

# ANALYSIS OF E-SPORTS EARNINGS

PRESENTED BY: **VAISHNAVI TASHI**  
INSTITUTE NAME: SYMBIOSIS SKILLS  
& PROFESSIONAL UNIVERSITY.  
DATA ASSOCIATE COURSE



UNDER THE GUIDANCE OF:  
MISS.  
VAISHNAVI SATAV

- **Introduction:**

This is scraped data from eSportsEarning.com that provides information on earnings 💰 of e-Sports players and teams.

- **Dataset Generation:**

The data is collected from the [www.kaggle.com](http://www.kaggle.com) website. There are two different .CSV files in the dataset. The dataset contains the details of information on earnings of e-Sports players and teams. The data set downloaded from this website was in the .CSV format. The .CSV file had data about 1000 different entries in both files. Firstly, I cleaned the dataset and then have loaded the data of .CSV file into MySQL database by using the 'load data infile' command in the command line.

Following are two CSV files that contain in the dataset:

highest\_earning\_teams.csv

highest\_earning\_players.csv

The data from CSV files (1) & (2) includes these games:

1. Dota 2
2. Counter-Strike: Global Offensive
3. Fortnite
4. League of Legends
5. StarCraft II
6. Overwatch
7. PUBG
8. Hearthstone
9. Heroes of the Storm
10. Arena of Valor

- **About CSV file:**

Commas separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV files stores tabular data (numbers and text) in a plaintext. Each line of the file is a data record. Each record consists of one or more fields separated by commas. The use of commas as a field separator is the source of the name for this file format.

- **Data Cleaning:** The data obtained from the [www.kaggle.com](http://www.kaggle.com) website was not cleaned. I sorted the data by removing duplicate values and null values.

- **Dataset Description:**

The data set that I generated had the data from the .CSV files. The data set consist of two different tables i.e. Highest\_earning\_players and Highest\_earning\_teams. The highest\_earning\_players contains 7 different columns and 929 recors in the table. similarly, the highest\_earning\_teams contains 6 columns and 926 records in the table. I have created two different tables in the My-SQL command line by using syntax of create table in the SQL.

- **Technology used: MySQL 8.0 Command Line:**

MySQL client is a common name for tools that are designed to connect to MySQL Server. Client programs are used to send commands or queries to the server and allow managing data in the databases stored on the server.

- **Tools Used:**

- Microsoft word 2010
- Microsoft Excel 2010

- **Problem Statement:**

- The main aim of the project is to identify the team with the highest total USD prize.
- To manage, retrieve and predict the data based on the given database.

- **Types of Analysis:**

- 1) **Descriptive Analysis:**

The e-sport's highest earnings players consist of fields such as Player\_id, First\_name, Last\_name, Current \_handle, Country\_code, TotalUSDprize and games etc. In this table, Player\_id refers to the primary key in the database.

Likewise,

The e-sport's highest earnings teams consist of fields such as Player\_id, Team\_id, Teamname, TotalUSDprize, and Total\_tournaments,games. In this table, player\_id indicates the foreign key references from the highest\_earning\_players table.

- 2) **Diagnostic analysis:**

Identifying the team with the highest total USD prize with the help of given fields between two tables such as Player\_id, First\_name, Last\_name, Current \_handle, Country\_code, TotalUSDprize and game etc.

And,

Player\_id, Team\_id, Teamname, TotalUSDprize, and Total\_tournaments.

- 3) **Prescriptive analysis:**

Suggesting a new different team with the highest total tournaments with the help of fields contained in the Highest\_earning\_teams table such as Player\_id, Team\_id, Teamname, TotalUSDprize, and Total\_tournaments,games.

- **Project Work:**

following are the important SQL Commands that I have performed in the My-SQL Command Line:

- Create
- Load Data Command
- Limit Keyword
- Select Statement
- Aggregate Functions
- Joins
- Subqueries
- Clauses such as Order by, Group by and Where
- Desc and Asc Keyword. etc

