DBMS - Mini Project

PESU SPORTS

MANAGEMENT SYSTEM

Submitted By:

Name : CHIRANTH R

SRN : PES1UG20CS116

V Semester Section :B

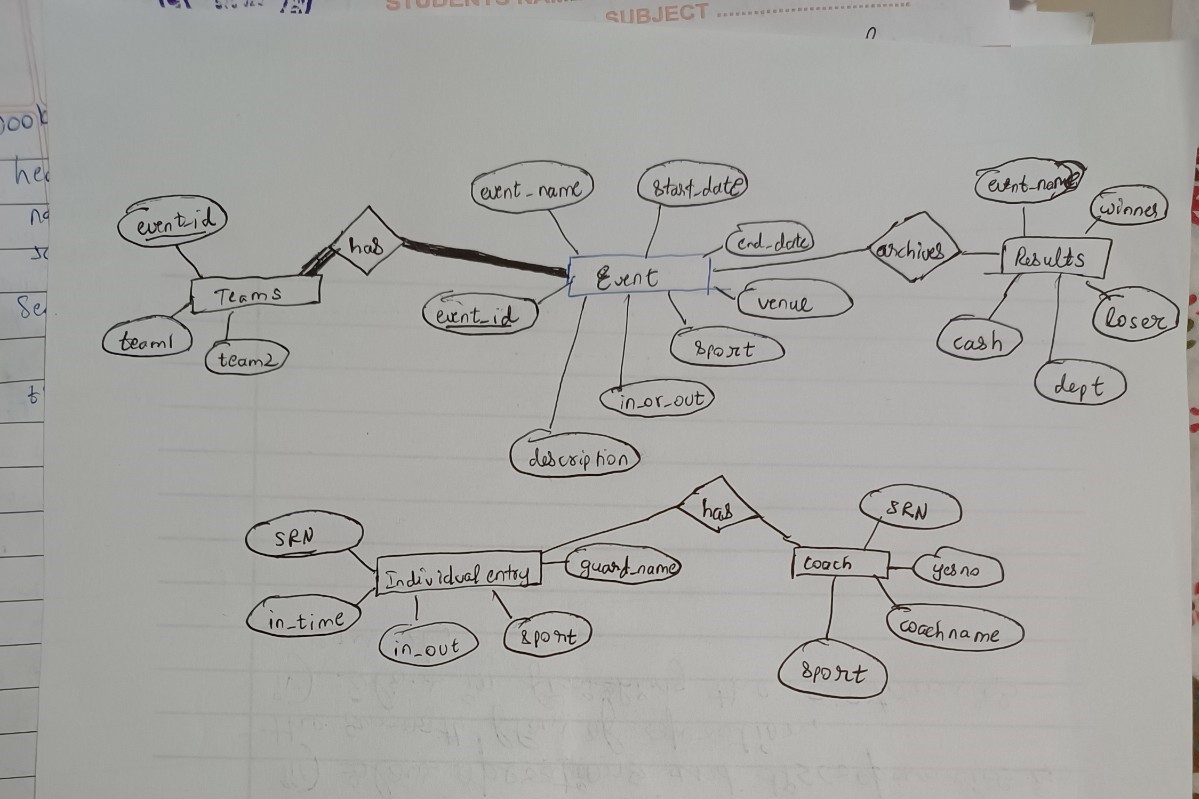
**Short Description and Scope of the Project**

* The main aim of the project is to manage sports events taking place in our university and also to manage entries at different venues for students.
* The events also comprise of teams who will need to be registered under that event
* Displaying result archives of different events
* Maintaining information about students who enter into a sports complex and also details of coach if they have a coach under them

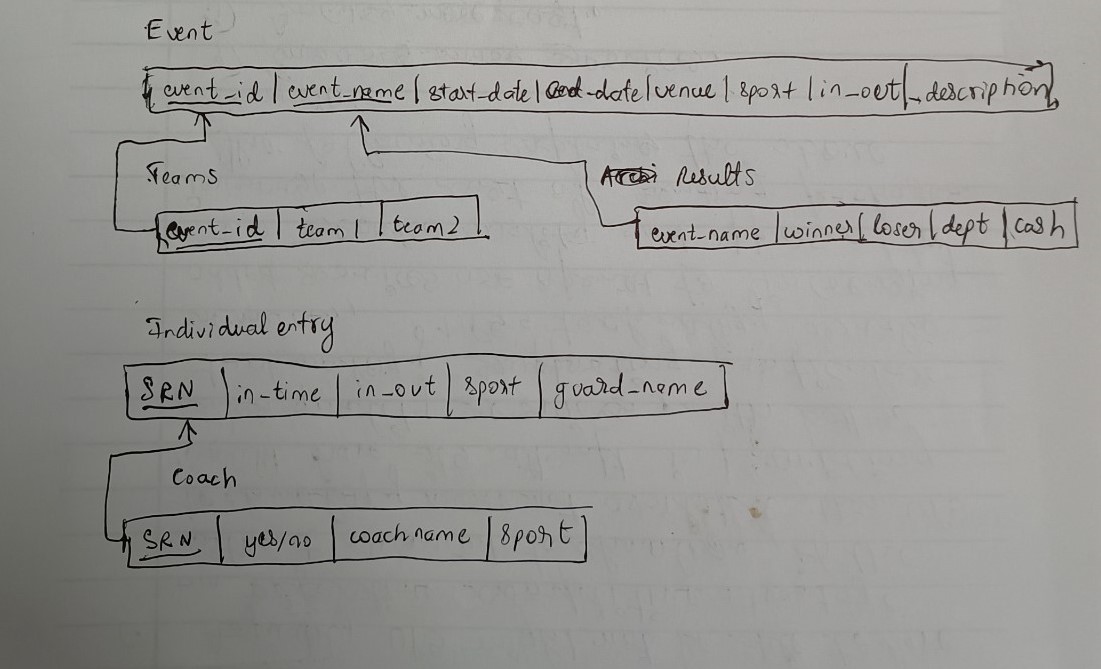
Scope:

The scope of the project is to make the sports event management of our university more efficient by proper tracking of different venues ,the dates of event, the registration of teams and so on

**ER Diagram**

****

**Relational Schema**

****

**DDL statements - Building the database**

create database pesu\_sports;

create table event(event\_name TEXT,start\_date TEXT,end\_date TEXT,venue TEXT,sport TEXT,in\_or\_out TEXT,description TEXT,event\_id varchar(20) NOT NULL,PRIMARY KEY(event\_id));

create table enter\_team(event\_id varchar(10) NOT NULL,team1 varchar(50),team2 varchar(50),FOREIGN KEY(event\_id) REFERENCES event(event\_id) ON DELETE CASCADE);

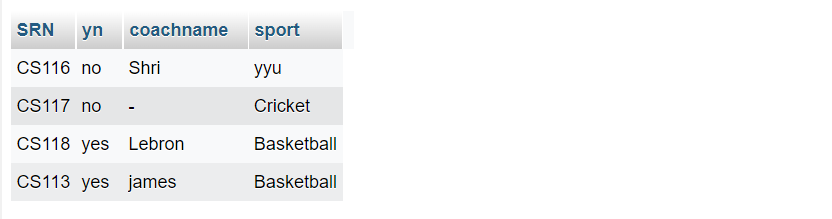
create table results(event\_name TEXT,winner TEXT,loser TEXT,dept TEXT,cash TEXT);

create table entry(SRN char(13) NOT NULL,in\_time datetime,in\_out varchar(5),sport varchar(20),guard\_name varchar(100),PRIMARY KEY(SRN));

create table coach(SRN char(13),yn varchar(3),coachname varchar(50),sport varchar(50));

**Populating the Database**

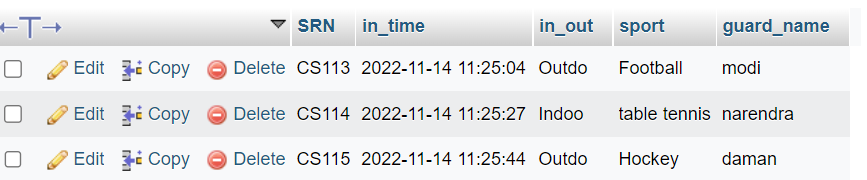
**Coach:**



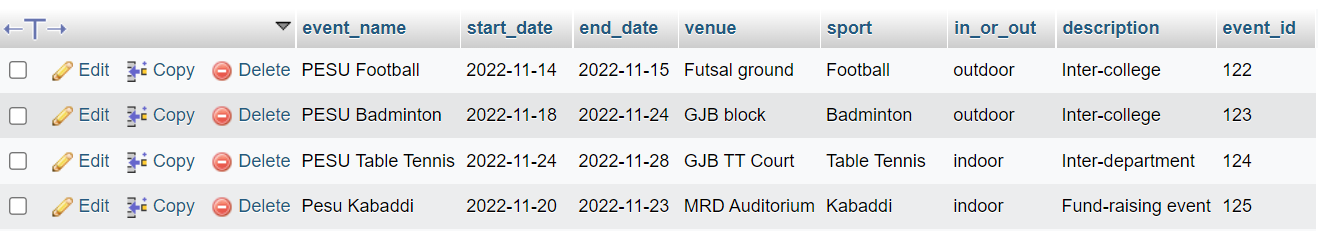
**Team:**



**Entry:**



**Event:**



**Results:**

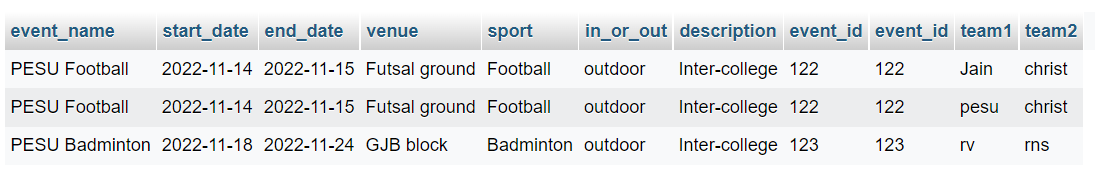


**Join Queries**

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

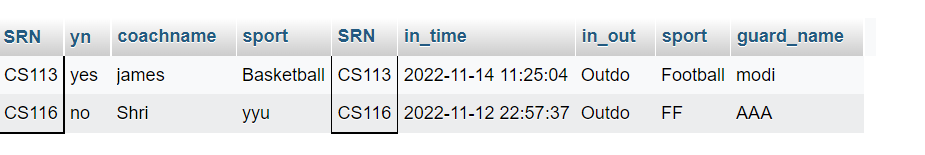
1. To display team details with event details

select \* from event inner join enter\_team on event.event\_id=enter\_team.event\_id;



