# CRICKET MANAGEMENT SYSTEM



NAME: KAVISH

ID NO. : 2021A7PS0559P

# INTRODUCTION

My DBMS project is based on Cricket database management system. It provides various information about the various teams participating in the World Cup, in which all the major countries participate. It also provides us with information about the various players participating in the tournament. The database contains details of players, coaches, matches, wicket keepers and umpires among others. All the useful information about the entire World Cup can be found here.

# DATA REQUIREMENTS

Entities:

1. Team: is an entity type which has many attributes like Team Name which uses the data type varchar. Every team has been given a Team ID which is the primary key which is of data type varchar. There are another attribute – Coach Name and Home Ground which accepts varchar data type.

Primary key cannot have null value.

1. Players: is an entity type which has an attribute – Player Name which is of the data type varchar. It has a primary key, Player ID, which cannot have null value. It has a foreign key, Team ID which is the primary key of the entity, Team. There is an attribute age of which accepts only integer values and another attribute Playing Type

which is of data type varchar.

1. Scores: is an entity type with a primary key, Score ID of data type INT. It has two foreign keys, i) Player id from table Players and ii) Match ID from table Matches. Runs Scored and Wickets Taken are also attributes of this table of data type INT. Aso have one more attribute Overs Bowled having data type float.
2. Matches: is an entity type with a primary key, Match ID, of INT data type. It has attributes like Team1 ID, Team2 ID, Winner Team ID of data type INT and Location of data type varchar. Match Date is an attribute which uses the data type date.

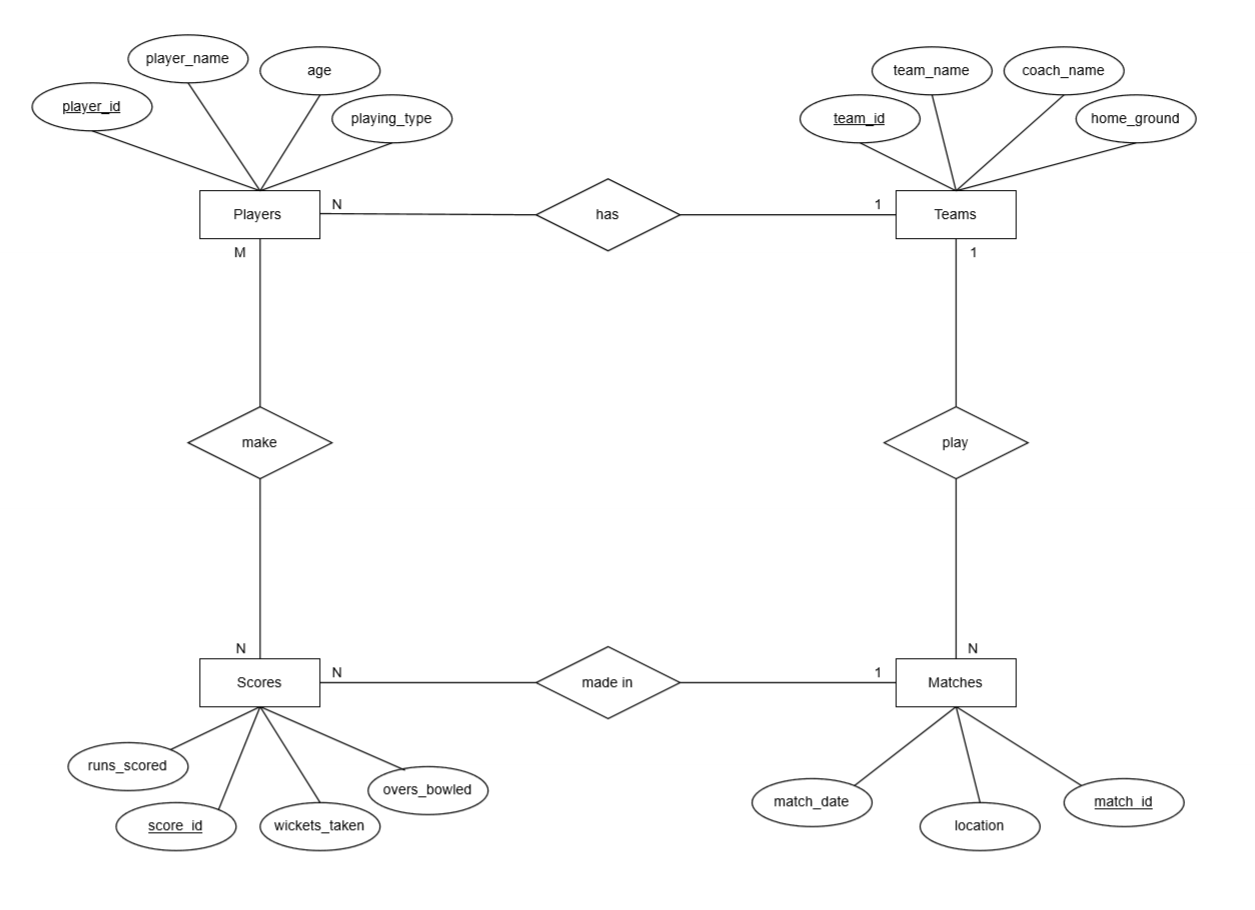
Relationships:

Cricket player plays in team: a cricket player can play in only one team but a team can have many players in it but a team must have players in it.

Team plays match: Team can play many matches and a match can be played by two teams.

Runs Scored by a player: Multiple players can score in a single match and single player can score in multiple matches.

**ER DIAGRAM:**



RELATIONAL SCHEMA

* **Team** (team\_id, team\_name, coach\_name, home\_ground)
* **Player** (player\_id, player\_name, age, playing\_type, team\_id)
* **Match** (match\_id, match\_date, location, team1\_id, team2\_id, winner\_team\_id)
* **Score** (score\_id, match\_id, player\_id, runs\_scored, wickets\_taken,overs\_bowled)

**NOTE:-**

**Candidate Keys Are:**

team\_id

player\_id

match\_id

score\_id

**Foreign Keys Are:**

Players.team\_id references Teams.team\_id

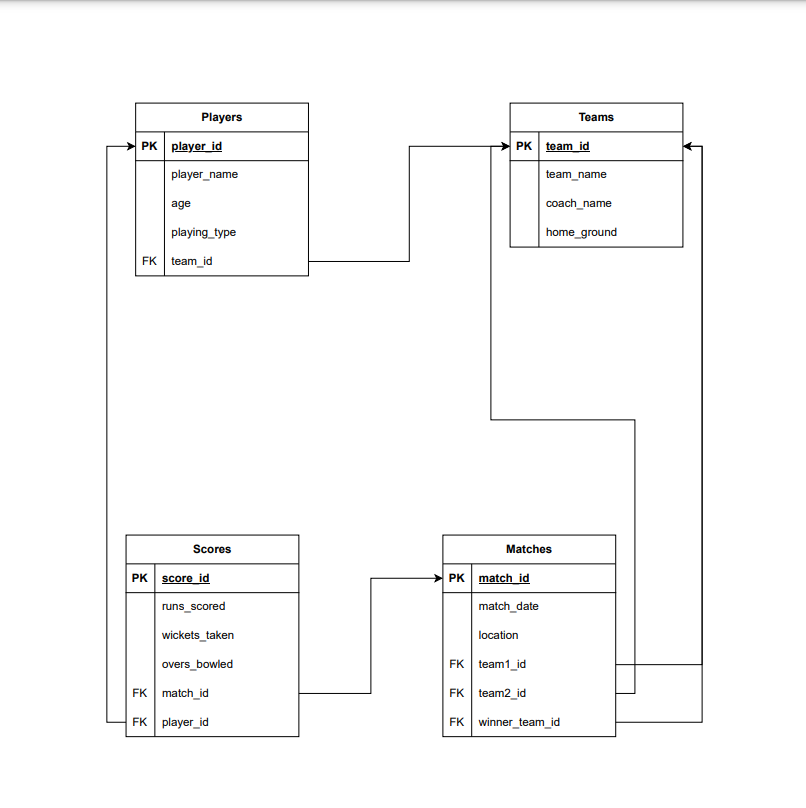
Matches.team1\_id references Teams.team\_id

Matches.team2\_id references Teams.team\_id

Matches.winner\_team\_id references Teams.team\_id

Scores.match\_id references Matches.match\_id

Scores.player\_id references Players.player\_id



NORMALIZATION

**Functional dependencies for each relation:**

* Functional dependencies for Players:

player\_id → player\_name, age, playing\_type,

team\_id

team\_id → team\_id, team name, coach\_name,

home\_ground

* Functional dependencies for Matches:

match\_id → match\_date, location, team1\_id

team2\_id, winner\_team\_id

* Functional dependencies for Scores:

score\_id → match\_id, player\_id, runs\_scored,

wickets\_taken, overs\_bowled

NOTE:-

* 1. There are no partial dependencies in any

Of the relations.

* 1. For Players, we have a transitive dependency: team\_id → team\_name, coach\_name, home\_ground.

We can remove this by creating a new relation

for Teams with team\_id as the primary key and

team\_name, coach\_name, and home\_ground as

attributes.

Teams (team\_id, team\_name, coach\_name,

home\_ground)

* 1. For Scores, we have a transitive dependency: match\_id → team1\_id, team2\_id, winner\_team\_id. We can remove this by creating a new relation for Matches with match\_id as the primary key and match\_date, location, team1\_id, team2\_id, and winner\_team\_id as attributes.

