

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 №1
на тему:
«ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ»
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

Приобретение навыков анализа предметной области и построения концептуальной модели. Предметная область: Футбольные матчи.

1 Выделить основные абстракции (сущность, атрибут, связь) в предметной области и определить их параметры.

2 Сформировать максимально полный перечень возможных запросов к базе данных на основе анализа предметной области.

3 Построить концептуальную модель в виде ER-диаграммы.

4 Представить концептуальную модель в терминах реляционной модели.

5 Описать домены (допустимые множества значений, которые могут принимать атрибуты), указывая типы соответствующих данных и их характеристики.

6 Определить ключи и внешние ключи (если они есть).

7 Выписать функциональные зависимости (рассматривая возможные значения полей таблицы).

8 Привести полученную концептуальную модель к третьей нормальной форме (показать, что она находится в соответствующей нормальной форме).

2 КРАТКИЕ ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ

Концептуальная модель базы данных представляет собой абстрактное описание предметной области, которое включает в себя основные сущности, их атрибуты и связи между ними. Основными абстракциями при построении такой модели являются сущности (объекты предметной области), атрибуты (свойства сущностей) и связи (отношения между сущностями). Ключевым этапом является анализ текстового описания предметной области для выделения этих элементов. На основе анализа формируются возможные запросы к базе данных, которые определяют требования к системе. Концептуальная модель часто визуализируется в виде ER-диаграммы, которая наглядно отображает сущности, атрибуты и связи. Далее модель представляется в терминах реляционной модели в виде таблиц, где каждый атрибут имеет свой домен – допустимое множество значений с указанием типа данных и характеристик. Для каждой таблицы определяются ключи (первичные и внешние), которые обеспечивают целостность данных и связи между таблицами. Важным этапом является анализ функциональных зависимостей между атрибутами, который позволяет провести нормализацию отношений.

Нормализация – это процесс организации данных в базе данных. Она включает в себя создание таблиц и установление связей между ними в соответствии с правилами, разработанными как для защиты данных, так и для повышения гибкости базы данных, устранив избыточность и несогласованную зависимость.

Переменная отношения находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

Отношение находится во второй нормальной форме (2НФ) тогда и только тогда, когда отношение находится в 1НФ и нет неключевых атрибутов, зависящих от части сложного ключа.

Отношение находится в третьей нормальной форме (3НФ) тогда и только тогда, когда отношение находится в 2НФ и все неключевые атрибуты взаимно независимы.

3 ВЫПОЛНЕНИЕ РАБОТЫ

1 Выделим сущности на основе нашей предметной области и определим для них атрибуты:

- Club (Клуб): id (PK), name, founded_date, coach;
- Player (Игрок): id (PK), club_id (FK), first_name, last_name, position, number, birth_date;
- Stadium (Стадион): id (PK), name, city, capacity;
- Match (Матч): id (PK), home_club_id (FK), away_club_id (FK), stadium_id (FK), match_date, status, home_score, away_score, referee;
- Sponsor (Спонсор): id (PK), name;
- ClubSponsorship (Спонсорство): id (PK), club_id (FK), sponsor_id (FK), start_date.

Определим связи между таблицами баз данных:

- Club содержит Player (Один-ко-многим): Один клуб может иметь много игроков, но один игрок принадлежит только одному клубу;
- Stadium имеет Match (Один-ко-многим): На одном стадионе может проводиться много матчей, но один матч проводится на одном стадионе;
- Club быть домашней/гостевой командой на Match (Один-ко-многим): Один клуб может быть домашней командой во многих матчах, но у одного матча только один домашний клуб;
- Club and Sponsor (Многие-ко-многим): Один клуб может иметь много спонсоров, один спонсор может спонсировать много клубов.

2 Сформируем максимально полный перечень возможных запросов к базе данных на основе анализа:

- Получить список всех клубов;
- Найти клуб по названию (или части названия);
- Найти все клубы, основанные после определенного года;
- Найти клуб, который основан раньше всех;
- Показать клубы с указанием количества игроков в каждом;
- Найти клубы, которые в настоящее время не имеют тренера;
- Получить всех игроков указанного клуба;
- Получить всех игроков указанного клуба;
- Найти игроков по фамилии (или части фамилии);
- Показать игроков определенной позиции;
- Найти игроков моложе определенного возраста;
- Показать игроков с номерами от X до Y;
- Показать самых молодых/старших игроков;
- Получить список всех стадионов с их вместимостью;
- Найти стадионы в определенном городе;
- Показать самый вместительный стадион;
- Найти стадионы с вместимостью больше указанной;
- Получить расписание предстоящих матчей;
- Найти все матчи определенного клуба;
- Показать матчи на определенном стадионе;

- Найти матчи, которые судил определенный арбитр;
- Получить список всех спонсоров;
- Найти спонсоров по названию;
- Показать спонсоров определенного клуба;
- Вывести клубы, которые спонсирует указанная компания;
- Показать среднюю вместимость стадионов в каждом городе;
- Вывести средний возраст игроков.

3 Построим концептуальную модель ER-диаграммы с помощью PlantUML, полученный результат можно увидеть на рисунке 1.

