«АКАКДЕМИЯ TOP»

**РЕФЕРАТ**

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

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

Выполнили: Михайлов, Кулягин(РПО 28/2)

Санкт-Петербург, 2025

**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.

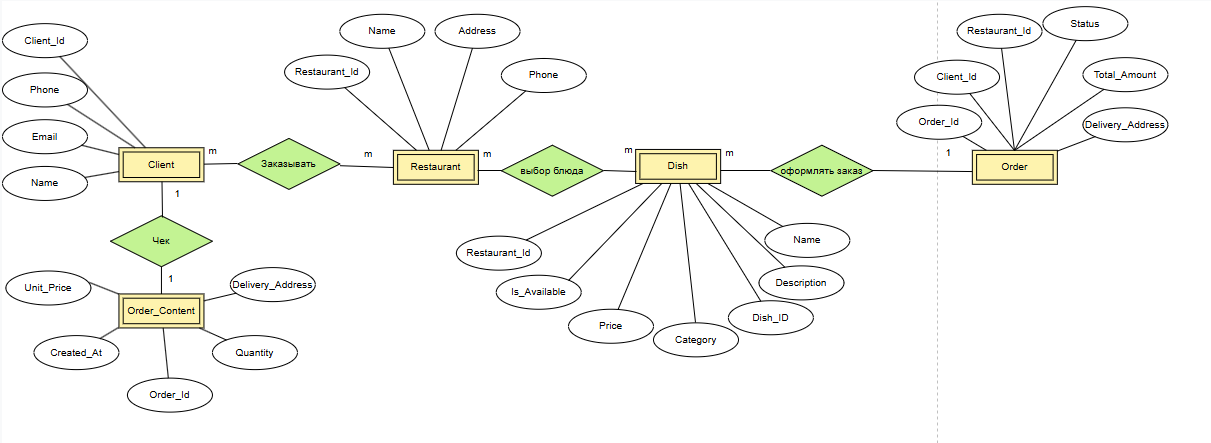
**Язык программирования:** C#

**СУБД:** SQLite

**Программа для реализации:** MS Visual Studio

**Объекты базы данных:** Client, Order, Dish, Order\_Content, Courier, Restaurant.

**ER Диаграмма**

****

**Описание ER-Диаграммы:**

**1. Client**

Client\_Id - уникальный идентификатор клиента

Name - Имя клиента

Email - почта

Phone - номер телефона

**2. Restaurant**

Restaurant\_Id - уникальный идентификатор ресторана

Name - имя

Address - адрес ресторана

Phone - номер телефон ресторана

**3. Dish**

Dish\_Id - уникальный идентификатор блюда

Restaurant\_Id - уникальный идентификатор ресторана

Is\_Available - доступно ли блюдо

Price - цена

Category - категория блюда

Description - описание блюда

Name - название

**4. Customer\_Order**

Order\_Id - уникальный идентификатор заказа

Client\_Id - уникальный идентификатор клиента

Restaurant\_Id - уникальный идентификатор ресторана

Dish\_Id - уникальный идентификатор блюда

Status - статус заказа(по умолчанию: Принят)

Total\_Amount - полная сумма заказа

Delivery\_Address - адрес доставки

**5. Order\_Content**

Order\_Id - уникальный идентификатор заказа

Delivery\_Address - адрес доставки

Quantity - количество позиций

Unit\_Price - конечная цена

Created\_At - время создания заказа

**Связи между сущностями:**

1 - Client M:M Restaurant - Много клиентов заказывает через различные рестораны.

2 - Restaurant M:M Dish - В разных ресторанах много различных блюд.

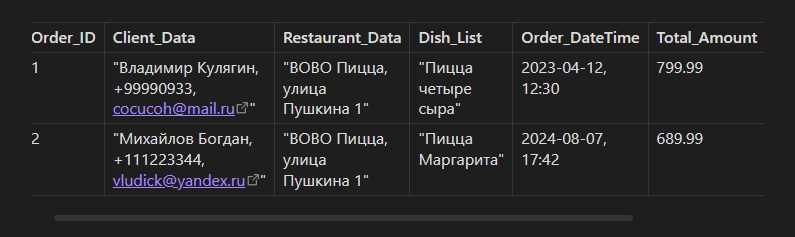
3 - Dish M:1 Order - Много блюд в один заказ.

4 - Client 1:1 Order\_Content - У каждого клиента уникальный чек.

**ПРОЦЕСС НОРМАЛИЗАЦИИ БАЗЫ ДАННЫХ**

**1. НЕНОРМАЛИЗОВАННАЯ БАЗА ДАННЫХ**

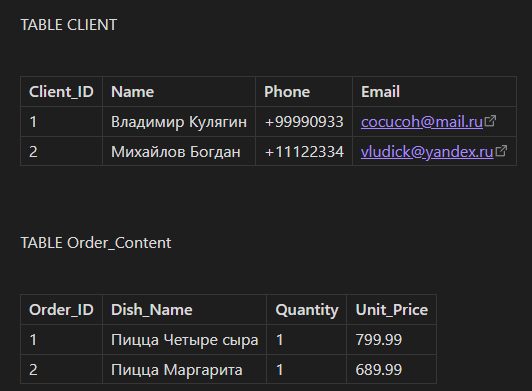
На начальном этапе все данные хранились в одной большой таблице. Это приводило к множеству проблем.