Matches (match\_id, match\_date, location, team1\_id, team2\_id, winner\_team\_id)

**The final normalized relations are:**

* Players (player\_id, player\_name, age, playing\_type, team\_id)
* Teams (team\_id, team\_name, coach\_name, home\_ground)
* Matches (match\_id, match\_date, location, team1\_id, team2\_id, winner\_team\_id)
* Scores (score\_id, match\_id, player\_id, runs\_scored, wickets\_taken, overs\_bowled)

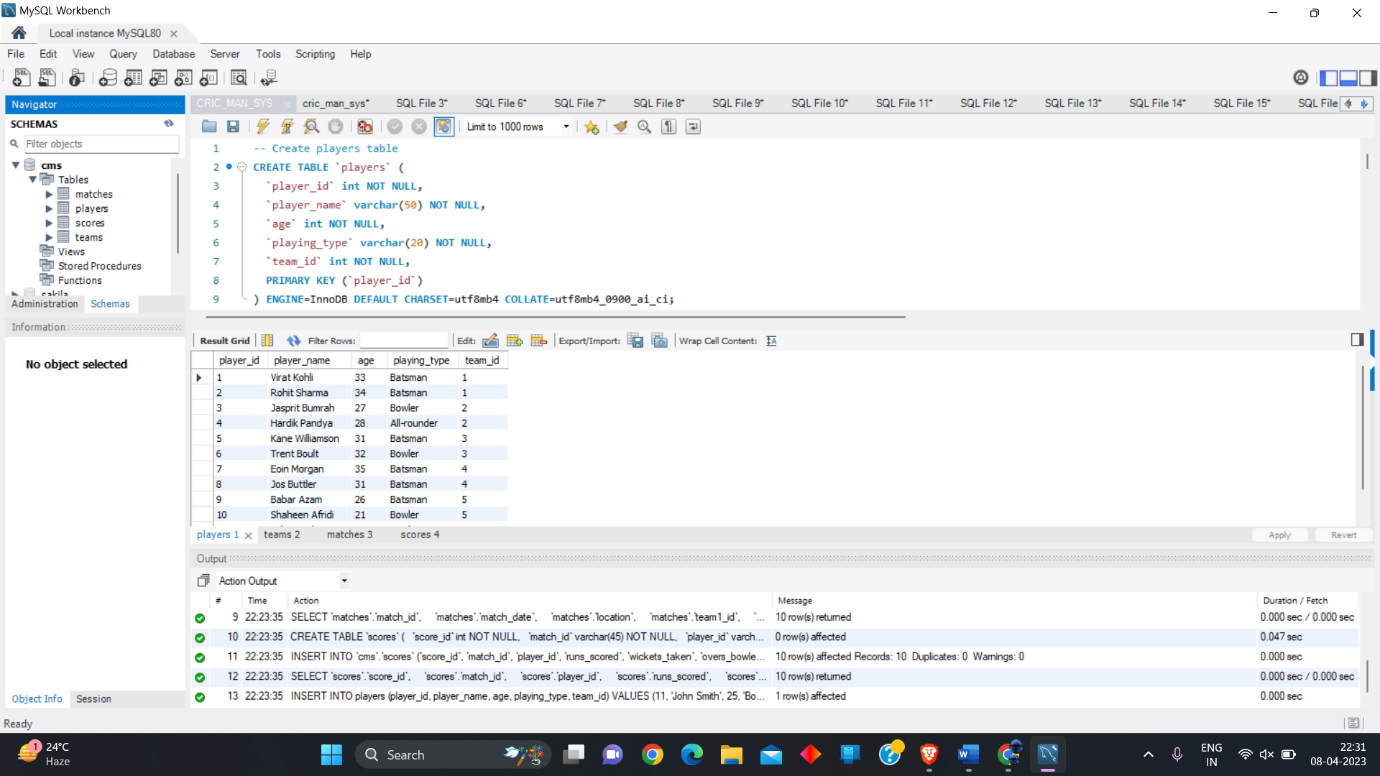
\*\*ALL RELATONDS ARE NOW IN 3NF, AND THERE ARE NO TRANSITIVE DEPENDENCIES.

# SQL OUERIES AND THEIR OUTPUT SCREENSHOTS

1.Create a table for players with fields such as

player\_id, player\_name, player\_age,

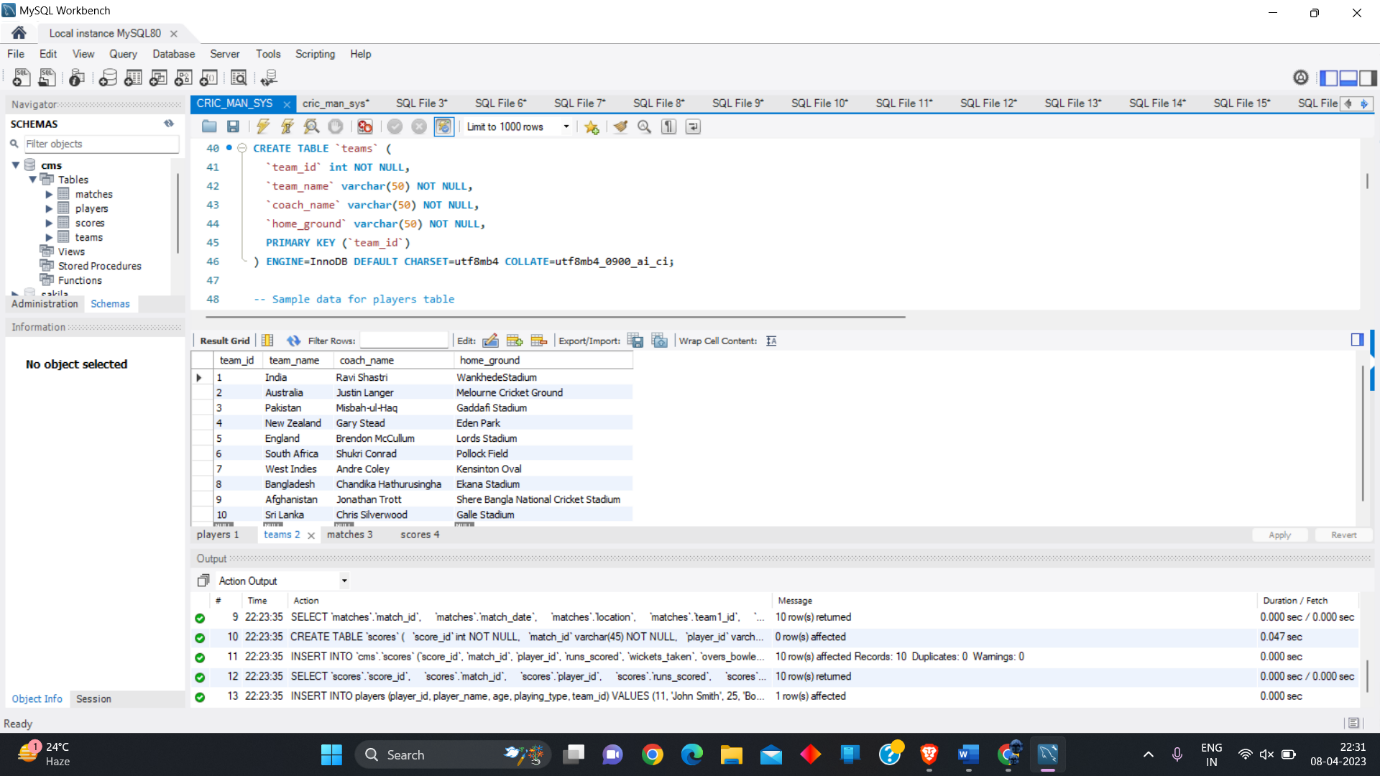
player\_position, and team\_id.



2.Create a table for teams with fields such as

team\_id, team\_name, coach\_name, and

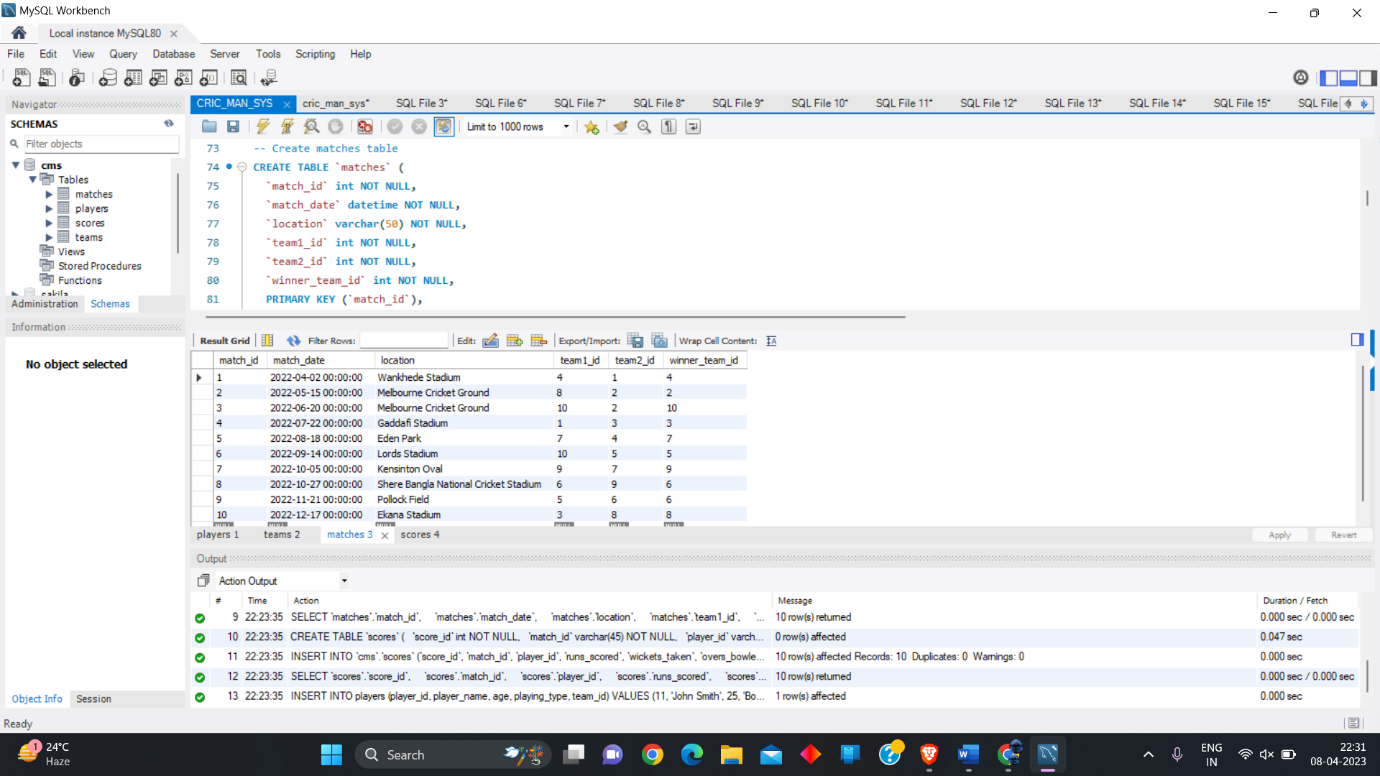
home\_ground.



