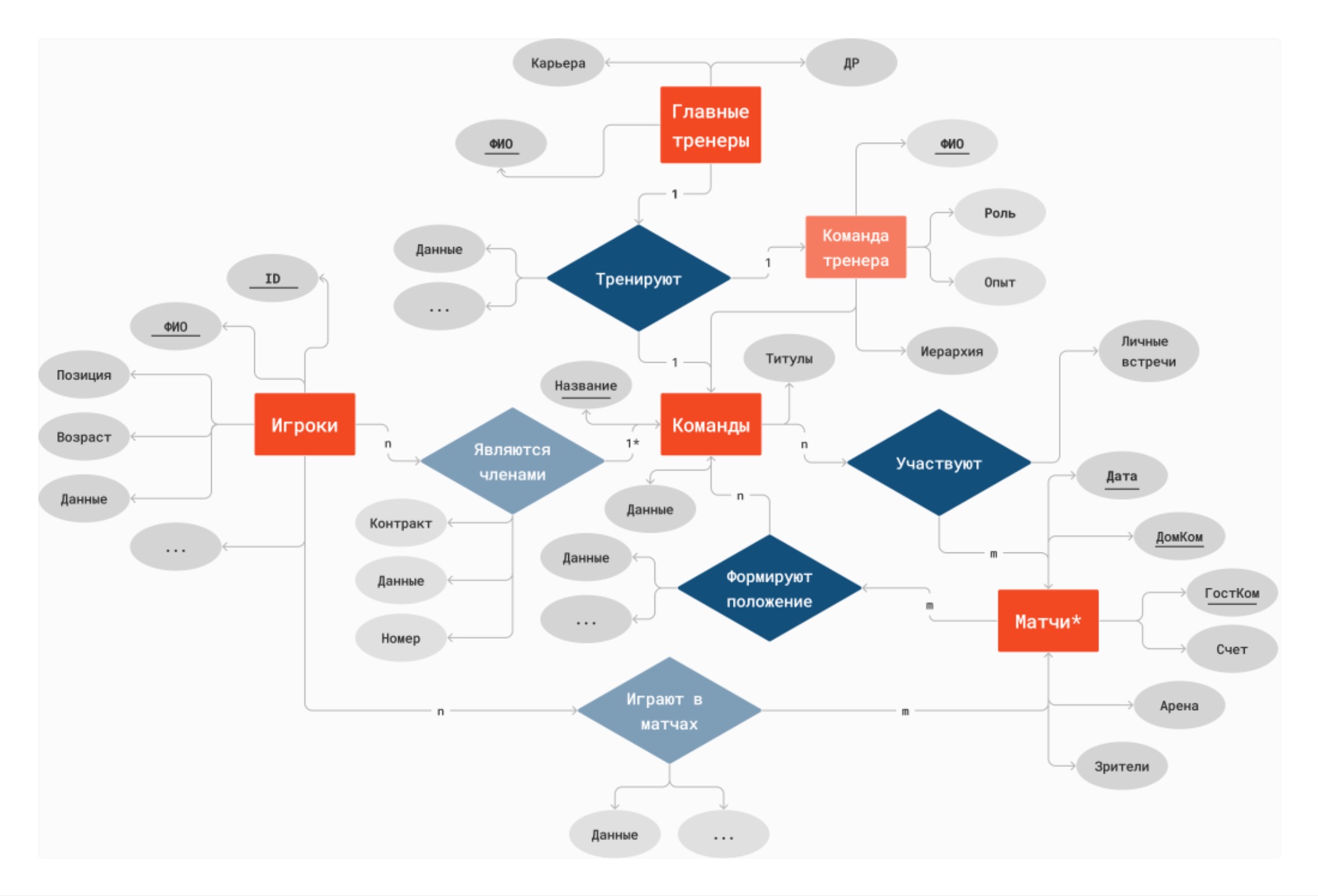
Краткое описание проекта:

Работать с выдуманными данными не интересно и не так информативно, поэтому я сделал небольшую базу данных национальной баскетбольной ассоциации (США), для этого я спарсил и предобработал данные с сайт [basketball-reference.com](https://www.basketball-reference.com/), использовав Google Sheet. Также данные были преобразованы в .csv и импортированы в [sqliteonline.com](https://sqliteonline.com/), где и завершилась настройка.

