

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Базы данных»
Тема: Реализация базы данных в СУБД MySQL

Студент гр. 9383

Нистратов Д.Г.

Преподаватель

Заславский М.М.

Санкт-Петербург

2021

Цель работы.

Изучение создания СУБД MySQL.

Задание (12 Вариант).

Необходимо развернуть MySQL локально:

- Написать запросы для создания таблиц из предыдущей лабораторной работы
- Заполнить тестовыми данными: 5-10 строк на каждую таблицу, обязательно наличие связи между ними, данные приближены к реальности.
- Написать запросы к БД, отвечающие на вопросы из предыдущей лабораторной работы
- Исходный код выложить на www.db-fiddle.com для проверки работоспособности
- Исходный код в виде .sql файла загрузить в виде PR в репо
- В отчете описать цель, текст задания в соответствии с вариантом, скриншоты работы с СУБД MySQL (Datagrip, MySQL Server, mysql cli), скриншоты на каждый запрос (или группу запросов) на изменение/таблицы с выводом результатов (ответ), исходный код в приложении, ссылку на исходный код www.db-fiddle.com в приложении, ссылка на PR в приложении, вывод

Описание предметной области:

Пусть требуется создать программную систему, предназначенную для организаторов соревнований по футболу в рамках первенства страны. Такая система должна обеспечивать хранение сведений о командах, участвующих в

первенстве, об игроках команд, о расписании встреч и их результатах, о цене билетов на игры. Сведения о команде — название команды, город, где она базируется, имя главного тренера, место в таблице прошлого сезона, расписание встреч. В один день команда может участвовать только в одной встрече. Сведения об игроке включают в себя фамилию и имя игрока, его возраст, номер и амплуа в команде. Сведения о стадионе, на котором происходит встреча содержат город, в котором он находится, название стадиона, и его вместимость. Цена билета на матч зависит от вместимости стадиона и положения встречающихся команд в турнирной таблице прошлого сезона (наибольшая - при игре тройки призеров, наименьшая — при игре тройки аутсайдеров). Организаторы соревнований должны иметь возможность внести изменения в данные о составе команд, перенести встречу. Им могут потребоваться следующие сведения:

- Даты встреч указанной команды, ее противники и счет?
- Номера и фамилии игроков команд, участвовавших во встрече, которая проходила в указанный день в указанном городе?
- Цена, билета на матч между указанными командами?
- Игрок, забивший в турнире наибольшее количество мячей?
- Команды, имеющие наилучшую и наихудшую разницу забитых и пропущенных мячей?
- Самый молодой участник турнира?
- Команды, занявшие призовые места?

Выполнение работы.

В ходе выполнения лабораторной работы была создана и заполнена СУБД данными из предыдущей лабораторной работы, см. Рисунок 1-6.

```
CREATE TABLE Team (
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    place smallint unsigned NULL,
    coach VARCHAR(50),
    PRIMARY KEY (team_name, team_city)
);
```

Рисунок 1

```
CREATE TABLE Matchevent (
    event_id bigint unsigned AUTO_INCREMENT PRIMARY KEY,
    event_date DATE,
    price decimal(10, 2)
);
```

Рисунок 2

```
CREATE TABLE Stadium (
    name VARCHAR(50) PRIMARY KEY,
    city VARCHAR(50),
    size bigint unsigned
);
```

Рисунок 3

```
CREATE TABLE Player (
    number smallint unsigned,
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    name varchar(20),
    second_name varchar(20),
    patronymic varchar(20) NULL,
    age tinyint unsigned,
    amplya enum('Guardian', 'Attacker', 'Goalkeeper'),
    PRIMARY KEY (number, team_name, team_city),
    FOREIGN KEY (team_name, team_city) REFERENCES Team (team_name, team_city)
);
```

Рисунок 4

```
CREATE TABLE Goal (
    ball_id SERIAL,
    event_id bigint unsigned,
    number smallint unsigned,
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    PRIMARY KEY (ball_id, event_id, number, team_name, team_city),
    FOREIGN KEY (number, team_name, team_city) REFERENCES Player (number, team_name, team_city),
    FOREIGN KEY (event_id) REFERENCES Matchevent (event_id)
);
```

