

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

# «МИРЭА ~~−~~ Российский технологический университет»

РТУ МИРЭА



Институт информационных технологий (ИИТ)

Кафедра промышленной информатики (ПИ)

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине «Разработка баз данных»

**Тема** «Разработка базы данных “Кондитерской”**»**

Студент группы ИКБО-27-20, Родионова А.В.

(подпись)

Преподаватель Королев Ф.А.

(подпись)

Отчет представлен « » 2022 г.

Москва 2022 г.

# ОГЛАВЛЕНИЕ

[Описание предметной области 3](#_TOC_250016)

1. [ЧАСТЬ 3](#_TOC_250015)

[ЗАДАНИЕ 3 3](#_TOC_250014)

[ЗАДАНИЕ 4 7](#_TOC_250013)

1. [ЧАСТЬ 9](#_TOC_250012)

[ЗАДАНИЕ 1 9](#_TOC_250011)

[ЗАДАНИЕ 2 14](#_TOC_250010)

[ЗАДАНИЕ 3 16](#_TOC_250009)

[Создание и заполнение таблиц на сервере 21](#_TOC_250008)

[Выборка данных, оператор SELECT 23](#_TOC_250007)

[Хранимые процедуры, функции и триггеры 23](#_TOC_250006)

[ЗАКЛЮЧЕНИЕ 29](#_TOC_250005)

ПРИЛОЖЕНИЯ 30

[Приложение 1 30](#_TOC_250004)

[Приложение 2 33](#_TOC_250003)

[Приложение 3 35](#_TOC_250002)

[Приложение 4 36](#_TOC_250001)

[Приложение 5 41](#_TOC_250000)

# Описание предметной области

Областью применения базы данных является Кондитерская. Кондитерская – это место, где продаются сладости. Сладости или конфеты, изготовленные в кондитерской (кондитерские изделия), также известны как кондитерские изделия. Следовательно, Кондитерская работает с базами данных, где хранится вся информация о выпускаемой продукции и клиентах. Менеджерам всегда необходимо следить за появлением новых клиентов, чтобы во время успевать перенаправлять заказы в кондитерские. А руководству и бухгалтерии необходимо информацию о своих сотрудниках. Для этого нужна общая база данных, включающая всю необходимую информацию.

В результате в базе данных "Кондитерская" используются следующие входные данные: информация о клиентах; информация о сотрудниках, о выпускаемой продукции, об активных и неактивных заказах

# 1 ЧАСТЬ

Для создания логической модели данных используем ERwin Data Modeler r7.3. Вначале запустим Erwin и создадим новый проект Logical/Physical, после чего в этом проекте создадим сущность “Кондитер” и добавим атрибут id, который будет иметь Primary Key (Рисунок 1).

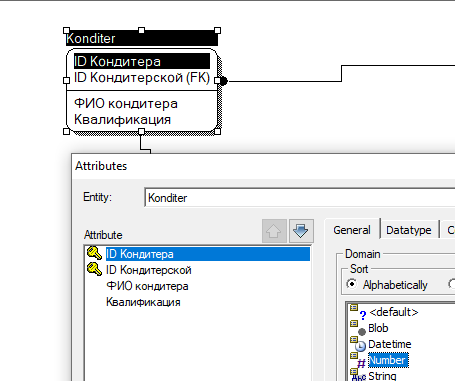


Рисунок 1. Сущность “Кондитерская”.

**Логическая и физическая модель данных**

Логическая модель базы данных (Рисунок 2).

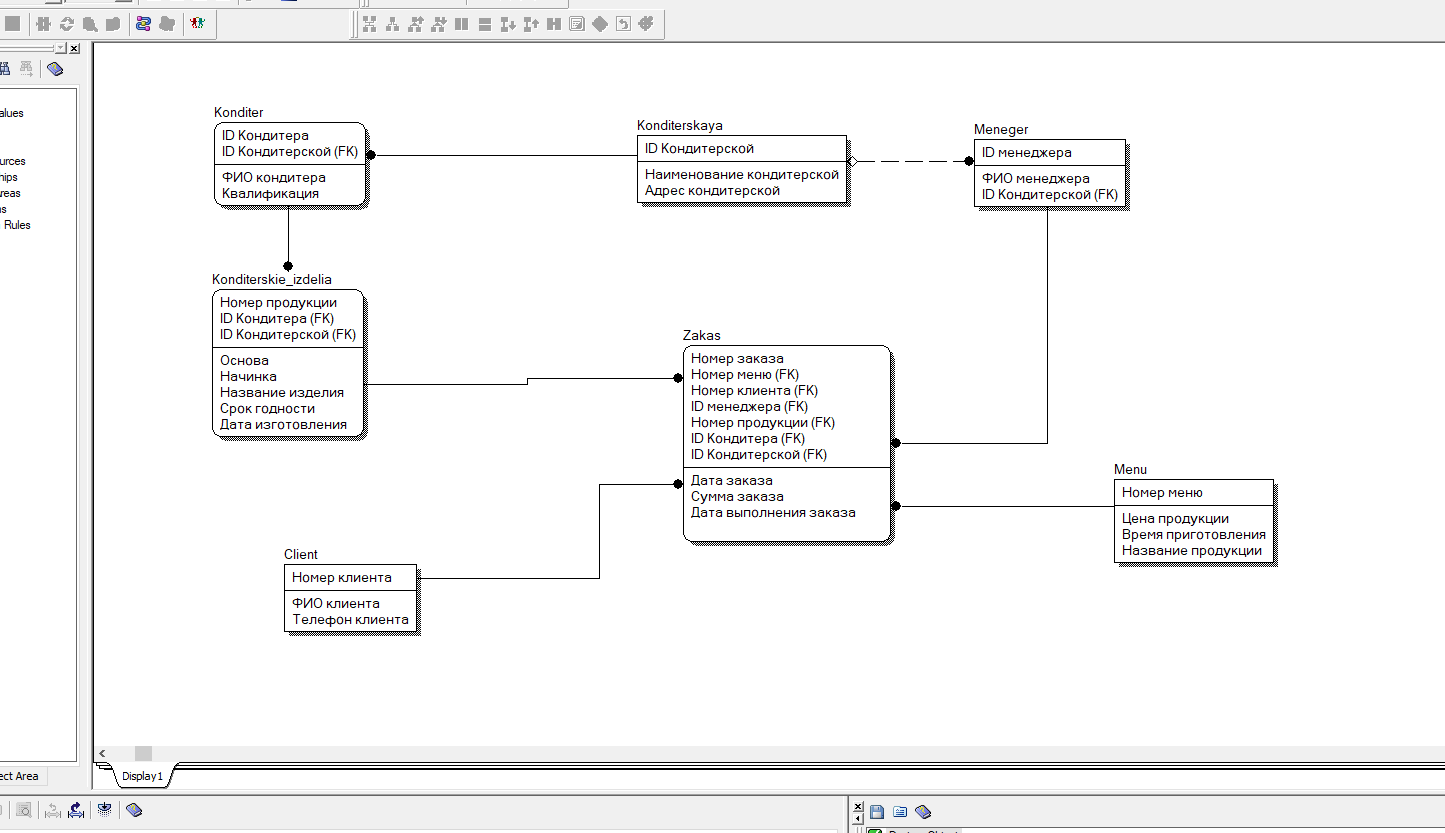


Рисунок 2. Логическая модель данных.

После построения логической модели перейдем на вкладку Physical на верхней панели.

Физическая модель данных (Рисунок 3.).

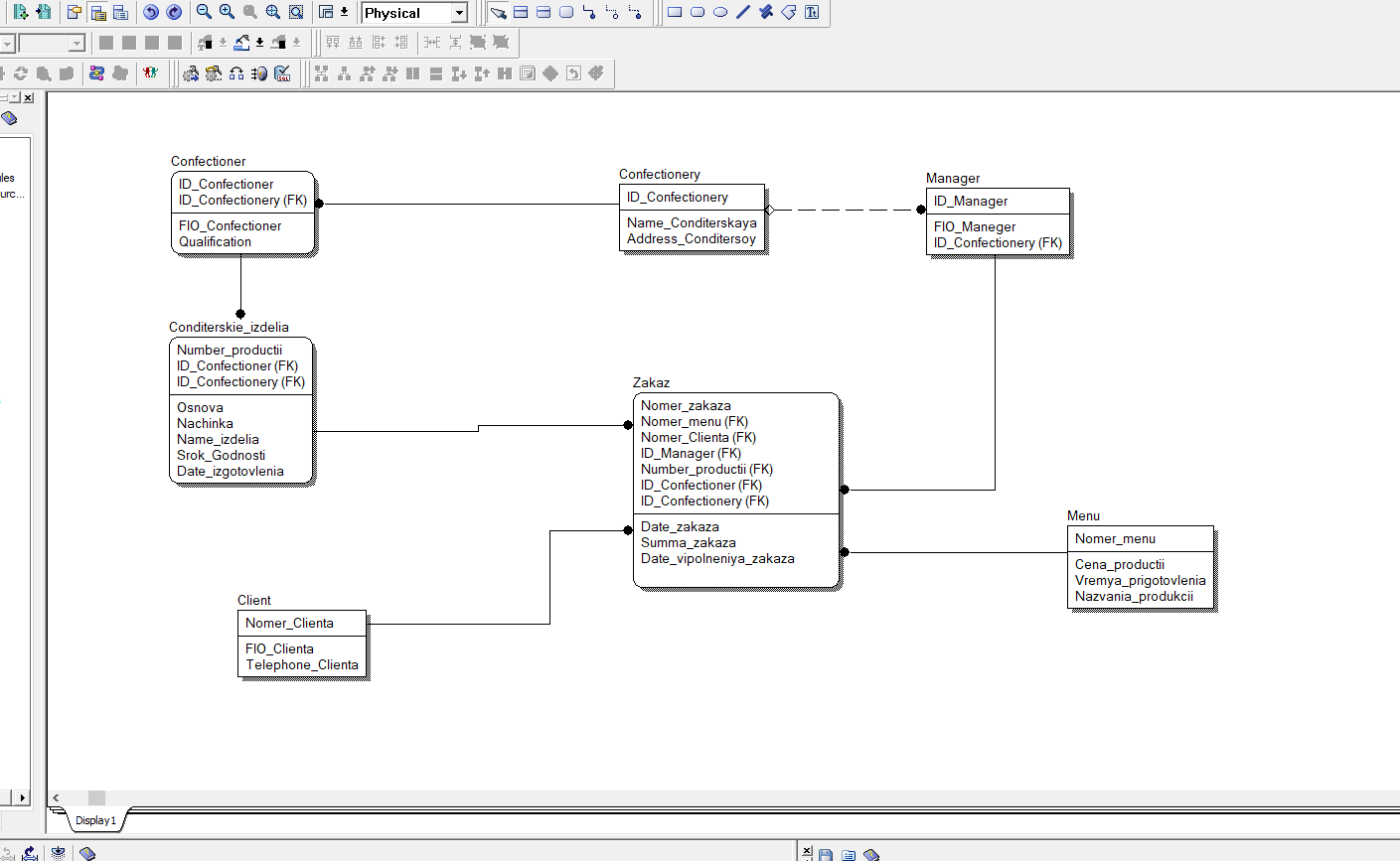


Рисунок 3. Физическая модель данных.

# ЗАДАНИЕ 1

Для создания базы данных и таблиц в ней используем MySQL Server

* 1. Прежде всего надо запустить сервер MySQL. Идем в системное меню Пуск — Программы — MySQL — MySQL Server 8.0 — MySQL Command Line Client. Откроется окно, предлагающее ввести пароль. Нажимаем Enter на клавиатуре, если вы не указывали пароль при настройке сервера или указываем пароль, если вы его задавали. Ждем приглашения mysql> (Рисунок 4).