Стоит отметить, что данных оказалось слишком много, поэтому код будет представлен здесь лишь частично, а все insert’ы будут вынесены отдельно. Дополнять данные до полной картины всегда и везде у меня не хватило сил, поэтому то как правильно организовывать данные лишь обозначено поверхностно (например, я добавляю столбец, говорю зачем он нужен, но не заполняю его данными полностью). Это все происходило, потому что далеко не все полезные данные были на представленных ресурсах.

ER-diagram:



Пояснения к ER-diagram:

Я делал ER-диаграмму, зная что буду использовать определенные статистические метрики, поэтому атрибуты: данные и … следует воспринимать именно как намеренные пропуски, чтобы не захламлять диаграмму (приведу пример данных главного тренера, который сейчас тренируют команду: Coach, Team, SeasonsWithFranchise, SeasonsOverall, GamesCurRegul, WinCurRegul, LoseCurRegul,...).

Приглушенные цвета в *некоторых* сущностях и связях я использовал, чтобы показать, что данные по ним не полны и представляют лишь ознакомительный характер (за основу таких данных взяты данные команды Golden State Warriors, их тренера Steve Kerr, и их главной звезды Stephen Curry).

Перейдем наконец к практическим соображениям таблицы. Важно отметить, что по большей части мы будем рассматривать регулярный сезон 2023-2024. Поэтому связи между главными тренерами и командами обозначены как 1 к 1, потому что в рамках 1 момента иначе быть не может. При этом стоить отметить, что, конечно тренеры меняют свою работу и за свою жизнь могут тренировать разные команды. То же самое касается Игроков и Команды в которых они играют. Поэтому на связи являются членами со стороны команд обозначена 1\*.Т.е. в рамках 1 момента игрок находится в одной команде. При этом даже за 1 сезон, как мы позже выясним, 75 игроков сменили команды.

Связь Формирует положение представлена во многом из-за статистического удобства, нежели вызвана большой необходимостью, она помогает хранить как итоговое положение команд в таблице, так и их средние метрики по сезону.

Матчи\* - представлены все результаты по матчам за весь сезон 2023-2024. Именно поэтому Игрок может участвовать в множестве матчей, а в одном матче может участвовать множество игроков (вообще до 24, но это уже детали).

Последнее замечание, во многом составные ключи являются скорее требованием задания, нежели практической необходимостью, ведь их всегда можно заменить на простые ID.Примерно также обстоит дело с командой тренера, это отношение нужно, чтобы просто показать материализованный путь, не более (хотя все данные и реально, придумана лишь их иерархия на примере тренерской команды Steve Kerr)

**SQL**

Ограничения целостности, обозначение ключей, создание таблиц:

Сначала просто загрузим все таблицы и укажем ограничения, которые придут в голову, также сразу укажем primary key.

//

DROP TABLE IF EXISTS matches;

CREATE TABLE matches (

Date DATE NOT NULL,

Visitor VARCHAR(50) NOT NULL,

PointsVisitor INTEGER CHECK (PointsVisitor >= 0),

Home VARCHAR(50) NOT NULL,

PointsHome INTEGER CHECK (PointsHome >= 0),

Attendance REAL CHECK (Attendance >= 0),

Arena VARCHAR(50) NOT NULL,

PRIMARY KEY (Date, Visitor, Home)

);

INSERT INTO matches (Date, Visitor, PointsVisitor, Home, PointsHome, Attendance, Arena) VALUES

('10/24/2023','Los Angeles Lakers','107','Denver Nuggets','119','19.842','Ball Arena'),

//

//

DROP TABLE IF EXISTS players;

CREATE TABLE players (

Rank INTEGER not null,

Player VARCHAR(50) NOT NULL,

Pos VARCHAR(20) NOT NULL,

Age INTEGER CHECK (Age > 0 AND Age < 100),

Team VARCHAR(5) NOT NULL,

Games INTEGER CHECK (Games >= 0),

MinutesPlayed REAL CHECK (MinutesPlayed >= 0),

EfficientFieldGoalsRate REAL CHECK (EfficientFieldGoalsRate >= 0 AND EfficientFieldGoalsRate <= 1),

