Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Рязанский государственный радиотехнический университет имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

дата подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Клиент-серверные приложения баз данных»**

На тему

«Создание клиент-серверного приложения сервиса для доставки еды»

Выполнил студент группы 245

Евдокимов А.В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

дата сдачи на проверку подпись

Руководитель работы

ассистент каф. ЭВМ

Баранова С. Н. \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

оценка дата защиты подпись

Рязань 2025

**Введение**

Клиент-серверные приложения являются удобным средством взаимодействия бизнеса и клиента. Современные информационные системы позволяют совершать сложные операции с данными быстро и эффективно.

С помощью информационных систем осуществляется много видов деятельности. В данной работе будет разработана система заказов еды с помощью доставки.

В настоящее время доставка – самый удобный и быстрый способ получения желаемого товара. Всё, что нужно клиенту, желающему сделать заказ, – выбрать нужное в приложении и совершить заказ.

Приложения для доставки пищи обычно имеют информацию о меню, список заказов пользователя и информацию о пользователе.

Пользователь может сделать заказ на свой адрес и получить его от курьера.

Цель работы заключается в получение навыков разработки клиент-серверного приложения.

Задачей работы является анализ выбранной предметной области, выявление требований и бизнес-правил предметной области и создание клиент-серверного приложения по предметной области.

**Постановка задачи**

**Описание автоматизируемой области**

Информационная система «Доставка еды» предназначена для совершения пользователем заказов: комплектации (выбора желаемых блюд) и выбора места для доставки. Помимо этого, пользователь может просматривать информацию о себе, своих заказах и все товары, существующие в информационной системе.

Система позволяет взаимодействовать бизнесу и пользователю. Взаимодействие происходит при помощи совершения пользователем заказов пищи. Пользователь может совершить заказ, выбрав блюда из списка всех блюд. При совершении заказа пользователь должен ввести данные, такие как адрес, куда будет доставлен заказ. Пока заказ не отправлен производителю, пользователь может редактировать его, добавляя и удаляя оттуда товары. После отправки заказа, производитель начинает выполнение заказа, уведомляя клиента о состоянии готовности заказа. После выполнения заказа пользователь получает доставку с помощью курьера предприятия, которое получило заказ.

**Выделение основных функций и бизнес-правил автоматизируемой области**

Информационная система выполняет следующие **функции**:

1. Идентификация пользователя
2. Представление пользователю информации
3. Совершение заказа
4. Уведомление пользователя о статусе заказа

Для выбранной предметной области были получены **бизнес-правила**:

1. Обслуживание будет проводиться только в городах на территории РФ
2. У пользователя должны быть обязательно известны логин, пароль, имя пользователя и телефон
3. Логин пользователя должен быть уникален
4. Логин должен быть больше 5 символов, но меньше 30.
5. Пароль должен быть больше 5 символов, но меньше 16
6. Пароли хранятся в hash кодах
7. Имя должно быть от 3 до 19 символов
8. Телефон пользователя должен быть уникален
9. Телефон пользователя должен быть представлен в формате номеров РФ (+7/8)
10. Пользователь может добавить электронную почту для отправки чеков. Почта должна иметь формат \*@\*.\*, где \* - любой символ.
11. При изменении пароля новый пароль должен отличаться от старого.
12. Пользователь имеет аккаунт, в который ему требуется войти для совершения заказа
13. В одном заказе могут быть блюда только одного производителя
14. Клиент может иметь много адресов
15. Выбор адреса для доставки осуществляется при добавлении первого блюда в заказ
16. Адреса не могут повторяться у одного пользователя, но могут у разных
17. Пользователь может собирать только один заказ
18. Пока заказ формируется, курьер неизвестен
19. Одни и те же блюда можно включать в заказ несколько раз
20. Логин производителя уникален и имеет больше 5, но меньше 30 символов
21. Наименование производителя имеет меньше или ровно 30 символам
22. Пока заказ собирается, нельзя удалить адрес формируемого заказа

**Разработка архитектуры ИС**

Архитектурой ИС будет клиент-серверная двухзвенная архитектура, представленная в виде сервера и клиентского приложения.

Моделью ИС будет являться распределённая модель. Из-за отсутствия прослойки между клиентом и базой данных, часть функций будет возложено на клиент, а часть на СУБД по средствам хранимых процедур.