3.Create a table for matches with fields such as

match\_id, match\_date, location, and

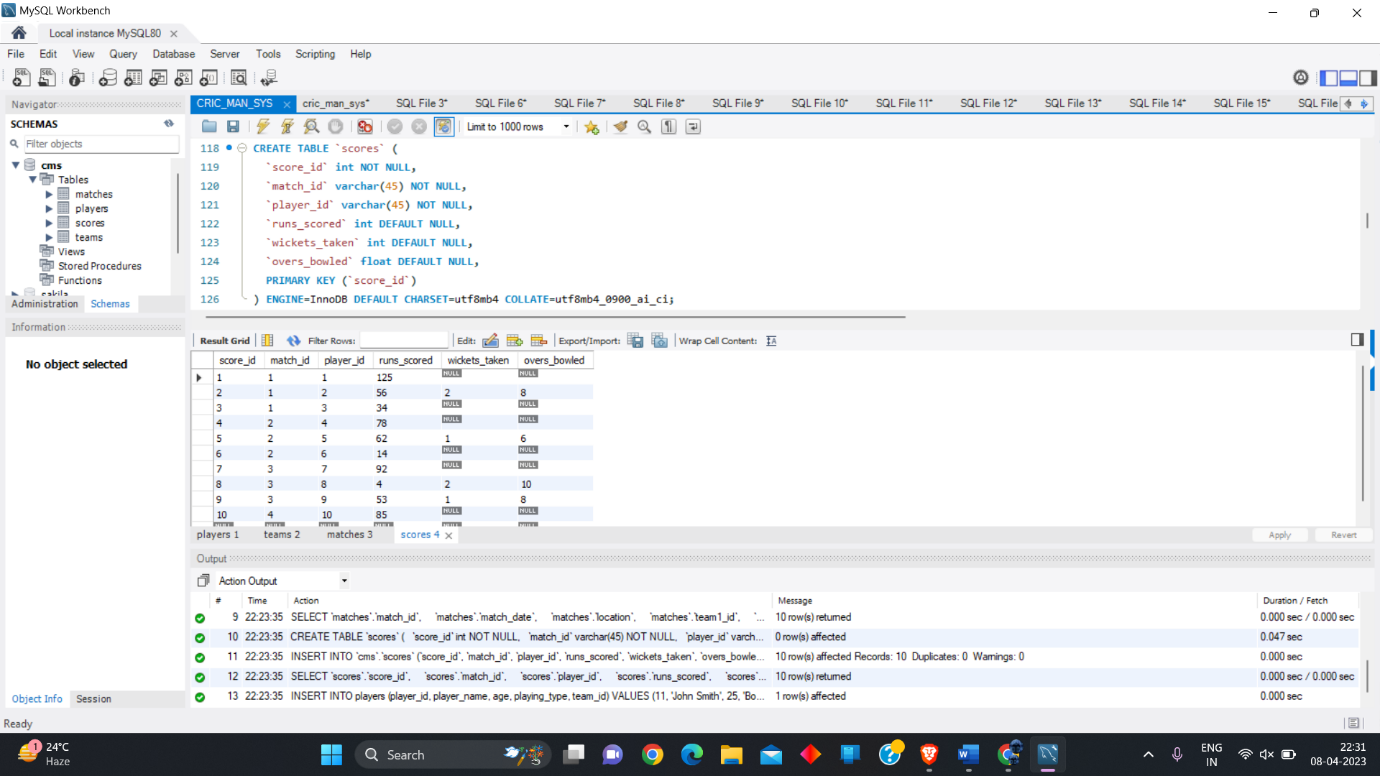
winner\_team\_id.



4.Create a table for scores with fields such as

score\_id, match\_id, player\_id, runs\_scored,

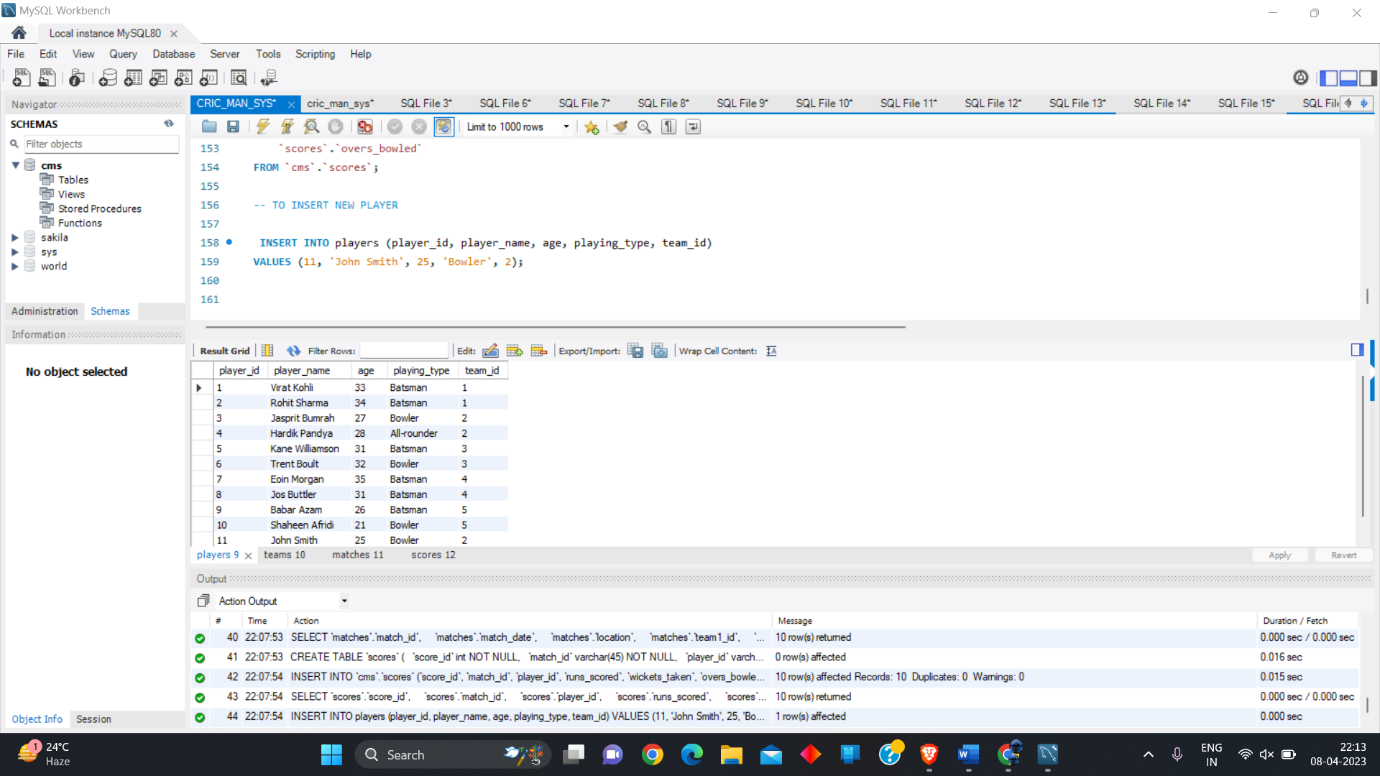
wickets\_taken, and overs\_bowled.



