JOptionPane.INFORMATION_MESSAGE);

            if(ID.equals("ADMIN")){
                new Admin();
                dispose();
            }
            else{
                new Customer(ID,Password);
                dispose();
            }
        }
        else{
            JOptionPane.showMessageDialog(this,"Invalid ID (or) Password!!","WARNING",
JOptionPane.WARNING_MESSAGE);
        }
    }
}
```

SQL_Connection.java

```
import java.sql.*;

public class SQL_Connection {

    Connection conn = null;
    Statement stmt=null;
    ResultSet rs;

    SQL_Connection()
    {
        try
        {
            //Loading Driver
            Class.forName("com.mysql.jdbc.Driver");
            String DB_URL = "jdbc:mysql://localhost:3306/CriccBook";

            // Open a connection
            conn = DriverManager.getConnection(DB_URL,"root","root");
            stmt = conn.createStatement();

        }
        catch(Exception e){ System.out.println(e);}
    }

    public int CID_Validation(String cid)
    {
        try
        {
            rs = stmt.executeQuery("select cid from Customer;");
            while(rs.next()){
                if(rs.getString(1).equals(cid))
                {
                    return 0;
                }
            }
        }
        catch(Exception e){
            return 0;
        }

        return 1;
    }

    public int ValidPassword(String cpass)
    {
        if(cpass.length()>=8)return 1;
        else return 0;
    }
    public void registration(String cid,String cname,String cpass)
    {
        try
        {
            String reg="insert into Customer values('"+cid+"','"+cname+"','"+cpass+"');";
            stmt.executeUpdate(reg);
        }
        catch(Exception e){ System.out.println(e);}
    }

    public boolean login(String id,String pass)
    {
        int flag=0;

        String log="";
        if(id.equals("ADMIN"))
        {
            log="select password from Admin where admin_id='"+id+"';";
        }
        else
        {
            log="select password from Customer where cid='"+id+"';";
        }

        try
        {
            rs = stmt.executeQuery(log);
            while(rs.next()){
                if(rs.getString(1).equals(pass)){
                    flag=1;
                }
            }
        }
        catch(Exception e){ System.out.println(e);}

        if(flag==1) return true;
        else return false;
    }
}
```

<u>Schedule.java</u>

```
import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class Schedule extends JFrame {

        private static final long serialVersionUID = 1L;

        private Connection conn = null;
        private Statement stmt=null;
        private ResultSet rs;

        private JTable ct;
        private JLabel label1, label2;

        Schedule()
        {
                try
                {
                        //Loading Driver
                        Class.forName("com.mysql.jdbc.Driver");
                        String DB_URL = "jdbc:mysql://localhost:3306/CriccBook";

                        // Open a connection
                        conn = DriverManager.getConnection(DB_URL,"root","root");
                        stmt = conn.createStatement();

                        Object[][] rows = new Object[8][6];

                       rs = stmt.executeQuery("select M.team1,M.team2,St.sname,S.mdate,S.day,S.mtime from "
                        + "Matches M,Stadium St,Schedule S where S.mid=M.mid and S.sid=St.sid;");

                        int i=0;
                        while(rs.next()){
                                rows[i][0] = rs.getString(1);rows[i][1] = rs.getString(2);
                                rows[i][2] = rs.getString(3);rows[i][3] = rs.getString(4);
                                rows[i][4] = rs.getString(5);rows[i][5] = rs.getString(6);
                                i++;
                        }

                        label1 = new JLabel("IPL Matches Schedule");

                        String[] cols = new String[]{"Team1","Team2","Stadium","Date","Day","Time"};

                        ct = new JTable(rows,cols);

                        JScrollPane scrollP = new JScrollPane(ct);
```

```
            DefaultTableCellRenderer tableRenderer = new DefaultTableCellRenderer();
            tableRenderer.setHorizontalAlignment(JLabel.CENTER);
            ct.setDefaultRenderer(Object.class, tableRenderer);

            scrollP.setBorder(BorderFactory.createEmptyBorder());
            scrollP.setPreferredSize(new Dimension(490, 180));

            label2 = new JLabel("Rows : " + ct.getRowCount() + ", Columns : "+
ct.getColumnCount() );

            add(label1);
            add(scrollP);
            add(label2);

            setTitle("Schedule");
            setSize(530,280);
            setLayout(new FlowLayout());
            setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);
        }
        catch(Exception e){ System.out.println(e);}
    }
}
```

Stadium.java

```
import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class Stadium extends JFrame {

    private static final long serialVersionUID = 1L;

    private Connection conn = null;
    private Statement stmt=null;
    private ResultSet rs;

    private JTable ct;
    private JLabel label1, label2;

    Stadium()
    {
        try
        {
            //Loading Driver
            Class.forName("com.mysql.jdbc.Driver");
            String DB_URL = "jdbc:mysql://localhost:3306/CriccBook";

            // Open a connection
            conn = DriverManager.getConnection(DB_URL,"root","root");
            stmt = conn.createStatement();

