Databases

Project Work

<National Basketball Association(NBA) Database>

Project elaborator: **Goncalves Charles Nicolas Kormyshev Mendes**, DWFD1I,
Charlesgoncalve07@gmail.com

University: Óbudai Egyetem – Neumann János Informatikai Kar

Major: Computer Science BSc

Semester: 2020/2021-1

Table of Contents

1 Project work's topics description

I chose the topic **National Basketball Association(NBA) Database** because I truly enjoy watching professional basketball leagues, and NBA being one of the most famous and watched in the world happened to be chosen as the topic for myproject's Work.

When you think about a basketball in general, the first thing that comes to your mind are the names of the famous players in the world(usually from the NBA),well known due to their outstanding performances throughout their career and the yearly individual awards they've got, because of their individual skills of course, but we can't forget that factors like age, hight, the way they play on their positions and deliver themselves to the game determines the salary/contract they'll be signing with a team on the following seasons.

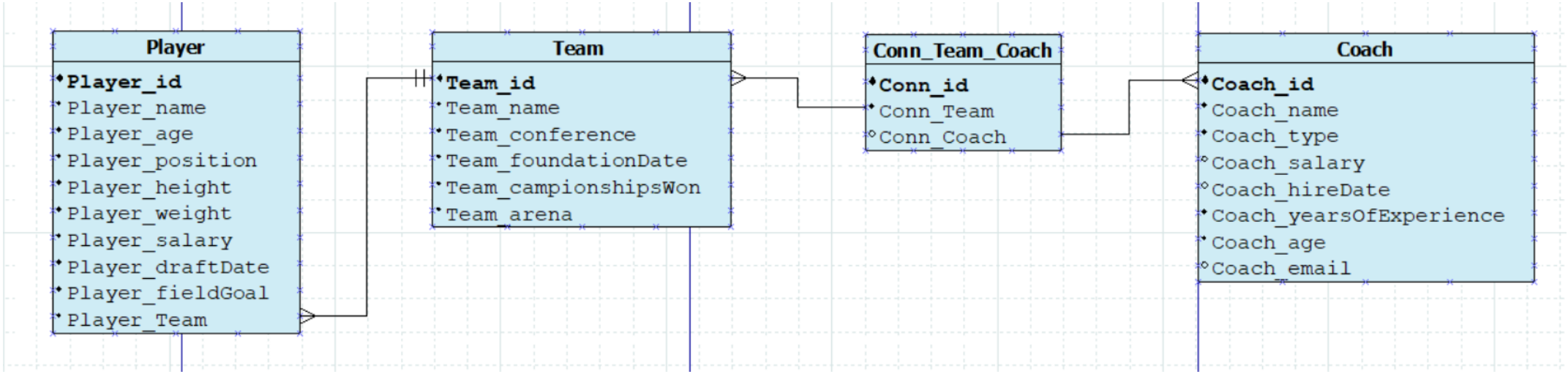
Now its time to give place for the amazing combination of players that each team has that makes the league more competitive and exiting. Its basically a race that every team form the different conferences (Western and Eastern) has at the end of the season to get the players that best suit their team, cause at the end of the day, they all want the same thing, winning the championship!

Thinking more deeply about what is behind the scenes, we realize that the coaches, inclouding shooting coach, handles coach and defense coach, are the ones that make all the show that we watch come true.