FreeThrowsRate REAL CHECK (FreeThrowsRate >= 0 AND FreeThrowsRate <= 1),

Rebounds REAL CHECK (Rebounds >= 0),

Assists REAL CHECK (Assists >= 0),

Steals REAL CHECK (Steals >= 0),

Blocks REAL CHECK (Blocks >= 0),

Turnovers REAL CHECK (Turnovers >= 0),

PF REAL CHECK (PF >= 0),

Points REAL CHECK (Points >= 0),

primary key(rank,player)

);

INSERT INTO players (Rank, Player, Pos, Age, Team, Games, MinutesPlayed, EfficientFieldGoalsRate, FreeThrowsRate, Rebounds, Assists, Steals, Blocks, Turnovers, PF, Points) VALUES

('1','Buddy Hield','SF-SG','31','TOT','84','25.7','.565','.881','3.2','2.8','0.8','0.5','1.2','2.0','12.1'),

//

Здесь будет небольшое пояснение - тк команды часто обозначают сокращенно у меня возникли проблемы с правильным неймингом, поэтому было принято решение в таблице teambase создать два одинаковых столбца с разными удобными обозначениями команд. Знаю, что для нормальных форм это очень плохо, но в данном случае так просто удобнее. К тому же primary key - укажем только Franchise, хотя и TeamCode уникален.

//

DROP TABLE IF EXISTS teamsbase;

CREATE TABLE teamsbase (

Franchise VARCHAR(100) PRIMARY KEY,

Years INTEGER CHECK (Years > 0),

WinLoseRate REAL CHECK (WinLoseRate >= 0.0 AND WinLoseRate <= 1.0),

Champ INTEGER CHECK (Champ >= 0),

TeamCode VARCHAR(5)

);

INSERT INTO teamsbase (Franchise, Years, WinLoseRate, Champ, TeamCode) VALUES

('Atlanta Hawks', 75, 0.493, 1, 'ATL'),

//

//

DROP TABLE IF EXISTS coachall;

CREATE TABLE coachall (

Coach VARCHAR(100) NOT NULL,

FromYear INTEGER CHECK (FromYear >= 1900 AND FromYear <= 2024),

ToYear INTEGER CHECK (ToYear >= 1900 AND ToYear <= 2024 AND ToYear >= FromYear),

Birth\_Date DATE,

PRIMARY KEY (Coach)

);

INSERT INTO coachall (Coach,FromYear,ToYear,Birth\_Date) VALUES

('Rick Adelman\*','1989','2014','6/16/1946'),

//

Здесь добавил материализованный путь

//

DROP TABLE IF EXISTS coaching\_staff;

CREATE TABLE coaching\_staff (

name VARCHAR(50) PRIMARY KEY,

role VARCHAR(50),

start\_season VARCHAR(10),

path VARCHAR(50)

);

INSERT INTO coaching\_staff (name, role, start\_season, path) VALUES

('Steve Kerr', 'Head Coach', '2014-15', '1.');

//

Теперь дополнительные таблицы -  таблицы связей, в некоторых есть foreign keys. А некоторые просто дополнительные (приглушенные тона в таблицу er-diagram)

//

DROP TABLE IF EXISTS coach;

CREATE TABLE coach (

Coach VARCHAR(50) NOT NULL REFERENCES coachall(coach),

Team VARCHAR(5) NOT NULL REFERENCES teambase(teamcode),

SeasonsWithFranchise INTEGER CHECK (SeasonsWithFranchise >= 0),

SeasonsOverall INTEGER CHECK (SeasonsOverall >= 0),

GamesCurRegul INTEGER CHECK (GamesCurRegul >= 0),

WinCurRegul INTEGER CHECK (WinCurRegul >= 0 AND WinCurRegul <= GamesCurRegul),

LoseCurRegul INTEGER CHECK (LoseCurRegul >= 0 AND LoseCurRegul <= GamesCurRegul),

GamesCareerRegul INTEGER CHECK (GamesCareerRegul >= 0),

WinCareerRegul INTEGER CHECK (WinCareerRegul >= 0 AND WinCareerRegul <= GamesCareerRegul),

LoseCareerRegul INTEGER CHECK (LoseCareerRegul >= 0 AND LoseCareerRegul <= GamesCareerRegul),

WinRateCareerRegul REAL CHECK (WinRateCareerRegul >= 0.0 AND WinRateCareerRegul <= 1.0),

PRIMARY KEY (Coach, Team)

);

INSERT INTO coach (Coach, Team, SeasonsWithFranchise, SeasonsOverall, GamesCurRegul, WinCurRegul, LoseCurRegul, GamesCareerRegul, WinCareerRegul, LoseCareerRegul, WinRateCareerRegul) VALUES

('Quin Snyder','ATL','2','10','82','36','46','739','418','321','0.566'),

//

//

DROP TABLE IF EXISTS standing;

CREATE TABLE standing (

Rk INTEGER PRIMARY KEY,

Team VARCHAR(50) NOT NULL REFERENCES teamsbase(franchise),

Overall VARCHAR(10) NOT NULL CHECK (length(Overall) >= 5 AND Overall LIKE '%-%'),

Home VARCHAR(10) NOT NULL CHECK (length(Home) >= 3 AND Home LIKE '%-%'),

Road VARCHAR(10) NOT NULL CHECK (length(Road) >= 3 AND Road LIKE '%-%')

);

INSERT INTO standing (Rk, Team, Overall, Home, Road) VALUES

(1, 'Boston Celtics', '64-18', '37-4', '27-14'),

//

//

DROP TABLE IF EXISTS teamtoteam;

CREATE TABLE teamtoteam (

Rk INTEGER PRIMARY KEY,

Team VARCHAR(50) NOT NULL UNIQUE REFERENCES teamsbase(franchise),

ATL VARCHAR(10),

BOS VARCHAR(10),

BRK VARCHAR(10),

CHI VARCHAR(10),

CHO VARCHAR(10),

CLE VARCHAR(10),

DAL VARCHAR(10),

DEN VARCHAR(10),

DET VARCHAR(10),

GSW VARCHAR(10),

HOU VARCHAR(10),

IND VARCHAR(10),

LAC VARCHAR(10),

LAL VARCHAR(10),

MEM VARCHAR(10),

MIA VARCHAR(10),

MIL VARCHAR(10),

MIN VARCHAR(10),

NOP VARCHAR(10),

NYK VARCHAR(10),

OKC VARCHAR(10),

ORL VARCHAR(10),

PHI VARCHAR(10),

PHO VARCHAR(10),

POR VARCHAR(10),

SAC VARCHAR(10),

SAS VARCHAR(10),

TOR VARCHAR(10),

UTA VARCHAR(10),

WAS VARCHAR(10)

);

INSERT INTO teamtoteam (Rk, Team, ATL, BOS, BRK, CHI, CHO, CLE, DAL, DEN, DET, GSW, HOU, IND, LAC, LAL, MEM, MIA, MIL, MIN, NOP, NYK, OKC, ORL, PHI, PHO, POR, SAC, SAS, TOR, UTA, WAS) VALUES

('1','Atlanta Hawks','','2-2','1-3','1-2','1-3','1-3','0-2','0-2','3-0','1-1','2-0','0-4','1-1','1-1','1-1','1-3','1-2','1-1','1-1','1-2','1-1','3-1','2-2','1-1','1-1','0-2','2-0','2-2','1-1','3-1'),

//

//

DROP TABLE IF EXISTS currystats;

CREATE TABLE currystats (

GameNumber INTEGER PRIMARY KEY,

Date DATE NOT NULL,

Age VARCHAR(10) NOT NULL,

Opp VARCHAR(3) NOT NULL REFERENCES teamsbase(teamcode),

GS INTEGER NOT NULL CHECK (GS IN (0, 1)),

PTS INTEGER,

Efficiency INTEGER

);

INSERT INTO currystats (GameNumber, Date, Age, Opp, GS, PTS, Efficiency) VALUES

('1','10/24/2023','35-224','PHO','1','27','-9'),

//

//

DROP TABLE IF EXISTS teamroster;

CREATE TABLE teamroster (

Number INTEGER NOT NULL,

Player VARCHAR(100) NOT NULL REFERENCES players(player),

Position VARCHAR(10) NOT NULL,

Height VARCHAR(10) NOT NULL,

Weight INTEGER CHECK (Weight > 0),

BirthDate DATE NOT NULL,

Country VARCHAR(50) NOT NULL,

YearsExperience VARCHAR(10) NOT NULL,

PRIMARY KEY (Number, Player)

);

INSERT INTO teamroster (Number,Player,Position,Height,Weight,BirthDate,country,YearsExperience) VALUES

('11','Klay Thompson','SF','6-6','215','February 8, 1990','us','10'),

//

Теперь переходим к запросам SQL (в презентации подробнее):

-- Матчи NBA очень популярны и даже в регулярный сезон там случаются аншлаги, но поскольку я не успел добавить вместимость стадионов, то попробуем найти самый посещаемый стадион, выбрав 10 самых посещаемых матчей. Скорее всего стадион, который нам выпадет - и есть самый большой стадион в лиге.

