**DBMS - MINI PROJECT**

**“IPL Auction management database”**

Submitted By:

Srinidhi Bharadwaj

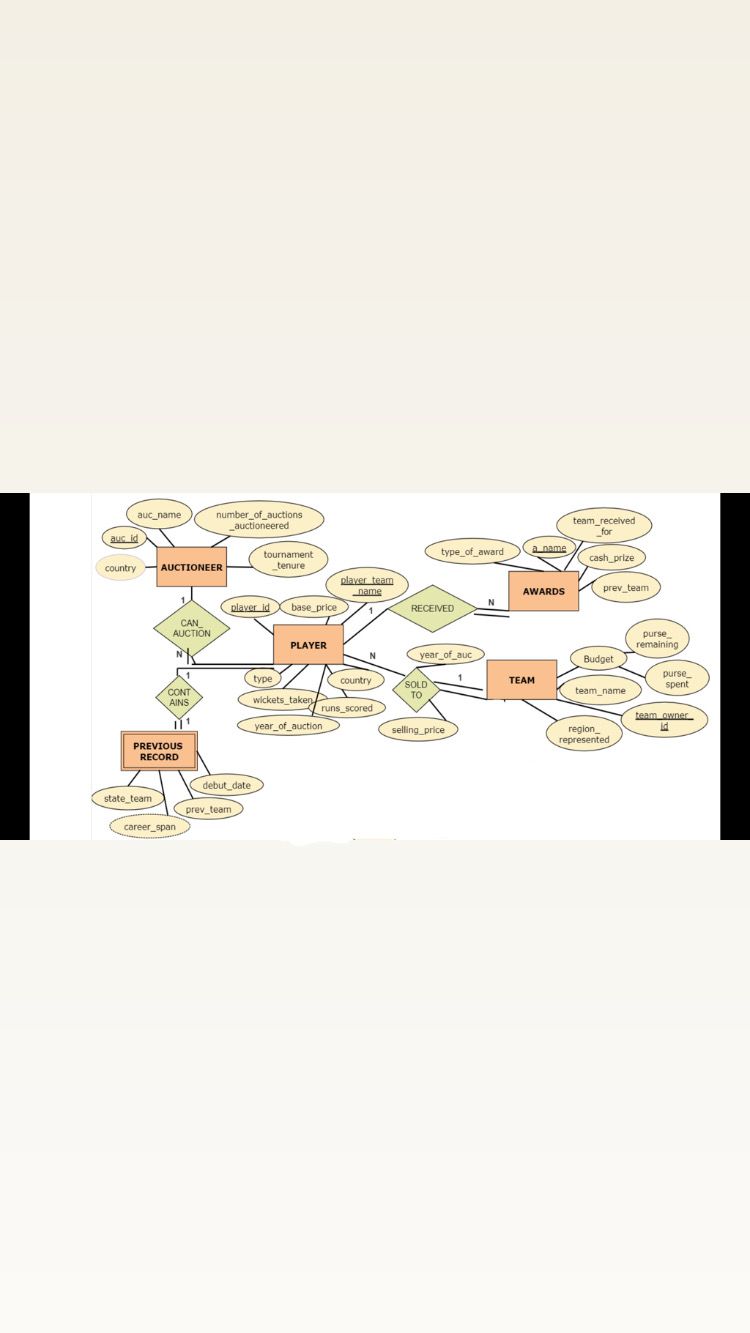
PES1UG20CS143

V Semester Section C

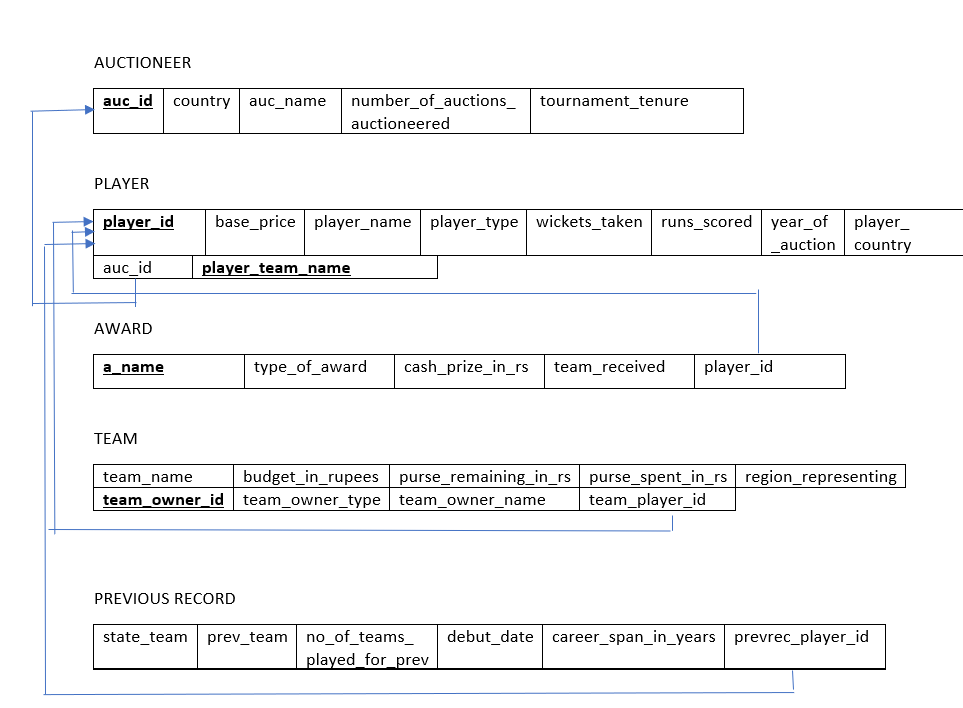
**ABSTRACT**

//Short Description and Scope of the Project

**ER Diagram**



**Relational Schema**



**DDL statements - Building the database**

1. create table auctioneer(auc\_id varchar(10),country varchar(10),auc\_name varchar(20),number\_of\_auctions\_auctioneered int,tournament\_tenure int,primary key(auc\_id));

2. create table player(player\_id varchar(20),player\_team\_name varchar(20),primary key(player\_id,player\_team\_name),base\_price int,player\_name varchar(50),player\_type varchar(20),wickets\_take int, runs\_scored int,player\_country varchar(20),year\_of\_auction varchar(20),player\_auc\_id varchar(10),foreign key(player\_auc\_id) references auctioneer(auc\_id));

3. create table award(a\_name varchar(10),type\_of\_award varchar(20),cash\_prize\_in\_rupees int,team\_received\_for varchar(10),award\_player\_id varchar(20),foreign key(award\_player\_id) references player(player\_id));