- **Questions & Answers that performed in the SQL for Analysis:**

**1) Create table highest\_earning\_players and load CSV file into SQL having all the entries. Also, show the command for the load data into a file.**

```
Database changed
mysql> CREATE TABLE HIGHEST_EARNING_PLAYERS(PlayerId INT,NameFirst VARCHAR(30),NameLast VARCHAR(30),CurrentHandle VARCHAR(40),CountryCode VARCHAR(30),TotalUSDPrize VARCHAR(30),Game varchar(60));
Query OK, 0 rows affected (0.02 sec)

mysql> LOAD DATA INFILE "D:/esports sql project/highest earning players.csv" INTO table HIGHEST_EARNING_PLAYERS FIELDS TERMINATED BY ',' IGNORE 1 LINES;
ERROR 1366 (HY000): Incorrect integer value: '' for column 'PlayerId' at row 929
mysql> LOAD DATA INFILE "D:/esports sql project/highest earning players.csv" INTO table HIGHEST_EARNING_PLAYERS FIELDS TERMINATED BY ',' IGNORE 1 LINES;
Query OK, 928 rows affected (0.02 sec)
Records: 928 Deleted: 0 Skipped: 0 Warnings: 0

mysql> select * from highest_earning_players limit 40;
```

PlayerId	NameFirst	NameLast	CurrentHandle	CountryCode	TotalUSDPrize	Game
3883	Peter	Rasmussen	dupreeh	dk	1822989.41	Counter-Strike: Global Offensive
3679	Andreas	Højsleth	Xyp9x	dk	1799288.57	Counter-Strike: Global Offensive
3885	Nicolai	Reedtz	dev1ce	dk	1787489.88	Counter-Strike: Global Offensive
3672	Lukas	Rossander	glalve	dk	1652350.75	Counter-Strike: Global Offensive
17800	Emil	Reif	Magisk	dk	1416448.64	Counter-Strike: Global Offensive
16800	Jakey	Yip	Stewie2k	us	1087340	Counter-Strike: Global Offensive
12183	Epitácio	de Melo	TACO	br	1063858.27	Counter-Strike: Global Offensive
12169	Fernando	Alvarenga	fer	br	1063038.92	Counter-Strike: Global Offensive
2455	Gabriel	Toledo	FalleN	br	1059938.92	Counter-Strike: Global Offensive
12182	Marcelo	David	coldzera	br	1021901.46	Counter-Strike: Global Offensive
10629	Keith	Markovic	NAF	ca	982765.66	Counter-Strike: Global Offensive
2452	Finn	Andersen	karrigan	dk	964635.39	Counter-Strike: Global Offensive
11788	Jonathan	Jablonowski	ELiGE	us	930696.42	Counter-Strike: Global Offensive
8635	Nick	Cannella	nitr0	us	920151.73	Counter-Strike: Global Offensive
3875	Jesper	Wecksell	JW	se	897760.68	Counter-Strike: Global Offensive
5001	Olof	Kajbjerg	oLofermeister	se	880011.52	Counter-Strike: Global Offensive
3878	Robin	Rönquist	flusha	se	877668.95	Counter-Strike: Global Offensive
5000	Freddy	Johansson	KRIMZ	se	867823.34	Counter-Strike: Global Offensive
20415	Russel	Van Dulken	Twistzz	ca	835376.43	Counter-Strike: Global Offensive
3888	Dan	Madesclaire	apEX	fr	814852.39	Counter-Strike: Global Offensive
3290	Nathan	Schmitt	NBK	fr	805444.66	Counter-Strike: Global Offensive
3882	Ladislav	Kovács	GuardiaN	sk	798520.8	Counter-Strike: Global Offensive
10630	Tarik	Celik	tarik	us	780039.21	Counter-Strike: Global Offensive
5483	Håvard	Nyggaard	rain	no	763799.57	Counter-Strike: Global Offensive
5783	Nikola	Kova?	NiKo	ba	762802.56	Counter-Strike: Global Offensive
8168	Egor	Vasilyev	flamie	ru	732920.07	Counter-Strike: Global Offensive

### Create table Query:

```
CREATE TABLE HIGHEST_EARNING_PLAYERS(PlayerId INT,NameFirst
VARCHAR(30),NameLast VARCHAR(30),CurrentHandle VARCHAR(40),CountryCode
VARCHAR(30),TotalUSDPrize VARCHAR(30),Game varchar(60));
```

### Load Data Command:

```
LOAD DATA INFILE "D:/esports sql project/highest earning players.csv" INTO table
HIGHEST_EARNING_PLAYERS FIELDS TERMINATED BY ',' IGNORE 1 lines
```

