

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Пензенский государственный технологический университет»

(ПензГТУ)

Факультет автоматизированных информационных технологий

Кафедра «Информационные технологии и системы»

Дисциплина «Управление данными»

КУРСОВАЯ РАБОТА

на тему: «Разработка базы данных интернет ресурса по доставке еды»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ПензГТУ 2.09.03.02.16.ПЗ

Выполнил: студент гр. 19ИС16п Степанчев В.С.

Проверил: ст. преподаватель каф. ИТС Пискаев К.Ю.

Работа защищена с оценкой: _____

Пенза 2021

Утверждаю
зав. каф. ИТС Михеев М.Ю.
«____» _____

ЗАДАНИЕ

на курсовую работу по дисциплине «Управление данными»

Студента Степанчев В.С.

Группа 19ИС1бп

Тема работы: «Разработка базы данных интернет ресурса по доставке еды»

Исходные данные (технические требования на работу)

1. Назначение базы данных: хранение данных интернет ресурса доставки еды
2. Состав и структура исходных данных: список пользователей, продуктов, магазинов
3. Перечень сущностей (может изменяться в процессе проектирования): продукты, список магазинов, пользователи
3. Содержание расчетной части: введение, анализ предметной области, моделирование предметной области, создание и запуск базовых запросов SQL, создание и запуск продвинутых запросов SQL, создание и использование представлений, заключение, список литературы, приложения.
4. Графическая часть:
 - схема БД (физическая) (____ лист(ов) формата А4);
 - схема БД MS SQL Server (____ лист(ов) формата А4);

5. Экспериментальная часть:

- разработка схемы БД с использованием Sparx Systems Enterprise Architect.
- составление и отладка запросов в Microsoft SQL Server Management Studio for Microsoft SQL Server 2008/10/12;

6. Срок выполнения работы по разделам:

- | | |
|--|-------|
| 1. Постановка задачи до | _____ |
| 2. Разработка схемы БД до | _____ |
| 3. Написание базовых запросов до | _____ |
| 4. Написание продвинутых запросов до | _____ |
| 5. Оформление графической части до | _____ |
| 6. Оформление пояснительной записки до | _____ |

Студент _____ / _____ /

Задание получил «___» _____ 201_г.

Руководитель _____ / _____ /

Задание выдал «___» _____ 201_г.

Дата защиты работы _____

Студент _____ / _____ /

Руководитель _____ / _____ /

Содержание

Введение.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1 Общие сведения	8
1.2 Выбор СУБД.....	8
1.3 Сравнительный анализ аналогов.....	9
1.5 Перечень хранимой информации: таблицы, поля, типы	14
1.6 Необходимые предполагаемые запросы к БД.....	16
1.7 Выводы по разделу	17
2. МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	18
2.1 Общие сведения	18
2.2 Выделение сущностей, атрибутов, ключей, связей.....	18
2.3 Проектирование БД и перенос ее на СУБД.	20
2.4 Создание БД в PostgreSQL. Перенос схемы из ЕА.....	21
2.5 Создание диаграммы средствами DataGrip.....	23
2.6 Начальное заполнение БД.....	24
2.7 Вывод по разделу	24
3. СОЗДАНИЕ И ЗАПУСК БАЗОВЫХ ЗАПРОСОВ SQL	25
3.1 Общие сведения	25
3.2 Базовые запросы.....	25
3.3 Выводы по разделу	32
4. СОЗДАНИЕ И ЗАПУСК ПРОДВИНУТЫХ ЗАПРОСОВ	33

Подпись и дата	Инд. № дубл.	2.3 Проектирование БД и перенос ее на СУБД.	20									
		2.4 Создание БД в PostgreSQL. Перенос схемы из ЕА.....	21									
		2.5 Создание диаграммы средствами DataGrip.....	23									
		2.6 Начальное заполнение БД.	24									
		2.7 Вывод по разделу	24									
Взам. инв. №	Подпись и дата	3. СОЗДАНИЕ И ЗАПУСК БАЗОВЫХ ЗАПРОСОВ SQL	25									
		3.1 Общие сведения	25									
		3.2 Базовые запросы.....	25									
		3.3 Выводы по разделу	32									
Подпись и дата	Инд. № подл.	4. СОЗДАНИЕ И ЗАПУСК ПРОДВИНУТЫХ ЗАПРОСОВ	33									
		ПензГТУ 2.09.03.02.16										
Инд. № подл.	Подпись и дата	Изм	Лист	№ докум.	Подпись	Дата	Разработка базы данных интернет ресурса по доставке еды	Лит	Лист	Листов		
		Разраб.	Степанчев В.С.									
		Провер.	Пискаев К.Ю.								4	47
		Реценз.										
		Н. Контр.										
		Утверд.										
							Гр. 19ИС16п					

4.1. Общие сведения	33
4.2 Продвинутое запросы	33
4.3 Выводы по разделу	38
Заключение	39
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	40
ПРИЛОЖЕНИЕ А. СХЕМА БАЗЫ ДАННЫХ	42
ПРИЛОЖЕНИЕ Б. SQL КОД БАЗЫ ДАННЫХ	43

Инв. № подл.	Подпись и дата				Инв. № дубл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата	
Изм	Лист	№ докум.	Подпись	Дата	ПензГТУ 02.09.03.02.16					Лист
										5

Введение

В наши дни большинство занятых на работе людей не имеют возможности заниматься приготовлением обедов на работу, ужинов дома. В такие моменты в голову приходит мысль о заказе готовой еды на дом либо в офис. Актуальность подобной идеи только растет из-за занятости населения и желания сэкономить время, чем стоять весь день у плиты. Неотъемлемой частью сервиса по доставке еды является база данных.

База данных — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

Целью курсовой работы является разработка собственной базы данных сервиса по доставке еды.

Задачами курсовой работы являются:

- 1) Анализ предметной области, выделение справочных и оперативных данных
- 2) Проектирование диаграммы сущность-связь, перенос диаграммы на целевую СУБД, заполнение базы данных
- 3) Написание базовых и продвинутых SQL-запросов, проверка их работоспособности

В первом разделе будет проведен анализ предметной области, приведены общие сведения о теме курсовой работы, будут выделены справочные и оперативные данные путем сравнения аналогов, будет представлен перечень хранимой информации (таблицы, поля, типы) и будут представлены предполагаемые запросы к базе данных.

Во втором разделе будут представлены общие сведения о моделировании предметной области, будут выделены сущности, атрибуты, ключи и связи, будет проведено проектирование диаграммы сущность-связь в Enterprise Architect, будет проведена подготовка диаграммы сущность-связь

для переноса на целевую СУБД, будет создана база данных, будет проведено наполнение БД данными.

В третьем разделе будут приведены общие сведения о создании и запуске базовых запросов SQL, будут написаны такие запросы, как select, update, insert, delete, будут проведены тесты и представлены результаты выполнения запросов

В четвертом разделе будут приведены общие сведения о создании и запуске продвинутых запросов SQL, будут написаны такие запросы, как multi table select, create table, drop table, alter table, create index, alter index, drop index, select into, будут проведены тесты и представлены их результаты

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Общие сведения

В первом разделе будут приведены общие сведения о предметной области курсовой работы, будут выделены справочные и оперативные данные, которые будет хранить созданная база данных, будет предоставлен перечень хранимой в базе данных информации, а именно таблицы, поля и типы, и будут предоставлены предполагаемые запросы к базе данных.

В курсовой работе будет создана база данных для сервиса доставки еды. Сервис доставки еды — некий ресурс, на котором собраны предложения всех ресторанов, кафе, пиццерий и других заведений, которые предоставляют услугу доставки блюд в том или ином городе. Доставка осуществляется с помощью курьеров.

База данных - это упорядоченный набор структурируемой информации, которая хранится в электронном виде. Базы данных управляются системой управления базами данных. В современных базах данных данные формируются в виде строк и столбцов в ряде таблиц, чтобы обеспечить эффективность обработки и запросов данных.

1.2 Выбор СУБД

База данных в курсовом проекте будет реализована с помощью PostgreSQL [8].

PostgreSQL – мощная система объектно-реляционной базы данных с открытым исходным кодом, активная разработка которой насчитывает более 30 лет, что принесло ей прочную репутацию за надежность, функциональность и производительность.

Сильными сторонами PostgreSQL считаются:

- 1) Высокопроизводительные и надежные механизмы транзакций и репликации
- 2) Расширяемая система встроенных языков программирования
- 3) Наследование

- 4) Возможность индексирования геометрических объектов и наличие базирующегося на ней расширения PostGIS
- 5) Встроенная поддержка слабоструктурированных данных в формате JSON с возможностью их индексации
- 6) Расширяемость

1.3 Сравнительный анализ аналогов

Для того, чтобы понять, как будет выглядеть готовая база данных необходимо провести анализ существующих аналогов сайта доставки еды. Первым таким аналогом является Яндекс Еда. На главной странице сайта мы видим список категорий продуктов, которые покупатель может заказать: суши, бургеры, пиццу, десерты, грузинскую еду, шашлык. Кроме того, на главной странице сайта показан список ресторанов, из которых пользователь может сделать заказ. Также, в правой верхнем углу страницы расположена кнопка перехода в ваш личный профиль.

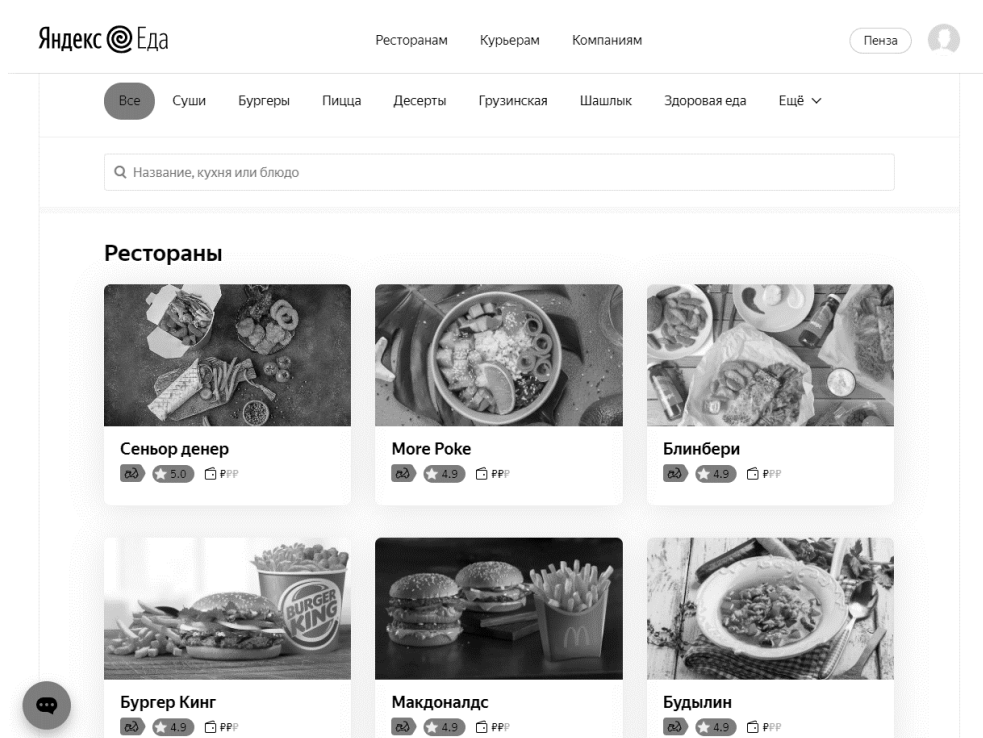


Рисунок 1 – Главная страница Яндекс Еды

Во вкладке профиля расположены следующие пункты:

- 1) Мои данные

2) Мои адреса

3) Мои заказы

Во вкладке «Мои данные» находятся ваши личные данные, а именно номер телефона, адрес электронной почты и имя пользователя.