4. create table team(team\_name varchar(20),budget\_in\_rupees int,purse\_remaining\_in\_rupees int,purse\_spent\_in\_rupees int,region\_representing varchar(20),team\_owner\_id varchar(10),primary key(team\_owner\_id),team\_owner\_type varchar(20),team\_owner\_name varchar(20),team\_player\_id varchar(20));

5. create table previous\_records(state\_team varchar(10),prev\_team varchar(10),no\_of\_teams\_played\_for\_prev int,debut\_date DATE,career\_span\_in\_years int,prevrec\_player\_id varchar(20),foreign key(prevrec\_player\_id)references player(player\_id));

**Populating the Database**

#for player table:

insert into player values("007","DC",2,"Axar Patel","All-rounder",86,1059,"India","2023","2k23001",8);

#for auctioneer

insert into auctioneer values("2k23002","Australia","Billy",8,2);

#for award:

insert into award values("MOT","Trophy and cash",100000,"DC","007");

#for previous\_record:

insert into previous\_records values("DC","CSK",3,2014-01-21,8,"009");

#for team:

insert into team values ("RCB",90,67,23,"Bangalore","OW001","Business man","Malya");

**Tools Used**

* **Tkinter**
* **Python**
* **Mysql**
* **Python mysql.connector**

**Queries**

**Join queries (at least 6)**

1. names of players who have won at least one award

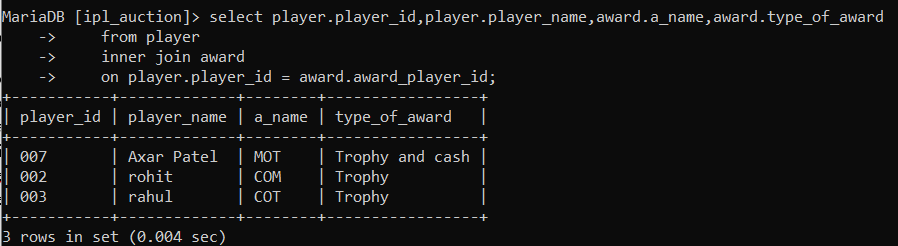
Query: select player.player\_id,player.player\_name,award.a\_name,award.type\_of\_award

from player

inner join award

on player.player\_id = award.award\_player\_id;

output screenshot:



2. names of players who have played more than 10 years of ipl.

Query: select player.player\_id,player.player\_name, previous\_records.career\_span\_in\_years,previous\_records.debut\_date

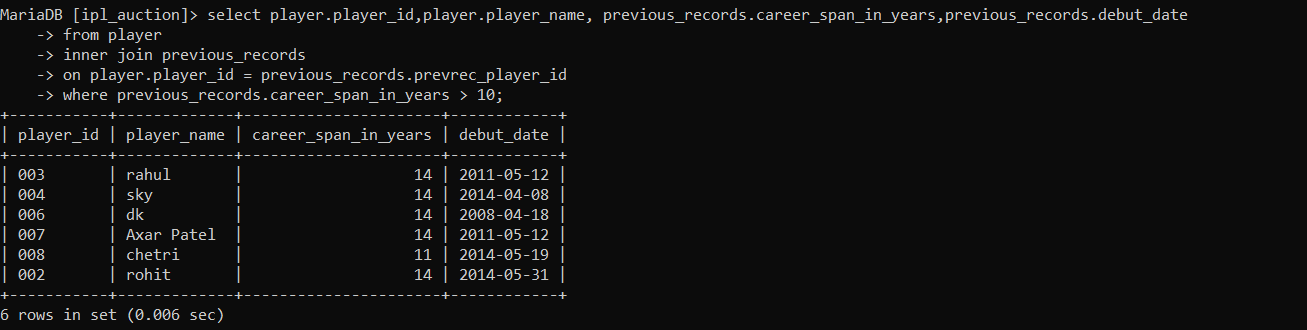
from player

inner join previous\_records

on player.player\_id = previous\_records.prevrec\_player\_id

where previous\_records.career\_span\_in\_years > 10;

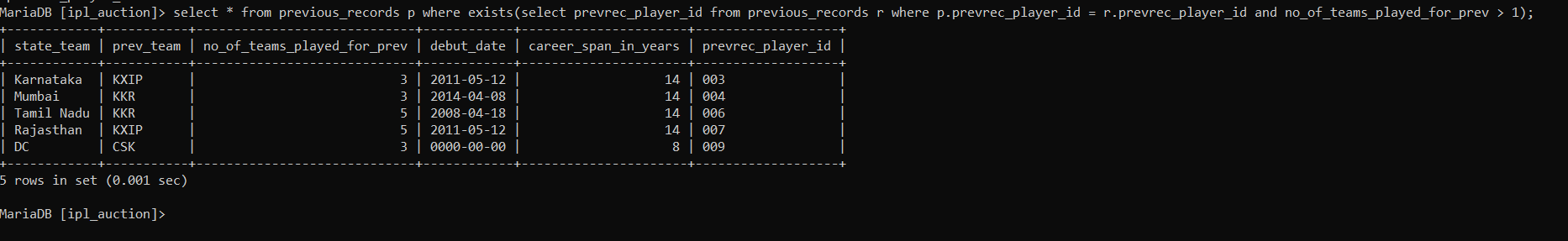
Output screenshot:



3. display previous records of those players, who have played for more than one team.

Query: select \* from previous\_records p where exists(select prevrec\_player\_id from previous\_records r where p.prevrec\_player\_id = r.prevrec\_player\_id and no\_of\_teams\_played\_for\_prev > 1);

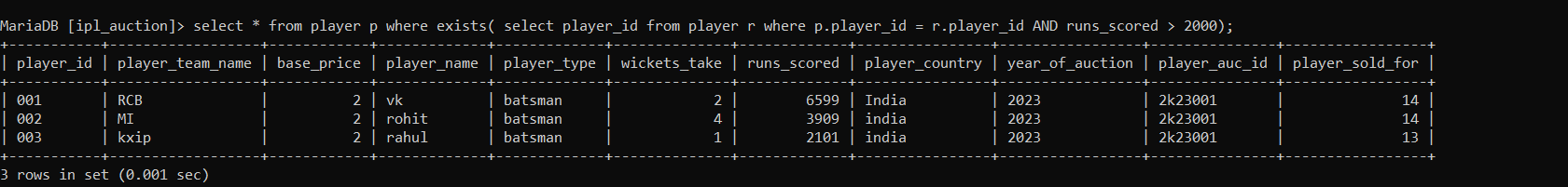
Output screenshot:



4. display player details of players who have scored more than 2000 runs.