Рисунок 5

```
CREATE TABLE Participant (
    participant_id SERIAL,
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    event_id bigint unsigned,
    name VARCHAR(50),
    PRIMARY KEY (team_name, team_city, event_id, name),
    FOREIGN KEY (team_name, team_city) REFERENCES Team (team_name, team_city),
    FOREIGN KEY (event_id) REFERENCES Matchevent (event_id),
    FOREIGN KEY (name) REFERENCES Stadium (name)
);
```

Рисунок 6

Также были описаны запросы выполняющие ответы на вопросы из предыдущей лабораторной работы, см. Рисунок 7-12.

- Даты встреч указанной команды, ее противники и счет?

	event_date	enemy	Scored	Missed
1	2012-12-30	PGA	<null>	1
2	2015-03-20	Barcelona	2	<null>
3	2018-07-20	Barcelona	1	2

Рисунок 7

- Номера и фамилии игроков команд, участвовавших во встрече, которая проходила в указанный день в указанном городе?

	number	second_name
1	1	Anome
2	4	Salsa
3	21	Aliaze
4	3	Petrov
5	4	Mashkov
6	11	Morozov

Рисунок 8

- Цена, билета на матч между указанными командами?

	price
1	1500.00
2	700.99

Рисунок 9

- Игрок, забивший в турнире наибольшее количество мячей?

	player.number	player.team_name	player.team_city	name	second_name	patronymic	age	amplua	ball_id	event
1	3	Zenit	Saint-Petersburg	Sergey	Petrov	<null>	18	Guardian	2	

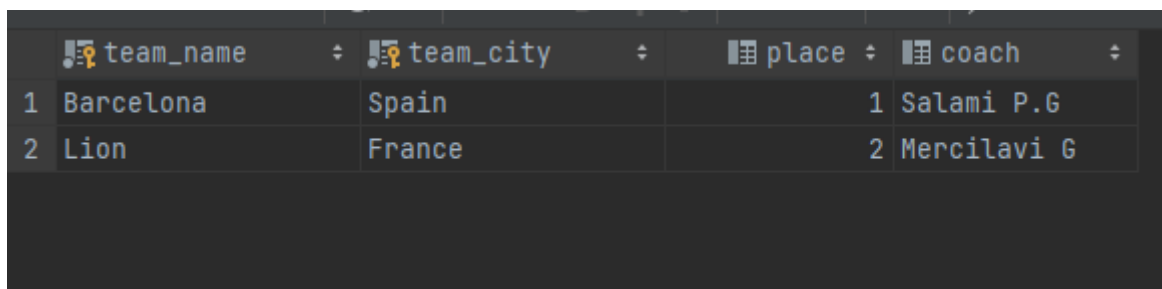
Рисунок 10

- Самый молодой участник турнира?

	number	team_name	team_city	name	second_name	patronymic	age	amplua
1	3	Zenit	Saint-Petersburg	Sergey	Petrov	<null>	18	Guardian
2	4	PGA	BRAZIL	Lamami	Salsa	Gevo	18	Attacker

Рисунок 11

- Команды, занявшие призовые места?



	team_name	team_city	place	coach
1	Barcelona	Spain	1	Salami P.G
2	Lion	France	2	Mercilavi G

Рисунок 12

Выводы.

В ходе выполнения лабораторной работы была произведена разработка собственной СУБД MySQL.

Приложение А.

