AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Course Title

INTRODUCTION TO DATABASE

Summer 2023-2024

Section: [M]

PROJECT TITLE

TOURNAMENT MANAGEMENT SYSTEM

Supervised By

Md. Sajid Bin Faisal

Submitted By: Group no: 01

Name	ID
Sirajum Munir	**22-48549-3**
Anindita Bhattacharjee	**22-48606-3**
Fariar Rahman Arpa	**22-49040-3**
Min Shariar Digonto	**23-50689-1**

TABLE OF CONTENTS

Introduction

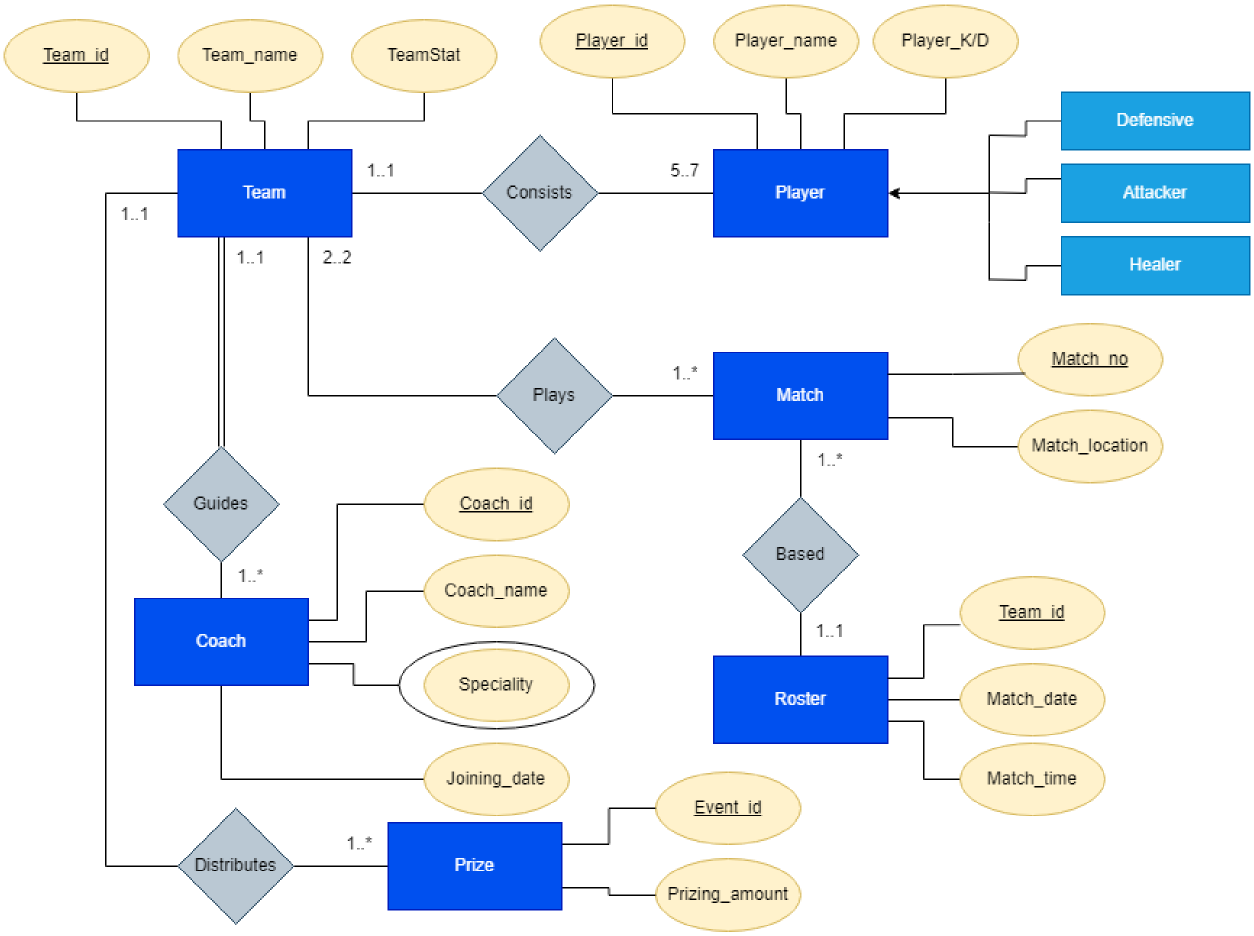
This project is about an online game tournament management system where data of different teams,their coaches,matches are stored orderly.This project ensures the smooth management of the online game tournament. While online gaming brings numerous economic benefits such as job creation, technological innovation, development of esports and so on. Online gaming stands as a powerhouse, driving economic growth and shaping modern entertainment landscapes worldwide.

Case Study

TOURNAMENT MANAGEMENT SYSTEM

In a tournament management system, there is information about teams, matches, players and coaches. Each team has a team id, team name, team state. Teams consist of five players per team. There can be a maximum number of seven players in a team. Players have name, id and date of joining and specialties are also listed. Each team must have a minimum number of one coach. Coaches have coach_id,coach_name,coach_joiningdate and speacilities.And each coach can only guide one team. There might be coaches without a team but there is no team without coaches. Teams play matches. Matches have match number, match location. Each team has many matches, but every match must have only two teams. Matches are based on rosters. Rosters have team id, match date and match time. In a roster, there can be information of many matches. Players can be specialized into 3 categories. These are defensive, attacker and healer. Prices are distributed among the teams. Each team can achieve many prizes and one prize can be distributed to one team.

ER_Diagram


Team_id
Team_name
TeamStat
Player_id
Player_name
Player_K/D
Team
1..1
Consists
5..7
Player
Defensive
Attacker
Healer
1..1
1..1
2..2
Plays
1..*
Match
Match_no
Match_location
1..*
Guides
Coach_id
Coach_name
Speciality
Based
1..*
Coach
1..1
Roster
Team_id
Match_date
Match_time
Joining_date
Event_id
Distributes
1..*
Prize
Prizing_amount

Normalization

Plays:

UNF: team_slot, team_id, team_no, mat_no, mat_loc.

1NF: team_slot, team_id(PK), team_no, mat_no(PK), mat_loc.

2NF: 1) team_slot, team_id(PK), team_no

2) mat_no, mat_loc(PK), team_id(FK)

3NF: 1) team_slot, team_id(PK), team_no

2) mat_no, mat_loc(PK), team_id(FK)

Based:

UNF: mat_no,mat_loc,team_id,mat_date,mat_time

1NF: mat_no(PK),mat_loc,team_id(PK),mat_date,mat_time

2NF: 1) mat_no(PK),mat_loc,team_id(FK)

2) team_id(PK),mat_date,mat_time

3NF: 1) mat_no(PK),mat_loc,team_id(FK)

2) team_id(PK),mat_date,mat_time

Consists:

UNF: pl_name,pl_id,pl_K/D ratio,team_no,team_id,team_stat

1NF: pl_name,pl_id(PK),pl_K/D ratio,team_no,team_id(PK),team_stat

2NF: 1) pl_name,pl_id(PK),pl_K/D ratio, ,team_id(FK)

2) team_no, team_id(PK),team_stat

3NF: : 1) pl_name,pl_id(PK),pl_K/D ratio, ,team_id(FK)

2) team_no, team_id(PK),team_stat

Guides:

UNF: team_id,team_no,team_stat,co_id,co_name,co_jd,co_spe.

1NF: team_id(PK),team_no,team_stat,co_id(PK),co_name,co_jd,co_spe.

2NF: 1) team_id(PK),team_no,team_stat

2) co_id(PK),co_name,co_jd,co_spe, team_id(FK),

3NF: 1) team_id(PK),team_no,team_stat

2) co_id(PK),co_name,co_jd,co_spe, team_id(FK),

Distributes:

UNF: team_stat,team_id,team_no,p_amount,event_id

1NF: team_stat,team_id(PK),team_no,p_amount,event_id(PK)

2NF: 1) team_stat,team_id(PK),team_no

2) p_amount,event_id(PK), team_id(FK),

3NF: 1) team_stat,team_id(PK),team_no

2) p_amount,event_id(PK), team_id(FK),

Tables after normalization:

1) team_stat, team_id(PK), team_no

2) mat_no(PK), mat_loc, team_id(FK)

3) mat_no(PK),mat_loc,team_id(FK)

4) team_id(PK),mat_date,mat_time

5) pl_name,pl_id(PK),pl_K/D ratio, ,team_id(FK)

6) team_no, team_id(PK),team_stat

7) team_id(PK),team_no,team_stat

8) co_id(PK),co_name,co_jd,co_spe, team_id(FK)

9) team_stat,team_id(PK),team_no

10) p_amount,event_id(PK), team_id(FK),

Final Tables(6 Tables):

1) mat_no(PK),mat_loc,team_id(FK)

2) team_id(PK),mat_date,mat_time

3) pl_name,pl_id(PK),pl_K/D ratio, ,team_id(FK)

