Анализ предметной области

Ресторан «Оникс» — это уютное и популярное место. Гости делают заказы, кухня готовит вкусные блюда, а официанты подают их. Клиенты наслаждаются едой и приятной атмосферой.

Ключевые задачи ресторана:

1. Принятие заказов от гостей.
2. Передача заказов на кухню и бар.
3. Приготовление блюд и сборка заказов.
4. Обслуживание гостей: подача блюд, решение вопросов, прием дополнительных заказов.
5. Расчет с гостями: формирование и печать чека, прием оплаты.
6. Учет и анализ: контроль популярных блюд, остатков продуктов и выручки.

До автоматизации процесс регистрации и учета заказов осуществляется устаревшими способами. Процесс происходит следующим образом:

1. Принятие заказов: Официант подходит к столику, записывает пожелания гостей в блокнот или на листок.
2. Передача на кухню: Официант относит бумажный лист с заказом на кухню и отдает его повару.
3. Приготовление: Повар получает заказ, вносит его в свой список и приступает к приготовлению. Если заказ потерялся или записан неразборчиво, он уточняет его у официанта.
4. Подача блюд: когда блюдо готово, повар сообщает об этом официанту. Официант забирает готовый заказ и относит его гостям.
5. Расчет: В конце визита официант вручную подсчитывает итоговую сумму счета на калькуляторе или в уме, выписывает чек и принимает оплату.

Как было отмечено ранее в бизнес процессе используются устаревшие способы учета заказов, такие как запись заказов на бумаге, подсчет суммы заказов вручную и т.д. В процессе учета заказов не используются существующие программные продукты для процесса учета заказов. Поэтому можно выделить ряд недостатков в её работе.

1. Риск человеческих ошибок: Неразборчивый почерк, неверно услышанное или записанное блюдо приводит к тому, что гостям приносят не тот заказ.
2. Потеря времени: Постоянные перемещения официанта между залом и кухней для передачи заказов и уточнений замедляют обслуживание.
3. Низкая организация: На кухне может скапливаться стопка бумажных заказов, которые сложно отсортировать по срочности или очередности. Легко потерять или перепутать листок.
4. Задержки в обслуживании: В час пик поварам и официантам сложно быстро обрабатывать большое количество бумажных заказов, что увеличивает время ожидания для гостей.
5. Сложности учета: Нет оперативных данных о том, какие блюда популярны, какие продукты заканчиваются, и какова выручка за текущий день.

Переход с ручного учета заказов на автоматизированную деятельность важен, т.к. это позволит снизить риск ошибок, улучшит предоставление услуг ресторана

На основании выявленных недостатков в бизнес-процессе, было принято решение о разработке программы, которая будет выполнять следующие функции:

1. Функции для официанта (Интерфейс приема заказов):

Выбор столика. Начало работы с новым заказом.

Формирование заказа.

На экране отображается актуальное меню (список блюд и их цены), подгружаемое из базы данных.

Официант добавляет выбранные гостями позиции в заказ.

Программа автоматически подсчитывает сумму.

Отправка заказа на кухню.

По нажатию кнопки «Отправить» заказ сохраняется в базу данных со статусом «Принят» и мгновенно появляется в программе на кухне.

Каждому заказу присваивается уникальный номер.

Отслеживание статуса. Официант видит, какие из его заказов готовятся, а какие уже готовы к подаче.

Формирование чека. После обслуживания программа формирует фискальный чек для печати. Вся информация для чека (состав заказа, суммы) берется из базы данных.

2. Функции для повара (Интерфейс кухни):

Просмотр очереди заказов. На главный экран выводятся все заказы со статусом «Принят».

Изменение статуса заказа.

Повар начинает готовить – меняет статус на «Готовится».

Когда блюдо готово, повар меняет статус заказа на «Готов».

Уведомление официантов. Как только статус заказа меняется на «Готов», это сразу отображается в интерфейсе официанта.

Пример заказа:

Заказ №: 142

Столик №: 5

Статус: Принят

Время заказа: 19:30

Пример чека:

РЕСТОРАН "ОНИКС"

ул. Медниковых, 4 ст1, г. Торжок

ИНН 7712345678 КПП 771201001

Кассовый чек № 4567

Дата: 12.11.2025 13:45

СТОЛИК №5

---------------------------------------------

№ Наименование Кол. Цена Сумма

---------------------------------------------

1 Тар-тар из говядины 1 450.00 450.00

2 Паста Карбонара 2 580.00 1160.00

3 Апельс. сок, 0.3л 2 300.00 600.00

---------------------------------------------

ИТОГО: 2210.00

---------------------------------------------

Спасибо за визит!

Пример меню:

|  |  |  |
| --- | --- | --- |
| Категория | Название блюда | Цена |
| Напитки | Сок | 300 |
| Вода | 100 |
| Десерты | Чизкейк | 200 |
| Фондан | 300 |
| Основные | Паста Карбонара | 400 |
| Пельмени | 300 |

