Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине «Программно-аппаратные средства хранения и обработки данных»

на тему «Создание, связывание и добавление данных в таблицу базы данных в среде MySQL»

Выполнили:

студенты группы 22ВВП1

Демин М. С.

Беляев Д. И.

Амиров И. Р.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2025

**Название**

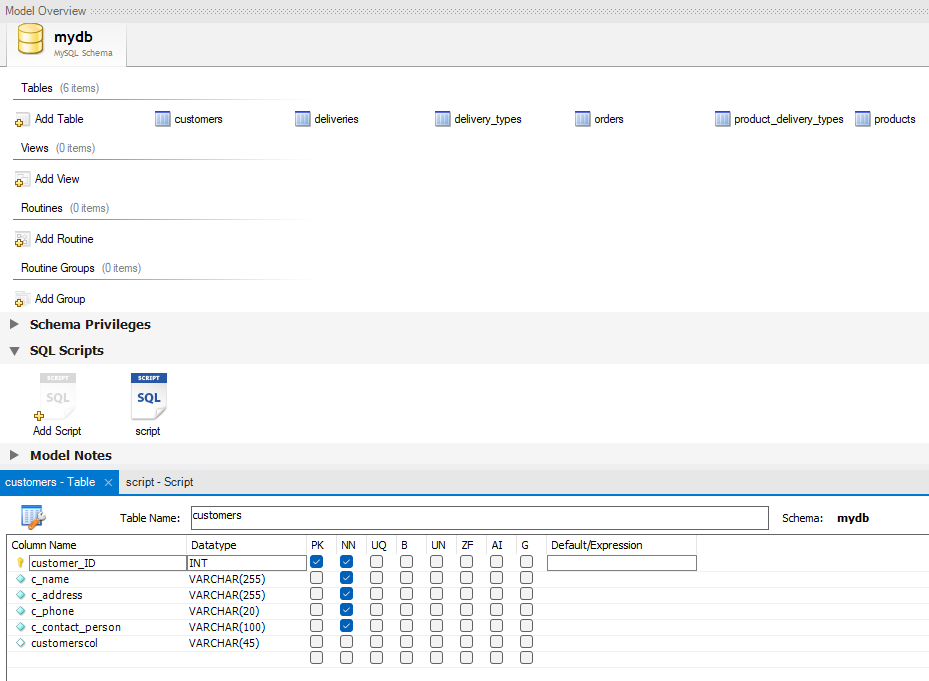
Создание, связывание и добавление данных в таблицу базы данных в среде MySQL

**Цель работы**

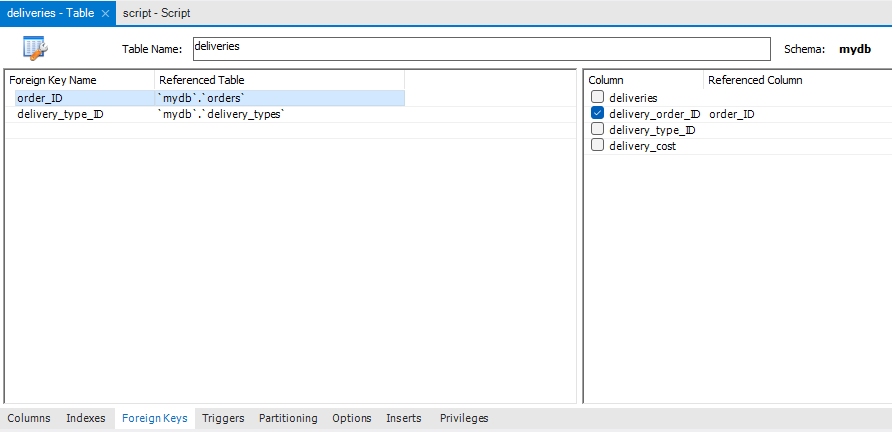
Научиться создавать таблицу в среде MySQL, добавлять, изменять и удалять данные из таблицы.

**Ход работы:**

1. Создали таблицу в MySQL из 1 лабораторной работы.



