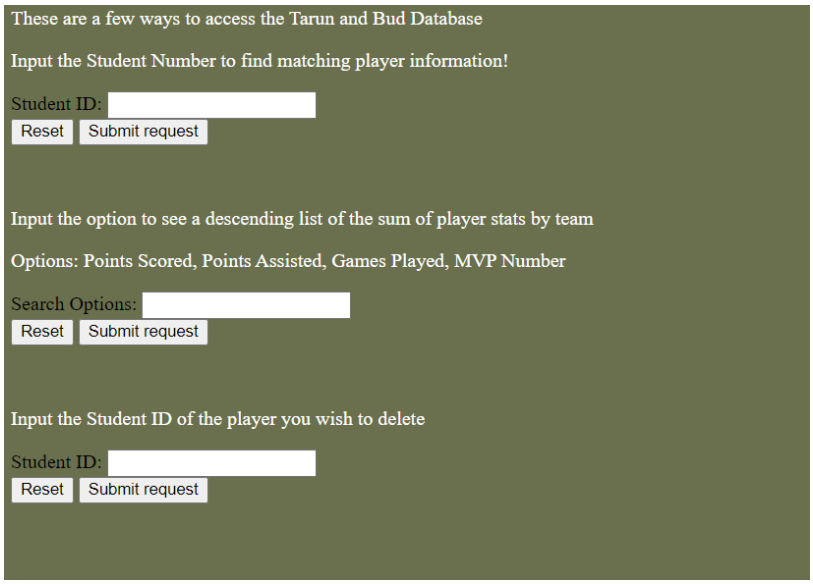
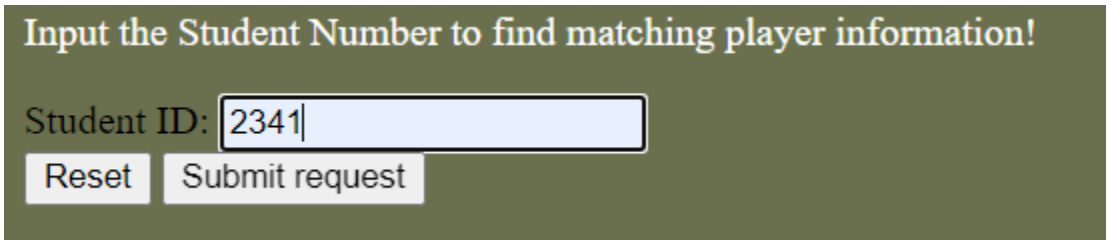
Project Finished: End of Winter Quarter 2022 - Sophomore Year
Context: This course was made to teach students about databases and how to work in SQL. There was a final project that involved student pairs working together to make a custom database. My partner Tarun Mathews and I made this without help, starting off with an idea, leading into a proposal. From there, we made an ER diagram and Relational diagram for our professor. After this, we programmed the tables and relationships in SQL before uploading it to the school server. To test our database more thoroughly, we made synthetic data using website generators. Finally, once everything was working, we made a website to interface with the database, shown below with the green background.

TL;DR - A project to make a custom database accompanied with a website to access it using SQL, PHP, and proper database planning procedures.

Website Screenshots **(The backend was deleted off of the school server when these were taken)**





Copy of Document 1: Proposal

CPSC 3300 Fundamental of Databases

PDA 1

Bud Robinson & Tarun Mathew

University Sports League

Description:

For out Database project we decided to create a database that would track hypothetical University Intermural Sports Leagues. We chose 3 sports to track: Soccer, Basketball, and Volleyball. Within these three sports we have decided to break them into 3 divisions: the Co-Ed, Men's, and Women's. Within each division there are a number of teams that we store and in each team are the players. A player can play in multiple sports, but only one team and one division within the sport. For the player we will be tracking statistics (Games played, Points Scored, Points Assisted). For the Team we will track their rank, the games played, and their record.