5.Insert a new player record with player\_name

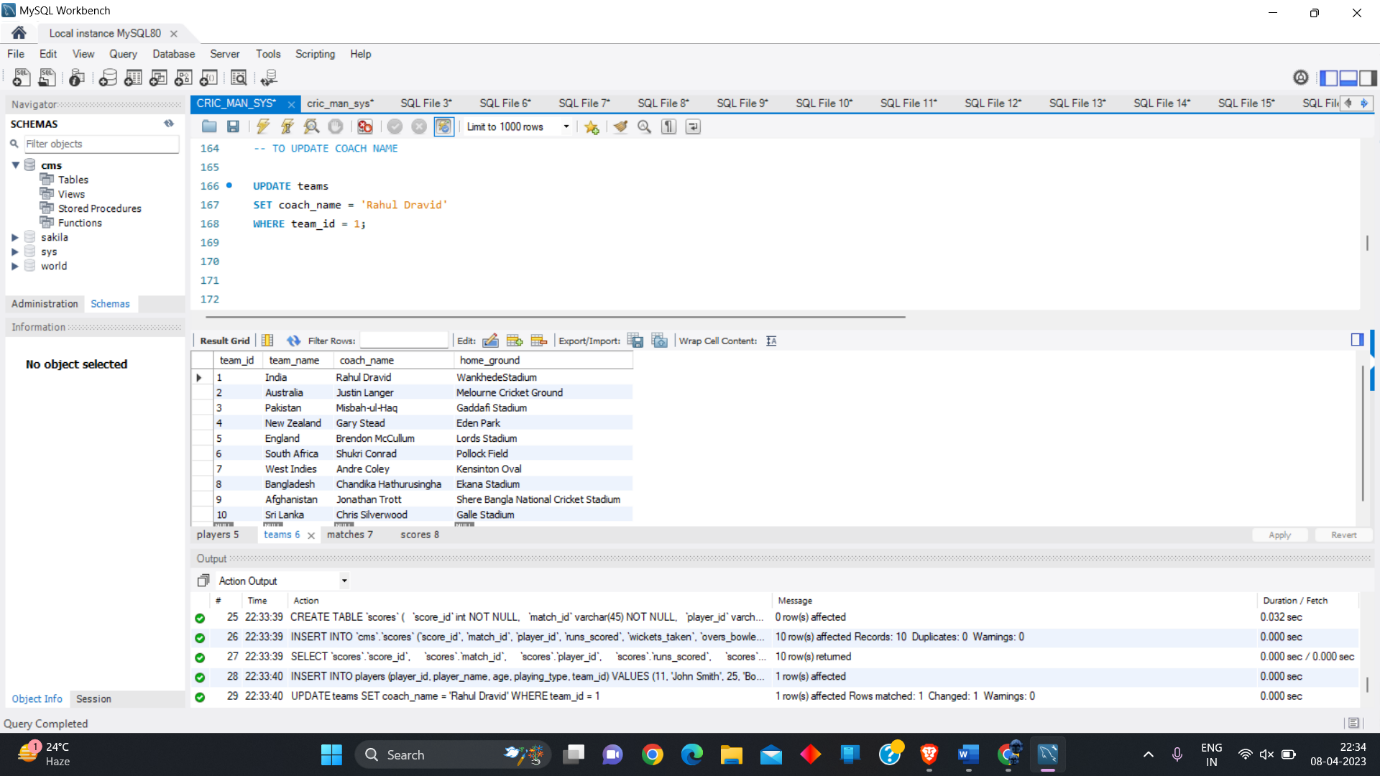
“John Smith”, player\_age 25, player\_type

“bowler”, and team\_id 2.



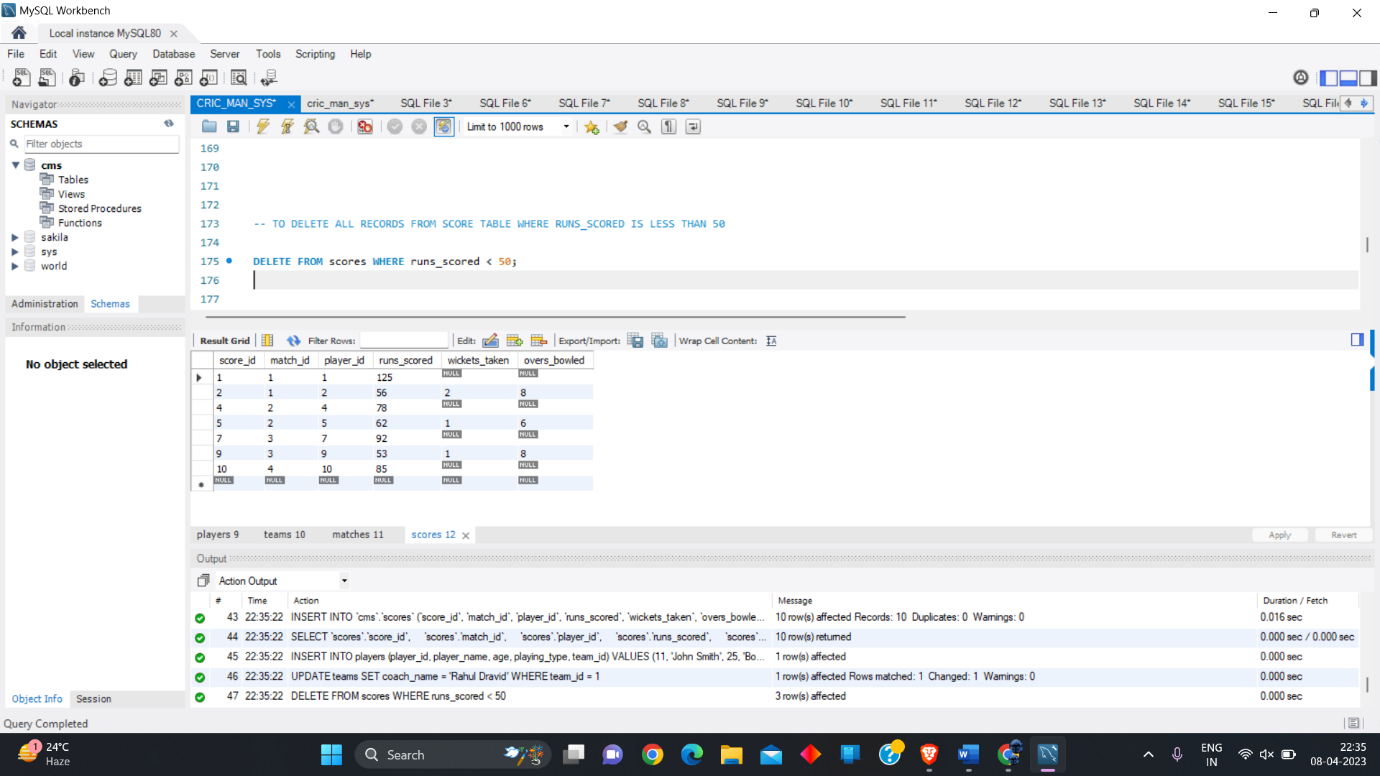
6.Update the coach\_name for team\_id 1 to “Rahul

Dravid”.



7.Delete all records from the scores table where

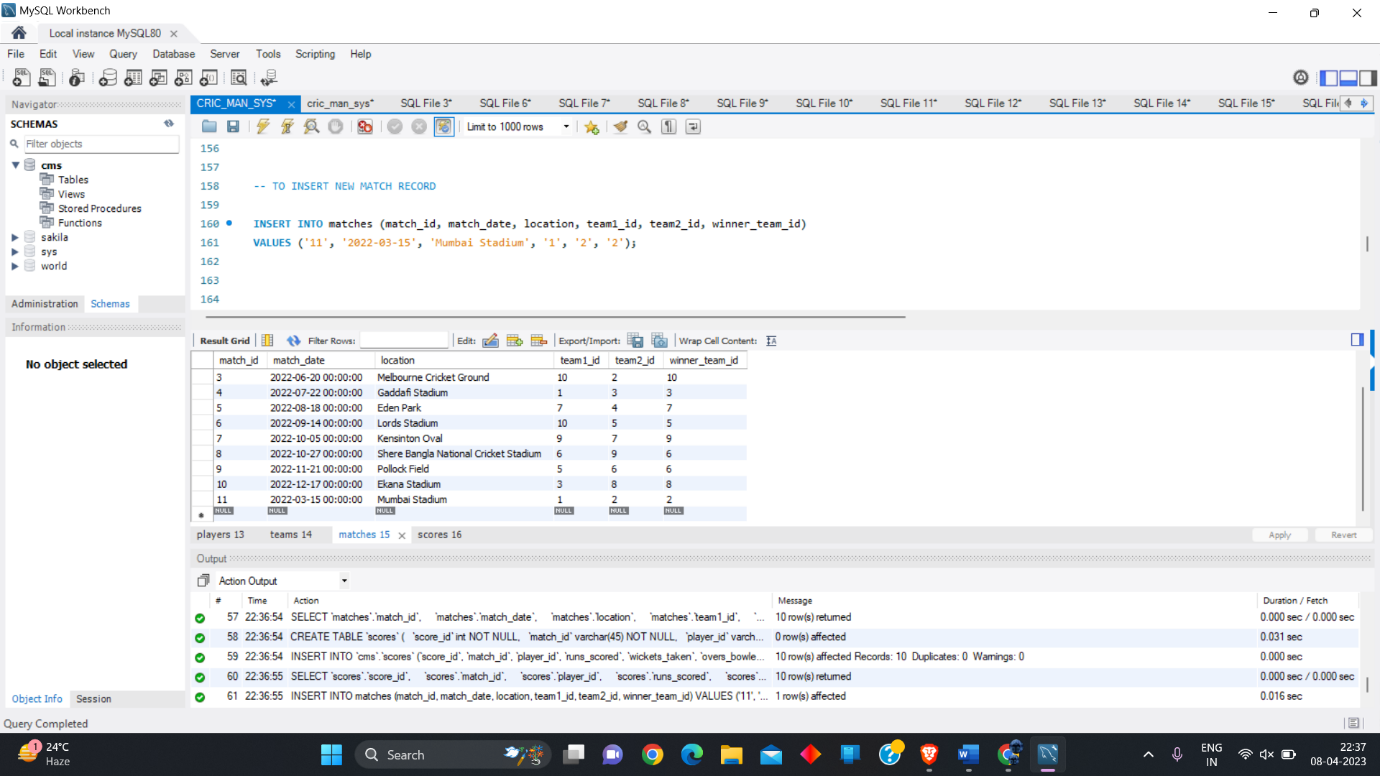
runs\_scored is less than 50.