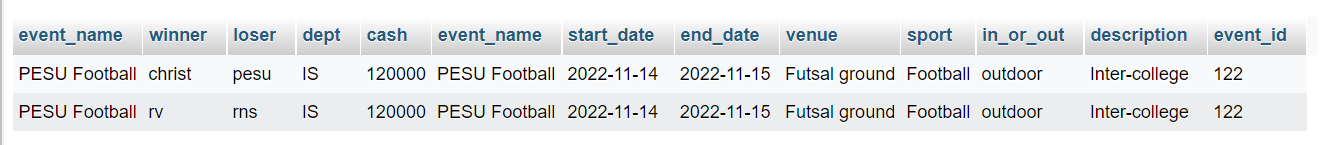
1. Display coach and student details

SELECT \* from coach join entry WHERE coach.SRN=entry.SRN;



1. Display team details with event

SELECT \* FROM results join event on event.event\_name=results.event\_name;



**Aggregate Functions**

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1. To count number of teams participating in each event

SELECT COUNT(\*),event\_id from enter\_team GROUP BY event\_id;



1. To count number of coaches for each student

SELECT SRN,COUNT(\*) FROM `coach` GROUP BY SRN;



1. To display the cash amount given till now

SELECT SUM(cash) FROM results;



1. To display number of students playing a particular sport

SELECT sport,COUNT(\*) FROM entry GROUP BY sport;



**Set Operations**

Creating another table(team2) which will have team details which are yet to confirm for a event

1. Union – Doing union operation of enter\_team and team2

SELECT \* FROM `team2` UNION SELECT \* FROM enter\_team;



1. Intersection – Same as above tables,common data for both tables

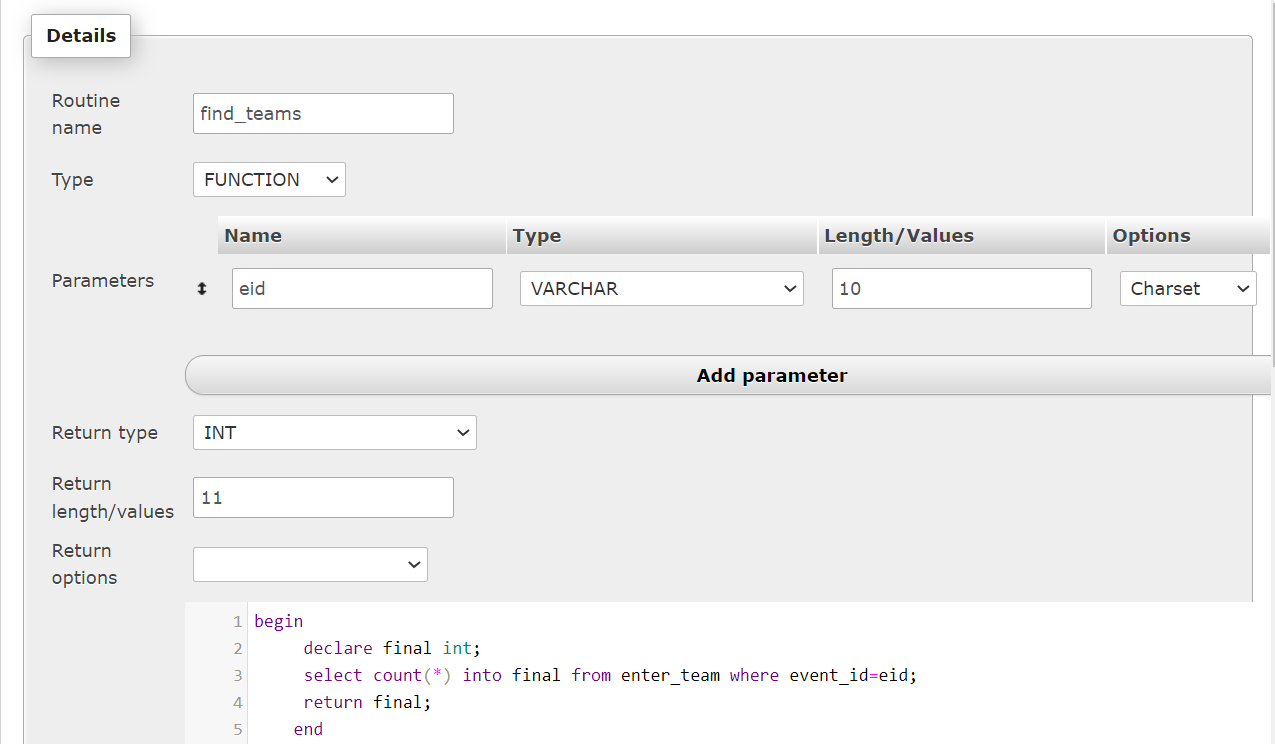
SELECT \* FROM `team2` INTERSECT SELECT \* FROM enter\_team;