            Object[][] rows = new Object[3][4];

            rs = stmt.executeQuery("select * from Stadium;");

            int i=0;
            while(rs.next()){
                rows[i][0] = rs.getString(1);
                rows[i][1] = rs.getString(2);
                rows[i][2] = rs.getString(3);
                rows[i][3] = rs.getString(4);
                i++;
            }

            label1 = new JLabel("List Of Stadiums");

            String[] cols = new String[]{"Stadium ID","Stadium Name","Location","Tiers"};

            ct = new JTable(rows,cols);

            JScrollPane scrollP = new JScrollPane(ct);
```

```
                        DefaultTableCellRenderer tableRenderer = new DefaultTableCellRenderer();
                        tableRenderer.setHorizontalAlignment(JLabel.CENTER);
                        ct.setDefaultRenderer(Object.class, tableRenderer);


                        scrollP.setBorder(BorderFactory.createEmptyBorder());
                        scrollP.setPreferredSize(new Dimension(420, 100));

                        label2 = new JLabel("Rows : " + ct.getRowCount() + ", Columns : "+
ct.getColumnCount() );

                        add(label1);
                        add(scrollP);
                        add(label2);


                        setTitle("Stadiums");
                        setSize(530,200);
                        setLayout(new FlowLayout());
                        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                        setVisible(true);
                }
                catch(Exception e){ System.out.println(e);}
        }
}
```

TMatches.java

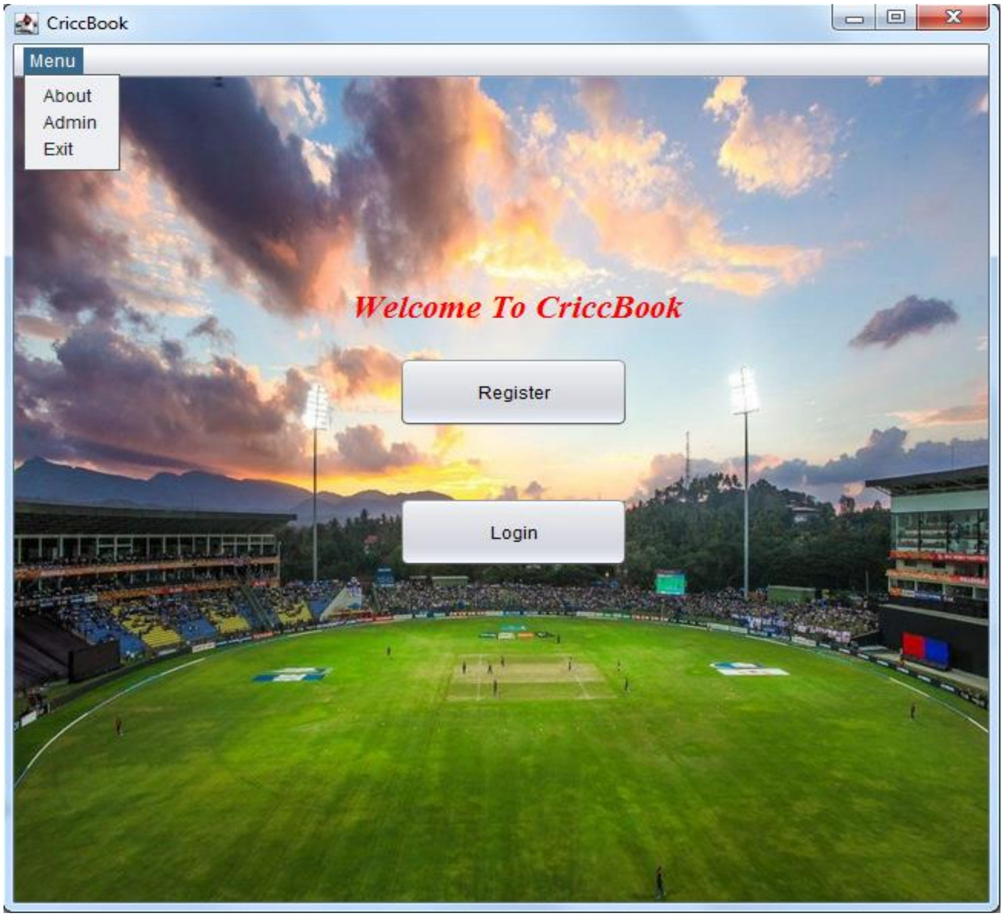
```
import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class TMatches extends JFrame {

        private static final long serialVersionUID = 1L;

        private Connection conn = null;
        private Statement stmt=null;
        private ResultSet rs;

        private JTable ct;
        private JLabel label1, label2;
        private JTextField match_no;
        private JButton next;
        private int n=0;

        private String CID;
        private String[] MID;
        private String[] SID;

        TMatches(String CID,String Team)
        {
                try
                {
                        this.CID = CID;