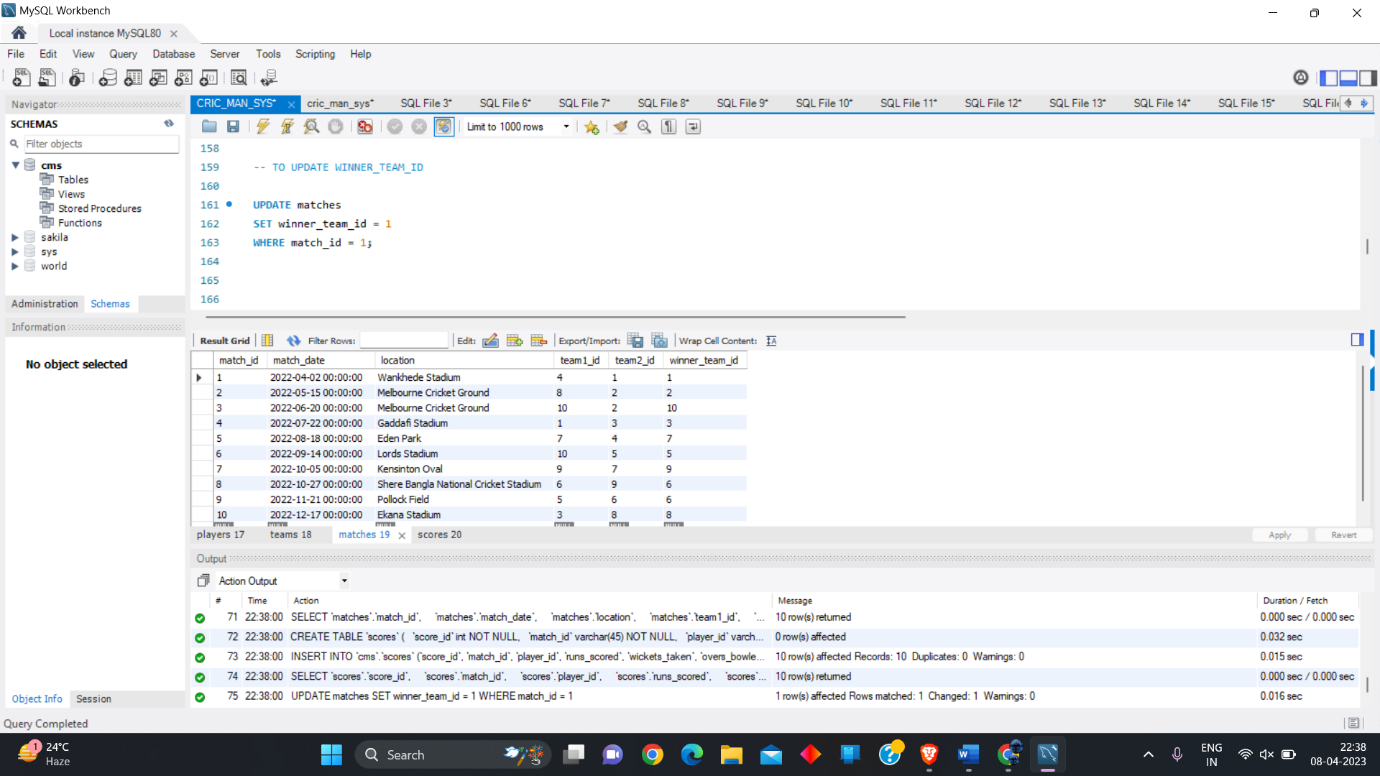
8.Insert a new match record with match\_date “2022

03-15", location “Mumbai”, and competing

team\_ids 1 and 2.

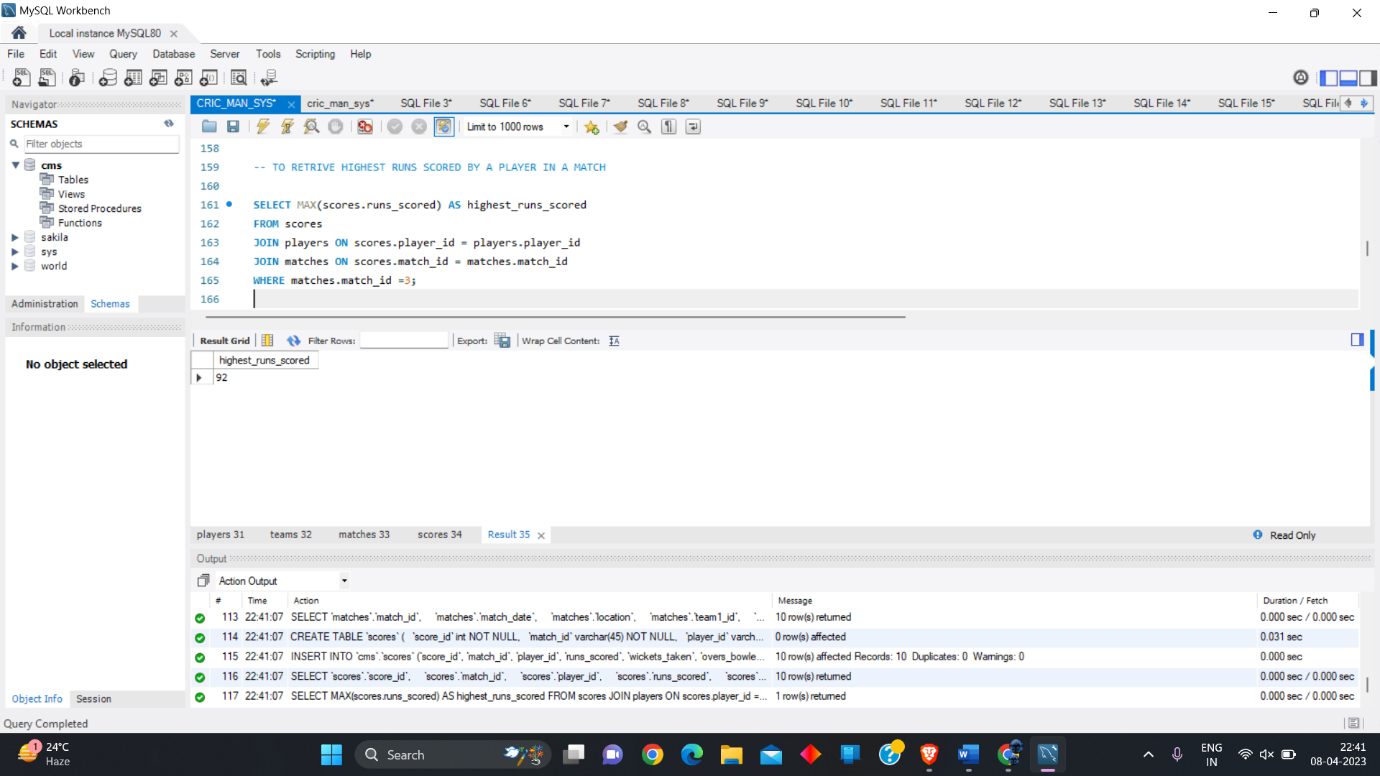


9.Update the winner\_team\_id for match\_id 1 to 1.



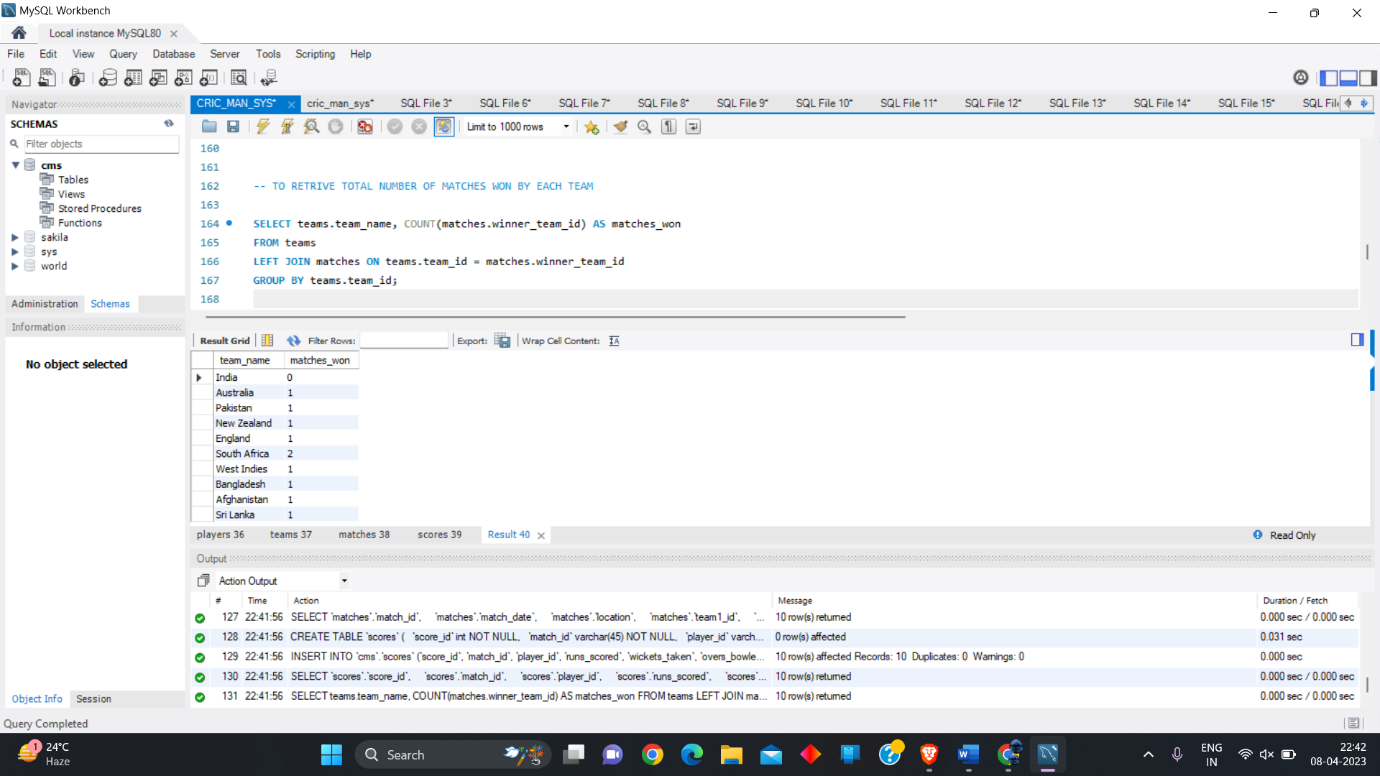
10.Retrieve the highest runs\_scored by a player in a

match.