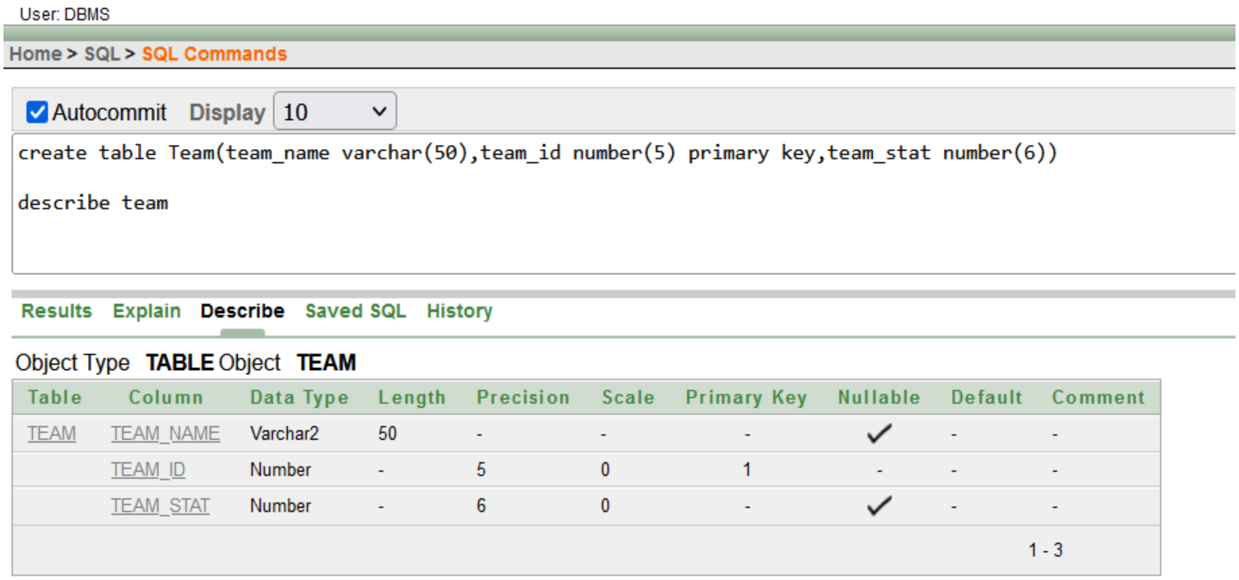
4) team_id(PK),team_no,team_stat

5) co_id(PK),co_name,co_jd,co_spe, team_id(FK)

6) p_amount,event_id(PK), team_id(FK)

Tables Creation:

TEAM



User: DBMS

Home > SQL > SQL Commands

☑ Autocommit Display 10

```
create table Team(team_name varchar(50),team_id number(5) primary key,team_stat number(6))

describe team
```

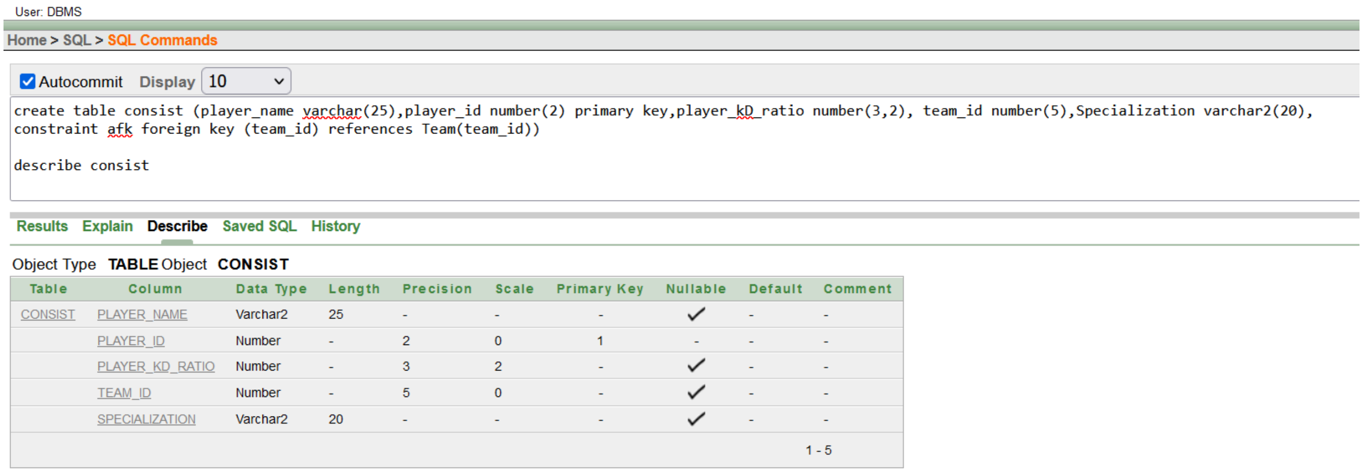
Object Type **TABLE** Object **TEAM**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TEAM	TEAM_NAME	Varchar2	50	-	-	-	✓	-	-
	TEAM_ID	Number	-	5	0	1	-	-	-
	TEAM_STAT	Number	-	6	0	-	✓	-	-

1 - 3

Fig 6.1 Team Table Creation

CONSIST


User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History

```
create table consist (player_name varchar(25),player_id number(2) primary key,player_kD_ratio number(3,2), team_id number(5),Specialization varchar2(20),
constraint afk foreign key (team_id) references Team(team_id))

describe consist
```

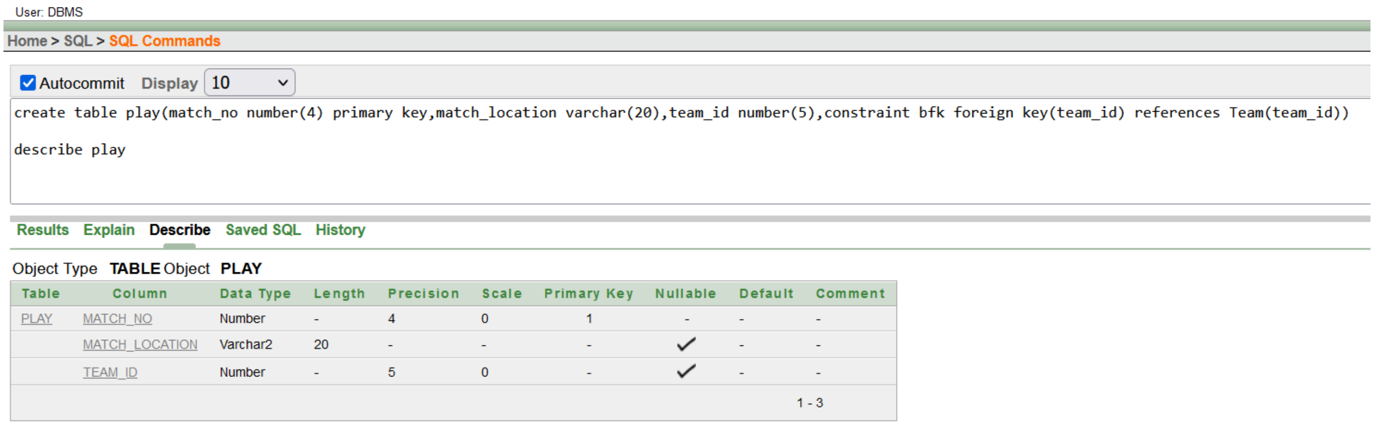
Object Type **TABLE** Object **CONSIST**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CONSIST	PLAYER_NAME	Varchar2	25	-	-	-	✓	-	-
	PLAYER_ID	Number	-	2	0	1	-	-	-
	PLAYER_KD_RATIO	Number	-	3	2	-	✓	-	-
	TEAM_ID	Number	-	5	0	-	✓	-	-
	SPECIALIZATION	Varchar2	20	-	-	-	✓	-	-

1 - 5

Fig 6.2 Consist Table Creation

PLAY


User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History

```
create table play(match_no number(4) primary key,match_location varchar(20),team_id number(5),constraint bfk foreign key(team_id) references Team(team_id))

describe play
```

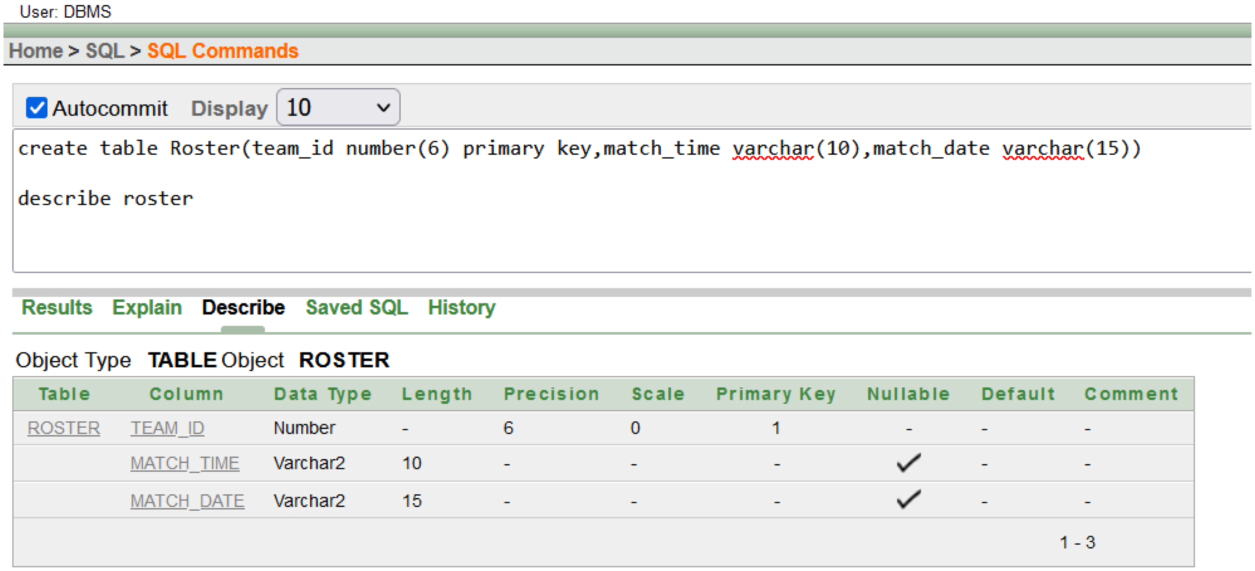
Object Type **TABLE** Object **PLAY**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PLAY	MATCH_NO	Number	-	4	0	1	-	-	-
	MATCH_LOCATION	Varchar2	20	-	-	-	✓	-	-
	TEAM_ID	Number	-	5	0	-	✓	-	-

