Group Milestone 2

We have updated our ER diagram to meet the needs of our database more accurately, the new ER diagram is as follows:

Diagram

Description automatically generated

(We made one error on the ER diagram with no time to fix Stage.Match\_No should not exist. )

Our Constraints Are:

|  |  |  |
| --- | --- | --- |
| On Table | Constraint | Notes |
| Matches | Team A <> Team B |  |
| Player | Position can only hold certain values |  |
| Player | Award can only hold certain values |  |
| Matches | Winner must be the team with the greater score, null otherwise | We have not yet implemented this constraint in our code. |
|  |  |  |
| Stage | Match\_Type can only hold specific values |  |
|  |  |  |

**Front End:**

Our front end will be implemented as a web page scripted in Python or Node.js. The proposed design looks like this Graphical user interface, table

Description automatically generated

The final draft will implement either a block style mechanic, to limit the number of searches and reduce security vulnerabilities, or a plain search bar. Once a user makes a search the table will be populated with the results. The table has a built-in search function if the list gets too long and can perform sort-by functions as well.

**Relational Algebra**:

|  |
| --- |
|  |
|  | **Player With Most Goals**  MostGoals := πGoal(PLAYER)- πGoal[PLAYER ⨝ Score<Score2 (ρName2/NameρScore2/Score PLAYER) |
|  |  |
|  | **Matches Held In Given Stadium** |
|  | π MATCH.Match\_Number(σ STAGE.Stadium = '\*'(STAGE)⨝ MATCH) |
|  |  |
|  | **All teams who had a tie game** |
|  | TiedGames←σScore\_1=Score\_2(MATCH)πName(σ(Name=TiedGames.Team\_1 ∨ Name=TiedGames.Team\_2)(Team×TiedGames)) |
|  |  |
|  | **All players from given match** |
|  | πName(PLAYER)σName='\*'(TEAM) |
|  |  |
|  | **Average Score For Team** |
|  | \* G avg(Team) (π Team\_A as Team (σ (team\_A = "x") (Matches)) ∪ π Team\_B as Team ((σ (team\_B = "x") Matches)) |
|  |  |
|  | **Highest Score of Any Team** |
|  | Team G max(Score)(π(Team\_1 as Team, Score\_1 as Score)(Matches) ∪ π(Team\_2 as Team, Score\_2 as Score)(Matches)) |
|  |  |
|  | **All scores for a Given Group Stage** |
|  |  |
|  | Team G sum(Team)(((π (Team\_A as Team)(σ (Team\_A = Winner) Match ⨝ (match\_Number = Match\_No)(σ(Match\_Type = "group x")(Stage\_Junction)) )) |
|  | ∪ (π (Team A as Team)(σ (Winner isNull)(Match ⨝ (match\_Number = Match\_No)(σ(Match\_Type = "group x")(Stage\_Junction))))) |
|  | ∪ (π (Team\_B as Team)(σ (Team\_B = Winner) Match ⨝ (match\_Number = Match\_No)(σ(Match\_Type = "group x")(Stage\_Junction)))) |
|  | ∪ (π (Team B as Team)(σ (Winner isNull)(Match ⨝ (match\_Number = Match\_No)(σ(Match\_Type = "group x")(Stage\_Junction)))))) |
|  |  |
|  | **Matches Won By a Given Team** |
|  | π(Match Number, Team\_1, Team\_2, Result, Winner)(σ(Winner='Team A')(MATCH)) |
|  |  |
|  | **Stats On a Team** |
|  | π(Name, Nick\_Name, Manager, Captian)(σ(Name='Team A')(TEAM)) |
|  |  |
|  | **Players Who Held x Position** |
|  | π(Name, Nick\_Name, Number, Team\_Name, Position, Goal, Award)(σ(Name=’somePosition’)(Player)) |
|  |  |
|  | **Player With x Name** |
|  | π(Name, Nick\_Name, Number, Team\_Name, Position, Goal, Award)(σ(Name=’name’)(Player)) |
|  |  |
|  | **Player by Jersey Number** |
|  | π(Name, Nick\_Name, Number, Team\_Name, Position, Goal, Award)(σ(Name=’num’)(Player)) |

