# Description

I made a database that represents *"Counter Strike : Global Offensive"* Esports scene as a high level overview. It can be used by organizing companies to manage events, team managing companies to manage their teams or as overall publicly available data for those who make money on eSports.

This database contains 8 tables.

**PLAYERS** which contains all of the information about the players, such as their full name, in-game name, nationality, age, id of the team they are playing in and their total winnings throughout career. Player does not have to be currently in the team. I assume every player will have their unique id created, when inserted into a database.

**TEAMS** which contains all of the information about the teams on the scene. It has team name, team nationality (usually based of nationalities of the majority of the players, but not always), a year of establishment, a city of their main headquarters, id of their main sponsor if any, id of their current coach if any and a number of tournaments won since establishment. I assume every team is given it's unique id to it when inserted into database.

**LEAGUE_HOST_COMPANIES** which has information about tournament organizers. It contains name of the company, year of official establishment, number of employees and country of origin of the company. I assume all organizing companies are given unique id when inserted into database.

**COACHES** which provides information about coaches in the teams. It can provide information about coaches full name, years of experience, an id of the team, in which they are currently coaching if they are employed, and their salary if the coach is active. I assume every coach gets a unique id when he enters the scene.

**TOURNAMENTS** contains information about scheduled tournaments, future, current and past. It contains tournament name, date, id of the organizer, prize pool in dollars, type of the tournament, which can be Major, Minor or Regional tournament. It also shows if tournament is played offline or offline and what is the bracket type of the tournament.

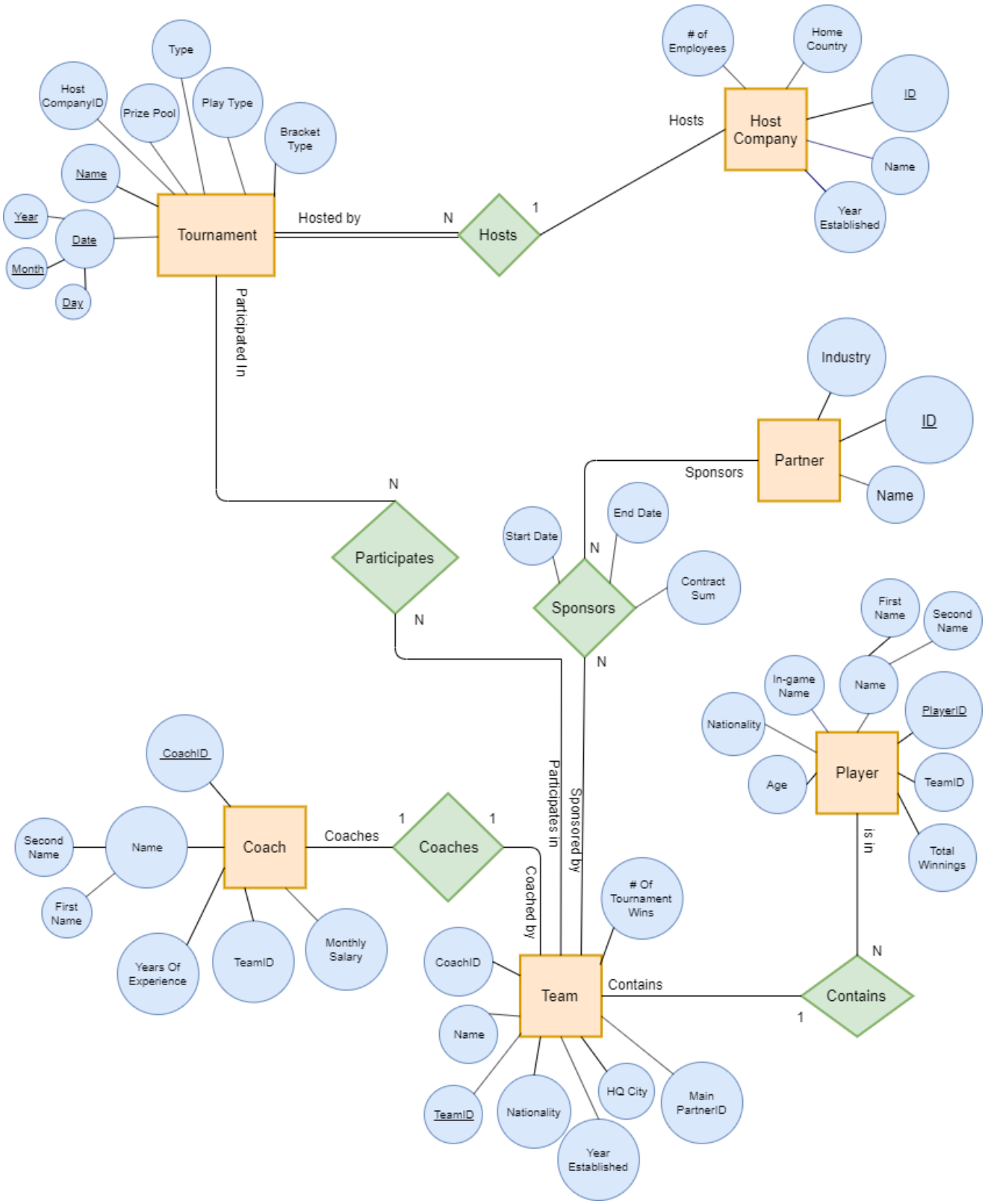
**PARTNERS** contains partnering companies that sponsor teams. We can see their name, industry that they specialize in and the unique id, which they should get when inserted in a database.

The last two table were created for normalization and to reduce redundancy.