1 - 3

Fig 6.3 Play Table Creation

ROSTER


User: DBMS
Home > SQL > SQL Commands
Results Explain Describe Saved SQL History

☑ Autocommit Display 10

```
create table Roster(team_id number(6) primary key,match_time varchar(10),match_date varchar(15))

describe roster
```

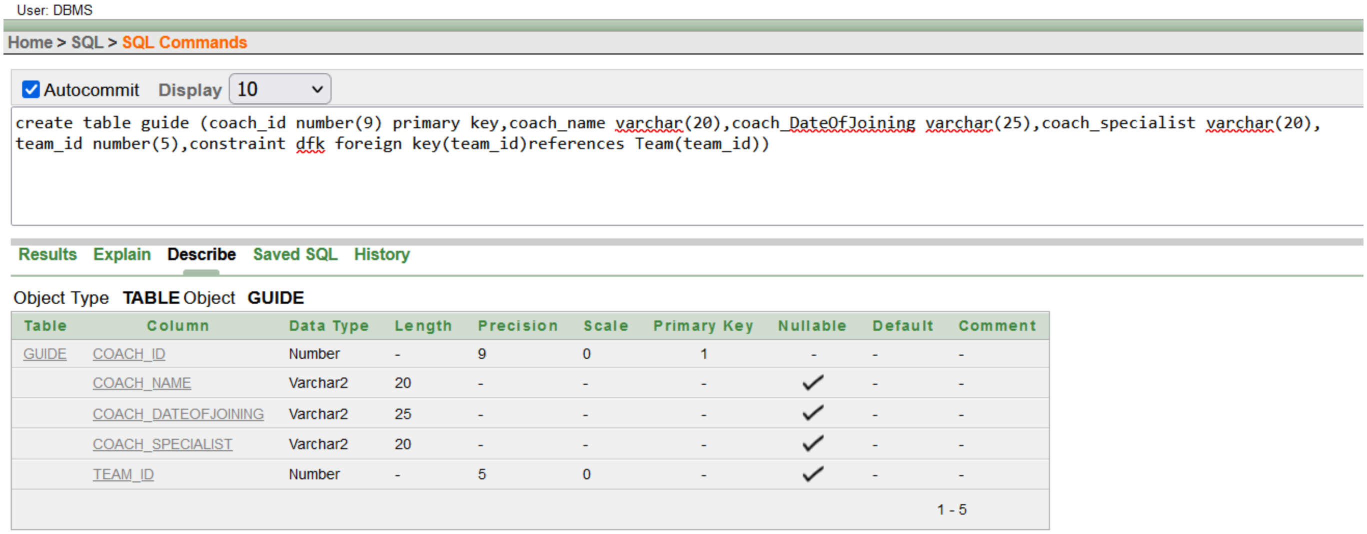
Object Type **TABLE** Object **ROSTER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ROSTER	TEAM_ID	Number	-	6	0	1	-	-	-
	MATCH_TIME	Varchar2	10	-	-	-	✓	-	-
	MATCH_DATE	Varchar2	15	-	-	-	✓	-	-

1 - 3

Fig 6.4 Roster Table Creation

GUIDE


User: DBMS
Home > SQL > SQL Commands
Results Explain Describe Saved SQL History

☑ Autocommit Display 10

```
create table guide (coach_id number(9) primary key,coach_name varchar(20),coach_DateOfJoining varchar(25),coach_specialist varchar(20),
team_id number(5),constraint dfk foreign key(team_id)references Team(team_id))
```

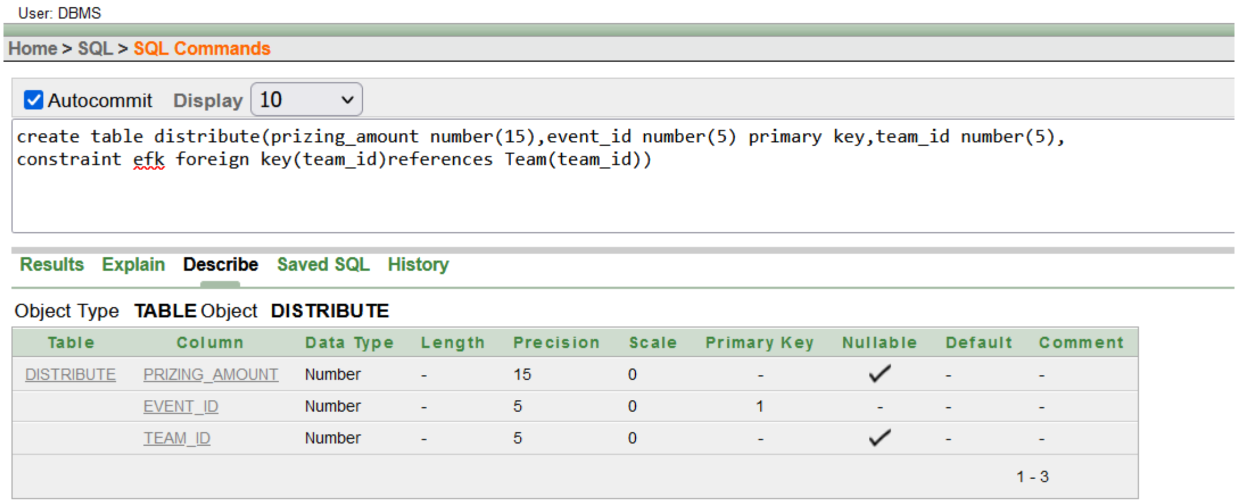
Object Type **TABLE** Object **GUIDE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GUIDE	COACH_ID	Number	-	9	0	1	-	-	-
	COACH_NAME	Varchar2	20	-	-	-	✓	-	-
	COACH_DATEOFJOINING	Varchar2	25	-	-	-	✓	-	-
	COACH_SPECIALIST	Varchar2	20	-	-	-	✓	-	-
	TEAM_ID	Number	-	5	0	-	✓	-	-

1 - 5

Fig 6.5 Guide Table Creation

DISTRIBUTE


User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History

```
create table distribute(prizing_amount number(15),event_id number(5) primary key,team_id number(5),
constraint efk foreign key(team_id)references Team(team_id))
```

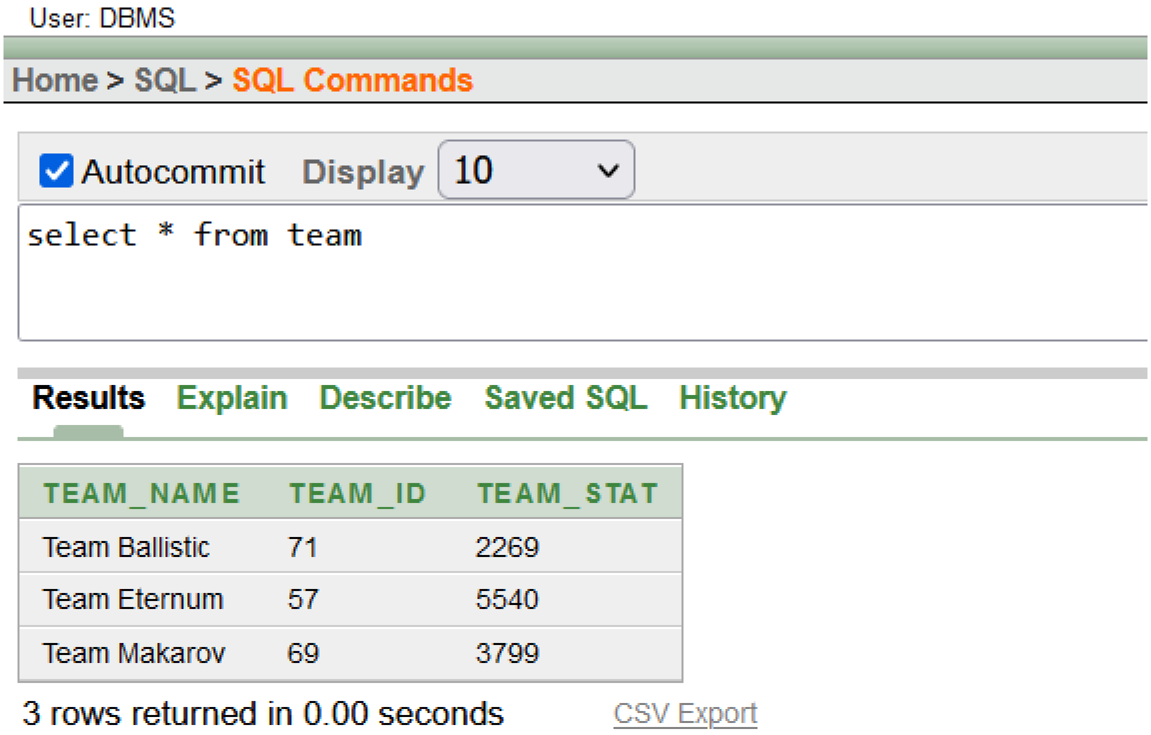
Object Type TABLE Object DISTRIBUTE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DISTRIBUTE	PRIZING_AMOUNT	Number	-	15	0	-	✓	-	-
	EVENT_ID	Number	-	5	0	1	-	-	-
	TEAM_ID	Number	-	5	0	-	✓	-	-

