Беласин Дмитрий, ФИТ 3-1

Лабораторная работа №1

Вариант 1: Система тестирования

Задание 1: Спроектировать базу данных по своему варианту, учитывая:

• список типичных бизнес-задач;

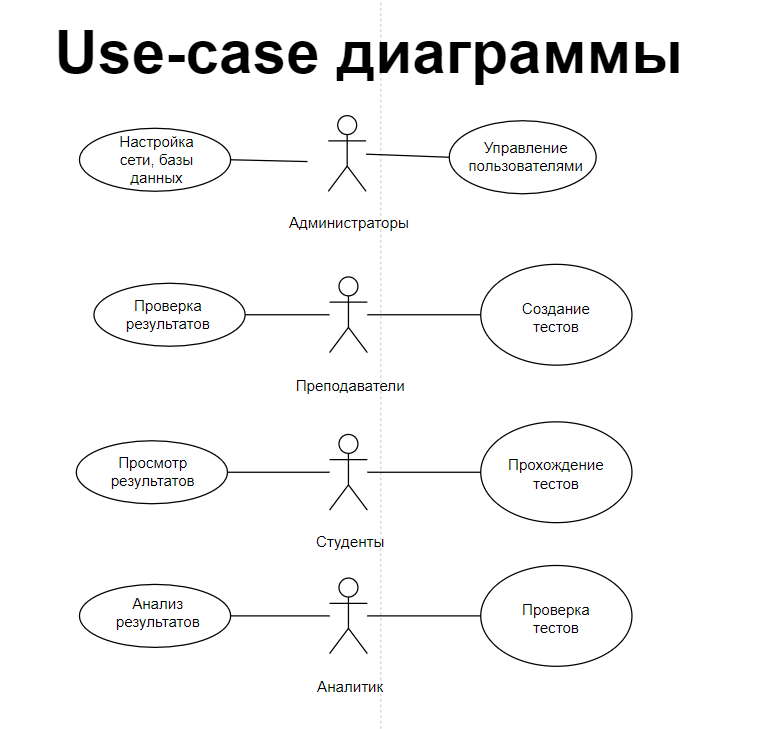
• список пользователей проекта;

Список типичных бизнес-задач:

1. Создание и управление тестами.
2. Управление пользователями (создание, редактирование, удаление).
3. Проведение тестов пользователями.
4. Сбор и анализ результатов тестирования.
5. Управление вопросами и вариантами ответов.
6. Отчеты по результатам тестирования.
7. Управление ролями и правами доступа.

Пользователи проекта:

1. Администраторы
2. Преподаватели (создают тесты)
3. Студенты (проходят тесты)
4. Аналитики (анализируют результаты)



Задание 2: Проверить логическую схему проекта на:

• нормальные формы до 4 НФ включительно;  
(Многозначная зависимость возникает, когда два или более атрибута независимо зависят от первичного ключа )