                        Class.forName("com.mysql.jdbc.Driver");
                        String DB_URL = "jdbc:mysql://localhost:3306/CriccBook";

                        // Open a connection
                        conn = DriverManager.getConnection(DB_URL,"root","root");
                        stmt = conn.createStatement();


                        rs = stmt.executeQuery("select mid from Matches where team1='"+Team+"' or
        team2='"+Team+"';");
                        int i=0;
                        while(rs.next()){
                                n++;
                        }

                        MID = new String[n];
                        SID = new String[n];

                        rs = stmt.executeQuery("select S.mid,S.sid from Schedule S,Matches M where "
                        + "M.mid in (select mid from Matches where team1='"+Team+"' || team2='"+Team+"')
and S.mid=M.mid;");
                         i=0;
                        while(rs.next()){
                                MID[i]=rs.getString(1);
                                SID[i]=rs.getString(2);
                                i++;
                        }

                        Object[][] rows = new Object[n][6];

                        rs = stmt.executeQuery("select M.team1,M.team2,St.sname,S.mdate,S.day,S.mtime from
Schedule S,Matches M,Stadium St where "+ "M.mid in (select mid from Matches where team1='"+Team+"' ||
team2='"+Team+"') and S.mid=M.mid and St.sid=S.sid order by S.mdate;");

                        i=0;
                        while(rs.next()){
                                rows[i][0] = rs.getString(1);rows[i][1] = rs.getString(2);
                                rows[i][2] = rs.getString(3);rows[i][3] = rs.getString(4);
                                rows[i][4] = rs.getString(5);rows[i][5] = rs.getString(6);
                                i++;
                        }

                        label1 = new JLabel(Team+" Team Schedule");

                        String[] cols = new String[]{"Team1","Team2","Stadium","Date","Day","Time"};

                        ct = new JTable(rows,cols);

                        JScrollPane scrollP = new JScrollPane(ct);

                        DefaultTableCellRenderer tableRenderer = new DefaultTableCellRenderer();
                        tableRenderer.setHorizontalAlignment(JLabel.CENTER);
                        ct.setDefaultRenderer(Object.class, tableRenderer);


                        scrollP.setBorder(BorderFactory.createEmptyBorder());
                        scrollP.setPreferredSize(new Dimension(490, 100));

                        label2 = new JLabel("Match No:");
                        match_no = new JTextField(5);
                        next = new JButton("Next");

                        add(label1);
                        add(scrollP);
                        add(label2);
                        add(match_no);
                        add(next);

                        registerListeners();
```

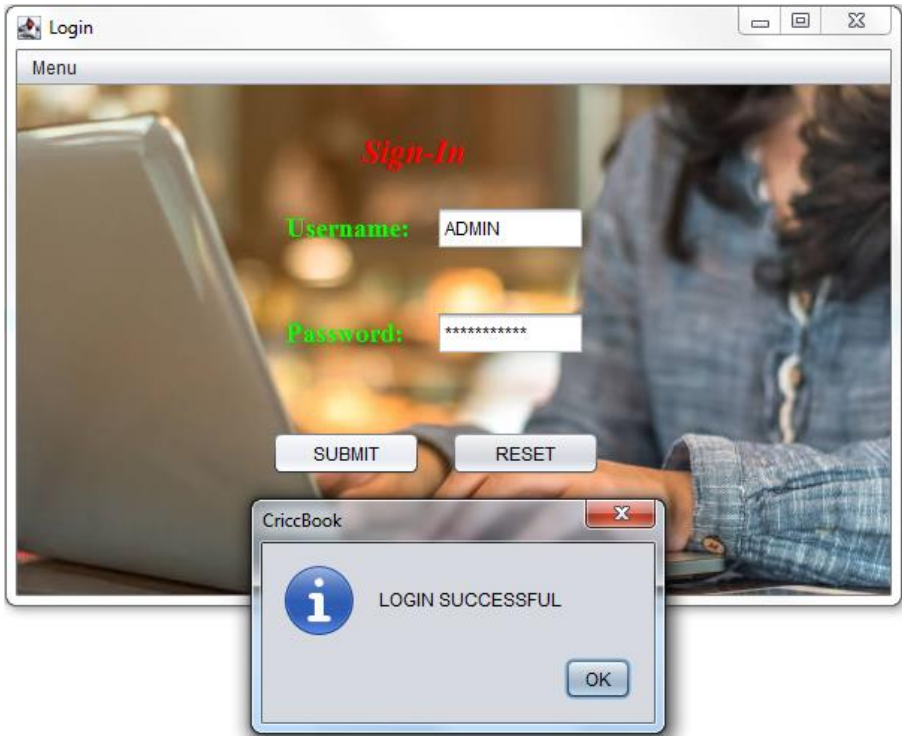
```
            setTitle("Schedule");
            setSize(530,200);
            setLayout(new FlowLayout());
            setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);
        }
        catch(Exception e){ System.out.println(e);}
    }

    public void registerListeners()
    {
        next.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent ae)
            {
                try
                {
                    int m_no = Integer.parseInt(match_no.getText());
                    if(m_no <=0 || m_no>n){
                        JOptionPane.showMessageDialog(TMatches.this, "Invalid Match
Number!!", "WARNING", JOptionPane.WARNING_MESSAGE);
                    }
                    else
                    {
                        new Tiers(CID,MID[m_no-1],SID[m_no-1]);
                        dispose();
                    }
                }
                catch(Exception E){
                    JOptionPane.showMessageDialog(TMatches.this, "Invalid Match
Number!!", "WARNING", JOptionPane.WARNING_MESSAGE);
                    System.out.println(E);
                }
       }
     });
    }
}
```

Tiers.java