**Проблемы ненормализованной формы:**

1. **Избыточность данных:** Данные о клиенте и ресторане дублировались в каждом заказе.
2. **Аномалии обновления:** При смене номера телефона ресторана "ВОВО Пицца" необходимо было обновить его в каждой строке таблицы.
3. **Аномалии удаления:** Если удалить все заказы для ресторана "ВОВО 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**

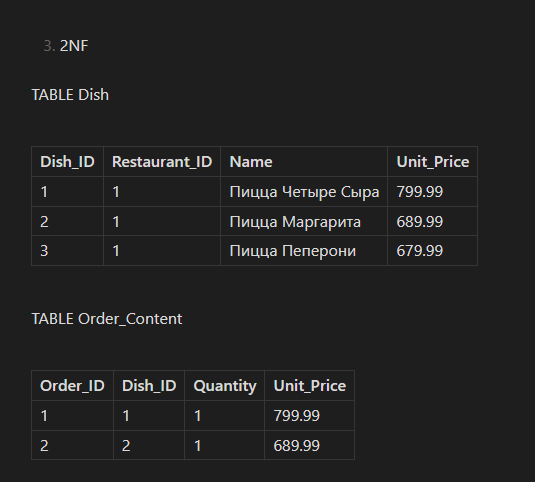
****

Разделение составных полей:

1. Поле Client\_Data было разделено на отдельные столбцы: Name, Phone, Email.
2. Поле Restaurant\_Data было разделено на Name, Address, Phone.
3. Для решения проблемы с Dish\_List была создана отдельная таблица ORDER\_CONTENT, где каждая строка представляет собой одну позицию из заказа.

Были созданы таблицы CLIENT и RESTAURANT с уникальными идентификаторами (ID).

**3. 2NF**

****

**Выявленная проблема:**

В таблице **ORDER\_CONTENT** первичный ключ был составным: **(Order\_ID, Dish\_Name).** Однако атрибут **Unit\_Price** зависел только от **Dish\_Name**, но не от **Order\_ID**. Это означало, что если бы цена на "Пицца Четыре сыра" изменилась, нам пришлось бы искать и обновлять все старые заказы, что неверно с точки зрения исторической целостности.

**Проведенные изменения:**

1. **Создание сущности DISH:** Была создана таблица DISH с уникальным Dish\_ID. В ней хранится актуальная информация о блюде: название, описание, текущая цена и связь с рестораном.
2. **Модификация ORDER\_CONTENT**: В таблице ORDER\_CONTENT ссылка на блюдо теперь осуществляется через Dish\_ID. Ключевым улучшением стало то, что Unit\_Price теперь сохраняет цену блюда на момент заказа. Таким образом, этот атрибут полностью зависит от всей пары (Order\_ID, Dish\_ID).

**4. 3NF**

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

* **В CLIENT**: Name, Phone и Email зависят от Client\_ID и независимы друг от друга.
* **В RESTAURANT**: Name, Address, Phone зависят от Restaurant\_ID.
* **В CUSTOMER\_ORDER**: Total\_Amount зависит от Order\_ID. Хотя эта сумма может быть вычислена из ORDER\_CONTENT, ее хранение является целесообразной денормализацией для повышения производительности отчетов и избежания постоянных сложных вычислений JOIN с агрегацией.

**БАЗА ДАННЫХ:**

**КЛИЕНТ (CLIENT)**

CREATE TABLE CLIENT (

Client\_ID INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL,

Phone TEXT NOT NULL,

Email TEXT

);

**ЗАКАЗ (ORDER)**

CREATE TABLE CUSTOMER\_ORDER (

Order\_ID INTEGER PRIMARY KEY AUTOINCREMENT,

Client\_ID INTEGER NOT NULL,

Restaurant\_ID INTEGER NOT NULL,

Total\_Amount REAL NOT NULL,

Status TEXT NOT NULL DEFAULT 'Принят',

Delivery\_Address TEXT NOT NULL,

FOREIGN KEY (Client\_ID) REFERENCES CLIENT(Client\_ID),

FOREIGN KEY (Restaurant\_ID) REFERENCES RESTAURANT(Restaurant\_ID),

);

**БЛЮДО (DISH)**

CREATE TABLE DISH (

Dish\_ID INTEGER PRIMARY KEY AUTOINCREMENT,

Restaurant\_ID INTEGER NOT NULL,

Name TEXT NOT NULL,

Description TEXT,

Price REAL NOT NULL,

Category TEXT,

Is\_Available INTEGER NOT NULL DEFAULT 1, -- 1 = true, 0 = false

FOREIGN KEY (Restaurant\_ID) REFERENCES RESTAURANT(Restaurant\_ID)

);

**РЕСТОРАН (RESTAURANT)**

CREATE TABLE RESTAURANT (

Restaurant\_ID INTEGER PRIMARY KEY AUTOINCREMENT,

Name TEXT NOT NULL,

Address TEXT NOT NULL,

Phone TEXT NOT NULL,

);

**СОСТАВ\_ЗАКАЗА (ORDER\_CONTENT)**

CREATE TABLE ORDER\_CONTENT (

Order\_ID INTEGER NOT NULL,

Quantity INTEGER NOT NULL DEFAULT 1,

Unit\_Price REAL NOT NULL,

Delivery\_Address TEXT NOT NULL,

Created\_At DATETIME DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (Order\_ID),

FOREIGN KEY (Order\_ID) REFERENCES CUSTOMER\_ORDER(Order\_ID),

);