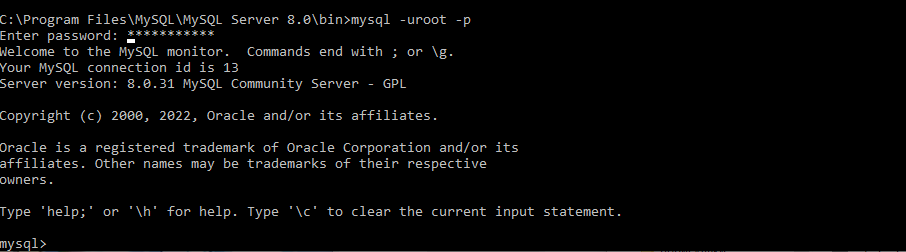


Рисунок 4. MySQL в командной строке.

Далее создадим базу данных и выберем ее для работы (Рисунок 5-6).

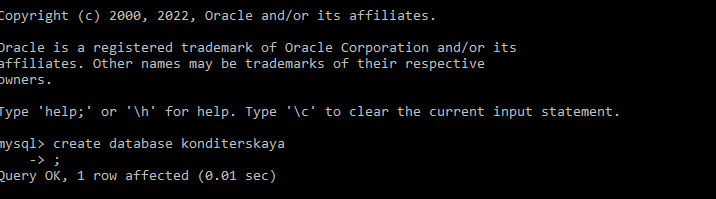


Рисунок 5. Создание базы данных “Кондитерская”.

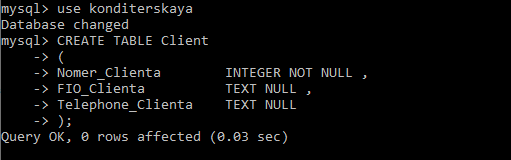


Рисунок 6. Выбор базы данных “Кондитерская”.

Напишем команды для создания таблиц базы данных ориентируясь на физическую модель данных (Приложение 1).

Отобразим список имеющихся таблиц и их описание (Рисунок 7-9).

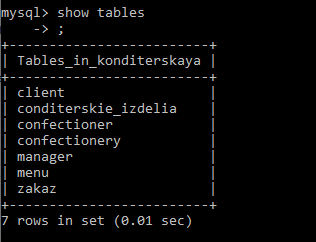


Рисунок 7. Таблицы в базе данных “Кондитерская”.

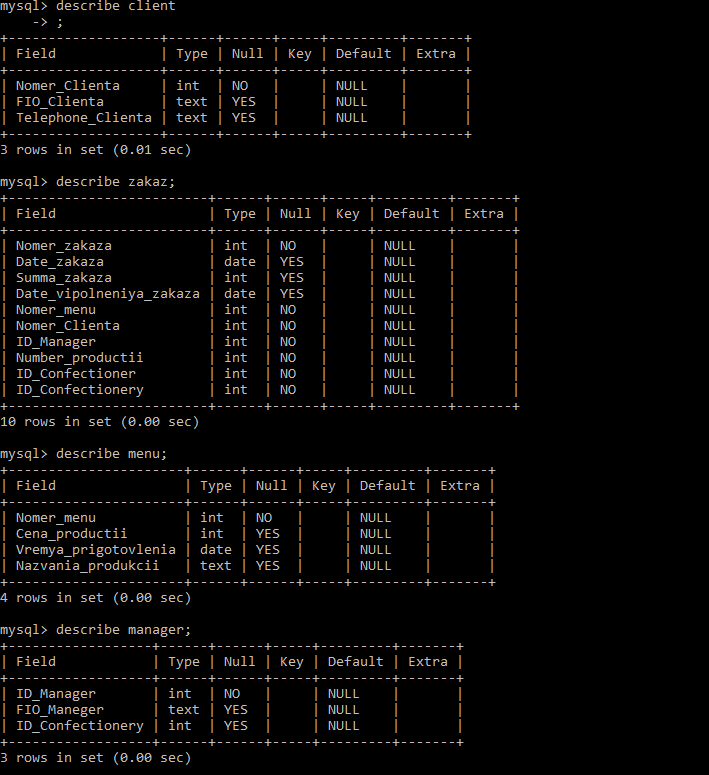


Рисунок 8. Описание 1ч.

Рисунок 9. Описание 2ч.

Далее необходимо заполнить таблицы. Для этого используется оператор INSERT. Будем использовать синтаксис для внесения данных во все поля таблицы (Приложение 2).

Давайте посмотрим все столбцы из таблиц (Рисунок 10-11).

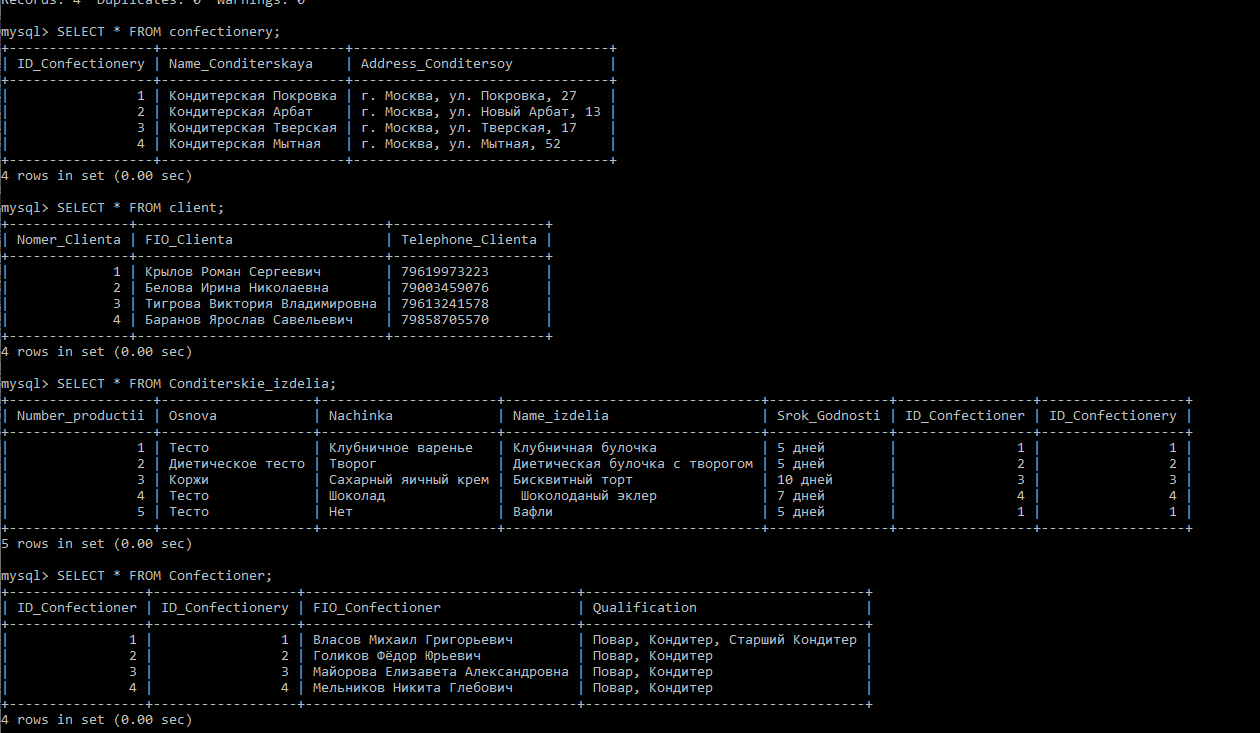


Рисунок 9. Заполненные столбцы 1ч.

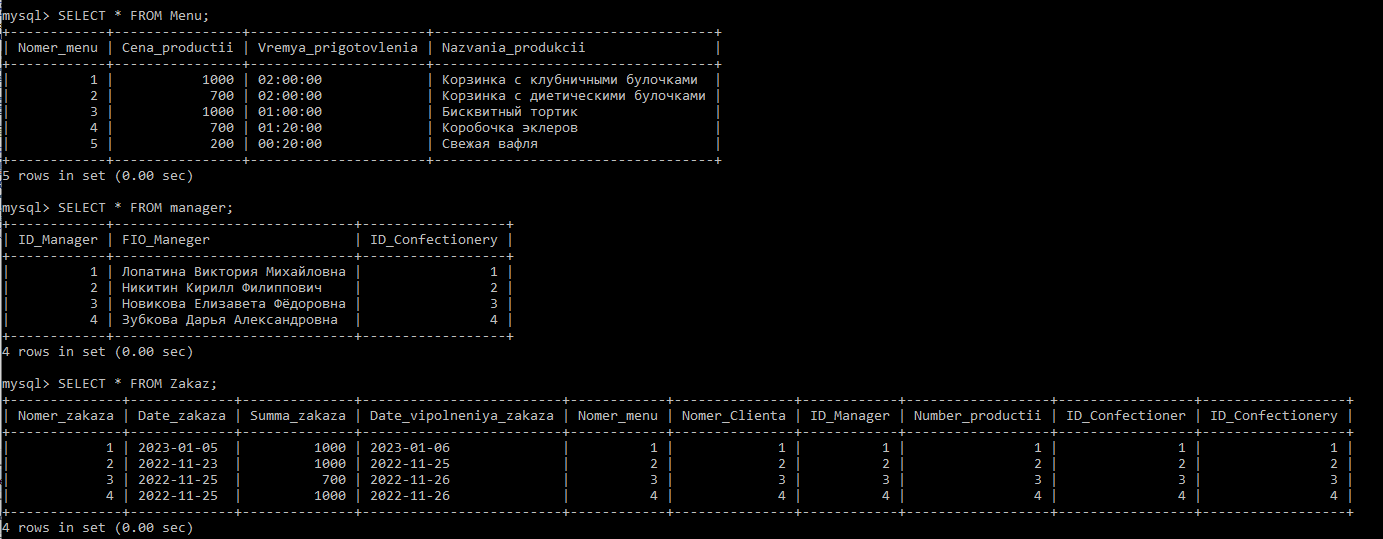


Рисунок 10. Заполненные столбцы 2ч.

# ЗАДАНИЕ 2

Если необходимо посмотреть, какие данные содержатся в таблицах, то для этого в SQL существует оператор SELECT. Синтаксис его использования следующий: SELECT что\_выбрать FROM откуда\_выбрать;

(Рисунок 11).

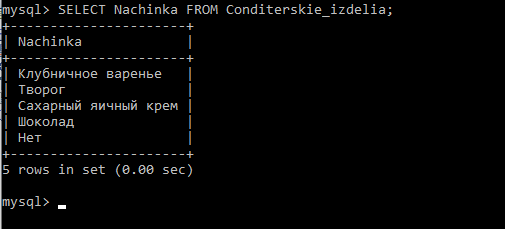


Рисунок 11. Оператор SELECT

Для более детальных сортировок используем операторы, которые есть в MySQL (Рисунок 12-16) . Метасимвол – %, означает любые символы. Метасимвол – \_ В отличие от %, который обозначает несколько или ни одного символа, нижнее подчеркивание обозначает ровно один символ.