Описание модели базы данных

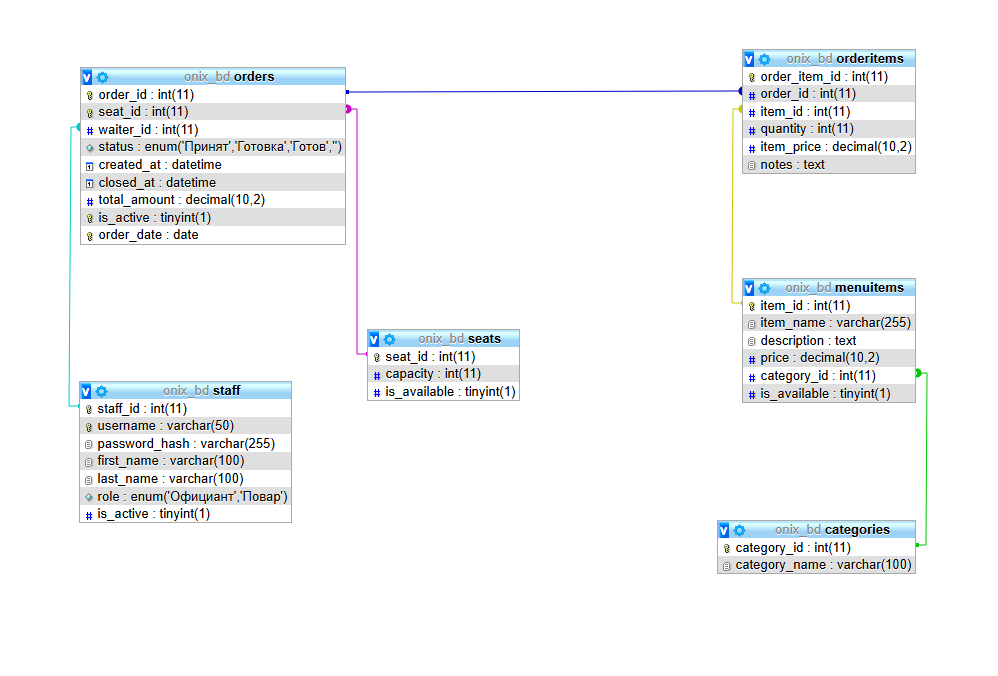


Рисунок 1 – ER-диаграмма

1. Таблица: categories (Категории меню)

Назначение: Группировка блюд по категориям

Первичный ключ (PK): category\_id

Атрибуты:

category\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор категории

category\_name (VARCHAR(100)) - Название категории

2. Таблица: menuitems (Позиции меню)

Назначение: Блюда и напитки, доступные для заказа

Первичный ключ (PK): item\_id

Атрибуты:

item\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор блюда

item\_name (VARCHAR(255)) - Название блюда

description (TEXT) - Описание блюда

price (DECIMAL(10,2)) - Текущая цена

category\_id (INT, FK) - Категория блюда

is\_available (BOOLEAN) - Доступно для заказа

3. Таблица: staff (Персонал)

Назначение: Данные сотрудников ресторана

Первичный ключ (PK): staff\_id

Атрибуты:

staff\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор сотрудника

username (VARCHAR(50)) - Логин для входа

password\_hash (VARCHAR(255)) - Хэш пароля

first\_name, last\_name (VARCHAR(100)) - Имя и фамилия

role (ENUM) - Должность сотрудника ('Официант', 'Повар')

is\_active (BOOLEAN) - Активен ли сотрудник

4. Таблица: seats (Столики)

Назначение: Информация о столиках в зале

Первичный ключ (PK): seat\_id

Атрибуты:

seat\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор столика

capacity (INT) - Вместимость (количество гостей)

is\_available (BOOLEAN) - Активен ли столик

5. Таблица: orders (Заказы)

Назначение: Основная информация о заказе

Первичный ключ (PK): order\_id

Атрибуты:

order\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор заказа

seat\_id (INT, FK) - Столик, для которого заказ

waiter\_id (INT, FK) - Официант, принявший заказ

status (ENUM) - Текущий статус заказа ('Принят', 'Готовка', 'Готов')

created\_at (DATETIME) - Время создания заказа

closed\_at (DATETIME) - Время оплаты/закрытия

total\_amount (DECIMAL(10,2)) - Общая сумма заказа

is\_active (BOOLEAN) - Активен ли заказ

order\_date (DATE) - Дата заказа

6. Таблица: orderitems (Состав заказа)

Назначение: Связь между заказами и позициями меню

Первичный ключ (PK): order\_item\_id

Атрибуты:

order\_item\_id (INT, AUTO\_INCREMENT, PK) - Уникальный идентификатор позиции заказа

order\_id (INT, FK) - Заказ

item\_id (INT, FK) - Позиция меню

quantity (INT) - Количество

item\_price (DECIMAL(10,2)) - Цена на момент заказа

notes (TEXT) - Пожелания гостя

Связи между сущностями модели БД

1. categories — menuitems

Тип связи: Один-ко-многим (1:N)

Описание: Одна категория может содержать много блюд

Необходима для: структурирования меню для удобного отображения

2. staff — orders

Тип связи: Один-ко-многим (1:N)

Описание: Один официант может принимать много заказов

Необходима для: учета ответственности за заказы

3. seats — orders

Тип связи: Один-ко-многим (1:N)

Описание: Один столик может иметь много заказов (в разное время)

Необходима для: привязки заказа к месту в зале

4. orders — orderitems

Тип связи: Один-ко-многим (1:N)

Описание: Один заказ может содержать много позиций

Необходима для: детализации состава заказа

5. menuitems — orderitems

Тип связи: Один-ко-многим (1:N)

Описание: Одно блюдо может быть во многих заказах

Необходима для: связи позиции заказа с описанием блюда

6. orders — menuitems (через orderitems)

Тип связи: Многие-ко-многим (N:M)

Описание: Многие заказы могут содержать многие блюда

Необходима для: реализации сложных заказов с несколькими позициями