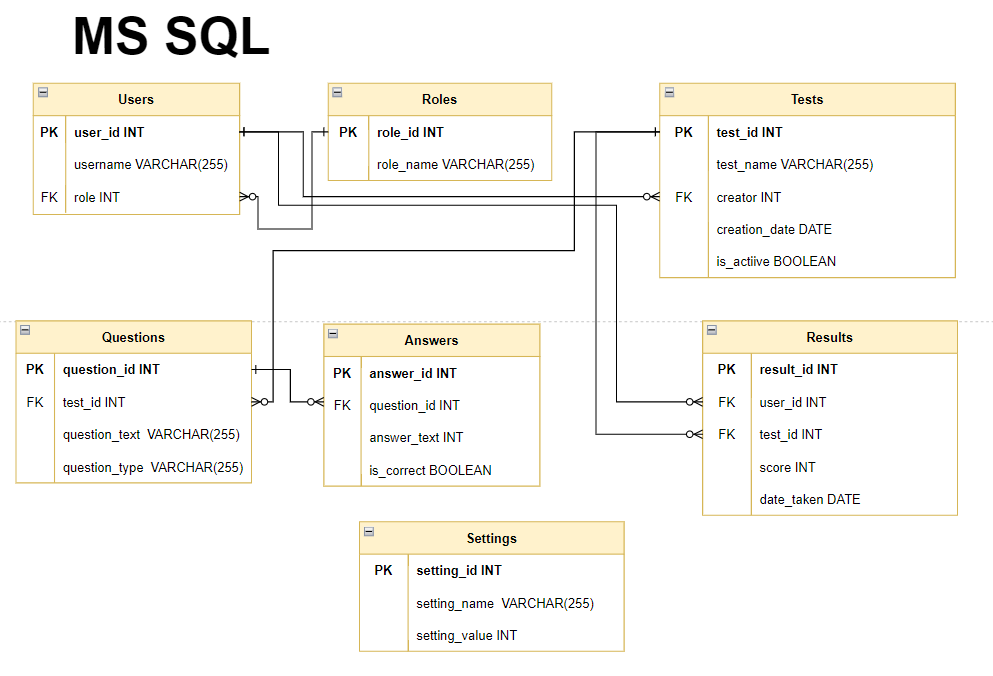
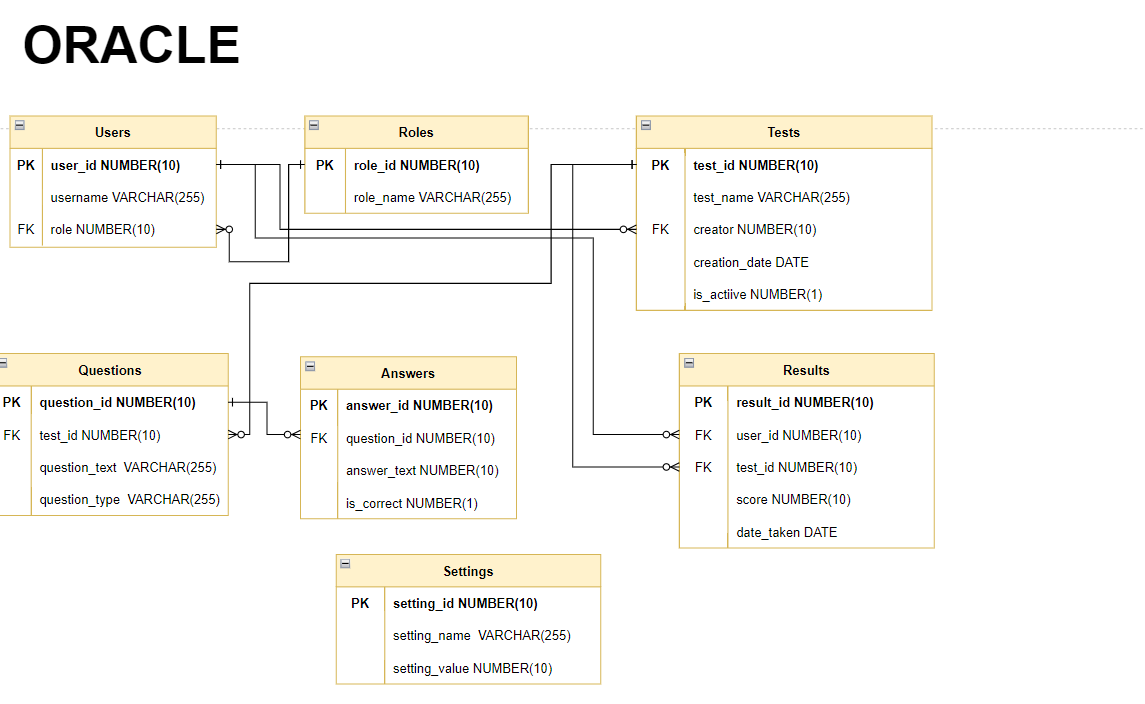
• необходимость наличия денормализованных атрибутов;  
(Денормализация — это процесс добавления избыточных данных в базу данных для повышения производительности запросов, когда данных много и таблицы сложные)

• обеспечение безопасности;

(паттерн ­– не хранить пароль в БД)

Логическая схема:



Физические схемы:  
 

Первая нормальная форма:

Правило: Каждая ячейка в таблице должна содержать только одно значение.

В схеме: Все таблицы удовлетворяют этому правилу. Нет таблиц, где в одной ячейке хранится несколько значений, разделенных запятой или другим разделителем.

Вторая нормальная форма

Правило: Таблица должна быть в 1 NF и все неключевые атрибуты должны полностью зависеть от первичного ключа.

В схеме: Все таблицы удовлетворяют этому правилу. Нет таблиц, где неключевые атрибуты не от ключевого.  
  
Третья нормальная форма:

Правило: Таблица должна быть в 2NF и неключевые атрибуты не должны зависеть друг от друга.

В схеме: Все таблицы удовлетворяют этому правилу. Нет таблиц, где неключевые атрибуты зависят друг от друга, а не от первичного ключа.

Четвертая нормальная форма:

Правило: Таблица должна быть в 3NF и не должна содержать многозначных зависимостей.

В схеме: Все таблицы удовлетворяют этому правилу. Нет таблиц, где один и тот же ключ может быть связан с несколькими записями в другой таблице.  
  
В данной схеме нет очевидной необходимости в денормализации. Все таблицы хорошо структурированы, и денормализация может привести к дублированию данных и проблемам с целостностью.