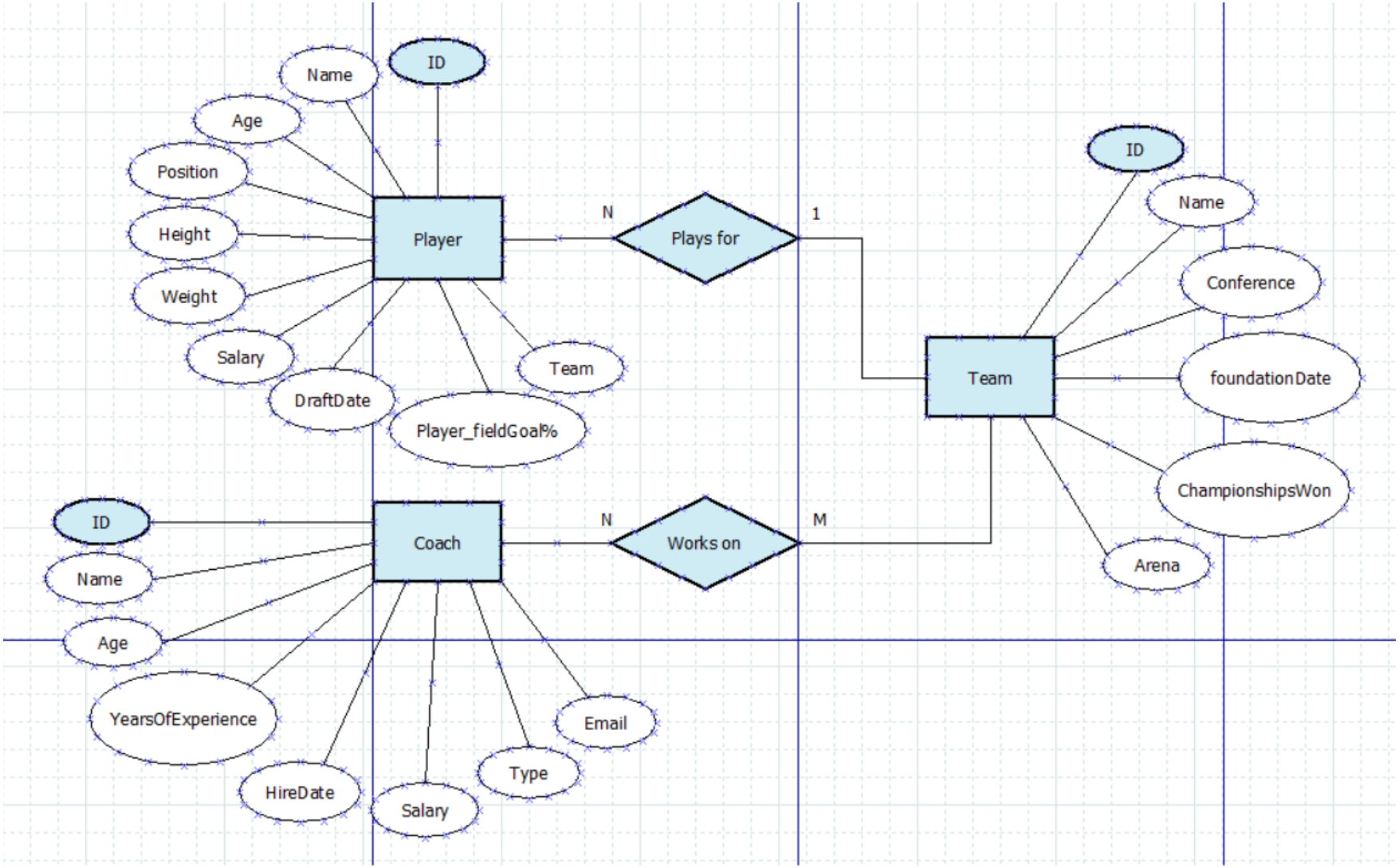
To sumarize the description of my project work's topic, I've chosen the 3 entities mentioned above for the tables that I've created, namely:

- Player: containing the ID as a primary key, name, age, position, height, weight, salary, draftDate as attributes and Team as foreign key that stablishes the one to many relation with the Team, where one player can play for one and only team at a time, and one team can have many players so it's a 1:N relation;
- Team: containing the ID as primary key and name, conference, foundation date, number of championships won, arena as attributes;
- Connector table Team-Coach: containing the ID as primary key and stablishes the many to many relation between the Team and the coach, because the team can have many coaches(considering the many types of coache) and a coach can work for many teams, depending on the need of the team, so it's a N:M relation;
- Coach: containing the ID as primary key and name, type, salary, years of experience, hire date, age, email as attributes.

2 Project work's table structure diagram



3 Project work's ER diagram



4 Proof that my table structure is BCNF

My table table structures are BCNF because they fulfil all the requirements for 0NF, 1NF, 2NF, 3NF and BCNF.

They fulfil the 0NF for not having any unnecessary data, having only the ones that I really intend store and add to the table;

They fulfil the 1NF for having only atomic cells in every row of the relation, so each rows has their own data field cells, which implies not having any merged rows/columns, and also it doesn't contain any complex data attributes, in other words, It doesn't have any cell that contains a list inside it, otherwise it would be unusable by the SQL.

They fulfil the 2NF, because on the tables that I've created the fields are either a key field or a field that functionally depends on the key fields, which basically means that I have an attribute that clrearly defines the value of the other attributes on the table.
Satisfying this way the definition of functional dependency(FD), I can also state that every table has one and only functional dependency(FD), where every secondary attribute functionally depends on the key of the relation.

They fulfil the 3NF, because every non key attribute or secondary attributes depend only on the key field, not functionally depending on any other attributes in the relation, this way there is no field that is transitively dependent on the key(s);

They end up also fulfilling the BCNF because it fulfils all the other normal forms.

5 Proof that my table structure is lossless decomposition (Using the Chase algorithm)

On my table structures I have 24 data fields and they are the following:

R={ p_id, p_name, p_age, p_position, p_height, p_weight, p_salary, p_draftDate, p_fieldGoal, t_id, t_name, t_conference, t_foundationDate, t_champWon, t_anena, c_id, c_name, c_type, c_salary, c_hireDate, c_YearsXP, c_age, c_email
}

And the Functional dependencies that could be made out of this data fields can be seen below:

F={

F_{player} : { p_id } => { p_name, p_age, p_position, p_height, p_weight, p_salary, p_draftDate, p_fieldGoal, }

F_{team} : { t_id } => { t_name, t_conference, t_foundationDate, t_champWon, t_anena }

F_{coach} : { c_id } => { c_name, c_type, c_salary, c_hireDate, c_YearsXP, c_age, c_email }

}

On the first row of the chase algorithm table that created to proof the lossless decomposition I will have every single data field from every single table that I've created and since I followed the rule that every single table must have at most 1 FD (2NF), I have 3 functional dependencies on 3 different tables, so they will be represented by more 3 rows on the chase algorithm table, plus another row representing the connector table for the many to many relation existing between the team and the coach tables, while the data fields will be represeted with by columns, which means that I will have a 5x24 table.
At the starting point we have only unknown values which are represented by B(i:j) values which will be replaced by A values if the current table/functional dependency contain the field.

