

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 №3
на тему:
«ПРОЕКТИРОВАНИЕ СХЕМЫ БАЗЫ ДАННЫХ В SQL»
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

Получить практические навыки построения баз данных с помощью диалекта DDL языка SQL. В результате выполнения работы вы должны иметь набор SQL-запросов позволяющий создать базу данных согласно предложенной схеме и вашей предметной области.

2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

Схема базы данных представляет собой её структуру, включая таблицы, столбцы, типы данных, ограничения и связи между таблицами. Основным инструментом для создания структуры базы данных в SQL является DDL (Data Definition Language) CREATE TABLE. Оно позволяет определить таблицу, указав её имя, перечислив столбцы с их типами данных и наложив ограничения (CONSTRAINTS).

В SQLite используются следующие типы данных: строковые (TEXT – строка переменной длины), числовые (INTEGER для целых чисел, REAL для чисел с плавающей точкой, NUMERIC для десятичных чисел), бинарные данные (BLOB). Связи между таблицами, обеспечивающие целостность данных, устанавливаются с помощью ограничений FOREIGN KEY, которые ссылаются на первичные или уникальные ключи в других таблицах. В результате проектирования создается набор SQL-запросов, которые при выполнении формируют структуру базы данных, готовую для хранения и обработки информации. Правильно спроектированная схема обеспечивает эффективность, надежность и согласованность данных.

3 ВЫПОЛНЕНИЕ РАБОТЫ

Представленный ниже код на языке SQL представляет собой законченный скрипт для создания реляционной базы данных, предназначенный для управления данными футбольной лиги или клуба. Структура базы данных включает в себя шесть взаимосвязанных таблиц, которые охватывают основные сущности предметной области. Каждая таблица спроектирована с учетом целостности данных: определены первичные и внешние ключи для установления связей, а также добавлены различные ограничения (CHECK, NOT NULL, UNIQUE), которые гарантируют корректность и непротиворечивость хранимой информации. Для автоматической генерации уникальных идентификаторов записей в таблицах используется ключевое слово AUTOINCREMENT. Политики обработки зависимых записей при удалении (ON DELETE CASCADE и ON DELETE SET NULL) обеспечивают поддержание целостности связей в базе данных при выполнении операций удаления.

```
CREATE TABLE club
(
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT UNIQUE NOT NULL,
    founded_date DATE NOT NULL,
    coach TEXT
);

CREATE TABLE sponsor(
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT UNIQUE NOT NULL
);

CREATE TABLE club_sponsorship (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    club_id INTEGER NOT NULL,
    sponsor_id INTEGER NOT NULL,
    start_date DATE NOT NULL,
    FOREIGN KEY (club_id) REFERENCES CLUB(id) ON DELETE CASCADE,
    FOREIGN KEY (sponsor_id) REFERENCES SPONSOR(id) ON DELETE CASCADE
);

CREATE TABLE stadium (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL UNIQUE,
    city TEXT,
    capacity INTEGER CHECK (capacity > 0)
);

CREATE TABLE player (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    club_id INTEGER NOT NULL,
```

```

first_name TEXT NOT NULL,
last_name TEXT NOT NULL,
position TEXT NOT NULL ,
number INTEGER CHECK (number BETWEEN 1 AND 99),
birth_date DATE NOT NULL,

FOREIGN KEY (club_id) REFERENCES CLUB(id) ON DELETE SET NULL
);

CREATE TABLE match (
id INTEGER PRIMARY KEY AUTOINCREMENT,

home_club_id INTEGER NOT NULL,
away_club_id INTEGER NOT NULL,
stadium_id INTEGER NOT NULL,

match_date DATETIME NOT NULL,
status TEXT DEFAULT 'expected' CHECK (status IN ('expected', 'playing',
'completed')),
home_score INTEGER DEFAULT 0 CHECK (home_score >= 0),
away_score INTEGER DEFAULT 0 CHECK (away_score >= 0),
referee TEXT,

FOREIGN KEY (home_club_id) REFERENCES CLUB(id) ON DELETE CASCADE,
FOREIGN KEY (away_club_id) REFERENCES CLUB(id) ON DELETE CASCADE,
FOREIGN KEY (stadium_id) REFERENCES STADIUM(id) ON DELETE CASCADE,
CHECK (home_club_id != away_club_id)
);

```

ВЫВОД

В ходе выполнения лабораторной работы были успешно получены и закреплены практические навыки проектирования и реализации реляционной базы данных с использованием языка структурированных запросов SQL, а именно его подмножества DDL (Data Definition Language). Была разработана схема базы данных для управления информацией о футбольных клубах, соответствующая принципам нормализации и целостности данных.

В процессе работы создан набор взаимосвязанных таблиц, отражающих предметную область. Для каждой таблицы продумана и задана структура: определены атрибуты с соответствующими типами данных, назначены первичные ключи для уникальной идентификации записей и установлены внешние ключи для обеспечения связей между таблицами.

Проведенная работа продемонстрировала полный цикл проектирования схемы БД: от концептуальной модели до ее физической реализации на языке SQL.