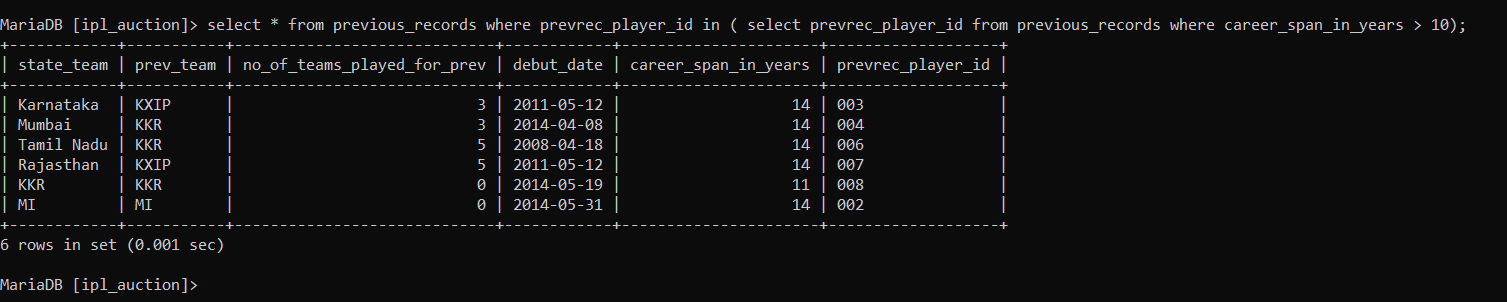
Query: select \* from player p where exists( select player\_id from player r where p.player\_id = r.player\_id AND runs\_scored > 2000);

Output screenshot:

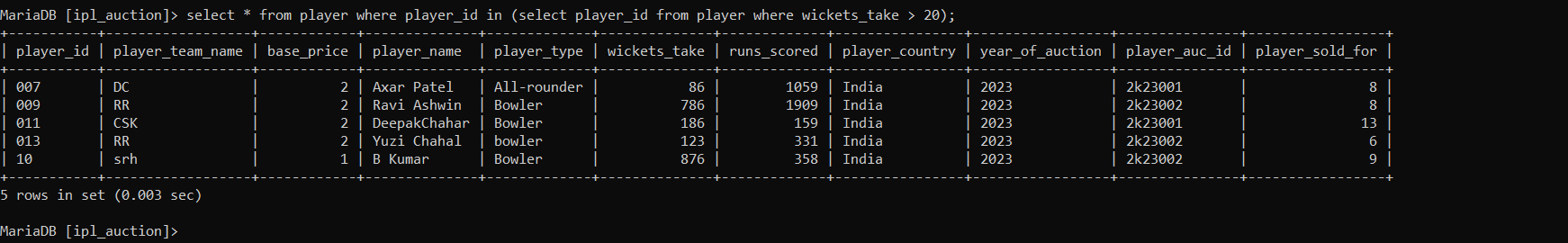


5. display those previous records who's career has lasted more than 10 years:

Query:



6. display those players details who have taken more than 20 wickets:

Query: select \* from player where player\_id in (select player\_id from player where wickets\_take > 20); 

**Aggregate Functions (at least 2)**

Showcase at least 2 Aggregate function queries. Write the query in English Language, Show the equivalent SQL statement and also screenshot of the query and the results

--The function is to display the amount of money spent an average of budget spent per team for each player.

Query and Aggregate funtions as a function:

delimiter $$

Create function get\_budget(T\_name varchar(255))

returns varchar(255)

deterministic

begin

declare temp int;

declare temp1 int;

select sum(player\_sold\_for) into temp

from player

where player\_team\_name= T\_name;

select avg(player\_sold\_for) into temp1 from player where player\_team\_name =T\_name;

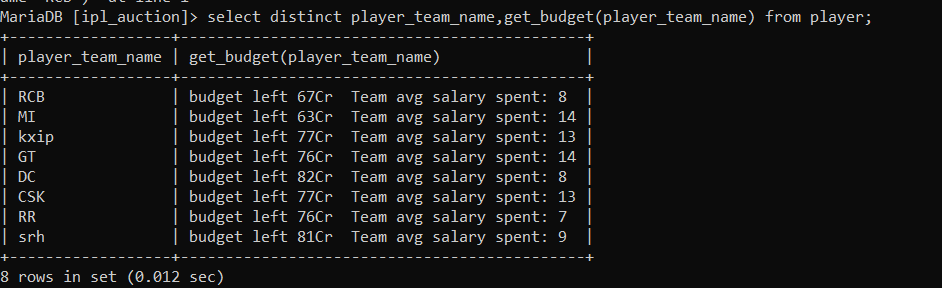
return(CONCAT("budget left ",(90-temp),"Cr"," Team avg salary spent: ",temp1));

end;

$$

delimiter ;

select distinct player\_team\_name,get\_budget(player\_team\_name) from player;



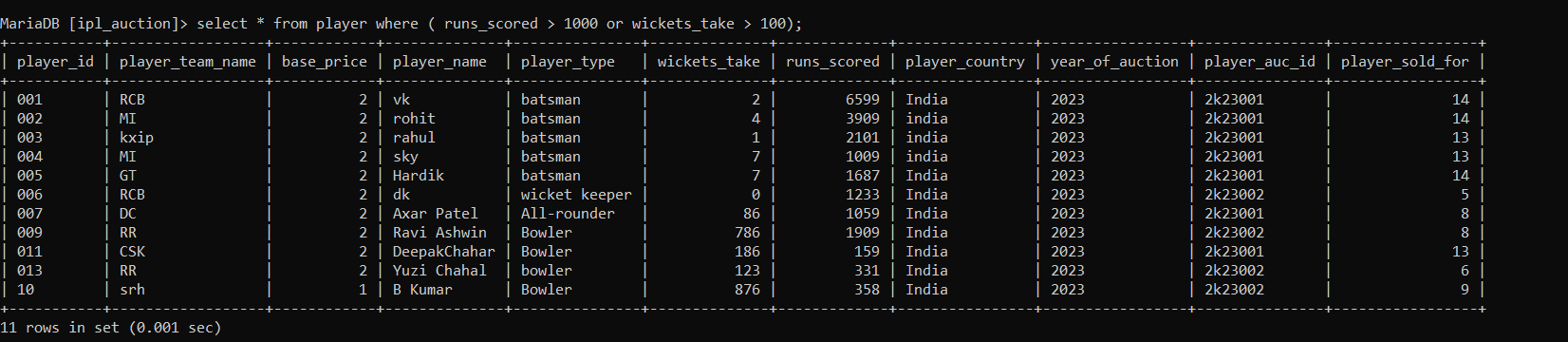
**Set Operations (at least 2)**

Showcase at least 2 Set Operations queries . Write the query in English Language, Show the equivalent SQL statement and also screenshot of the query and the results

1. Display those player details who either have 1000 runs or 100 wickets

Query: select \* from player where ( runs\_scored > 1000 or wickets\_take > 100);