Entities:

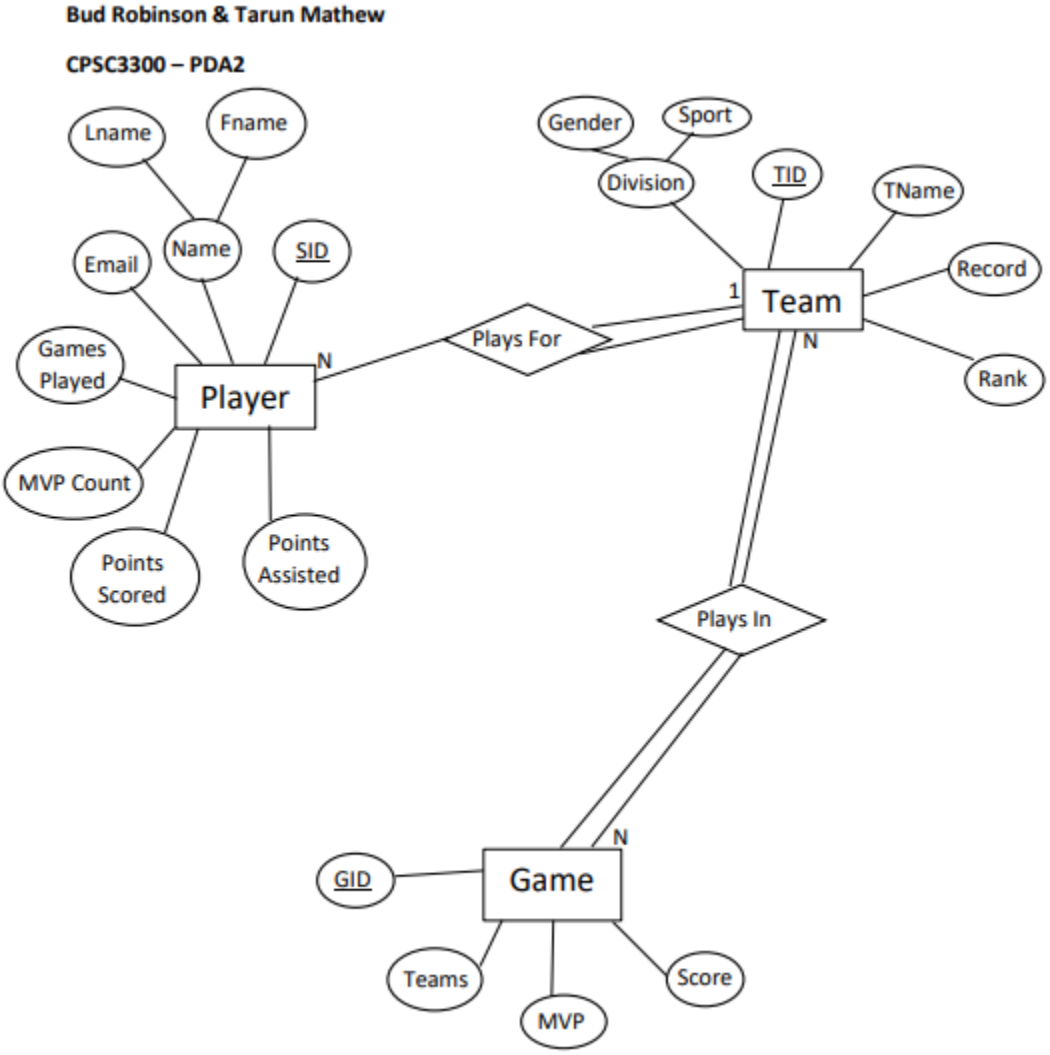
- **Player:** an individual who is enrolled at the university
  - *Student ID*: Unique ID of the student – primary key
  - *Name*: Full name of the student – non-null
  - *Email*: Email address of the student – non-null, unique
  - *Games Played*: Number of games played
  - *MVP Count:* All time number of MVPs
  - *Points Scored*: All time number of points earned personally
  - *Points assisted*: All time number of points with the student's direct involvement
- **Team:** a group of players who work together at games
  - *Team ID*: Unique ID of the team – primary key
  - *Team Name*: A customizable name for each group of players – non-null, unique
  - *Record*: The number of wins, losses, and games played
  - *Rank*: The standings of a team relative to other teams, derived from their record
  - *Division:* Team categorization (Co-Ed, Female, Male)(Basketball, Soccer, Volleyball)
- **Game:**
  - *Score:* The number of points for each team. Winner derived
  - *MVP:* Awarded to the player that has been most impactful
  - *Teams:* The two teams that are participating in the Game.
  - *GameID:* The specific ID that the game will have – primary key

