Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Проектирование и разработка баз данных интернет-приложений

Студент: Трубач Д.С.

ФИТ 3 курс 5 группа

Преподаватель: Нистюк О.А.

Минск 2024

# **Лабораторная работа № 1**

1. Описать целевую аудиторию и стратегии использования интернет-приложения.

Целевая аудитория: студенты и учащиеся, преподаватели.

Стратегия использования: обучение и последующее оценивание, самостоятельное обучение, аналитика и отчетность.

1. Разработать UML диаграммы для каждой роли приложения.

На картинках

1. Выделить основные сущности, описывающие предметную область.
2. Определить и уточнить атрибуты для каждой сущности.
3. Определить связи между сущностями.

Все на картинках

1. Преобразовать сущности в таблицы базы данных согласно выбранной модели.

MSSQL:

|  |
| --- |
| CREATE DATABASE LabBD;  USE LabBD;  CREATE TABLE Тест (  ID INT PRIMARY KEY,  Название VARCHAR(100),  Описание VARCHAR(255),  Предмет VARCHAR(100)  );  CREATE TABLE Вопрос (  ID INT PRIMARY KEY,  Текст VARCHAR(255),  Варианты\_ответов VARCHAR(255),  ID\_теста INT,  FOREIGN KEY (ID\_теста) REFERENCES Тест(ID)  );  CREATE TABLE Ответ (  ID INT PRIMARY KEY,  Текст VARCHAR(255),  Правильный BIT,  ID\_вопроса INT,  FOREIGN KEY (ID\_вопроса) REFERENCES Вопрос(ID)  );  CREATE TABLE Пользователь (  ID INT PRIMARY KEY,  Имя VARCHAR(100),  Роль VARCHAR(50),  Данные VARCHAR(255)  );  CREATE TABLE Результат\_теста (  ID\_пользователя INT,  ID\_теста INT,  Дата\_и\_время TIMESTAMP,  Итоговый\_балл FLOAT,  PRIMARY KEY (ID\_пользователя, ID\_теста),  FOREIGN KEY (ID\_пользователя) REFERENCES Пользователь(ID),  FOREIGN KEY (ID\_теста) REFERENCES Тест(ID)  ); |

ORACLE:

|  |
| --- |
| CREATE TABLE Тест (  ID NUMBER(10) PRIMARY KEY,  Название VARCHAR2(100),  Описание VARCHAR2(255),  Предмет VARCHAR2(100)  );  CREATE TABLE Вопрос (  ID NUMBER(10) PRIMARY KEY,  Текст VARCHAR2(255),  Варианты\_ответов VARCHAR2(255),  ID\_теста NUMBER(10),  FOREIGN KEY (ID\_теста) REFERENCES Тест(ID)  );  CREATE TABLE Ответ (  ID NUMBER(10) PRIMARY KEY,  Текст VARCHAR2(255),  Правильный NUMBER(1),  ID\_вопроса NUMBER(10),  FOREIGN KEY (ID\_вопроса) REFERENCES Вопрос(ID)  );  CREATE TABLE Пользователь (  ID NUMBER(10) PRIMARY KEY,  Имя VARCHAR2(100),  Роль VARCHAR2(50),  Данные VARCHAR2(255)  );  CREATE TABLE Результат\_теста (  ID\_пользователя NUMBER(10),  ID\_теста NUMBER(10),  Дата\_и\_время TIMESTAMP,  Итоговый\_балл FLOAT,  PRIMARY KEY (ID\_пользователя, ID\_теста),  FOREIGN KEY (ID\_пользователя) REFERENCES Пользователь(ID),  FOREIGN KEY (ID\_теста) REFERENCES Тест(ID)  ); |

1. Разработать логическую схему базы данных.
2. Проверить логическую схему базы данных на:

* соответствие нормальным формам до 4 НФ включительно;
  1. Чтобы таблица соответствовала 1-й нормальной форме (1NF), необходимо, чтобы все значения ее полей были неделимыми и не вычисляемыми, а все записи – уникальными (не должно быть полностью совпадающих строк). **ПРИМЕР:** в таблице Пользователь, каждое поле (ID, Имя, Роль, Данные) содержит атомарные значения, то есть каждое поле содержит только одно значение.
  2. Чтобы таблица соответствовала 2-й нормальной форме (2NF), необходимо, чтобы она находилась в 1-й нормальной форме и все не ключевые поля полностью зависели от ключевого. **ПРИМЕР:** В таблице Результат\_теста, все не ключевые атрибуты (Дата\_и\_время, Итоговый\_балл) полностью зависят от первичного ключа (ID\_пользователя, ID\_теста).
  3. Для перехода к 3-й нормальной форме (3NF), необходимо обеспечить, чтобы все таблицы находились во 2-й нормальной форме и все не ключевые поля в таблицах не зависели взаимно друг от друга. **ПРИМЕР:** в таблице Ответ, все не ключевые атрибуты (Текст, Правильный) взаимно независимы. Например, текст ответа не зависит от того, является ли ответ правильным, и наоборот.
  4. Для перехода к 4-ой нормальной форме (4NF), требуется отсутствие многозначных зависимостей. **ПРИМЕР:** в таблице Тест, нет многозначных зависимостей. Например, предмет теста не зависит от названия теста, и наоборот.
* присутствие денормализованных атрибутов;

**Денормализация** в базах данных обычно используется для улучшения производительности за счет добавления избыточных данных или группирования данных. В моей базе данных системы тестирования не видно явных денормализованных атрибутов. Все атрибуты в каждой таблице кажутся уникальными и необходимыми для этой таблицы.   
Например, в таблице Тест атрибуты Название, Описание и Предмет уникальны для каждого теста и не дублируются в других таблицах. Однако Описание и Предмет могут дублироватся.

1. Оценить сроки проекта и требуемые ресурсы.

Семестр, 4-5 месяца.