

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра информатики  
Дисциплина «Модели данных и системы управления базами данных»

**ОТЧЕТ**  
к лабораторной работе №6  
на тему:  
**«ПОСТРОЕНИЕ ПРОСТЫХ SQL ЗАПРОСОВ»**  
БГУИР 6-05-0612-02 67

Выполнил студент группы 353503  
КОХАН Артём Игоревич

---

(дата, подпись студента)

Проверил начальник отдела ОИАСУ  
КОЖЕМЯКО Евгения Александровна

---

(дата, подпись преподавателя)

Минск 2025

## **1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

Написать запросы позволяющие получить следующую информацию из базы данных на основе вашей предметной области:

- 1 Определить количество завершенных матчей.
- 2 Вывести 3 самых молодых клуба.
- 3 Получить количество клубов и спонсоров, по которым заключены контракты.
- 4 Вывести список стадионов, отсортированный по количеству запланированных матчей.
- 5 Вывести клубы, у которых нет в базе данных ни одного игрока.
- 6 Вывести клубы, у которых средний возраст игроков максимален. Клубы, где в базе данных ни одного игрока не учитывать. Округлить до двух знаков после запятой.

## **2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ**

SQL состоит из нескольких подмножеств, наиболее важными из которых являются DDL (Data Definition Language) и DML (Data Manipulation Language). DDL используется для определения структуры базы данных – создания, изменения и удаления таблиц и других объектов. Основные команды DDL: CREATE, ALTER, DROP. DML используется для манипуляции с данными – выборки, вставки, обновления и удаления записей. Основные команды DML: SELECT, INSERT, UPDATE, DELETE. В данной лабораторной работе основное внимание уделяется написанию запросов SELECT для извлечения информации из базы данных. Запрос SELECT позволяет выбирать данные из одной или нескольких таблиц с использованием различных условий фильтрации (WHERE), сортировки (ORDER BY) и объединения таблиц (JOIN). Для соединения таблиц используются различные типы JOIN: INNER JOIN (возвращает строки, имеющие соответствия в обеих таблицах), LEFT JOIN (возвращает все строки из левой таблицы и соответствующие строки из правой). Для работы с шаблонами строк применяется оператор LIKE с wildcard-символами (% – любое количество любых символов, \_ – один любой символ). Агрегатные функции (COUNT, SUM, AVG, MAX, MIN) позволяют выполнять вычисления над группами строк. Группировка данных осуществляется с помощью оператора GROUP BY. Вложенные запросы (подзапросы) используются для выполнения сложных операций, когда результат одного запроса используется в другом запросе. Правильное проектирование структуры базы данных и владение языком SQL являются фундаментальными навыками для работы с любыми системами управления базами данных.

### 3 ВЫПОЛНЕНИЕ РАБОТЫ

Ниже представлен код SQL-запросов, обеспечивающих получение аналитической информации из базы данных футбольной лиги. Для выполнения поставленных задач используются различные конструкции языка SQL, включая агрегатные функции (COUNT, SUM, AVG, MAX, MIN).

```
-- 1.1 Определить количество завершенных матчей

SELECT COUNT(*) AS completed_matches_count
FROM match
WHERE status = 'completed';

-- 1.2 Вывести 3 самых молодых клуба

SELECT name, founded_date
FROM club
ORDER BY founded_date DESC
LIMIT 3;

-- 1.3 Получить количество клубов и спонсоров, по которым заключены контракты

SELECT
    COUNT(DISTINCT club_id) AS active_clubs_count,
    COUNT(DISTINCT sponsor_id) AS active_sponsors_count
FROM club_sponsorship;

-- 1.4 Вывести список стадионов, отсортированный по количеству запланированных матчей

SELECT
    s.name,
    s.city,
    COUNT(m.id) AS matches_hosted
FROM stadium s
LEFT JOIN match m ON s.id = m.stadium_id
WHERE m.status = 'expected'
GROUP BY s.id
ORDER BY matches_hosted DESC;

-- 1.5 Вывести клубы, у которых нет в базе данных ни одного игрока

SELECT
    c.name,
    0 AS count_players
FROM club c
LEFT JOIN player p ON c.id = p.club_id
WHERE p.id IS NULL;

-- 1.6 Вывести клубы, у которых средний возраст игроков максимальен.
-- Клубы, где в базе данных ни одного игрока не учитывать
-- Округлить до двух знаков после запятой

SELECT
    c.name,
    ROUND(
        AVG(
            (strftime('%Y', 'now') - strftime('%Y', p.birth_date)) -
            (strftime('%m-%d', 'now') < strftime('%m-%d', p.birth_date))
        ),
        2) AS avg_age
```

```
FROM club c
JOIN player p ON c.id = p.club_id
GROUP BY c.id
HAVING COUNT(p.id) > 0
ORDER BY avg_age DESC;
```

## **ВЫВОД**

В рамках выполнения лабораторной работы был разработан комплекс SQL-запросов, направленных на анализ данных футбольной лиги. Представленные запросы демонстрируют практическое применение различных конструкций языка SQL для решения аналитических задач. В работе использованы агрегатные функции COUNT и AVG для подсчета и вычисления средних значений, оператор JOIN для объединения данных из нескольких таблиц, фильтрация с помощью WHERE для отбора записей по определенным условиям, а также группировка GROUP BY для агрегации данных на уровне клубов и стадионов. Особое внимание уделено обработке исключительных ситуаций: с помощью LEFT JOIN и проверки на IS NULL выявляются клубы без игроков, а условие HAVING гарантирует, что в расчете среднего возраста учитываются только клубы с существующими игроками. Для улучшения читаемости результатов применено округление числовых значений до двух знаков после запятой и сортировка выводов по релевантным критериям. Все запросы оптимизированы для работы с реальными данными и обеспечивают получение точной статистической информации о футбольных клубах, матчах и связанных сущностях.