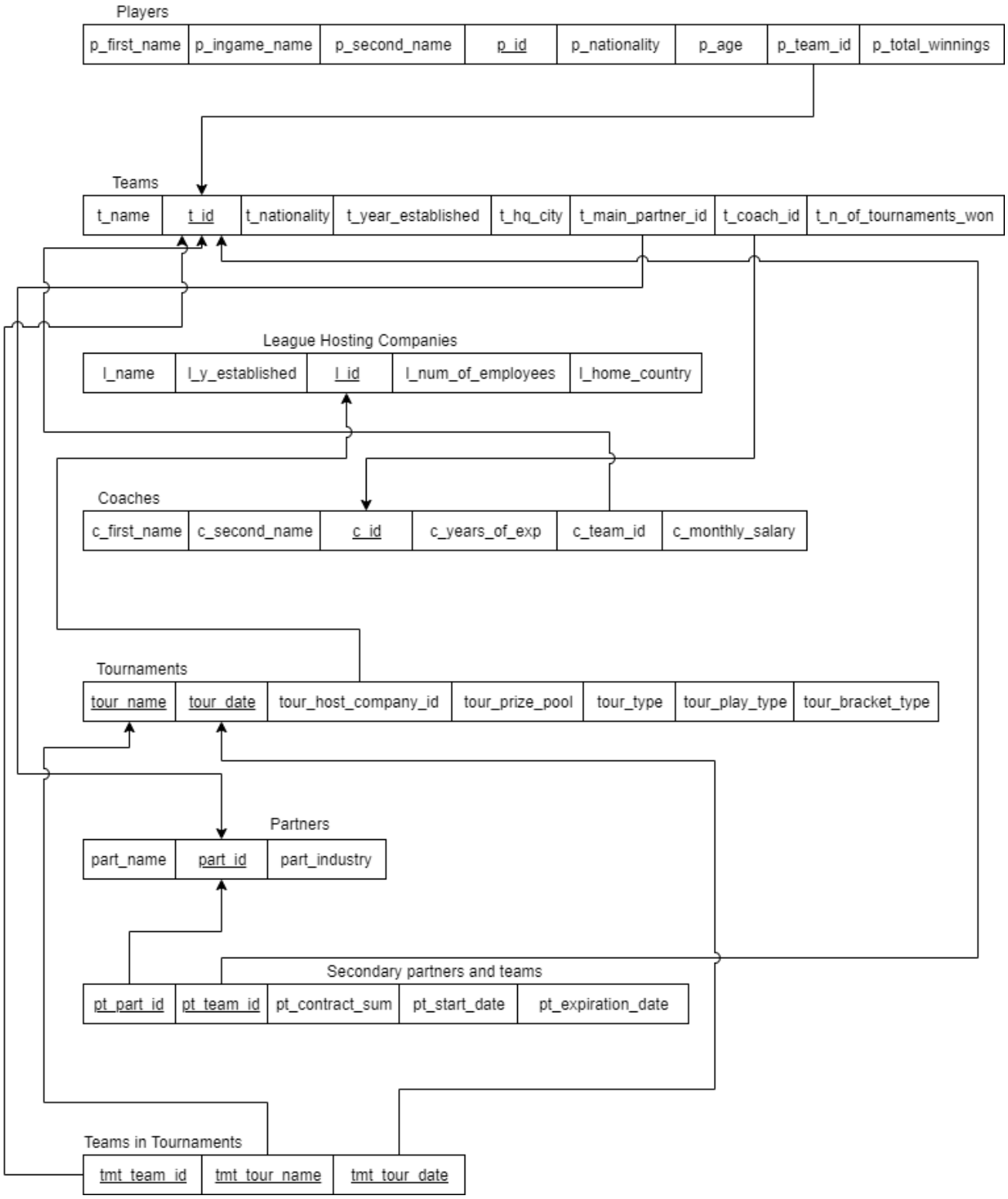
**SECONDARY_PARTNERS_AND_TEAMS** is created since team can have multiple partners. TEAMS contains main partner id, but this table represents the rest of them. It contains team id and partner id, which are partnered with each other. It also contains date of the start of the partnership and it's expiration date. We can also see the sum that team receives upon this partnership.

**TEAMS_IN_TOURNAMENTS** shows us the teams that are registered to participate in certain tournaments. It contains name and the date of the tournament as well as it has id of the team that is participating in this tournament.