```
import java.awt.*;
import java.sql.*;
import javax.swing.*;

public class Tiers extends JFrame {

    private static final long serialVersionUID = 1L;

    private Connection conn = null;
    private Statement stmt=null;
    private ResultSet rs;

    private JTable ct;
    private JLabel label1,tid,seats;
    private JTextField tier,seat;
    private JButton book;
    private String CID,MID,SID;

    int n;

    Tiers(String CID,String MID,String SID)
    {
        try
        {
            this.CID=CID;this.MID=MID;this.SID=SID;

            //Loading Driver
            Class.forName("com.mysql.jdbc.Driver");
            String DB_URL = "jdbc:mysql://localhost:3306/CriccBook";

            // Open a connection
            conn = DriverManager.getConnection(DB_URL,"root","root");
            stmt = conn.createStatement();

            rs = stmt.executeQuery("select tiers from Stadium where sid='"+SID+"';");
            while(rs.next()){
                n=rs.getInt(1);
            }

            Object[][] rows = new Object[n][4];

            rs = stmt.executeQuery("select tid,tname,seats from Tier where sid='"+SID+"';");
            int i=0;
            while(rs.next()){
                rows[i][0] = rs.getString(1);rows[i][1] = rs.getString(2);
                rows[i][2] = rs.getString(3);
                i++;
            }

            for(int p=0;p<n;p++){
                rs = stmt.executeQuery("select price from Seat where seat_id='A_01' and
tid='"+rows[p][0]+"' and sid='"+SID+"';");
                while(rs.next()){
                    rows[p][3]=rs.getString(1);
                }
            }

            label1 = new JLabel("Stadium Tier Details");
            tid = new JLabel("Tier ID:");
            seats = new JLabel("No.of Seats:");

            tier = new JTextField(5);
            seat = new JTextField(5);

            book = new JButton("Book Now");

            String[] cols = new String[]{"Tier ID","Tier Name","Available Seats","Price"};

            ct = new JTable(rows,cols);

            JScrollPane scrollP = new JScrollPane(ct);

            DefaultTableCellRenderer tableRenderer = new DefaultTableCellRenderer();
            tableRenderer.setHorizontalAlignment(JLabel.CENTER);
            ct.setDefaultRenderer(Object.class, tableRenderer);


            scrollP.setBorder(BorderFactory.createEmptyBorder());
            scrollP.setPreferredSize(new Dimension(480, 140));

            add(label1);
            add(scrollP);
            add(tid);add(tier);
            add(seats);add(seat);
            add(book);

            registerListeners();