Relationships:

- Plays In: (Player, Team), (Team, Game)

There are three essential entities for a league. The first is the player, who is a single individual. A player's attributes are a student ID, a name (first and last), an email, the number of games they've played, the number of MVP's, the number of points scored, and the number of points assisted. The second entity is the team, which is a collection of individuals who play together. A team's attributes are its team ID, name, win-loss-tie record, rank compared to other teams, and division (Co-Ed, Female, Male) (Basketball, Soccer, Volleyball). The last entity is the game, which is where teams face off against each other. The team's attributes are the team ID, the score, the MVP of the game, and the names of the two participating teams.
There are two relationships for a league. A player plays for a team. Many players play for a single team (many-to-one). A player is partial participation, as a player can make an account and fill out all their information but may not have a team to join. A team is total participation, because a team must have players to still be a team. A team plays a game. Many teams play in a game, but there is only a single game. A team is total participation because a team must play games as a requirement for establishing a team. Games are total participation because a game needs enough teams to properly function.

Copy of Document 2: ER Diagram



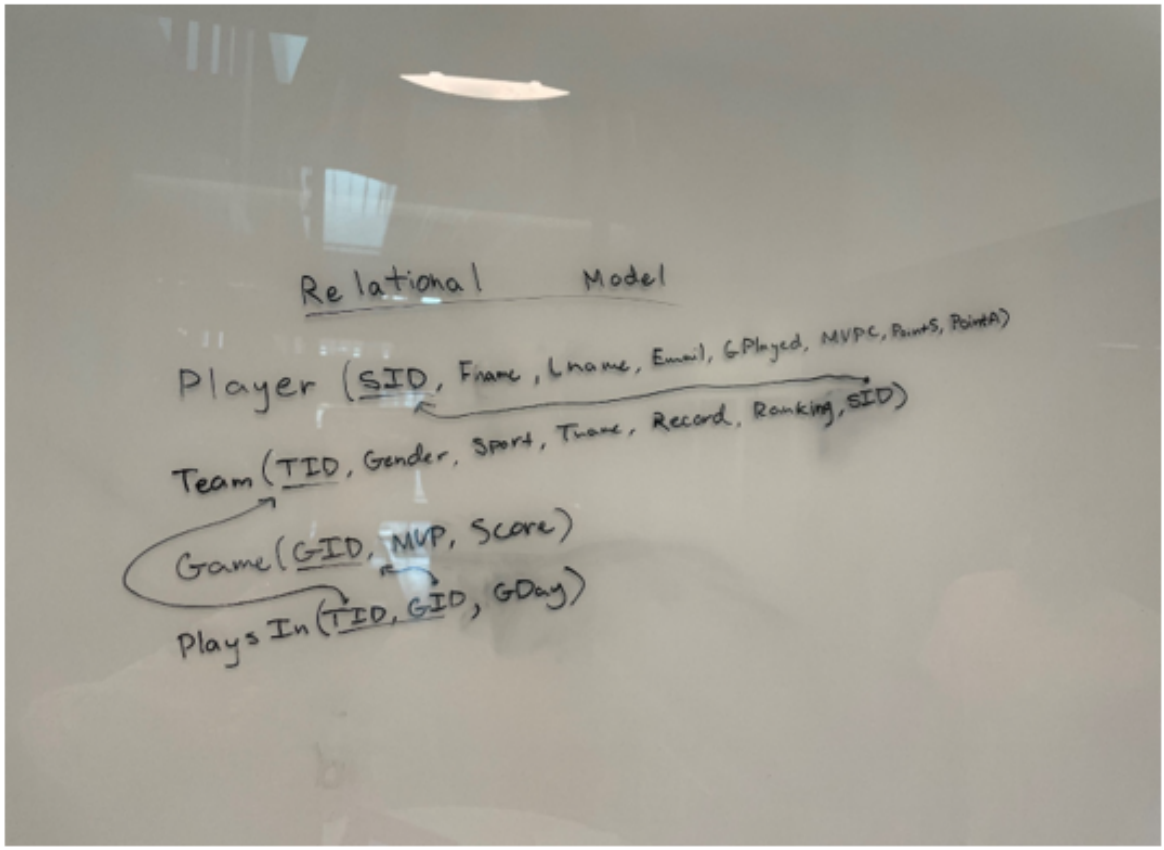
There are three essential entities for a league. The first is the player, who is a single individual. A player's attributes are a student ID, a name (first and last), an email, the number of games they've played, the number of MVP's, the number of points scored, and the number of points assisted. The second entity is the team, which is a collection of individuals who play together. A team's attributes are its team ID, name, win-loss-tie record, rank compared to other teams, and division (Co-Ed, Female, Male) (Basketball, Soccer, Volleyball). The last entity is the game, which is where teams face off against each other. The team's attributes are the team ID, the score, the MVP of the game, and the names of the two participating teams. There are two relationships for a league. A player plays for a team. Many players play for a single team (many-to-one). A player is partial participation, as a player can make an account and fill out all their information but may not have a team to join. A team is total participation because a team must have players to still be a team. A team plays a game. Many teams play in a game, but there is only a single game. A team is total participation because a team must play games as a requirement for establishing a team. Games are total participation because a game needs enough teams to properly function