**Разработка серверной части информационной системы**

**Инфологическое проектирование БД**

При проектировании базы данных был использован ER метод, включающий в себя:

1. **Выделение сущностей и связей между ними**

Для выбранной предметной области были выделены следующие сущности: Клиент, Адрес Клиента, Заказ, Курьер, Производитель, Блюдо, Ингредиенты, Уведомления

1. **Построение ER диаграммы классов, включающий все выделенные сущности и определения характеристик этих связей.**

Связь Клиент-Уведомление

Клиенту направляется личное сообщение, которое прочитать может только он. Пользователю может приходить много сообщений с указанием, какому пользователю. Пользователь необязательно должен получить хотя бы одно сообщение, но если сообщение отправлено, то известно кому оно отправлено. Отношение 1:N с классом принадлежности Необязательный –Обязательный. Отношение приведено на рисунке 1.



Рисунок 1 - Отношение Клиент-Уведомление

Связь Клиент-Адрес клиента

Клиент может иметь много адресов (бизнес-правило). Адрес не может существовать без привязки к Клиенту. Адрес принадлежит конкретному клиенту (при повторении адресов они считаются разными атрибутами, поскольку принадлежат разным клиентам). Клиент необязательно может иметь адрес. Отношение 1:N с классом принадлежности Необязательный – Обязательный. Отношение приведено на рисунке 2.



Рисунок 2 - Отношение Клиент-Адрес клиента

Связь Заказ – Адрес Клиента

Заказ совершается на конкретный адрес клиента. На один и тот же адрес может быть совершено несколько заказов. Заказ обязательно имеет адрес клиента. Также на адрес может быть не совершено заказов вообще. Отношение N:1 с классом принадлежности Обязательный – Необязательный. Отношение приведено на рисунке 3.



Рисунок 3 - Отношение Заказ – Адрес Клиента

Связь Курьер – Заказ

Каждый отдельный курьер может доставлять много заказов. Один заказ доставляется одним курьером. Курьер необязательно доставляет хотя бы один заказ. Курьер не известен при создании заказа (бизнес-правило). Отношение 1:N с классом принадлежности Необязательный – Необязательный. Отношение приведено на рисунке 4.



Рисунок 4 - Отношение Курьер – Заказ

Связь Заказ-Блюдо

Заказ содержит много блюд. Много блюд входят в заказ. Каждое блюдо необязательно содержатся в заказе, но заказ не существует без любого блюда. Отношение N:N с классом принадлежности Обязательный – Необязательный. Отношение приведено на рисунке 5.



Рисунок 5 - Отношение Заказ-Блюдо

Связь Производитель – Блюдо

Производитель производит много блюд. Каждое блюдо принадлежит конкретному производителю. Производитель необязательно производит блюда, но блюдо обязательно имеет производителя. Отношение 1:N с классом принадлежности Необязательный – Обязательный. Отношение приведено на рисунке 6.



Рисунок 6 - Отношение Производитель – Блюдо

Связь Блюдо-Ингредиенты

Каждое блюдо необязательно имеет ингредиенты. Ингредиенты могут не быть в блюде вообще (если не указано производителем). Отношение N:N с классом принадлежности Необязательный – Необязательный. Отношение приведено на рисунке 7.



Рисунок 7 - Отношение Блюдо-Ингредиенты

Общая ER диаграмма приведена на рисунке 8.



Рисунок 8 – ER диаграмма

1. **Формирование набора предварительных отношений по ER диаграмме с указанием первичного ключа для каждого отношения**

По связям из ER диаграммы выявлены следующие отношения:

**Связь Клиент-Уведомление**

Данная связь будет отображена в базе данных, по правилу 4, как два отношения: по одному для каждой сущности, где ключом является ключ соответствующей сущности и в отношении для N-связной сущности добавляется ключ односвязной сущности.

В отношение Уведомление войдёт поле Логин Клиента и станет частью первичного ключа.

**Связь Клиент-Адрес клиента**

Данная связь будет отображена в базе данных, по правилу 4, как два отношения: по одному для каждой сущности, где ключом является ключ соответствующей сущности и в отношении для N-связной сущности добавляется ключ односвязной сущности.

В отношение Адрес Клиента войдёт поле Логин Клиента и станет частью первичного ключа.

**Связь Заказ – Адрес Клиента**

Данная связь будет отображена в базе данных, по правилу 4, как два отношения: по одному для каждой сущности, где ключом является ключ соответствующей сущности и в отношении для N-связной сущности добавляется ключ односвязной сущности.

В отношение Заказ войдёт поле Адрес Клиента.

**Связь Курьер – Заказ**

Данная связь будет отображена в базе данных, по правилу 5, как 3 отношения: по одному для каждой сущности, где ключом является ключ соответствующей сущности, и одно для связи. Последнее содержит ключи обеих сущностей; первичным ключом является ключ N связной сущности.