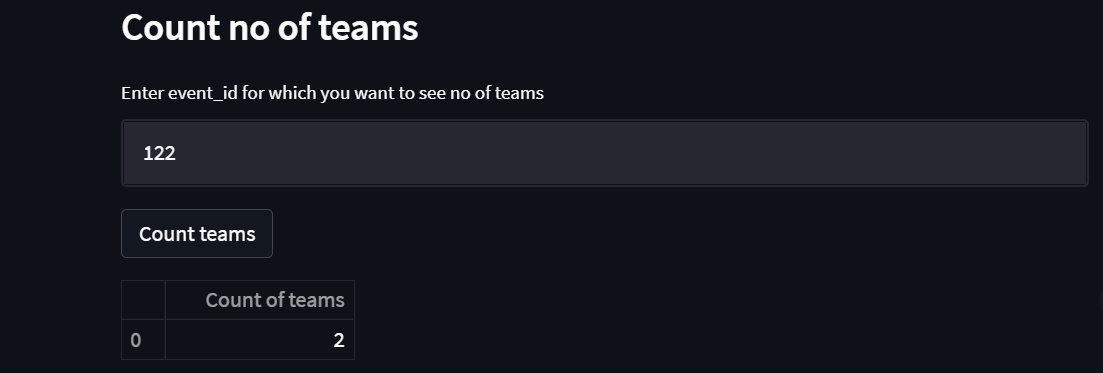
**Functions and Procedures**

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

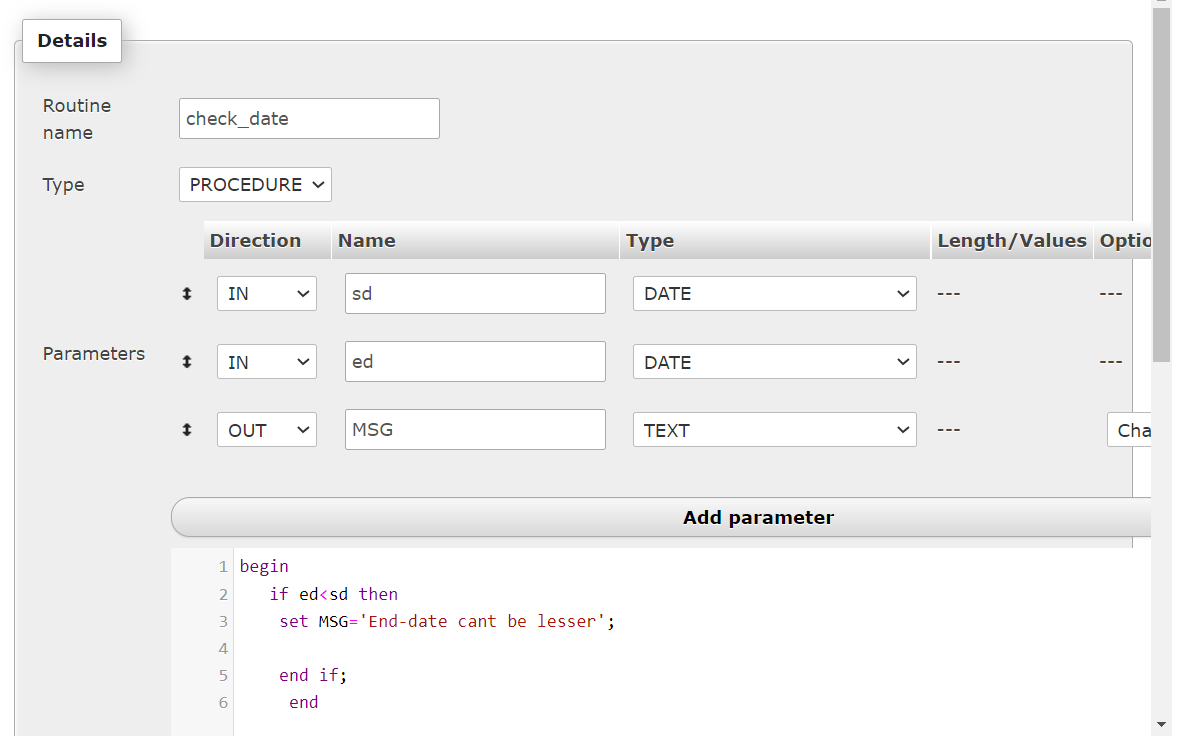
1. **Creating a function which will takein the event\_id and display the No of teams playing in that event.**



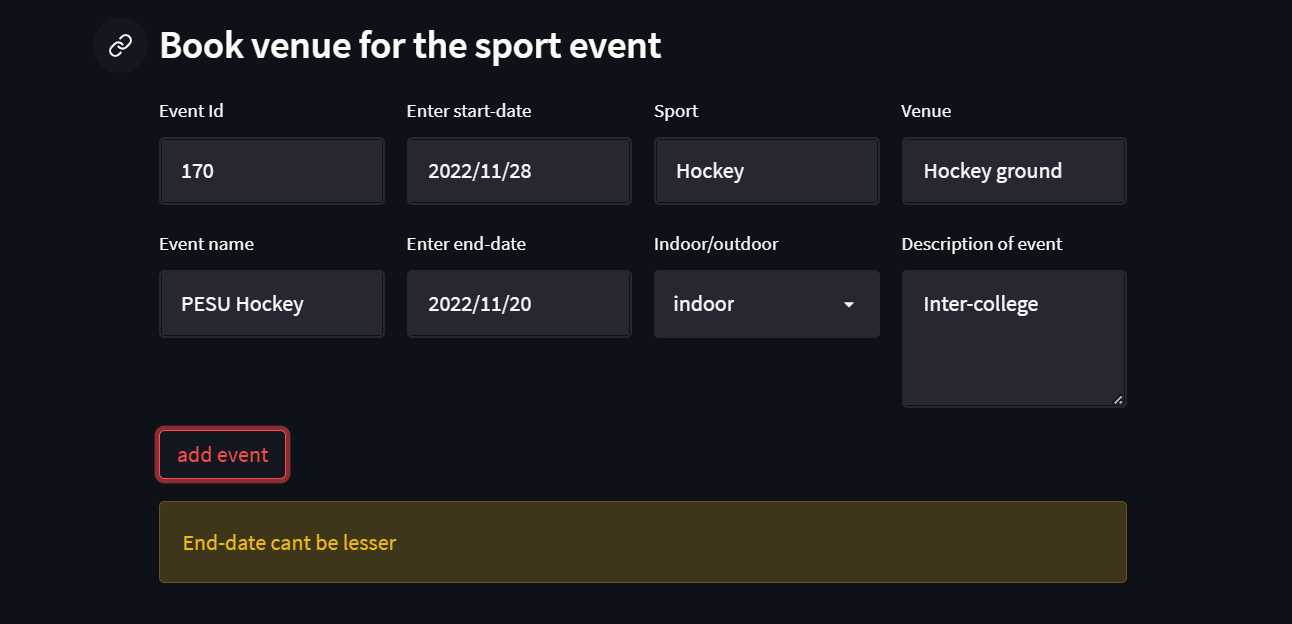
Result(shown in frontend):