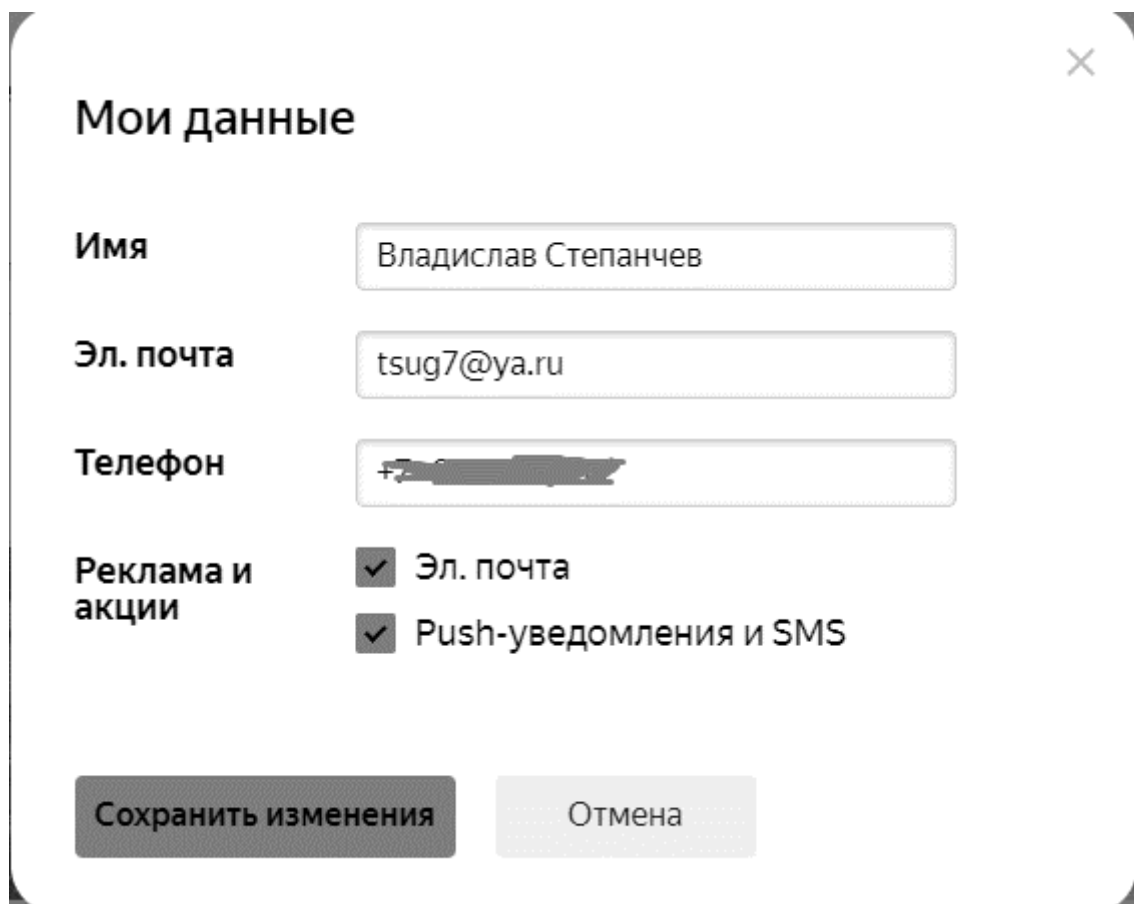


Рисунок 2 – Вкладка «Мои данные»

Во вкладке «Мои адреса» находятся адреса доставки, которые укажет пользователь.

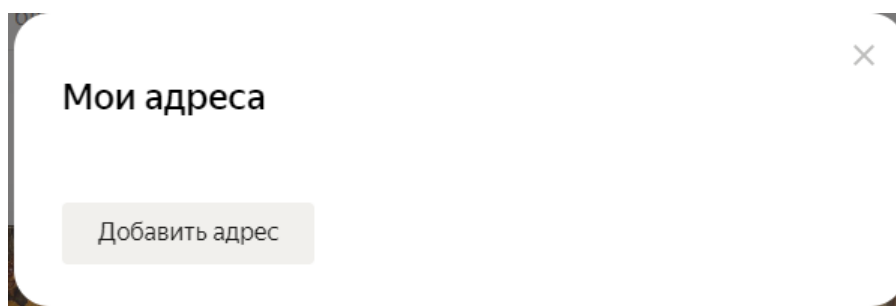


Рисунок 3 – Вкладка «Мои адреса»

Во вкладке «Мои заказы» находятся заказы, которые пользователь когда-либо делал.



Рисунок 4 – Вкладка «Мои заказы»

Для оформления заказа пользователю необходимо добавить выбранный им товар в корзину. После добавления товара, система высчитывает сумму заказа и отображает ее пользователю.

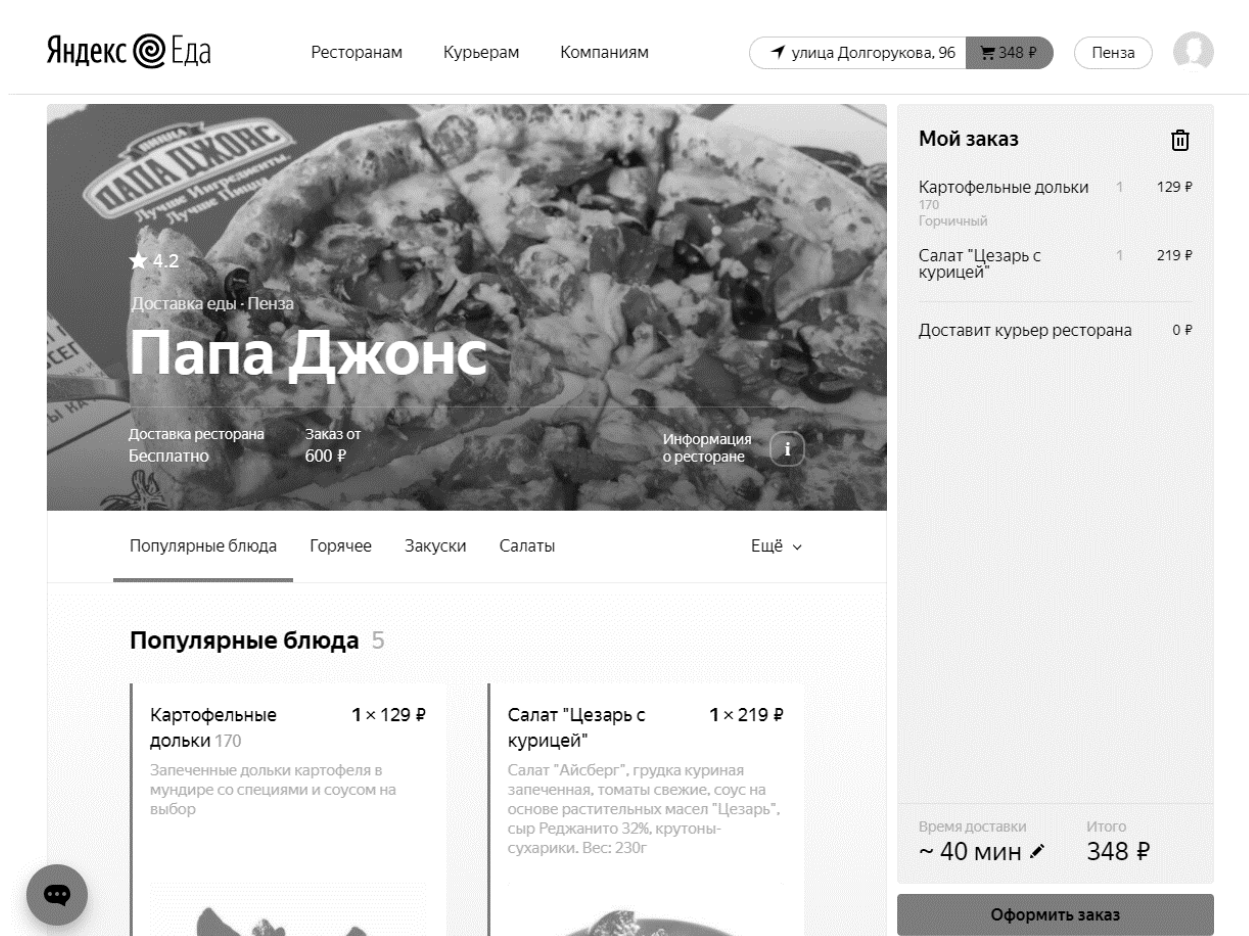


Рисунок 5 – Корзина

В качестве следующего аналога рассмотрим сервис Delivery Club. На главной странице расположен список ресторанов, из которых пользователь может заказать еду.

Кроме того, на главной странице расположен список продуктов, которые возможно заказать.

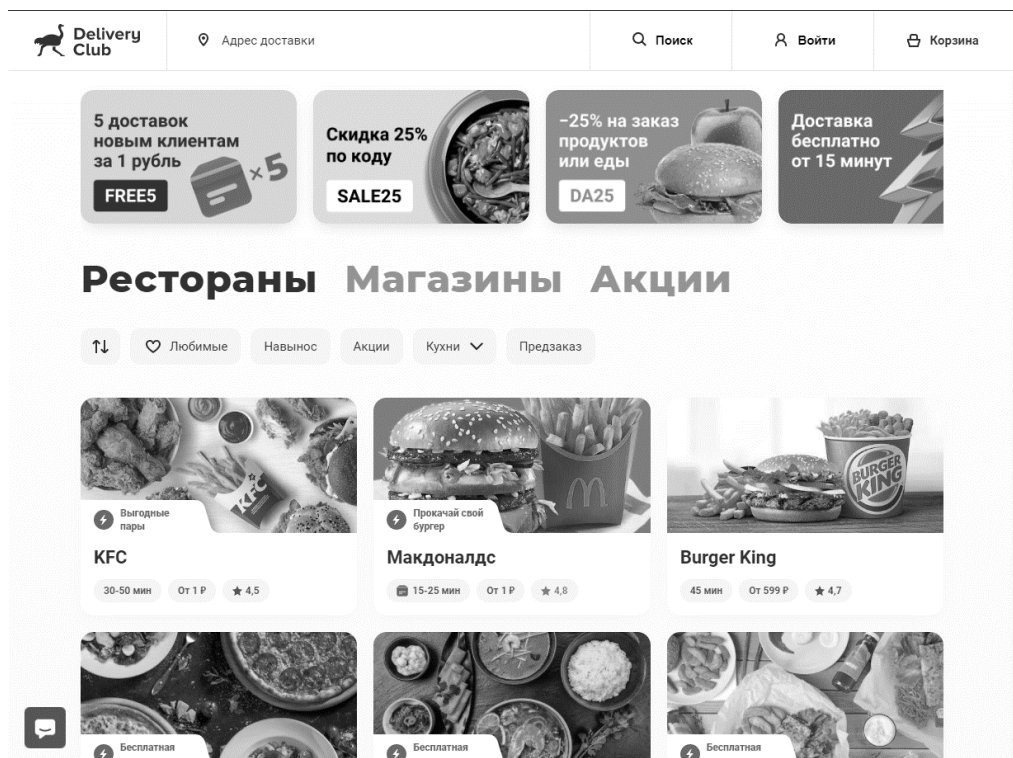


Рисунок 6 – Главная страница

В правом верхнем углу странице расположена кнопка входа в личный аккаунт и перехода в корзину. Во вкладке личного аккаунта расположены следующие пункты: личные данные, адреса доставки, мои заказы, мои скидки, **ВЫХОД**

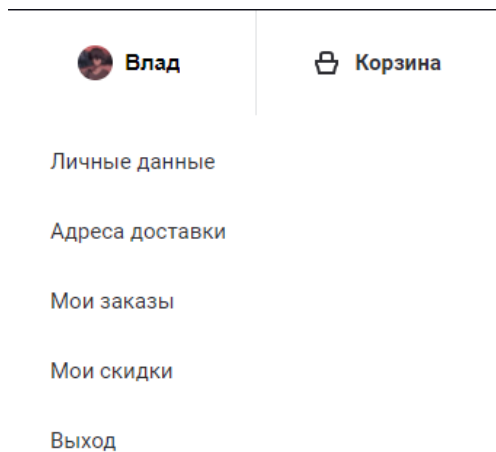


Рисунок 7 – Личный профиль

Для оформления заказа пользователю необходимо добавить выбранный им товар в корзину. После добавления товара, система высчитывает сумму заказа и отображает ее пользователю.

