

«АКАДЕМИЯ TOP»

ПРОЕКТ

По дисциплине: «Технология доступа к базам данных ADO.NET»

На тему: «Система управления заказами службы доставки еды.»

Выполнили: Михайлов,
Кулягин(РПО 28/2)

ВВЕДЕНИЕ

Цель проекта:

Разработать систему управления заказами службы доставки еды, позволяющий отображать сам заказ, статус заказа, пользователей.

Задачи проекта:

1. Разработать саму базу данных,
2. Заполнить их нужными данными для работы программы,
3. Создать интерфейс позволяющий добавлять заказ,
4. Смотреть заказ, проверить его статус, редактировать
5. Провести тестирование.

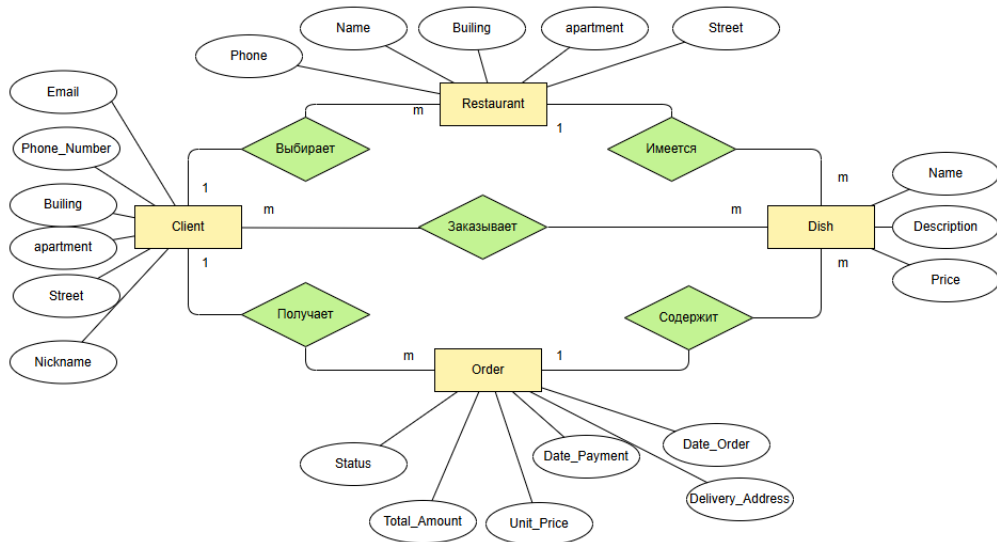
Язык программирования: C#

СУБД: SQLite

Программа для реализации: MS Visual Studio

Объекты базы данных: Client, Restaurant, Dish, Order, OrderStatus, OrderItem.

ER Диаграмма



Описание ER-Диаграммы:

1. Client

Client_Id - уникальный идентификатор пользователя

Email - эл. почта

Building - номер здания

apartment - номер квартиры

street - улица

Nickname - имя пользователя

2. Restaurant

Restaurant_Id - уникальный идентификатор ресторана

Name - название

Building - номер здания

Apartment - номер квартиры

Street - улица

Phone - номер телефона

3. Dish

Dish_Id - уникальный идентификатор блюда

Name - название

Description - описание

Price - цена

4. Order

Order_Id - уникальный идентификатор чека

Dish_Id – уникальный идентификатор блюда

Status - Статус заказа

Total_Amount - общая сумма заказа

Unit_Price - цена каждой позиции отдельно

Date_Payment - дата оплаты

Date_Order - дата оформления заказа

Delivery_Address - адрес доставки

Связи между сущностями:

Client M:M Dish - много клиентов может заказать разные блюда

Client 1:M Order - клиент может иметь много чеков, но не наоборот

Client 1:M Restaurant - клиент выбирает ресторан

Restaurant 1:M Dish - Один ресторан имеет много блюд

Order 1:M Dish - В одном чеке может быть много блюд

ПРОЦЕСС НОРМАЛИЗАЦИИ БАЗЫ ДАННЫХ

1. НЕНОРМАЛИЗОВАННАЯ БАЗА ДАННЫХ

На начальном этапе все данные хранились в одной большой таблице. Это приводило к множеству проблем.

