

Отчёт по лабораторной работе номер 7
студента группы И-1-24
Белялова Таира

По дисциплине: База данных

Тема: Проектирование базы данных с помощью нормализации.

1. Построение модели уровня ключей (фаза 3)

Разрешение неопределённых связей и новые сущности:

Множество сущностей	Определение	Назначение
Course_Enrollments	Запись студентов на курсы	Разрешает связь M:N между Students и Courses
Lesson_Materials	Материалы уроков	Разрешает связь M:N между Lessons и Materials
User_Messages	Сообщения между пользователями	Учитывает отправителя и получателя

Описание связей:

Связь	От сущности	К сущности	Тип связи	Кратность	Описание
1	Users	Students	Наследование	1:1	Пользователь может быть студентом
2	Users	Teachers	Наследование	1:1	Пользователь может быть преподавателем
3	Teachers	Courses	Создание	1:N	Преподаватель

Связь	От сущности	К сущности	Тип связи	Кратность	Описание
					создаёт курсы
4	Courses	Lessons	Состав	1:N	Курс состоит из уроков
5	Students	Course_Enrollments	Запись	1:N	Студент записывается на курсы
6	Courses	Course_Enrollments	Набор	1:N	Курс имеет записи студентов
7	Lessons	Assignments	Содержание	1:N	Урок содержит задания
8	Students	Grades	Оценивание	1:N	Студент получает оценки
9	Assignments	Grades	Проверка	1:N	Задание имеет оценки
10	Quizzes	Grades	Результат	1:N	Тест имеет оценки

2. Определение ключевых атрибутов и доменов

Users (Пользователи)

PK: UserID (Номер)

AK1: Email (Строка)

Атрибуты: FirstName (Имя), LastName (Имя), RegistrationDate (Дата)

Students (Студенты)

PK: StudentID (Номер) + **FK:** UserID → Users

Атрибуты: EnrollmentDate (Дата), Status (Строка)

Teachers (Преподаватели)

PK: TeacherID (Номер) + **FK:** UserID → Users

Атрибуты: HireDate (Дата), Specialization (Имя)

Courses (Курсы)

PK: CourseID (Номер)

AK1: CourseName (Имя)

FK: TeacherID → Teachers

Атрибуты: Description (Описание), Price (Денежная сумма), Level (Строка)

Course_Enrollments (Записи на курсы)

PK: составной (StudentID + CourseID)