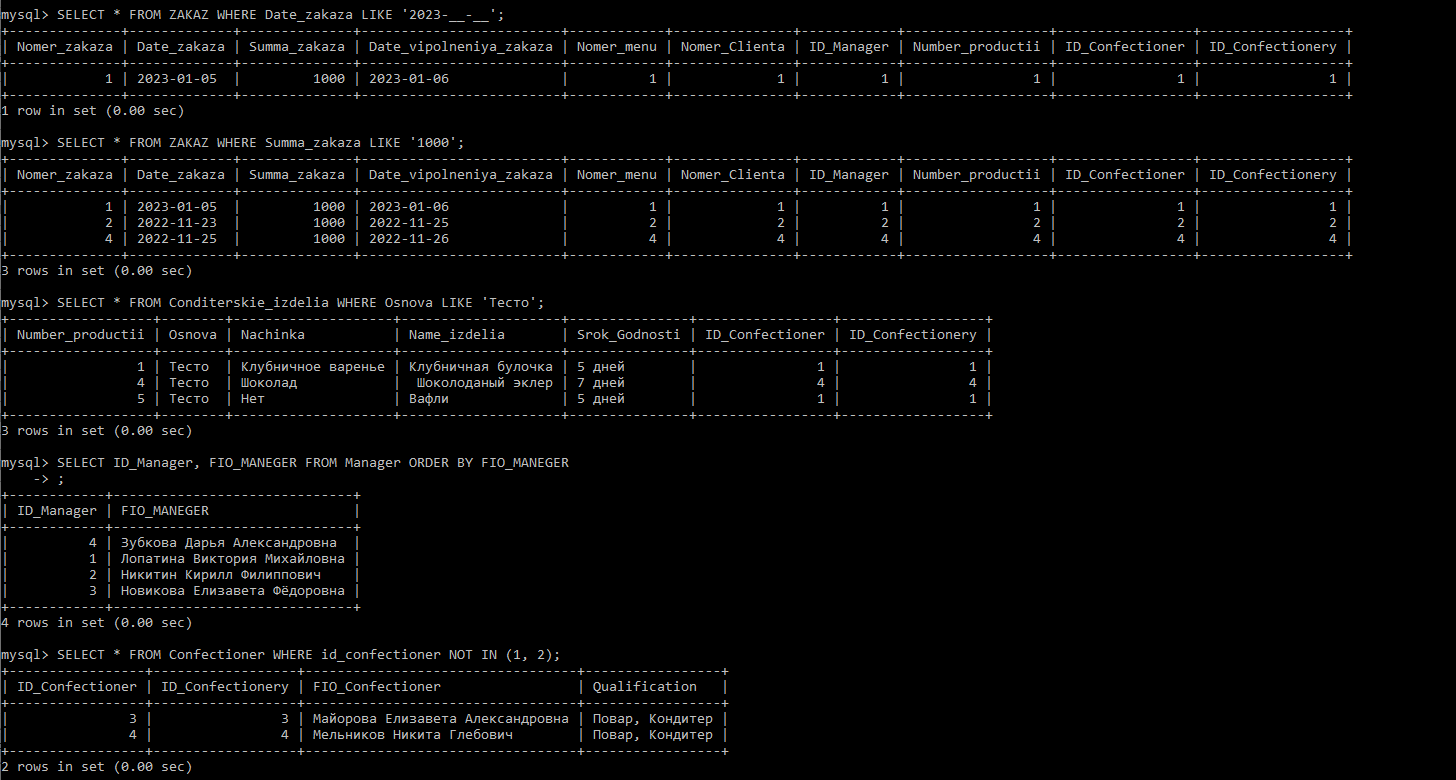


Рисунок 12. SELECT ч1.

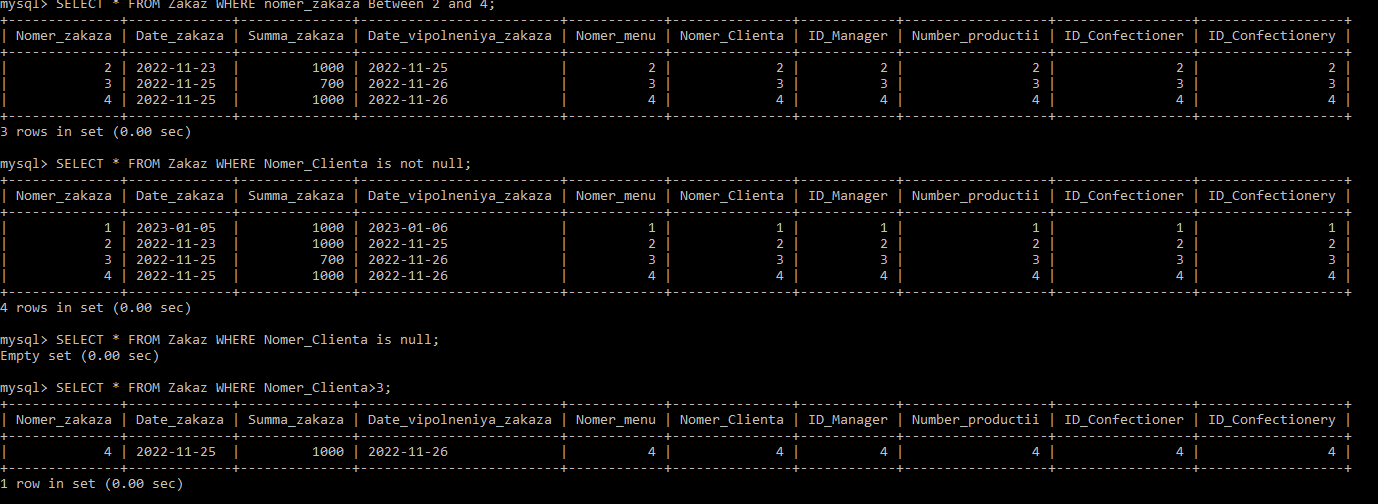


Рисунок 13. SELECT ч2.

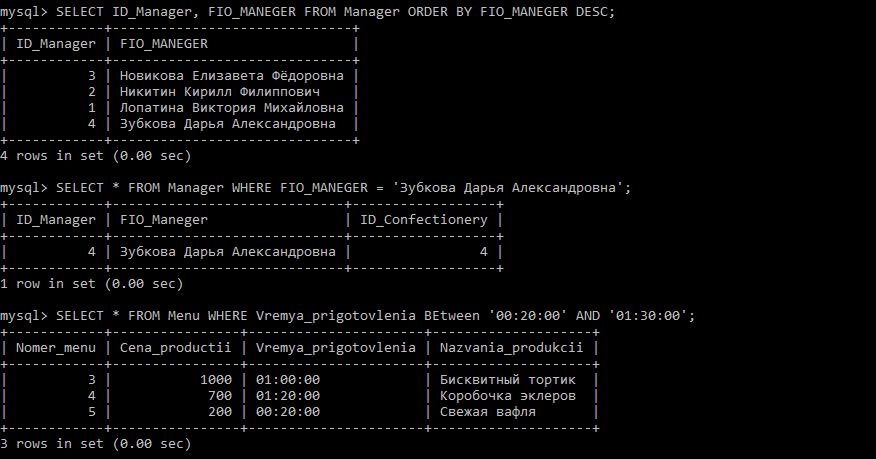


Рисунок 14. SELECT ч3.

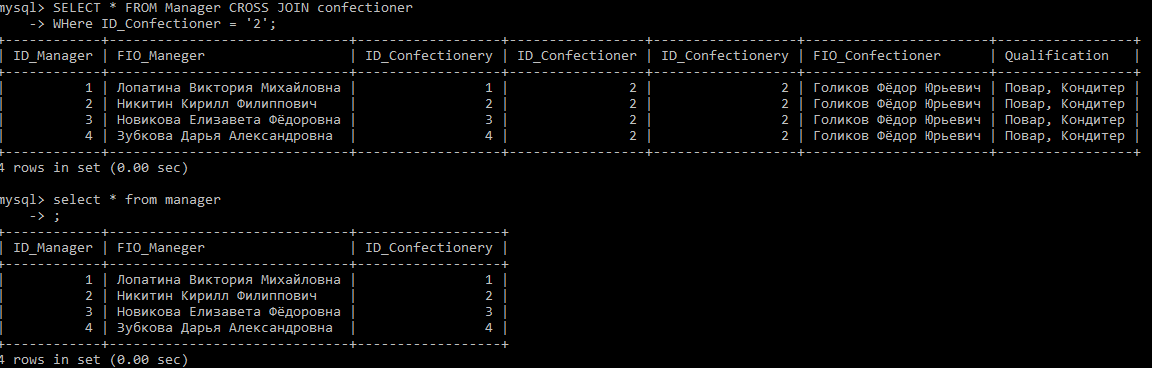


Рисунок 15. SELECT ч4.

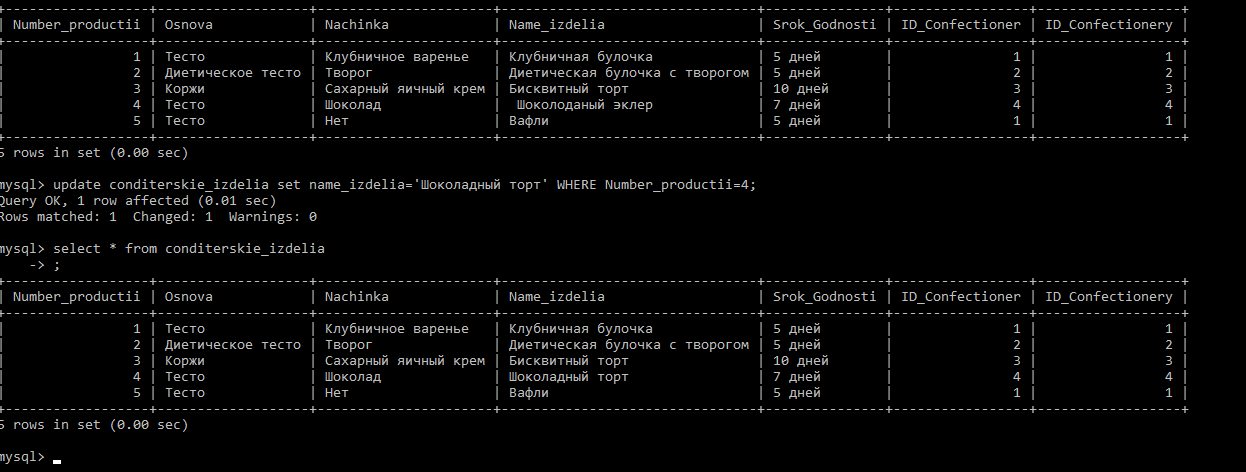


Рисунок 16. SELECT ч5.

# ЗАДАНИЕ 3

Для формирования модели БД используем MySQL Workbench.

Перейдем на вкладку Models и создадим новую модель (Рисунок 16). На вкладке настройки модели нажмем “Add Diagram” (Рисунок 17). Далее укажем название схемы и выберем collation — “utf8\_general\_ci” (Рисунок 18).

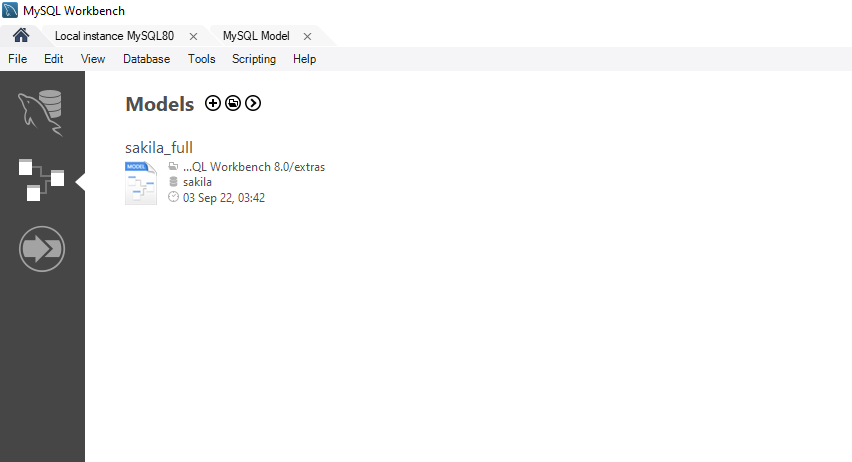


Рисунок 17. Создание модели.