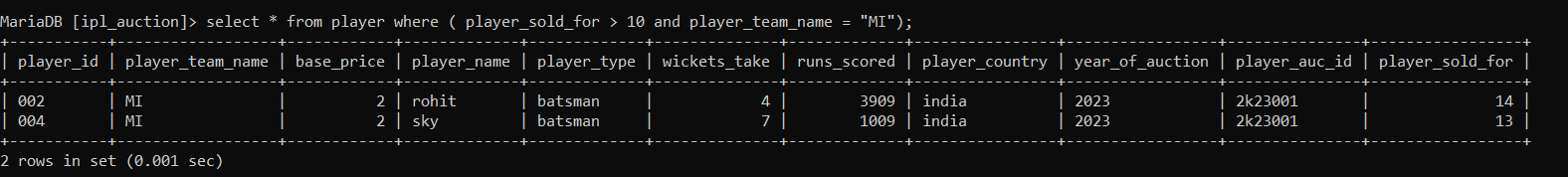
**Output screenshot:**

****

**2) display those I players who have been sold for more than 10cr:**

**Query: select \* from player where ( player\_sold\_for > 10 and player\_team\_name = "MI");**

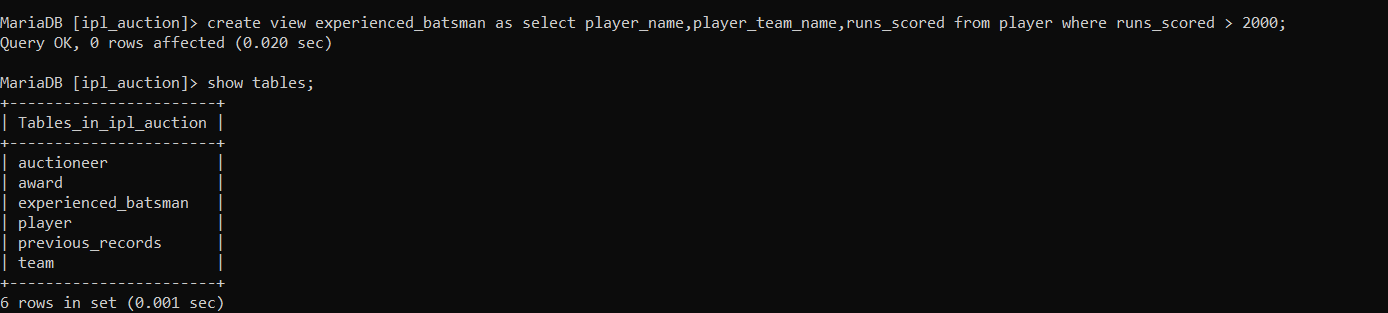
**Output screenshot:**

****

**View (atleast 1)**

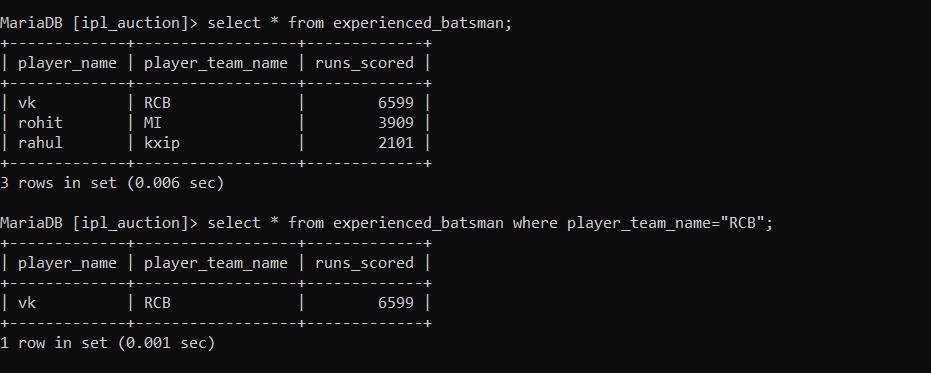
Creation:

Cmd: create view experienced\_batsman as select player\_name,player\_team\_name,runs\_scored from player where runs\_scored > 2000;



Create a query to find details of any experienced batsman who play for RCB

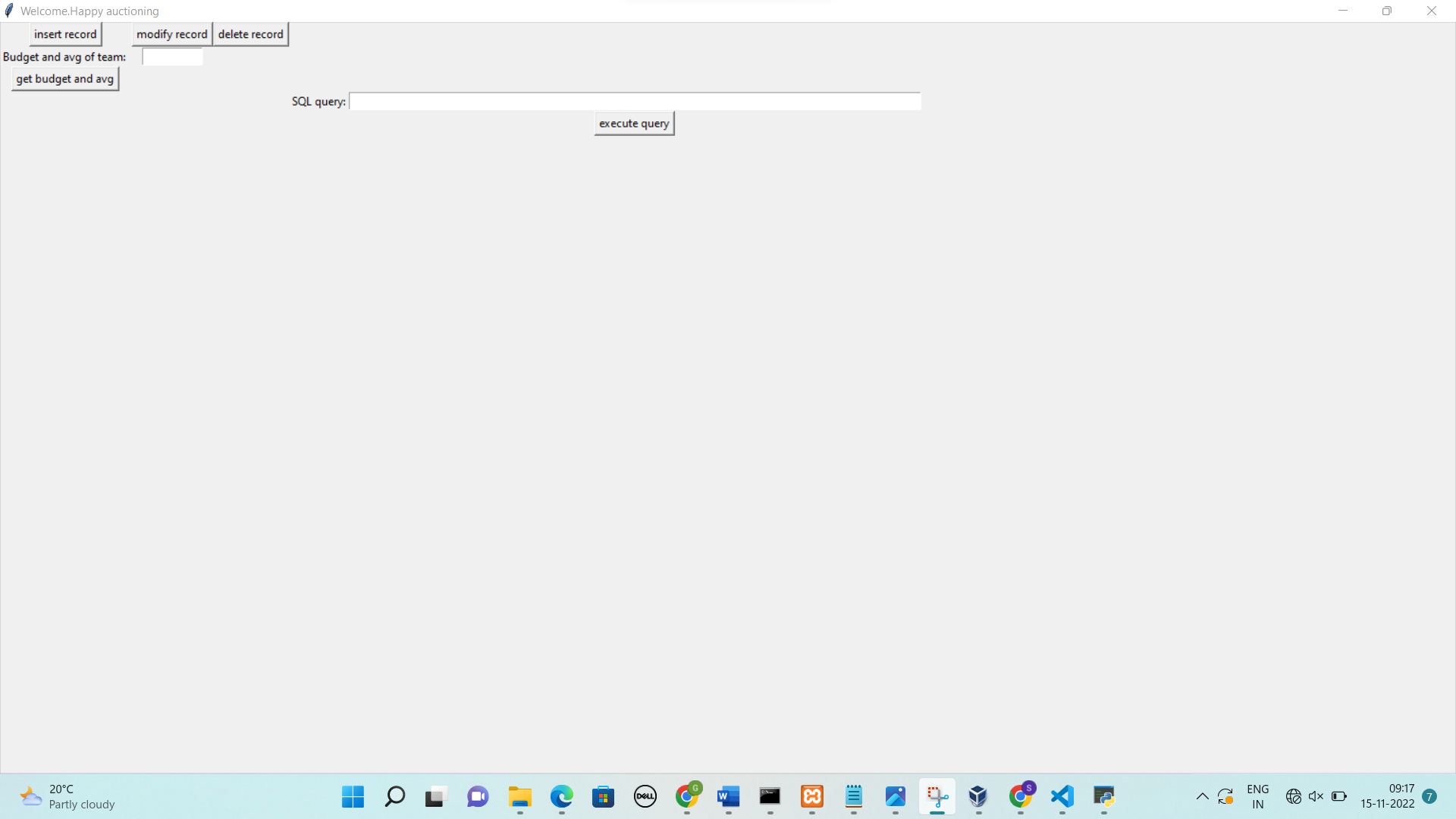
Query: select \* from experienced\_batsman where player\_team\_name="RCB";