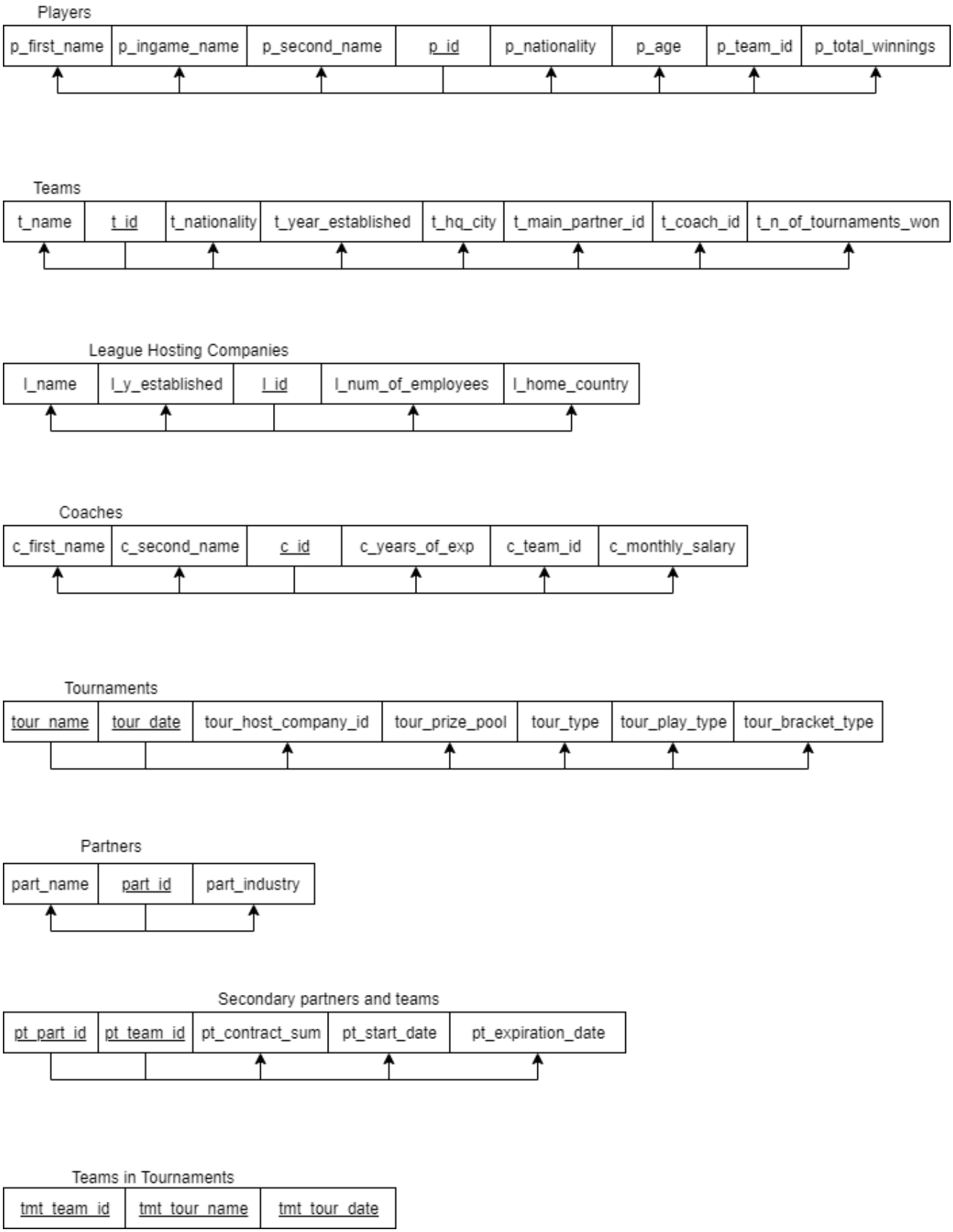
# Entity Relationship Diagram



# Mapping to Relational Schema



# *Functional Dependency Diagram*



# *Semantic Constraints*

## Key constraints

Every Table in the has a **primary key defined**, that is **unique**. For most of the cases it's a **unique id**.

In terms of tournaments its a composite primary key consisting of the name of the tournament and the date. In TEAMS_IN_TOURNAMENTS and SECONDARY_PARTNERS_AND_TEAMS there are also **composite primary keys**. For former it's a combination of teamID and tournament name and date, for latter its team and partner ids.

## Entity Integrity constraints

It is ensured that **none of the Primary keys** in the tables can be **NULL**.

## Referential Integrity Constraints

Most of the **foreign keys are allowed to have NULL value**, since our schema allows it. However, we ensure that foreign keys are **always referring to existing tuples**. In case if referred tuple is deleted or updated, we make use of **ON UPDATE** and **ON DELETE** with occasional **CASCADE** clauses to either delete a tuple or update it with a new value, or set foreign key to NULL.

## Additional Constraints

In addition to constraints above, **all IDs have to be positive numbers**. This is ensured through series of commands:

**ALTER TABLE PLAYERS ADD CHECK(p_id>0);**

**ALTER TABLE TEAMS ADD CHECK(t_id>0);**

**ALTER TABLE LEAGUES_HOST_COMPANIES ADD CHECK(l_id>0);**

**ALTER TABLE COACHES ADD CHECK(c_id>0);**

**ALTER TABLE PARTNERS ADD CHECK(part_id>0);**

There are also a couple of **CHECKs** specified, where only certain values can be accepted.

Examples in **TOURNAMENTS**:

**CHECK(tour_type IN('Major','Minor','Regional')** – tournaments officially can only be in of these three categories, therefore we restrict possible values to them.

**CHECK(tour_play_type IN('Online','Offline')** - tournaments can be played online or offline, so it makes sense to accept only these two values.

**CHECK(tour_bracket_type IN('Round Robin','Single Elimination','Swiss','Double Elimination')** – Bracket type varies from tournament to tournament, but for official tournaments only these four are accepted, therefore we restrict possible values there.

# *Security and views*

In terms of security, for this database roles would work the best. Certain roles will have permission to view or modify tables only that are relevant to them. Some of the roles will also have restricted views on tables through **VIEWs** created for them.

Example roles would be:

- **CREATE ROLE Coach;**

Since coach only needs to see the team members that he is coaching, we create a special view for him, that will only show to him the players that are in the **same team** as coach is. **We will assume out coach is in team with id = 1.**

**CREATE VIEW Coach_View(f_name, nickname, s_name, id, nationality, age, team_id) AS SELECT p_first_name, p_ingame_name, p_second_name, p_id, p_nationality, p_age, p_team_id FROM PLAYERS WHERE 1 = p_team_id;**

**GRANT SELECT ON Coach_View TO Coach;**

- **CREATE ROLE Player;**

Player will technically only need to see other players in the scene, to know who his opponents are and what team they belong to. Therefore we grant them read permissions for these two tables.