**2) Create table `highest_earning_teams` and load CSV file into SQL having all the entries. Also, show the command for the load data into a file.**

```
mysql> use esports;
Database changed
mysql> CREATE TABLE HIGHEST_EARNING_PLAYERS(PlayerId INT,NameFirst VARCHAR(30),NameLast VARCHAR(30),CurrentHandle VARCHAR(40),CountryCode VARCHAR(30),TotalUSDPrize VARCHAR(30),Game varchar(60));
Query OK, 0 rows affected (0.02 sec)

mysql> LOAD DATA INFILE "D:/esports sql project/highest earning players.csv" INTO table HIGHEST_EARNING_PLAYERS FIELDS TERMINATED BY ',' IGNORE 1 LINES;
ERROR 1366 (HY000): Incorrect integer value: '' for column 'PlayerId' at row 929
mysql> LOAD DATA INFILE "D:/esports sql project/highest earning players.csv" INTO table HIGHEST_EARNING_PLAYERS FIELDS TERMINATED BY ',' IGNORE 1 LINES;
Query OK, 928 rows affected (0.02 sec)
Records: 928 Deleted: 0 Skipped: 0 Warnings: 0
```

**Create table query:**

create table `highest_earning_teams`(PlayerId int,TeamId int,TeamName varchar(40),TotalUSDPrize varchar(30),TotalTournaments varchar(40),Games varchar(50));

**Load Data Command:**

LOAD DATA INFILE "D:/esports sql project/highest earning teams.csv" INTO table `HIGHEST_EARNING_TEAMS` FIELDS TERMINATED BY ',' IGNORE 1 LINES;

**3) Display the first 20 records of Highest earning players.**

```
mysql> select * from highest_earning_players limit 20;
```

PlayerId	NameFirst	NameLast	CurrentHandle	CountryCode	TotalUSDPrize	Game
3883	Peter	Rasmussen	dupreeh	dk	1822989.41	Counter-Strike: Global Offensive
3679	Andreas	Højsleth	Xyp9x	dk	1799288.57	Counter-Strike: Global Offensive
3885	Nicolai	Reedtz	device	dk	1787489.88	Counter-Strike: Global Offensive
3672	Lukas	Rossander	glalve	dk	1652350.75	Counter-Strike: Global Offensive
17800	Emil	Reif	Magisk	dk	1416448.64	Counter-Strike: Global Offensive
16800	Jakey	Yip	Stewie2k	us	1087340	Counter-Strike: Global Offensive
12183	Epitácio	de Melo	TACO	br	1063858.27	Counter-Strike: Global Offensive
12169	Fernando	Alvarenga	fer	br	1063038.92	Counter-Strike: Global Offensive
2455	Gabriel	Toledo	FalleN	br	1059938.92	Counter-Strike: Global Offensive
12182	Marcelo	David	coldzera	br	1021901.46	Counter-Strike: Global Offensive
10629	Keith	Markovic	NAF	ca	982765.66	Counter-Strike: Global Offensive
2452	Finn	Andersen	karrigan	dk	964635.39	Counter-Strike: Global Offensive
11788	Jonathan	Jablonowski	ELiGE	us	930696.42	Counter-Strike: Global Offensive
8635	Nick	Cannella	nitr0	us	920151.73	Counter-Strike: Global Offensive
3875	Jesper	Wecksell	JW	se	897760.68	Counter-Strike: Global Offensive
5001	Olof	Kajbjer	olofmeister	se	880011.52	Counter-Strike: Global Offensive
3878	Robin	Rönnquist	flusha	se	877668.95	Counter-Strike: Global Offensive
5000	Freddy	Johansson	KRiMZ	se	867823.34	Counter-Strike: Global Offensive
20415	Russel	Van Dulken	Twistzz	ca	835376.43	Counter-Strike: Global Offensive
3888	Dan	Madesclaire	apEX	fr	814852.39	Counter-Strike: Global Offensive

20 rows in set (0.00 sec)

Query: select \* from `highest_earning_players` limit 20;

#### 4) Display first 30 records from highest earning teams.

```
mysql> select * from highest_earning_teams limit 30;
```