1 - 3

Fig 6.6 Distribute Table Creation

Value Insert:

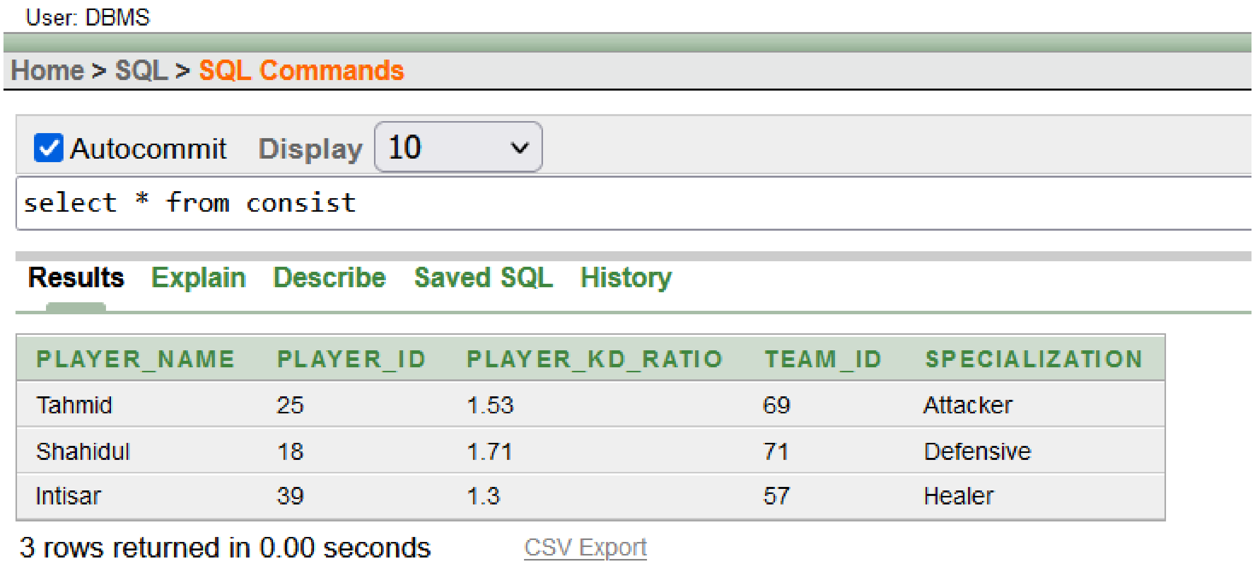

User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select * from team
```

TEAM_NAME	TEAM_ID	TEAM_STAT
Team Ballistic	71	2269
Team Eternum	57	5540
Team Makarov	69	3799

3 rows returned in 0.00 seconds

Fig 7.1 Value insertion for Team Table



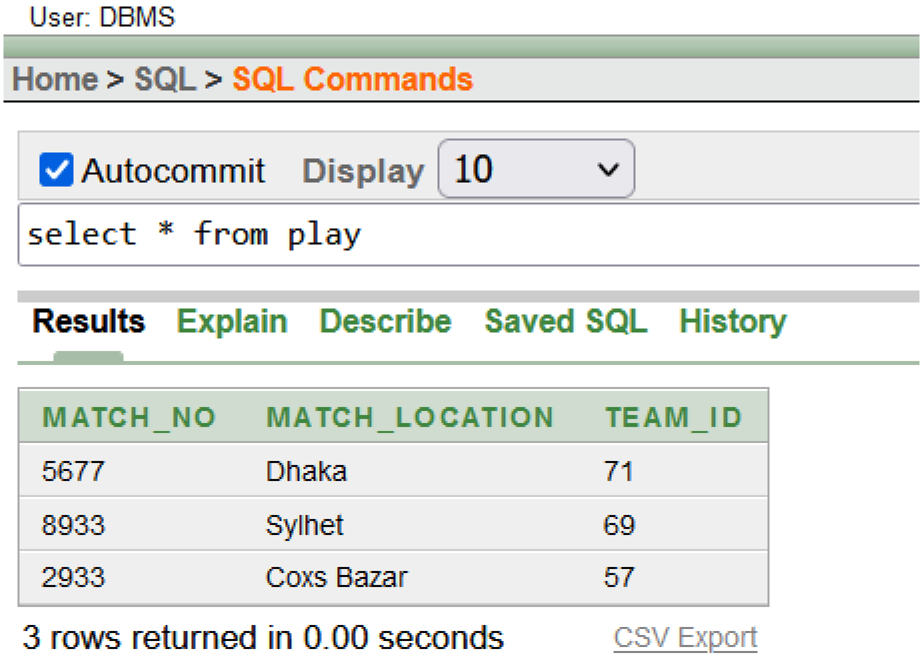
☑ Autocommit Display 10

```
select * from consist
```

PLAYER_NAME	PLAYER_ID	PLAYER_KD_RATIO	TEAM_ID	SPECIALIZATION
Tahmid	25	1.53	69	Attacker
Shahidul	18	1.71	71	Defensive
Intisar	39	1.3	57	Healer

3 rows returned in 0.00 seconds

Fig 7.2 Value insertion for Consist Table



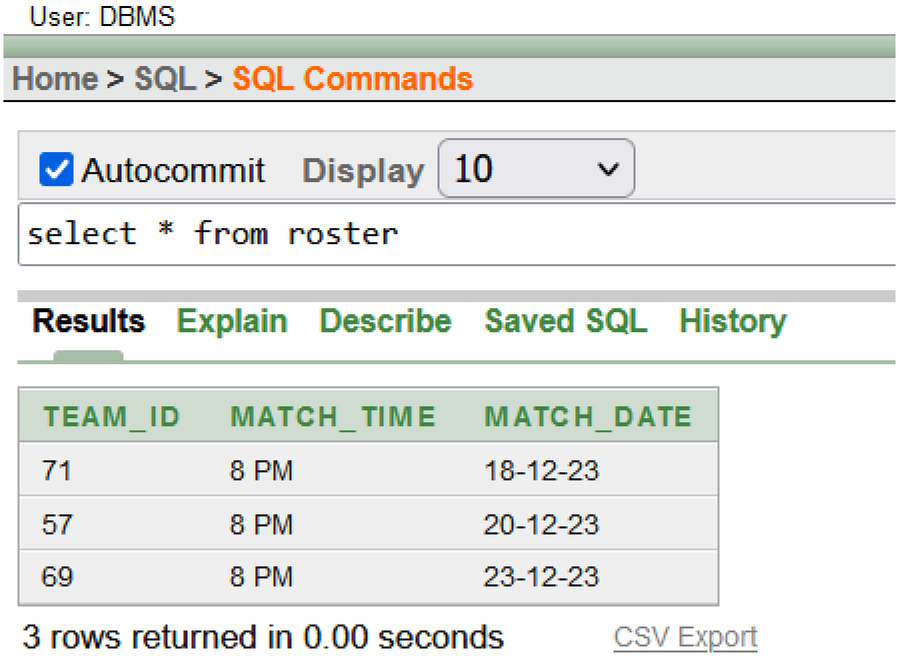
☑ Autocommit Display 10

```
select * from play
```

MATCH_NO	MATCH_LOCATION	TEAM_ID
5677	Dhaka	71
8933	Sylhet	69
2933	Coxs Bazar	57

3 rows returned in 0.00 seconds

Fig 7.3 Value insertion for Play Table



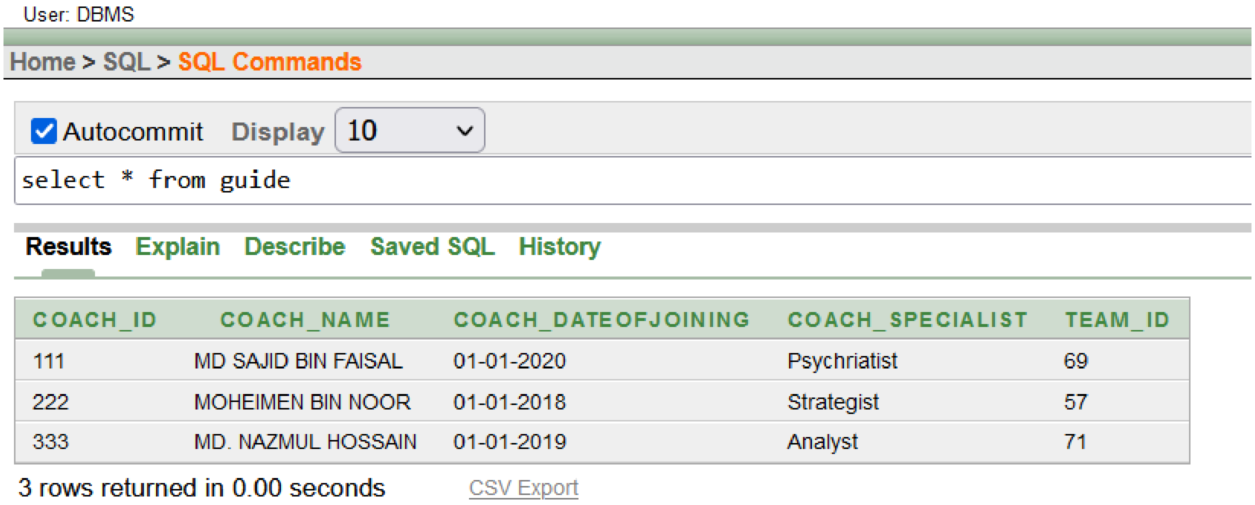
☑ Autocommit Display 10