	p_id	p_name	p_age	p_pos	p_height	p_weight	p_sal	p_dd	p_FG	t_id	t_name	t_conf	t_FD	t_ChW	t_arena	Conn_id	c_id	c_name	c_type	c_sal	c_HD	c_yearXP	c_age	c_email
Player	B(1,1)	B(1,2)	B(1,3)	B(1,4)	B(1,5)	B(1,6)	B(1,7)	B(1,8)	B(1,9)	B(1,10)	B(1,11)	B(1,12)	B(1,13)	B(1,14)	B(1,15)	B(1,16)	B(1,17)	B(1,18)	B(1,19)	B(1,20)	B(1,21)	B(1,22)	B(1,23)	B(1,24)
Team	B(2,1)	B(2,2)	B(2,3)	B(2,4)	B(2,5)	B(2,6)	B(2,7)	B(2,8)	B(2,9)	B(2,10)	B(2,11)	B(2,12)	B(2,13)	B(2,14)	B(2,15)	B(2,16)	B(2,17)	B(2,18)	B(2,19)	B(2,20)	B(2,21)	B(2,22)	B(2,21)	B(2,24)
Conn_team_coach	B(3,1)	B(3,2)	B(3,3)	B(3,4)	B(3,5)	B(3,6)	B(3,7)	B(3,8)	B(3,9)	B(3,10)	B(3,11)	B(3,12)	B(3,13)	B(3,14)	B(3,15)	B(3,16)	B(3,17)	B(3,18)	B(3,19)	B(3,20)	B(3,21)	B(3,22)	B(3,23)	B(3,24)
Coach	B(4,1)	B(4,2)	B(4,3)	B(4,4)	B(4,5)	B(4,6)	B(4,7)	B(4,8)	B(4,9)	B(4,10)	B(4,11)	B(4,12)	B(4,13)	B(4,14)	B(4,15)	B(4,16)	B(4,17)	B(4,18)	B(4,19)	B(4,20)	B(4,21)	B(4,22)	B(4,23)	B(4,24)

After replacing those B(i:j) values by A values we get:

	p_id	p_name	p_age	p_pos	p_height	p_weight	p_sal	p_dd	p_FG	t_id	t_name	t_conf	t_FD	t_ChW	t_arena	Conn_id	c_id	c_name	c_type	c_sal	c_HD	c_yearXP	c_age	c_email
Player	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	B(1,11)	B(1,12)	B(1,13)	B(1,14)	B(1,15)	B(1,16)	B(1,17)	B(1,18)	B(1,19)	B(1,20)	B(1,21)	B(1,22)	B(1,23)	B(1,24)
Team	B(2,1)	B(2,2)	B(2,3)	B(2,4)	B(2,5)	B(2,6)	B(2,7)	B(2,8)	B(2,9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	B(2,16)	B(2,17)	B(2,18)	B(2,19)	B(2,20)	B(2,21)	B(2,22)	B(2,21)	B(2,24)
Conn_team_coach	B(3,1)	B(3,2)	B(3,3)	B(3,4)	B(3,5)	B(3,6)	B(3,7)	B(3,8)	B(3,9)	A(10)	B(3,11)	B(3,12)	B(3,13)	B(3,14)	B(3,15)	A(16)	A(17)	B(3,18)	B(3,19)	B(3,20)	B(3,21)	B(3,22)	B(3,23)	B(3,24)
Coach	B(4,1)	B(4,2)	B(4,3)	B(4,4)	B(4,5)	B(4,6)	B(4,7)	B(4,8)	B(4,9)	B(4,10)	B(4,11)	B(4,12)	B(4,13)	B(4,14)	B(4,15)	B(4,16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)

Now that we already have all the fields present in the tables replaced by A values, we can start looping through the functional dependencies to see if we can replace some more values accordingly to the relation stablished between the tables, in other words replacing all the values that depend on the the a known value.
On this case all the fields in the FD relation that has the team_id and coahc_id as known values will be replaced by known values:

	p_id	p_name	p_age	p_pos	p_height	p_weight	p_sal	p_dd	p_FG	t_id	t_name	t_conf	t_FD	t_ChW	t_arena	Conn_id	c_id	c_name	c_type	c_sal	c_HD	c_yearXP	c_age	c_email
Player	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	B(1,16)	B(1,17)	B(1,18)	B(1,19)	B(1,20)	B(1,21)	B(1,22)	B(1,23)	B(1,24)
Team	B(2,1)	B(2,2)	B(2,3)	B(2,4)	B(2,5)	B(2,6)	B(2,7)	B(2,8)	B(2,9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	B(2,16)	B(2,17)	B(2,18)	B(2,19)	B(2,20)	B(2,21)	B(2,22)	B(2,21)	B(2,24)
Conn_team_coach	B(3,1)	B(3,2)	B(3,3)	B(3,4)	B(3,5)	B(3,6)	B(3,7)	B(3,8)	B(3,9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	A(16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)
Coach	B(4,1)	B(4,2)	B(4,3)	B(4,4)	B(4,5)	B(4,6)	B(4,7)	B(4,8)	B(4,9)	B(4,10)	B(4,11)	B(4,12)	B(4,13)	B(4,14)	B(4,15)	B(4,16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)