PlayerId	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Games
3883	760	San Francisco Shock	3105000	7	Overwatch
3679	776	London Spitfire	1591136.5	13	Overwatch
3885	768	New York Excelsior	1572618.5	18	Overwatch
3672	773	Philadelphia Fusion	1186278.5	15	Overwatch
17800	766	Seoul Dynasty	1130000	6	Overwatch
16800	856	Vancouver Titans	950000	4	Overwatch
12183	769	Shanghai Dragons	755000	5	Overwatch
12169	774	Los Angeles Gladiators	709605.19	13	Overwatch
2455	861	Atlanta Reign	596098	9	Overwatch
12182	770	Los Angeles Valiant	535000	6	Overwatch
10629	216	Team Envy	504391.4	32	Overwatch
2452	860	Hangzhou Spark	425000	3	Overwatch
11788	732	RunAway	420808.84	19	Overwatch
8635	613	Lunatic-hai	320633.66	10	Overwatch
3875	771	Boston Uprising	288606	7	Overwatch
5001	765	GC Busan	262117.51	10	Overwatch
3878	863	Washington Justice	250000	1	Overwatch
5000	557	Rogue	233623.47	22	Overwatch
20415	808	Team Gigantti	224542.69	13	Overwatch
3888	775	Florida Mayhem	212672.5	8	Overwatch
3290	529	Misfits	172484.12	16	Overwatch
3882	227	LGD Gaming	165687.71	14	Overwatch
10630	24950	Osh-Tekk Warriors	161534	2	Overwatch
5483	24708	Talon Esports	159952	7	Overwatch
5783	184	Flash Wolves	157081.66	3	Overwatch
8168	762	Miraculous Youngster	148594.21	7	Overwatch
3289	482	Team Kongdoo	145971.95	15	Overwatch
1511	212	Cloud9	133251.4	36	Overwatch
8169	734	Blank Esports	132105.66	7	Overwatch

Query: select \* from highest\_earning\_teams limit 30;

#### 5) Display the maximum USD prize for highest earning players along with the first name and last name.

```
mysql> select max(TotalUSDPrize),NameFirst,NameLast from highest_earning_players;
```

max(TotalUSDPrize)	NameFirst	NameLast
991583.2	Peter	Rasmussen

1 row in set (0.01 sec)

Query: select max(TotalUSDPrize),NameFirst,NameLast from highest\_earning\_players;

**Q 6) Display second highest USD prize of the highest earning teams along with team name.**

```
mysql> select max(TotalUSDPrize),TeamName from highest_earning_teams where TotalUSDPrize<(select max(TotalUSDPrize)from highest_earning_teams);
```

max(TotalUSDPrize)	TeamName
99817.22	San Francisco Shock

1 row in set (0.01 sec)

Query: select max(TotalUSDPrize),TeamName from highest\_earning\_teams where TotalUSDPrize<(select max(TotalUSDPrize)from highest\_earning\_teams);

**Q 7) Count total tournaments along with the team name.**

```
mysql> select count(TotalTournaments),TeamName from highest_earning_teams;
```

count(TotalTournaments)	TeamName
925	San Francisco Shock

1 row in set (0.00 sec)

Query: select count (TotalTournaments),TeamName from highest\_earning\_teams;

**Q 8) Display the list of the games of highest-earning player.**

```
mysql> select games from highest_earning_players limit 10;
ERROR 1054 (42S22): Unknown column 'games' in 'field list'
mysql> select game from highest_earning_players limit 10;
```

game
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive
Counter-Strike: Global Offensive

10 rows in set (0.00 sec)

Query: select game from highest\_earning\_players limit 10;



**Q 9) delete one row whose teamname is Lunatic-hai.**

```
mysql> delete from highest_earning_teams where TeamName="Lunatic-hai";
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from highest_earning_team limit 15;
ERROR 1146 (42S02): Table 'esports.highest_earning_team' doesn't exist
mysql> select * from highest_earning_teams limit 15;
```

PlayerId	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Games
3883	760	San Francisco Shock	3105000	7	Overwatch
3679	776	London Spitfire	1591136.5	13	Overwatch
3885	768	New York Excelsior	1572618.5	18	Overwatch
3672	773	Philadelphia Fusion	1186278.5	15	Overwatch
17800	766	Seoul Dynasty	1130000	6	Overwatch
16800	856	Vancouver Titans	950000	4	Overwatch
12183	769	Shanghai Dragons	755000	5	Overwatch
12169	774	Los Angeles Gladiators	709605.19	13	Overwatch
2455	861	Atlanta Reign	596098	9	Overwatch
12182	770	Los Angeles Valiant	535000	6	Overwatch