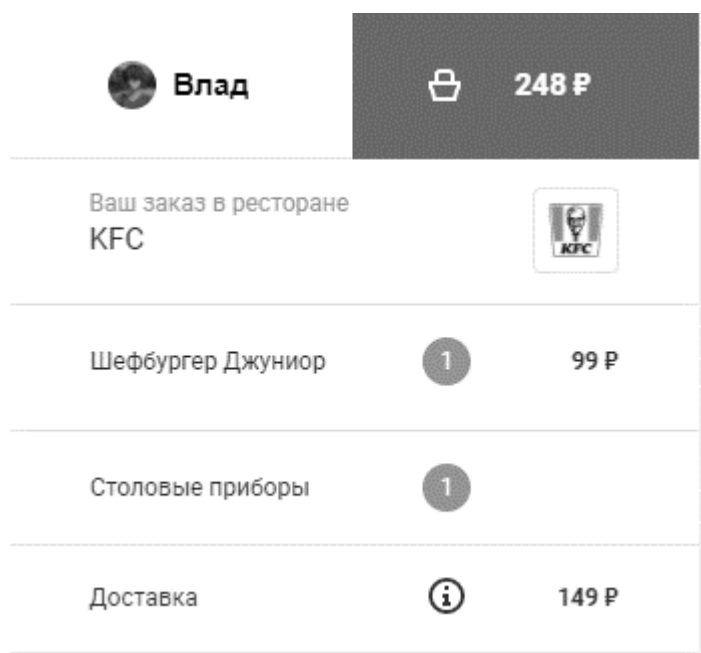


Рисунок 8 – Корзина

Исходя из предоставленного списка аналогов можно составить следующую таблицу:

Таблица 1 – Сравнение аналогов

	Широкий выбор ресторанов	История заказов	Добавление нескольких адресов доставки
Яндекс Еда	Да	Да	Да
Delivery Club	Да	Да	Да

На данной таблице представлено сравнение аналогов сервиса по доставке еды.

1.4 Выделение справочных и оперативных данных

Исходя из предоставленной информации об аналогах можно сделать вывод о справочных и оперативных данных, которые должны храниться в базе данных.

К справочным данным можно отнести:

- 1) Категории товаров
- 2) Товары

- 3) Список магазинов
- 4) Личные данных конкретного пользователя
- 5) Цены товаров
- 6) Описание товаров
- 7) Состав продуктов
- 8) Описание магазина

К оперативным данным можно отнести:

- 1) Сумма заказа
- 2) Количество позиций в корзине
- 3) Список пользователей
- 4) Текущий адрес доставки

Исходя из вышеперечисленных пунктов можно сделать вывод о том, что в базе данных, разрабатываемой в рамках данной курсовой работы, должны присутствовать все те данные, что присутствуют в базе данных аналогов.

1.5 Перечень хранимой информации: таблицы, поля, типы

Первым делом необходимо определить, какие таблицы будут присутствовать в проектируемой базе данных.

Для базы данных курсовой работы необходимы следующие таблицы:

- 1) Пользователи
- 2) Товары
- 3) Магазины
- 4) Корзина

В таблице «Пользователи» должны присутствовать такие поля, как электронная почта, номер телефона, имя пользователя. В таблице «Товары» должны находиться все товары, которые пользователь может заказать, а также их описание и стоимость. В таблице «Магазины» нужно хранить список магазинов, из которых пользователь может выбрать товар для заказа. В таблице «Корзина» должны содержаться следующие пункты:

- 1) Количество позиций
- 2) Наименование товара
- 3) Имя пользователя
- 4) Общая стоимость

Количество позиций и общая стоимость заказа будут вычисляться и после будут добавлены в поля.

Таблица 2 – Предполагаемый список таблиц БД

Таблица «Пользователи» (users)			
Название столбца	Тип данных	Назначение	Примечание
Id	Число	Порядковый номер	
Name	Текст	Имя пользователя	Максимальная длина – 255 символов
Phone_number	Число	Номер телефона пользователя	
email	Текст	Электронная почта пользователя	Максимальная длина – 255 символов
Таблица «Товары» (products)			
Название столбца	Тип данных	Назначение	Примечание
Id	Число	Порядковый номер	
Name	Текст	Наименование товара	Максимальная длина – 255 символов
price	Число	Цена товара	
description	Текст	Описание товара	Максимальная длина – 255 символов

Таблица 2 (продолжение)

Таблица «Магазины» (shops)			
Название столбца	Тип данных	Назначение	Примечание
id	Число	Порядковый номер	
name	Текст	Название магазина	Максимальная длина – 255 символов
Таблица «Корзина» (orders)			
Название столбца	Тип данных	Назначение	Примечание
id	Число	Порядковый номер	
User_id	Число	Порядковый номер пользователя	
Product_id	Число	Порядковый номер товара	
Total_price	Число	Итоговая цена	

Исходя из вышеперечисленных таблиц можно сделать вывод о необходимых предполагаемых запросах к БД.

1.6 Необходимые предполагаемые запросы к БД

В качестве предполагаемых бизнес операций можно выделить следующее:

1) Составление чека:

- Запрос данных заказа:

- а) выбрать номер заказа;
- б) выбрать дату по коду заказа;
- в) выбрать товары из чека по коду заказа;

- Расчет итоговой суммы.

- Формирование отчета (чек).
 - Отправка копии отчета (чека) пользователю.
- 2) Покупка еды пользователем:
- Добавление товаров в корзину.
 - Расчет итоговой суммы.
 - Получение данных об оплате.
 - Получение данных об адресе доставки.
- 3) Вывод информации о пользователе:
- Получение кода пользователя.
 - Получение информации о пользователе:
 - а) выбрать имя пользователя;
 - б) выбрать номер телефона пользователя;
 - в) выбрать почту пользователя;
 - б) выбрать адрес пользователя;
 - Вывод информации.
- 4) Вывод информации на главную страницу о новинках
- Получение кода товара.
 - Получение информации о товаре:
 - а) выбрать название товара;
 - б) выбрать описание товара;
 - в) выбрать состав продукта;
 - г) выбрать изображение товара;
 - д) выбрать цену товара;
 - Отобразить информацию на главной странице.

1.7 Выводы по разделу

В данном разделе были предоставлены сведения о предметной области, выделены справочные и оперативные данные, был предоставлен список предполагаемых таблиц и были написаны предполагаемые запросы к БД.

2. МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

2.1 Общие сведения

В данной главе будут выделены сущности, атрибуты, ключи и связи для каждой таблицы, будет проведена подготовка для переноса диаграммы сущность-связь в ЕА к переносу на целевую и будет создана модель базы данных с помощью средств Enterprise Architect. Enterprise Architect (EA) – CASE-инструмент для проектирования и конструирования программного обеспечения.

2.2 Выделение сущностей, атрибутов, ключей, связей

Исходя из проанализированных данных в первой главе, можно сделать вывод о содержании базы данных, а именно какие данные будут храниться в базе данных и какого типа они будут. Ниже представлены таблицы с данными и их типами.

Таблица 3 – Список таблиц БД

Таблица «Товары» (products)	
Поля	Типы
Id	Bigint
Name	Varchar(255)
Price	Bigint
Description	Varchar(255)
Shop_id	bigint
Таблица «Магазины» (shops)	
Id	Bigint
name	Varchar(255)
Связующая таблица «Магазины и товары» (ShopProduct)	
Id	Bigint
Product_id	Bigint
Shop_id	bigint
Таблица «Список заказов» (Order_list)	

Таблица 3 (продолжение)

Поля	Типы
Id	Bigint
Order_id	Bigint
ShopProduct_id	bigint
Таблица «Корзина» (orders)	
Id	Bigint
User_id	Bigint
Product_id	Bigint
Total_price	bigint
Таблица «Пользователи» (users)	
Id	Bigint
Name	Varchar(255)
Phone_number	Varchar(255)
email	Varchar(255)
Таблица «Отзыв» (review)	
Id	Bigint
User_id	Bigint
Text_review	Varchar(255)
Таблица «Пользовательские данные» (user_data)	
Id	Bigint
Address	Varchar(255)
Login	Bigint
Password	Bigint
Card_number	Bigint
User_id	bigint
Таблица «Чек» (receipt)	
Id	Bigint
User_id	Bigint
Order_id	bigint
Таблица «Статус заказа» (state_order)	
Id	Bigint
State	Varchar(255)
Order_id	Bigint
User_id	bigint

Исходя из данных, предоставленных в таблице 3 можно начать проектирование диаграммы сущность-связь.

2.3 Проектирование БД и перенос ее на СУБД.

После того, как были определены поля таблиц и их типы, было необходимо смоделировать диаграмму сущность-связь с помощью ЕА (См. Приложение А).

После создание диаграммы сущность-связь и верной расстановки связей средствами ЕА, необходимо перенести данную диаграмму на целевую СУБД. Для этого нужно также воспользоваться встроенными функциями ЕА.

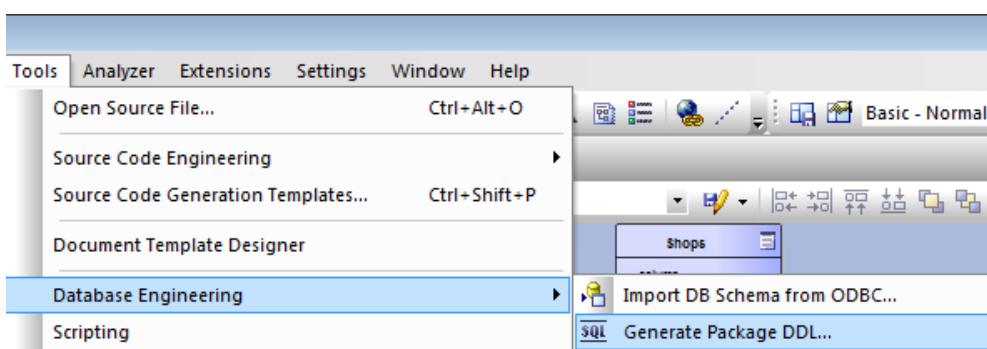


Рисунок 8 – Генерация SQL-кода

На рисунке показано, какими функциями ЕА необходимо воспользоваться для генерации SQL-кода [1].