FK1: StudentID → Students

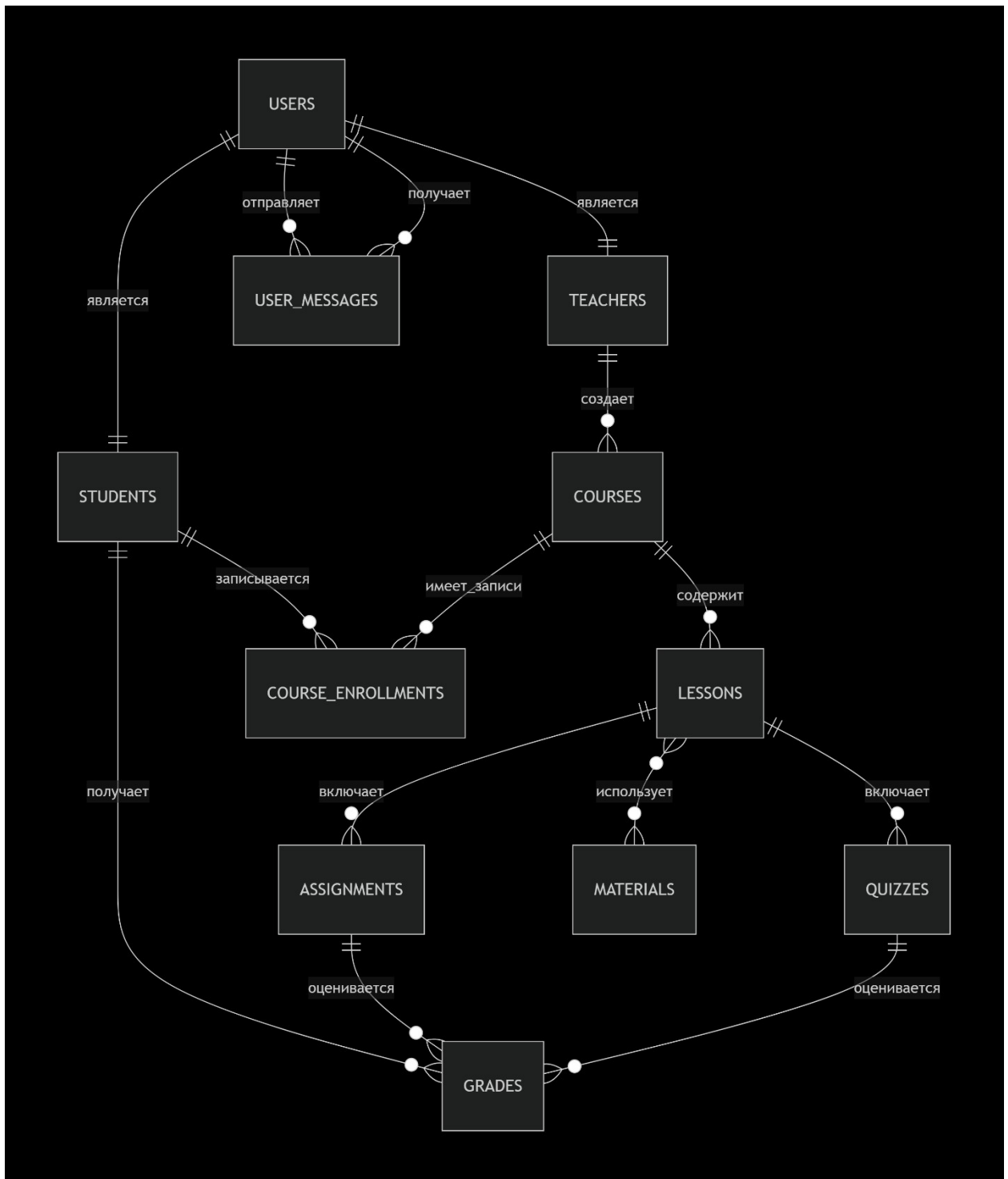
FK2: CourseID → Courses

Атрибуты: EnrollmentDate (Дата), Progress (Число)

3. Описание доменов

Домен	Общий домен	Описание
Номер	Число	Натуральные числа (ID)
Имя	Строка	Строки до 50 символов
Описание	Строка	Строки до 256 символов
Дата/время	-	Дата и время
Денежная сумма	Число	Decimal(10,2)
Строка	-	Текстовые данные

4. Концептуальная схема



Все таблицы приведены к **третьей нормальной форме (3NF)**:

1NF: Все атрибуты атомарны, нет повторяющихся групп

2NF: Нет частичных зависимостей от составного ключа

3NF: Нет транзитивных зависимостей, все неключевые атрибуты зависят только от первичного ключа

Пример проверки для таблицы Courses:

PK: CourseID

Все атрибуты (CourseName, TeacherID, Description, Price, Level, CreatedDate) зависят только от CourseID

Нет транзитивных зависимостей

5. Ограничения целостности

1. Временные ограничения:

Дата окончания курса > даты начала

Время отправки сообщения ≤ текущему времени

2. Бизнес-ограничения:

Оценка от 0 до 100 баллов

Прогресс обучения от 0 до 100%

Цена курса ≥ 0

3. Ссылочная целостность:

Каскадное удаление: User → Student/Teacher

Ограничение удаления: Course → Lessons

6. Даталогическая модель (SQL DDL)

sql

-- Таблица пользователей

CREATE TABLE Users (

 UserID INTEGER PRIMARY KEY,

 Email VARCHAR(100) UNIQUE NOT NULL,

 FirstName VARCHAR(50) NOT NULL,

 LastName VARCHAR(50) NOT NULL,

 RegistrationDate DATE NOT NULL,

 UserType VARCHAR(20) NOT NULL CHECK (UserType IN ('Student', 'Teacher'))

);

-- Таблица студентов

```
CREATE TABLE Students (  
    StudentID INTEGER PRIMARY KEY,  
    UserID INTEGER NOT NULL UNIQUE,  
    EnrollmentDate DATE NOT NULL,  
    Status VARCHAR(20) DEFAULT 'Active',  
    FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE  
);
```

-- Таблица преподавателей

```
CREATE TABLE Teachers (  
    TeacherID INTEGER PRIMARY KEY,  
    UserID INTEGER NOT NULL UNIQUE,  
    HireDate DATE NOT NULL,  
    Specialization VARCHAR(50) NOT NULL,  
    FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE  
);
```