**Query:** select \* from highest\_earning\_teams limit 15;

**Q 10) Display game wise average of the total USD prize.**

```
mysql> select avg(TotalUSDPrize) from highest_earning_teams group by games;
```

avg(TotalUSDPrize)
203481.1515151515
278561.4524
646655.6092000002
399991.3494000001
1101606.5147999993
1905938.8331999993
140098.3391
129709.09779999999
147094.17050000004
117768.83844444444

```
10 rows in set (0.01 sec)
```

**Query:** select avg(TotalUSDPrize) from highest\_earning\_teams group by games;



### Q 11) Perform inner join between two tables.

```
mysql> select highest_earning_players.PlayerID,NameFirst,TeamID from highest_earning_players inner join highest_earning_teams on highest_earning_players.PlayerID=highest_earning_teams.PlayerId;
```

PlayerID	NameFirst	TeamID
3883	Peter	760
3679	Andreas	776
3885	Nicolai	768
3672	Lukas	773
17860	Emil	766
16800	Jakey	856
12183	Epitácio	769
12169	Fernando	774
2455	Gabriel	861
12182	Marcelo	770
10629	Keith	216
2452	Finn	860
11788	Jonathan	732
8635	Nick	613
3875	Jesper	771
5001	Olof	765
3878	Robin	863
5000	Freddy	557
20415	Russel	808
3888	Dan	775
3290	Nathan	529
3882	Ladislav	227
10630	Tarik	24950
5483	Håvard	24708
5783	Nikola	184
8168	Egor	762
3289	Kenny	482
1511	Daniil	212
8169	Alexander	734
3881	Richard	767
4998	Janusz	425
10736	Markus	704
1510	Ioann	468
1485	Filip	687
3951	Chris	411
1484	Wiktor	169
4311	Will	851
11888	Youssef	816

**Query:** select highest\_earning\_players.PlayerID, NameFirst,TeamID from Highest\_earning\_players inner join highest\_earning\_teams on highest\_earning\_players.PlayerID=highest\_earning\_teams.PlayerID;

### Q 12) Perform left join between two tables.

```
mysql> select NameFirst,TeamName,TotalTournaments,CurrentHandle from highest_earning_players left join highest_earning_teams on highest_earning_players.PlayerID=highest_earning_teams.PlayerID;
```

NameFirst	TeamName	TotalTournaments	CurrentHandle
Peter	San Francisco Shock	7	dupreeh
Andreas	London Spitfire	13	Xyp9x
Nicolai	New York Excelsior	18	dev1ce
Lukas	Philadelphia Fusion	15	glaive
Emil	Seoul Dynasty	6	Magisk
Jakey	Vancouver Titans	4	Stewie2k
Epitácio	Shanghai Dragons	5	TACO
Fernando	Los Angeles Gladiators	13	fer
Gabriel	Atlanta Reign	9	FalleN
Marcelo	Los Angeles Valiant	6	coldzera
Keith	Team Envy	32	NAF
Finn	Hangzhou Spark	3	karrigan
Jonathan	RunAway	19	ELiGE
Nick	NULL	NULL	nitr0
Jesper	Boston Uprising	7	JW
Olof	GC Busan	10	olofmeister
Robin	Washington Justice	1	flusha
Freddy	Rogue	22	KRiMz
Russel	Team Gigantti	13	Twistzz
Dan	Florida Mayhem	8	apEX
Nathan	Misfits	16	NBK
Ladislav	LGD Gaming	14	GuardiaN
Tarik	Osh-Tekk Warriors	2	tarik

**Query:** select NameLast, TeamName, TotalTournaments, CurrentHandle from highest\_earning\_players left join highest\_earning\_teams on highest\_earning\_players.PlayerID=highest\_earning\_teams.PlayerID;

**Q 13) display the top 3 earning teams according to the total USD prize.**

```
mysql> SELECT * FROM highest_earning_teams ORDER By TotalUSDPrize DESC limit 3;
```

PlayerId	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Game
2581	315	Invasion eSport	99897.23	125	Starcraft II
12671	726	OSC Elite	99817.22	147	Starcraft II
19074	24678	Brave Star Gaming	99294.24	99	Starcraft II