After executing the chase algorithm and replacing all the possible values we get didn't get any row that contains only know data due to the many to many relation, because there is a hidden dependency, from the chase algorithm point of view we have a composite dependency. On the connector table we have that the connector table id determines the coach id and the team id, there is a dependency between them that the FD hides which is the following:

{ t_id } => { c_name, c_type, c_salary, c_hireDate, c_YearsXP, c_age, c_email }^N

executing this we get the corret result with the full row of know values which means that our **table structure is lossless.**

	p_id	p_name	p_age	p_pos	p_height	p_weight	p_sal	p_dd	p_FG	t_id	t_name	t_conf	t_FD	t_ChW	t_arena	Conn_id	c_id	c_name	c_type	c_sal	c_HD	c_yearXP	c_age	c_email
Player	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	A(16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)
Team	B(2,1)	B(2,2)	B(2,3)	B(2,4)	B(2,5)	B(2,6)	B(2,7)	B(2,8)	B(2,9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	A(16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)
Conn_team_coach	B(3,1)	B(3,2)	B(3,3)	B(3,4)	B(3,5)	B(3,6)	B(3,7)	B(3,8)	B(3,9)	A(10)	A(11)	A(12)	A(13)	A(14)	A(15)	A(16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(24)
Coach	B(4,1)	B(4,2)	B(4,3)	B(4,4)	B(4,5)	B(4,6)	B(4,7)	B(4,8)	B(4,9)	B(4,10)	B(4,11)	B(4,12)	B(4,13)	B(4,14)	B(4,15)	B(4,16)	A(17)	A(18)	A(19)	A(20)	A(21)	A(22)	A(23)	A(

6 All CREATE TABLE + INSERT INTO statements

```
IF object_id('conn_coach_team', 'U') is not null DROP TABLE conn_coach_team;
IF object_id('coach', 'U') is not null DROP TABLE coach;
IF object_id('player', 'U') is not null DROP TABLE player;
IF object_id('team', 'U') is not null DROP TABLE team;
GO

CREATE TABLE team(
      team_id int primary key,
      team_name nvarchar(100),
      team_conference nvarchar(100),
      team_foundationDate date,
      team_championshipsWon int,
      team_arena nvarchar(100)
)

CREATE TABLE coach(
      coach_id int primary key,
      coach_name nvarchar(100),
      coach_type nvarchar(100),
      coach_yearsOfExperience int,
      coach_hireDate date,
      coach_salary int,
      coach_age int,
      coach_email nvarchar(100)
)

CREATE TABLE player(
      player_id int primary key,
      player_name nvarchar(100),
      player_age int,
      player_position nvarchar(100),
      player_height float,
      player_weight float,
      player_salary int,
      player_fieldGoal float,
      player_draftDate date,
      player_team int NOT NULL references team(team_id)
)


CREATE TABLE conn_coach_team(
      conn_id int identity primary key,
      conn_coach int references coach(coach_id),
      conn_team int references team(team_id)
)
GO
```

```
INSERT INTO team VALUES(1, 'Los Angeles Lakers', 'Western', '1947-10-06', 17, 'Staples Center')
INSERT INTO team VALUES(2, 'Los Angeles Clippers', 'Western', '1970-09-30', 0, 'Staples Center')
INSERT INTO team VALUES(3, 'Houston Rockets', 'Western', '1967-11-26', 2, 'Toyota Center')
INSERT INTO team VALUES(4, 'Milwaukee Bucks', 'Eastern', '1968-05-07', 1,'Fiserv Forum')
INSERT INTO team VALUES(5, 'Boston Celtics', 'Eastern', '1946-06-09', 17,'TD Garden');

INSERT INTO coach VALUES(1, 'Joseph Clanaghan', 'Handles coach', 10, '2019-06-05', 900000, 38, 'Josephclanaghan07@gmail.com')
INSERT INTO coach VALUES(2, 'James Cooper', 'Shooting coach', 15, '2018-03-16', 1500000, 40, 'Jamescooper3445@gmail.com')
INSERT INTO coach VALUES(3, 'Jacob Smith', 'Defense coach', 6, '2020-07-20', 5000000, 32, 'Jacobsmith867@gmail.com')
INSERT INTO coach VALUES(4, 'Theo Williams', 'Defense coach',8, '2019-11-23', 7000000, 37, 'Theowilliams34@gmail.com')
INSERT INTO coach VALUES(5, 'Marc Thompson', 'Handles coach',9, '2017-01-31', 8000000, 35, 'Marcthompson6541@gmail.com')
INSERT INTO coach VALUES(6, 'Harry Addams', 'Shooting coach',12, '2020-06-11', 1300000, 42, 'Marcthompson6541@gmail.com');


INSERT INTO player VALUES(1, 'Lebron James', 35, 'PF', 2.06, 113.0, 37000000, 49.3,'2003-06-09', 1)
INSERT INTO player VALUES(2, 'Anthony Davis', 27, 'C', 2.08, 114.0, 27000000, 50.3,'2012-10-15', 1)

INSERT INTO player VALUES(3, 'Kawhi Leonard', 29, 'SF', 2.01, 102.0, 32000000, 49.1,'2011-03-14', 2)
INSERT INTO player VALUES(4, 'Paul George', 30, 'SG', 2.03, 99.0, 33000000, 43.3,'2010-04-21', 2)

INSERT INTO player VALUES(5, 'James Harden', 31, 'SG', 1.96, 99.0, 38000000, 44.4,'2009-01-22', 3)
INSERT INTO player VALUES(6, 'Russell Westbrook', 27, 'PG', 1.90, 90.0, 38000000, 43.7,'2008-04-10', 3)

INSERT INTO player VALUES(7, 'Giannis Antetokounmpo', 25, 'PF', 2.11, 109.0, 24000000, 55.3,'2013-02-03', 4)
INSERT INTO player VALUES(8, 'George Hill', 34, 'PG', 1.90, 85.0, 8000000, 51.6,'2008-07-17', 4)

INSERT INTO player VALUES(9, 'Jayson Tatum', 22, 'SF', 2.03, 95.0, 7000000, 45.0,'2017-12-03', 5)
INSERT INTO player VALUES(10, 'Tacko Fall', 24, 'C', 2.26, 141.0, 1000000, 78.6,'2019-01-30', 5);

INSERT INTO conn_coach_team (conn_coach, conn_team) VALUES (1,1), (2,1), (2,2), (3,2), (3,3), (4,3), (4,4), (5,4), (5,5), (6,5), (1,6);
GO
```