Появится дополнительное отношение **Курьер-Заказ**, в котором первичным ключом будет выступать первичный ключ отношения Заказ.

**Связь Заказ-Блюдо**

Данная связь будет отображена в базе данных, по правилу 6, как 3 отношения: по одному на каждую сущности, где ключом является ключ соответствующей сущности, и одно для связи, в котором предварительный ключ содержит ключи обеих сущностей.

Появится дополнительное отношение **Заказ-Блюдо**, в котором первичным ключом будет выступать первичные ключи отношений Заказ и Блюдо.

**Связь Производитель-Блюдо**

Данная связь будет отображена в базе данных, по правилу 4, как два отношения: по одному для каждой сущности, где ключом является ключами будут ключи соответствующей сущности и в отношении для N-связной сущности добавляется ключ односвязной сущности.

В отношение Блюдо войдёт поле Логин Производителя.

**Связь Блюдо-Ингредиенты**

Данная связь будет отображена в базе данных, по правилу 6, как 3 отношения: по одному на каждую сущности, где ключом является ключ соответствующей сущности, и одно для связи, в котором предварительный ключ содержит ключи обеих сущностей.

Появится дополнительное отношение **Блюдо-Ингредиенты**, в котором первичным ключом будет выступать первичные ключи отношений Ингредиенты и Блюдо.

В таблице 1 приведены первичные ключи для всех сущностей.

Таблица 1 – Первичные ключи

|  |  |
| --- | --- |
| **Отношение** | **Первичный ключ** |
| Клиент | Логин Клиента |
| Адрес Клиента | Логин Клиента, Город, Район, Улица, Здание, Помещение |
| Заказ | Номер Заказа |
| Курьер | Номер-Серия Паспорта |
| Производитель | Логин Производителя |
| Блюдо | Название, Логин Производителя |
| Ингредиенты | Название Ингредиента |
| Уведомления | Логин Клиента, Дата/Время, Сообщение |
| Курьер-Заказ | Номер Заказа |
| Заказ-Блюдо | Название Блюда, Логин Производителя, Номер заказа |
| Блюдо-Ингредиенты | Название Блюда, Логин Производителя, Название Ингредиента |

Пояснение к отношению Адрес Клиента: адреса разных клиентов могут повторяться, но у одного клиента не может быть двух одинаковых адресов.

Пояснение к отношению Курьер: и Серия-номер паспорта, и Номер трудовой книги – уникальные идентификаторы, то есть потенциальные ключи. Был выбран первичный ключ Серия-номер паспорта, так как номер паспорта содержит больше символов (10), чем номер трудовой книги (7). Значит его уникальность выше.

Пояснение к отношению Блюдо: блюда с одним и тем же названием могут встречаться у производителей, но не могут встречаться у одного производителя.

Пояснение к отношению Уведомления: в одно и то же время многим пользователям могут прийти одинаковые сообщения (например, при массовой рассылке об акциях). Также пользователю в одно время могут прийти уведомления о разных заказах.

1. **Подготовка списка всех представляющих интерес атрибутов, кроме тех, которые уже были выделены. Распределение атрибутов по отношениям, где каждый атрибут может попасть только в одно отношение**

Для выделенных сущностей были выбраны и распределены атрибуты:

Для отношения Клиент: Пароль, Имя, Телефон, Электронная почта, Активность

Для отношения Адрес Клиента: Активность

Для отношения Заказ: Время Заказа, Статус Заказа, Оценка Заказа, Время Завершения Заказа

Для отношения Курьер: Фамилия, Имя, Отчество, Телефон, Трудовая книга

Для отношения Блюдо: Изображение, Цена, Описание, Калории, Масса, Видимость

Для отношения Производитель: Пароль, Наименование, Оценка, Регион, Город, Район, Улица, Здание, Этаж, Помещение

Для отношения Уведомления: Просмотрено

Для отношения Блюдо-Заказ: Количество блюд в заказе

1. Проверка отношений на БКНФ

Все отношения соответствуют 1 НФ, так как имеют ровно одно значение в атрибуте.

Для проверки на 2 НФ были получены зависимости:

**Отношение Клиент**

**Отношение Адрес Клиента**

**Отношение Заказ**

**Отношение Курьер**

**Отношение Блюдо**

{Название, Производитель} → Изображение, Цена, Описание, Калории, Масса, Видимость

**Отношение Производитель**

**Отношение Уведомление**

**Отношение Курьер-Заказ**

**Отношение Блюдо-Заказ**

Не упомянутые отношения не имеют зависимых от первичного ключа атрибутов.

В отношениях нет зависимости не ключевых атрибутов от части составного ключа, то есть они соответствуют 2НФ.