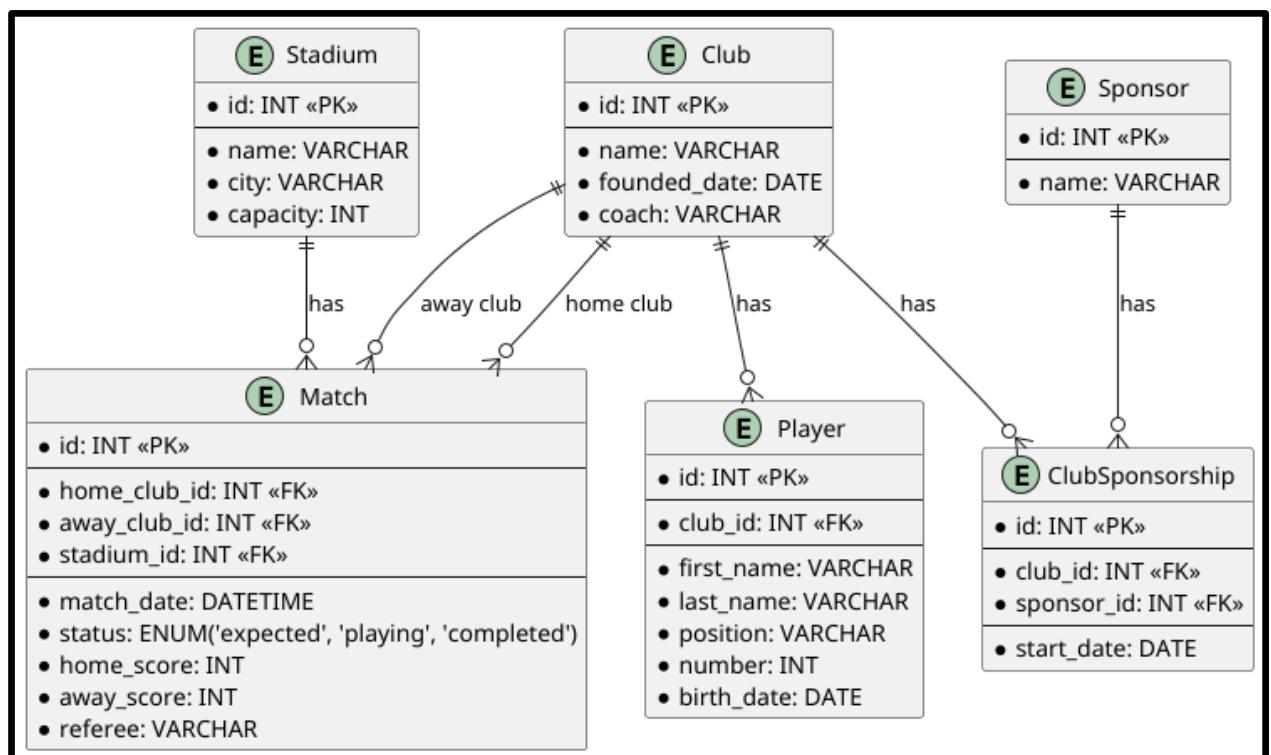


Рисунок 1 – Концептуальная модель в виде ER-диаграммы

4 На основании концептуальной модели можно выписать следующие функциональные зависимости:

- Club: id определяет name, founded_date, coach. Поле name определяет id, founded_date, coach.
- Sponsor: id определяет name. Поле name определяет id.
- Stadium: id определяет name, city, capacity. Поле name определяет id, city, capacity.
- Player: id определяет club_id, first_name, last_name, position, number, birth_date. Поля first_name, last_name и club_id определяет id, position, number, birth_date.
- Match: id определяет home_club_id, away_club_id, stadium_id, match_date, status, home_score, away_score, referee. Поля home_club_id, away_club_id и match_date определяет id, stadium_id, status, home_score, away_score, referee. Поле stadium_id определяет city, capacity.

5 Представленная модель находится в третьей нормальной форме (3НФ). Это можно показать, проверив выполнение условий для каждой таблицы:

- 1НФ: Все атрибуты атомарны (неделимы). Нет повторяющихся групп или массивов;
- 2НФ: Все таблицы имеют первичный ключ. Все не ключевые атрибуты полностью функционально зависят от всего первичного ключа;
- 3НФ: В таблицах отсутствуют транзитивные зависимости. Нет не ключевых атрибутов, которые зависели бы от других не ключевых атрибутов.

Таким образом, декомпозиция проведена корректно, и модель нормализована до 3НФ.

ВЫВОД

В ходе выполнения лабораторной работы были успешно приобретены практические навыки анализа предметной области и построения концептуальной модели базы данных для системы учета футбольных матчей.

Была проведена комплексная работа по выделению основных абстракций предметной области: определены сущности (Клуб, Игрок, Стадион, Матч, Спонсор, Спонсорство), их атрибуты и связи между ними. Сформирован перечень возможных запросов к базе данных, что подтверждает глубокий анализ функциональных требований системы.

Построенная ER-диаграмма наглядно демонстрирует структуру данных и взаимосвязи между сущностями. Разработанная реляционная модель корректно отображает концептуальную модель в терминах таблиц с четко определенными доменами атрибутов, первичными и внешними ключами.

Проведенный анализ функциональных зависимостей и последующая нормализация отношений до третьей нормальной формы обеспечили устранение избыточности данных и потенциальных аномалий обновления, что подтверждает корректность проектных решений.

Полученная модель базы данных является полностью нормализованной, соответствует требованиям ЗНФ и готова к физической реализации в любой современной СУБД. Модель обеспечивает целостность данных и эффективное выполнение всех запросов, определенных в ходе анализа предметной области.