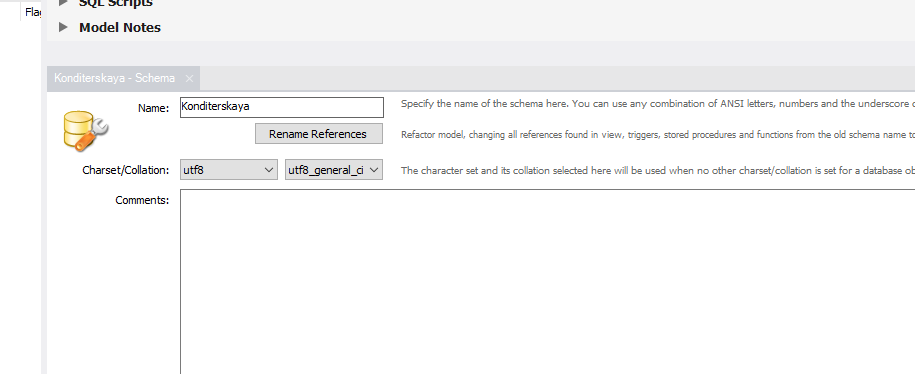


Рисунок 18. Настройки схемы.

Создадим таблицы базы данных, выполним соединения таблиц и настройку наследуемых ключей (Рисунок 19).

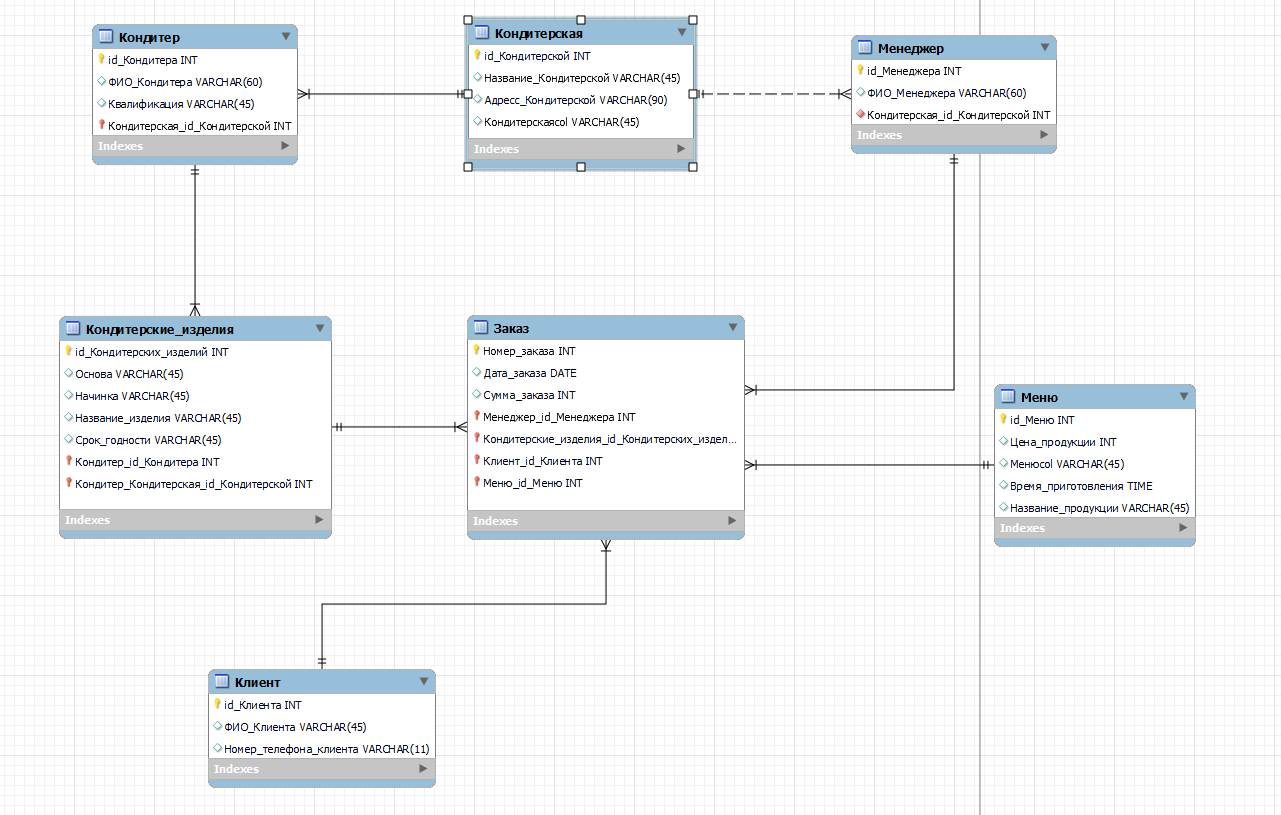


Рисунок 19. Модель “konditerskaya”.

Выполним перенос БД на сервер. Выберем сервер и настройки соединения (Рисунок 20-21).

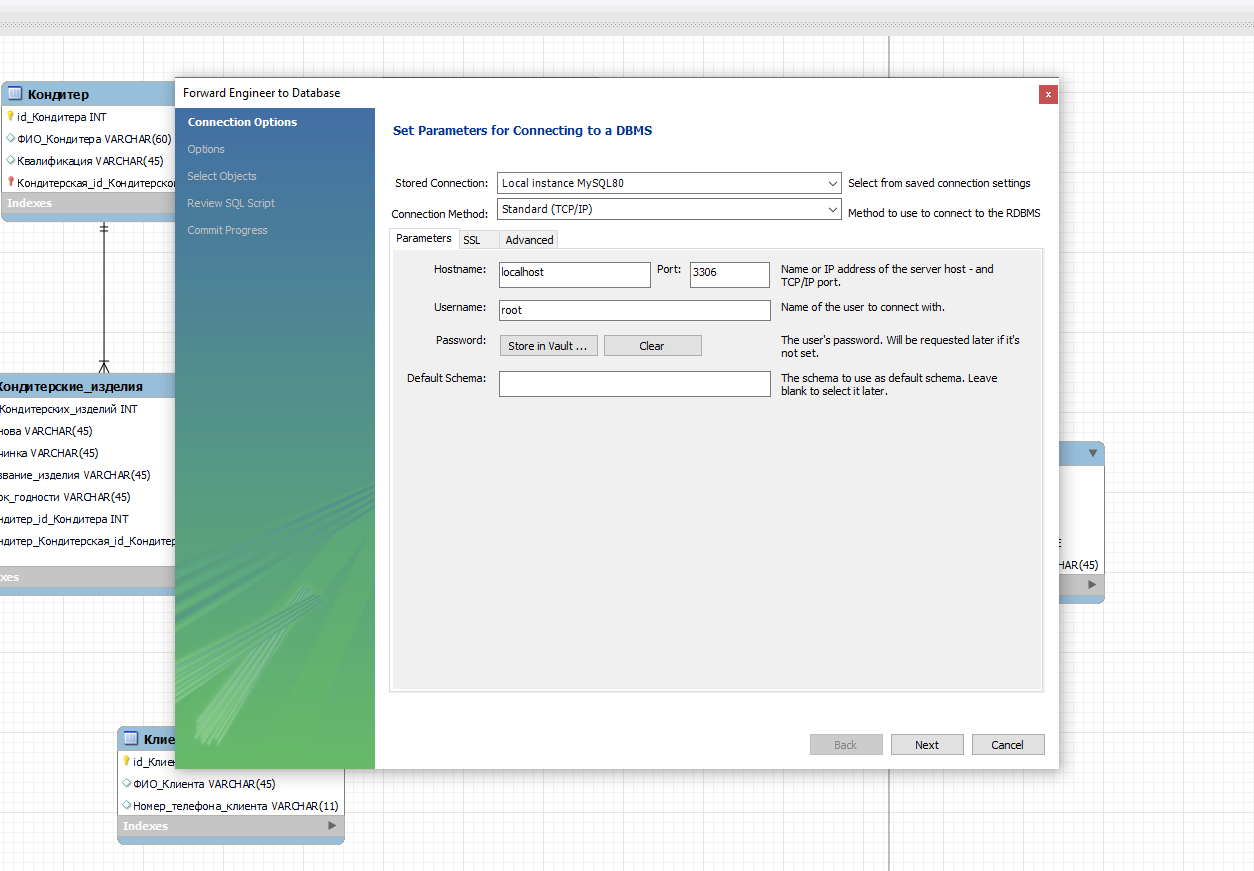


Рисунок 20. Соединение.

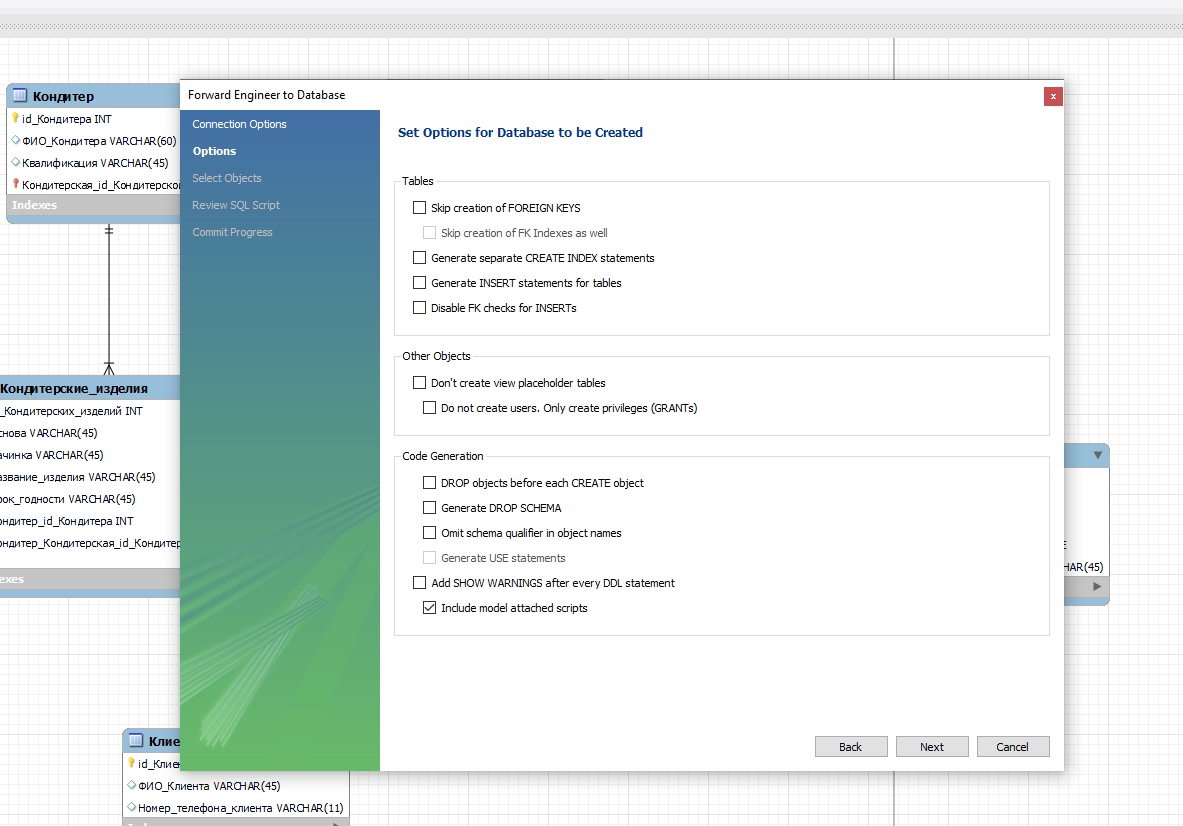


Рисунок 21. Настройки.

Выбор объектов генерации (Рисунок 22).