Order_ID	Client_Data	Restaurant_Data	Dish_List	Order_DateTime	Total_Amount
1	"Владимир Кулягин, +99990933, cocucoh@mail.ru "	"ОВОО Пицца, улица Пушкина 1"	"Пицца четыре сыра"	2023-04-12, 12:30	799.99
2	"Михайлов Богдан, +111223344, vludick@yandex.ru "	"ОВОО Пицца, улица Пушкина 1"	"Пицца Маргарита"	2024-08-07, 17:42	689.99

Проблемы ненормализованной формы:

1. **Избыточность данных:** Данные о клиенте и ресторане дублировались в каждом заказе.
2. **Аномалии обновления:** При смене номера телефона ресторана "ОВОО Пицца" необходимо было обновить его в каждой строке таблицы.
3. **Аномалии удаления:** Если удалить все заказы для ресторана "ОВОО Пицца", то информация о самом ресторане безвозвратно терялась.
4. **Аномалии добавления:** Нельзя было добавить в систему новый ресторан, пока ему не сделали ни одного заказа.

2. 1NF

Цель: Устранить повторяющиеся группы, обеспечить атомарность данных.

Шаги:

1. Выделить отдельные сущности
2. Создать первичные ключи
3. Убедиться в атомарности всех полей

TABLE Client					
Client_ID	Email	Building	Apartment	Street	NickName
1	gavgav@gmail.com	дом 43	350	Макаренко	Bobr22

TABLE Restaurant					
Restaurant_ID	Name	Building	Apartment	Street	Phone
1	ОВОПИЦЦА	дом 42	1	Макаренко	+79799549893

TABLE Dish				
Dish_Id	Name	Description	Price	Restaurant_ID
1	Пиццы Маргарита	(описание пиццы)	899.00	1

TABLE Order							
Order_ID	Client_ID	Dish_ID	Status	Total_Amount	Date_Order	Date_Payment	D_Addre:
1	1	1	принят	899.00	2025-05-23	2025-05-23	Макарен дом 43 к 350

Проблема после 1NF: В таблице Order поле Dish_Id нарушает 1NF - один заказ может содержать несколько блюд.

3. 2NF

Цель: Устранить частичные зависимости от составного ключа.

Шаги:

1. Вынести статусы в отдельную таблицу
2. Создать таблицу для позиций заказа (OrderItem)

TABLE OrderStatus			
Status_Id	Status_Code	Status_Name	Description
1	confirmed	Подтвержден	Ресторан принял заказ и начал готовку

TABLE OrderItem				
Order_Item_Id	Order_Id	Dish_Id	Quantity	Unit_Price
1	1	1	3	2779.0

Что улучшилось:

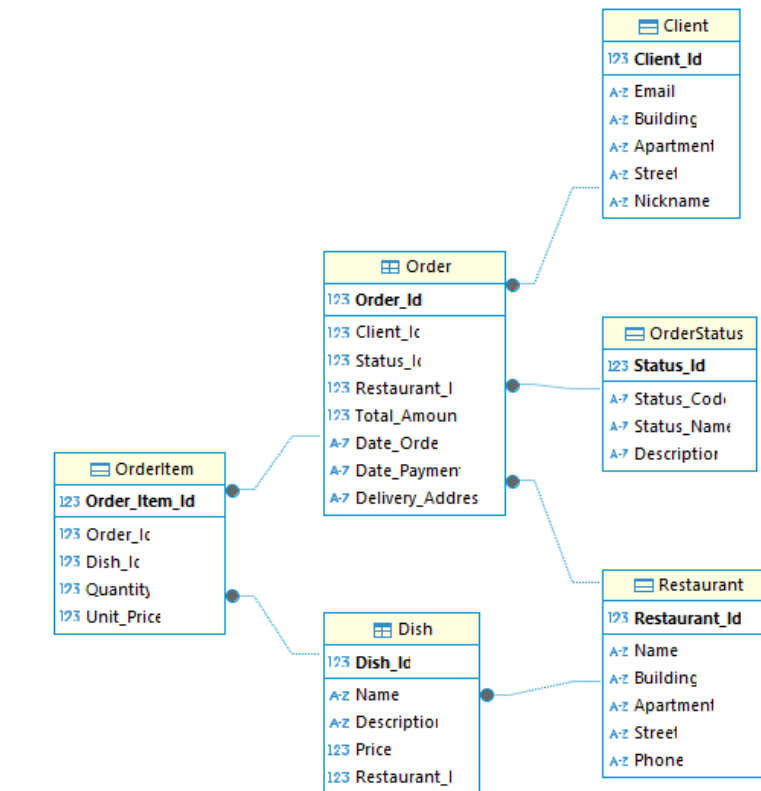
1. Статусы вынесены в отдельную таблицу
2. Один заказ теперь может содержать несколько блюд через OrderItem
3. Устранены частичные зависимости