3 rows in set (0.00 sec)

Select \* from highest\_earning\_teams order by TotalUSDPrize Desc limit 3;

**Q 14) find the second-highest team according to the tournament.**

```
mysql> select TeamName,max(TotalTournaments)from highest_earning_teams where TotalTournaments<(select max(TotalTournaments)
from highest_earning_teams);
```

TeamName	max(TotalTournaments)
San Francisco Shock	98

1 row in set (0.00 sec)

Query: select TeamName,max(TotalTournaments)from highest\_earning\_teams where TotalTournaments<(select max(TotalTournaments) from highest\_earning\_teams);

**Q 15) Perform Right Join Between Two Tables.**

```
mysql> select NameLast, TeamName, TotalTournaments, CurrentHandle from highest_earning_players right join highest_earning_teams on highest_earning_players.
PlayerId=highest_earning_teams.PlayerId;
```

NameLast	TeamName	TotalTournaments	CurrentHandle
Rasmussen	San Francisco Shock	7	dupreeh
Hejsleth	London Spitfire	13	Xyp9x
Reedtz	New York Excelsior	18	device
Rossander	Philadelphia Fusion	15	glalve
Reif	Seoul Dynasty	6	Magisk
Yip	Vancouver Titans	4	Stewie2k
de Melo	Shanghai Dragons	5	TACO
Alvarenga	Los Angeles Gladiators	13	fer
Toledo	Atlanta Reign	9	FalleN
David	Los Angeles Valiant	6	coldzera
Markovic	Team Envy	32	NAF
Andersen	Hangzhou Spark	3	karrigan
Jablonowski	RunAway	19	ELiGE
Wecksell	Boston Uprising	7	JW
Kajbjer	GC Busan	10	oLoFmeister
Rönnquist	Washington Justice	1	flusha
Johansson	Rogue	22	KRiMz
Van Dulken	Team Gigantti	13	Twistzz
Madesclaire	Florida Mayhem	8	apEX
Schmitt	Misfits	16	NBK
Kovács	LGD Gaming	14	GuardiaN
Celik	Osh-Tekk Warriors	2	tarik
Nygaard	Talon Esports	7	rain
Kovač	Flash Wolves	3	NiKo
Vasilyev	Miraculous Youngster	7	flamie
Schrub	Team Kongdoo	15	kennyS
Teslenko	Cloud9	36	Zeus
Kostylev	Blank Esports	7	sImple
Papillon	Houston Outlaws	4	shox
Pogorzelski	NRG Esports	10	snax
Kjærbye	Hong Kong Attitude	6	Kjaerbye

Query: select NameLast, TeamName, TotalTournaments, CurrentHandle from  
highest\_earning\_players right join highest\_earning\_teams on highest\_earning\_players.  
PlayerId=highest\_earning\_teams.PlayerId;

- **Final Conclusion of Analysis:**

The main aim of my Analysis is to display the top 3 highest earning teams. Following is the SQL query that matches the aim of analysis:

Question: **display the top 3 earning teams according to the total USD prize.**

```
mysql> SELECT * FROM highest_earning_teams ORDER By TotalUSDPrize DESC limit 3;
```

PlayerId	TeamId	TeamName	TotalUSDPrize	TotalTournaments	Game
2581	315	Invasion eSport	99897.23	125	Starcraft II
12671	726	OSC Elite	99817.22	147	Starcraft II
19074	24678	Brave Star Gaming	99294.24	99	Starcraft II

3 rows in set (0.00 sec)

Query: Select \* from highest\_earning\_teams order by TotalUSDPrize Desc limit 3;

Note for himani:( mention this on sepearate slide)

As per the Prescriptive Analysis , the result displayed below suggests the second-highest team according to the tournaments.

Question: **find the second-highest team according to the tournament.**

```
mysql> select TeamName,max(TotalTournaments)from highest_earning_teams where TotalTournaments<(select max(TotalTournaments)
from highest_earning_teams);
```

TeamName	max(TotalTournaments)
San Francisco Shock	98

1 row in set (0.00 sec)

Query :select TeamName, max(TotalTournaments) from highest\_earning\_teams where  
TotalTournaments < (select max(TotalTournaments) from highest\_earning\_teams);

- **References:**

[www.kaggle.com](http://www.kaggle.com)

[www.w3schools.com](http://www.w3schools.com)