Copy of Document 3: Relational Diagram and Start of SQL

Tarun Mathew and Bud Robinson
CPSC3300

PDA 3

A) Relational Model



B) Create Tables

```
create table Player
(
    sid int,
    Fname varchar(20),
    Lname varchar(20),
    email varchar(40),
    gplayed int,
    MVPC int,
    PointS int,
    PointA int,
    primary key (sid)
);

desc Player;

create table Team
(
    tid int,
```

Field	Type	Null	Key	Default	Extra
sid	int	NO	PRI	NULL	
Fname	varchar(20)	YES		NULL	
Lname	varchar(20)	YES		NULL	
email	varchar(40)	YES		NULL	
gplayed	int	YES		NULL	
MVPC	int	YES		NULL	
PointS	int	YES		NULL	
PointA	int	YES		NULL	

```
create table Team
(
    tid int,
    gender varchar(10),
    sport varchar(15),
    Tname varchar(30),
    record varchar(10),
    ranking int,
    sid int,
    primary key (tid),
    foreign key (sid) references Player (sid)
);

desc Team;
```

Field	Type	Null	Key	Default	Extra
tid	int	NO	PRI	NULL	
gender	varchar(10)	YES		NULL	
sport	varchar(15)	YES		NULL	
Tname	varchar(30)	YES		NULL	
record	varchar(10)	YES		NULL	
ranking	int	YES		NULL	
sid	int	YES	MUL	NULL	