4. 3NF

На этом этапе мы проверили, что в получившихся таблицах неключевые поля зависят непосредственно от первичного ключа, а не друг от друга.

- **Нет избыточности данных** - каждая сущность в своей таблице
- **Минимизированы аномалии обновления** - изменение адреса клиента не влияет на исторические заказы
- **Правильные связи** - внешние ключи обеспечивают целостность
- **Отсутствие транзитивных зависимостей** - все атрибуты зависят только от первичного ключа своей таблицы

БАЗА ДАННЫХ:



ИНТЕРФЕЙС ПРОГРАММЫ

1 - Начало программы: Главное меню

```
=====
=== СЛУЖБА ДОСТАВКИ ЕДЫ | КЛИЕНТ ID: 1 ===
=====
Выберите действие:
1. Оформить новый заказ
2. Отследить мои заказы / Редактировать
3. Посмотреть мой профиль
4. Редактировать адрес доставки
0. Выход
Ваш выбор:
```

2 – Оформление нового заказа, адрес указывается автоматически(берется из профиля), после нужно выбрать ресторан, блюдо и количество блюд(можно выбрать несколько позиций для одного заказа), после чего подтвердить оформление заказа.

```
--- Адрес доставки ---
Используется адрес из профиля: ул. Ключикино, д. 52, кв. 250

--- Выбор ресторана ---
1. Пиццерия "Маргарита" (ул. Цветочная, 22)
2. Суши-бар "Сакура" (пр. Победы, 10)
Введите ID ресторана: █
```

```
--- Меню: Суши-бар "Сакура" ---

Добавление блюда:
3. Ролл Филадельфия (450,00 Р)
4. Мисо суп (150,00 Р)
Введите ID блюда для добавления (или 0 для завершения): 3
Введите количество для 'Ролл Филадельфия': 30
Добавлено: 30 x Ролл Филадельфия. Всего позиций в заказе: 1

Добавление блюда:
3. Ролл Филадельфия (450,00 Р)
4. Мисо суп (150,00 Р)
Введите ID блюда для добавления (или 0 для завершения): 0

--- Подтверждение заказа ---
Адрес: ул. Ключикино, д. 52, кв. 250
Сумма к оплате: 13 500,00 Р
Оформить заказ? (y/n): y
```

3 – Управление заказами. Можно взаимодействовать с различными заказами: Поиск, Редактировать Заказ, Удалить Заказ, Сбросить поиск/Обновить поиск.

```
--- Управление Заказами ---
Ваши последние заказы:
Заказ №7 от 16.10.2025 06:14 | Ресторан: Суши-бар "Сакура" | Сумма: 13 500,00 Р | СТАТУС: Новый
Заказ №6 от 16.10.2025 05:27 | Ресторан: Суши-бар "Сакура" | Сумма: 18 000,00 Р | СТАТУС: Новый
Заказ №5 от 16.10.2025 05:21 | Ресторан: Пиццерия "Маргарита" | Сумма: 8 400,00 Р | СТАТУС: Новый
Заказ №4 от 13.10.2025 11:04 | Ресторан: Суши-бар "Сакура" | Сумма: 1 350,00 Р | СТАТУС: Новый
Заказ №3 от 13.10.2025 10:53 | Ресторан: Пиццерия "Маргарита" | Сумма: 970,00 Р | СТАТУС: Отменен
Заказ №2 от 13.10.2025 09:23 | Ресторан: Суши-бар "Сакура" | Сумма: 150,00 Р | СТАТУС: Новый
Заказ №1 от 13.10.2025 09:10 | Ресторан: Пиццерия "Маргарита" | Сумма: 970,00 Р | СТАТУС: Готовится

Выберите действие:
S. Поиск (по ID, статусу или ресторану)
E. Редактировать заказ (статус/адрес)
D. Удалить заказ
R. Сбросить поиск / Обновить список
0. Назад в главное меню
Ваш выбор: _
```