**GRANT SELECT ON PLAYERS TO Player;**

**GRANT SELECT ON TEAMS TO Player;**

- **CREATE ROLE Event_Organiser;**

Event organizer would have a special view on all the teams and their information, as long as they are participating in their tournaments. They would be able to remove them from the tournament or update the information about them if needed. **We will assume out league host id is 1115.**

**CREATE VIEW Host_view(team_name, team_id, team_nationality, team_year_established, t_main_partner_id, team_coach_id, team_n_of_tournaments_won) AS SELECT DISTINCT t_name, t_id, t_nationality, t_year_established, t_main_partner_id, t_coach_id, t_n_of_tournaments_won FROM TEAMS, TEAMS_IN_TOURNAMENTS, LEAGUES_HOST_COMPANIES, TOURNAMENTS WHERE t_id = tmt_team_id AND tmt_tour_name IN (SELECT tour_name FROM TOURNAMENTS, LEAGUES_HOST_COMPANIES WHERE tour_host_company_id = 1115) AND tmt_tour_date IN (SELECT tour_date FROM TOURNAMENTS, LEAGUES_HOST_COMPANIES WHERE tour_host_company_id = 1115);**

**GRANT ALL ON Host_view TO Event_Organiser;**

We can also specify specific tournament if needed.

- **CREATE ROLE Analytics;**

We might want to create some of the more interesting roles. Analytics for example could be someone who does research on how much players earn from tournaments on average, but didn't want to access personal data of the players. Therefore, we can grant him the **view on players winnings, without access to other attributes**.

**CREATE VIEW Analytics_view(total_winnings) AS SELECT p_total_winnings FROM PLAYERS;**

**GRANT SELECT ON Analytics_view TO Analytics;**

# Triggers

We also have two triggers in our database to **preserve Referential Integrity** in some of the cases.

**CREATE TRIGGER Team_deletion_update_tournaments**

**AFTER DELETE ON TEAMS**

**FOR EACH ROW**

**BEGIN**

**DECLARE**

**deleted_t_id int;**

**IF (OLD.t_id IS NOT NULL) THEN**

**SET @deleted_t_id := OLD.t_id;**

**DELETE FROM TEAMS_IN_TOURNAMENTS**

**WHERE tmt_team_id=deleted_t_id;**

**END IF;**

**END$$**

This one will help if we delete one of the teams that was scheduled to participate in a tournament. To preserve Referential and Entity integrity, we would **delete all the rows, that would include the id of deleted teams**.

```
CREATE TRIGGER Team_deletion_update_sponsors
AFTER DELETE ON TEAMS
FOR EACH ROW
BEGIN
DECLARE
        deleted_t_id int;
IF (OLD.t_id IS NOT NULL) THEN
        SET @deleted_t_id := OLD.t_id;
        DELETE FROM SECONDARY_PARTNERS_AND_TEAMS
        WHERE  pt_team_id=deleted_t_id;
END IF;
END$$
```

This one would work similar to tournament one, with the difference being the table that we would delete rows in. If team is disbanded and removed from the eSports scene, it loses all partnerships. Therefore, to preserve DB integrity, we **remove all rows from SECONDARY_PARTNERS_AND_TEAMS that has deleted team id in it.**

# *Graphical Examples*

Here are some examples from querying the database:

This is the result we would get by running

**SELECT * FROM TEAMS;**

| t_name | t_id | t_nationality | t_year_established | t_hq_city | t_main_partner_id | t_coach_id | t_n_of_tournaments_won |
|---|---|---|---|---|---|---|---|
| Astralis | 1 | Danish | 2016 | Copenhagen | 10000 | 100000 | 7 |
| Virtus Pro | 2 | Polish | 2003 | Moscow | 10001 | 100001 | 6 |
| SK Gaming | 3 | German | 2010 | Cologne | 10002 | 100002 | 7 |
| fnatic | 4 | Swedish | 2004 | London | 10003 | 100003 | 10 |
| Team Liquid | 5 | American | 2000 | Santa Monica | 10004 | 100004 | 5 |
| Natus Vincere | 6 | Ukrainian | 2009 | Kyiv | 10005 | 100005 | 6 |