1. **Creating a Procedure which will validate the dates(the end date shouldn’t be lesser than the start date) which will take start and end dates as input and ouputs a MSG**



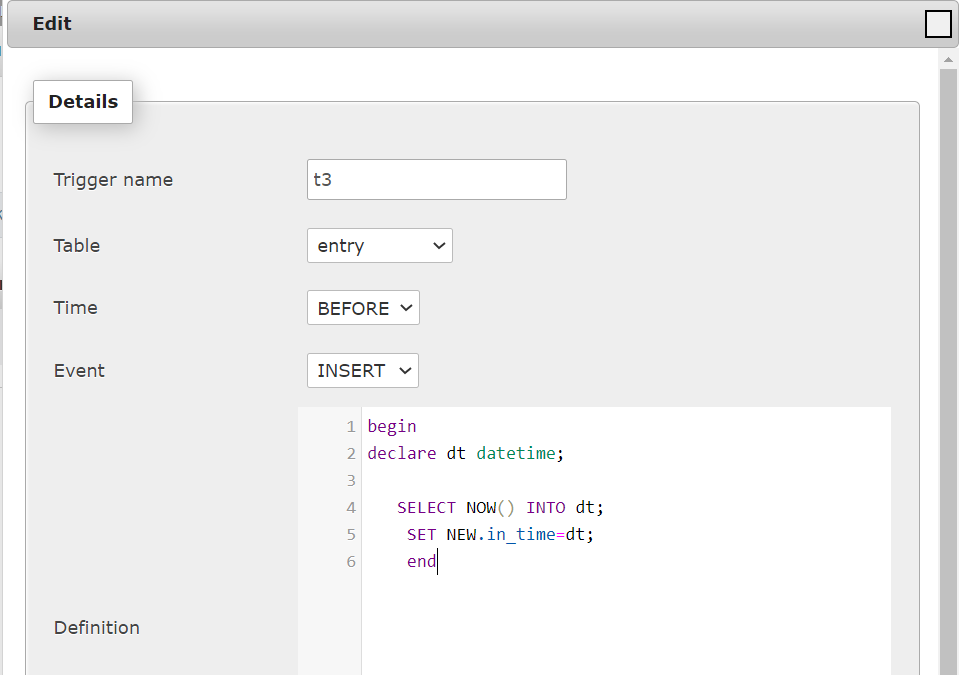
Result(Shown in frontend itself):



**Triggers and Cursors**

Create a Trigger and a Cursor. State the objective. Run and display the results.

**A trigger t3 was created on the entry table in which as soon as the student enter details there will be no manual entry of datetime but a automatic one due to this trigger**



Result:

After doing this:

INSERT INTO `entry`(`SRN`, `in\_out`, `sport`, `guard\_name`) VALUES ('CS125','INDOOR','Cricket','Shyam');