Для успешной генерации SQL-кода необходимо выставить правильные настройки в выплывающем окне.

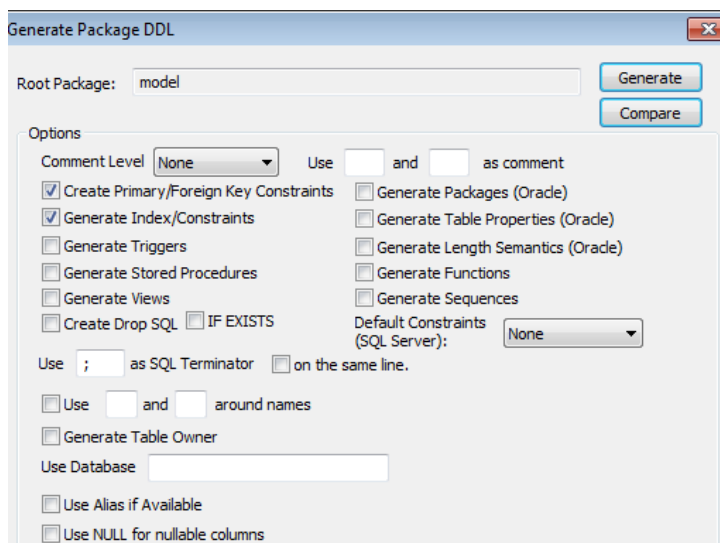


Рисунок 10 – Настройки для генерации кода

На данном рисунке показаны настройки, выставленные для успешной генерации SQL-кода

После проделанных действий ЕА сгенерирует SQL-код, который впоследствии можно использовать для переноса диаграммы сущность-связь на целевую СУБД (См. приложение Б).

2.4 Создание БД в PostgreSQL. Перенос схемы из ЕА.

Для создания базы данных необходимо установить pgAdmin. pgAdmin — это платформа с открытым исходным кодом для администрирования и разработки для PostgreSQL и связанных с ней систем управления базами данных. Чтобы создать БД воспользуемся средствами pgAdmin [12].

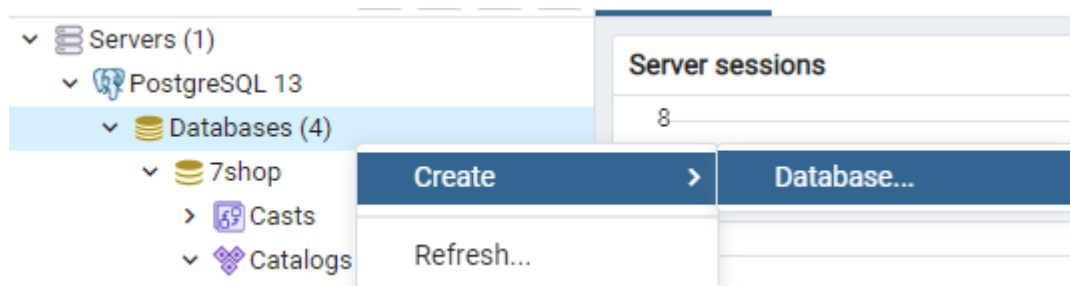


Рисунок 11 – Создание базы данных

На рисунке показано, как была создана база данных с помощью pgAdmin. Для переноса схемы из ЕА нужно установить DataGrip. Программное обеспечение JetBrains DataGrip представляет собой IDE-инструмент для работы с базами данных MySQL, PostgreSQL, Oracle, SQL Server, Sybase, DB2, SQLite, HyperSQL, Apache Derby и H2. После установки, необходимо подключиться к своему локальному серверу [15].

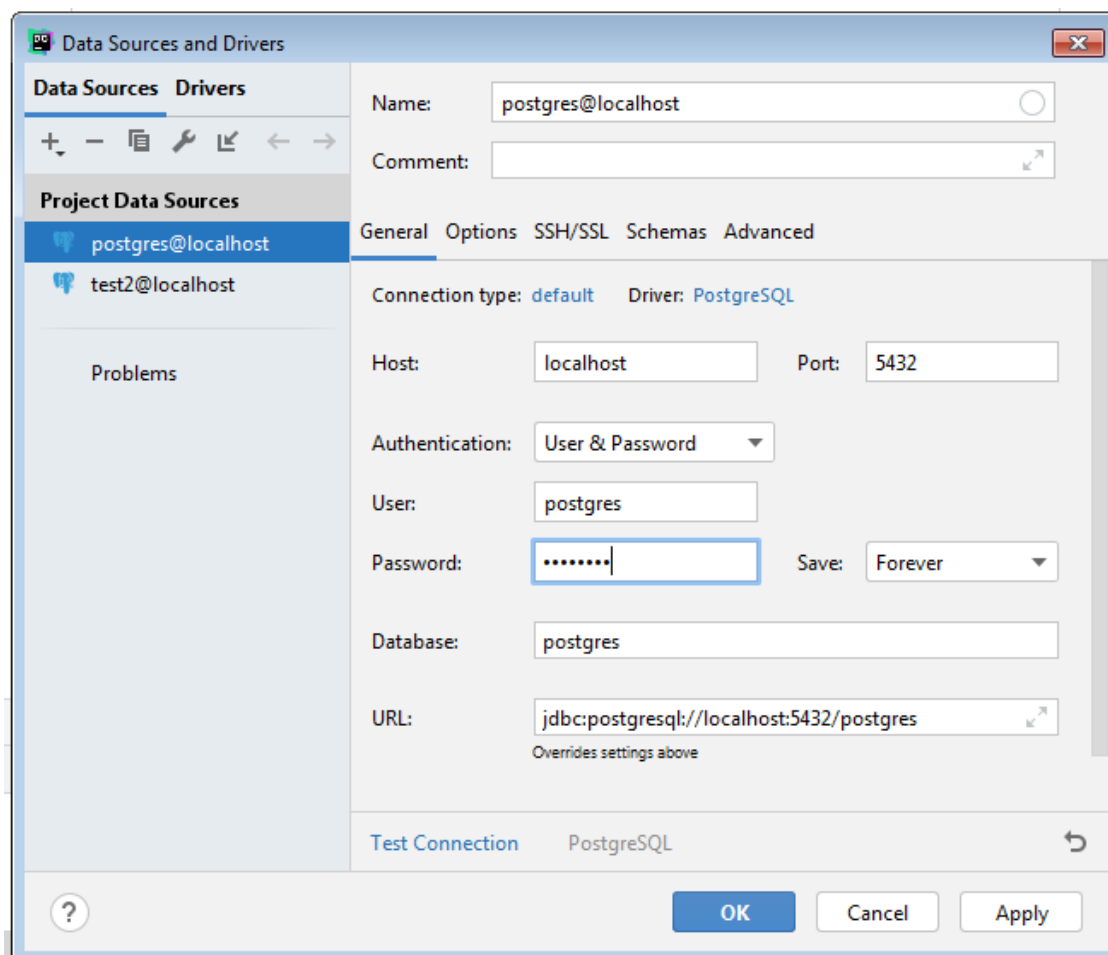


Рисунок 12 – Подключение к локальному серверу

Из рисунка видно, что для подключения к серверу необходимо ввести логин и пароль, а также указать куда именно мы подключаемся.

После того, как подключение к серверу прошло успешно, необходимо перенести схему из ЕА на СУБД. Для этого нужно воспользоваться консолью в DataGrip.

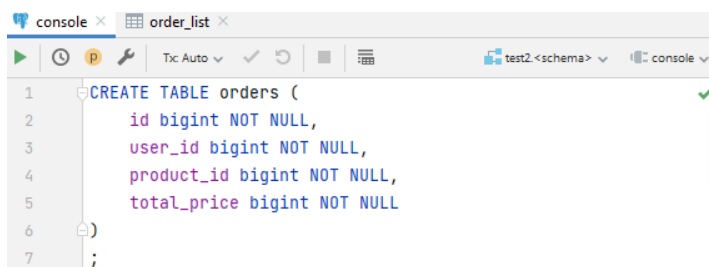


Рисунок 13 – Перенос схемы на СУБД

После того, как сгенерированный средствами ЕА SQL-код был введен в консоль, все таблицы были перенесены на СУБД.

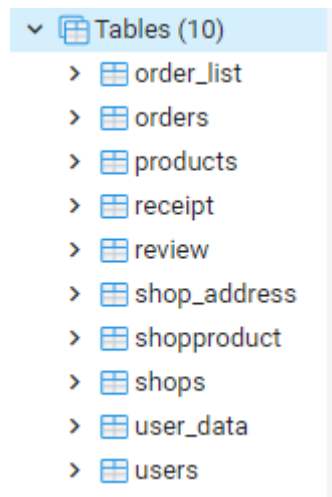


Рисунок 14 – Таблицы

2.5 Создание диаграммы средствами DataGrip.

После успешного переноса таблиц, необходимо создать схему средствами DataGrip.

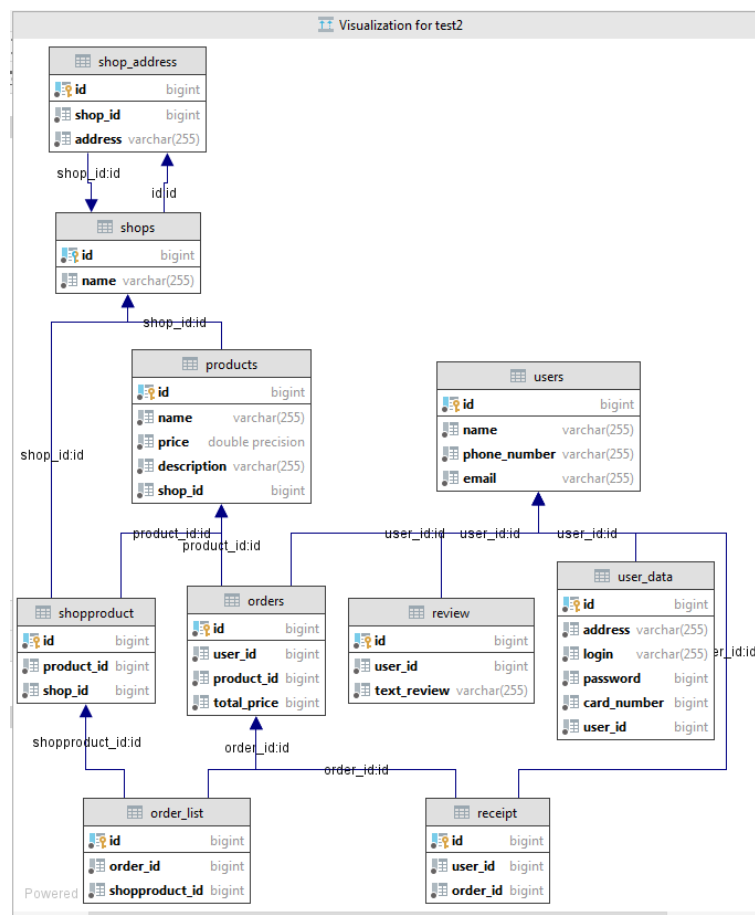
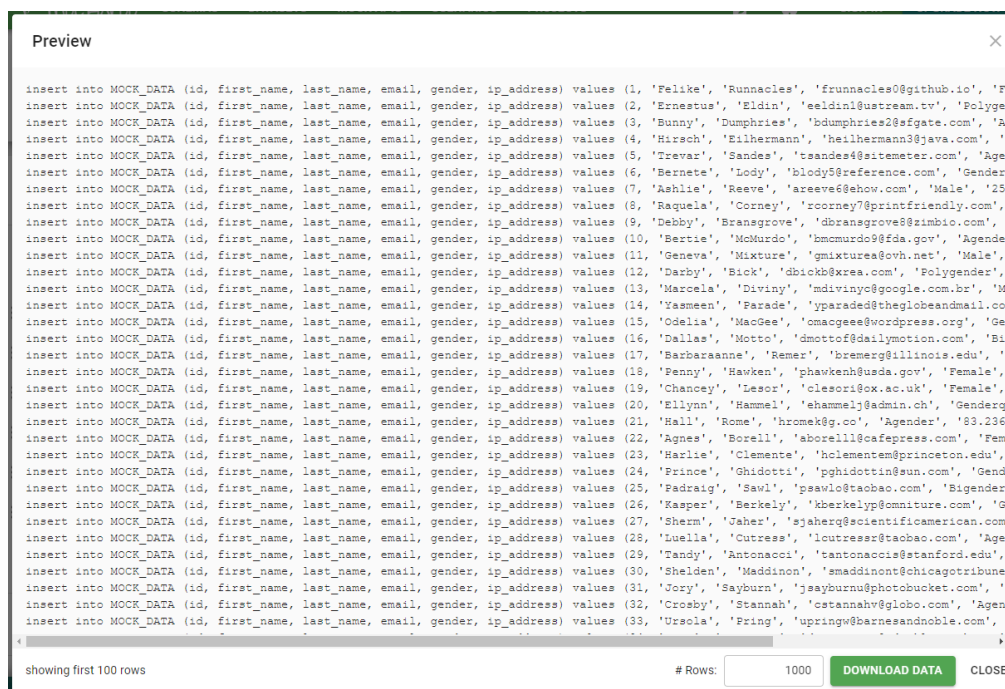