7 Simple single-table select statements(Q1-Q4)

```
Q1 => SELECT * FROM team;
```

Results Messages

	team_id	team_name	team_conference	team_foundationDate	team_championshipsWon	team_arena
1	1	Los Angeles Lakers	Western	1947-10-06	17	Staples Center
2	2	Los Angeles Clippers	Western	1970-09-30	0	Staples Center
3	3	Houston Rockets	Western	1967-11-26	2	Toyota Center
4	4	Milwaukee Bucks	Eastern	1968-05-07	1	Fiserv Forum
5	5	Boston Celtics	Eastern	1946-06-09	17	TD Garden

```
Q2 => SELECT * FROM player;
```

Results Messages

	player_id	player_name	player_age	player_position	player_height	player_weight	player_salary	player_fieldGoal	player_draftDate	player_team
1	1	Lebron James	35	PF	2.06	113	37000000	49.3	2003-06-09	1
2	2	Russell Westbrook	27	PG	1.9	90	38000000	43.7	2008-04-10	3
3	3	George Hill	34	PG	1.9	85	8000000	51.6	2008-07-17	4
4	4	James Harden	31	SG	1.96	99	38000000	44.4	2009-01-22	3
5	5	Paul George	30	SG	2.03	99	33000000	43.3	2010-04-21	2
6	6	Kawhi Leonard	29	SF	2.01	102	32000000	49.1	2011-03-14	2
7	7	Anthony Davis	27	C	2.08	114	27000000	50.3	2012-10-15	1
8	8	Giannis Antetokounmpo	25	PF	2.11	109	24000000	55.3	2013-02-03	4
9	9	Jayson Tatum	22	SF	2.03	95	7000000	45	2017-12-03	5
10	10	Tacko Fall	24	C	2.26	141	1000000	78.6	2019-01-30	5

```
Q3 => SELECT * FROM coach;
```

Results | Messages

	coach_id	coach_name	coach_type	coach_yearsOfExperience	coach_hireDate	coach_salary	coach_age	coach_email
1	1	Joseph Clanaghan	Handles coach	10	2019-06-05	900000	38	Josephclanaghan07@gmail.com
2	2	James Cooper	Shooting coach	15	2018-03-16	1500000	40	Jamescooper3445@gmail.com
3	3	Jacob Smith	Defense coach	6	2020-07-20	5000000	32	Jacobsmith867@gmail.com
4	4	Theo Williams	Defense coach	8	2019-11-23	7000000	37	Theowilliams34@gmail.com
5	5	Marc Thompson	Handles coach	9	2017-01-31	8000000	35	Marcthompson6541@gmail.com
6	6	Harry Addams	Shooting coach	12	2020-06-11	1300000	42	Marcthompson6541@gmail.com

```
Q4 => SELECT * FROM conn_coach_team;
```

Results | Messages

	conn_id	conn_coach	conn_team
1	1	1	1
2	2	2	1
3	3	2	2
4	4	3	2
5	5	3	3
6	6	4	3
7	7	4	4
8	8	5	4
9	9	5	5
10	10	6	5

8 Simple single-table group by(Q5-Q7)

Q5 => List the number of teams belonging to each conference and the total number of championships won in each conference:

```
SELECT team_conference , count(team_id) as numberOfTeams, sum(team_championshipsWon)
as championshipsWon
FROM team
GROUP BY team_conference;
```

Results Messages

	team_conference	numberOfTeams	championshipsWon
1	Eastern	2	18
2	Western	3	19

Q6 => List the number of players for each position and the best and the worse field goal on the current position:

```
SELECT player_position, count(player_id) as numberPerPosition, max(player_fieldGoal)
bestFGperPosition, min(player_fieldGoal) worseFGperPosition
from player
group by player_position;
```

Results Messages

	player_position	numberPerPosition	bestFGperPosition	worseFGperPosition
1	C	2	78.6	50.3
2	PF	2	55.3	49.3
3	PG	2	51.6	43.7
4	SF	2	49.1	45
5	SG	2	44.4	43.3