The curent date and time is automatically added



**A cursor is normally used in the python scripting which is needed to handle all the CRUD operations:**



**Developing a Frontend**

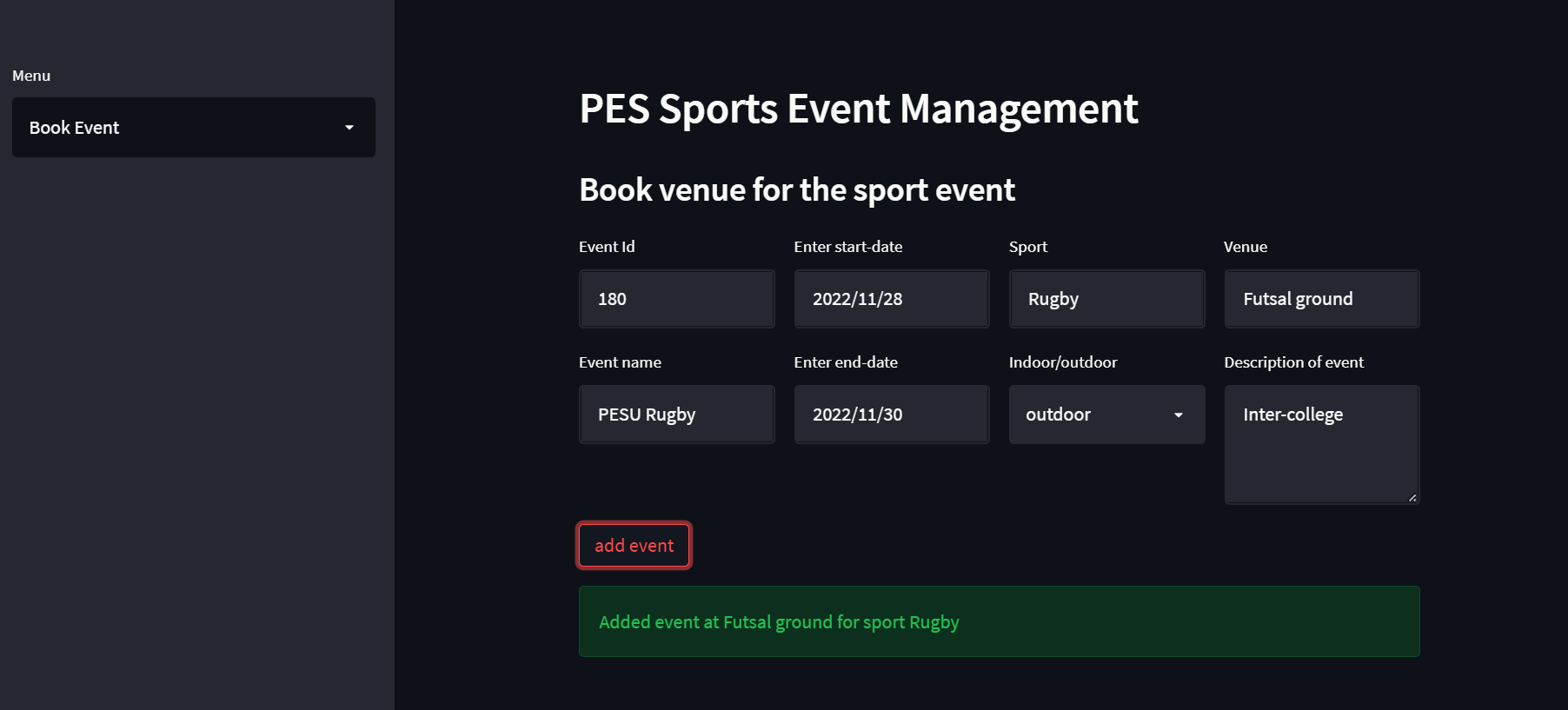
The frontend should support

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

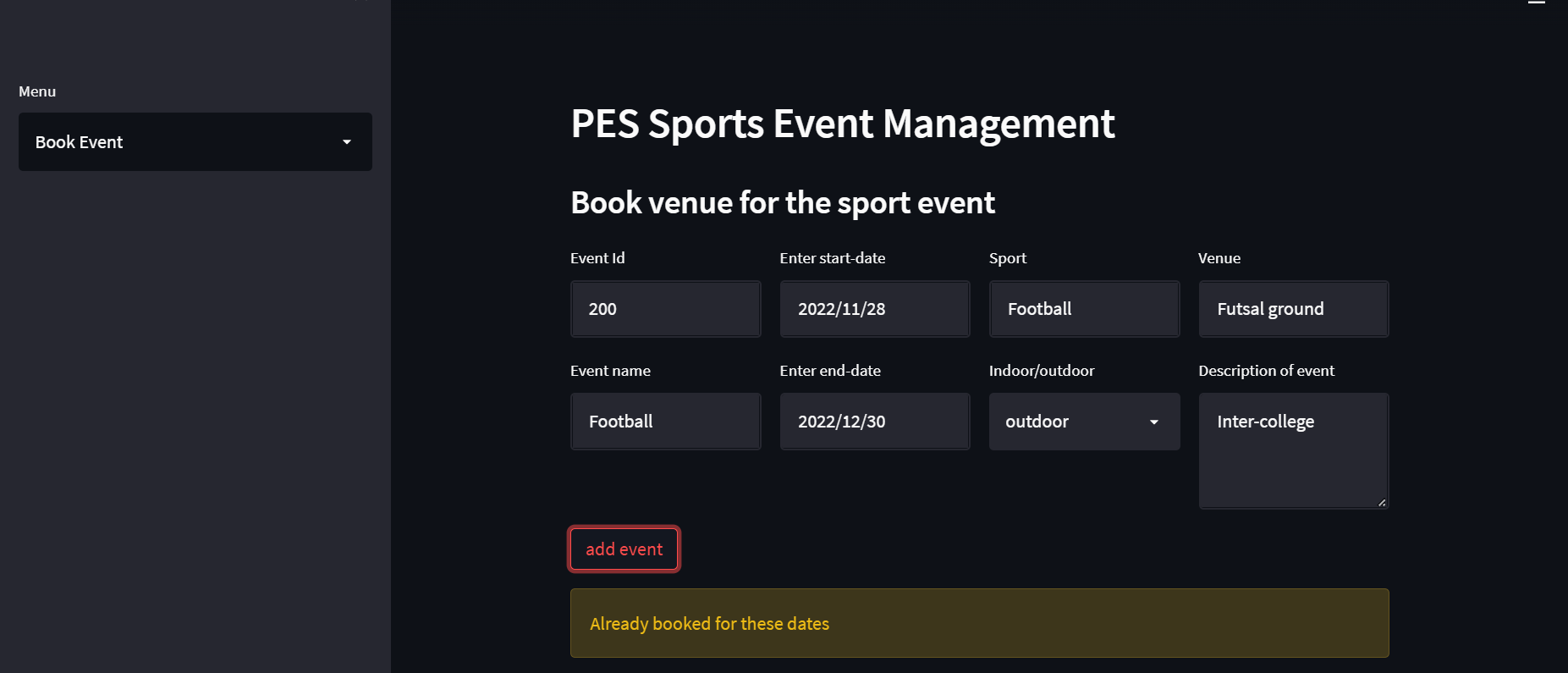
2. There should be an window to accept and run any SQL statement and display the result