Result 2

Output

Action Output

#	Time	Action	Message
1	14:32:42	create table Player (sid int, Fname varchar(20), Lname varchar(20), email varchar(40), gplay...	0 row(s) affected
2	14:33:45	desc Player	8 row(s) returned
3	14:34:47	create table Team (tid int, gender varchar(10), sport varchar(15), Tname varchar(30), recor...	0 row(s) affected
4	14:34:53	desc Team	7 row(s) returned

```
36
37 • create table Game
38 (
39     gid int,
40     mvp varchar(40),
41     score varchar(10),
42     primary key (gid)
43 );
44 • desc Game;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
gid	int	NO	PRI	NULL	
mvp	varchar(40)	YES		NULL	
score	varchar(10)	YES		NULL	

Result 3

Output

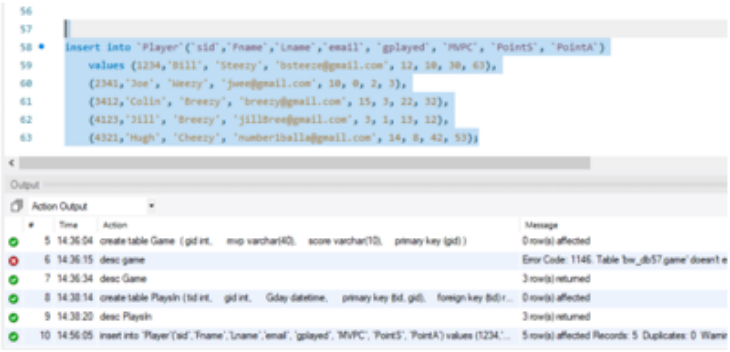
Action Output

#	Time	Action	Message
2	14:33:45	desc Player	8 row(s) returned
3	14:34:47	create table Team (tid int, gender varchar(10), sport varchar(15), Tname varchar(30), rec...	0 row(s) affected
4	14:34:53	desc Team	7 row(s) returned
5	14:36:04	create table Game (gid int, mvp varchar(40), score varchar(10), primary key (gid))	0 row(s) affected
6	14:36:15	desc game	Error Code: 1146. Table 'bw_db57.game' doesn't exist
7	14:36:34	desc Game	3 row(s) returned

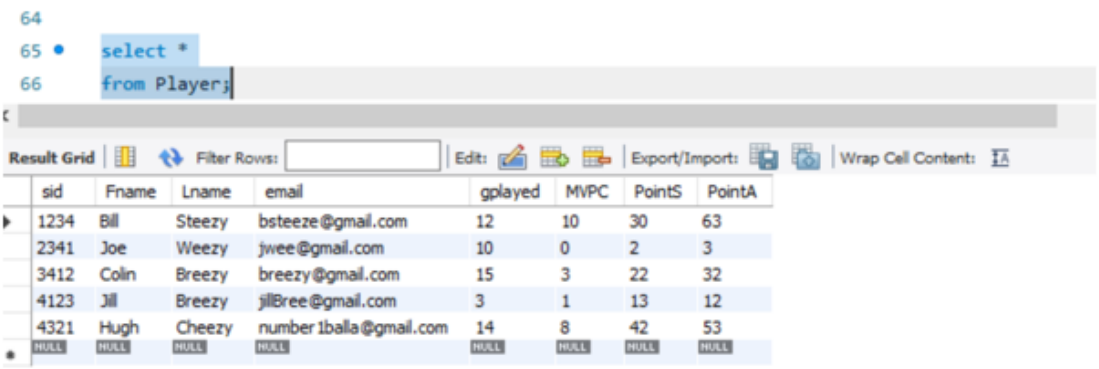
```
45
46 • create table PlaysIn
47 (
48     tid int,
49     gid int,
50     Gday datetime,
51     primary key (tid, gid),
52     foreign key (tid) references Team (tid),
53     foreign key (gid) references Game (gid)
54 );
55 • desc PlaysIn;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
tid	int	NO	PRI	NULL	
gid	int	NO	PRI	NULL	
Gday	datetime	YES		NULL	



D) Retrieving Data from Table



sid	Fname	Lname	email	gplayed	MVPC	PointS	PointA
1234	Bill	Steezy	bsteeze@gmail.com	12	10	30	63
2341	Joe	Weezy	jwee@gmail.com	10	0	2	3
3412	Colin	Breezy	breezy@gmail.com	15	3	22	32
4123	Jill	Breezy	jillBree@gmail.com	3	1	13	12
4321	Hugh	Cheezy	number1balla@gmail.com	14	8	42	53
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

E) Report

We did our Relational Model on a whiteboard and expanded on/edited small parts of the ER Diagram that we submitted for PDA2. Specifically, we added GDay to the plays in relationship to have game times. For the SQL tables and tuples, we emulated other assignments and adapted the syntax.