Рисунок 15 – Диаграмма DataGrip

Исходя из рисунка можно сделать вывод, что перенос таблиц прошел успешно.

2.6 Начальное заполнение БД.

Для заполнения БД тестовыми данными было принято решение воспользоваться средствами mockaroo. Данный интернет ресурс позволяет сгенерировать SQL-запросы для заполнения БД случайными данными.



```
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (1, 'Felike', 'Runnacles', 'frunnacles0@github.io', 'F',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (2, 'Ernestus', 'Eldin', 'eeldin1@ustream.tv', 'Polyge
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (3, 'Bunny', 'Dumphries', 'bdumphries2@sfgate.com', 'A
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (4, 'Hirsch', 'Ellhermann', 'hellhermann3@java.com', 'A
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (5, 'Trevor', 'Sandee', 'tsandee4@siteimeter.com', 'Age
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (6, 'Bernete', 'Lody', 'blody5@reference.com', 'Gende
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (7, 'Ashlie', 'Reeve', 'areeve6@show.com', 'Male', '25
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (8, 'Raquela', 'Corney', 'rcoorney7@printfriendly.com',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (9, 'Debby', 'Bransgrove', 'dbransgrove8@zimbio.com',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (10, 'Bertie', 'McMurdo', 'bmcmurdo9@fda.gov', 'Agende
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (11, 'Geneva', 'Mixture', 'gmixturea@ovh.net', 'Male',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (12, 'Darby', 'Bick', 'dbickb@xrea.com', 'Polygender',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (13, 'Marcela', 'Diviny', 'mdiviny@google.com.br', 'Mi
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (14, 'Yasmeen', 'Parade', 'yparaded@theglobeandmail.co
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (15, 'Odelia', 'MacGee', 'omacgeea@wordpress.org', 'Ge
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (16, 'Dallas', 'Motto', 'dmottof@dailymotion.com', 'Bi
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (17, 'Barbaraanne', 'Remer', 'bremerg@illinois.edu', 'I
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (18, 'Penny', 'Hawken', 'phawkenh@usda.gov', 'Female',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (19, 'Chancey', 'Lesor', 'clesori@ox.ac.uk', 'Female',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (20, 'Ellynn', 'Hammel', 'ehammel@admin.ch', 'Genderq
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (21, 'Hall', 'Rome', 'hrome@ig.co', 'Agender', '83.236
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (22, 'Agnes', 'Bozell', 'aborell1@cafepress.com', 'Fem
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (23, 'Harlie', 'Clemente', 'hclementem@princeton.edu',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (24, 'Prince', 'Ghidotti', 'pghidottin@sun.com', 'Gend
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (25, 'Padraig', 'Sawl', 'psawlo@taobao.com', 'Bigender
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (26, 'Kasper', 'Berkely', 'kberkelyp@omniure.com', 'G
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (27, 'Sherm', 'Jaher', 'sjaherg@scientificamerican.com
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (28, 'Luelia', 'Cutress', 'lcutress@taobao.com', 'Age
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (29, 'Tandy', 'Antonacci', 'tantonaccis@stanford.edu',
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (30, 'Shelden', 'Maddinon', 'smaddinont@chicagotribune
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (31, 'Jory', 'Sayburn', 'jsayburnu@photobucket.com', 'I
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (32, 'Crosby', 'Stannah', 'cstannahv@globo.com', 'Agene
insert into MOCK_DATA (id, first_name, last_name, email, gender, ip_address) values (33, 'Ursola', 'Pring', 'upringw@barnesandnoble.com',
```

Рисунок 16 – Генерация случайных данных

После того, как были указаны поля и типы данных для заполнения, данный ресурс генерирует SQL-запросы. Пример можно увидеть на рисунке. Подобным образом были заполнены все таблицы.

2.7 Вывод по разделу

В данном разделе были приведены общие сведения о моделировании предметной области, были выделены атрибуты, ключи, связи и сущности, была спроектирована диаграмма сущность-связь в ЕА, была проведена подготовка диаграммы сущность-связь для переноса на целевую СУБД, была создана БД и был проведен перенос диаграммы сущность-связь на целевую СУБД, было проведено заполнение таблиц тестовыми данными.

3. СОЗДАНИЕ И ЗАПУСК БАЗОВЫХ ЗАПРОСОВ SQL

3.1 Общие сведения

SQL или Structured Query Language (язык структурированных запросов) — язык программирования, предназначенный для управления данными в СУБД. Все современные СУБД поддерживают SQL.

В данной главе будут написаны SQL-запросы к имеющейся БД и предоставлены скриншоты результатов.

3.2 Базовые запросы

Для того, чтобы выбрать данные из таблицы, необходимо воспользоваться командой SELECT. Данная команда возвращает данные из таблицы по выбранному критерию. Для того, чтобы получить все данные из таблицы необходимо написать следующую команду:

```
SELECT * FROM users
```

Листинг 1 – Выборка всех данных из таблицы

После выполнения данной команды выводится следующий результат:

	🔍 id	🔍 name	🔍 phone_number	🔍 email
1	1	Roosevelt	224-570-4216	rmeeron0@prlog.org
2	2	Russ	805-805-7189	rroberson1@boston.com
3	3	Ephrayim	205-871-3465	eyitshak2@addtoany.com
4	4	Gavra	319-568-1680	geverley3@unblog.fr
5	5	Ann	348-365-6486	abalogun4@com.com
6	6	Wilbert	444-295-1609	wcazereau5@opera.com
7	7	Claudian	957-386-6707	cspear6@redcross.org
8	8	Arlin	753-254-2139	abaltrushaitis7@oracle.com

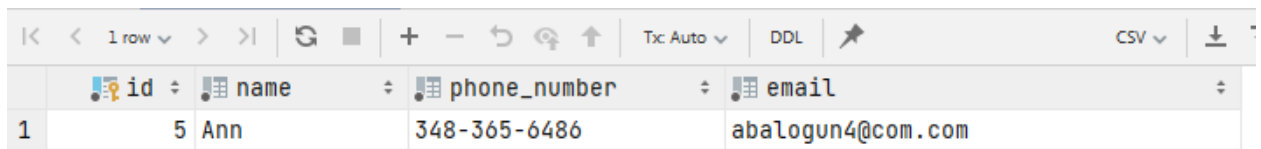
Рисунок 17 – Результат выполнения команды SELECT.

Для того, чтобы получить данные с определенным критерием, необходимо указать условие в запросе. Например, для того, чтобы получить данные пользователем с именем Ann необходимо написать следующий запрос:

```
SELECT * FROM users WHERE name = 'Ann'
```

Листинг 2 – Выборка данных пользователя Ann из таблицы.

После выполнения данной команды выводится следующий результат:



	id	name	phone_number	email
1	5	Ann	348-365-6486	abalogun4@com.com

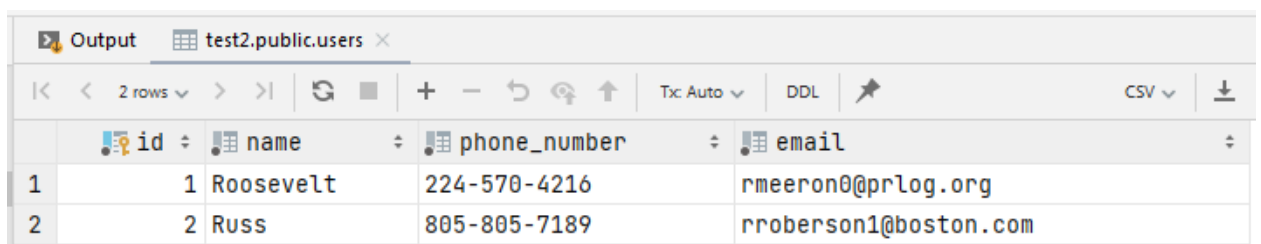
Рисунок 18 – Результат выборки данных с условием

Для того, чтобы указать несколько условий для выборки, необходимо воспользоваться операторами OR или AND. Для получения данных из таблицы, где id будет равен 1 или 2 необходимо написать следующий запрос:

```
SELECT * FROM users WHERE id = 1 OR id = 2
```

Листинг 3 – Выборка данных с условием OR

После выполнения команды выводится следующий результат:



	id	name	phone_number	email
1	1	Roosevelt	224-570-4216	rmeeron0@prlog.org
2	2	Russ	805-805-7189	rroberson1@boston.com

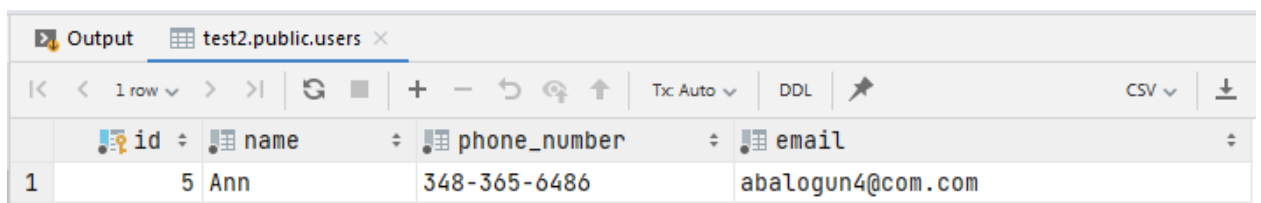
Рисунок 19 – Результат работы запроса с условием OR

Для получения данных из таблицы, где id будет равен 5 и имя пользователя будет равным Ann, необходимо написать следующий запрос:

```
SELECT * FROM users WHERE id = 5 AND name = 'Ann'
```

Листинг 4 – Выборка данных с условием AND

После выполнения команды выводится следующий результат:



	id	name	phone_number	email
1	5	Ann	348-365-6486	abalogun4@com.com

Рисунок 20 – Результат работы запроса с условием AND

Для добавления данных в таблицу необходимо использовать запрос INSERT. Например, для того, чтобы добавить в таблицу users нового пользователя с именем 'Vlad', телефонным номером '0' и почтой 'vlad@gmail.com', необходимо написать следующий запрос:

```
INSERT INTO users (id, name, phone_number, email) VALUES (101, 'Vlad', 0, 'vlad@gmail.com')
```

Листинг 5 – Запрос INSERT

Для проверки работоспособности запроса, необходимо посмотреть на таблицу до и после проведения теста.

