

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 №5
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 Вывод всех данных обо всех футбольных клубах.
- 2 Вывод имён и фамилий игроков футбольного клуба "Славия-Мозырь", отсортированных по фамилии в алфавитном порядке.
- 3 Вывод данных об играющих, фамилии которых начинаются на букву "М".
- 4 Вывод всех матчей клубов, которые основаны в апреле.
- 5 Вывод всех игроков моложе 25 лет, не играющих за футбольный клуб "Славия-Мозырь".
- 6 Вывод информации о матче с указанием названий клубов, стадиона и даты проведения.
- 7 Вывод информации о матчах, которые проводятся не на стадионе "Юность".
- 8 Вывод отсортированного списка спонсоров и тренеров клуба.
- 9 Вывод отсортированного списка всех тренеров и названий их клубов с указанием количества игроков в клубе.
- 10 Вывод названия стадионов, на которых еще не проводились матчи.

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, обеспечивающий вставку минимально необходимого набора данных во все таблицы базы данных. Для обеспечения ссылочной целостности и корректности связей между сущностями активно применяются подзапросы с оператором SELECT, позволяющие динамически определять идентификаторы связанных записей в родительских таблицах на основе их атрибутов. Конструкция UNION ALL используется для эффективного объединения множественных наборов данных при заполнении таблиц, содержащих внешние ключи.

```
-- 1.1 Вывод всех данных обо всех футбольных клубах
SELECT *
FROM club;

-- 1.2 Вывод имён и фамилий игроков футбольного клуба "Славия-Мозырь",
отсортированных по фамилии в алфавитном порядке
SELECT first_name, last_name
FROM player
WHERE club_id = (SELECT id FROM club WHERE name = 'Славия-Мозырь')
ORDER BY last_name;

-- 1.3 Вывод данных об играющих, фамилии которых начинаются на букву "М"
SELECT *
FROM player
WHERE last_name LIKE 'M%';

-- 2.1 Вывод всех матчей клубов, которые основаны в апреле
SELECT
    hc.name AS home_club,
    hc.founded_date AS home_founded_date,
    ac.name AS away_club,
    ac.founded_date AS away_founded_date,
    s.name AS stadium,
    m.match_date,
    m.status,
    m.referee
FROM match m
JOIN club hc ON m.home_club_id = hc.id
JOIN club ac ON m.away_club_id = ac.id
JOIN stadium s ON m.stadium_id = s.id
WHERE (hc.founded_date LIKE '____-04-__' OR ac.founded_date LIKE '____-04-__');

-- 2.2 Вывод всех игроков моложе 25 лет, не играющих за футбольный клуб "Славия-
Мозырь"
SELECT
    p.first_name,
    p.last_name,
    p.position,
    p.number,
    p.birth_date,
    c.name AS club_name,
    (strftime('%Y', 'now') - strftime('%Y', p.birth_date)) -
    (strftime('%m-%d', 'now') < strftime('%m-%d', p.birth_date)) AS age
```

```

FROM player p
JOIN club c ON p.club_id = c.id
WHERE (strftime('%Y', 'now') - strftime('%Y', p.birth_date)) -
(strftime('%m-%d', 'now') < strftime('%m-%d', p.birth_date)) < 25
AND c.name != 'Славия-Мозырь';

-- 2.3 Вывод информации о матче с указанием названий клубов, стадиона и даты
проведения

SELECT
    hc.name AS home_club,
    ac.name AS away_club,
    s.name AS stadium,
    m.match_date,
    m.status,
    m.referee,
    m.home_score,
    m.away_score
FROM match m
JOIN club hc ON m.home_club_id = hc.id
JOIN club ac ON m.away_club_id = ac.id
JOIN stadium s ON m.stadium_id = s.id;

-- 2.4 Вывод информации о матчах, которые проводятся не на стадионе "Юность"

SELECT
    hc.name AS home_club,
    ac.name AS away_club,
    s.name AS stadium,
    m.match_date,
    m.status,
    m.referee,
    m.home_score,
    m.away_score
FROM match m
JOIN club hc ON m.home_club_id = hc.id
JOIN club ac ON m.away_club_id = ac.id
JOIN stadium s ON m.stadium_id = s.id
WHERE s.name != 'Юность';

-- 2.5 Вывод отсортированного списка спонсоров и тренеров клуба

SELECT
    s.name,
    c.coach,
    c.name
FROM club_sponsorship cs
JOIN club c ON cs.club_id = c.id
JOIN sponsor s ON cs.sponsor_id = s.id
ORDER BY s.name;

-- 2.6 Вывод отсортированного списка всех тренеров и названий их клубов с
указанием количества игроков в клубе

SELECT
    c.coach,
    c.name,
    count(p.id) AS count_players
FROM club c
LEFT JOIN player p ON p.club_id = c.id
WHERE c.coach IS NOT NULL
GROUP BY c.coach, c.name
ORDER BY c.coach;

```

-- 2.7 Вывод названия стадионов, на которых еще не проводились матчи

```
SELECT
    s.name AS stadium_name,
    s.city,
    s.capacity
FROM stadium s
LEFT JOIN match m ON s.id = m.stadium_id
WHERE m.id IS NULL
```

ВЫВОД

В ходе выполнения лабораторной работы был разработан комплекс SQL-запросов для извлечения различной информации из базы данных футбольных матчей. Запросы демонстрируют практическое применение ключевых конструкций языка SQL для решения реальных задач анализа данных.

Разработанные запросы охватывают основные сценарии работы с реляционной базой данных: выборку данных из одной таблицы, соединение нескольких таблиц, агрегацию данных, работу с датами и фильтрацию по сложным условиям.