```
select * from roster
```

TEAM_ID	MATCH_TIME	MATCH_DATE
71	8 PM	18-12-23
57	8 PM	20-12-23
69	8 PM	23-12-23

3 rows returned in 0.00 seconds

Fig 7.4 Value insertion for Roster Table



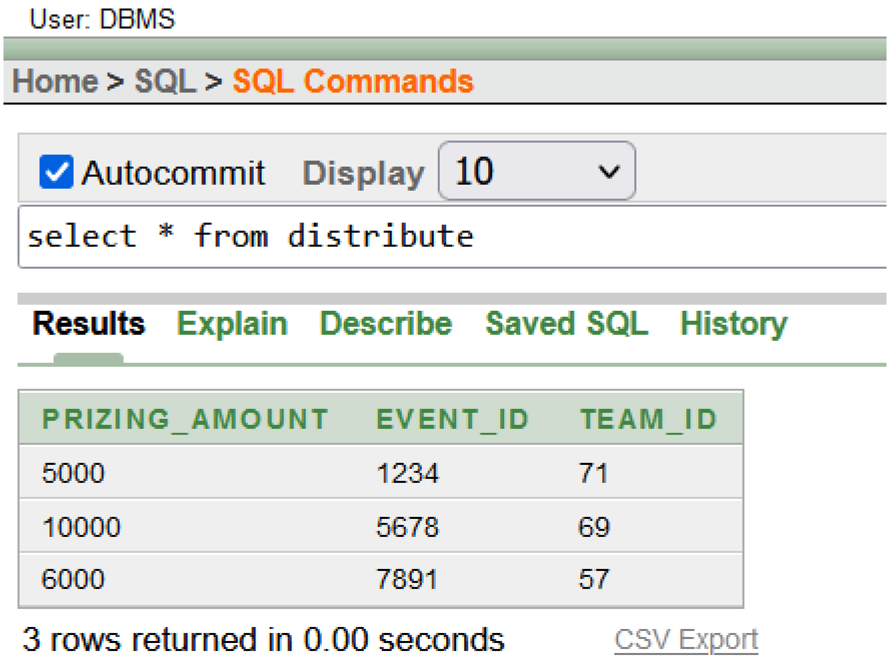
☑ Autocommit Display 10

```
select * from guide
```

COACH_ID	COACH_NAME	COACH_DATEOFJOINING	COACH_SPECIALIST	TEAM_ID
111	MD SAJID BIN FAISAL	01-01-2020	Psychriatist	69
222	MOHEIMEN BIN NOOR	01-01-2018	Strategist	57
333	MD. NAZMUL HOSSAIN	01-01-2019	Analyst	71

3 rows returned in 0.00 seconds

Fig 7.5 Value insertion for Guide Table



☑ Autocommit Display 10

```
select * from distribute
```

PRIZING_AMOUNT	EVENT_ID	TEAM_ID
5000	1234	71
10000	5678	69
6000	7891	57

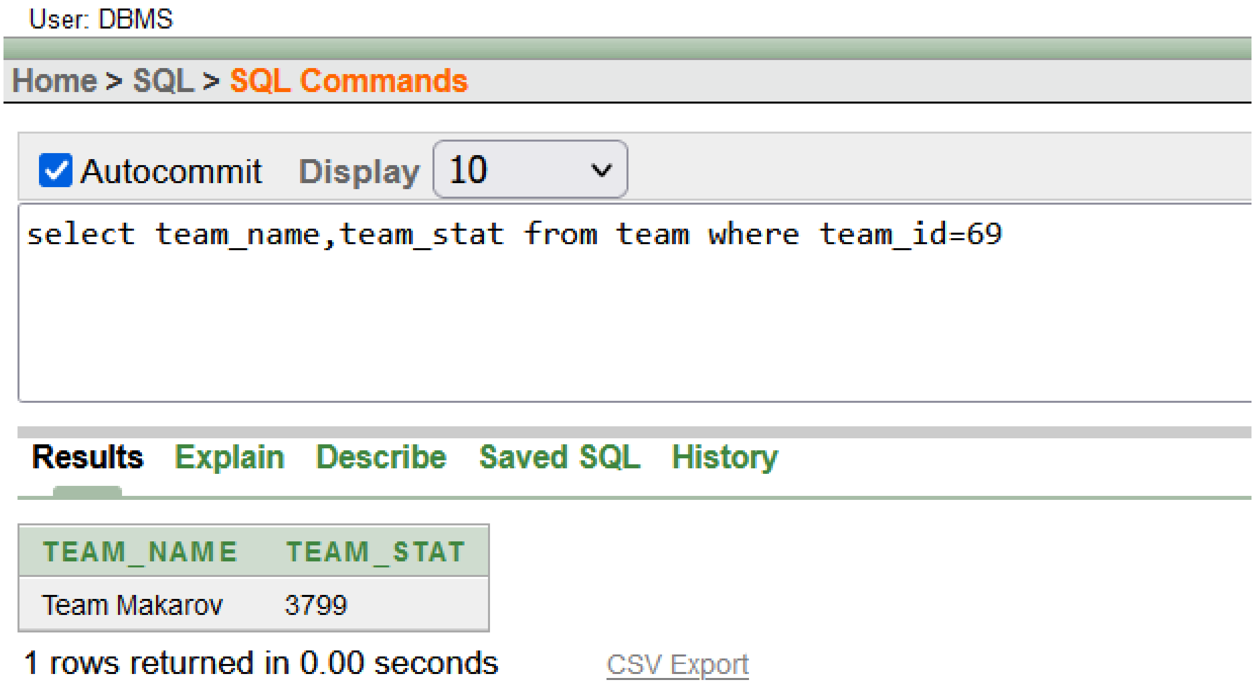
3 rows returned in 0.00 seconds

Fig 7.6 Value insertion for Distribute Table

Query Test:

1.Simple Query:

Show the team name and team stat from table team where team id is 69.



☑ Autocommit Display 10

```
select team_name,team_stat from team where team_id=69
```

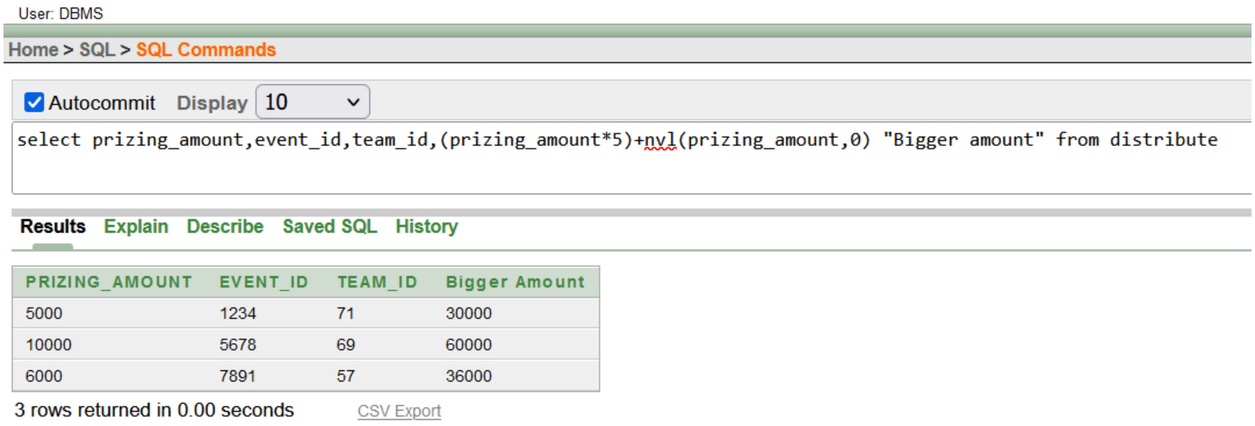
TEAM_NAME	TEAM_STAT
Team Makarov	3799

1 rows returned in 0.00 seconds

Fig 8.1 Simple Query

2.Single Row Function

Show prizing amount, event id, team id and prizing amount *5 replacing null with 0


User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select prizing_amount,event_id,team_id,(prizing_amount*5)+nvl(prizing_amount,0) "Bigger amount" from distribute
```