4 – Посмотреть профиль

```
--- Мой Профиль ---
Никнейм: Иван_И
Email: ivanov@example.com
Адрес доставки: ул. Ключкино, д. 52, кв. 250
```

5 – Редактирование адреса: Нужно ввести улицу, дом, квартиру

```
4. Редактировать адрес доставки
0. Выход
Ваш выбор: 4

--- Редактирование адреса доставки ---
Текущий адрес: ул. Ключкино, д. 52, кв. 250
Введите новую улицу (текущая: ул. Ключкино): ул. Пушкино
Введите новый дом/корпус (текущий: 52): 42
Введите новую квартиру (текущая: 250, можно оставить пустым): 111

[✓] Адрес успешно обновлен!
Новый адрес: ул. Пушкино, д. 42, кв. 111
```

ТЕСТЫ

1. Создание заказа и расчёт суммы

Параметр	Описание
Цель Теста	Убедиться что метод PlaceOrder корректно создает заказ в базе, включая позиции, и точно рассчитывает общую сумму(TotalAmount).
Предусловия	1. Клиент с ID существует. 2. Ресторан с ID 1 существует. 3. Блюда с ID 1(цена 10.00) и ID 2(цена 20.00) существуют.
Действие	Вызов PlaceOrder для клиента 1 с позициями 5 шт. Блюда 1(5 на 10.00 = 50.00) - 1 шт. Блюда 2(1 на 20.00 = 20.00)
Ожидаемый результат	1. Возвращается новый OrderId > 0. 2. При последующем запросе(через GetClientOrders) проверяется, что TotalAmount нового заказа равно 70.00. 3. Тестовый заказ успешно удаляется.

2. Фильтрация и сортировка

Параметр	Описание
Цель Теста	Проверить логику сортировки заказов по сумме (DESC) и способность фильтровать список по конкретному ID заказа.
Предусловия	У клиента с ID 1 должно быть минимум два существующих заказа с разными суммами
Действие 1	Запрос всех заказов клиента 1, отсортированных по полю TotalAmount в убывающем порядке(DESC).
Ожидаемый результат 1(Сортировка)	Список заказов должен быть отсортирован таким образом, что TotalAmount каждого заказа больше или равно сумме следующего заказа в списке.
Действие 2	Запрос заказов клиента 1 с фильтром ID самого первого заказа из отсортированного списка
Ожидаемый результат(Фильтрация)	Возвращает список, содержащий ровно один заказ, ID которого совпадает с искомым ID

Результат тестов:

```
=====
=== ЗАПУСК РЕАЛЬНЫХ ИНТЕГРАЦИОННЫХ ТЕСТОВ ===
=== Клиент ID: 1, Ресторан ID: 1 ===
=====

--- ТЕСТ 1: Создание Заказа и Расчет Суммы ---
[+] 1.1: Заказ успешно создан с ID: 10
[+] 1.2: Проверка данных заказа. Сумма 70,00 ₽ и адрес верны.
[+] Тестовый заказ №10 удален.

--- ТЕСТ 2: Сортировка по Сумме (DESC) и Фильтрация по ID ---
[+] 2.1: Сортировка 6 заказов по сумме (DESC) прошла успешно.
[+] 2.2: Фильтрация по ID заказа (7) прошла успешно.

=== ТЕСТЫ ЗАВЕРШЕНЫ ===

Нажмите любую клавишу для продолжения...
```

ВЫВОД

Поставленная цель и задачи были выполнены.

Были сложности с:

1. Созданием ER-Диаграммы;
2. Часто изменялась и дорабатывалась база данных;
3. Программа очень часто крашилась, изза чего его приходилось часто пересобирать.

Но у нас получилось преодолеть трудности и выполнить нашу цель и задачи.