This is partial result (since we have 30 players) from running

**SELECT * FROM PLAYERS;**

| p_first_name | p_ingame_name | p_second_name | p_id | p_nationality | p_age | p_team_id | p_total_winnings |
|---|---|---|---|---|---|---|---|
| Nicolai | dev1ce | Reedtz | 10 | Danish | 24 | 1 | 1590501 |
| Peter | dupreeh | Rasmussen | 11 | Danish | 26 | 1 | 1621123 |
| Andreas | Xyp9x | Højsleth | 12 | Danish | 24 | 1 | 1624934 |
| Lukas | gla1ve | Rossander | 13 | Danish | 24 | 1 | 1323552 |
| Emil | Magisk | Reif | 14 | Danish | 21 | 1 | 1203818 |
| Janusz | Snax | Pogorzelski | 15 | Polish | 26 | 2 | 637630 |
| Paweł | byali | Bieliński | 16 | Polish | 25 | 2 | 570881 |
| Jarosław | pashaBiceps | Jarząbkowski | 17 | Polish | 31 | 2 | 620576 |
| Filip | NEO | Kubski | 18 | Polish | 32 | 2 | 743495 |
| Wiktor | TaZ | Wojtas | 19 | Polish | 33 | 2 | 725958 |
| Gabriel | FalleN | Toledo | 20 | Brazilian | 28 | 3 | 948740 |
| Fernando | fer | Alvarenga | 21 | Brazilian | 28 | 3 | 935596 |
| Marcelo | coldzera | David | 22 | Brazilian | 25 | 3 | 942699 |
| Ricardo | boltz | Prass | 23 | Brazilian | 22 | 3 | 344150 |
| Freddy | KRIMZ | Johansson | 24 | Swedish | 25 | 4 | 782580 |
| Jesper | JW | Wecksell | 25 | Swedish | 24 | 4 | 812521 |
| Ludvig | Brollan | Brolin | 26 | Swedish | 17 | 4 | 140437 |
| Robin | flusha | Rönnquist | 27 | Swedish | 26 | 4 | 786441 |
| Richard | Xizt | Landström | 28 | Swedish | 28 | 4 | 541198 |
| Nicholas | nitr0 | Cannella | 29 | United States | 24 | 5 | 838332 |
| Jonathan | EliGE | Jablonowski | 30 | United States | 22 | 5 | 829482 |
| Russel | Twistzz | Van Dulken | 31 | Canadian | 19 | 5 | 404272 |
| Keith | NAF | Markovic | 32 | Canadian | 21 | 5 | 882649 |
| Jacky | "Stewie2K | Yip | 33 | United States | 21 | 5 | 979640 |
| Epitácio | TACO | de Melo | 34 | Brazilian | 24 | 3 | 944996 |

Here is the result from running

**SELECT * FROM COACHES;**

| c_first_name | c_second_name | c_id | c_years_of_exp | c_team_id | c_monthly_salary |
|---|---|---|---|---|---|
| Danny | Sørensen | 100000 | 3 | 1 | 22000 |
| Jakub | Gurczyński | 100001 | 5 | 2 | 24000 |
| Wilton | Prado | 100002 | 3 | 3 | 7000 |
| Andreas | Samuelsson | 100003 | 0 | 4 | 23000 |
| Eric | Hoag | 100004 | 1 | 5 | 20000 |
| Andrey | Gorodenskiy | 100005 | 5 | 6 | 9000 |

To give an idea of some views as well, here is the result from running

**SELECT * FROM host_view;**

Assuming we are host with **l_id = 1115**

| team_name | team_id | team_nationality | team_year_established | t_main_partner_id | team_coach_id | team_n_of_tournaments_won |
|---|---|---|---|---|---|---|
| Virtus Pro | 2 | Polish | 2003 | 10001 | 100001 | 6 |
| fnatic | 4 | Swedish | 2004 | 10003 | 100003 | 10 |
| Natus Vincere | 6 | Ukrainian | 2009 | 10005 | 100005 | 6 |

And this is the result of

**SELECT * FROM coach_view;** assuming we are querying **as a coach of the team 1**.

| f_name | nickname | s_name | id | nationality | age | team_id |
|---|---|---|---|---|---|---|
| Nicolai | dev1ce | Reedtz | 10 | Danish | 24 | 1 |
| Peter | dupreeh | Rasmussen | 11 | Danish | 26 | 1 |
| Andreas | Xyp9x | Højsleth | 12 | Danish | 24 | 1 |
| Lukas | gla1ve | Rossander | 13 | Danish | 24 | 1 |
| Emil | Magisk | Reif | 14 | Danish | 21 | 1 |