Copy of Document 4:

PDA 4 Summary

(a-b) We chose to use the website https://www.mockaroo.com/ to make our data. We chose between the custom inputs present for each column and synthesized usable data for the purposes of this PDA. **Our code is on bw_db57.**

(c) Screenshots of Count(*):

Player:

```
70
71 •  select count(*)
72     from Player;
```

Result Grid | Filter Rows:

count(*)
1005

Team:

```
71 •  select count(*)
72     from Team;
```

Result Grid | Filter Rows:

count(*)
100

Game:

```
71 •  select count(*)
72     from Game;
```

Result Grid | Filter Rows:

count(*)
200

PlaysIn:

```
71 •  select count(*)
72    from PlaysIn;
```

Result Grid | Filter Rows:

count(*)
200

(d) Screenshot Examples:

Player:

```
71 •  select *
72    from Player
73    Limit 5;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

sid	tid	Fname	Lname	email	gplayed	MVPC	PointS	PointA
1	30	Zandra	Pardon	zpardon0@webeden.co.uk	23	94	15	46
2	38	Dicky	De Lasci	ddelasci1@blogger.com	93	49	48	93
3	64	Sonny	Marchetti	smarchetti2@arstechnica.com	25	4	8	38
4	63	Devina	Pavinese	dpavinese3@fastcompany.com	34	27	60	5
5	68	Ronnie	Fittes	rfittes4@delicious.com	3	4	48	55
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Team:

```
71 • select *
72   from Team
73   Limit 5;
```

Result Grid | Filter Rows: | Edit:

tid	gender	sport	Tname	record	ranking
1	mens	basketball	Dragons	19-0	1
2	F	rgskyoxf	rjzpsiwgvnww	96689	15
3	F	bkrqfsbn	fpneqjtzdneq	11749	20
4	F	iwizurlb	mhjefajblrkp	12895	13
5	F	chrydhgh	smrydmenobgj	14350	14
NULL	NULL	NULL	NULL	NULL	NULL

PlaysIn:

```
71 • select *
72   from PlaysIn
73   Limit 5;
```

Result Grid | Filter Rows:

tid	gid	Gday
1	3	2021-07-08
1	30	2022-07-24
1	33	2023-08-09
1	54	2021-07-06
1	60	2022-08-05
NULL	NULL	NULL

Reflection: We have successfully made synthetic data for our project. It took a while to iron out the data inconsistencies and issues with our SQL code, but it all works. Both of us are now confident in making sample data in the future and have further understood SQL.

Copy of Document 5:

PDA 5

A. Write Five database queries

a. Find the full name of five players

```
select Fname, Lname
from Player
Limit 5;
```

i.

ii.

	Fname	Lname
▶	Zandra	Pardon
	Dicky	De Lasci
	Sonny	Marchetti
	Devina	Pavinese
	Ronnie	Fittes

ii.

iii.

b. Find the Full Name of Five Players on Team and their team

```
select Fname, Lname, Tname
from Player P, Team T
where P.tid = T.tid
Limit 5;
```

i.

ii.

	Fname	Lname	Tname
▶	Billie	Rolland	sjggllzchkfr
	Gina	Whorlton	sjggllzchkfr
	Slade	Yitzhakov	sjggllzchkfr
	Christoffer	De Maria	sjggllzchkfr
	Sigvard	Theriot	sjggllzchkfr

ii.

iii.

c. Find the Full Name of Five Players and the Team (Without using Join)