Безопасность:

Контроль доступа: Ограничить доступ к данным, чтобы только администратор мог управлять ими

Аудит: Вести журнал результатов, чтобы отслеживать изменения.

Валидация данных: Проверять вводимые данные на корректность и соответствие форматам.  
  
Задание 3: оценить сроки и требуемые ресурсы

Этап 1: Планирование и дизайн (1 месяц):

Этап 2: Разработка (2 месяца):

Этап 3: Тестирование (1-2 недели):

Этап 4: Развертывание (1 неделя):

Итого: около 4 месяцев

Требуемые сотрудники:

* 1 разработчик БД
* 1 дизайнер
* 2 разработчика приложения: разработчик серверной и клиентской части
* 2 тестировщика
* 1 системный администратор: (работа с серверами, настройкой сети и безопасностью)

Скрипт для создания таблиц в MS SQL:

|  |
| --- |
| use [Testing System]  CREATE TABLE Roles (  role\_id INT PRIMARY KEY,  role\_name NVARCHAR(255) NOT NULL UNIQUE  );  CREATE TABLE Users (  user\_id INT PRIMARY KEY,  username NVARCHAR(255) NOT NULL UNIQUE,  role INT NOT NULL,  FOREIGN KEY (role) REFERENCES Roles(role\_id)  );  CREATE TABLE Tests (  test\_id INT PRIMARY KEY,  test\_name NVARCHAR(255) NOT NULL,  creator INT NOT NULL,  date\_created DATETIME NOT NULL DEFAULT GETDATE(),  is\_active BIT NOT NULL DEFAULT 1,  FOREIGN KEY (creator) REFERENCES Users(user\_id)  );  CREATE TABLE Questions (  question\_id INT PRIMARY KEY,  test\_id INT NOT NULL,  question\_text NVARCHAR(255) NOT NULL,  question\_type NVARCHAR(255) NOT NULL,  FOREIGN KEY (test\_id) REFERENCES Tests(test\_id)  );  CREATE TABLE Answers (  answer\_id INT PRIMARY KEY,  question\_id INT NOT NULL,  answer\_text NVARCHAR(255) NOT NULL,  is\_correct BIT NOT NULL,  FOREIGN KEY (question\_id) REFERENCES Questions(question\_id)  );  CREATE TABLE Results (  result\_id INT PRIMARY KEY,  user\_id INT NOT NULL,  test\_id INT NOT NULL,  score INT NOT NULL,  date\_taken DATE NOT NULL DEFAULT GETDATE(),  FOREIGN KEY (user\_id) REFERENCES Users(user\_id),  FOREIGN KEY (test\_id) REFERENCES Tests(test\_id)  );  CREATE TABLE Settings (  setting\_id INT PRIMARY KEY,  setting\_name NVARCHAR(255) NOT NULL UNIQUE,  setting\_value NVARCHAR(255) NOT NULL  );  BEGIN TRANSACTION;  COMMIT TRANSACTION;  BEGIN TRANSACTION;  INSERT INTO Roles (role\_id, role\_name) VALUES  (1, 'Администратор'),  (2, 'Преподаватель'),  (3, 'Студент'),  (4, 'Аналитик');  INSERT INTO Users (user\_id, username, role) VALUES  (1, 'admin', 1),  (2, 'teacher1', 2),  (3, 'student1', 3),  (4, 'analyst1', 4);  INSERT INTO Tests (test\_id, test\_name, creator, date\_created, is\_active) VALUES  (1, 'Тест по математике', 2, GETDATE(), 1),  (2, 'Тест по истории', 2, GETDATE(), 1);  INSERT INTO Questions (question\_id, test\_id, question\_text, question\_type) VALUES  (1, 1, 'Сколько будет 2 + 2?', 'multiple-choice'),  (2, 1, 'Какое число является квадратом 16?', 'multiple-choice'),  (3, 2, 'Кто был первым президентом США?', 'open-ended');  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES  (1, 1, '4', 1),  (2, 1, '5', 0),  (3, 2, '256', 1),  (4, 2, '32', 0),  (5, 3, 'Джордж Вашингтон', 1),  (6, 3, 'Авраам Линкольн', 0);  INSERT INTO Results (result\_id, user\_id, test\_id, score, date\_taken) VALUES  (1, 3, 1, 10, GETDATE()),  (2, 3, 2, 8, GETDATE());  INSERT INTO Settings (setting\_id, setting\_name, setting\_value) VALUES  (1, 'MaxScore', '100'),  (2, 'PassScore', '60'),  (3, 'TestDuration', '30'); -- в минутах  COMMIT TRANSACTION; |