В отношениях нет транзитивных зависимостей и соответствуют 3НФ, то есть ни одни не ключевой атрибут не определяет другой не ключевой атрибут.

В отношениях детерминанты всех функциональных зависимостей являются потенциальными ключами, то есть они находятся в БКНФ.

**Даталогическое проектирование**

По полученным отношениям была построена схема базы данных. Результат приведён на рисунке 2.



Рисунок 2 – Схема базы данных

Для технической оптимизации строковые значения были заменены на целочисленные значения.

Пояснение: сравнение строковых значений – сложная операция, требующая больших вычислительных мощностей, чем сравнение целочисленных значений типа integer.

Были изменены поля:

1. Уведоление.Логин -> Уведомление.ID клиента
2. Клиент.Логин -> Клиент.ID (поле Логин остаётся уникальным, но больше не является первичным ключом)
3. Адрес.Логин пользователя -> Адрес.ID клиента
4. Производитель.Логин производителя -> Производитель.Номер производителя (поле логин остаётся уникальным, но больше не является первичным ключом)
5. Курьер.Серия-номер паспорта -> Курьер.ID (поле Серия-номер паспорта остаётся уникальным, но больше не является первичным ключом).
6. Ингредиент.Название -> Ингредиент.ID ингредиента (поле Название остаётся уникальным, но больше не является первичным ключом)

Для оптимизации хранимых значений поле Статус отношения Заказ будет вынесено в отдельную таблицу Статусы.

Для ускорения выборки данных в отношение Заказ был добавлен атрибут ID клиента.

Схема базы данных приведена на рисунке 3.



Рисунок 3 – Финальная схема базы данных

**Разработка ограничений БД**

**Поля таблицы Клиент**

Id клиента будет использовать тип данных int, предоставляющий значений, то есть 4 294 967 296 значений.

Из бизнес-правил известно, что логин должен быть меньше 30 символов, но начальное значение 5 символов. Для данного поля будет использоваться тип nvarchar(30). Выбор данного типа обоснован наличием в строках кириллицы, поэтому тип varchar не подходит.

Также из бизнес-правила известно, что логин уникален, на данное поле будет наложено ограничение UNIQUE.

Из бизнес-правила известно, что пароль будет в hash кодах. В качестве алгоритма хеширования будет использоваться SHA256. Он имеет длину выходного значения 32 байта. Для хранения пароля будет использоваться тип данных binary(32).

Из бизнес-правила известно, что имя содержит от 3 до 19 символов. Для хранения имени будет использоваться nvarchar(20). Выбор данного типа обоснован наличием в строках кириллицы.

Из бизнес-правила известно, что телефон представлен в формате номеров РФ, то есть начинается с +7 или 8 и имеет 12/11 символов.

Для реализации данного поля будет использоваться пользовательский тип данных, наследующийся от типа char(12), на который будет наложена проверка на соответствие формату номера +7\*\*\*\*\*\*\*\*\*\*, где \* - цифра 0-9. Реализация номера через 8 будет производиться на клиентском приложении.

Электронная почта будет использовать varchar(50), так как в электронной почте используются только латинские символы, что соответствует кодировке ASCII. Также поле электронная почта будет проверяться на соответствие формату \*@\*.\*,

Активность будет использовать tinyint для передачи логического значения (правда-ложь).

**Поля таблицы Адрес Клиента**

Номер адреса будет использовать тип int.

ID клиента будет взят из таблицы Клиент, так как это внешний ключ.

Поле ID помещения будет использовать тип данных int, взятый из таблицы Помещений как внешний ключ.

Поле активность будет иметь тип данных tinyint.

**Поля таблицы Производитель**

Номер производителя будет использовать тип int.

Из бизнес-правил известно, что логин производителя, так же, как и логин пользователя, должен быть меньше 30 символов, но начальное значение 5 символов. Для данного поля будет использоваться тип nvarchar(30).

Также из бизнес-правила известно, что логин производителя уникален, на данное поле будет наложено ограничение UNIQUE.

Пароль, хранящийся в hash-кодах будет использовать тип binary(32).

Из бизнес-правил известно, что наименование производителя будет иметь 30 символов, для данного поля будет использоваться тип nvarchar(30).

Оценка будет использовать тип данных tinyint, на который будет наложено ограничение от 0 до 5 (в соответствии со звёздами в подобных приложениях).

Поле помещение будет иметь тип данных int из таблицы Помещение.

**Поля таблицы Статус**

Поле ID будет иметь тип данных smallint, так как набор данных будет ограничиваться небольшим константным значение количества статусов.