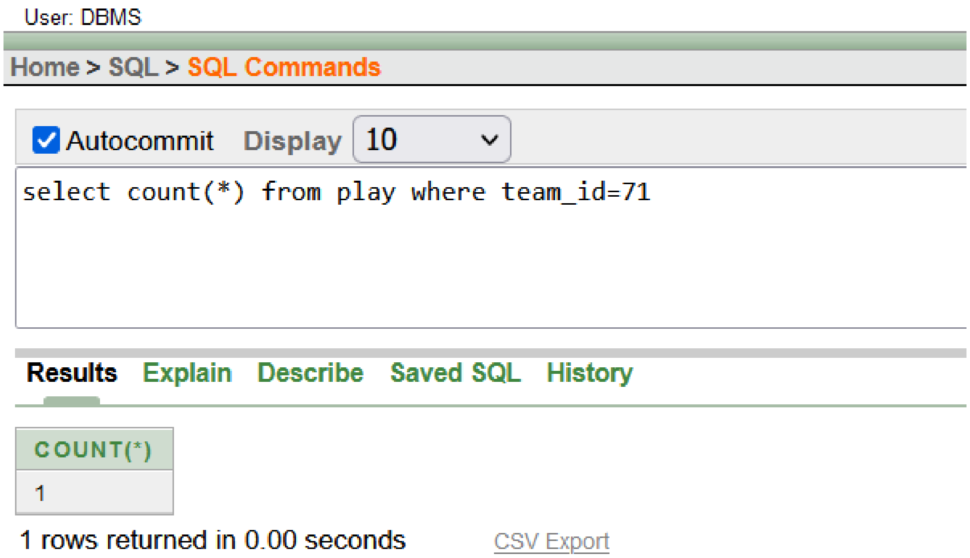
PRIZING_AMOUNT	EVENT_ID	TEAM_ID	Bigger Amount
5000	1234	71	30000
10000	5678	69	60000
6000	7891	57	36000

3 rows returned in 0.00 seconds

Fig 8.2 Single row function subquery

3.Group Function

Show the count of team id 71 from table play.


User: DBMS
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select count(*) from play where team_id=71
```

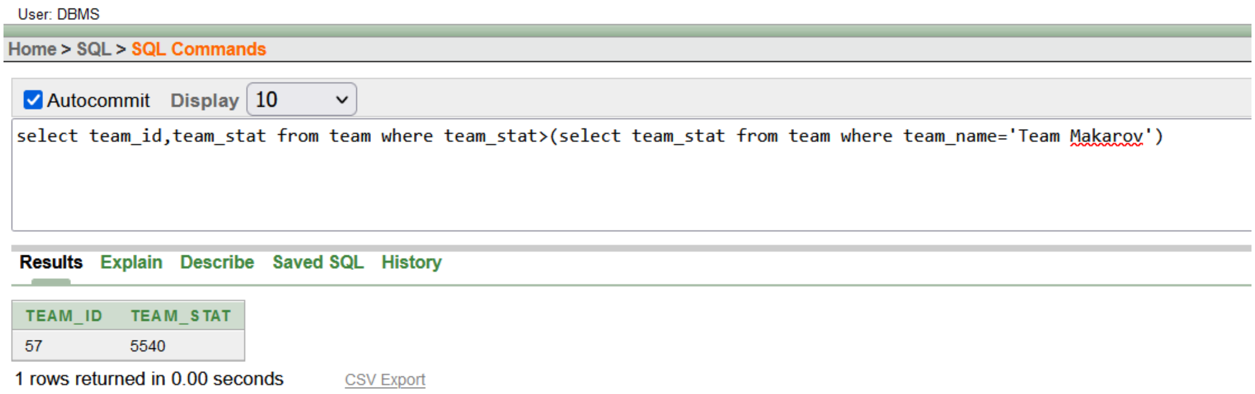
COUNT(*)
1

1 rows returned in 0.00 seconds

Fig 8.3 Group Function Query

4.Single Row Subquery

Show the team id, team stat higher than team Makarov from team table.



☑ Autocommit Display 10

```
select team_id,team_stat from team where team_stat>(select team_stat from team where team_name='Team Makarov')
```

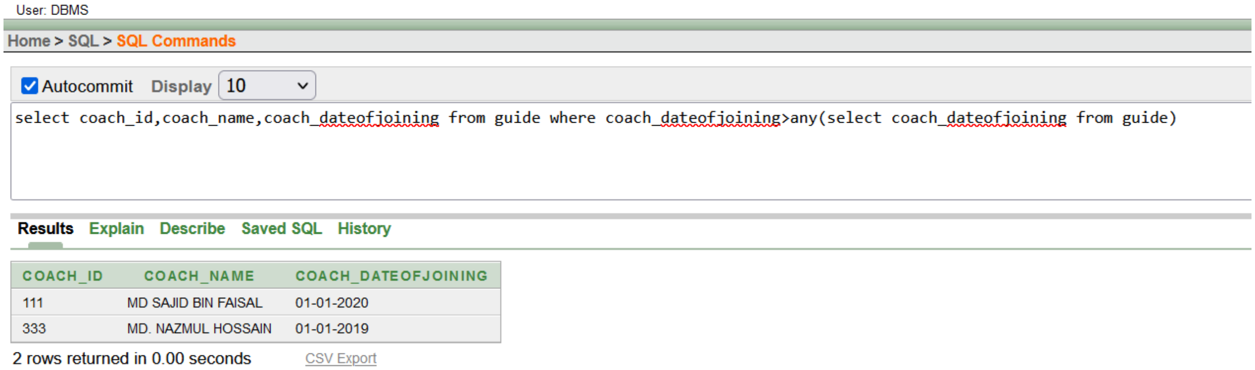
TEAM_ID	TEAM_STAT
57	5540

1 rows returned in 0.00 seconds

Fig 8.4 Single Row Subquery

5.Multiple Row Subquery

Show the coach id, name, date of joining where date of joining is higher than any other date of joining.



☑ Autocommit Display 10

```
select coach_id,coach_name,coach_dateofjoining from guide where coach_dateofjoining>any(select coach_dateofjoining from guide)
```

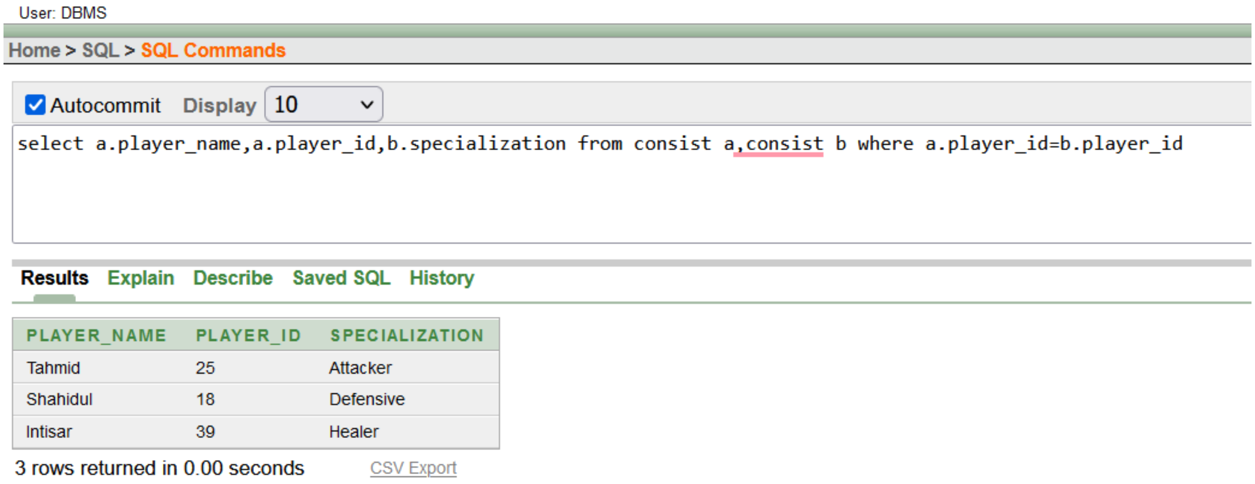
COACH_ID	COACH_NAME	COACH_DATEOFJOINING
111	MD SAJID BIN FAISAL	01-01-2020
333	MD. NAZMUL HOSSAIN	01-01-2019

2 rows returned in 0.00 seconds

Fig 8.5 Multiple Row Subquery

6.Joining (Self-Join)

Show the self-join from consist of table.