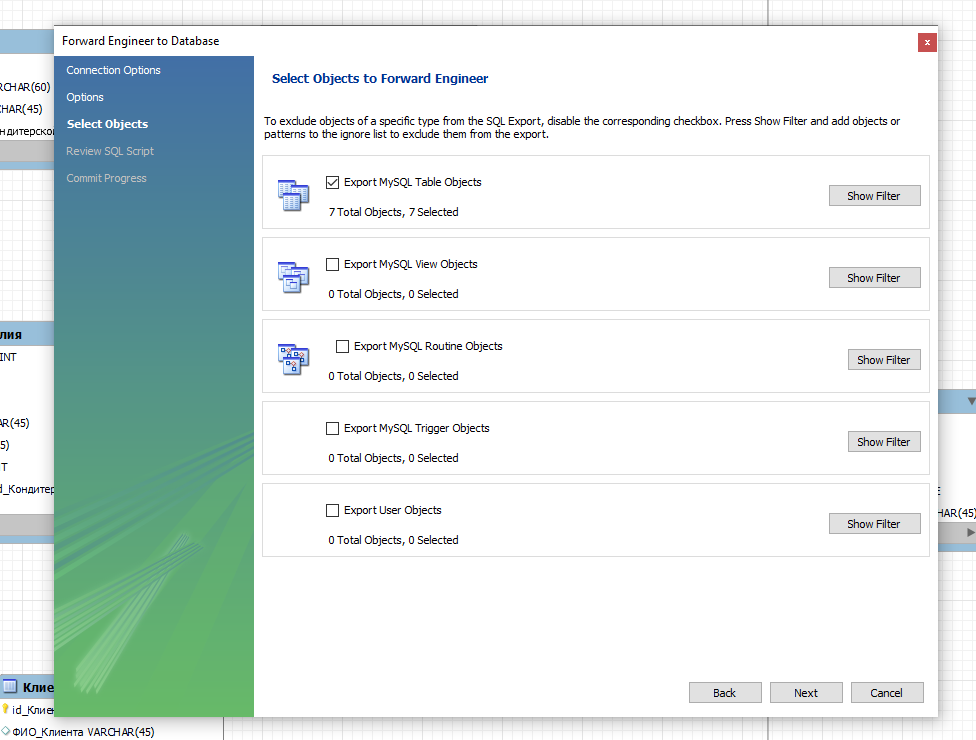


Рисунок 22. Объекты.

Просмотр сгенерированного скрипта SQL (Рисунок 23) (Приложение 3).

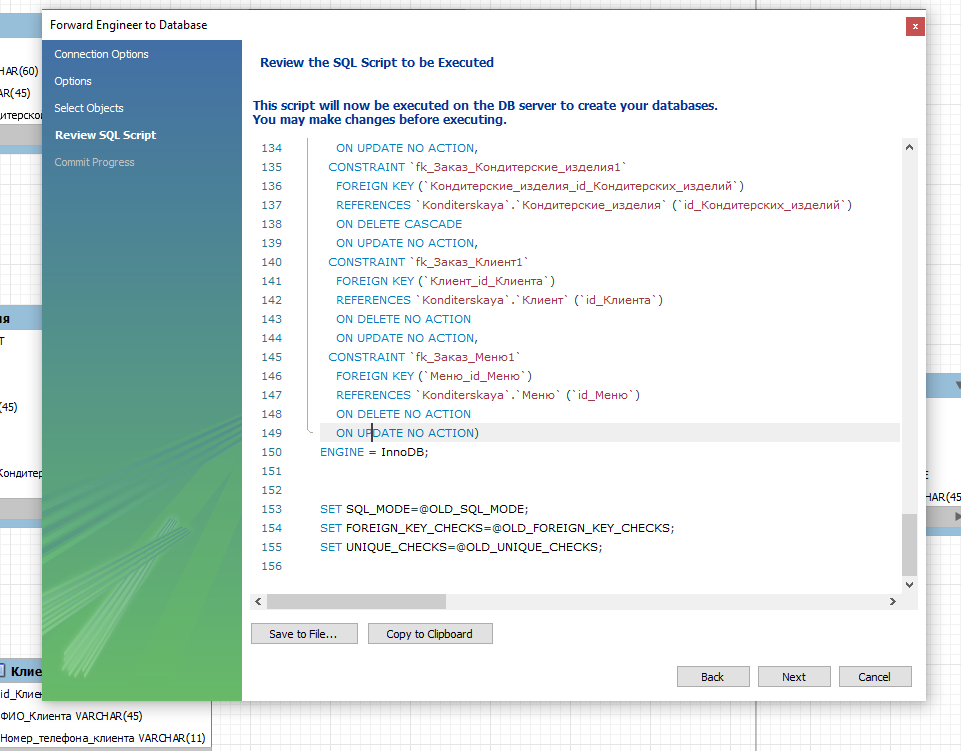


Рисунок 23. SQL скрипт.

Выполнение скрипта (Рисунок 24).

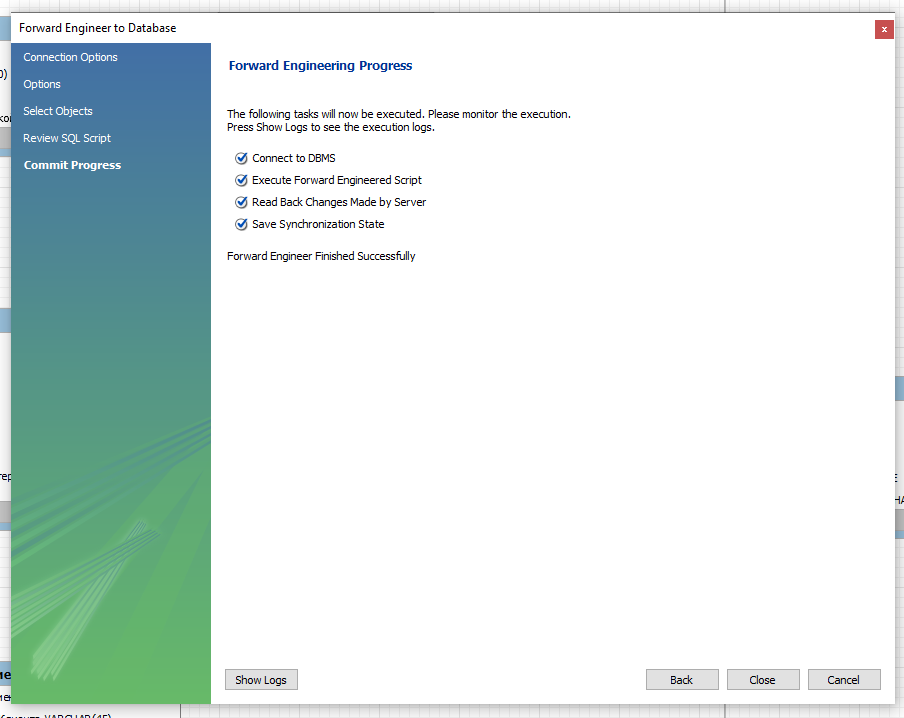


Рисунок 24. Отправка.

# Создание и заполнение таблиц на сервере

Запуск скрипта на сервере (Рисунок 25-26).

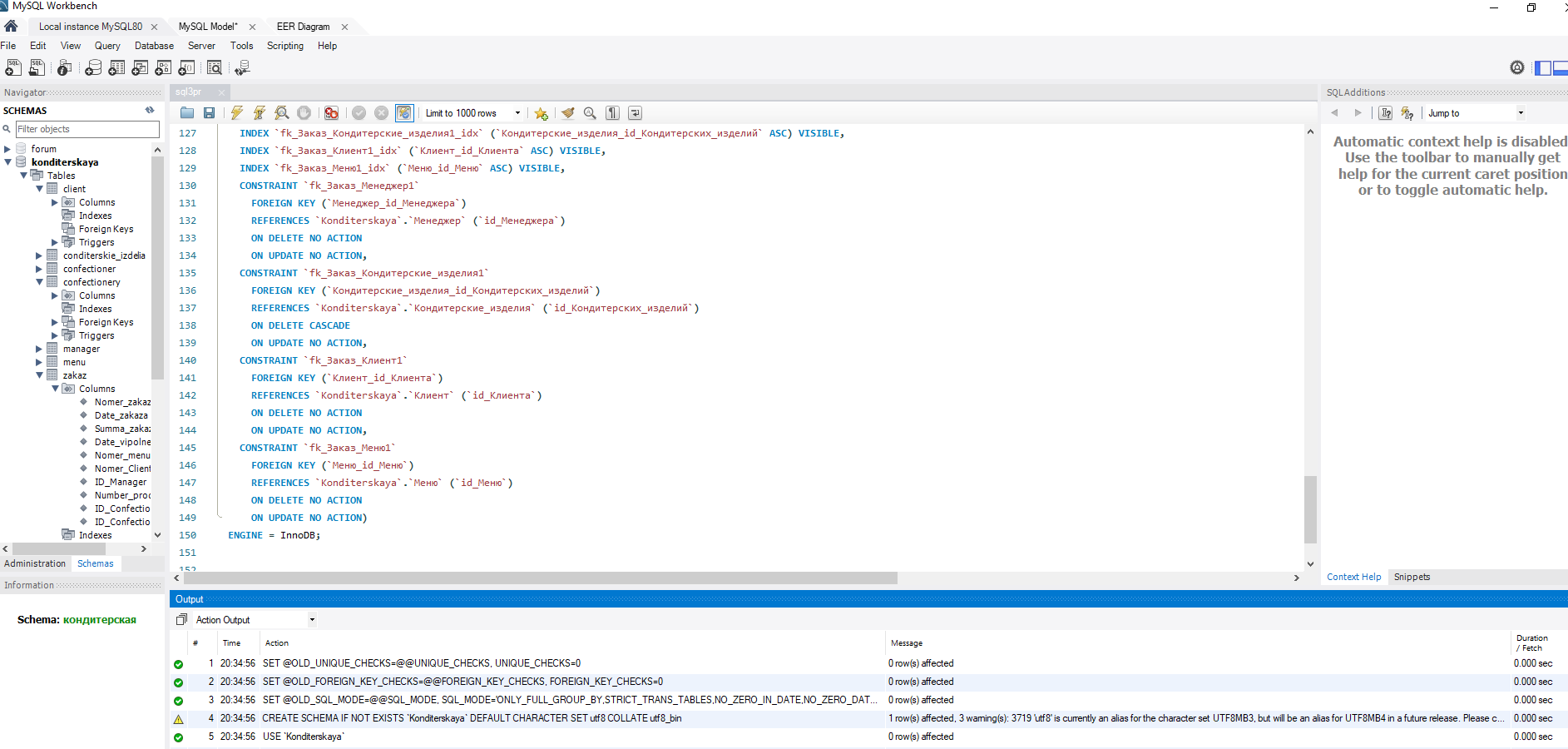


Рисунок 25. Запуск скрипта

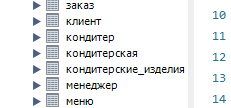


Рисунок 26. БД “konditerskaya”.

Заполним таблицы (Рисунок 27-28) (Приложение 2).