Q7 => List the number of coaches accordingly to the type of coach and the average salary for each type:

Results Messages

	coach_type	CoachNum	AVGsal
1	Defense coach	2	6000000
2	Handles coach	2	4450000
3	Shooting coach	2	1400000

9 Complex multi-table select (Q8-Q12)

```
Q8 => List the  number of coaches for each team:

SELECT team_name, count(coach_id) as numCoach
from team inner join conn_coach_team on team_id = conn_team
          inner join coach on conn_coach = coach_id
group by team_name;
```

Results | Messages

	team_name	numCoach
1	Boston Celtics	2
2	Houston Rockets	2
3	Los Angeles Clippers	2
4	Los Angeles Lakers	2
5	Milwaukee Bucks	2

```
Q9 => List the  number of players in each team:

SELECT team_name, count(player_id) as numbeOfPlayers, avg(player_fieldGoal) as
AvgFGTeam
from player inner join team on player_team = team_id
group by team_name;
```

Results | Messages

	team_name	numbeOfPlayers	AvgFGTeam
1	Boston Celtics	2	61.8
2	Houston Rockets	2	44.05
3	Los Angeles Clippers	2	46.2
4	Los Angeles Lakers	2	49.8
5	Milwaukee Bucks	2	53.45

```
Q10 => List the team name and the coach name working for them:

select team_name, coach_name
from team
          inner join conn_coach_team on (team_id = conn_team)
          inner join coach on (conn_coach = coach_id)
order by team_name;
```

Results | Messages

	team_name	coach_name
1	Boston Celtics	Marc Thompson
2	Boston Celtics	Harry Addams
3	Houston Rockets	Jacob Smith
4	Houston Rockets	Theo Williams
5	Los Angeles Clippers	James Cooper
6	Los Angeles Clippers	Jacob Smith
7	Los Angeles Lakers	Joseph Clanaghan
8	Los Angeles Lakers	James Cooper
9	Milwaukee Bucks	Theo Williams
10	Milwaukee Bucks	Marc Thompson

Q11 => List the name of the players and the name of the team where they are playing for:

```
SELECT team_name, player_name
FROM player inner join team on player_team = team_id
order by team_name;
```

Results | Messages

	team_name	player_name
1	Boston Celtics	Jayson Tatum
2	Boston Celtics	Tacko Fall
3	Houston Rockets	Russell Westbrook
4	Houston Rockets	James Harden
5	Los Angeles Clippers	Paul George
6	Los Angeles Clippers	Kawhi Leonard
7	Los Angeles Lakers	Lebron James
8	Los Angeles Lakers	Anthony Davis
9	Milwaukee Bucks	George Hill
10	Milwaukee Bucks	Giannis Antetokounmpo

Q12 => List the coach type that each teams are currently working with:

```
SELECT team_name, coach_type
FROM team inner join conn_coach_team on team_id = conn_team
          inner join coach on conn_coach = coach_id
ORDER BY team_name;
```

Results | Messages

	team_name	coach_type
1	Boston Celtics	Handles coach
2	Boston Celtics	Shooting coach
3	Houston Rockets	Defense coach
4	Houston Rockets	Defense coach
5	Los Angeles Clippers	Shooting coach
6	Los Angeles Clippers	Defense coach
7	Los Angeles Lakers	Handles coach
8	Los Angeles Lakers	Shooting coach
9	Milwaukee Bucks	Defense coach
10	Milwaukee Bucks	Handles coach

10 Complex subquery select (Q13-Q17)

```
Q13/14 => Who are the oldest(Veteran) and the most recent(Rookie) players in the
league?

if object_id('v1', 'v') is not null drop view v1;
go

create view v1 as
      select datediff(year, player_draftDate, GETDATE()) yearsInTheLeague
        from player
go

select player_name as Veteran, player_draftDate, (select max(yearsInTheLeague) from
v1) as yearsInTheLeague
from (select player_name, player_draftDate from player) as subquery
order by player_draftDate
offset 0 rows
fetch first 1 row only;

select player_name as Rookie, player_draftDate, (select min(yearsInTheLeague) from v1)
as yearsInTheLeague
from (select player_name, player_draftDate from player) as subquery
order by player_draftDate desc
offset 0 rows
fetch first 1 row only;
```

Results Messages

	Veteran	player_draftDate	yearsInTheLeague
1	Lebron James	2003-06-09	17

	Rookie	player_draftDate	yearsInTheLeague
1	Tacko Fall	2019-01-30	1

```
Q15 => What was the longest interval of time(in years) that the NBA stayed without
getting a new player in the league?

select max(subquery.intervalInYears) as intervalInYears
from (select datediff(year, p1.player_draftDate, p2.player_draftDate) as
intervalInYears
      from player p1 inner join player p2 on p1.player_id = p2.player_id - 1) as
subquery;
```

Results Messages

	intervalInYears
1	5

```
Q16 => How many years after the teams foundation did each of them hire one of their
actual coach? how many coaches does each team  have?

select team_name, min(interval) as yearsAfterFoundation, count(coach_id) as
CoachesHiredSinceThen
from (
      select coach_id, coach_hireDate, team_name, team_foundationDate, datediff(year,
team_foundationDate, coach_hireDate) as interval
        from team inner join conn_coach_team on team_id = conn_team
                  inner join coach on conn_coach = coach_id
        order by coach_name offset 0 rows
) as subquery
group by team_name;
```