***CRUD Operations shown are done on the event table***

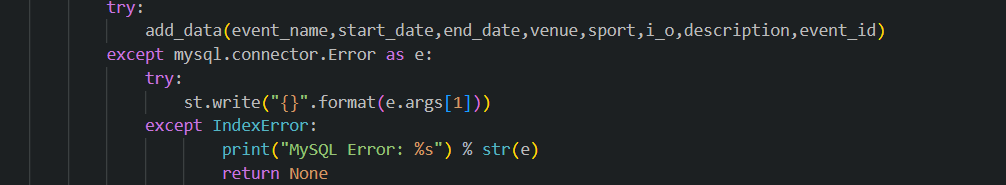
Create/Add:

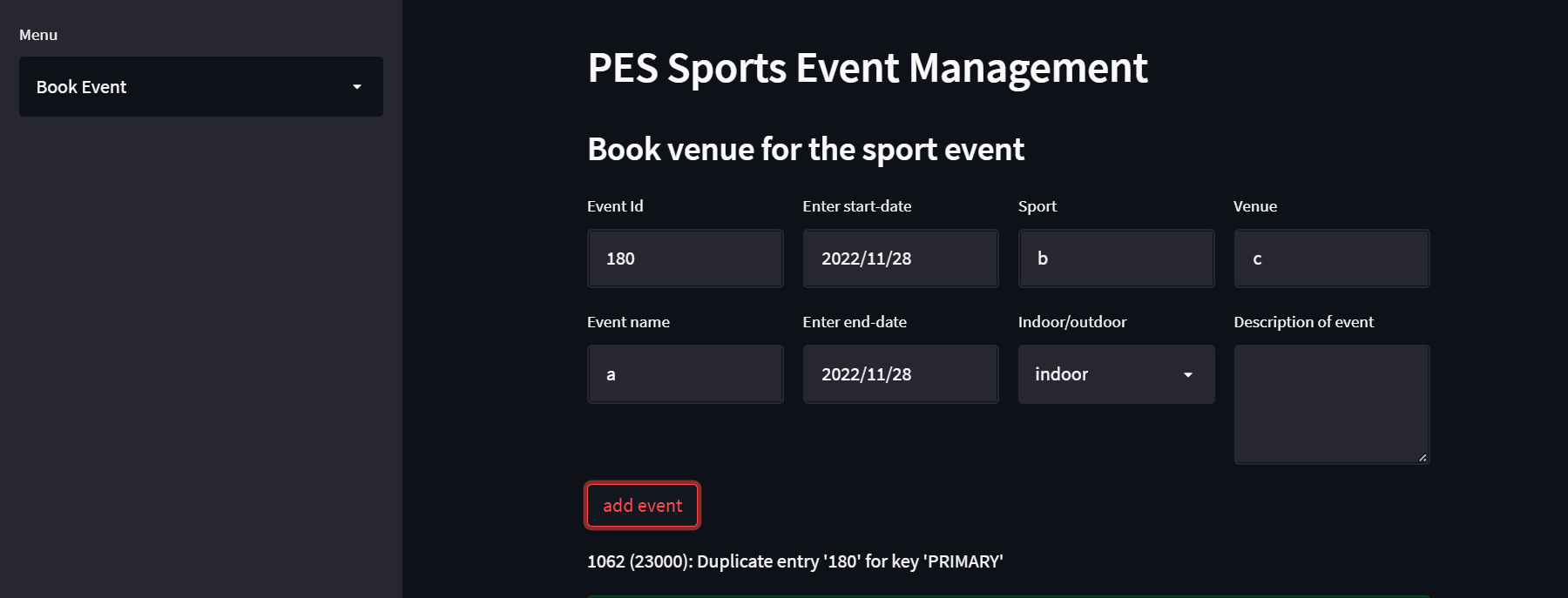


If at the same venue we check the dates and display a ‘booked message’:



Handling error conditions and displaying message:





Read:

Shown 3 tables for example

Event:

Entry:



Team:

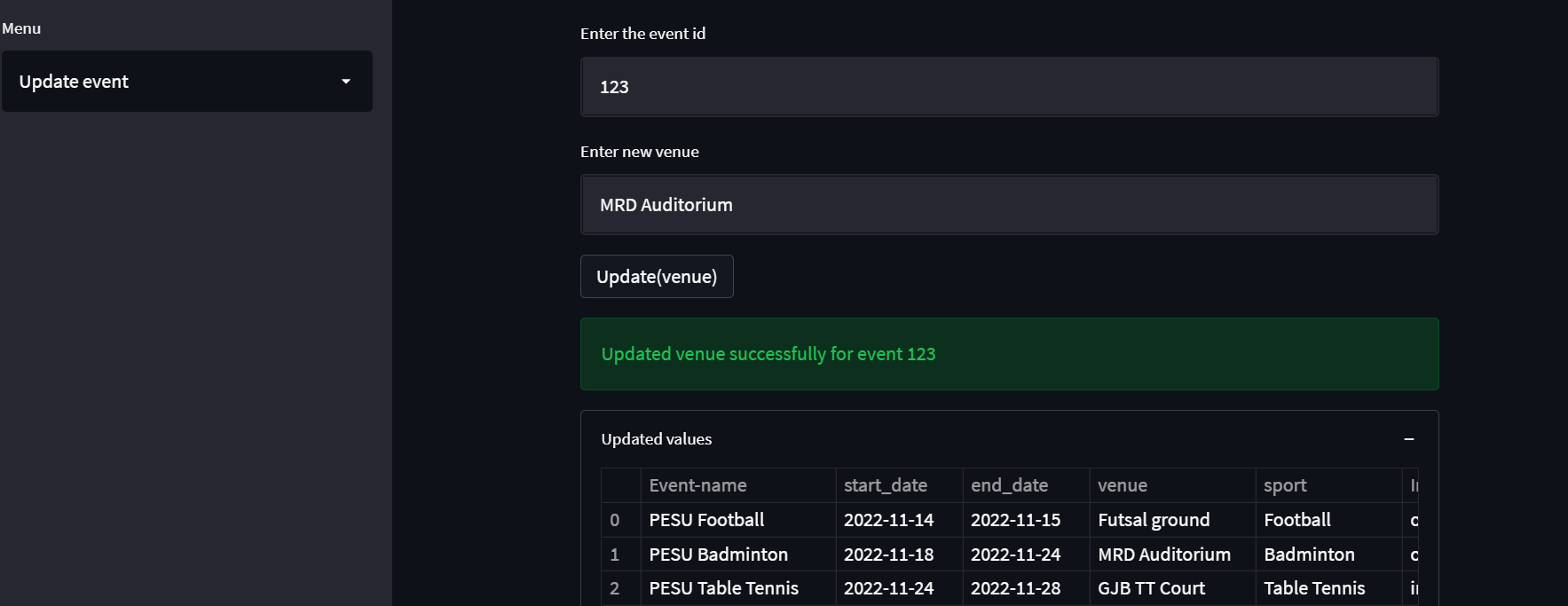


Update:

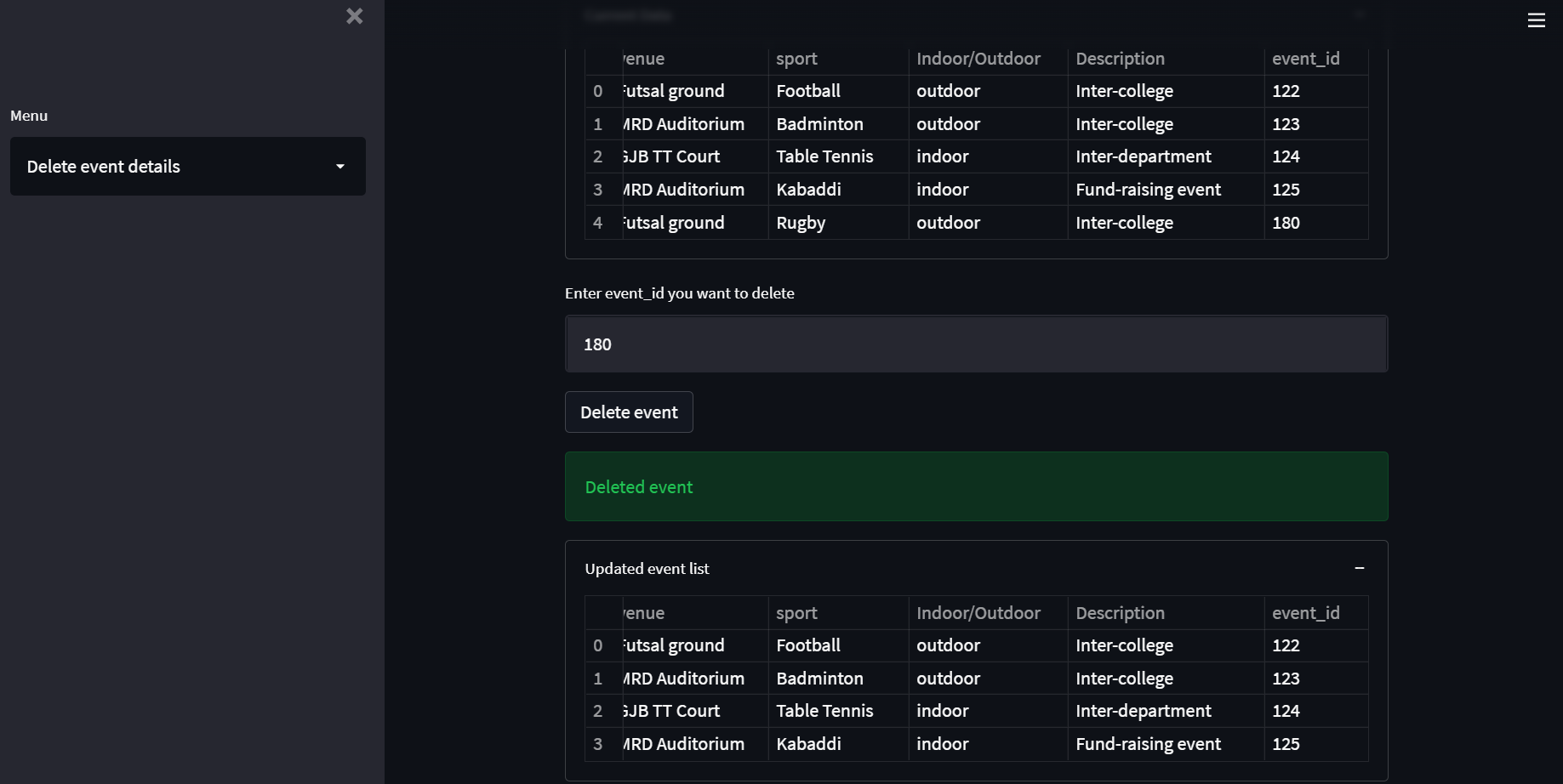
Before updating venue of PESU Badminton



After updating:



Delete:



Showing SQL Queries via Front-end:





