Федеральное государственное бюджетное образовательное учреждение высшего образования «Поволжский государственный технологический университет»

факультет: «информатики и вычислительной техники»

кафедра: «информатики и системного прогаммирования»

КУРСОВАЯ РАБОТА

по дисциплине «базы данных»

по теме «Информационный портал хоккейной лиги»

Выполнил:

студент группы ПС-31 Микушов М. В.

Проверил:

к.т.н., доцент кафедры ИиСП

Лучинин Захар Сергеевич

Йошкар-Ола 2018

**Описания предметной области, ее правил и ограничений.**

Структура лиги:

1. Восточная конференция:
   1. Дивзион метрополитан
   2. Дивзион атлантический
2. Западная конференция

2.1. Дивизион Центральный

2.2. Дивизион Тихоокеанский

Всего 31 команд. Каждай команда играет 82 матча.

Игры проходят в гостях и дома. 3 периода по 20 минут. Заявка на матч: 2 вратаря, 18 полевых игроков. На поле одновременно могут находиться 5 полевых игроков одной команды (минимальное кол-во игроков, с которыми можно начать матч: 5 игроков, 1 вратарь; минимальное число игроков, с которыми можно продолжать матч: 3 игрока; 1 вратарь. Если у Иначе техническое поражение 0:3.) Если обе команды не удовлетворяют требованиям – перенос матча на неопределенный срок. В игре м. участовать только игроки, занесенные в протокол матча.

За победу в основное время и в овертайме (в случае ничьи в основное время) +2 очка, поражение в овертайме +1 очко, в основное время 0 очков. Овертайм до первого гола или 10 мин. В случае ничьи +1 очко каждой команде участнице.

Роли в команде:

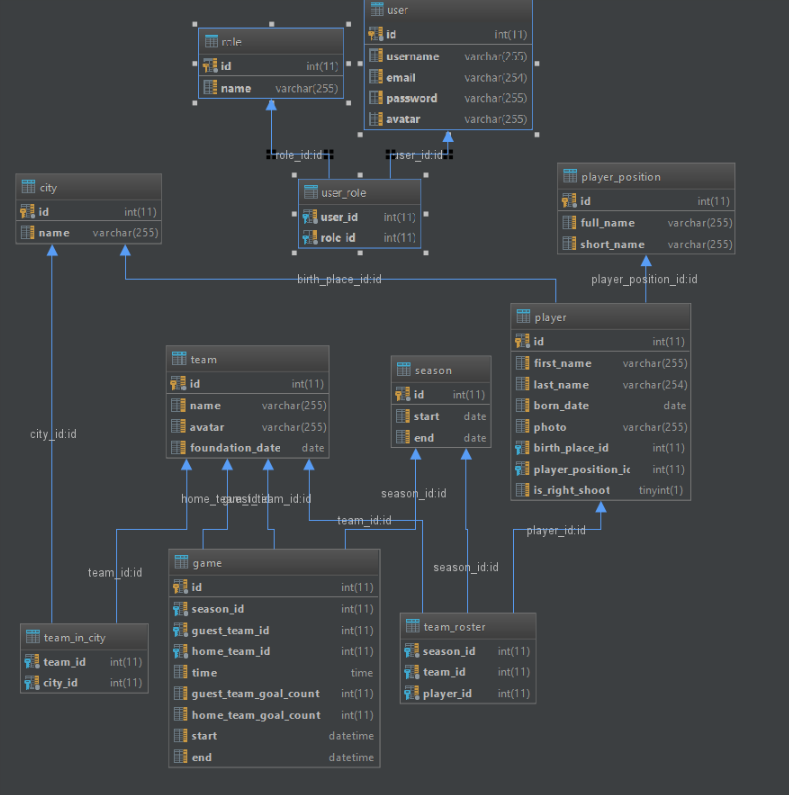
* Капитан (0 или 1 полевой игрок)
* Альтернативный капитан (2 или 3 полевых игрока).

Роль имеет только полевые игроки (<= 1).

События в игре:

* Гол (автор, 1-ая передача и 2-ая передача. Передачи м.б и м. не б.)
* Нарущение (тип нарущения наказние, наказуемый, время); Минимальное число полевых игроков 3. В случае удаления вратаря за отбывает полевой игрок.
* Потеря
* Перехват
* Силовой прием
* Офсайд
* Вбрасывание

**ER диаграмма**



**Примеры текстов SQL-запросов к базе данных**

*# 1) Получить таблицу результатов***SELECT** team\_id,  
 team\_name,  
 *SUM*(game\_count) **AS** game\_count,  
 *SUM*(goals\_scored) **AS** goals\_scored,  
 *SUM*(goals\_missed) **AS** goals\_missed,  
 *SUM*(goals\_scored) - *SUM*(goals\_missed) **AS** difference,  
 *SUM*(is\_win) **AS** count\_win,  
 *SUM*(is\_lose) **AS** count\_lose,  
 *SUM*(is\_overtime\_win) **AS** count\_overtime\_win,  
 *SUM*(is\_overtime\_lose) **AS** count\_overtime\_lose,  
 *SUM*(**point**) **AS** points  
**FROM** ((**SELECT** game.**id AS** game\_id,  
 *COUNT*(game.**id**) **AS** guest\_game\_count,  
 team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 *COUNT*(game.**id**) **AS** game\_count,  
 *SUM*(game.**home\_team\_goal\_count**) **AS** goals\_scored,  
 *SUM*(game.**guest\_team\_goal\_count**) **AS** goals\_missed,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_lose,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** > **'01:00:00'**) **AS** is\_overtime\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** > **'01:00:00'**) **AS** is\_overtime\_lose,  
 *SUM*(*getGamePoint*(**guest\_team\_goal\_count**, **home\_team\_goal\_count**, **time**)) **AS point  
 FROM** game  
 **LEFT JOIN** team **ON** game.**guest\_team\_id** = team.**id  
 LEFT JOIN** season s **on** game.**season\_id** = s.**id  
 WHERE** s.**start** >= **'2017-10-01' AND** s.**end** <= **'2018-05-01' AND** game.**end IS NOT NULL  
 GROUP BY** team\_id)  
 **UNION ALL** (**SELECT** game.**id AS** game\_id,  
 *COUNT*(game.**id**) **AS** home\_game\_count,  
 team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 *COUNT*(game.**id**) **AS** game\_count,  
 *SUM*(game.**home\_team\_goal\_count**) **AS** goals\_scored,  
 *SUM*(game.**guest\_team\_goal\_count**) **AS** goals\_missed,  
 *SUM*(game.**home\_team\_goal\_count** > game.**guest\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_win,  
 *SUM*(game.**home\_team\_goal\_count** < game.**guest\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_lose,  
 *SUM*(game.**home\_team\_goal\_count** > game.**guest\_team\_goal\_count AND** game.**time** > **'01:00:00'**) **AS** is\_overtime\_win,  
 *SUM*(game.**home\_team\_goal\_count** < game.**guest\_team\_goal\_count AND** game.**time** > **'01:00:00'**) **AS** is\_overtime\_lose,  
 *SUM*(*getGamePoint*(**home\_team\_goal\_count**, **guest\_team\_goal\_count**, **time**)) **AS point  
 FROM** game  
 **LEFT JOIN** team **ON** game.**home\_team\_id** = team.**id  
 LEFT JOIN** season s **on** game.**season\_id** = s.**id  
 WHERE** s.**start** >= **'2017-10-01' AND** s.**end** <= **'2018-05-01' AND** game.**end IS NOT NULL  
 GROUP BY** team\_id)) all\_game  
**GROUP BY** team\_id  
**ORDER BY** points **DESC**;  
  
*# /teams  
# Получить все камнды  
# 2)***SELECT** team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 team.**avatar As** team\_avatar,  
 city.**name AS** city\_name  
**FROM** team  
 **LEFT JOIN** team\_in\_city **ON** team.**id** = team\_in\_city.**team\_id  
 LEFT JOIN** city **ON** team\_in\_city.**city\_id** = city.**id**;  
  
*# /teams/{team\_name}  
# Получить основную инофрмацию о клубе  
# 3)***SELECT** team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 team.**avatar AS** team\_avatar,  
 city.**name AS** city\_name,  
 **foundation\_date  
FROM** team  
 **LEFT JOIN** team\_in\_city **ON** team.**id** = team\_in\_city.**team\_id  
 LEFT JOIN** city **ON** team\_in\_city.**city\_id** = city.**id**;  
  
*# 4) Получить результат последних 5 игр***SELECT DISTINCT** game.**id AS** game\_id,  
 home\_team.**id AS** home\_team\_id,  
 guest\_team.**name AS** guest\_team\_name,  
 guest\_team.**id AS** guest\_team\_id,  
 home\_team.**name AS** home\_team\_name,  
 **home\_team\_goal\_count**,  
 **guest\_team\_goal\_count**,  
 guest\_team\_city.**name AS** guest\_team\_city\_name,  
 home\_team\_city.**name AS** home\_team\_city\_name,  
 game.**start**,  
 game.**end  
FROM** game  
 **INNER JOIN** (  
 **SELECT  
 id**,  
 **home\_team\_id**,  
 **guest\_team\_id  
 FROM** game  
 **WHERE** (**end IS NOT NULL**) **AND** (**home\_team\_id** = 1 **OR guest\_team\_id** = 1)  
 **ORDER BY start DESC  
 LIMIT** 5  
 ) **AS** last\_game\_id **USING** (**id**)  
  
 **LEFT JOIN** team **AS** home\_team **ON** last\_game\_id.**home\_team\_id** = home\_team.**id  
 LEFT JOIN** team\_in\_city **AS** home\_team\_in\_city **ON** home\_team.**id** = home\_team\_in\_city.**team\_id  
 LEFT JOIN** city **AS** home\_team\_city **ON** home\_team\_in\_city.**city\_id** = home\_team\_city.**id  
  
 LEFT JOIN** team **AS** guest\_team **ON** last\_game\_id.**guest\_team\_id** = guest\_team.**id  
 LEFT JOIN** team\_in\_city **AS** guest\_team\_in\_city **ON** guest\_team.**id** = guest\_team\_in\_city.**team\_id  
 LEFT JOIN** city **AS** guest\_team\_city **ON** guest\_team\_in\_city.**city\_id** = guest\_team\_city.**id**;  
  
*# /teams/{team\_name}/players  
# 5)Получить состав на текущий сезон (2018/2019) сезон***SELECT  
 player\_id**,  
 player.**first\_name**,  
 player.**last\_name**,  
 player\_position.**full\_name**,  
 player.**born\_date  
FROM** team  
 **LEFT JOIN** team\_roster **ON** team.**id** = team\_roster.**team\_id  
 LEFT JOIN** season **ON** team\_roster.**season\_id** = season.**id  
 LEFT JOIN** player **ON** team\_roster.**player\_id** = player.**id  
 LEFT JOIN** player\_position **ON** player.**player\_position\_id** = player\_position.**id  
WHERE** (season.**start** >= **'2018-10-02' AND** season.**end** <= **'2019-05-01'**) **AND team\_id** = 1;  
  
*# /players/  
# Получить информацию о 30 игроках игроках всех команд в алвафитном порядке на текущий момент (2018/2019)  
# 6)***SELECT** player.**id AS** player\_id,  
 player.**last\_name**,  
 player.**first\_name**,  
 player\_position.**full\_name**,  
 player.**born\_date  
FROM** player  
 **INNER JOIN** (  
 **SELECT** *\** **FROM** player  
 **ORDER BY** player.**last\_name  
 LIMIT** 1, 30  
 ) **AS** players **USING** (**id**)  
 **LEFT JOIN** team\_roster **ON** players.**id** = team\_roster.**player\_id  
 LEFT JOIN** season **ON** team\_roster.**season\_id** = season.**id  
 LEFT JOIN** player\_position **ON** player.**player\_position\_id** = player\_position.**id  
WHERE** (season.**start** >= **'2018-10-02' AND** season.**end** <= **'2019-05-01'**);  
  
*# /players/{player\_id}  
# 7) Получить общую информацию об игроке***SELECT** player.**id AS** player\_id,  
 player.**last\_name**,  
 player.**first\_name**,  
 player.**born\_date**,  
 *calcAge*(**born\_date**) **AS** age,  
 *GROUP\_CONCAT*(**DISTINCT** team.**name SEPARATOR ', '**) **AS** player\_teams,  
 city.**name AS** born\_city,  
 player.**photo**,  
 player\_position.**short\_name  
FROM** player  
 **LEFT JOIN** city **ON** player.**birth\_place\_id** = city.**id  
 LEFT JOIN** team\_roster **ON** player.**id** = team\_roster.**player\_id  
 LEFT JOIN** team **ON** team\_roster.**team\_id** = team.**id  
 LEFT JOIN** player\_position **ON** player.**player\_position\_id** = player\_position.**id  
WHERE** player.**id** = 7;  
  
*# 8) Получить стаистику команд (количество побед, поражений, забитых и пропущенных голов и т.д.) , за которые выступал игрок по сезонам***SELECT** player.**id AS** player\_id,  
 season.**id AS** season\_id,  
 season.**start**,  
 season.**end**,  
 team\_name,  
 game\_count,  
 goals\_scored,  
 goals\_scored,  
 difference,  
 count\_win,  
 count\_lose,  
 count\_overtime\_win,  
 count\_overtime\_lose,  
 points  
**FROM** player  
 **LEFT JOIN** team\_roster **ON** player.**id** = team\_roster.**player\_id  
 lEFT JOIN** season **ON** team\_roster.**season\_id** = season.**id  
 LEFT JOIN** (**SELECT  
 season\_id**,  
 team\_id,  
 team\_name,  
 *SUM*(game\_count) **AS** game\_count,  
 *SUM*(goals\_scored) **AS** goals\_scored,  
 *SUM*(goals\_missed) **AS** goals\_missed,  
 *SUM*(goals\_scored) - *SUM*(goals\_missed) **AS** difference,  
 *SUM*(is\_win) **AS** count\_win,  
 *SUM*(is\_lose) **AS** count\_lose,  
 *SUM*(is\_overtime\_win) **AS** count\_overtime\_win,  
 *SUM*(is\_overtime\_lose) **AS** count\_overtime\_lose,  
 *SUM*(**point**) **AS** points  
 **FROM** ((**SELECT** game.**season\_id**,  
 game.**id AS** game\_id,  
 *COUNT*(  
 game.**id**) **AS** guest\_game\_count,  
 team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 *COUNT*(  
 game.**id**) **AS** game\_count,  
 *SUM*(  
 game.**home\_team\_goal\_count**) **AS** goals\_scored,  
 *SUM*(  
 game.**guest\_team\_goal\_count**) **AS** goals\_missed,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_lose,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_lose,  
 *SUM*(*getGamePoint*(**guest\_team\_goal\_count**, **home\_team\_goal\_count**, **time**)) **AS point  
 FROM** game  
 **LEFT JOIN** team **ON** game.**guest\_team\_id** = team.**id  
 LEFT JOIN** season **ON** game.**season\_id** = season.**id  
 WHERE** game.**end IS NOT NULL  
 GROUP BY season\_id**)  
 **UNION ALL** (**SELECT** game.**season\_id**,  
 game.**id AS** game\_id,  
 *COUNT*(  
 game.**id**) **AS** home\_game\_count,  
 team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 *COUNT*(  
 game.**id**) **AS** game\_count,  
 *SUM*(  
 game.**home\_team\_goal\_count**) **AS** goals\_scored,  
 *SUM*(  
 game.**guest\_team\_goal\_count**) **AS** goals\_missed,  
 *SUM*(game.**home\_team\_goal\_count** > game.**guest\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_win,  
 *SUM*(game.**home\_team\_goal\_count** < game.**guest\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_lose,  
 *SUM*(game.**home\_team\_goal\_count** > game.**guest\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_win,  
 *SUM*(game.**home\_team\_goal\_count** < game.**guest\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_lose,  
 *SUM*(*getGamePoint*(**home\_team\_goal\_count**, **guest\_team\_goal\_count**, **time**)) **AS point  
 FROM** game  
 **LEFT JOIN** team **ON** game.**home\_team\_id** = team.**id  
 LEFT JOIN** season **ON** game.**season\_id** = season.**id  
 WHERE** game.**end IS NOT NULL  
 GROUP BY season\_id**)) all\_game  
 **GROUP BY season\_id  
 ORDER BY** points **DESC**) **AS** stats **ON** stats.**season\_id** = season.**id  
WHERE** player.**id** = 1  
**GROUP BY** season.**id**;  
  
*# 9) Создать игрока***INSERT** player (**first\_name**, **last\_name**, **born\_date**, **birth\_place\_id**, **player\_position\_id**, **is\_right\_shoot**) **VALUE** (**'Александр'**, **'Овечкин'**, **'1985-09-17'**, 1, 1, 1);  
  
*# 10 )Удалить игрока***DELETE FROM** player  
**WHERE id** = 1;  
  
*# 11) Обновить фото игрока***UPDATE** player  
**SET** player.**photo** = **'/img/some-photo.jpg'**;  
  