WHERE		ORDER BY		
	id	name	phone_number	email
89	89	Robena	600-353-1890	rcanedo2g@foxnews.com
90	90	Kania	654-722-9038	kjickles2h@ifeng.com
91	91	Haven	887-584-3457	hshorten2i@msn.com
92	92	Kit	423-917-7514	klinzee2j@europa.eu
93	93	Pace	579-293-3951	pginglell2k@qq.com
94	94	Tandi	505-763-0985	tbiasi2l@examiner.com
95	95	Malissia	706-102-9784	mdanielsohn2m@tripadvis...
96	96	Tressa	990-826-7954	tcrannach2n@meetup.com
97	97	Creighton	549-570-8411	crabley2o@apple.com
98	98	Gilberte	743-768-4403	gpossa2p@opera.com
99	99	Shaylynn	664-365-6246	smcrinn2q@behance.net
100	100	Calypso	800-591-3943	clynett2r@cpanel.net

Рисунок 21 – Таблица users до запроса

На рисунке 21 показана таблица до проведения теста.

WHERE		ORDER BY		
	id	name	phone_number	email
90	90	Kania	654-722-9038	kjickles2h@ifeng.com
91	91	Haven	887-584-3457	hshorten2i@msn.com
92	92	Kit	423-917-7514	klinzee2j@europa.eu
93	93	Pace	579-293-3951	pginglell2k@qq.com
94	94	Tandi	505-763-0985	tbiasi2l@examiner.com
95	95	Malissia	706-102-9784	mdanielsohn2m@tripadviso...
96	96	Tressa	990-826-7954	tcrannach2n@meetup.com
97	97	Creighton	549-570-8411	crabley2o@apple.com
98	98	Gilberte	743-768-4403	gpossa2p@opera.com
99	99	Shaylynn	664-365-6246	smcrinn2q@behance.net
100	100	Calypso	800-591-3943	clynett2r@cpanel.net
101	101	Vlad	0	vlad@gmail.com

Рисунок 22 – Таблица users после запроса

Исходя из предоставленных рисунков можно сделать вывод о том, что запрос работает корректно.

Кроме того, используя INSERT можно добавлять сразу несколько строк в таблицу.

```
INSERT INTO users (id, name, phone_number, email) VALUES
(102, 'Andrey', 1, 'andrey@gmail.com'),
(103, 'Nikita', 2, 'nikita@gmail.com'),
(104, 'Roman', 3, 'roman@gmail.com')
```

Листинг 6 – Вставка нескольких строк

После выполнения команды выводится следующий результат:

WHERE		ORDER BY		
	id	name	phone_number	email
90	90	Kania	654-722-9038	kjickles2h@ifeng.com
91	91	Haven	887-584-3457	hshorten2i@msn.com
92	92	Kit	423-917-7514	klinzee2j@europa.eu
93	93	Pace	579-293-3951	pginglell2k@qq.com
94	94	Tandi	505-763-0985	tbiasi2l@examiner.com
95	95	Malissia	706-102-9784	mdanielsohn2m@tripadvis...
96	96	Tressa	990-826-7954	tcrannach2n@meetup.com
97	97	Creighton	549-570-8411	crabley2o@apple.com
98	98	Gilberte	743-768-4403	gpossa2p@opera.com
99	99	Shaylynn	664-365-6246	smcrinn2q@behance.net
100	100	Calypso	800-591-3943	clynett2r@cpanel.net
101	101	Vlad	0	vlad@gmail.com
102	102	Andrey	1	andrey@gmail.com
103	103	Nikita	2	nikita@gmail.com
104	104	Roman	3	roman@gmail.com

Рисунок 23 – Таблица users после запроса

Исходя из предоставленного рисунка можно сделать вывод о том, что запрос работает корректно. Для обновления данных в таблице используется команда UPDATE. Например, для того, чтобы изменить имя пользователя с id = 101 необходимо написать следующий запрос:

```
UPDATE users
SET name = 'Georg' WHERE id = 101;
```

Листинг 7 – Изменение поля в таблице

После выполнения команды выводится следующий результат:

WHERE		ORDER BY id			
	...	1	name	phone_number	email
90		90	Kania	654-722-9038	kjickles2h@ifeng.com
91		91	Haven	887-584-3457	hshorten2i@msn.com
92		92	Kit	423-917-7514	klinzee2j@europa.eu
93		93	Pace	579-293-3951	pginglell2k@qq.com
94		94	Tandi	505-763-0985	tbiasi2l@examiner.com
95		95	Malissia	706-102-9784	mdanielsohn2m@tripadvis...
96		96	Tressa	990-826-7954	tcrannach2n@meetup.com
97		97	Creighton	549-570-8411	crabley2o@apple.com
98		98	Gilberte	743-768-4403	gpossa2p@opera.com
99		99	Shaylynn	664-365-6246	smcrinn2q@behance.net
100		100	Calypso	800-591-3943	clynett2r@cpanel.net
101		101	Georg	0	vlad@gmail.com
102		102	Andrey	1	andrey@gmail.com
103		103	Nikita	2	nikita@gmail.com
104		104	Roman	3	roman@gmail.com

Рисунок 24 – Результат работы запроса

Как видно на рисунке, пользователю с id = 101 было изменено имя, следовательно, данный запрос работает корректно.

С помощью UPDATE можно изменять сразу все поля таблицы. Например, чтобы указать максимальную цену для всех товаров необходимо написать следующий запрос:

```
UPDATE products
SET price = (SELECT MAX(price)
FROM products
);
```

Листинг 8 – Запрос на обновление данных

После выполнения команды выводится следующий результат:

WHERE		ORDER BY id			
	1	name	price	description	
1	1	ARANESP	99.15	felis ut at dolor quis ...	
2	2	Levothyroxine Sodium	18.15	nisi nam ultrices liber...	
3	3	Acne/Pimple Control	50.15	tempus vivamus in felis...	
4	4	Kaleta	30.15	dui maecenas tristique ...	

Рисунок 25 – Таблица до запроса

На данном рисунке представлены данные таблицы до запроса.

	1	name	price	description
1	1	ARANESP	99.15	felis ut at dolor quis ...
2	2	Levothyroxine Sodium	99.15	nisi nam ultrices liber...
3	3	Acne/Pimple Control	99.15	tempus vivamus in felis...
4	4	Kaletra	99.15	dui maecenas tristique ...

Рисунок 26 – Таблица после запроса

Исходя из рисунков можно сделать вывод, что запрос работает корректно. Для удаления строк из таблицы необходимо воспользоваться оператором DELETE. Например, для удаления пользователя с id = 101 необходимо написать следующий запрос:

```
DELETE FROM users WHERE id = 101
```

Листинг 9 – Запрос на удаление данных

После выполнения команды выводится следующий результат:

	1	name	phone_number	email
90	90	Kania	654-722-9038	kjickles2h@ifeng.com
91	91	Haven	887-584-3457	hshorten2i@msn.com
92	92	Kit	423-917-7514	klinzee2j@europa.eu
93	93	Pace	579-293-3951	pginglell2k@qq.com
94	94	Tandi	505-763-0985	tbiasi2l@examiner.com
95	95	Malissia	706-102-9784	mdanielsohn2m@tripadvis...
96	96	Tressa	990-826-7954	tcrannach2n@meetup.com
97	97	Creighton	549-570-8411	crabley2o@apple.com
98	98	Gilberte	743-768-4403	gpossa2p@opera.com
99	99	Shaylynn	664-365-6246	smcrinn2q@behance.net
100	100	Calypso	800-591-3943	clynett2r@cpanel.net
101	101	Georg	0	vlad@gmail.com
102	102	Andrey	1	andrey@gmail.com
103	103	Nikita	2	nikita@gmail.com
104	104	Roman	3	roman@gmail.com

Рисунок 27 – Таблица users до запроса

После выполнения запроса таблица приняла следующий вид:

103 rows					CSV					users				
WHERE					ORDER BY id									
		name		phone_number		email								
90	90	Kania		654-722-9038		kjickles2h@ifeng.com								
91	91	Haven		887-584-3457		hshorten2i@msn.com								
92	92	Kit		423-917-7514		klinzee2j@europa.eu								
93	93	Pace		579-293-3951		pginglell2k@qq.com								
94	94	Tandi		505-763-0985		tbiasi2l@examiner.com								
95	95	Malissia		706-102-9784		mdanielsohn2m@tripadvis...								
96	96	Tressa		990-826-7954		tcrannach2n@meetup.com								
97	97	Creighton		549-570-8411		crabley2o@apple.com								
98	98	Gilberte		743-768-4403		gpossa2p@opera.com								
99	99	Shaylynn		664-365-6246		smcrinn2q@behance.net								
100	100	Calypso		800-591-3943		clynett2r@cpanel.net								
101	102	Andrey		1		andrey@gmail.com								
102	103	Nikita		2		nikita@gmail.com								
103	104	Roman		3		roman@gmail.com								

Рисунок 28 – Таблица users после запроса

Как видно из рисунков, запрос работает корректно.

Для удаления полей, где id будет больше, чем 100, необходимо написать следующий запрос:

```
DELETE FROM users WHERE id > 100
```

Листинг 10 – Удаление данных из таблицы

После выполнения команды выводится следующий результат:

WHERE					ORDER BY id									
		name		phone_number		email								
89	89	Robena		600-353-1890		rcanedo2g@foxnews.com								
90	90	Kania		654-722-9038		kjickles2h@ifeng.com								
91	91	Haven		887-584-3457		hshorten2i@msn.com								
92	92	Kit		423-917-7514		klinzee2j@europa.eu								
93	93	Pace		579-293-3951		pginglell2k@qq.com								
94	94	Tandi		505-763-0985		tbiasi2l@examiner.com								
95	95	Malissia		706-102-9784		mdanielsohn2m@tripadvis...								
96	96	Tressa		990-826-7954		tcrannach2n@meetup.com								
97	97	Creighton		549-570-8411		crabley2o@apple.com								
98	98	Gilberte		743-768-4403		gpossa2p@opera.com								
99	99	Shaylynn		664-365-6246		smcrinn2q@behance.net								
100	100	Calypso		800-591-3943		clynett2r@cpanel.net								

Рисунок 29 – Таблица users после запроса

Как видно из рисунка, запрос работает корректно.

3.3 Выводы по разделу

В данном разделе были приведены общие сведения о создании и запуске запросов SQL, были написаны запросы на выборку, вставку, обновление и удаление.