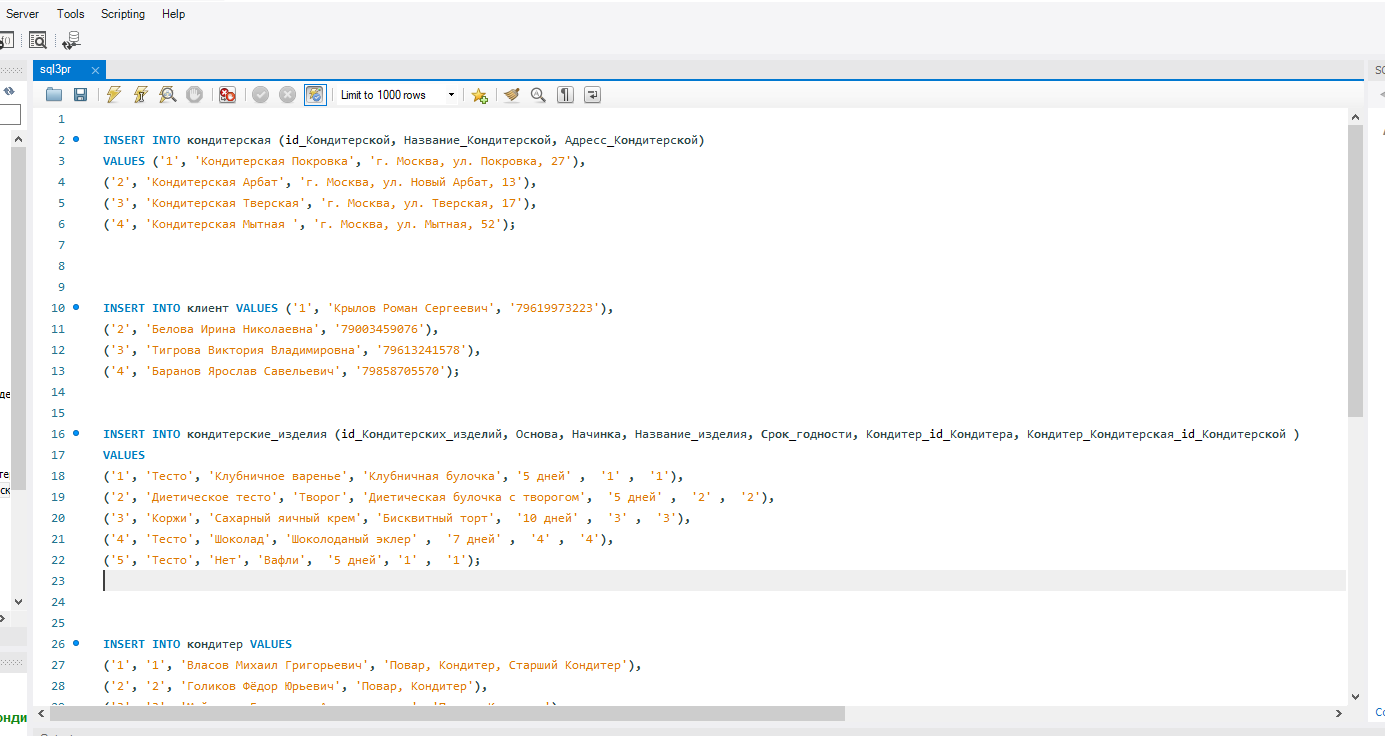


Рисунок 27. Заполнение.

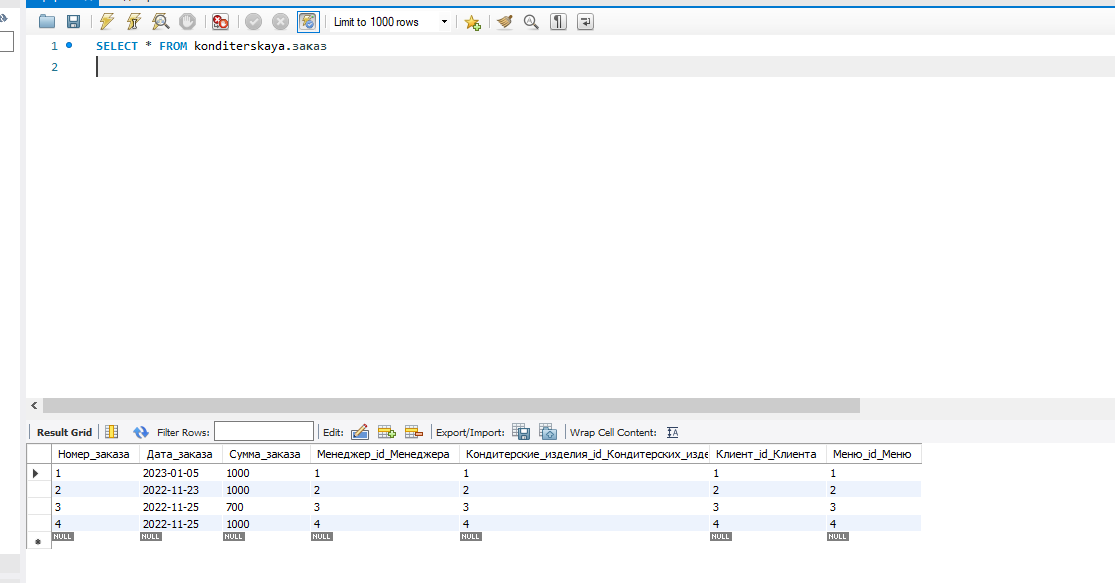


Рисунок 28. Результат заполнения.

# Выборка данных, оператор SELECT

Выполним сортировки (Рисунок 29).

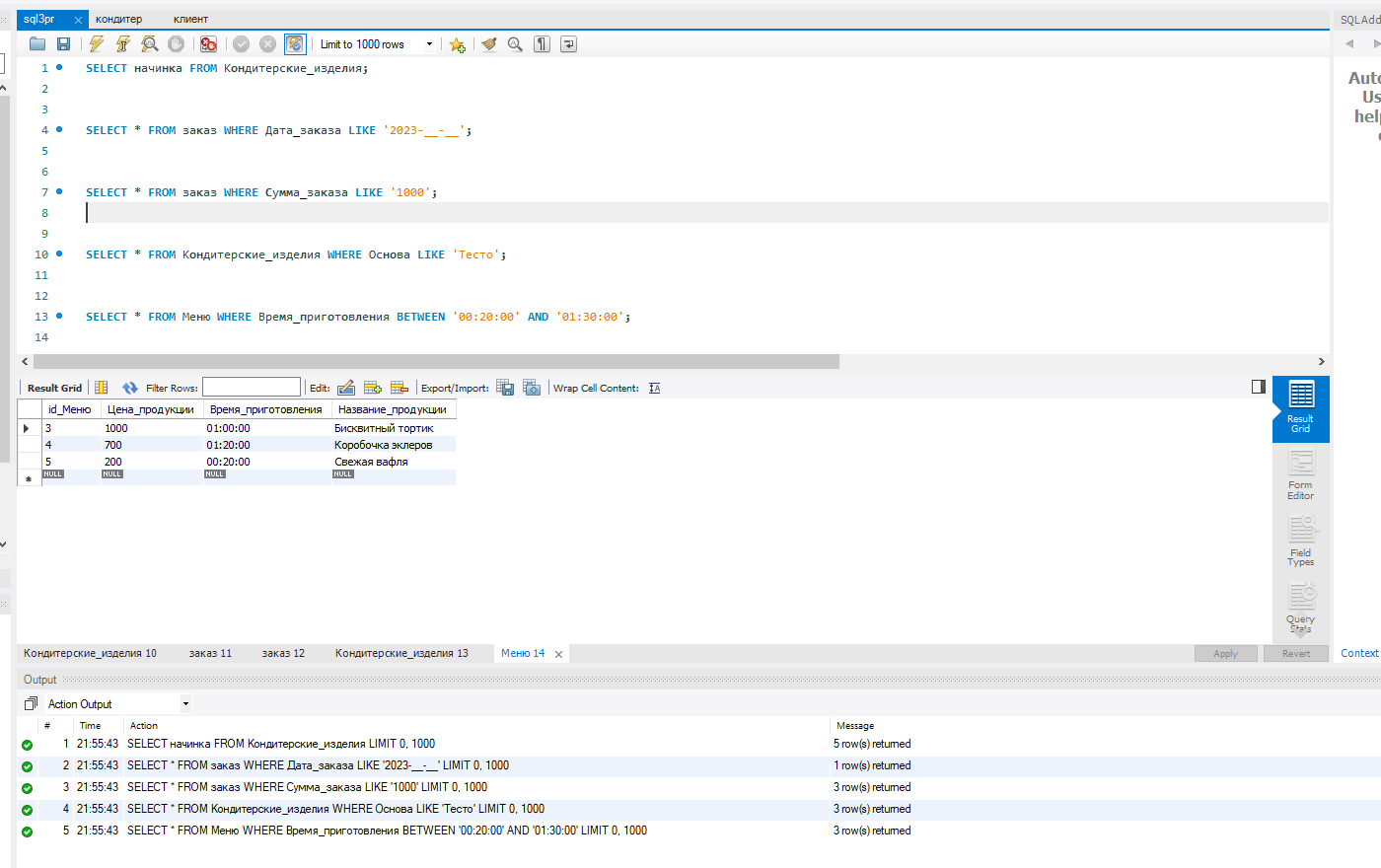


Рисунок 29. Сортировки.

# Хранимые процедуры, функции и триггеры

Напишем процедуру по подсчету ФИО\_Клиента в таблице *клиент* и вызовем ее (Рисунок 30-31).

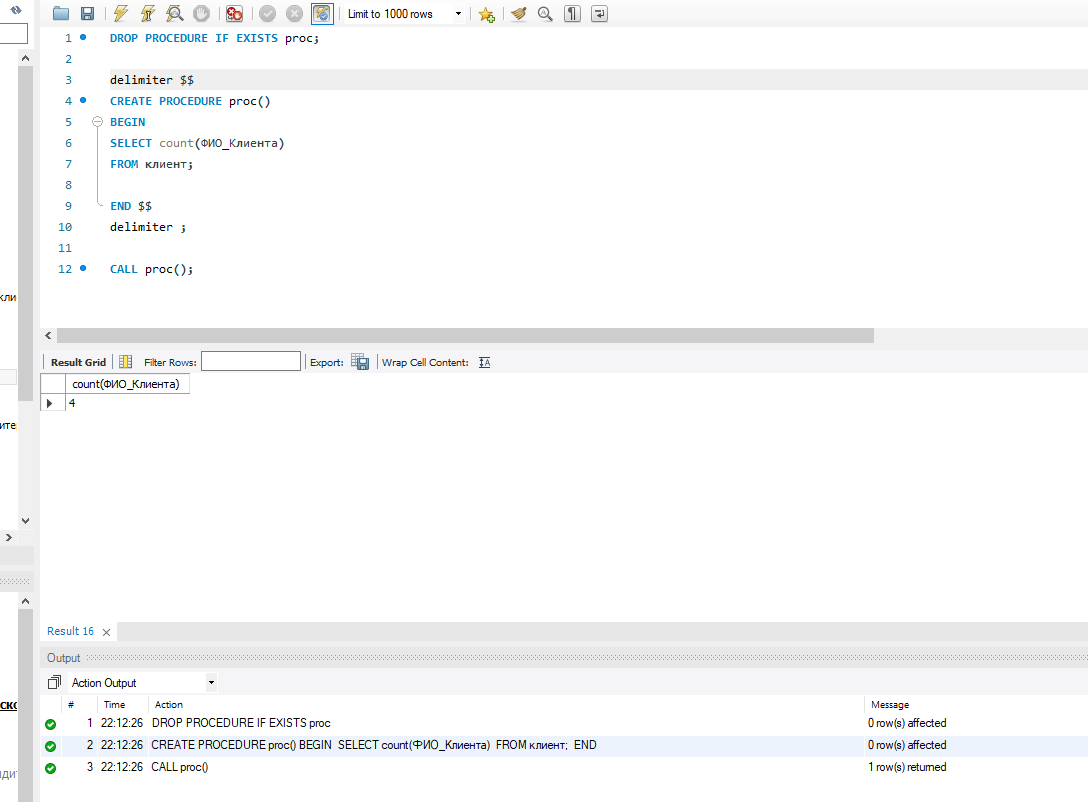


Рисунок 30. Процедура proc.