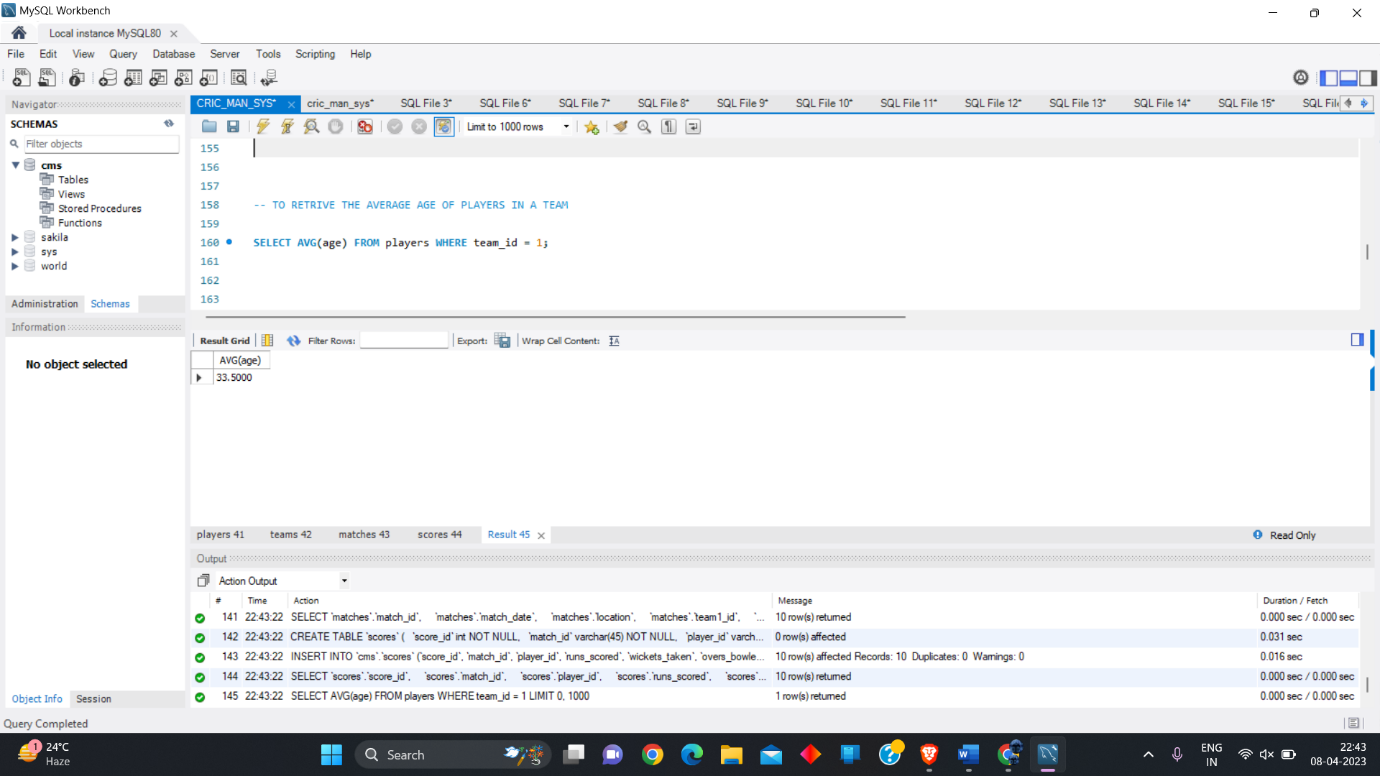


11.Retrieve the total number of matches won by

each team.

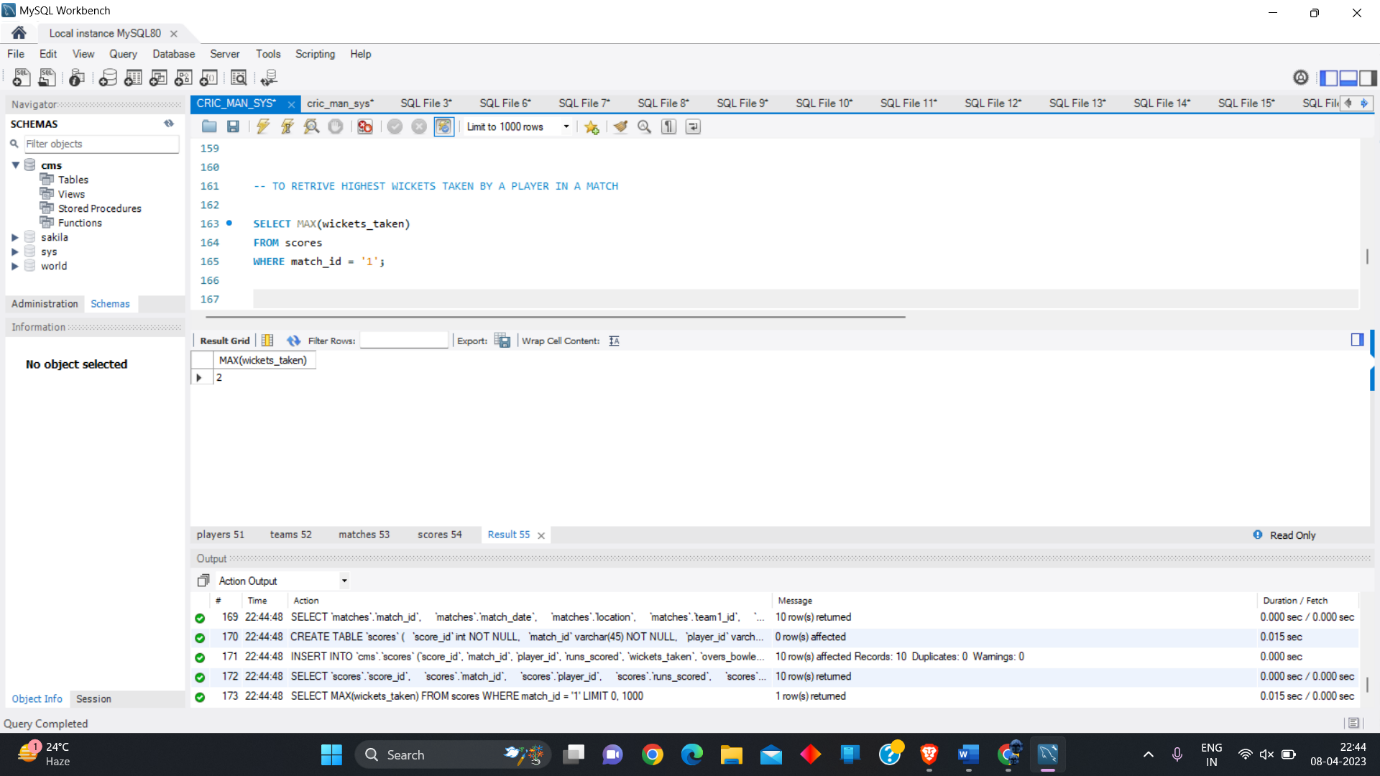


12.Retrieve the average age of players in a team.



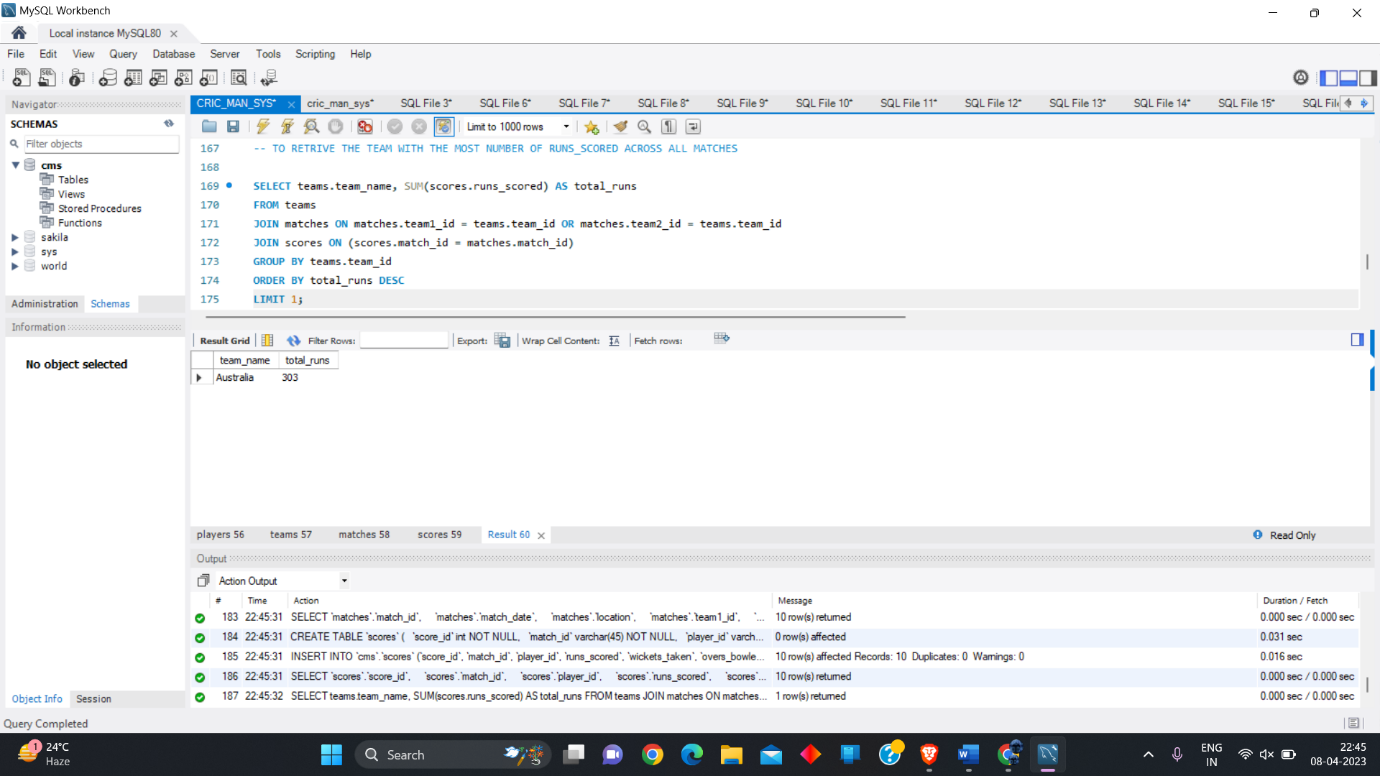
13.Retrieve the highest wickets\_taken by a player in

a match.