# *Appendix*

```
CREATE DATABASE csgo_database;
USE csgo_database;


CREATE table PLAYERS(
    p_first_name NATIONAL VARCHAR(30) NOT NULL,
    p_ingame_name VARCHAR(20) NOT NULL,
    p_second_name NATIONAL VARCHAR(30) NOT NULL,
    p_id INT NOT NULL PRIMARY KEY,
    p_nationality VARCHAR(20) NOT NULL,
    p_age INT NOT NULL,
    p_team_id INT,
    p_total_winnings INT
);
CREATE table TEAMS(
    t_name VARCHAR(20) NOT NULL,
    t_id INT NOT NULL PRIMARY KEY,
    t_nationality VARCHAR(20),
    t_year_established INT NOT NULL,
    t_hq_city VARCHAR(20),
    t_main_partner_id INT,
    t_coach_id INT,
    t_n_of_tournaments_won INT
);
```

```
CREATE table LEAGUES_HOST_COMPANIES(
    l_name VARCHAR(20) NOT NULL,
    l_y_established INT NOT NULL,
    l_id INT NOT NULL PRIMARY KEY,
    l_num_of_employees INT NOT NULL,
    l_home_country VARCHAR(15)
);

CREATE table COACHES(
    c_first_name NATIONAL VARCHAR(30) NOT NULL,
    c_second_name NATIONAL VARCHAR(30)  NOT NULL,
    c_id INT NOT NULL PRIMARY KEY,
    c_years_of_exp INT NOT NULL,
    c_team_id INT,
    c_monthly_salary INT
);

CREATE table TOURNAMENTS(
    tour_name VARCHAR(50) NOT NULL,
    tour_date DATE NOT NULL,
    tour_host_company_id INT NOT NULL,
    tour_prize_pool INT NOT NULL,
    tour_type VARCHAR(10) NOT NULL CHECK(tour_type IN('Major','Minor','Regional')),
    tour_play_type VARCHAR(7) NOT NULL CHECK(tour_play_type IN('Online','Offline')),
    tour_bracket_type VARCHAR(30) NOT NULL CHECK(tour_bracket_type IN('Round
Robin','Single Elimination','Swiss','Double Elimination')),
    CONSTRAINT PK_tour PRIMARY KEY(tour_name,tour_date)
);

CREATE table PARTNERS(
    part_name VARCHAR(35) NOT NULL,
    part_id INT NOT NULL PRIMARY KEY,
    part_industry VARCHAR(50) NOT NULL
```

```
);

CREATE table SECONDARY_PARTNERS_AND_TEAMS(
    pt_part_id INT NOT NULL,
    pt_team_id INT NOT NULL,
    pt_contract_sum INT NOT NULL,
    pt_start_date DATE NOT NULL,
    pt_expiration_date DATE NOT NULL,
    CONSTRAINT PK_pt PRIMARY KEY(pt_part_id,pt_team_id),
    CONSTRAINT FK_pt_part FOREIGN KEY(pt_part_id) REFERENCES PARTNERS(part_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_pt_team FOREIGN KEY(pt_team_id) REFERENCES TEAMS(t_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE table TEAMS_IN_TOURNAMENTS(
    tmt_team_id INT NOT NULL,
    tmt_tour_name VARCHAR(50) NOT NULL,
    tmt_tour_date DATE NOT NULL,
    CONSTRAINT PK_tmt PRIMARY KEY(tmt_team_id,tmt_tour_name,tmt_tour_date),
    CONSTRAINT FK_tmt_team FOREIGN KEY(tmt_team_id) REFERENCES TEAMS(t_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_tmt_tour FOREIGN KEY(tmt_tour_name,tmt_tour_date) REFERENCES
TOURNAMENTS(tour_name,tour_date)
    ON DELETE CASCADE ON UPDATE CASCADE
);

INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('ESL', 2000, 1111, 211, 'Germany' );
INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('ESEA', 2003, 1112, 54, NULL);
INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('FACE_IT', 2012, 1113, 131, 'United
Kingdom');
INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('DreamHack', 1994, 1114, 263,
'Sweden');
INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('StarLadder', 2012, 1115, 86,
'Ukraine');
```

```
INSERT INTO LEAGUES_HOST_COMPANIES VALUES ('RFRSH', 2016, 1116, 59, 'Denmark');

INSERT INTO TEAMS VALUES ('Astralis', 1, 'Danish', 2016, 'Copenhagen', 10000, 100000, 7);
INSERT INTO TEAMS VALUES ('Virtus Pro', 2, 'Polish', 2003, 'Moscow', 10001, 100001, 6);
INSERT INTO TEAMS VALUES ('SK Gaming', 3, 'German', 2010, 'Cologne', 10002, 100002, 7);
INSERT INTO TEAMS VALUES ('fnatic', 4, 'Swedish', 2004, 'London', 10003, 100003, 10);
INSERT INTO TEAMS VALUES ('Team Liquid', 5, 'American' , 2000, 'Santa Monica', 10004, 100004, 5);
INSERT INTO TEAMS VALUES ('Natus Vincere', 6, 'Ukrainian', 2009, 'Kyiv', 10005, 100005, 6);

INSERT INTO PARTNERS VALUES ('JACK & JONES', 10000, 'Clothing');
INSERT INTO PARTNERS VALUES ('Pari Match', 10001, 'Betting');
INSERT INTO PARTNERS VALUES ('Deutsche Telekom', 10002, 'Telecommunications');
INSERT INTO PARTNERS VALUES ('OnePlus', 10003, 'Consumer Electroncis');
INSERT INTO PARTNERS VALUES ('Monster Beverage Corporation', 10004, 'Energy Drinks');
INSERT INTO PARTNERS VALUES ('GG.BET', 10005, 'Betting');
INSERT INTO PARTNERS VALUES ('Intel', 10006, 'Semiconductors');
INSERT INTO PARTNERS VALUES ('MSI', 10007, 'Computer hardware and electronics');
INSERT INTO PARTNERS VALUES ('Alienware', 10008, 'Computer hardware');
INSERT INTO PARTNERS VALUES ('Omen by HP', 10009, 'Computer hardware and software');