Results | Messages

	team_name	yearsAfterFoundation	CoachesHiredSinceThen
1	Boston Celtics	71	2
2	Houston Rockets	52	2
3	Los Angeles Clippers	48	2
4	Los Angeles Lakers	71	2
5	Milwaukee Bucks	49	2

```
Q17 => Considering this coaches, what age was the earlient that a coach got into the
league?

select min(carrierStart) as carrierStart
from (
      select coach_yearsOfExperience, coach_age, (coach_age-coach_yearsOfExperience)
as carrierStart
         from coach
) as subquery;
```

Results | Messages

	carrierStart
1	25

11 Analytics/advanced grouping query (Q18-Q22)

```
Q18 => List the rank of the teams in the league accordingly to the championships won

SELECT
DENSE_RANK() OVER(ORDER BY team_championshipsWon desc) AS RankInTheLeague, team_name
as Name,team_championshipsWon as ChampionshipsWon, (select
sum(team_championshipsWon)from team) as OutOf
FROM team
ORDER BY RankInTheLeague ASC;
select * from player
```

Results | Messages

	RankInTheLeague	Name	ChampionshipsWon	OutOf
1	1	Los Angeles Lakers	17	37
2	1	Boston Celtics	17	37
3	2	Houston Rockets	2	37
4	3	Milwaukee Bucks	1	37
5	4	Los Angeles Clippers	0	37

```
Q19/20 => List the rank of the better paid coaches and players

SELECT
DENSE_RANK() OVER(ORDER BY coach_salary desc) AS RankInTheLeague, coach_name as
Name,coach_salary as Salary
FROM coach
ORDER BY RankInTheLeague ASC;

SELECT
DENSE_RANK() OVER(ORDER BY player_salary desc) AS RankInTheLeague, player_name as
Name,player_salary as Salary
FROM player
ORDER BY RankInTheLeague ASC;
```

Results | Messages

	RankInTheLeague	Name	Salary
1	1	Marc Thompson	8000000
2	2	Theo Williams	7000000
3	3	Jacob Smith	5000000
4	4	James Cooper	1500000
5	5	Harry Addams	1300000
6	6	Joseph Clanaghan	900000

	RankInTheLeague	Name	Salary
1	1	James Harden	38000000
2	1	Russell Westbrook	38000000
3	2	Lebron James	37000000
4	3	Paul George	33000000
5	4	Kawhi Leonard	32000000
6	5	Anthony Davis	27000000
7	6	Giannis Antetoko...	24000000
8	7	George Hill	8000000
9	8	Jayson Tatum	7000000
10	9	Tacko Fall	1000000

Q21 => How many players got draftet on the same month and year, also only on the same year, what about in total drafted?

```
select DATEPART(YEAR,player_draftDate) as 'Year', DATEPART(MONTH,player_draftDate)as
'Month',count(player_id)
from player inner join team on player_team = team_id
group by rollup(DATEPART(YEAR,player_draftDate), DATEPART(MONTH,player_draftDate));
```

Results | Messages

	Year	Month	(No column name)
1	2003	6	1
2	2003	NULL	1
3	2008	4	1
4	2008	7	1
5	2008	NULL	2
6	2009	1	1
7	2009	NULL	1
8	2010	4	1
9	2010	NULL	1
10	2011	3	1
11	2011	NULL	1
12	2012	10	1
13	2012	NULL	1
14	2013	2	1
15	2013	NULL	1
16	2017	12	1
17	2017	NULL	1
18	2019	1	1
19	2019	NULL	1
20	NU...	NULL	10

```
Q21 => What is the total amount spent to pay the players? how much exactly does each of the team pay for their players, and the
average of the salary of the players on that team

select team_name, sum(player_salary) as TotalSpent, avg(player_salary) as AvgSalary
from player inner join team on player_team = team_id
group by rollup(team_name);
```

Results | Messages

	team_name	TotalSpent	AvgSalary
1	Boston Celtics	8000000	4000000
2	Houston Rockets	76000000	38000000
3	Los Angeles Clippers	65000000	32500000
4	Los Angeles Lakers	64000000	32000000
5	Milwaukee Bucks	32000000	16000000
6	NULL	245000000	24500000

12 DML: 2 insert, 2 update, 2 delete (Q23-Q28)

```
Q23 => Insert a random player in the league.

INSERT INTO player
VALUES(11, 'Charles Gonçalves', Rand()*(30-20)+20, 'PG',
       ROUND(RAND()*(2.10-1.70)+1.70,1),
       ROUND(RAND()*(85-65)+65,1),
       RAND()*(15000000-5000000)+5000000,
       ROUND(Rand()*(100-0)+0,1),'2003-06-09', Rand()*(5-1)+1);

select * from player inner join team on player_team = team_id where player_id = 11
```

Results | Messages

	player_id	player_name	player_age	player_position	player_height	player_weight	player_salary	player_fieldGoal	player_draftDate	player_team	team_id	team_name	team_conference	team_foundationDate	team_championshipsWon	team_arena
1	11	Charles Gonçalves	28	PG	1.7	74.1	10030142	48.5	2003-06-09	2	2	Los Angeles Clippers	Western	1970-09-30	0	Staples Center