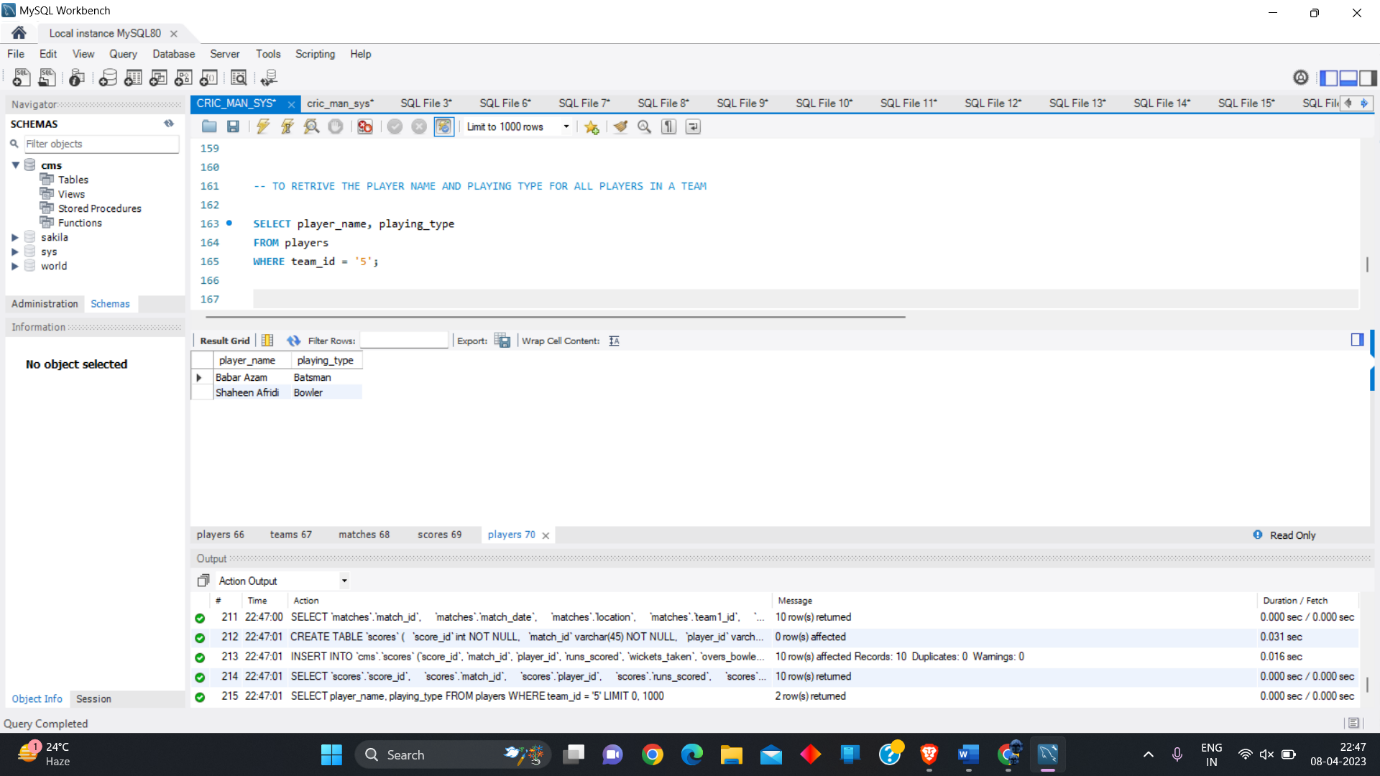
14.Retrieve the team with the most number of

runs\_scored across all matches.



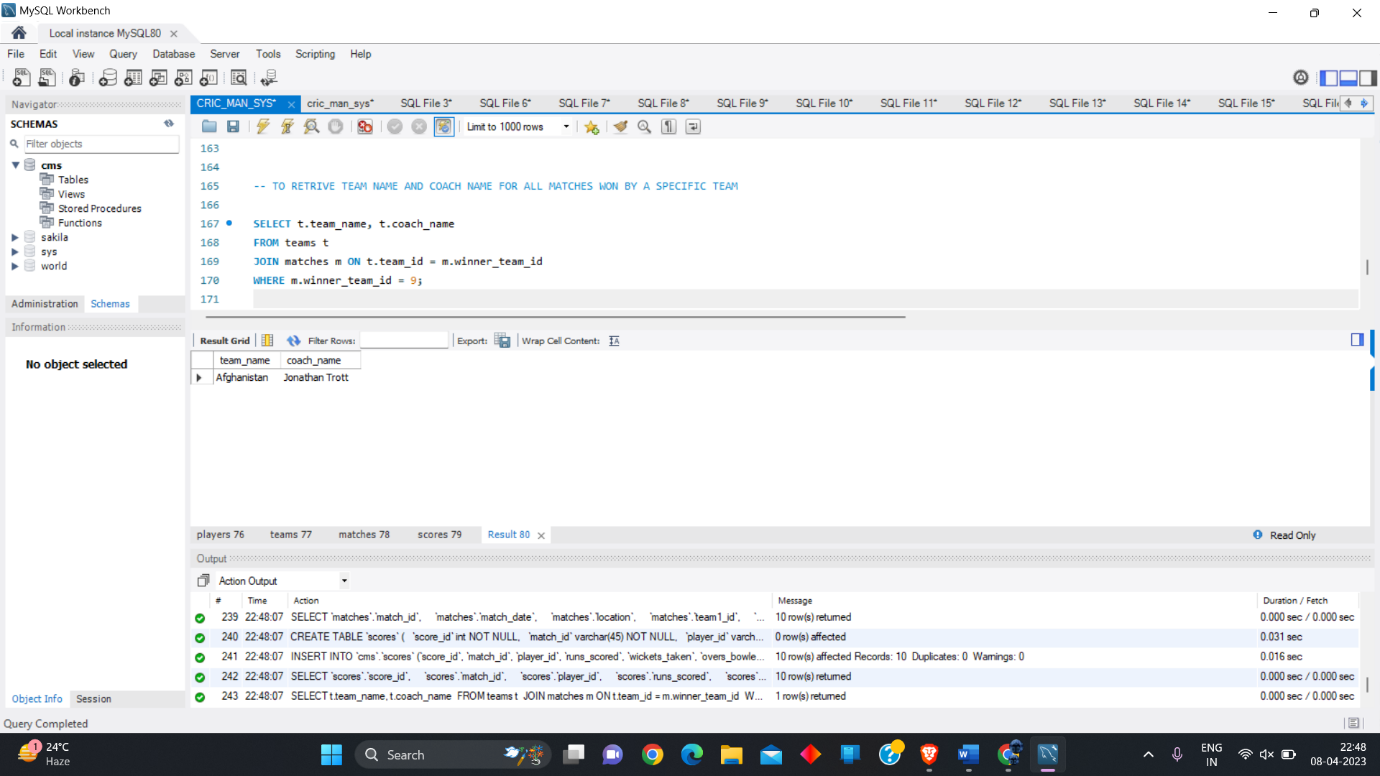
15.Retrieve the player\_name and player\_type for

all players in a specific team.