Скрипт для создания таблиц в ORACLE:

|  |
| --- |
| CREATE TABLE Roles (      role\_id NUMBER PRIMARY KEY,      role\_name NVARCHAR2(255) NOT NULL UNIQUE  );  CREATE TABLE Users (      user\_id NUMBER PRIMARY KEY,      username NVARCHAR2(255) NOT NULL UNIQUE,      role NUMBER NOT NULL,      FOREIGN KEY (role) REFERENCES Roles(role\_id)  );  CREATE TABLE Tests (      test\_id NUMBER PRIMARY KEY,      test\_name NVARCHAR2(255) NOT NULL,      creator NUMBER NOT NULL,      date\_created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,      is\_active NUMBER(1) DEFAULT 1,      FOREIGN KEY (creator) REFERENCES Users(user\_id)  );  CREATE TABLE Questions (      question\_id NUMBER PRIMARY KEY,      test\_id NUMBER NOT NULL,      question\_text NVARCHAR2(255) NOT NULL,      question\_type NVARCHAR2(255) NOT NULL,      FOREIGN KEY (test\_id) REFERENCES Tests(test\_id)  );  CREATE TABLE Answers (      answer\_id NUMBER PRIMARY KEY,      question\_id NUMBER NOT NULL,      answer\_text NVARCHAR2(255) NOT NULL,      is\_correct NUMBER(1) NOT NULL,      FOREIGN KEY (question\_id) REFERENCES Questions(question\_id)  );  CREATE TABLE Results (      result\_id NUMBER PRIMARY KEY,      user\_id NUMBER NOT NULL,      test\_id NUMBER NOT NULL,      score NUMBER NOT NULL,      date\_taken DATE DEFAULT CURRENT\_DATE,      FOREIGN KEY (user\_id) REFERENCES Users(user\_id),      FOREIGN KEY (test\_id) REFERENCES Tests(test\_id)  );  CREATE TABLE Settings (      setting\_id NUMBER PRIMARY KEY,      setting\_name NVARCHAR2(255) NOT NULL UNIQUE,      setting\_value NVARCHAR2(255) NOT NULL  );  INSERT INTO Roles (role\_id, role\_name) VALUES (1, 'Администратор');  INSERT INTO Roles (role\_id, role\_name) VALUES (2, 'Преподаватель');  INSERT INTO Roles (role\_id, role\_name) VALUES (3, 'Студент');  INSERT INTO Roles (role\_id, role\_name) VALUES (4, 'Аналитик');  INSERT INTO Users (user\_id, username, role) VALUES (1, 'admin', 1);  INSERT INTO Users (user\_id, username, role) VALUES (2, 'teacher1', 2);  INSERT INTO Users (user\_id, username, role) VALUES (3, 'student1', 3);  INSERT INTO Users (user\_id, username, role) VALUES (4, 'analyst1', 4);  INSERT INTO Tests (test\_id, test\_name, creator, is\_active) VALUES (1, 'Тест по математике', 2, 1);  INSERT INTO Tests (test\_id, test\_name, creator, is\_active) VALUES (2, 'Тест по истории', 2, 1);  INSERT INTO Questions (question\_id, test\_id, question\_text, question\_type) VALUES (1, 1, 'Сколько будет 2 + 2?', 'multiple-choice');  INSERT INTO Questions (question\_id, test\_id, question\_text, question\_type) VALUES (2, 1, 'Какое число является квадратом 16?', 'multiple-choice');  INSERT INTO Questions (question\_id, test\_id, question\_text, question\_type) VALUES (3, 2, 'Кто был первым президентом США?', 'open-ended');  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (1, 1, '4', 1);  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (2, 1, '5', 0);  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (3, 2, '256', 1);  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (4, 2, '32', 0);  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (5, 3, 'Джордж Вашингтон', 1);  INSERT INTO Answers (answer\_id, question\_id, answer\_text, is\_correct) VALUES (6, 3, 'Авраам Линкольн', 0);  INSERT INTO Results (result\_id, user\_id, test\_id, score) VALUES (1, 3, 1, 10);  INSERT INTO Results (result\_id, user\_id, test\_id, score) VALUES (2, 3, 2, 8);  INSERT INTO Settings (setting\_id, setting\_name, setting\_value) VALUES (1, 'MaxScore', '100');  INSERT INTO Settings (setting\_id, setting\_name, setting\_value) VALUES (2, 'PassScore', '60');  INSERT INTO Settings (setting\_id, setting\_name, setting\_value) VALUES (3, 'TestDuration', '30'); -- в минутах  SELECT \* FROM Roles;  SELECT \* FROM Users;  SELECT \* FROM Tests;  SELECT \* FROM Questions;  SELECT \* FROM Answers;  SELECT \* FROM Results;  SELECT \* FROM Settings;  COMMIT; |