Ссылка на pull-request: <https://github.com/moevm/sql-2021-9383/pull/20>

Ссылка на db-fiddle: <https://www.db-fiddle.com/f/eNtvge1BFsqEyhWsbbLBS/6>

Приложение Б.

```
CREATE DATABASE competition;
```

```
USE competition;
```

```
CREATE TABLE Team (  
    team_name VARCHAR(20),  
    team_city VARCHAR(50),  
    place smallint unsigned NULL,  
    coach VARCHAR(50),  
    PRIMARY KEY (team_name, team_city)  
);
```

```
INSERT INTO Team (team_name, team_city, place, coach)  
VALUES('Zenit', 'Saint-Petersburg', 5, 'Andreev A.A');
```

```
INSERT INTO Team (team_name, team_city, coach)  
VALUES('PGA', 'BRAZIL', 'Pergo O');
```

```
INSERT INTO Team (team_name, team_city, place, coach)  
VALUES('Barcelona', 'Spain', 1, 'Salami P.G');
```

```
INSERT INTO Team (team_name, team_city, place, coach)  
VALUES('Lion', 'France', 2, 'Mercilavi G');
```

```
INSERT INTO Team (team_name, team_city, coach)  
VALUES('Milan', 'France', 'Giorvano J.J');
```

```
CREATE TABLE Matchevent (  
    event_id bigint unsigned AUTO_INCREMENT PRIMARY KEY,  
    event_date DATE,  
    price decimal(10, 2)  
);
```

```
INSERT INTO Matchevent (event_date, price)  
VALUES('2012-12-30', 1000.12);
```

```
INSERT INTO Matchevent (event_date, price)
VALUES('15.03.2015', 1500);
```

```
INSERT INTO Matchevent (event_date, price)
VALUES('18.07.2019', 700.99);
```

```
INSERT INTO Matchevent (event_date, price)
VALUES('15.06.2015', 550);
```

```
INSERT INTO Matchevent (event_date, price)
VALUES('18.11.2021', 9999.99);
```

```
CREATE TABLE Stadium (
  name VARCHAR(50) PRIMARY KEY,
  city VARCHAR(50),
  size bigint unsigned
);
```

```
INSERT INTO Stadium (name, city, size)
VALUES('Power', 'Moscow', 10000);
```

```
INSERT INTO Stadium (name, city, size)
VALUES('California stadium', 'California', 3000);
```

```
INSERT INTO Stadium (name, city, size)
VALUES('LLSP', 'Tokyo', 100000);
```

```
INSERT INTO Stadium (name, city, size)
VALUES('Beaver Stadium', 'North America', 106572);
```

```
INSERT INTO Stadium (name, city, size)
VALUES('Camp Nou', 'Barcelona', 99354);
```

```

CREATE TABLE Player (
    number smallint unsigned,
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    name varchar(20),
    second_name varchar(20),
    patronymic varchar(20) NULL,
    age tinyint unsigned,
    amplua enum('Guardian', 'Attacker', 'Goalkeeper'),
    PRIMARY KEY (number, team_name, team_city),
    FOREIGN KEY (team_name, team_city) REFERENCES Team (team_name, team_city)
);

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, age, amplua)
VALUES (21, 'PGA', 'BRAZIL', 'Pedro', 'Aliaze', 24, 'Guardian');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, age, amplua)
VALUES (1, 'PGA', 'BRAZIL', 'Chili', 'Anome', 29, 'Attacker');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, patronymic, age,
amplua)
VALUES (4, 'PGA', 'BRAZIL', 'Lamami', 'Salsa', 'Gevo', 18, 'Attacker');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, age, amplua)
VALUES (11, 'Zenit', 'Saint-Petersburg', 'Andrey', 'Morozov', 25, 'Guardian');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, age, amplua)
VALUES (3, 'Zenit', 'Saint-Petersburg', 'Sergey', 'Petrov', 18, 'Guardian');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, patronymic, age,
amplua)
VALUES (4, 'Zenit', 'Saint-Petersburg', 'Vladimir', 'Mashkov', 'Andreevich', 23, 'Goalkeeper');

```

```

INSERT INTO Player (number, team_name, team_city, name, second_name, age, amplua)
VALUES (3, 'Barcelona', 'Spain', 'Barachelo', 'Si', 19, 'Attacker');

```

INSERT INTO Player (number, team_name, team_city, name, second_name, patronymic, age, amplua)