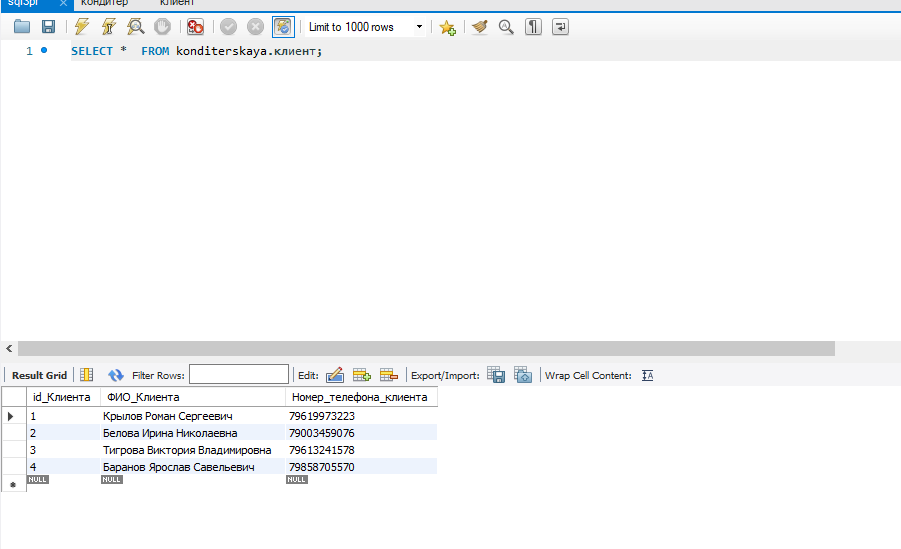


Рисунок 31. Таблица клиент.

Процедура по нахождению заказов по дате создания (Рисунок 32).

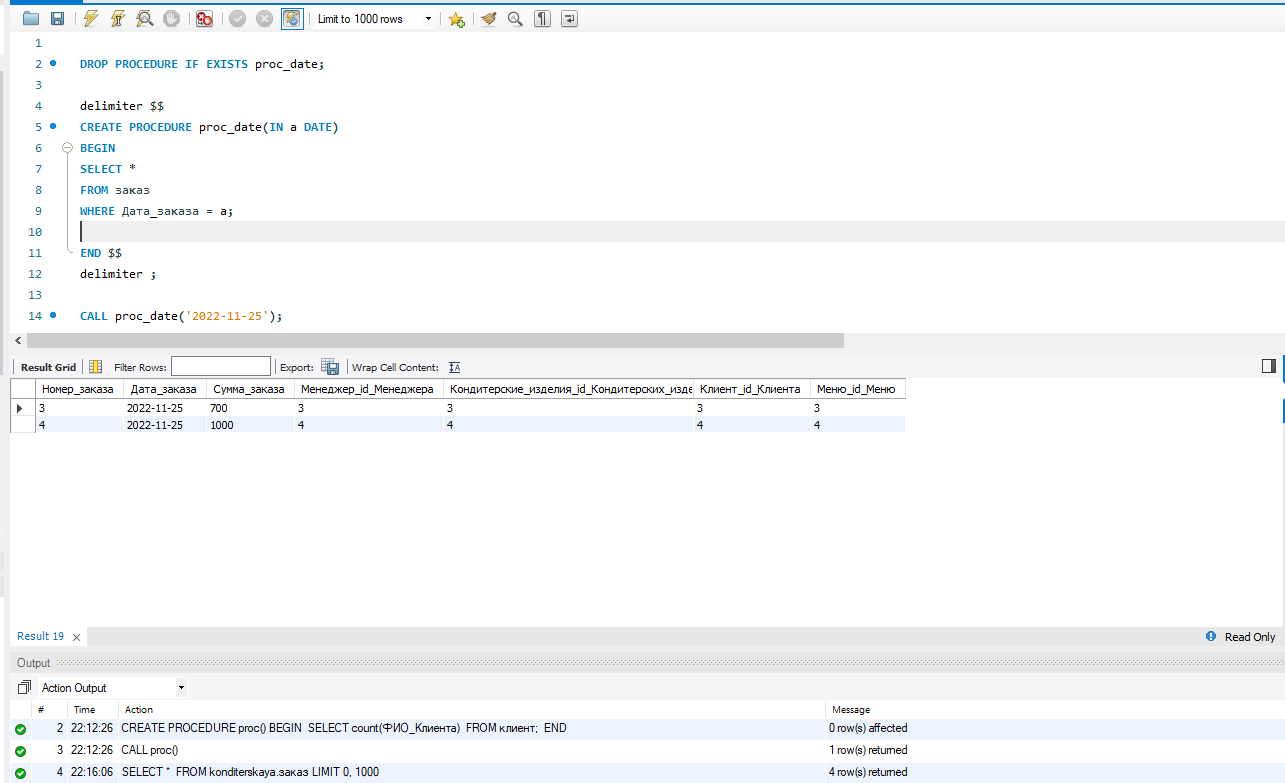


Рисунок 32. Процедура proc\_date.

Функция по проверке номера (Рисунок 33).

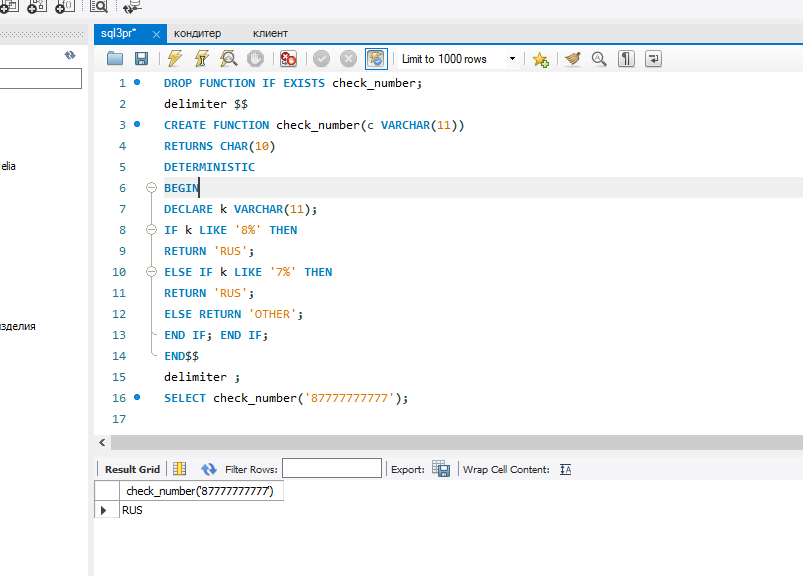


Рисунок 33. Функция check\_number

Триггер на проверку вхождения неправильной даты в таблице “заказ” (Рисунок 34).

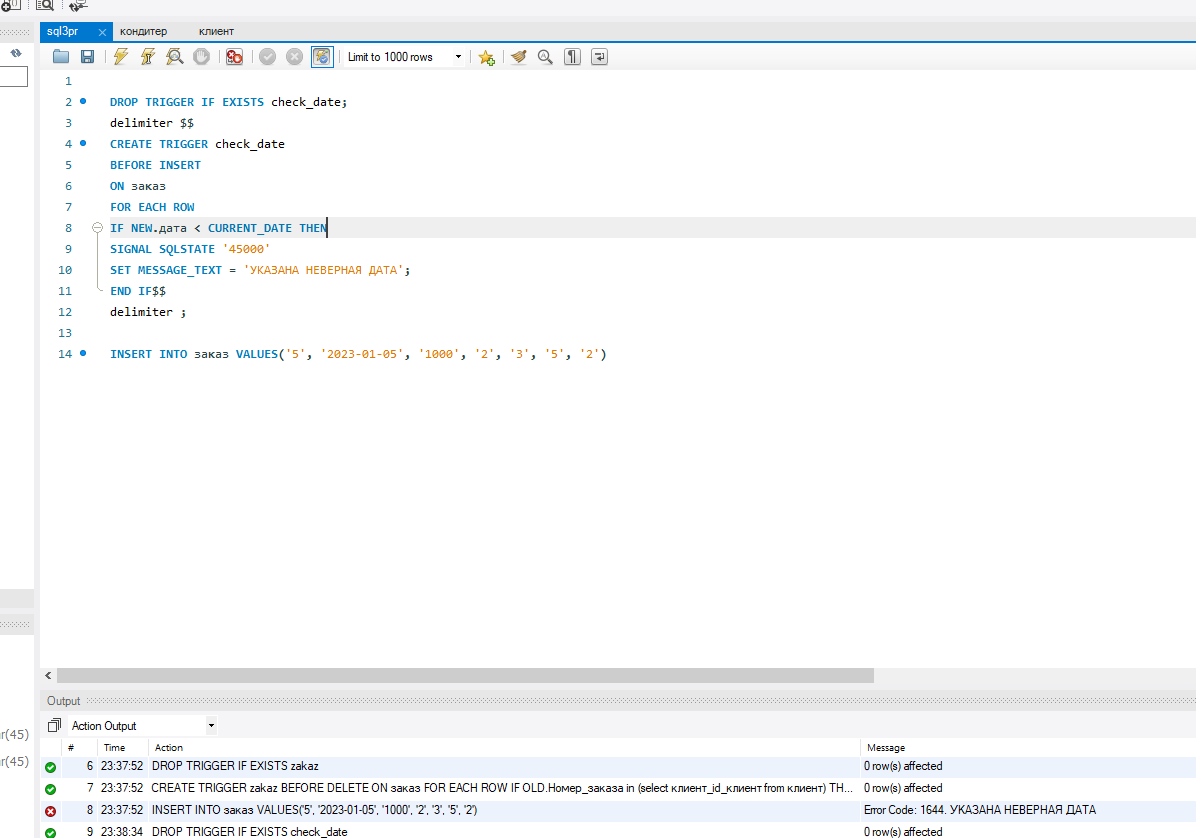


Рисунок 39. Триггер check\_date

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практических работ была создана база данных кондитерской, позволяющая автоматизировать поиск необходимого заказа и просматривать данные о клиентах. Были построены логическая и физическая модели данных в программном обеспечении “Erwin Data Modeler”. В процессе выполнения работы были изучены теоретические аспекты рассматриваемой темы, была дана характеристика предметной области.

При проектировании базы данных были реализованы:

* Установка и настройка MySQL Workbench
* Создание базы данных в MySQL 8.0 Command Line Client
* Создание модели в MySQL Workbench

Анализ работы объектов показал, что они в полной мере выполнили возложенные на них функции. Таким образом, все поставленные в практических работах задачи выполнены, цель работы можно считать достигнутой.

# Список использованных источников и литературы:

* + 1. Материалы раздела “Разработка баз данных” на СДО МИРЭА – Российского технологического университета [Электронный ресурс] Режим доступа: https://online-edu.mirea.ru/course/view.php?id=7699 (Дата обращения: 12.12.2022).
    2. Продан Р. К. Проектирование баз данных: Лекции / Серебрянкин В. А.: Метододические указания. Российский технологический университет, 2021.
    3. Инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов Erwin Process Modeler [Электронный ресурс] – URL: https://erwin.com. (Дата обращения: 30.03.2022).
    4. CASE-средство IBM Rational Rose [Электронный ресурс] –

URL:

https://[www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001.](http://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001) (Дата обращения: 27.04.2022).