INSERT INTO COACHES VALUES ('Danny', 'Sørensen', 100000, 3, 1, 22000);
INSERT INTO COACHES VALUES ('Jakub', 'Gurczyński', 100001, 5, 2, 24000);
INSERT INTO COACHES VALUES ('Wilton', 'Prado', 100002, 3, 3, 7000);
INSERT INTO COACHES VALUES ('Andreas', 'Samuelsson', 100003, 0, 4, 23000);
INSERT INTO COACHES VALUES ('Eric', 'Hoag', 100004, 1, 5, 20000);
INSERT INTO COACHES VALUES ('Andrey', 'Gorodenskiy', 100005, 5, 6, 9000);

INSERT INTO TOURNAMENTS VALUES ('Intel Extreme Masters XIII - Katowice Major 2019', '2019-02-13', 1111, 1000000, 'Major', 'Offline', 'Swiss');
INSERT INTO TOURNAMENTS VALUES ('StarSeries & i-League CS:GO Season 7', '2019-03-30', 1115, 500000, 'Minor', 'Offline', 'Swiss');
INSERT INTO TOURNAMENTS VALUES ('DreamHack Masters Dallas 2019', '2019-05-28', 1114, 250000, 'Minor', 'Offline', 'Double Elimination');
```

```
INSERT INTO TOURNAMENTS VALUES ('Esports Championship Series Season 7 - Finals', '2019-06-06', 1113, 500000, 'Minor', 'Offline', 'Double Elimination');

INSERT INTO TOURNAMENTS VALUES ('ESL One: Cologne 2019', '2019-07-02', 1111, 300000, 'Minor', 'Offline', 'Single Elimination');

INSERT INTO TOURNAMENTS VALUES ('StarLadder Berlin Major 2019', '2019-08-23', 1115, 1000000, 'Major', 'Offline', 'Swiss');

INSERT INTO TOURNAMENTS VALUES ('BLAST Pro Series: Global Final 2019', '2019-12-12', 1116, 500000, 'Minor', 'Offline', 'Double Elimination');

INSERT INTO TOURNAMENTS VALUES ('ESEA Season 32: Premier Division - Europe', '2019-09-09', 1112, 75000, 'Regional', 'Online', 'Round Robin');


INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(1,'Intel Extreme Masters XIII - Katowice Major 2019', '2019-02-13');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(4,'Intel Extreme Masters XIII - Katowice Major 2019', '2019-02-13');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(6,'Intel Extreme Masters XIII - Katowice Major 2019', '2019-02-13');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(4,'StarLadder Berlin Major 2019', '2019-08-23');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(6,'StarLadder Berlin Major 2019', '2019-08-23');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(2,'StarLadder Berlin Major 2019', '2019-08-23');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(1,'Esports Championship Series Season 7 - Finals', '2019-06-06');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(3,'Esports Championship Series Season 7 - Finals', '2019-06-06');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(5,'DreamHack Masters Dallas 2019', '2019-05-28');

INSERT INTO TEAMS_IN_TOURNAMENTS VALUES(4,'DreamHack Masters Dallas 2019', '2019-05-28');


INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10006,1, 750000, '2019-01-21', '2020-01-21');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10006,4, 800000, '2019-05-11', '2020-05-11');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10006,6, 650000, '2019-03-05', '2020-03-05');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10007,1, 600000, '2019-07-12', '2020-07-12');
```

```
INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10007,4, 250000, '2019-02-02', '2020-02-02');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10008,5, 150000, '2019-09-23', '2020-09-23');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10008,3, 750000, '2019-01-29', '2020-01-29');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10009,2, 1000000, '2019-10-09', '2020-10-09');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10009,5, 450000, '2019-05-17', '2020-05-17');

INSERT INTO SECONDARY_PARTNERS_AND_TEAMS VALUES(10005,3, 770000, '2019-03-15', '2020-03-15');


INSERT INTO PLAYERS VALUES('Nicolai', 'dev1ce', 'Reedtz', 10, 'Danish', 24, 1, 1590501);

INSERT INTO PLAYERS VALUES('Peter', 'dupreeh', 'Rasmussen', 11, 'Danish', 26, 1, 1621123);

INSERT INTO PLAYERS VALUES('Andreas', 'Xyp9x', 'Højsleth', 12, 'Danish', 24, 1, 1624934);

INSERT INTO PLAYERS VALUES('Lukas', 'gla1ve', 'Rossander', 13, 'Danish', 24, 1, 1323552);

INSERT INTO PLAYERS VALUES('Emil', 'Magisk', 'Reif', 14, 'Danish', 21, 1, 1203818);

INSERT INTO PLAYERS VALUES('Janusz', 'Snax', 'Pogorzelski', 15, 'Polish', 26, 2, 637630);

INSERT INTO PLAYERS VALUES('Paweł', 'byali', 'Bieliński', 16, 'Polish', 25, 2, 570881);

INSERT INTO PLAYERS VALUES('Jarosław', 'pashaBiceps', 'Jarząbkowski', 17, 'Polish', 31, 2, 620576);

INSERT INTO PLAYERS VALUES('Filip', 'NEO', 'Kubski', 18, 'Polish', 32, 2, 743495);

INSERT INTO PLAYERS VALUES('Wiktor', 'TaZ', 'Wojtas', 19, 'Polish', 33, 2, 725958);

INSERT INTO PLAYERS VALUES('Gabriel', 'FalleN', 'Toledo', 20, 'Brazilian', 28, 3, 948740);