VALUES (4, 'Barcelona', 'Spain', 'Kero', 'Messo', 'El', 32, 'Goalkeeper');

CREATE TABLE Goal (

ball_id SERIAL,

event_id bigint unsigned,

number smallint unsigned,

team_name VARCHAR(20),

team_city VARCHAR(50),

PRIMARY KEY (ball_id, event_id, number, team_name, team_city),

FOREIGN KEY (number, team_name, team_city) REFERENCES Player (number, team_name, team_city),

FOREIGN KEY (event_id) REFERENCES Matchevent (event_id)

);

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (1, 21, 'PGA', 'BRAZIL');

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (2, 3, 'Zenit', 'Saint-Petersburg');

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (2, 3, 'Zenit', 'Saint-Petersburg');

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (3, 3, 'Zenit', 'Saint-Petersburg');

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (3, 3, 'Barcelona', 'Spain');

INSERT INTO Goal (event_id, number, team_name, team_city)

VALUES (3, 4, 'Barcelona', 'Spain');

CREATE TABLE Participant (

```

    participant_id SERIAL,
    team_name VARCHAR(20),
    team_city VARCHAR(50),
    event_id bigint unsigned,
    name VARCHAR(50),
    PRIMARY KEY (team_name, team_city, event_id, name),
    FOREIGN KEY (team_name, team_city) REFERENCES Team (team_name, team_city),
    FOREIGN KEY (event_id) REFERENCES Matchevent (event_id),
    FOREIGN KEY (name) REFERENCES Stadium (name)
);

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('PGA', 'BRAZIL', 1, 'LLSP');

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('Zenit', 'Saint-Petersburg', 1, 'LLSP');

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('Zenit', 'Saint-Petersburg', 2, 'Power');

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('Barcelona', 'Spain', 2, 'Power');

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('Zenit', 'Saint-Petersburg', 3, 'LLSP');

```

```

INSERT INTO Participant (team_name, team_city, event_id, name)
VALUES ('Barcelona', 'Spain', 3, 'LLSP');

```

```

# Даты встреч указанной команды, ее противники и счет?
SELECT event_date, team_name as enemy, (SELECT COUNT(*) from GOAL
WHERE team_name = 'Zenit' AND GOAL.event_id = participant.event_id
GROUP BY team_city, team_name) as Scored,
(SELECT COUNT(*) from GOAL
WHERE team_name != 'Zenit' AND GOAL.event_id = participant.event_id
GROUP BY team_city, team_name) as Missed

```

```

from participant
INNER JOIN matchevent m on participant.event_id = m.event_id
WHERE m.event_id in (SELECT event_id FROM participant z WHERE z.team_name = 'Zenit') AND
team_name != 'Zenit';

```

Номера и фамилии игроков команд, участвовавших во встрече, которая проходила в указанный день в указанном городе?

```

SELECT number, second_name FROM player
LEFT JOIN participant p ON player.team_name = p.team_name and player.team_city = p.team_city
LEFT JOIN matchevent m on p.event_id = m.event_id
LEFT JOIN stadium s on p.name = s.name
WHERE event_date = '2012-12-30' AND city = 'Tokyo';

```

#Цена, билета на матч между указанными командами?

```

SELECT price FROM matchevent
INNER JOIN participant p on matchevent.event_id = p.event_id
WHERE p.team_name = 'Zenit' and p.event_id in (SELECT event_id FROM participant WHERE
team_name = 'Barcelona');

```

#Игрок, забивший в турнире наибольшее количество мячей?

```

SELECT * FROM player
INNER JOIN (SELECT * from goal
group by number,team_name, team_city
ORDER BY COUNT(*) DESC
LIMIT 1) as v ON player.number = v.number and player.team_city = v.team_city and
player.team_name = v.team_name;

```

Самый молодой участник турнира ?

```

SELECT * FROM Player
WHERE age = (
SELECT MIN(age) FROM Player
);

```

#Команды, занявшие призовые места?

```

SELECT * from team
WHERE place BETWEEN 1 and 3;

```