**Current SQL Code**:

create table Team

(

Team\_Name varchar(30) not null,

Nick\_Name varchar(30) not null,

Manager varchar(30) not null,

Player varchar(30) not null,

Captain varchar(30) not null,

primary key (Team\_Name),

unique (Team\_Name)

);

create table Stage

(

Match\_Type varchar(30) primary key,

Stadium varchar(30) not null,

Match\_time timestamp

);

create table Match\_junction

(

Match\_number integer not null,

Match\_Type varchar(30) not null,

primary key (Match\_number,Match\_Type)

);

create table Matches

(

Match\_Number integer not null,

Team\_1 varchar(50) not null,

Team\_2 varchar(50) not null,

Score\_1 integer not null,

Score\_2 integer not null,

Winner varchar(30),

Primary key (Match\_Number),

Unique (Match\_Number)

);

create table Player

(First\_Name varchar(30), Last\_Name varchar(30), Nick\_Name varchar(30),

Number int, Team\_Name varchar(50), Goals int);

Alter Table Player

ADD constraint pk\_Player primary key (Number, Team\_Name);

Alter Table Player

ADD constraint fk\_Player foreign key (Team\_Name) references Team(Team\_Name);

create table Position\_Constraints

(Position varchar(30) primary key);

insert into Position\_Constraints

values ("LB"), ("RB"), ("CM"), ("DM"), ("AM"), ("SS"), ("CF"), ("LW"), ("RW");

create table Player\_Position

( Number int, TName varchar(30), Position varchar(30));

Alter table Player\_Position

Add constraint pk\_Player\_Position primary key (TName, Position);

Alter Table Player\_Position

ADD constraint fk\_Player\_Position foreign key (Number, TName) references Player (Number, Team\_Name) On delete cascade;

Alter table Player\_Position

add constraint fk\_Position\_Constriaints foreign key (Position) references Position\_Constraints (Position);

create table Award\_Constraints

(Award varchar(30) primary key);

Insert into Award\_Constraints

values ("Golden Ball"), ("Golden Boot"), ("Golden Glove"), ("Man of the Match");

create table Player\_Award

(Number int, TName varchar(30), Award varchar(30));

alter table Player\_Award

ADD constraint pk\_Player\_Award Primary Key (TName, Award);

ALter table Player\_Award

ADD constraint fk\_Player\_Award foreign key (Number, TName) references Player (Number, Team\_Name) on delete cascade;

Alter table Player\_Award

ADD constraint fk\_Award\_Exists foreign key (Award) references Award\_Constraints (Award);

alter table Matches add constraint fk1\_TeamName

foreign key (Team\_1) references Team(Team\_Name);

alter table Matches add constraint fk2\_TeamName

foreign key (Team\_2) references Team(Team\_Name);

alter table Match\_junction add constraint fk\_MatchNo\_Junctiontable

foreign key (Match\_Number) references Matches(Match\_number);

alter table Match\_junction add constraint fk\_Matchtype

foreign key (Match\_Type) references Stage(Match\_Type);

CREATE TABLE Stage\_Names(Stage\_name varchar(30));

INSERT INTO Stage\_Names(Stage\_name) VALUES

('Group 1'),

('Group 2'),

('Group 3'),

('Group 4'),

('Group 5'),

('Group 6'),

('Group 7'),

('Group 8'),

('Sixteen'),

('Quarter-Finals'),

('Semi-Finals'),

('Finals');

ALTER TABLE Stage\_Names

Add Primary Key(Stage\_Name);

ALTER TABLE Stage

ADD constraint Stage\_name FOREIGN KEY Stage(Match\_Type) REFERENCES Stage\_Names(Stage\_Name);

**Schema Diagram:**

**Diagram

Description automatically generated**

If you would like direct access to our online database through MySQL you can input the following:

Standard TCP/IP

Hostname: cs470project.cy4ntiputuud.us-east-2.rds.amazonaws.com

Port: 3306

Username: admin

Connection name: cs470project

Password: Upon Request

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | |
|  |  | |
|  |  |  |
|  |  | |