*# 12) Поиск по фамилии или имени игрока***SELECT** player.**id AS** player\_id,  
 player.**last\_name**,  
 player.**first\_name**,  
 player.**born\_date**,  
 *calcAge*(**born\_date**) **AS** age,  
 *GROUP\_CONCAT*(**DISTINCT** team.**name SEPARATOR ', '**) **AS** player\_teams,  
 city.**name AS** born\_city,  
 player.**photo**,  
 player\_position.**short\_name  
FROM** player  
 **LEFT JOIN** city **ON** player.**birth\_place\_id** = city.**id  
 LEFT JOIN** team\_roster **ON** player.**id** = team\_roster.**player\_id  
 LEFT JOIN** team **ON** team\_roster.**team\_id** = team.**id  
 LEFT JOIN** player\_position **ON** player.**player\_position\_id** = player\_position.**id  
WHERE first\_name LIKE '%Ник%' OR last\_name LIKE '%Ник%'  
GROUP BY player\_id**;  
  
*# 13) Добавить игрока в команду***INSERT INTO** team\_roster (**season\_id**, **team\_id**, **player\_id**) **VALUES** (1, 2, 1);

*# /calendar*  
*# 13) Получить игры за период [текущая дата - 2 недели; текущая дата + 2 недели] текущего сезона;***SELECT** game.**id AS** game\_id,  
 home\_team.**id AS** home\_team\_id,  
 home\_team.**name AS** home\_team\_name,  
 guest\_team\_city.**name**,  
 guest\_team.**id AS** guest\_team\_id,  
 guest\_team.**name AS** guest\_team\_id,  
 guest\_team\_city.**name**,  
 game.**start AS** game\_start  
**FROM** game  
 **LEFT JOIN** team **AS** home\_team **ON** game.**home\_team\_id** = home\_team.**id  
 LEFT JOIN** team **AS** guest\_team **ON** game.**guest\_team\_id** = guest\_team.**id  
 LEFT JOIN** team **AS** home\_team\_city **ON** game.**home\_team\_id** = home\_team\_city.**id  
 LEFT JOIN** team **AS** guest\_team\_city **ON** game.**guest\_team\_id** = guest\_team\_city.**id  
WHERE** game.**season\_id** = *getSeasionId*(*CURDATE*()) **AND** (game.**start** >= *DATE\_SUB*(*CURDATE*(), **INTERVAL** 14 **DAY**) **AND** game.**start** <= *DATE\_ADD*(*CURDATE*(), **INTERVAL** 14 **DAY**));  
  
*# 14) Получить игры за период [текущая дата - 2 недели; текущая дата + 2 недели] текущего сезона одной команды;***SELECT** game.**id AS** game\_id,  
 home\_team.**id AS** home\_team\_id,  
 home\_team.**name AS** home\_team\_name,  
 guest\_team\_city.**name**,  
 guest\_team.**id AS** guest\_team\_id,  
 guest\_team.**name AS** guest\_team\_id,  
 guest\_team\_city.**name**,  
 game.**start AS** game\_start  
**FROM** game  
 **LEFT JOIN** team **AS** home\_team **ON** game.**home\_team\_id** = home\_team.**id  
 LEFT JOIN** team **AS** guest\_team **ON** game.**guest\_team\_id** = guest\_team.**id  
 LEFT JOIN** team **AS** home\_team\_city **ON** game.**home\_team\_id** = home\_team\_city.**id  
 LEFT JOIN** team **AS** guest\_team\_city **ON** game.**guest\_team\_id** = guest\_team\_city.**id  
WHERE** (home\_team.**id** = 1 **OR** guest\_team.**id** = 1) **AND** game.**season\_id** = *getSeasionId*(*CURDATE*())  
 **AND** (game.**start** >= *DATE\_SUB*(*CURDATE*(), **INTERVAL** 14 **DAY**) **AND** game.**start** <= *DATE\_ADD*(*CURDATE*(), **INTERVAL** 14 **DAY**));  
  
*# 15) Добавить игру на сезон***INSERT INTO** game (**season\_id**, **guest\_team\_id**, **home\_team\_id**, **time**, **guest\_team\_goal\_count**, **home\_team\_goal\_count**, **start**, **end**) **VALUES** (1, 2, 1, **null**, 0, 0, **'2018-10-09 19:00'**, **null**);  
  
*# 16) Изменить счет в игре***UPDATE** game **SET home\_team\_goal\_count WHERE** game.**id** = 1  
  