4. СОЗДАНИЕ И ЗАПУСК ПРОДВИНУТЫХ ЗАПРОСОВ

4.1. Общие сведения

Продвинутые SQL запросы – это запросы, которые работают с двумя или более таблицами. Данные запросы используются как для более точного исполнения запроса, так и для масштабной выборки данных из нескольких таблиц.

При использовании их в больших БД и не только, значительно уменьшает время на поиск данных среди многочисленных таблиц и многочисленных строк.

В данном разделе будут реализованы и описаны сложные запросы к базе данных.

4.2 Продвинутые запросы

Чтобы получить данные из нескольких таблиц, например, имя пользователя из таблицы users и его номер банковской карты из таблицы user_data, необходимо написать следующий запрос:

```
SELECT users.name, user_data.card_number FROM users, user_data
```

Листинг 11 – Пример запроса для получение данных из нескольких таблиц.

После выполнения команды выводится следующий результат:

	name	card_number
1	Roosevelt	0
2	Roosevelt	4
3	Roosevelt	2814
4	Roosevelt	74736
5	Roosevelt	6595
6	Roosevelt	68
7	Roosevelt	1
8	Roosevelt	43

Рисунок 30 – Данные, полученные запросом

Чтобы получить имя пользователя из таблицы users и адрес из таблицы user_data, где user.id = user_data.id, необходимо написать следующий запрос:

```
SELECT users.name, user_data.address FROM users, user_data WHERE users.id =
user_data.id
```

Листинг 12 – Запрос с условием

После выполнения команды выводится следующий результат:

	name	address
1	Roosevelt	510 Utah Circle
2	Russ	061 Talisman Junction
3	Ephrayim	1372 Linden Plaza
4	Gavra	5564 Sheridan Junction
5	Ann	17 Glendale Lane
6	Wilbert	5027 Oakridge Alley
7	Claudian	5 Daystar Court
8	Arlin	33414 Maple Wood Terrace

Рисунок 31 – Результат выполнения запроса

Для создания таблицы необходимо воспользоваться оператором CREATE. Например, для создания таблицы ban_list с полями id, name, description, необходимо написать следующий запрос:

```
CREATE TABLE ban_list (
id BIGINT,
name VARCHAR(255),
description VARCHAR(255))
```

Листинг 13 – Запрос на создание таблицы

После выполнения команды выводится следующий результат:

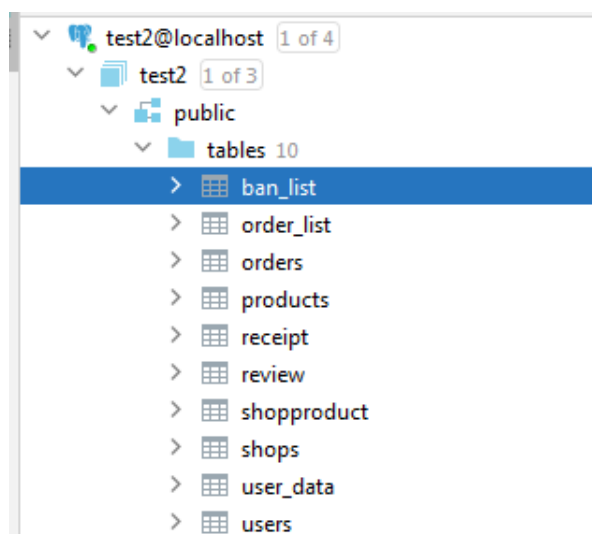


Рисунок 32 – Результат работы запроса

Исходя из рисунка, можно сделать вывод, что запрос работает корректно.

Для удаления таблицы необходимо использовать оператор DROP TABLE. Например, для удаления таблицы ban_list, необходимо написать следующий запрос:

```
DROP TABLE ban_list
```

Листинг 14 – Запрос на удаление таблицы

После выполнения команды выводится следующий результат:

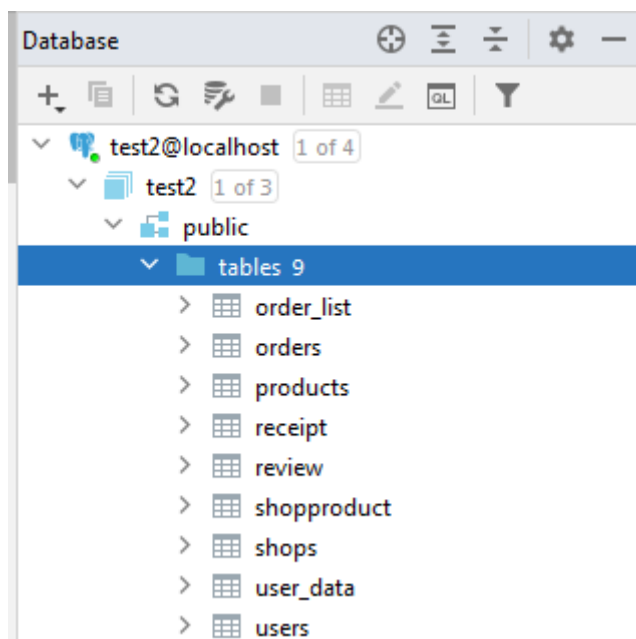


Рисунок 33 – Результат работы запроса

Исходя из рисунка, можно сделать вывод, что запрос работает корректно.

Для добавления новой строки в уже существующую таблицу, необходимо воспользоваться оператором ALTER TABLE. Например, для того, чтобы добавить новое поле last_name в таблицу users, необходимо написать следующий запрос:

```
ALTER TABLE users ADD last_name VARCHAR(255)
```

Листинг 15 – Запрос на добавление нового поля в таблицу

После выполнения команды выводится следующий результат:

WHERE		ORDER BY id				
	...	1	name	phone_number	email	
89		89	Robena	600-353-1890	rcanedo2g@foxnews.com	
90		90	Kania	654-722-9038	kjickles2h@ifeng.com	
91		91	Haven	887-584-3457	hshorten2i@msn.com	
92		92	Kit	423-917-7514	klinzee2j@europa.eu	
93		93	Pace	579-293-3951	pginglell2k@qq.com	
94		94	Tandi	505-763-0985	tbiasi2l@examiner.com	
95		95	Malissia	706-102-9784	mdanielsohn2m@tripadvis...	
96		96	Tressa	990-826-7954	tcrannach2n@meetup.com	
97		97	Creighton	549-570-8411	crabley2o@apple.com	
98		98	Gilberte	743-768-4403	gpossa2p@opera.com	
99		99	Shaylynn	664-365-6246	smcrinn2q@behance.net	
100		100	Calypso	800-591-3943	clynett2r@cpanel.net	

Рисунок 34 – Таблица users до запроса

После запроса таблица users принимает следующий вид:

WHERE		ORDER BY id				
			phone_number	email	last_name	
89			600-353-1890	rcanedo2g@foxnews.com	<null>	
90			654-722-9038	kjickles2h@ifeng.com	<null>	
91			887-584-3457	hshorten2i@msn.com	<null>	
92			423-917-7514	klinzee2j@europa.eu	<null>	
93			579-293-3951	pginglell2k@qq.com	<null>	
94			505-763-0985	tbiasi2l@examiner.com	<null>	
95	ia		706-102-9784	mdanielsohn2m@tripadvis...	<null>	
96			990-826-7954	tcrannach2n@meetup.com	<null>	
97	ton		549-570-8411	crabley2o@apple.com	<null>	
98	te		743-768-4403	gpossa2p@opera.com	<null>	
99	nn		664-365-6246	smcrinn2q@behance.net	<null>	
100	o		800-591-3943	clynett2r@cpanel.net	<null>	

Рисунок 35 – Результат работы запроса

Исходя из рисунка можно сделать вывод о том, что запрос работает корректно. Для создания индекса в таблицах необходимо использовать оператор CREATE INDEX.

Индексы используются для быстрого извлечения данных из базы данных. Пользователи не видят индексы, они просто используются для ускорения поисков/запросов.

Например, для создания индекса для поля name таблицы users, необходимо написать следующий запрос:

```
CREATE INDEX name_index  
ON users (name);
```

Листинг 16 – Запрос на создание индекса

Для изменения определения индекса необходимо написать следующий запрос:

```
ALTER INDEX IF EXISTS name_index RENAME TO name_index_test
```

Листинг 17 – Изменение индекса

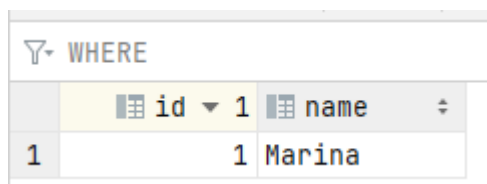
Для удаления индекса необходимо написать следующий запрос:

```
DROP INDEX name_index_test;
```

Листинг 18 – Запрос на удаление индекса

Для копирования данных из одной таблицы в другую, необходимо воспользоваться оператором SELECT INTO.

Для проверки работоспособности запроса была создана тестовая таблица users_test с полями id и name.



	id	name
1	1	Marina

Рисунок 36 – Тестовая таблица

Для копирования данных из тестовой таблицы в таблицу users, необходимо написать следующий запрос:

```
SELECT * INTO users_test_data1 FROM users
```

Листинг 19 – Запрос на копирование данных

	id	name	phone_number	email
1	1	Roosevelt	224-570-4216	rmeeron0@prlog.org
2	2	Russ	805-805-7189	rroberson1@boston.com
3	3	Ephrayim	205-871-3465	eyitshak2@addtoany.com
4	4	Gavra	319-568-1680	geverley3@unblog.fr
5	5	Ann	348-365-6486	abalogun4@com.com
6	6	Wilbert	444-295-1609	wcazereau5@opera.com
7	7	Claudian	957-386-6707	cspear6@redcross.org
8	8	Arlin	753-254-2139	abaltrushaitis7@oracle.co
9	9	Eldin	724-347-8302	ecorrison8@tinyurl.com
10	10	Adolph	130-436-8968	aship9@unblog.fr
11	11	Ben	700-397-4702	bgraynea@eepurl.com
12	12	Carrie	445-834-3281	cotteeb@miitbeian.gov.cn
13	13	Roderick	931-587-2873	rallsuppc@webeden.co.uk
14	14	Findley	698-566-7958	fsautterd@ca.gov
15	15	Micheil	166-272-3447	mkeatche@apple.com
16	16	Mark	160-349-0413	mbottellf@xing.com
17	17	Rickie	367-183-1686	rdrabbleg@nifty.com

Рисунок 37 – Результат выполнения запроса

Исходя из рисунка можно сделать вывод, что запрос работает корректно.

4.3 Выводы по разделу

В данном разделе были предоставлены общие сведения о продвинутых запросах, реализованы запросы на выбор данных из нескольких таблиц, на создание, изменение и удаление индекса, на изменение поля таблицы и на копирование данных из таблицы.

Заключение

В первом разделе был проведен анализ предметной области, приведены общие сведения о теме курсовой работы, были выделены справочные и оперативные данные путем сравнения аналогов, был представлен перечень хранимой информации (таблицы, поля, типы) и были представлены предполагаемые запросы к базе данных

Во втором разделе были представлены общие сведения о моделировании предметной области, были выделены сущности, атрибуты, ключи и связи, было проведено проектирование диаграммы сущность-связь средствами ЕА, была проведена подготовка диаграммы сущность-связь для переноса на целевую СУБД, была создана база данных, было проведено заполнение базы данных тестовыми данными.