INSERT INTO PLAYERS VALUES('Fernando', 'fer', 'Alvarenga', 21, 'Brazilian', 28, 3, 935596);

INSERT INTO PLAYERS VALUES('Marcelo', 'coldzera', 'David', 22, 'Brazilian', 25, 3, 942699);

INSERT INTO PLAYERS VALUES('Ricardo', 'boltz', 'Prass', 23, 'Brazilian', 22, 3, 344150);

INSERT INTO PLAYERS VALUES('Freddy', 'KRIMZ', 'Johansson', 24, 'Swedish', 25, 4, 782580);

INSERT INTO PLAYERS VALUES('Jesper', 'JW', 'Wecksell', 25, 'Swedish', 24, 4, 812521);

INSERT INTO PLAYERS VALUES('Ludvig', 'Brollan', 'Brolin', 26, 'Swedish', 17, 4, 140437);

INSERT INTO PLAYERS VALUES('Robin', 'flusha', 'Rönnquist', 27, 'Swedish', 26, 4, 786441);

INSERT INTO PLAYERS VALUES('Richard', 'Xizt', 'Landström', 28, 'Swedish', 28, 4, 541198);

INSERT INTO PLAYERS VALUES('Nicholas', 'nitr0', 'Cannella', 29, 'United States', 24, 5, 838332);
```

```
INSERT INTO PLAYERS VALUES('Jonathan', 'EliGE', 'Jablonowski', 30, 'United States', 22, 5, 829482);

INSERT INTO PLAYERS VALUES('Russel', 'Twistzz', 'Van Dulken', 31, 'Canadian', 19, 5, 404272);

INSERT INTO PLAYERS VALUES('Keith', 'NAF', 'Markovic', 32, 'Canadian', 21, 5, 882649);

INSERT INTO PLAYERS VALUES('Jacky', '"Stewie2K', 'Yip', 33, 'United States', 21, 5, 979640);

INSERT INTO PLAYERS VALUES('Epitácio', 'TACO', 'de Melo', 34, 'Brazilian', 24, 3, 944996);

INSERT INTO PLAYERS VALUES('Egor', 'flamie', 'Vasilev', 35, 'Russian', 22, 6, 610060);

INSERT INTO PLAYERS VALUES('Oleksandr', 's1mple', 'Kostyliev', 36, 'Ukrainian', 22, 6, 486886);

INSERT INTO PLAYERS VALUES('Ladislav', 'GuardiaN', 'Kovács', 37, 'Slovakian', 28, 6, 788607);

INSERT INTO PLAYERS VALUES('Danylo', 'Zeus', 'Teslenko', 38, 'Ukrainian', 32, 6, 785002);

INSERT INTO PLAYERS VALUES('Ioann', 'Edward', 'Sukhariev', 39, 'Ukrainian', 31, 6, 710977);


ALTER TABLE PLAYERS ADD CONSTRAINT FK_p_team FOREIGN KEY(p_team_id)
REFERENCES TEAMS(t_id)

   ON DELETE SET NULL

   ON UPDATE CASCADE;

ALTER TABLE TEAMS ADD CONSTRAINT FK_t_partner FOREIGN KEY(t_main_partner_id)
REFERENCES PARTNERS(part_id)

   ON DELETE SET NULL

   ON UPDATE CASCADE;

ALTER TABLE TEAMS ADD CONSTRAINT FK_t_coach FOREIGN KEY(t_coach_id)
REFERENCES COACHES(c_id)

   ON DELETE SET NULL

   ON UPDATE CASCADE;

ALTER TABLE COACHES ADD CONSTRAINT FK_c_team FOREIGN KEY(c_team_id)
REFERENCES TEAMS(t_id)

   ON DELETE SET NULL

   ON UPDATE CASCADE;

ALTER TABLE TOURNAMENTS ADD CONSTRAINT FK_host_id FOREIGN
KEY(tour_host_company_id) REFERENCES LEAGUES_HOST_COMPANIES(l_id)

   ON DELETE CASCADE

   ON UPDATE CASCADE;


ALTER TABLE PLAYERS ADD CHECK(p_id>0);
```

```
ALTER TABLE TEAMS ADD CHECK(t_id>0);

ALTER TABLE LEAGUES_HOST_COMPANIES ADD CHECK(l_id>0);

ALTER TABLE COACHES ADD CHECK(c_id>0);

ALTER TABLE PARTNERS ADD CHECK(part_id>0);

CREATE VIEW Coach_View(f_name, nickname, s_name, id, nationality, age, team_id) AS
SELECT p_first_name, p_ingame_name, p_second_name, p_id, p_nationality, p_age,
p_team_id

FROM PLAYERS WHERE 1 = p_team_id;


CREATE VIEW Host_view(team_name, team_id, team_nationality, team_year_established,
t_main_partner_id,  team_coach_id, team_n_of_tournaments_won) AS

SELECT DISTINCT t_name, t_id, t_nationality, t_year_established, t_main_partner_id,
t_coach_id, t_n_of_tournaments_won

FROM TEAMS, TEAMS_IN_TOURNAMENTS, LEAGUES_HOST_COMPANIES,
TOURNAMENTS

WHERE t_id = tmt_team_id AND tmt_tour_name IN (SELECT tour_name FROM
TOURNAMENTS, LEAGUES_HOST_COMPANIES WHERE tour_host_company_id = 1115)

AND tmt_tour_date IN (SELECT tour_date FROM TOURNAMENTS,
LEAGUES_HOST_COMPANIES WHERE tour_host_company_id = 1115);


CREATE VIEW Analytics_view(total_winnings) AS

SELECT p_total_winnings FROM PLAYERS;


CREATE ROLE Coach;

CREATE ROLE Player;

CREATE ROLE Event_Organiser;

CREATE ROLE Analytics;


GRANT SELECT ON Coach_view TO Coach;

GRANT ALL ON Host_view TO Event_Organiser WITH GRANT OPTION;

GRANT SELECT ON PLAYERS TO Player;

GRANT SELECT ON TEAMS TO Player;

GRANT SELECT ON Analytics_view TO Analytics;
```

```
DELIMITER $$
CREATE TRIGGER Team_deletion_update_tournaments
AFTER DELETE ON TEAMS
FOR EACH ROW
BEGIN
DECLARE
   deleted_t_id int;
IF (OLD.t_id IS NOT NULL) THEN
   SET @deleted_t_id := OLD.t_id;
   DELETE FROM TEAMS_IN_TOURNAMENTS
   WHERE tmt_team_id=deleted_t_id;
END IF;
END$$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER Team_deletion_update_sponsors
AFTER DELETE ON TEAMS
FOR EACH ROW
BEGIN
DECLARE
   deleted_t_id int;
IF (OLD.t_id IS NOT NULL) THEN
   SET @deleted_t_id := OLD.t_id;
   DELETE FROM SECONDARY_PARTNERS_AND_TEAMS
   WHERE  pt_team_id=deleted_t_id;
END IF;
END$$
DELIMITER ;
```