Q24 => Update the team table by adding a new column “jersey Color”

```
alter table team
add team_jerseyColor nvarchar(50)
go

update team
set team_jerseyColor = case team_id
                       when 1 then  'Black and Yellow'
                                    when 2 then  'White, Red and Blue'
                                    when 3 then  'Black and Red'
                                    when 4 then  'Black and Green'
                                    when 5 then  'White and Green'
                                end;
select team_name, team_jerseyColor
from team;
go
```

Results | Messages

	team_name	team_jerseyColor
1	Los Angeles Lakers	Black and Yellow
2	Los Angeles Clippers	White, Red and Blue
3	Houston Rockets	Black and Red
4	Milwaukee Bucks	Black and Green
5	Boston Celtics	White and Green

Q25 => The player’s Name, Salary, draft date, by which team, and the years of waiting for the next draft

```
if object_id('tableau', 'U') is not null drop table tableau;
go

create table tableau(
        Drafted nvarchar(50),
              DraftDate date,
              TeamName nvarchar(50),
              Salary int,
              YearsToNextDraft int
)
insert into tableau
select Drafted, DraftDate, team_name, salary, YearsToNextDraft
from (select p1.player_team as team, p1.player_name as Drafted,  p1.player_draftDate
as DraftDate, datediff(year, p1.player_draftDate, p2.player_draftDate) as
YearsToNextDraft,
             p1.player_salary as salary
      from player p1 inner join player p2 on p1.player_id = p2.player_id - 1) as
subquery inner join team on team = team_id

select * from tableau;
```

Results | Messages

	Drafted	DraftDate	TeamName	Salary	YearsToNextDraft
1	Lebron James	2003-06-09	Los Angeles Lakers	37000000	5
2	Russell Westbrook	2008-04-10	Houston Rockets	38000000	0
3	George Hill	2008-07-17	Milwaukee Bucks	8000000	1
4	James Harden	2009-01-22	Houston Rockets	38000000	1
5	Paul George	2010-04-21	Los Angeles Clipp...	33000000	1
6	Kawhi Leonard	2011-03-14	Los Angeles Clipp...	32000000	1
7	Anthony Davis	2012-10-15	Los Angeles Lakers	27000000	1
8	Giannis Antetoko...	2013-02-03	Milwaukee Bucks	24000000	4
9	Jayson Tatum	2017-12-03	Boston Celtics	7000000	2
10	Tacko Fall	2019-01-30	Boston Celtics	1000000	-16

```
Q26 => Add the a column team_playOffMAde using Rand(), which Is equal to 1 if the team passed or 0 if the team didn't pass to the play
offs
alter table team
add team_playOffMade nvarchar(50)
go

update team
set team_playOffMade = case team_id
                       when 1 then Convert(nvarchar,floor( Rand()*(2 - 0) + 0))
                                   when 2 then Convert(nvarchar,floor( Rand()*(2 - 0) + 0))
                                   when 3 then Convert(nvarchar,floor( Rand()*(2 - 0) + 0))
                                   when 4 then Convert(nvarchar,floor( Rand()*(2 - 0) + 0))
                                   when 5 then Convert(nvarchar,floor( Rand()*(2 - 0) + 0))
                                   end;

update team
set team_playOffMade = case team_playOffMade
                       when '0' then 'Did not make to the Play Off'
                                   when '1' then 'Made to the PLay Off'
                                   end;
select * from team;
```

	team_id	team_name	team_conference	team_foundationDate	team_championshipsWon	team_arena	team_jerseyColor	team_playOffMade
1	1	Los Angeles Lakers	Western	1947-10-06	17	Staples Center	Black and Yellow	Made to the PLay Off
2	2	Los Angeles Clippers	Western	1970-09-30	0	Staples Center	White, Red and Blue	Made to the PLay Off
3	3	Houston Rockets	Western	1967-11-26	2	Toyota Center	Black and Red	Did not make to th...
4	4	Milwaukee Bucks	Eastern	1968-05-07	1	Fiserv Forum	Black and Green	Made to the PLay Off
5	5	Boston Celtics	Eastern	1946-06-09	17	TD Garden	White and Green	Did not make to th...

Q27 => All the players that belong to the teams that didn't make to the play off should be deleted.

```
delete player
from player inner join team on player_team = team_id
where team.team_playOffMade = 'Did not make to the Play Off';
go
```

```
select team_name as remainingTeam, count(player_id) as players, avg(player_fieldGoal)
as TeamFG
from player inner join team on player_team = team_id
group by team_name;
```

	remainingTeam	players	TeamFG
1	Boston Celtics	2	61.8
2	Los Angeles Lakers	2	49.8
3	Milwaukee Bucks	3	62.9

Q28 => If at this stage there are still teams with 0 championships won, their players should be deleted from the table.

	player_name	team_name	team_championshipsWon
1	Lebron James	Los Angeles Lakers	17
2	Anthony Davis	Los Angeles Lakers	17
3	Jayson Tatum	Boston Celtics	17
4	Tacko Fall	Boston Celtics	17
5	George Hill	Milwaukee Bucks	1
6	Giannis Antetokounmpo	Milwaukee Bucks	1
7	Charles Gonçalves	Milwaukee Bucks	1