В третьем разделе были приведены общие сведения о создании и запуске базовых запросов SQL, были написаны такие запросы, как select, update, insert, delete, были проведены тесты базовых запросов и представлены результаты тестирования.

В четвертом разделе были приведены общие сведения о создании и запуске продвинутых запросов SQL, были написаны такие запросы, как multi table select, create table, drop table, alter table, create index, alter index, drop index, select into, были проведены тесты и представлены их результаты.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Язык запросов SQL [Электронный ресурс]. – Режим доступа: <https://htmlacademy.ru/tutorial/php/sql>, Загл. С экрана. Дата обращения 16.03.2021
2. Основные команды SQL [Электронный ресурс]. – Режим доступа: <https://tproger.ru/translations/sql-recap/>, Загл. С экрана. Дата обращения 16.04.2021
3. Назначение языка SQL [Электронный ресурс]. – Режим доступа: <https://oracle-patches.com/db/sql/4039>, Загл. С экрана. Дата обращения 16.04.2021
4. Основы языка SQL [Электронный ресурс]. – Режим доступа: <https://www.opennet.ru/docs/RUS/sql/>, Загл. С экрана. Дата обращения 16.04.2021
5. Введение в SQL [Электронный ресурс]. – Режим доступа: <https://sql-language.ru/sqlinto.html>, Загл. С экрана. Дата обращения 16.04.2021
6. Язык SQL [Электронный ресурс]. – Режим доступа: http://www.mstu.edu.ru/study/materials/zelenkov/ch_4_7.html, Загл. С экрана. Дата обращения 16.04.2021
7. Обзор основных SQL запросов [Электронный ресурс]. – Режим доступа: <https://itvdn.com/ru/blog/article/m-sql>, Загл. С экрана. Дата обращения 15.04.2021
8. Работа с PostgreSQL [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/learn-postgresql>, Загл. С экрана. Дата обращения 15.04.2021
9. Начало работы с PostgreSQL [Электронный ресурс]. – Режим доступа: <https://eax.me/postgresql-install/>, Загл. С экрана. Дата обращения 17.04.2021
10. PosgresPro [Электронный ресурс]. – Режим доступа: <https://postgrespro.ru/docs/postgresql/9.5/app-psql>, Загл. С экрана. Дата обращения 18.04.2021

11. Работы с PostgreSQL в Python [Электронный ресурс]. – Режим доступа: <https://dev-gang.ru/article/rabota-s-postgresql-v-python-xn8721sq0g/>, Загл. С экрана. Дата обращения 20.04.2021

12. Установка и базовая настройка PostgreSQL [Электронный ресурс]. – Режим доступа: <https://winitpro.ru/index.php/2019/10/25/ustanovka-nastrojka-postgresql-v-windows/>, Загл. С экрана. Дата обращения 20.04.2021

13. PostgreSQL или MySQL [Электронный ресурс]. – Режим доступа: <https://mcs.mail.ru/blog/postgresql-ili-mysql-kakaya-iz-etih-relyacionnyh-subd>, Загл. С экрана. Дата обращения 14.05.2021

14. Установка PostgreSQL [Электронный ресурс]. – Режим доступа: <https://java-course.ru/begin/postgresql/>, Загл. С экрана. Дата обращения 16.01.2021

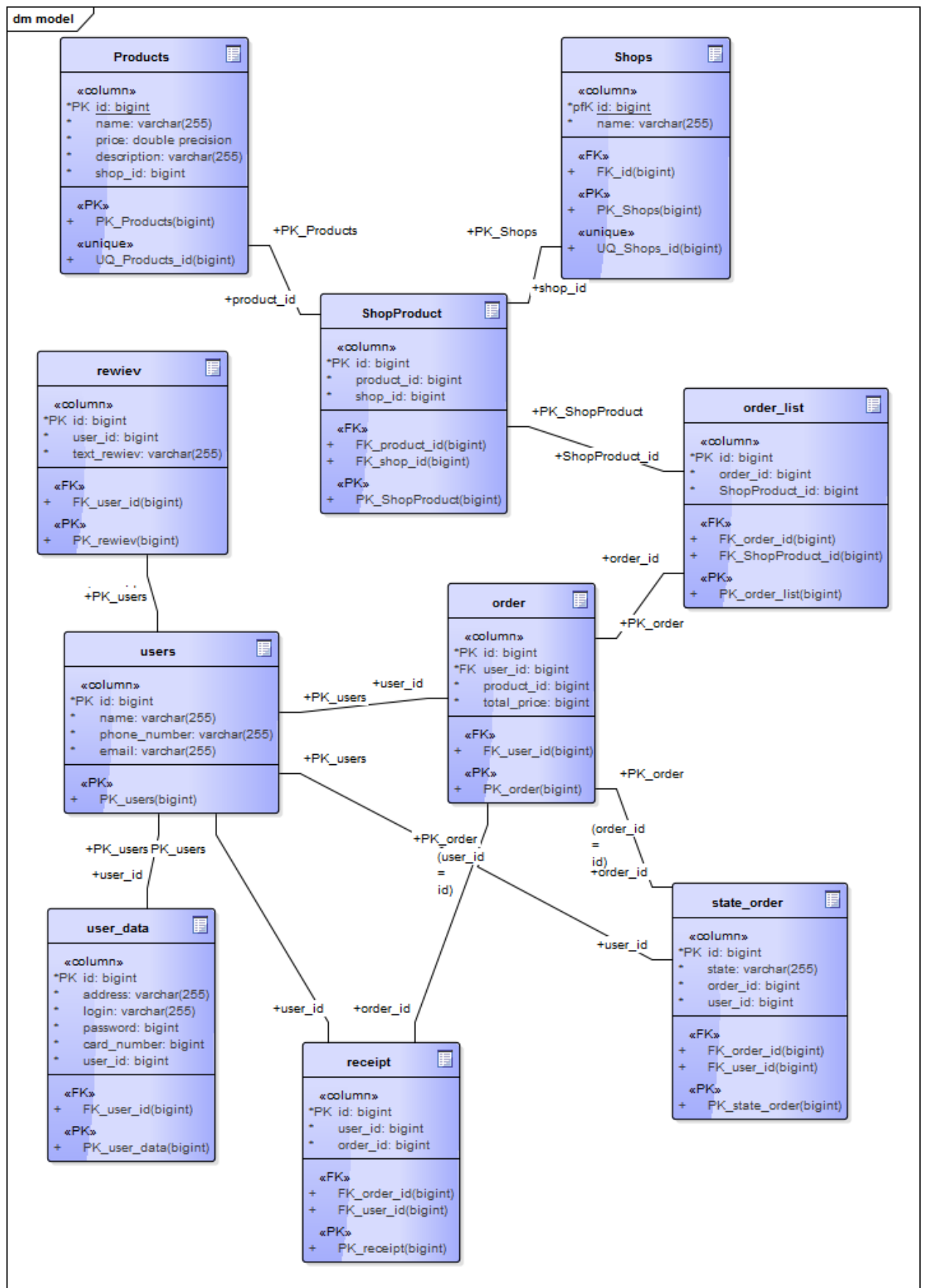
15. Что умеет DataGrip [Электронный ресурс]. – Режим доступа: <https://www.jetbrains.com/ru-ru/datagrip/features/>, Загл. С экрана. Дата обращения 20.04.2021

16. Как искать в DataGrip [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/JetBrains/blog/327006/>, Загл. С экрана. Дата обращения 20.04.2021

17. Переключение базы данных PostgreSQL [Электронный ресурс]. – Режим доступа: <https://coderoad.ru/48891255/Переключение-базы-данных-PostgreSQL-или-схемы-в-DataGrip-JetBrains>, Загл. С экрана. Дата обращения 20.04.2021

18. DataGrip [Электронный ресурс]. – Режим доступа: <http://cyclowiki.org/wiki/DataGrip>, Загл. С экрана. Дата обращения 20.04.2021

ПРИЛОЖЕНИЕ А. СХЕМА БАЗЫ ДАННЫХ



ПРИЛОЖЕНИЕ Б. SQL КОД БАЗЫ ДАННЫХ

```
CREATE TABLE orders (  
    id bigint NOT NULL,  
    user_id bigint NOT NULL,  
    product_id bigint NOT NULL,  
    total_price bigint NOT NULL  
)  
;  
CREATE TABLE order_list (  
    id bigint NOT NULL,  
    order_id bigint NOT NULL,  
    ShopProduct_id bigint NOT NULL  
)  
;  
CREATE TABLE Products (  
    id bigint NOT NULL,  
    name varchar(255) NOT NULL,  
    price double precision NOT NULL,  
    description varchar(255) NOT NULL,  
    shop_id bigint NOT NULL  
)  
;  
CREATE TABLE receipt (  
    id bigint NOT NULL,  
    user_id bigint NOT NULL,  
    order_id bigint NOT NULL  
)  
;  
CREATE TABLE review (  
    id bigint NOT NULL,  
    user_id bigint NOT NULL,  
    text_review varchar(255) NOT NULL  
)  
;  
CREATE TABLE ShopProduct (  
    id bigint NOT NULL,  
    product_id bigint NOT NULL,  
    shop_id bigint NOT NULL  
)  
;  
CREATE TABLE Shops (  
    id bigint NOT NULL,  
    name varchar(255) NOT NULL  
)  
;  
CREATE TABLE user_data (  
    id bigint NOT NULL,  
    address varchar(255) NOT NULL,  
    login varchar(255) NOT NULL,  
    password bigint NOT NULL,  
    card_number bigint NOT NULL,  
    user_id bigint NOT NULL  
)  
;  
CREATE TABLE users (  
    id bigint NOT NULL,  
    name varchar(255) NOT NULL,  
    phone_number varchar(255) NOT NULL,  
    email varchar(255) NOT NULL  
)  
;
```

```

ALTER TABLE Products
    ADD CONSTRAINT UQ_Products_id UNIQUE (id)
;
ALTER TABLE Shop_address
    ADD CONSTRAINT UQ_Shop_address_id UNIQUE (id)
;
ALTER TABLE Shops
    ADD CONSTRAINT UQ_Shops_id UNIQUE (id)
;
ALTER TABLE orders ADD CONSTRAINT PK_order
    PRIMARY KEY (id)
;

ALTER TABLE order_list ADD CONSTRAINT PK_order_list
    PRIMARY KEY (id)
;

ALTER TABLE Products ADD CONSTRAINT PK_Products
    PRIMARY KEY (id)
;

ALTER TABLE receipt ADD CONSTRAINT PK_receipt
    PRIMARY KEY (id)
;

ALTER TABLE review ADD CONSTRAINT PK_review
    PRIMARY KEY (id)
;

ALTER TABLE Shop_address ADD CONSTRAINT PK_Shop_address
    PRIMARY KEY (id)
;

ALTER TABLE ShopProduct ADD CONSTRAINT PK_ShopProduct
    PRIMARY KEY (id)
;
ALTER TABLE Shops ADD CONSTRAINT PK_Shops
    PRIMARY KEY (id)
;
ALTER TABLE user_data ADD CONSTRAINT PK_user_data
    PRIMARY KEY (id)
;
ALTER TABLE users ADD CONSTRAINT PK_users
    PRIMARY KEY (id)
;

ALTER TABLE Shops ADD CONSTRAINT FK_id
    FOREIGN KEY (id) REFERENCES Shop_address (id)
;

```