```
select a.player_name,a.player_id,b.specialization from consist a,consist b where a.player_id=b.player_id
```

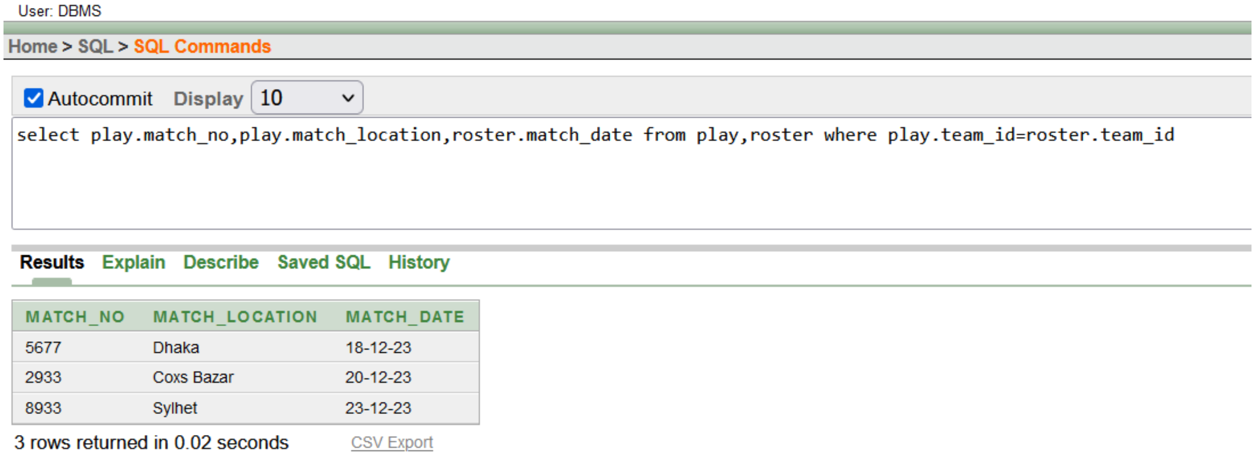
PLAYER_NAME	PLAYER_ID	SPECIALIZATION
Tahmid	25	Attacker
Shahidul	18	Defensive
Intisar	39	Healer

3 rows returned in 0.00 seconds

Fig 8.6.1 Self Join Query

Equijoin

Show the match number, location, date from Play and Roster Table



```
select play.match_no,play.match_location,roster.match_date from play,roster where play.team_id=roster.team_id
```

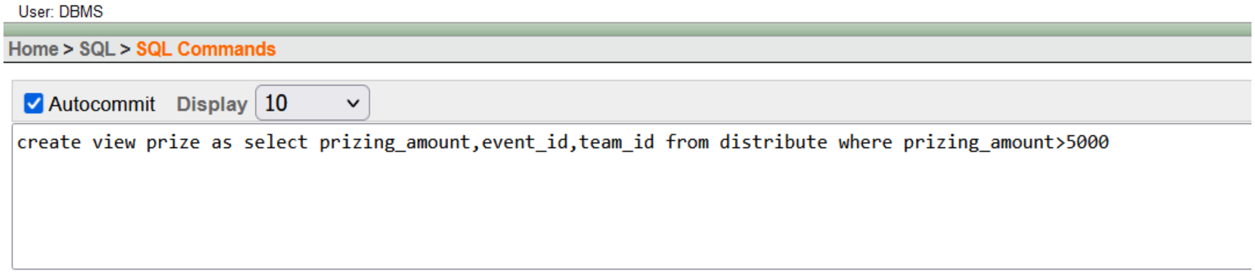
MATCH_NO	MATCH_LOCATION	MATCH_DATE
5677	Dhaka	18-12-23
2933	Coxs Bazar	20-12-23
8933	Sylhet	23-12-23

3 rows returned in 0.02 seconds

Fig 8.6.2 EquiJoin Query

Simple view

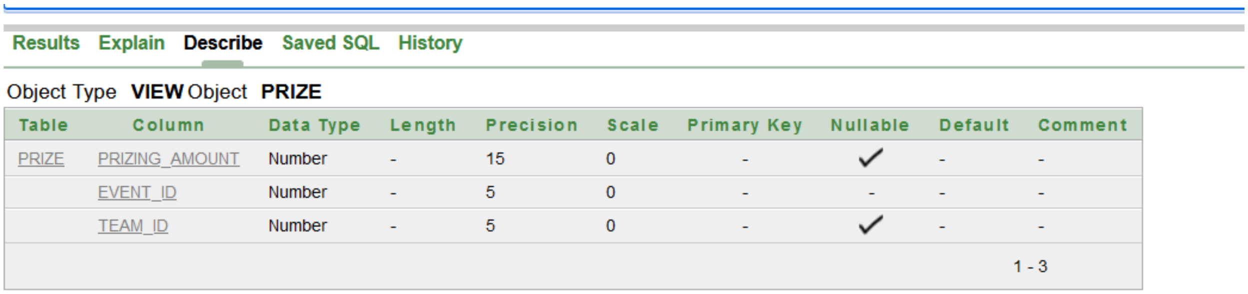
Create a view names as prize where the prizing amount over 5000 will be shown over the columns event_id,prizing_amount and team_id



☑ Autocommit Display 10

```
create view prize as select prizing_amount,event_id,team_id from distribute where prizing_amount>5000
```

Fig 8.7.1 Simple View Creation Command

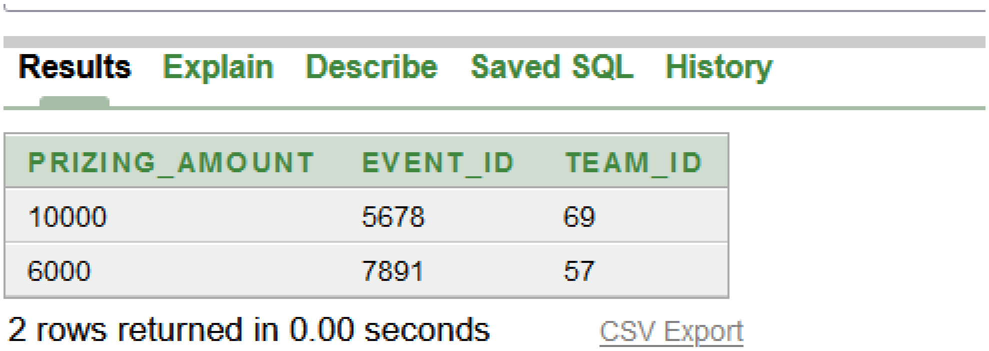


Object Type **VIEW** Object **PRIZE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRIZE	PRIZING_AMOUNT	Number	-	15	0	-	✓	-	-
	EVENT_ID	Number	-	5	0	-	-	-	-
	TEAM_ID	Number	-	5	0	-	✓	-	-

1 - 3

Fig 8.7.2 Description of the Simple View



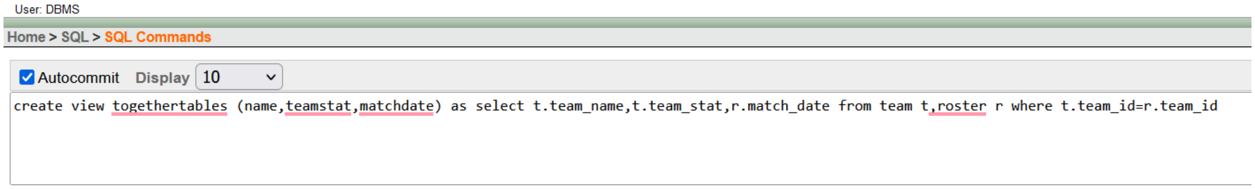
PRIZING_AMOUNT	EVENT_ID	TEAM_ID
10000	5678	69
6000	7891	57

2 rows returned in 0.00 seconds CSV Export

Fig 8.7.3 Result of the simple view as a whole table

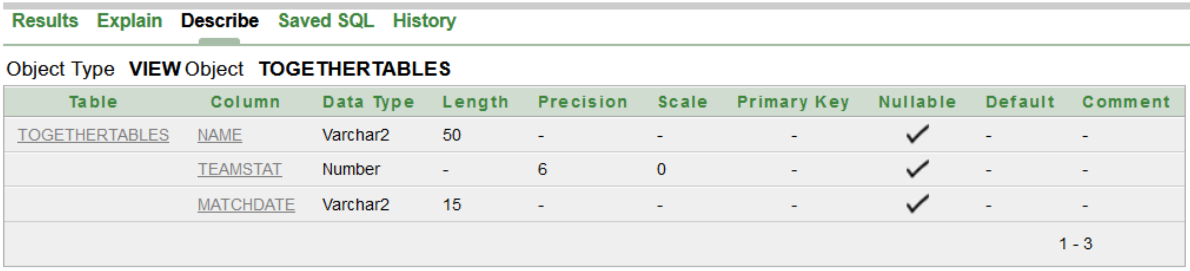
Complex View

Create a view named togthertables where team name,team stat and match date will be shown from table team and roster as they have team ID column in common.