            setTitle("Tiers");
            setSize(530,280);
            setLayout(new FlowLayout());
            setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
            setVisible(true);
        }
        catch(Exception e){ System.out.println(e);}
    }
    public void registerListeners()
    {
        book.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent ae){
                try{
                    int nos = Integer.parseInt(seat.getText());
                    String ts = tier.getText();
                    int t = ts.charAt(4) - '0';
                    if(nos <=0 || nos > 10){
                        JOptionPane.showMessageDialog(Tiers.this, "Invalid Seats!!",
"WARNING", JOptionPane.WARNING_MESSAGE);
                    }
                    if(t <=0 || t > n)
                    {
                        JOptionPane.showMessageDialog(Tiers.this, "Invalid Tier ID!!",
"WARNING", JOptionPane.WARNING_MESSAGE);
                    }
                    else{
                        JOptionPane.showMessageDialog(Tiers.this,"Booking
Successful!!", "CriccBook", JOptionPane.INFORMATION_MESSAGE);
                        dispose();
                    }
                }
                catch(Exception E){
                    JOptionPane.showMessageDialog(Tiers.this, "Invalid Seats!!",
"WARNING", JOptionPane.WARNING_MESSAGE);
                    System.out.println(E);
                }
       }
     });
    }
}
```

SCREENSHOTS:

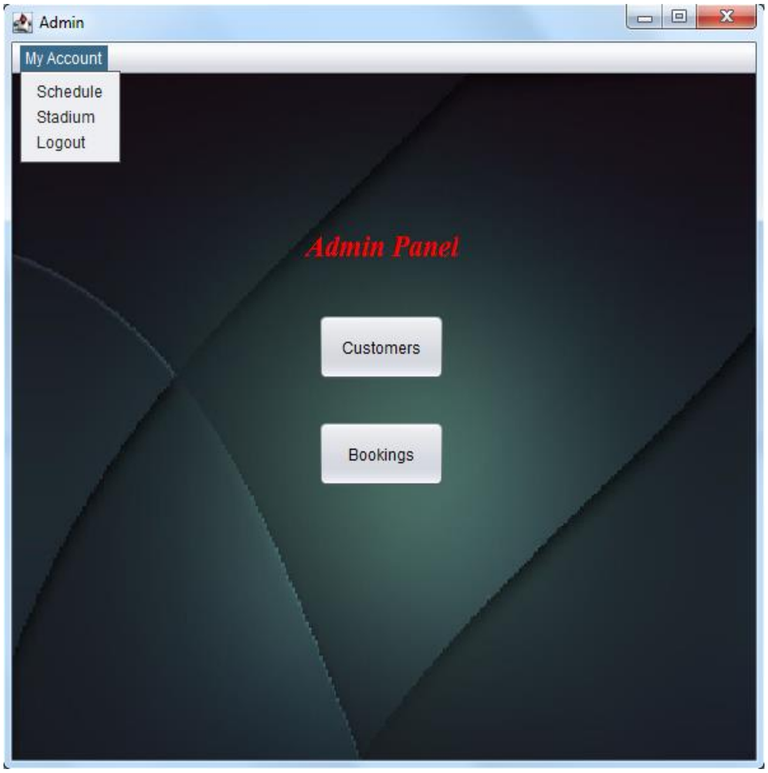


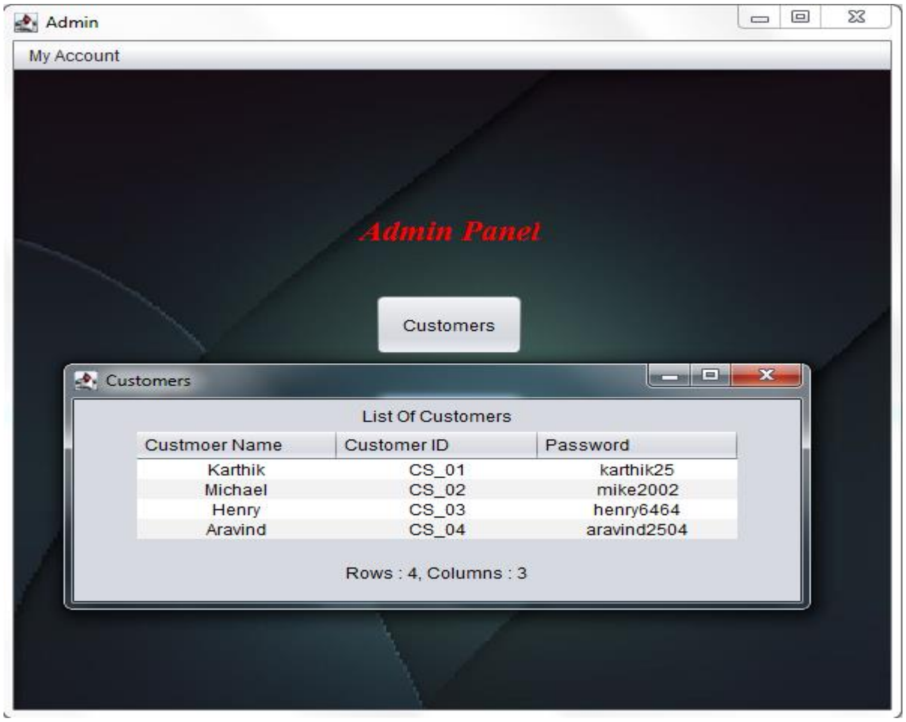


Username: ADMIN

Password: ***********

LOGIN SUCCESSFUL



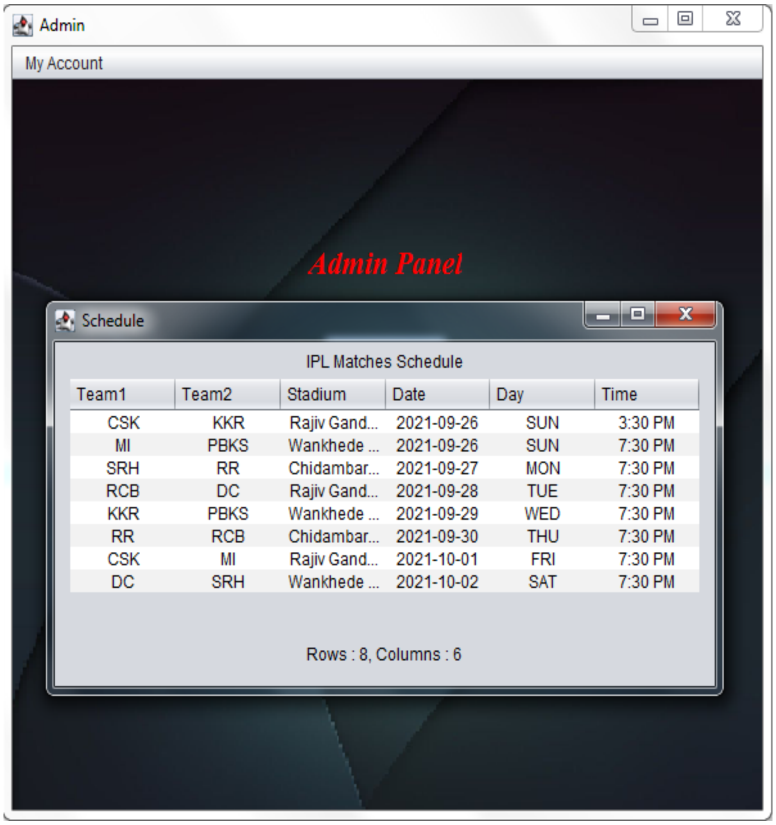


Customers

List Of Customers

Custmoer Name	Customer ID	Password
Karthik	CS_01	karthik25
Michael	CS_02	mike2002