-- Таблица курсов

```
CREATE TABLE Courses (  
    CourseID INTEGER PRIMARY KEY,  
    CourseName VARCHAR(100) NOT NULL UNIQUE,  
    TeacherID INTEGER NOT NULL,  
    Description VARCHAR(500),  
    Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0),  
    Level VARCHAR(30) NOT NULL,  
    CreatedDate DATE NOT NULL,  
    StartDate DATE NOT NULL,  
    EndDate DATE NOT NULL,
```

```
CHECK (EndDate > StartDate),  
FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID) ON DELETE  
RESTRICT  
);
```

-- Таблица уроков

```
CREATE TABLE Lessons (  
LessonID INTEGER PRIMARY KEY,  
CourseID INTEGER NOT NULL,  
Title VARCHAR(100) NOT NULL,  
Content TEXT,  
OrderNumber INTEGER NOT NULL,  
FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) ON DELETE CASCADE  
);
```

-- Таблица заданий

```
CREATE TABLE Assignments (  
AssignmentID INTEGER PRIMARY KEY,  
LessonID INTEGER NOT NULL,  
Title VARCHAR(100) NOT NULL,  
Description TEXT,  
MaxScore INTEGER NOT NULL CHECK (MaxScore > 0),  
Deadline DATE,  
FOREIGN KEY (LessonID) REFERENCES Lessons(LessonID) ON DELETE CASCADE  
);
```

-- Таблица тестов

```
CREATE TABLE Quizzes (  
QuizID INTEGER PRIMARY KEY,
```

```
LessonID INTEGER NOT NULL,  
Title VARCHAR(100) NOT NULL,  
QuestionCount INTEGER NOT NULL CHECK (QuestionCount > 0),  
TimeLimit INTEGER,  
FOREIGN KEY (LessonID) REFERENCES Lessons(LessonID) ON DELETE CASCADE  
);
```

-- Таблица материалов

```
CREATE TABLE Materials (  
    MaterialID INTEGER PRIMARY KEY,  
    Title VARCHAR(100) NOT NULL,  
    FilePath VARCHAR(255) NOT NULL,  
    FileType VARCHAR(50) NOT NULL  
);
```

-- Таблица оценок

```
CREATE TABLE Grades (  
    GradeID INTEGER PRIMARY KEY,  
    StudentID INTEGER NOT NULL,  
    AssignmentID INTEGER,  
    QuizID INTEGER,  
    Score INTEGER NOT NULL CHECK (Score >= 0 AND Score <= 100),  
    GradedDate DATE NOT NULL,  
    CHECK (AssignmentID IS NOT NULL OR QuizID IS NOT NULL),  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE  
RESTRICT,  
    FOREIGN KEY (AssignmentID) REFERENCES Assignments(AssignmentID) ON  
DELETE SET NULL,  
    FOREIGN KEY (QuizID) REFERENCES Quizzes(QuizID) ON DELETE SET NULL  
);
```


-- Таблица записей на курсы

```
CREATE TABLE Course_Enrollments (  
    StudentID INTEGER NOT NULL,  
    CourseID INTEGER NOT NULL,  
    EnrollmentDate DATE NOT NULL,  
    Progress INTEGER NOT NULL CHECK (Progress >= 0 AND Progress <= 100),  
    PRIMARY KEY (StudentID, CourseID),  
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID) ON DELETE  
    CASCADE,  
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) ON DELETE CASCADE  
);
```

-- Таблица материалов уроков

```
CREATE TABLE Lesson_Materials (  
    LessonID INTEGER NOT NULL,  
    MaterialID INTEGER NOT NULL,  
    PRIMARY KEY (LessonID, MaterialID),  
    FOREIGN KEY (LessonID) REFERENCES Lessons(LessonID) ON DELETE CASCADE,  
    FOREIGN KEY (MaterialID) REFERENCES Materials(MaterialID) ON DELETE  
    CASCADE  
);
```

-- Таблица сообщений пользователей

```
CREATE TABLE User_Messages (  
    MessageID INTEGER PRIMARY KEY,  
    SenderID INTEGER NOT NULL,  
    ReceiverID INTEGER NOT NULL,  
    MessageText TEXT NOT NULL,  
    SentDate TIMESTAMP NOT NULL,
```

```
IsRead BOOLEAN DEFAULT FALSE,  
CHECK (SentDate <= CURRENT_TIMESTAMP),  
FOREIGN KEY (SenderID) REFERENCES Users(UserID) ON DELETE CASCADE,  
FOREIGN KEY (ReceiverID) REFERENCES Users(UserID) ON DELETE CASCADE  
);
```

7. Концептуальная схема