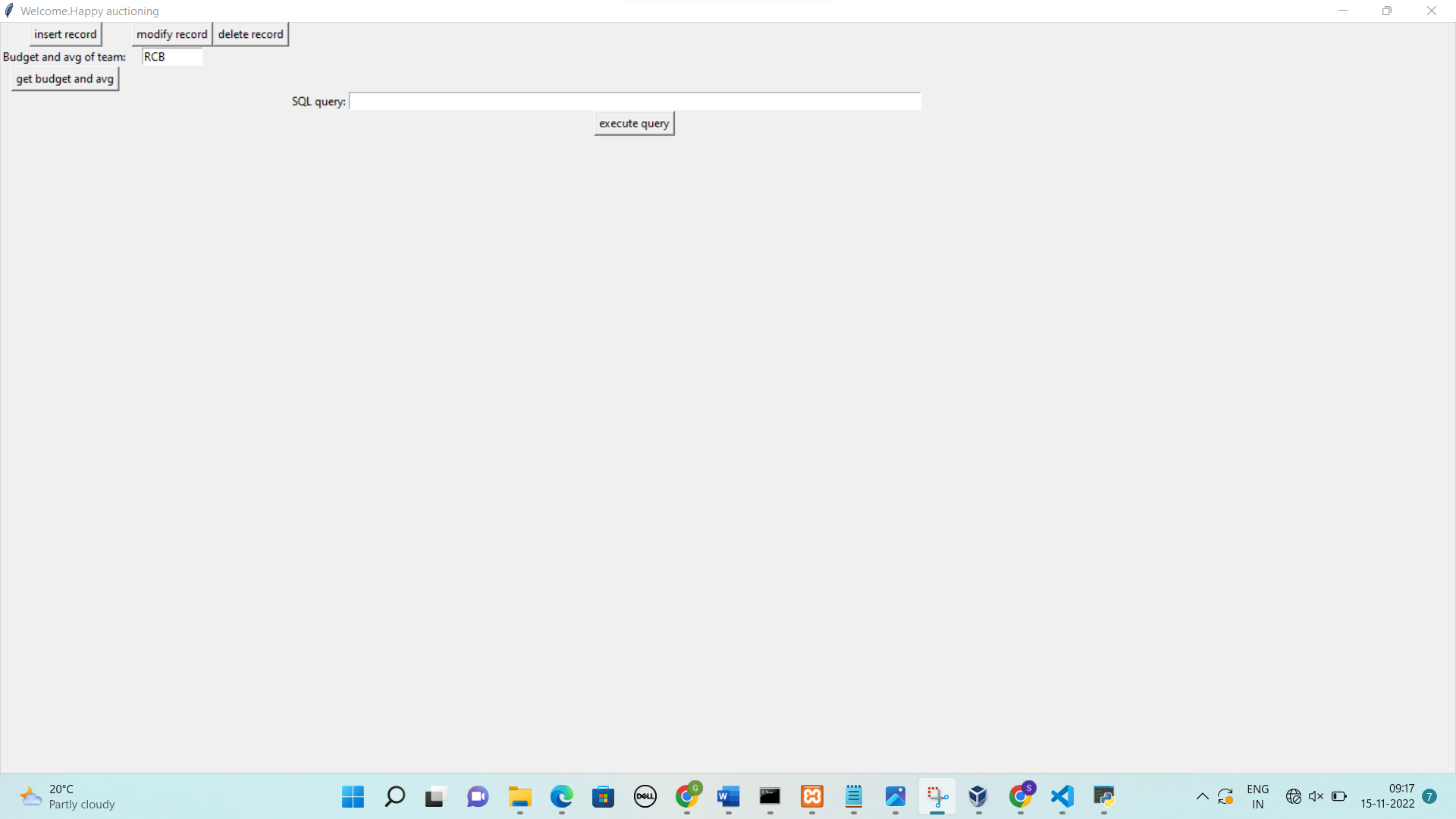
Output screenshot: 