Henry	CS_03	henry6464
Aravind	CS_04	aravind2504

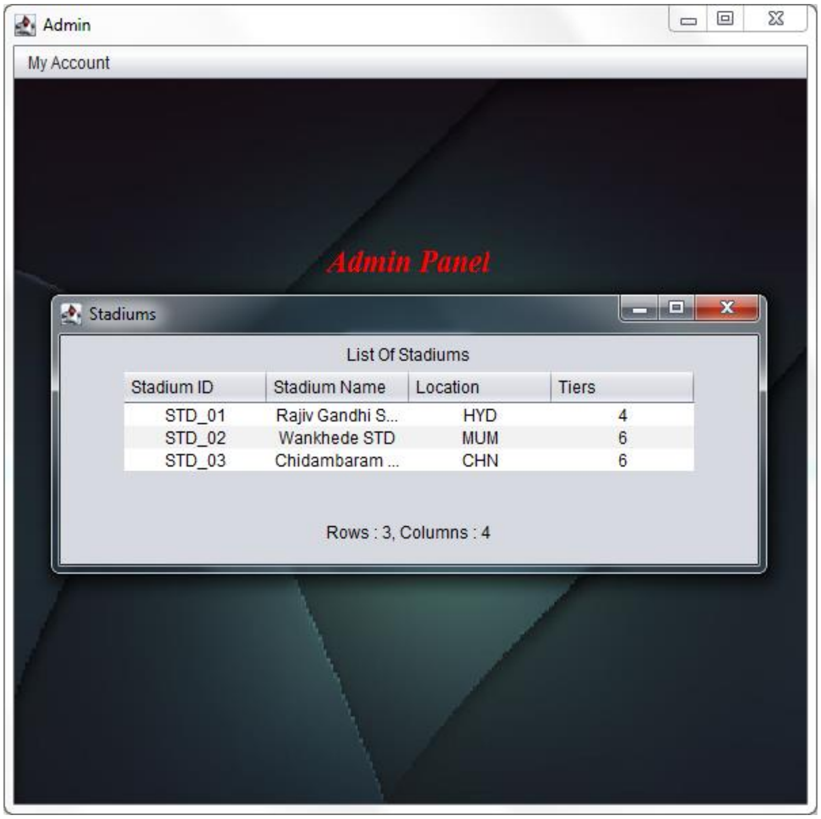
Rows : 4, Columns : 3



IPL Matches Schedule

Team1	Team2	Stadium	Date	Day	Time
CSK	KKR	Rajiv Gand...	2021-09-26	SUN	3:30 PM
MI	PBKS	Wankhede ...	2021-09-26	SUN	7:30 PM
SRH	RR	Chidambar...	2021-09-27	MON	7:30 PM
RCB	DC	Rajiv Gand...	2021-09-28	TUE	7:30 PM
KKR	PBKS	Wankhede ...	2021-09-29	WED	7:30 PM
RR	RCB	Chidambar...	2021-09-30	THU	7:30 PM
CSK	MI	Rajiv Gand...	2021-10-01	FRI	7:30 PM
DC	SRH	Wankhede ...	2021-10-02	SAT	7:30 PM

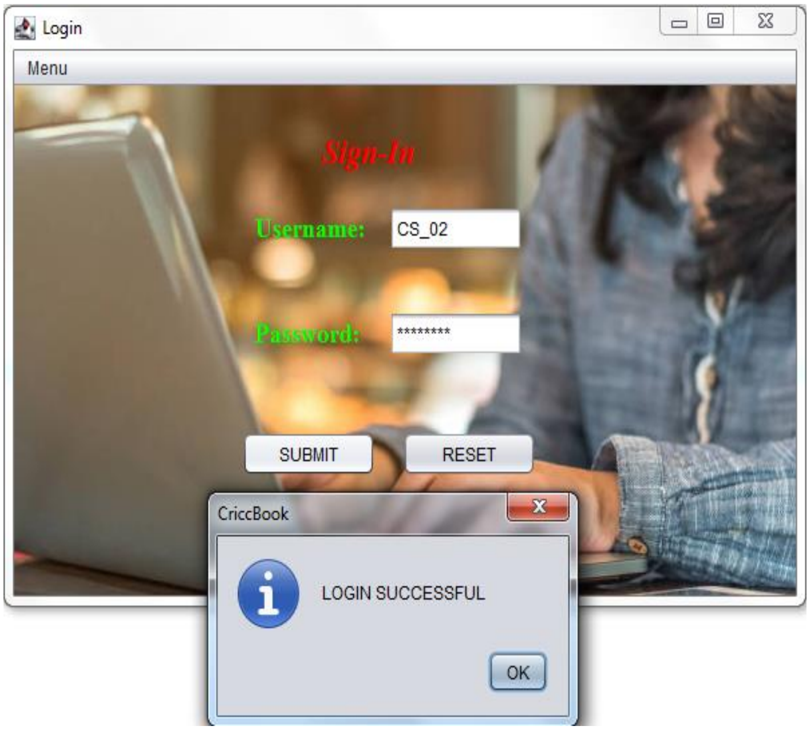
Rows : 8, Columns : 6



List Of Stadiums

Stadium ID	Stadium Name	Location	Tiers
STD_01	Rajiv Gandhi S...	HYD	4
STD_02	Wankhede STD	MUM	6
STD_03	Chidambaram ...	CHN	6

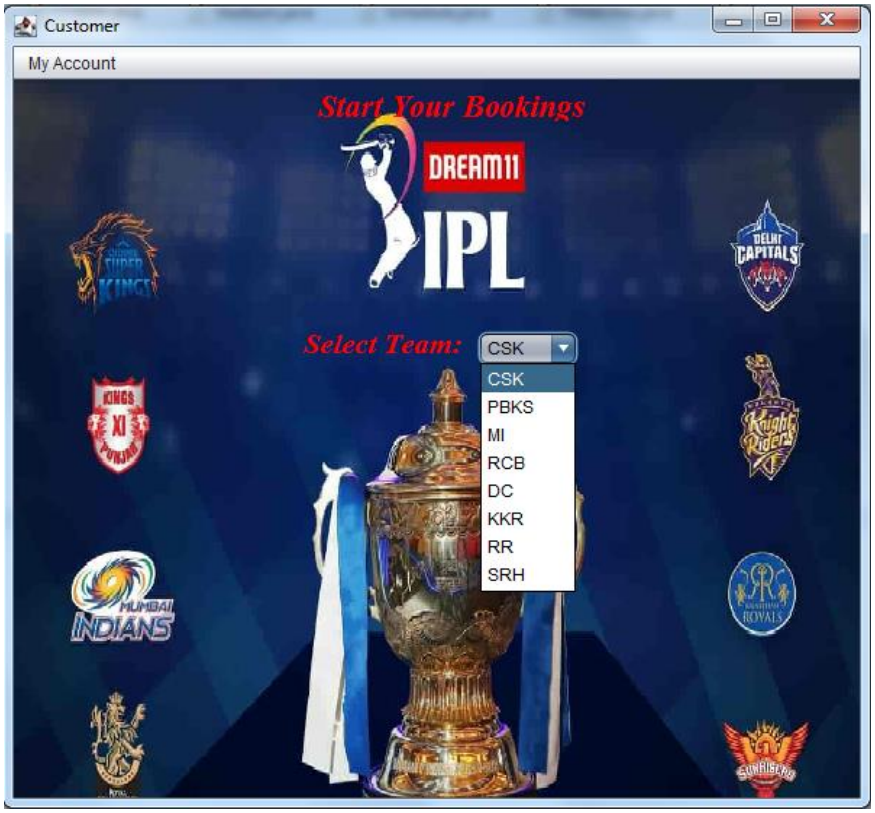
Rows : 3, Columns : 4



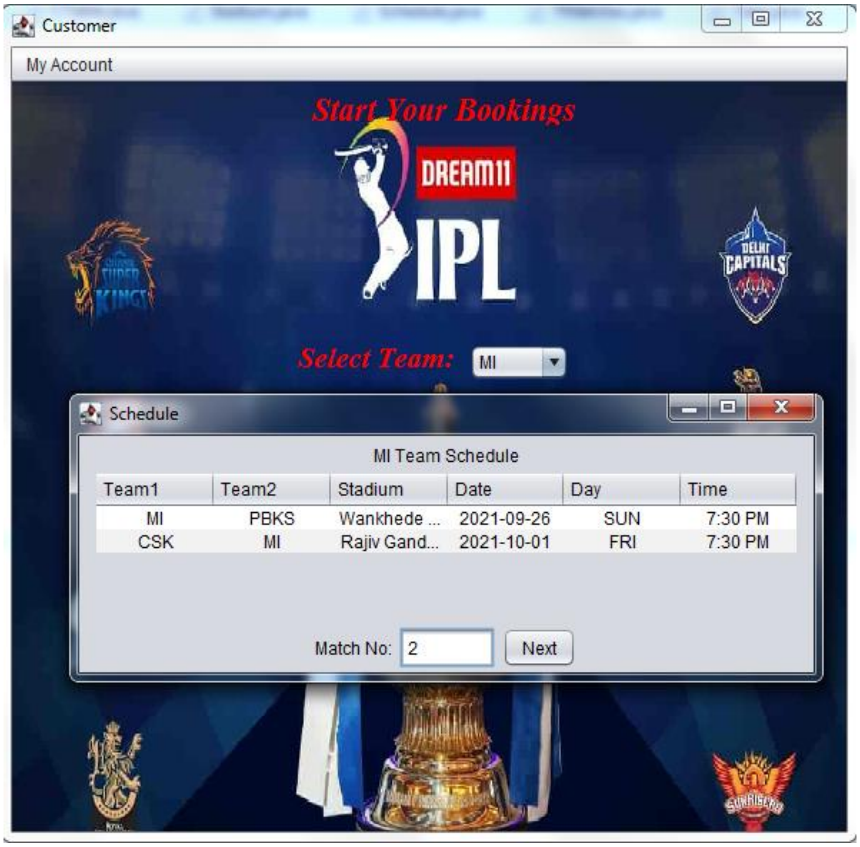
Username: CS_02

Password: ********

LOGIN SUCCESSFUL



