MINI PROJECT

CRICKET PLAYYER ANALYSIS

Aim:

To construct a database for the Cricket player analysis and connect it with my SQL using java.

Algorithm:

1.**Initialize Data Structures:**

- Create an ArrayList<Player> to store player details.
- Use a Scanner object for user input.

2.**Main Menu (Loop):**

- Display the main menu with the following options:
    1. Add a player
    2. View all players
    3. Analyze players
    4. Exit
- Continuously show the menu until the user selects option 4 to exit.

3.**Add Player (Option 1):**

- Prompt the user for player details: name, age, total runs, total wickets, and strike rate.
- Create a Player object with the provided details and add it to the players list.
- Confirm successful addition of the player.

4.**View All Players (Option 2):**

- Check if the players list is empty:
    - If empty, display a message stating no players are available.
    - If not empty, display a table with player details (name, age, runs, wickets, strike rate).

5.**Analyze Players (Option 3):**

- Check if the players list is empty:
    - If empty, display a message stating no players available for analysis.
    - If not empty, perform the analysis:
        - Find the player with the highest total runs.
        - Find the player with the highest strike rate.

- Display the results of the analysis: highest runs and best strike rate.

6. Stop.

PROGRAM:

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Scanner;

class Player {
    private String name;
    private int age;
    private int runs;
    private int wickets;
    private float strikeRate;

    // Constructor
    public Player(String name, int age, int runs, int wickets, float strikeRate) {
        this.name = name;
        this.age = age;
        this.runs = runs;
        this.wickets = wickets;
        this.strikeRate = strikeRate;
    }

    // Getters
    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
```

```
    }

    public int getRuns() {
        return runs;
    }

    public int getWickets() {
        return wickets;
    }

    public float getStrikeRate() {
        return strikeRate;
    }

    // Display Player Information
    public void display() {
        System.out.printf("%-20s %-5d %-10d %-10d %-10.2f\n", name, age, runs, wickets,
strikeRate);
    }
}

public class CricketPlayerAnalysis {

    private static ArrayList<Player> players = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        int choice;
        do {
            System.out.println("\n--- Cricket Player Analysis ---");
            System.out.println("1. Add Player");
```

```
        System.out.println("2. View All Players");
        System.out.println("3. Analyze Players");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();

        switch (choice) {
            case 1:
                addPlayer();
                break;
            case 2:
                viewAllPlayers();
                break;
            case 3:
                analyzePlayers();
                break;
            case 4:
                System.out.println("Exiting... Thank you!");
                break;
            default:
                System.out.println("Invalid choice! Please try again.");
        }
    } while (choice != 4);
}

// Method to add a new player
private static void addPlayer() {
    System.out.print("Enter player name: ");
    String name = scanner.next();
    System.out.print("Enter age: ");
```

```
        int age = scanner.nextInt();
        System.out.print("Enter total runs: ");
        int runs = scanner.nextInt();
        System.out.print("Enter total wickets: ");
        int wickets = scanner.nextInt();
        System.out.print("Enter strike rate: ");
        float strikeRate = scanner.nextFloat();

        Player player = new Player(name, age, runs, wickets, strikeRate);
        players.add(player);
        System.out.println("Player added successfully!");
    }

    // Method to view all players
    private static void viewAllPlayers() {
        if (players.isEmpty()) {
            System.out.println("No players available.");
            return;
        }
        System.out.printf("%-20s %-5s %-10s %-10s %-10s\n", "Name", "Age", "Runs",
"Wickets", "Strike Rate");
        System.out.println("-----------------------------------------------------------");
        for (Player player : players) {
            player.display();
        }
    }

    // Method to analyze players
    private static void analyzePlayers() {
        if (players.isEmpty()) {
            System.out.println("No players available for analysis.");
```

```
            return;
        }

        Player maxRunsPlayer =
players.stream().max(Comparator.comparingInt(Player::getRuns)).orElse(null);
        Player bestStrikeRatePlayer =
players.stream().max(Comparator.comparingDouble(Player::getStrikeRate)).orElse(null);

        System.out.println("\n--- Analysis Results ---");
        if (maxRunsPlayer != null) {
            System.out.println("Player with Highest Runs:");
            maxRunsPlayer.display();
        }

        if (bestStrikeRatePlayer != null) {
            System.out.println("\nPlayer with Best Strike Rate:");
            bestStrikeRatePlayer.display();
        }
    }
}
```

OUTPUT:

RESULT:

The database construction for the cricket player analysis has been successfully complected and connected with mySQL using java.