*# 17) /calendar/{game\_id}/preview  
# Получить результат последнего матча***SELECT** game.**id AS** game\_id,  
 game.**home\_team\_id**,  
 game.**guest\_team\_id**,  
 game.**home\_team\_goal\_count**,  
 home\_team.**id AS** home\_team\_id,  
 home\_team.**name AS** home\_team\_name,  
 guest\_team\_city.**name**,  
 guest\_team.**id AS** guest\_team\_id,  
 guest\_team.**name AS** guest\_team\_id,  
 guest\_team\_city.**name**,  
 game.**start AS** game\_start  
**FROM** game  
 **LEFT JOIN** team **AS** home\_team **ON** game.**home\_team\_id** = home\_team.**id  
 LEFT JOIN** team **AS** guest\_team **ON** game.**guest\_team\_id** = guest\_team.**id  
 LEFT JOIN** team **AS** home\_team\_city **ON** game.**home\_team\_id** = home\_team\_city.**id  
 LEFT JOIN** team **AS** guest\_team\_city **ON** game.**guest\_team\_id** = guest\_team\_city.**id**;  
  
*# 18) /calendar/{game\_id}/preview  
# Получить результаты домашние игры команды А с командой Б***SELECT** game.**season\_id**,  
 game.**id AS** game\_id,  
 *COUNT*(  
 game.**id**) **AS** guest\_game\_count,  
 team.**id AS** team\_id,  
 team.**name AS** team\_name,  
 *COUNT*(  
 game.**id**) **AS** game\_count,  
 *SUM*(  
 game.**home\_team\_goal\_count**) **AS** goals\_scored,  
 *SUM*(  
 game.**guest\_team\_goal\_count**) **AS** goals\_missed,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** = **'01:00:00'**) **AS** is\_lose,  
 *SUM*(game.**guest\_team\_goal\_count** > game.**home\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_win,  
 *SUM*(game.**guest\_team\_goal\_count** < game.**home\_team\_goal\_count AND** game.**time** >  
 **'01:00:00'**) **AS** is\_overtime\_lose,  
 *SUM*(*getGamePoint*(**guest\_team\_goal\_count**, **home\_team\_goal\_count**, **time**)) **AS point  
FROM** game  
 **LEFT JOIN** team **ON** game.**home\_team\_id** = team.**id  
 LEFT JOIN** season **ON** game.**season\_id** = season.**id  
WHERE** game.**end IS NOT NULL  
GROUP BY** game\_id;

Тексты необходимых процедур, функций и триггеров.

Функция вычисления количества очко полученной командой за игру

**DROP FUNCTION IF EXISTS** *getGamePoint*;  
**DELIMITER** //  
**CREATE FUNCTION** *getGamePoint*(guest\_goal\_count **INT**(11), home\_goal\_count **INT**(11), **time TIME**)  
 **RETURNS INT**(11) **DETERMINISTIC  
 BEGIN  
 DECLARE point INT**(11);  
 **IF** (guest\_goal\_count > home\_goal\_count)  
 **THEN  
 SET point** = 2;  
 **ELSEIF** (guest\_goal\_count <= home\_goal\_count **AND time** > **'01:00:00'**)  
 **THEN  
 SET point** = 1;  
 **ELSE  
 SET point** = 0;  
 **END IF**;  
 **RETURN** (**point**);  
 **END** //  
**DELIMITER** ;

Функция вычисления возраста игрока**DROP FUNCTION IF EXISTS** *calcAge*;  
**DELIMITER** //  
**CREATE FUNCTION** *calcAge*(bornDate **DATE**)  
 **RETURNS INT**(11) **DETERMINISTIC  
 BEGIN  
 RETURN** (*floor*(*datediff*(*curdate*(), bornDate) / 365));  
 **END** //  
**DELIMITER** ;**DROP FUNCTION IF EXISTS** *getSeasionId*;

Функция вычисления идентификатора сезона по дате  
**DELIMITER** //  
**CREATE FUNCTION** *getSeasionId*(**date DATE**)  
 **RETURNS INT**(11) **DETERMINISTIC  
 BEGIN  
 RETURN** (**SELECT id  
 FROM** season  
 **WHERE** season.**start** <= **date AND** season.**end** >= **date**);  
 **END** //  
**DELIMITER** ;

**Описание и вид созданных приложения и отчетов**

Страницы:

1. /schedules
2. /schedules/{schedule\_id}
3. /standings
4. /teams
5. /teams/{team\_id}
6. /teams/{team\_id}/roster
7. /players
8. /players/{player\_id}