```
create view togethertables (name,teamstat,matchdate) as select t.team_name,t.team_stat,r.match_date from team t,roster r where t.team_id=r.team_id
```

Fig 8.7.4 Complex view creation command

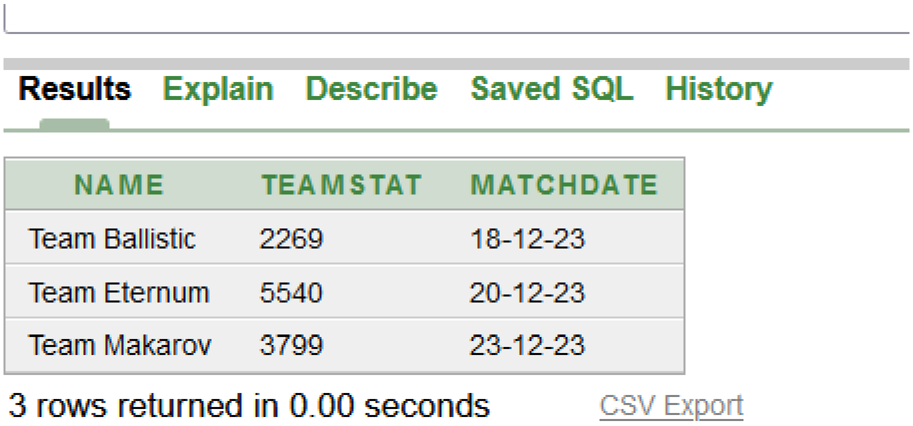


Object Type **VIEW** Object **TOGETHERTABLES**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TOGETHERTABLES	NAME	Varchar2	50	-	-	-	✓	-	-
	TEAMSTAT	Number	-	6	0	-	✓	-	-
	MATCHDATE	Varchar2	15	-	-	-	✓	-	-

1 - 3

Fig 8.7.5 Description of the Complex view



NAME	TEAMSTAT	MATCHDATE
Team Ballistic	2269	18-12-23
Team Eternum	5540	20-12-23
Team Makarov	3799	23-12-23

3 rows returned in 0.00 seconds CSV Export

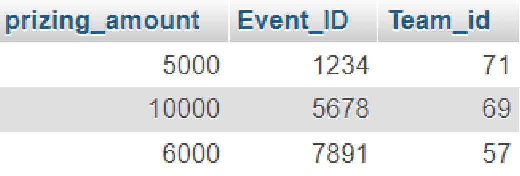
Fig 8.7.6 Result of the Complex view as a whole table.

Name: Anindita Bhattacharjee

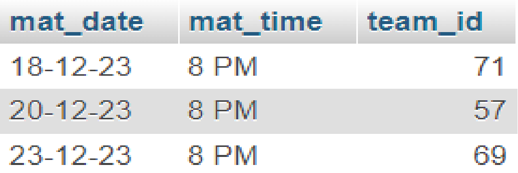
DATABASE CONNECTION

Procedure:

- Firstly, created a database named Tournament1 using XAMPP software and then started Apache and MySQL admin panel. Then, created two tables named Distribution and Roster.

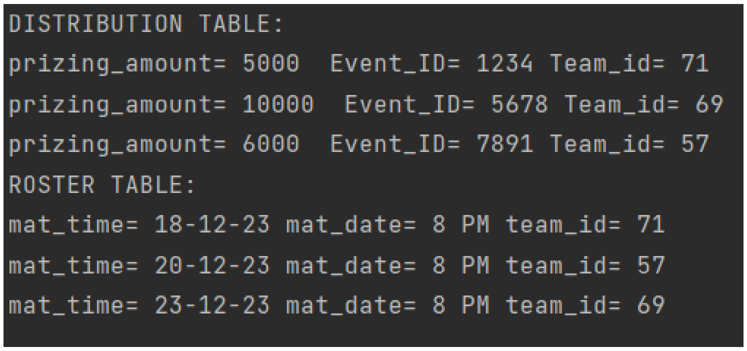


prizing_amount	Event_ID	Team_id
5000	1234	71
10000	5678	69
6000	7891	57



mat_date	mat_time	team_id
18-12-23	8 PM	71
20-12-23	8 PM	57
23-12-23	8 PM	69

- After that, added a jar file (mysql-connector-java-8.0.28 version) to the project library classpath.
- Using IDE (IntelliJ IDEA), registered the driver(“com.mysql.cj.jdbc.Driver”).
- Connected the MySQL server with the IDE (IntelliJ IDEA) through Connection function.
- Created a statement object to execute SQL queries and connecting with the connection function to collect the data from the MySQL server to execute.
- Then, executed the SQL query using executeQuery() and store results in a ResultSet.
- After that, used the Connection Close() function for disconnecting database and the program executed with this output.



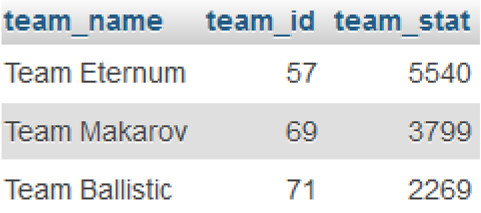
```
DISTRIBUTION TABLE:
prizing_amount= 5000  Event_ID= 1234 Team_id= 71
prizing_amount= 10000  Event_ID= 5678 Team_id= 69
prizing_amount= 6000  Event_ID= 7891 Team_id= 57
ROSTER TABLE:
mat_time= 18-12-23 mat_date= 8 PM team_id= 71
mat_time= 20-12-23 mat_date= 8 PM team_id= 57
mat_time= 23-12-23 mat_date= 8 PM team_id= 69
```

Database Connection

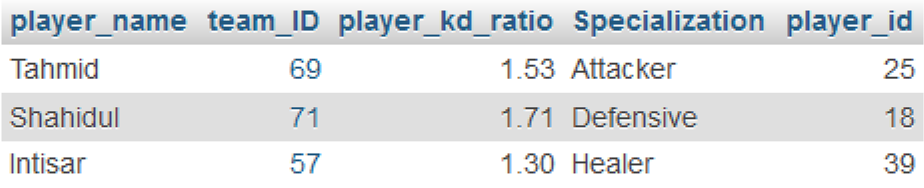
Name: Sirajum Munir

Procedure:

- Firstly, jar file “mysql-connector-j-8.2.0” was downloaded and kept in the C drive.
- XAMPP Control Panel v3.3.0 was downloaded and installed and from there Apache and MySql was started. From the Admin action, MySql was accessed.
- In MySQL, two tables named Team and Consist were created with value insertion.

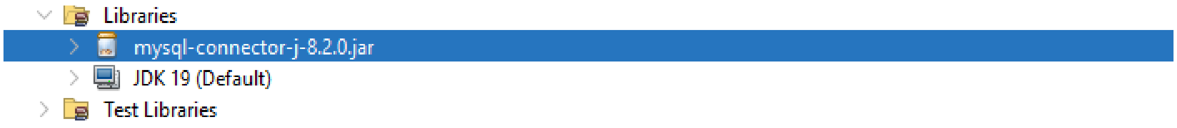


team_name	team_id	team_stat
Team Eternum	57	5540
Team Makarov	69	3799
Team Ballistic	71	2269

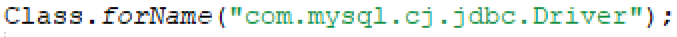


player_name	team_ID	player_kd_ratio	Specialization	player_id
Tahmid	69	1.53	Attacker	25
Shahidul	71	1.71	Defensive	18
Intisar	57	1.30	Healer	39

- The NetBeans IDE was used and the jarfile was added to the project library class path.

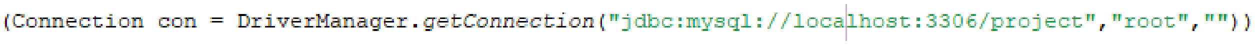


- The driver was registered with this line of code.



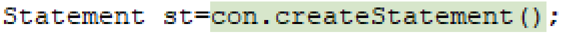
```
Class.forName("com.mysql.cj.jdbc.Driver");
```

- MySQL server was connected with the java code through this line.



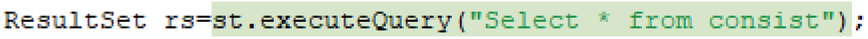
```
(Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root",""))
```

- Created a statement object st to Create statement.



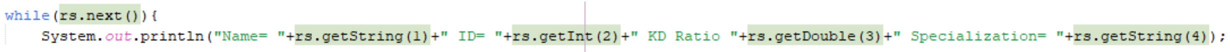
```
Statement st=con.createStatement();
```

- Then Created an object rs of ResultSet and executed the query using executeQuery() to call all data from table.



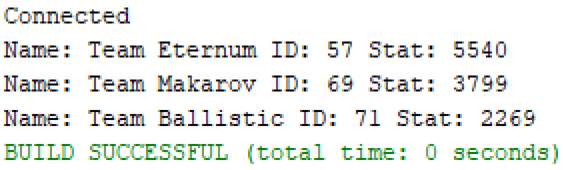
```
ResultSet rs=st.executeQuery("Select * from consist");
```

- Then with a line of code the columns of the tables were printed.

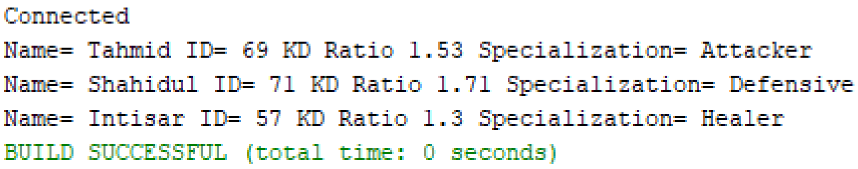


```
while(rs.next()){
    System.out.println("Name= "+rs.getString(1)+" ID= "+rs.getInt(2)+" KD Ratio "+rs.getDouble(3)+" Specialization= "+rs.getString(4));
```

- After that, used the Connection Close () function.
- Exception throwing statements were also used in this code to catch and throw any type of exceptions.



```
Connected
Name: Team Eternum ID: 57 Stat: 5540
Name: Team Makarov ID: 69 Stat: 3799
Name: Team Ballistic ID: 71 Stat: 2269
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
Connected
Name= Tahmid ID= 69 KD Ratio 1.53 Specialization= Attacker
Name= Shahidul ID= 71 KD Ratio 1.71 Specialization= Defensive
Name= Intisar ID= 57 KD Ratio 1.3 Specialization= Healer
BUILD SUCCESSFUL (total time: 0 seconds)
```

Conclusion:

This project was about a Tournament management system and made by Oracle DBMS. Firstly, an ER diagram was made then the ER diagram was normalized to the finalization point and from those final tables, Database was created in Oracle with multiple commands and then value was inserted. Several queries were tested. Then the project tables were connected to Java code with several steps individually by every team member. This was the project overall.