SELECT Home, Visitor, Arena, MAX(Attendance) AS MaxAttendance

FROM matches

GROUP BY Home, Visitor

ORDER BY MaxAttendance DESC

LIMIT 10;

и это United Center стадион команды Chicago Bulls (где когда-то играл Джордан)

—найдем уникальные позиции игроков

SELECT DISTINCT Pos FROM players;

много же их (там есть и полупозиции, дело в том, что как и в футболе - игрок может выходить на разных позициях, тем более важный и принципиальный отличий между смежными позициями нет)

-- Кто больше забивает, если ориентироваться на позицию

SELECT Pos, AVG(Points) AS AvgPoints

FROM players

GROUP BY Pos

ORDER BY AvgPoints DESC;

Limit 1;

Оказалось, что это PointGuard-ShooterGuard

—-Теперь посмотрим на всех, кто забивает больше пары трехочковых

SELECT Pos, AVG(Points) AS AvgPoints

FROM players

GROUP BY Pos

HAVING AVG(Points) > 6

ORDER BY AvgPoints DESC;

–Теперь посмотрим на личностей, кто больше всех забивает в команде

SELECT Player, Points

FROM players

WHERE Points = (

SELECT MAX(Points)

FROM players p

WHERE p.Team = players.Team

);

Наш Стефен Кари на месте, хотя и забивает в среднем сильно меньше Луки Дончича

–А есть ли среди них эффективные игроки, которые забивают при этом тоже достаточно

SELECT Player, EfficientFieldGoalsRate, points

FROM players

where points>10

ORDER BY EfficientFieldGoalsRate DESC

limit 10;

Нету.

Из звезд я вижу только Зайона Уильямса, он центровой, так что понятно, почему он точно бросает.

–А сколько в среднем очков забивает игрок за сезон

SELECT AVG(Points) AS AVGPoints

FROM players;

неплохо - 9.22

–Выберем матчи, где домашняя команда забила не просто больше, а больше всех за сезон

SELECT Date, Home, Visitor, PointsHome, PointsVisitor

FROM matches

WHERE PointsHome = (

SELECT MAX(PointsHome)

FROM matches

);

157 впечатляет!

–переходим к тренерам, раз уж нам нравится Стив Керр, то добавим как будто он сыграл еще одну игру

UPDATE coach

SET GamesCurRegul = GamesCurRegul + 1

WHERE Coach = 'Steve Kerr';

–А как же нам связаться с таким прекрасным Стивом, добавим ему email

ALTER TABLE coach

ADD COLUMN Email VARCHAR(100);

UPDATE coach

SET Email = 'steve.kerr@warriorsfun.com'

WHERE Coach = 'Steve Kerr';

–Удалим теперь, чтобы не его никто не смог побеспокоить

ALTER TABLE coach

DROP COLUMN Email;

–Теперь проверим насколько наш стив керр отлично себя показал в этом сезоне

SELECT Coach, WinCurRegul, GamesCurRegul, (WinCurRegul \* 1.0 / GamesCurRegul) AS CurrentWinRate

FROM coach

WHERE (WinCurRegul \* 1.0 / GamesCurRegul) > (

SELECT AVG(WinCurRegul \* 1.0 / GamesCurRegul)

FROM coach

WHERE GamesCurRegul > 0)

Order by CurrentWinRate desc;

Эх, он где-то в середине

–Ух, теперь наконец дойдем до сложного, у нас есть таблица standing - проверим по матчам - правильно ли она подсчитано, для этого, например, посчитаем домашние победы команд

WITH HomeWins AS (

SELECT Home AS Team, COUNT(\*) AS Wins

FROM matches

WHERE PointsHome > PointsVisitor

GROUP BY Home

)

SELECT hw.Team, hw.Wins AS HomeWins, s.Home AS RecordedHome

FROM HomeWins hw

JOIN standing s ON hw.Team = s.Team;

Ну пожалуй все, мы использовали все функции какие только можно.