```
select Fname, Lname, Tname
from Player P, Team T
where P.tid IN (Select tid
                from Team)
Limit 5;
```

i.

ii.

Result Grid | Filter Rows:

	Fname	Lname	Tname
▶	Billie	Rolland	viuobbfqcxlw
	Gina	Whorlton	viuobbfqcxlw
	Slade	Yitzhakov	viuobbfqcxlw
	Christoffer	De Maria	viuobbfqcxlw
	Sigvard	Theriot	viuobbfqcxlw

ii.

iii.

d. Find Players from teams who are men

i.
```sql
Select P.Fname, P.Lname, T.gender
From Player P, Team T
where exists (
select *
from Team T2
where P.tid = T2.tid and T2.gender = 'M'
)
Limit 5;
```

ii.

	Fname	Lname	gender
▶	Stacia	MacQuarrie	mens
	Miof mela	MacMaster	mens
	Aloysius	Djurisic	mens
	Christin	Kembrey	mens
	Rayna	Sparshott	mens

ii.

iii.

e. Find the number of players on each team, group by team ID and only display teams having 6+ players

i.
```sql
Select count(P.tid) as countOfPlayers, T.Tname
From Player P, Team T
where P.tid = T.tid
group by P.tid
having count(P.tid)>5
Limit 5;
```

ii.

	countOfPlayers	Tname
▶	14	Dragons
	13	rjzpsiwgvnww
	12	fpneqjtzdneq
	7	mhjefajblrkp
	11	smrydmenobgj

ii.

iii.

B. Write SQL commands for data modification

a. Insert

i.
```sql
insert into `Team`(`tid`, `gender`, `sport`,`Tname`,`record`,`ranking`)
values(1453, 'mens', 'basketball', 'Crocs', '13-0', 3);
```

ii.

ii. 93 15:43:27 insert into `Team`(`tid`, `gender`, `sport`,`Tname`,`record`,`ranking`) values(1453, 'mens', 'basketball', 'Crocs', ... 1 row(s) affected

iii.

b. Update

i.
```sql
update Team
Set ranking = 3
where tid > 5 and tid < 10;
```

ii.

ii. 97 15:50:37 update Team Set ranking = 3 where tid > 5 and tid < 10 4 row(s) affected Rows matched: 4 Changed: 4 Warnings: 0

iii.

c. Delete

```
delete from Team where tid = 1453;
```

i.

ii.

ii. 99 15:52:35 delete from Team where tid = 1453 1 row(s) affected

iii.

C. VIEWS

```
Create View Player_stats as
select Fname, Lname, PointS, PointA
from Player
```

a.

b.

b. 100 15:56:48 Create View Player_stats as select Fname, Lname, PointS, PointA from Player

c.

```
select Fname, Lname, PointS
from Player_stats
where PointS > 10
limit 5;
```

c.

d.

Fname	Lname	PointS
Zandra	Pardon	15
Dicky	De Lasci	48
Devina	Pavinese	60
Ronnie	Fittes	48
Layla	Watson-Brown	63

d.

e.

e. Before:

```
88 •  select Fname
89    from Player
90    where sid = 8127;
91
```

Result Grid | Filter Rows:

Fname

i.

ii.

```
select Fname
from Player_stats
where Fname = 'Test';
```

Result Grid | Filter Rows: | Ex

Fname

ii.

iii.

f. Inserting

```
insert into `Player`(`sid`, `tid`, `Fname`,`Lname`,`email`, `gplayed`, `MVPC`, `PointS`, `PointA`)
    values (8127, 0001, 'Test', 'Player', 'testing@gmail.com', 43, 23, 823, 344);
```

i.

ii.

ii. 107 16:06:21 insert into `Player`(`sid`, `tid`, `Fname`,`Lname`,`email`, `gplayed`, `MVPC`, `PointS`, `PointA`) values (8127, 0... 1 row(s) affected

iii.

g. After

```
select Fname
from Player
where Fname = 'Test';
```

Result Grid | Filter Rows:

Fname
Test

i.

ii.

```
select Fname
from Player_stats
where Fname = 'Test';
```

Result Grid | Filter Rows:

Fname
Test

ii.

iii.

D. Triggers

a. A database trigger is a piece of code the executes when an event occurs. This event is a named system that passively runs in the background. When a certain condition is met by a database process (ex. Insert into a certain table), an action occurs. This could be inserting default data for an entry to avoid NULL data entries, or any other actions which the database admin sees fit.

b.

```
87 •  Create trigger autoData
88    after insert on Team
89    for each  row
90    update new.Team
91    set ranking = 20
92    where ranking>20;
93
94
```

Output

Action Output

#	Time	Action	Message
109	16:08:02	select Fname from Player_stats where Fname = 'Te...	1 row(s) returned
110	16:24:48	Create trigger autoData after insert on Team for eac...	0 row(s) affected

c.

Copy of Document 6:

Consist of the PHP files and HTML file in the repository, was making the website