16.Retrieve the team\_name and coach\_name for all

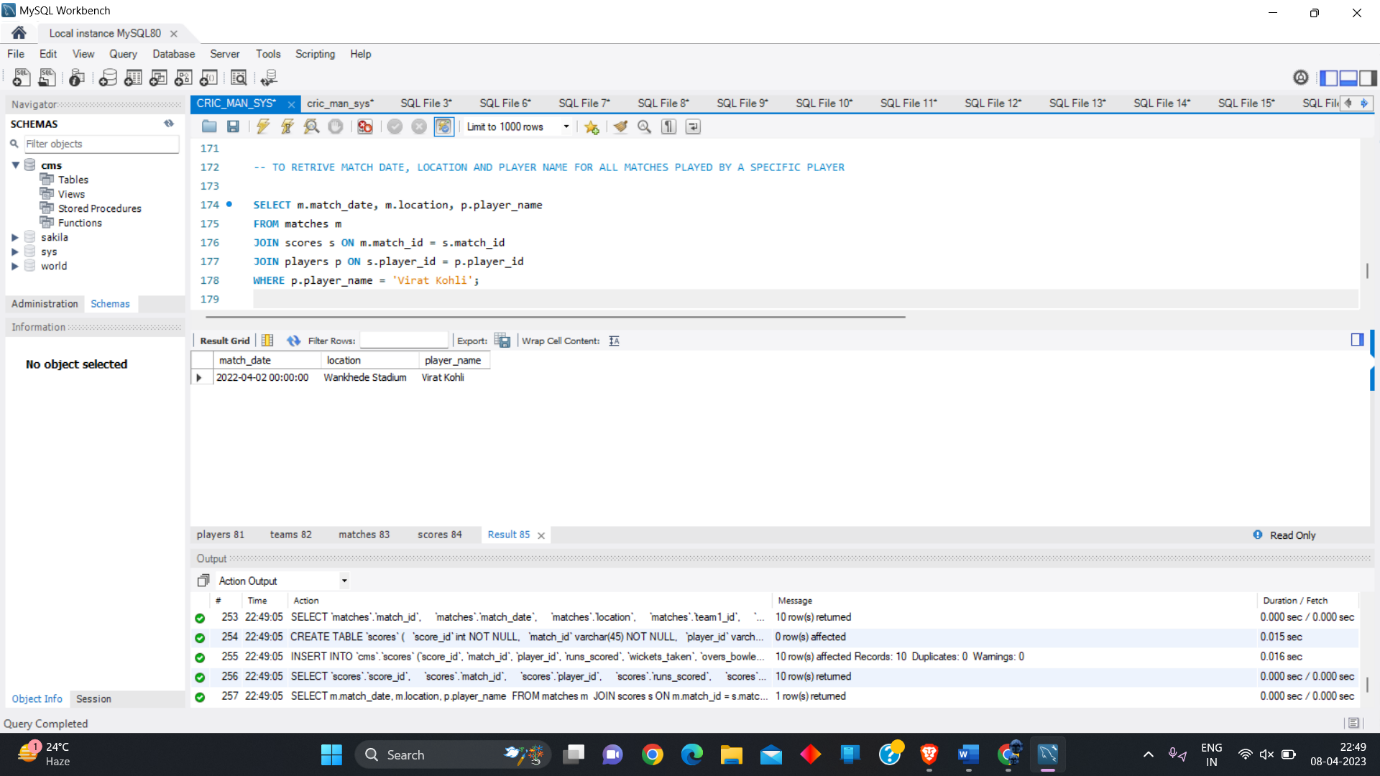
matches won by a specific team.



17.Retrieve the match\_date, location, and

player\_name for all matches played by a specific

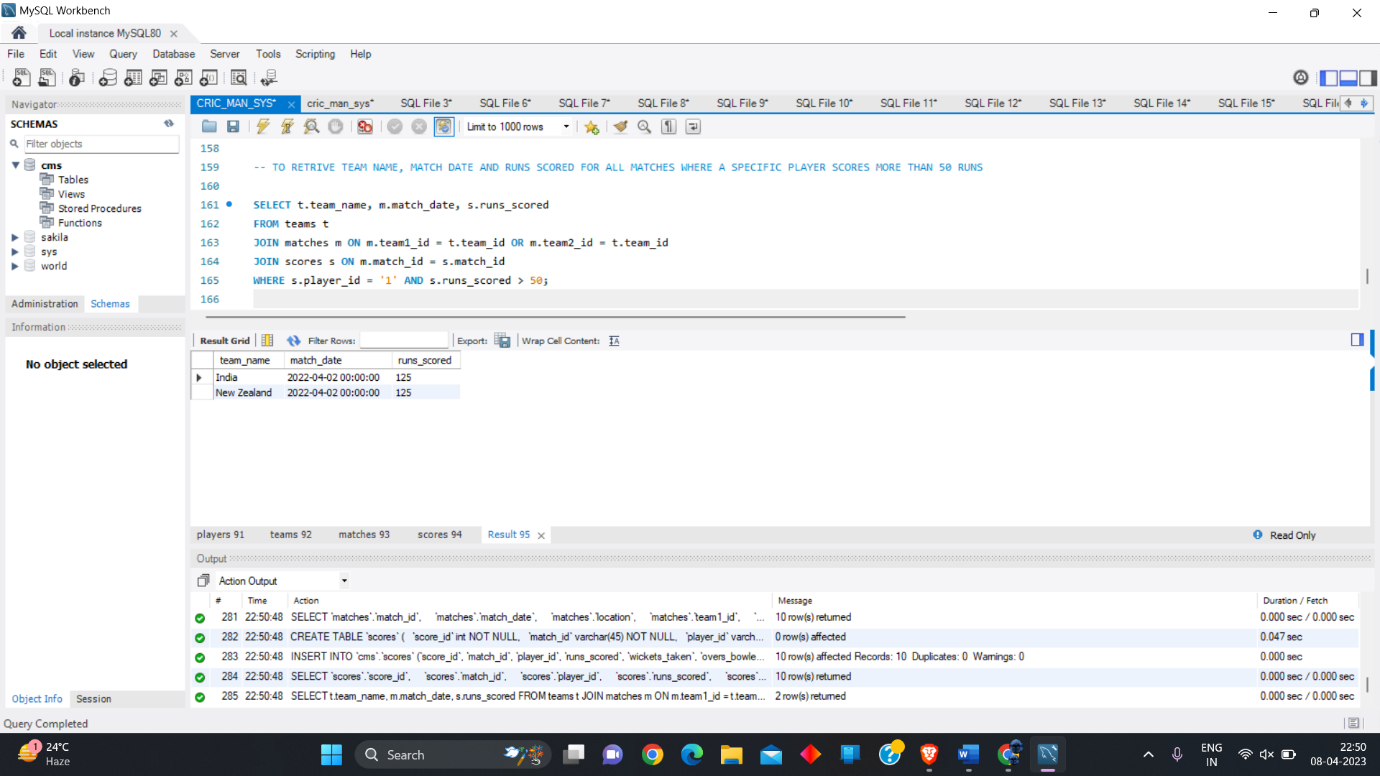
player.



18.Retrieve the team\_name, match\_date, and

runs\_scored for all matches where a specific

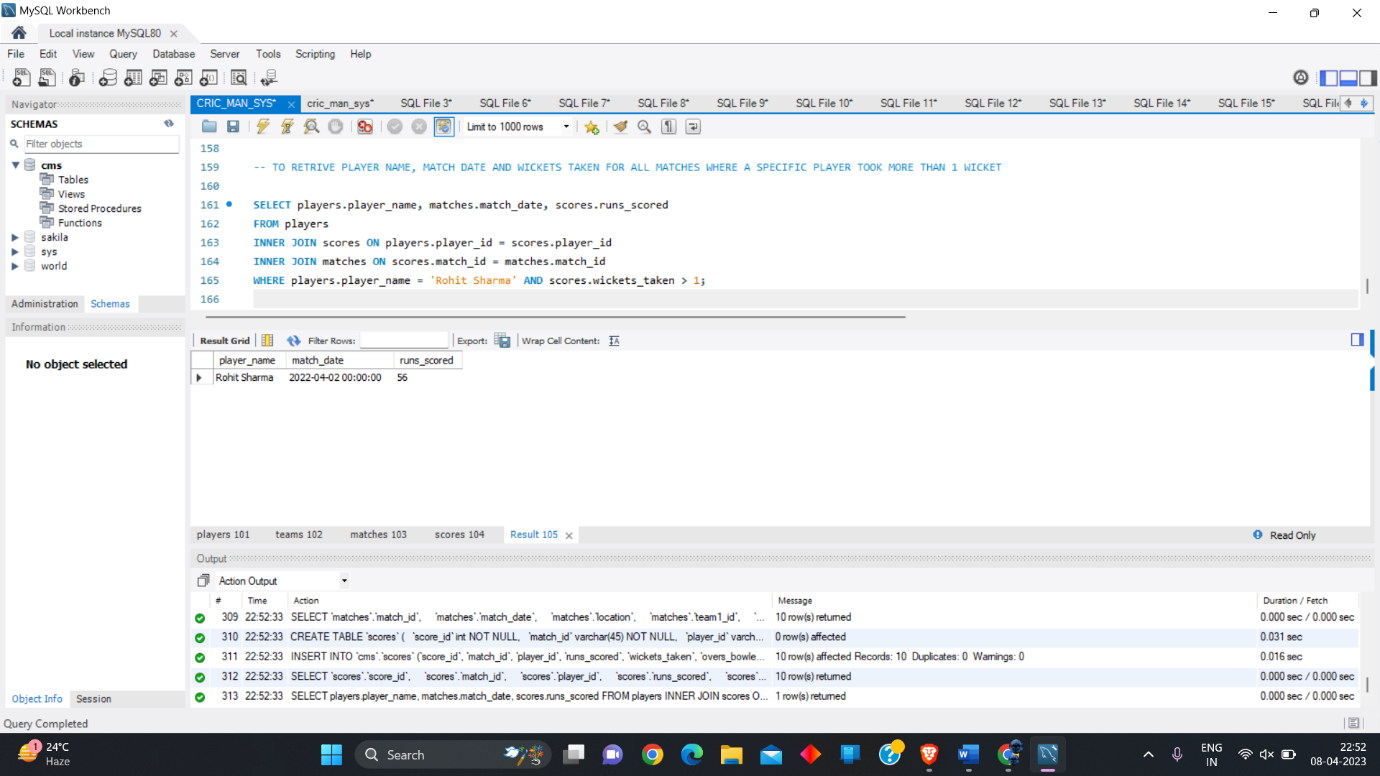
player scored more than 50 runs.



19.Retrieve the player\_name, match\_date, and

wickets\_taken for all matches where a specific

player took more than 1 wickets.



# VIDEO LINK

<https://drive.google.com/file/d/1rQsTXOuHK-1Ssb0-a4PRfein20XJvz7U/view?usp=share_link>