* + 1. Документация MySQL [Электронный ресурс] – URL: https://dev.mysql.com/doc/ (Дата обращения: 05.11.2022).
    2. MySQL Database Triggers [Электронный ресурс] – URL: https://[www.digitalocean.com/community/tutorials/how-to-manage-and-use-mys](http://www.digitalocean.com/community/tutorials/how-to-manage-and-use-mys) ql-database-triggers-on-ubuntu-18-04 (Дата обращения: 20.11.2022). **ПРИЛОЖЕНИЯ**

# Приложение 1

# CREATE TABLE Client

# (

# Nomer\_Clienta INTEGER NOT NULL ,

# FIO\_Clienta TEXT NULL ,

# Telephone\_Clienta TEXT NULL

# );

# CREATE TABLE Conditerskie\_izdelia

# (

# Number\_productii INTEGER NOT NULL ,

# Osnova TEXT NULL ,

# Nachinka TEXT NULL ,

# Name\_izdelia TEXT NULL ,

# Srok\_Godnosti DATE NULL ,

# Date\_izgotovlenia DATE NULL ,

# ID\_Confectioner INTEGER NOT NULL ,

# ID\_Confectionery INTEGER NOT NULL

# );

# CREATE TABLE Confectioner

# (

# ID\_Confectioner INTEGER NOT NULL ,

# ID\_Confectionery INTEGER NOT NULL ,

# FIO\_Confectioner TEXT NULL ,

# Qualification TEXT NULL

# );

# CREATE TABLE Confectionery

# (

# ID\_Confectionery INTEGER NOT NULL ,

# Name\_Conditerskaya TEXT NULL ,

# Address\_Conditersoy TEXT NULL

# );

# CREATE TABLE Manager

# (

# ID\_Manager INTEGER NOT NULL ,

# FIO\_Maneger TEXT NULL ,

# ID\_Confectionery INTEGER NULL

# );

# CREATE TABLE Menu

# (

# Nomer\_menu INTEGER NOT NULL ,

# Cena\_productii INTEGER NULL ,

# Vremya\_prigotovlenia DATE NULL ,

# Nazvania\_produkcii TEXT NULL

# );

# CREATE TABLE Zakaz

# (

# Nomer\_zakaza INTEGER NOT NULL ,

# Date\_zakaza DATE NULL ,

# Summa\_zakaza INTEGER NULL ,

# Date\_vipolneniya\_zakaza DATE NULL ,

# Nomer\_menu INTEGER NOT NULL ,

# Nomer\_Clienta INTEGER NOT NULL ,

# ID\_Manager INTEGER NOT NULL ,

# Number\_productii INTEGER NOT NULL ,

# ID\_Confectioner INTEGER NOT NULL ,

# ID\_Confectionery INTEGER NOT NULL

# );

# Приложение 2

# Кондитерская

# INSERT INTO confectionery VALUES ('1', 'Кондитерская Покровка', 'г. Москва, ул. Покровка, 27'),

# ('2', 'Кондитерская Арбат', 'г. Москва, ул. Новый Арбат, 13'),

# ('3', 'Кондитерская Тверская', 'г. Москва, ул. Тверская, 17'),

# ('4', 'Кондитерская Мытная ', 'г. Москва, ул. Мытная, 52');

# Клиент

# INSERT INTO client VALUES ('1', 'Крылов Роман Сергеевич', '79619973223'),

# ('2', 'Белова Ирина Николаевна', '79003459076'),

# ('3', 'Тигрова Виктория Владимировна', '79613241578'),

# ('4', 'Баранов Ярослав Савельевич', '79858705570');

# Кондитерские изделья

# INSERT INTO Conditerskie\_izdelia VALUES

# ('1', 'Тесто', 'Клубничное варенье', 'Клубничная булочка', '5 дней' , '1' , '1'),

# ('2', 'Диетическое тесто', 'Творог', 'Диетическая булочка с творогом', '5 дней' , '2' , '2'),

# ('3', 'Коржи', 'Сахарный яичный крем', 'Бисквитный торт', '10 дней' , '3' , '3'),

# ('4', 'Тесто', 'Шоколад', ' Шоколоданый эклер' , '7 дней' , '4' , '4'),

# ('5', 'Тесто', 'Нет', 'Вафли', '5 дней', '1' , '1');

# Кондитер

# INSERT INTO Confectioner VALUES

# ('1', '1', 'Власов Михаил Григорьевич', 'Повар, Кондитер, Старший Кондитер'),

# ('2', '2', 'Голиков Фёдор Юрьевич', 'Повар, Кондитер'),

# ('3', '3', 'Майорова Елизавета Александровна', 'Повар, Кондитер'),

# ('4', '4', 'Мельников Никита Глебович', 'Повар, Кондитер');

# Меню

# INSERT INTO Menu VALUES

# ('1', '1000', '02:00:00', 'Корзинка с клубничными булочками'),

# ('2', '700', '02:00:00', 'Корзинка с диетическими булочками'),

# ('3', '1000', '01:00:00', 'Бисквитный тортик'),

# ('4', '700', '01:20:00', 'Коробочка эклеров'),

# ('5', '200', '00:20:00', 'Свежая вафля');

# Менеджер

# INSERT INTO manager VALUES

# ('1', 'Лопатина Виктория Михайловна', '1'),

# ('2', 'Никитин Кирилл Филиппович', '2'),

# ('3', 'Новикова Елизавета Фёдоровна', '3'),

# ('4', 'Зубкова Дарья Александровна', '4');

# Заказ

# INSERT INTO Zakaz VALUES

# ('1', '2023-01-05', '1000', '2023-01-06', '1' , '1' , '1' , '1' , '1' , '1'),

# ('2', '2022-11-23', '1000', '2022-11-25', '2' , '2' , '2' , '2' , '2' , '2'),

# ('3', '2022-11-25', '700', '2022-11-26', '3' , '3' , '3' , '3' , '3' , '3'),

# ('4', '2022-11-25', '1000', '2022-11-26' , '4' , '4' , '4' , '4' , '4' , '4');

# Приложение 3

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

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

-- Schema Konditerskaya

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

--

--

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

-- Schema Konditerskaya

--

--

--

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

CREATE SCHEMA IF NOT EXISTS `Konditerskaya` DEFAULT CHARACTER SET utf8 COLLATE utf8\_bin ;

USE `Konditerskaya` ;

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

-- Table `Konditerskaya`.`Кондитерская`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Кондитерская` (

`id\_Кондитерской` INT NOT NULL,

`Название\_Кондитерской` VARCHAR(45) NULL,

`Адресс\_Кондитерской` VARCHAR(90) NULL,

`Кондитерскаяcol` VARCHAR(45) NULL,

PRIMARY KEY (`id\_Кондитерской`))

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Кондитер`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Кондитер` (

`id\_Кондитера` INT NOT NULL,

`ФИО\_Кондитера` VARCHAR(60) NULL,

`Квалификация` VARCHAR(45) NULL,

`Кондитерская\_id\_Кондитерской` INT NOT NULL,

PRIMARY KEY (`id\_Кондитера`, `Кондитерская\_id\_Кондитерской`),

INDEX `fk\_Кондитер\_Кондитерская\_idx` (`Кондитерская\_id\_Кондитерской` ASC) VISIBLE,

CONSTRAINT `fk\_Кондитер\_Кондитерская`

FOREIGN KEY (`Кондитерская\_id\_Кондитерской`)

REFERENCES `Konditerskaya`.`Кондитерская` (`id\_Кондитерской`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Кондитерские\_изделия`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Кондитерские\_изделия` (

`id\_Кондитерских\_изделий` INT NOT NULL,

`Основа` VARCHAR(45) NULL,

`Начинка` VARCHAR(45) NULL,

`Название\_изделия` VARCHAR(45) NULL,

`Срок\_годности` VARCHAR(45) NULL,

`Кондитер\_id\_Кондитера` INT NOT NULL,

`Кондитер\_Кондитерская\_id\_Кондитерской` INT NOT NULL,

PRIMARY KEY (`id\_Кондитерских\_изделий`, `Кондитер\_id\_Кондитера`, `Кондитер\_Кондитерская\_id\_Кондитерской`),

INDEX `fk\_Кондитерские\_изделия\_Кондитер\_idx` (`Кондитер\_id\_Кондитера` ASC, `Кондитер\_Кондитерская\_id\_Кондитерской` ASC) VISIBLE,

CONSTRAINT `fk\_Кондитерские\_изделия\_Кондитер1`

FOREIGN KEY (`Кондитер\_id\_Кондитера` , `Кондитер\_Кондитерская\_id\_Кондитерской`)

REFERENCES `Konditerskaya`.`Кондитер` (`id\_Кондитера` , `Кондитерская\_id\_Кондитерской`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Менеджер`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Менеджер` (

`id\_Менеджера` INT NOT NULL,

`ФИО\_Менеджера` VARCHAR(60) NULL,

`Кондитерская\_id\_Кондитерской` INT NOT NULL,

PRIMARY KEY (`id\_Менеджера`),

INDEX `fk\_Менеджер\_Кондитерская1\_idx` (`Кондитерская\_id\_Кондитерской` ASC) VISIBLE,

CONSTRAINT `fk\_Менеджер\_Кондитерская1`

FOREIGN KEY (`Кондитерская\_id\_Кондитерской`)

REFERENCES `Konditerskaya`.`Кондитерская` (`id\_Кондитерской`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Клиент`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Клиент` (

`id\_Клиента` INT NOT NULL,

`ФИО\_Клиента` VARCHAR(45) NULL,

`Номер\_телефона\_клиента` VARCHAR(11) NULL,

PRIMARY KEY (`id\_Клиента`))

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Меню`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Меню` (

`id\_Меню` INT NOT NULL,

`Цена\_продукции` INT NULL,

`Менюcol` VARCHAR(45) NULL,

`Время\_приготовления` TIME NULL,

`Название\_продукции` VARCHAR(45) NULL,

PRIMARY KEY (`id\_Меню`))

ENGINE = InnoDB;

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

-- Table `Konditerskaya`.`Заказ`

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

CREATE TABLE IF NOT EXISTS `Konditerskaya`.`Заказ` (

`Номер\_заказа` INT NOT NULL,

`Дата\_заказа` DATE NULL,

`Сумма\_заказа` INT NULL,

`Менеджер\_id\_Менеджера` INT NOT NULL,

`Кондитерские\_изделия\_id\_Кондитерских\_изделий` INT NOT NULL,

`Клиент\_id\_Клиента` INT NOT NULL,

`Меню\_id\_Меню` INT NOT NULL,

PRIMARY KEY (`Номер\_заказа`, `Менеджер\_id\_Менеджера`, `Кондитерские\_изделия\_id\_Кондитерских\_изделий`, `Клиент\_id\_Клиента`, `Меню\_id\_Меню`),

INDEX `fk\_Заказ\_Менеджер1\_idx` (`Менеджер\_id\_Менеджера` ASC) VISIBLE,

INDEX `fk\_Заказ\_Кондитерские\_изделия1\_idx` (`Кондитерские\_изделия\_id\_Кондитерских\_изделий` ASC) VISIBLE,

INDEX `fk\_Заказ\_Клиент1\_idx` (`Клиент\_id\_Клиента` ASC) VISIBLE,

INDEX `fk\_Заказ\_Меню1\_idx` (`Меню\_id\_Меню` ASC) VISIBLE,

CONSTRAINT `fk\_Заказ\_Менеджер1`

FOREIGN KEY (`Менеджер\_id\_Менеджера`)

REFERENCES `Konditerskaya`.`Менеджер` (`id\_Менеджера`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Заказ\_Кондитерские\_изделия1`

FOREIGN KEY (`Кондитерские\_изделия\_id\_Кондитерских\_изделий`)

REFERENCES `Konditerskaya`.`Кондитерские\_изделия` (`id\_Кондитерских\_изделий`)

ON DELETE CASCADE

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Заказ\_Клиент1`

FOREIGN KEY (`Клиент\_id\_Клиента`)

REFERENCES `Konditerskaya`.`Клиент` (`id\_Клиента`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Заказ\_Меню1`

FOREIGN KEY (`Меню\_id\_Меню`)

REFERENCES `Konditerskaya`.`Меню` (`id\_Меню`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;