Select Team: CSK

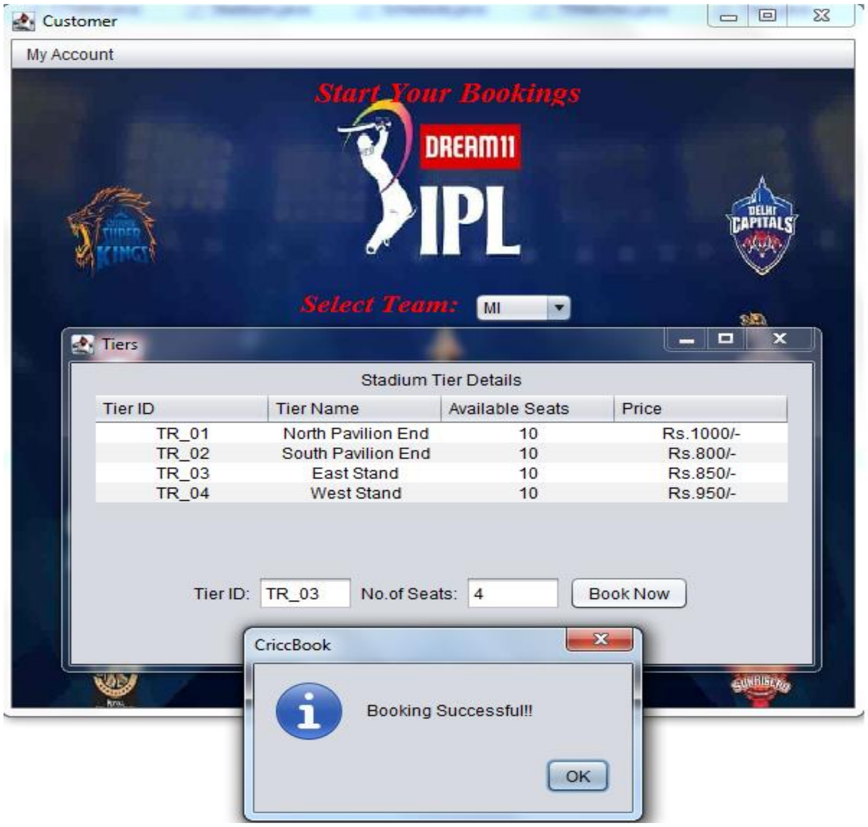


Select Team: MI

MI Team Schedule

Team1	Team2	Stadium	Date	Day	Time
MI	PBKS	Wankhede ...	2021-09-26	SUN	7:30 PM
CSK	MI	Rajiv Gand...	2021-10-01	FRI	7:30 PM

Match No: 2



Select Team: MI

Stadium Tier Details

Tier ID	Tier Name	Available Seats	Price
TR_01	North Pavilion End	10	Rs.1000/-
TR_02	South Pavilion End	10	Rs.800/-
TR_03	East Stand	10	Rs.850/-
TR_04	West Stand	10	Rs.950/-

Tier ID: TR_03 No.of Seats: 4

Booking Successful!!

CONCLUSION:

To conclude we build an application which is a very handy for all the followers of the cricket world where they can book tickets from the comfort of their home, putting fully-fledged queues, a full stop and saves a lot of time for the user. Additionally, it provides match schedule, timings, venues and many more. It is like a fully equipped cricket application named ***CriccBook***.