**Assignment 1**.

**Q1[total 2: (-1 point for every mistake)]** Consider a table **Payment**. It has the following fields: **payment\_id**, **customer\_id**, **staff\_id**, **amount**, **payment\_date**. The **payment\_id** field identifies the tuple. The **customer\_id** and **staff\_id** fields are foreign keys to the **Customer** and **Staff** table, respectively. It tells us which customer made the payment and which staff member processed the payment. The last field denotes the date when the payment was made. Obviously, a customer can make several payments on the same date. Complete the following picture by putting checkmarks where appropriate. The picture is for the **Payment** table. For example, there is a checkmark for **payment\_id** and **Primary key** because **payment\_id** is the primary key for the **Payment** table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Primary key | Candidate Key | Super key |
| payment\_id | X | X | X |
| customer\_id |  |  |  |
| payment\_id,customer\_id |  |  | X |
| amount, payment\_date |  |  |  |
| payment\_id,staff\_id |  |  | X |

**Q2[total 8: 2 per table]** Suppose you want to record information about soccer players, their teams, the games where they played, and how many goals they scored in each game. Create a **SoccerPlayer** table that stores information about the soccer players. Create a **Game** table that stores information about the games. Every game is between two teams. Record the number of goals scored by each team. Create a **Team** table that stores information about all the teams. Every soccer player belongs to exactly one team (i.e., add a **team** field to the **SoccerPlayer** table). Lastly, create a **Participate** table that shows which player played in which game and how many goals they scored. If a player didn’t score, then 0 will appear for **goalsScored**.

Submit the **create table** statements for the four tables. Include primary key and foreign key constraints.

CREATE TABLE Team (

name VARCHAR(45) NOT NULL,

city VARCHAR(45) NOT NULL,

PRIMARY KEY (name) );

CREATE TABLE SoccerPlayer (

id INT NOT NULL,

name VARCHAR(45) NOT NULL,

teamName VARCHAR(45) NOT NULL,

PRIMARY KEY (id,teamName),

FOREIGN KEY (teamName)

REFERENCES team(name));

CREATE TABLE Game (

id INT AUTO\_INCREMENT,

homeTeamName VARCHAR(45) NOT NULL,

awayTeamName VARCHAR(45) NOT NULL,

homeGoals INT NOT NULL,

awayGoals INT NOT NULL,

date DATE NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (homeTeamName) REFERENCES team(name),

FOREIGN KEY (awayTeamName) REFERENCES team(name));

CREATE TABLE Participate (

id INT AUTO\_INCREMENT,

soccerPlayerID INT NOT NULL,

soccerPlayerTeamName VARCHAR(45) NOT NULL,

gameID INT NOT NULL,

goals INT NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (soccerPlayerID,soccerPlayerTeamName) REFERENCES soccerplayer(id,teamName),

FOREIGN KEY (gameID) REFERENCES game(id));

**Q3[4: 1 per table]** Show a sequence of **insert into** statements that populate all four tables. The order of the statements matters!

INSERT INTO team (name, city) VALUES ('Montreal Impact', 'Montreal');

INSERT INTO team (name, city) VALUES ('Toronto FC', 'Toronto');

INSERT INTO team (name, city) VALUES ('LA Galaxy', 'Los Angeles');

INSERT INTO team (name, city) VALUES ('Vancouver Whitecaps FC', 'Vancouver');

INSERT INTO team (name, city) VALUES ('New York Red Bulls', 'New York');

INSERT INTO soccerplayer VALUES ('1', 'Bob Kick', 'Toronto FC');

INSERT INTO soccerplayer VALUES ('1', 'Tim Howard', 'LA Galaxy');

INSERT INTO soccerplayer VALUES ('9', 'Clint Dempsey', 'Vancouver Whitecaps FC');

INSERT INTO soccerplayer VALUES ('9', 'David Beckham', 'Montreal Impact');

INSERT INTO game(homeTeamName,awayTeamName,homeGoals,awayGoals,date) VALUES ('Montreal Impact', 'LA Galaxy', '4', '1', '2012/1/1');

INSERT INTO game(homeTeamName,awayTeamName,homeGoals,awayGoals,date) VALUES ('Toronto FC', 'New York Red Bulls', '3', '2', '2013/4/4');

INSERT INTO participate (`soccerPlayerID`, `soccerPlayerTeamName`, `gameID`, `goals`) VALUES ('9', 'Montreal Impact', '2', '3');

**Q4[total 8: 2 per query]** Create the tables in MySQL and populate with random data. Then write, test, and submit the SQL for the following queries. Add the **distinct** keyword if you want to eliminate duplicates.

1. Print the names of soccer players that have scored a hat-trick (3 or more goals per game).
2. Print the names of teams that have one or more players that have scored a hat-trick.
3. Print the names of teams that have scored more than 3 goals in a single game.
4. Suppose that ``Cal Poly” is one of the teams. Print the names of all teams that Cal Poly has beaten. Cal Poly has beaten an opponent if we have scored more goals than the opponent in a single game.

Submit via PolyLearn. Due Date October 13th before start of class. I will solve the assignment in class, so no late assignments will be accepted.

select distinct s.name

from soccerPlayer s, participate p

where s.id = p.soccerPlayerID and s.teamName = p.soccerPlayerTeamName and p.goals >= 3;

select t.name

from soccerPlayer s, participate p, Team t

where s.id = p.soccerPlayerID and s.teamName = p.soccerPlayerTeamName and p.goals >= 3 and s.teamName = t.name;

select t.name

from Game g, Team t

where (g.homeTeamName = t.name and g.homeGoals>3) or (g.awayTeamName = t.name and g.awayGoals >3)

select t.name

from Game g, Team t

where (g.homeTeamName = 'Cal Poly' and g.awayTeamName = t.name and g.homeGoals>g.awayGoals) or

(g.awayTeamName = 'Cal Poly' and g.homeTeamName = t.name and g.homeGoals<g.awayGoals)