**Triggers (Functions or Procedures)**

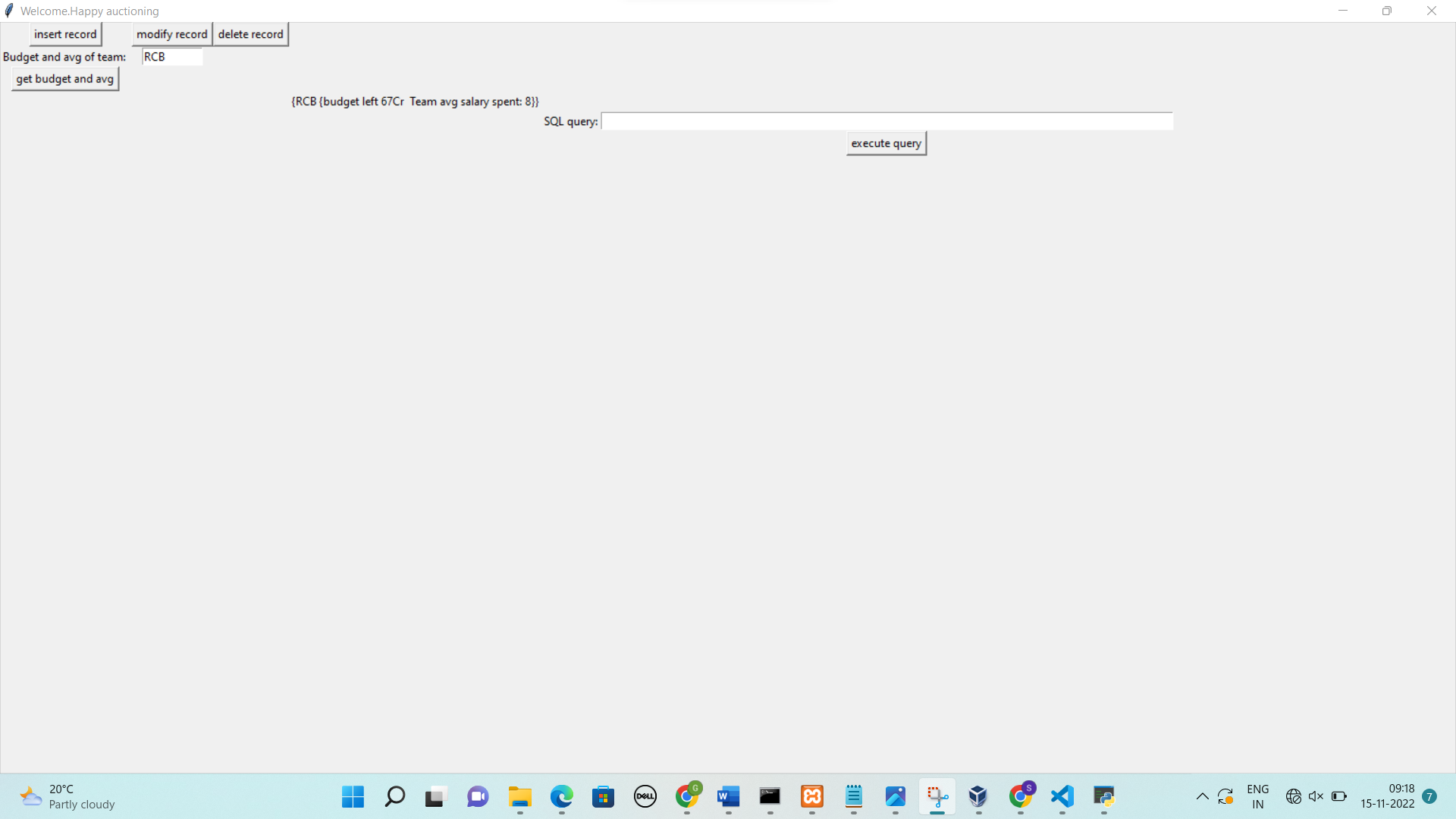
The function is called when the following button is clicked in the front end. It outputs your team’s budget left and average salary spent on a player.

Screenshots:

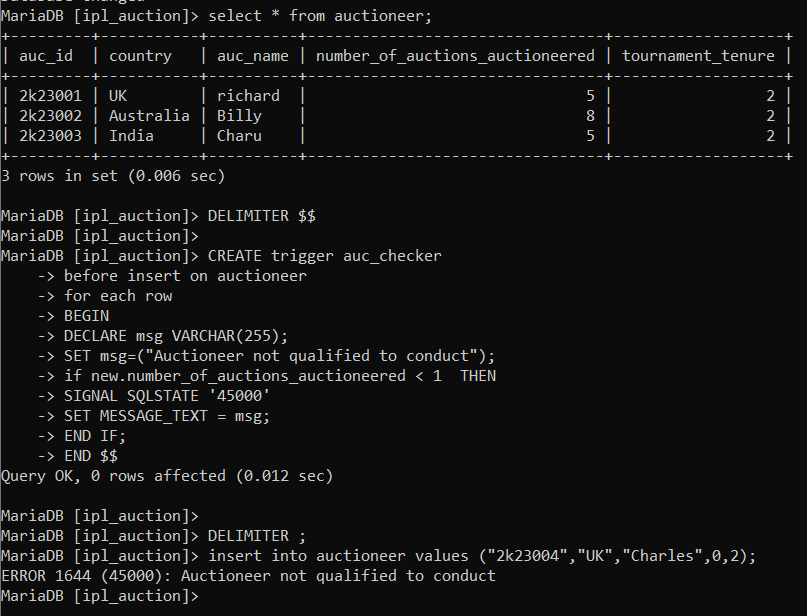




After you click on ( [get budget and avg]):



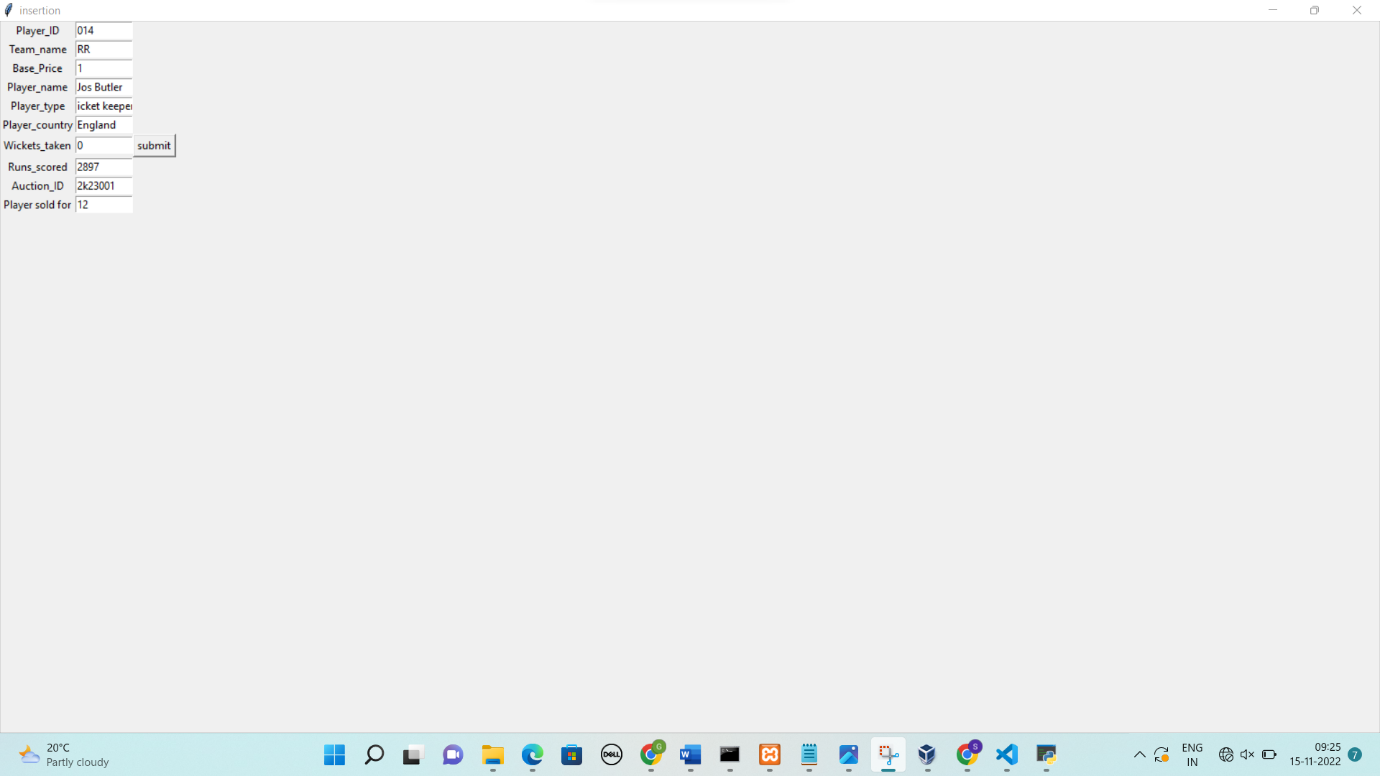
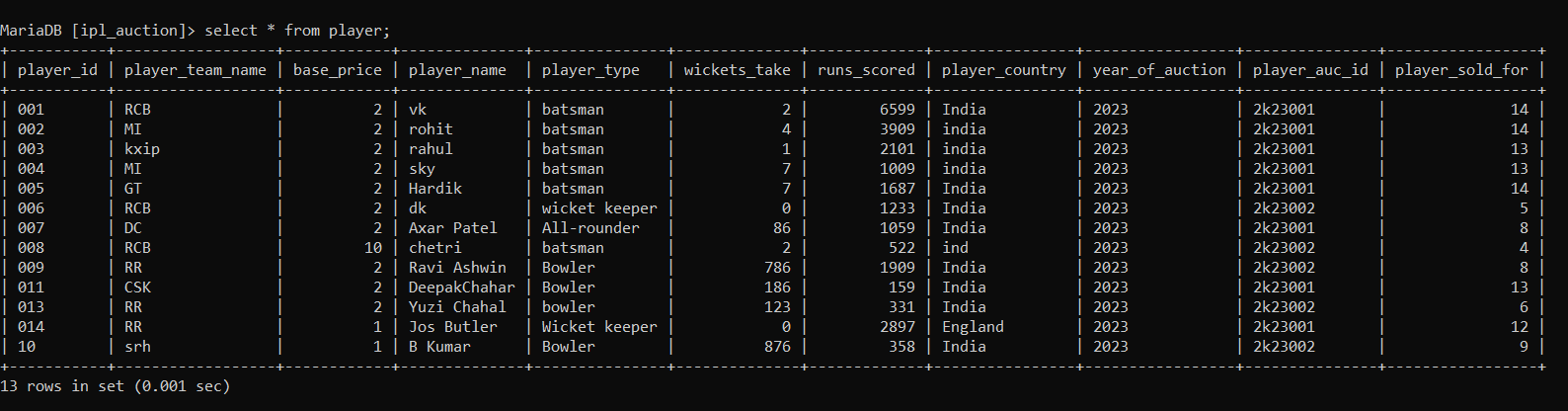
Trigger: auctioneer not qualified enough if he has no experience previously.

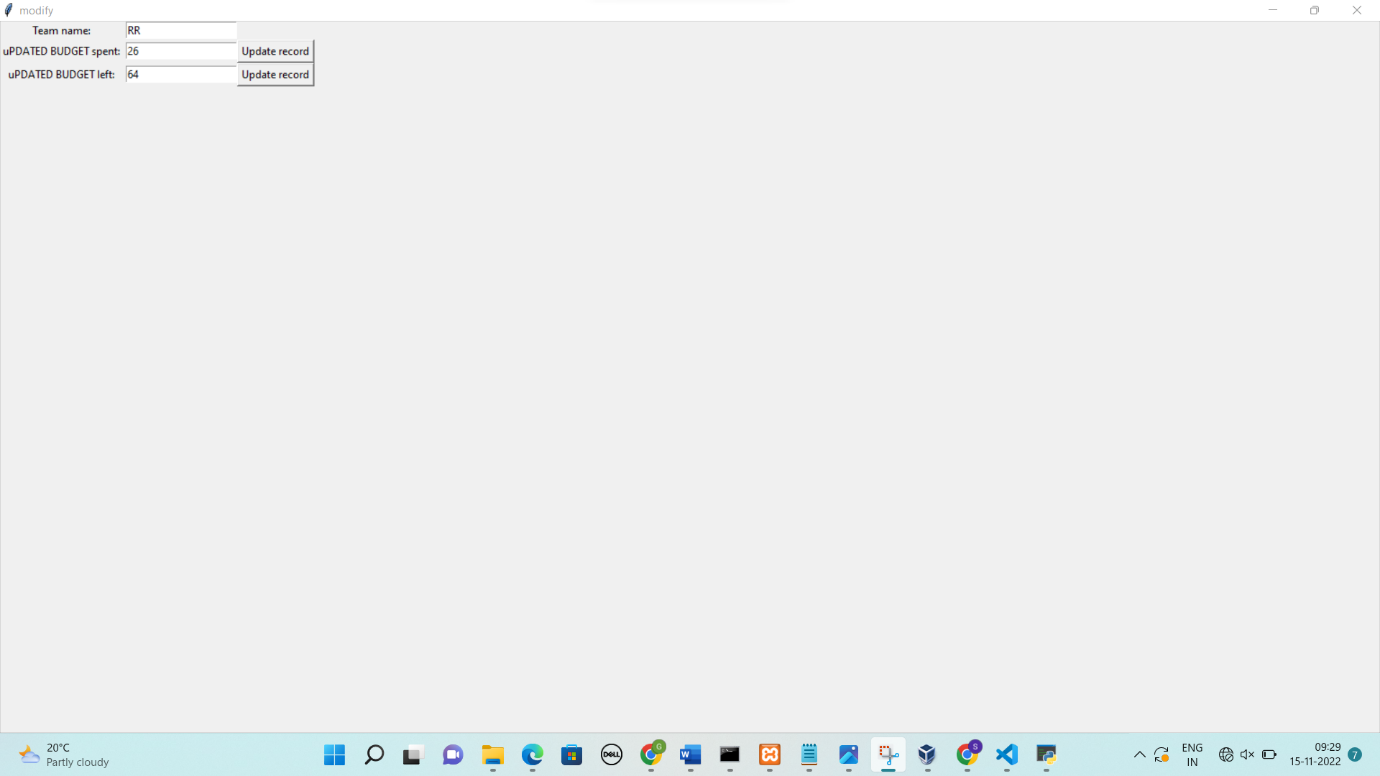


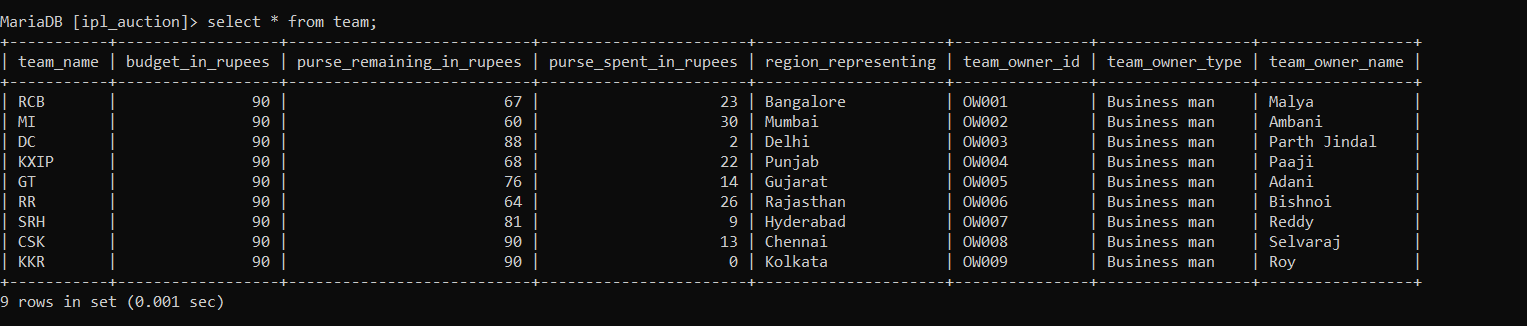
**Developing a Frontend**

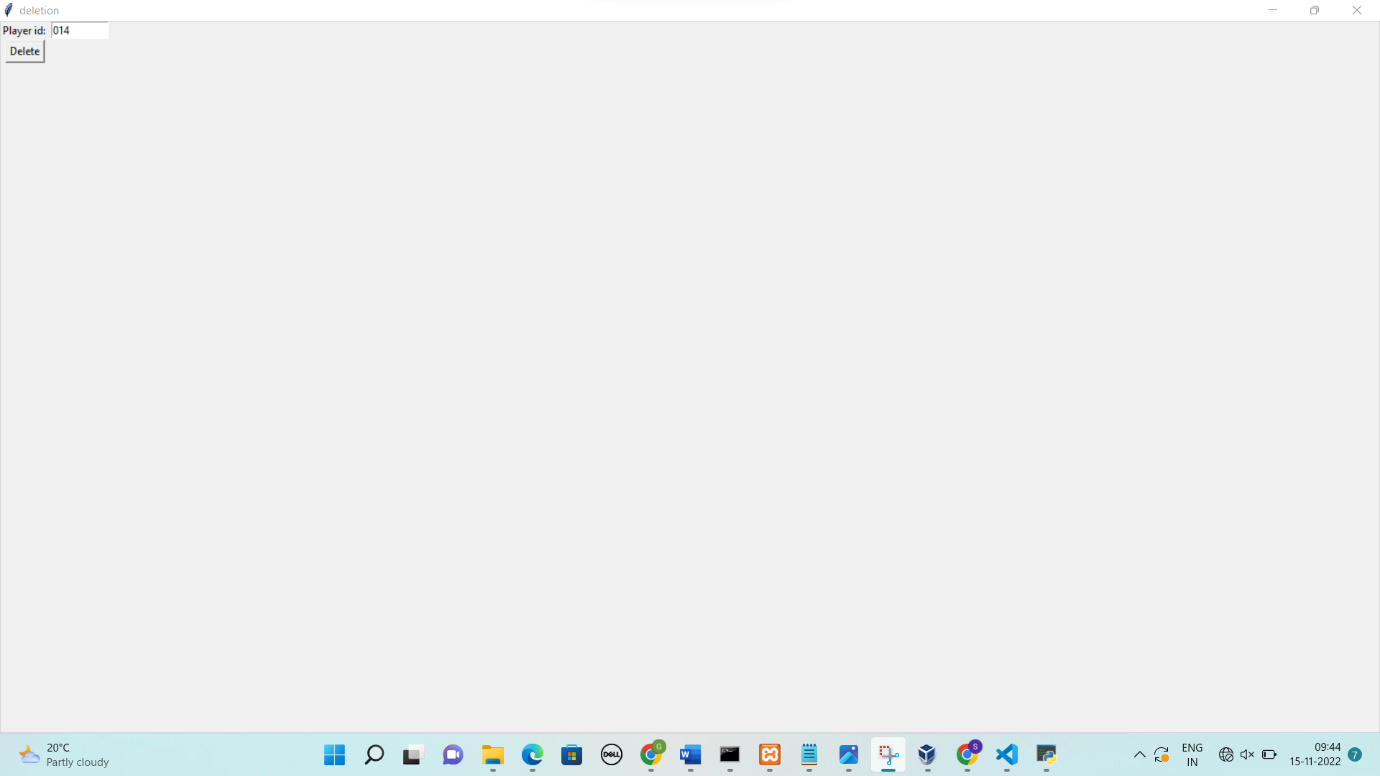
The frontend should support

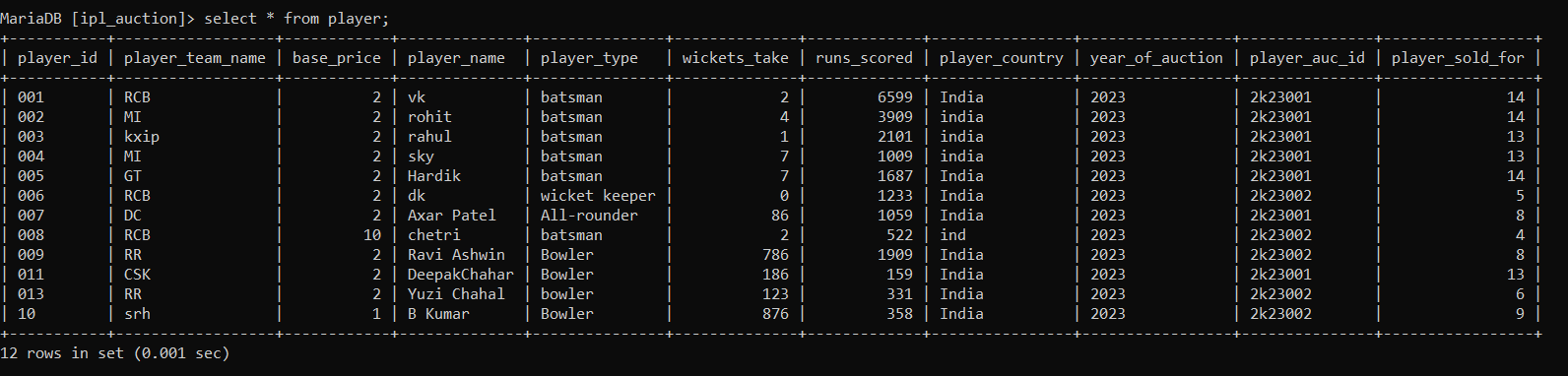
1. Addition, Modification and Deletion of records from any chosen table
   1. Addition:



* 1. Modify:



* 1. Delete:



1. There should be a window to accept and run any SQL statement and display the result