1. Создали foreign key



1. С помощью скрипта создали таблицу

**Листинг:**

-- MySQL Script generated by MySQL Workbench

-- Sat Feb 22 21:32:07 2025

-- Model: New Model Version: 1.0

-- 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 mydb

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

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

-- Schema mydb

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

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

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

-- Table `mydb`.`products`

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

CREATE TABLE IF NOT EXISTS `mydb`.`products` (

`product\_ID` INT NOT NULL,

`p\_name` VARCHAR(255) NOT NULL,

`p\_price` DECIMAL(10,2) NOT NULL,

`p\_info` TEXT NULL,

`p\_delivery\_available` TINYINT NOT NULL,

PRIMARY KEY (`product\_ID`))

ENGINE = InnoDB;

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

-- Table `mydb`.`customers`

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

CREATE TABLE IF NOT EXISTS `mydb`.`customers` (

`customer\_ID` INT NOT NULL,

`c\_name` VARCHAR(255) NOT NULL,

`c\_address` VARCHAR(255) NOT NULL,

`c\_phone` VARCHAR(20) NOT NULL,

`c\_contact\_person` VARCHAR(100) NOT NULL,

`customerscol` VARCHAR(45) NULL,

PRIMARY KEY (`customer\_ID`))

ENGINE = InnoDB;

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

-- Table `mydb`.`orders`

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

CREATE TABLE IF NOT EXISTS `mydb`.`orders` (

`order\_ID` INT NOT NULL,

`order\_customer\_ID` INT NOT NULL,

`order\_product\_ID` INT NOT NULL,

`order\_quantity` INT NOT NULL,

`order\_date` DATE NOT NULL,

PRIMARY KEY (`order\_ID`),

INDEX `customer\_ID\_idx` (`order\_customer\_ID` ASC) VISIBLE,

INDEX `product\_ID\_idx` (`order\_product\_ID` ASC) VISIBLE,

CONSTRAINT `customer\_ID`

FOREIGN KEY (`order\_customer\_ID`)

REFERENCES `mydb`.`customers` (`customer\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `product\_ID`

FOREIGN KEY (`order\_product\_ID`)

REFERENCES `mydb`.`products` (`product\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `mydb`.`delivery\_types`

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

CREATE TABLE IF NOT EXISTS `mydb`.`delivery\_types` (

`delivery\_type\_ID` INT NOT NULL,

`dt\_name` VARCHAR(100) NOT NULL,

`dt\_price` DECIMAL(10,2) NOT NULL,

`dt\_speed` VARCHAR(50) NOT NULL,

PRIMARY KEY (`delivery\_type\_ID`))

ENGINE = InnoDB;

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

-- Table `mydb`.`deliveries`

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

CREATE TABLE IF NOT EXISTS `mydb`.`deliveries` (

`deliveries` INT NOT NULL,

`delivery\_order\_ID` INT NOT NULL,

`delivery\_type\_ID` INT NULL,

`delivery\_cost` DECIMAL(10,2) NOT NULL,

PRIMARY KEY (`deliveries`),

INDEX `order\_ID\_idx` (`delivery\_order\_ID` ASC) VISIBLE,

INDEX `delivery\_type\_ID\_idx` (`delivery\_type\_ID` ASC) VISIBLE,

CONSTRAINT `order\_ID`

FOREIGN KEY (`delivery\_order\_ID`)

REFERENCES `mydb`.`orders` (`order\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `delivery\_type\_ID`

FOREIGN KEY (`delivery\_type\_ID`)

REFERENCES `mydb`.`delivery\_types` (`delivery\_type\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `mydb`.`product\_delivery\_types`

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

CREATE TABLE IF NOT EXISTS `mydb`.`product\_delivery\_types` (

`product\_ID` INT NOT NULL,

`delivery\_type\_ID` INT NOT NULL,

INDEX `product\_ID\_idx` (`product\_ID` ASC) VISIBLE,

INDEX `delivery\_type\_ID\_idx` (`delivery\_type\_ID` ASC) VISIBLE,

CONSTRAINT `product\_ID`

FOREIGN KEY (`product\_ID`)

REFERENCES `mydb`.`products` (`product\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `delivery\_type\_ID`

FOREIGN KEY (`delivery\_type\_ID`)

REFERENCES `mydb`.`delivery\_types` (`delivery\_type\_ID`)

ON DELETE CASCADE

ON UPDATE CASCADE)

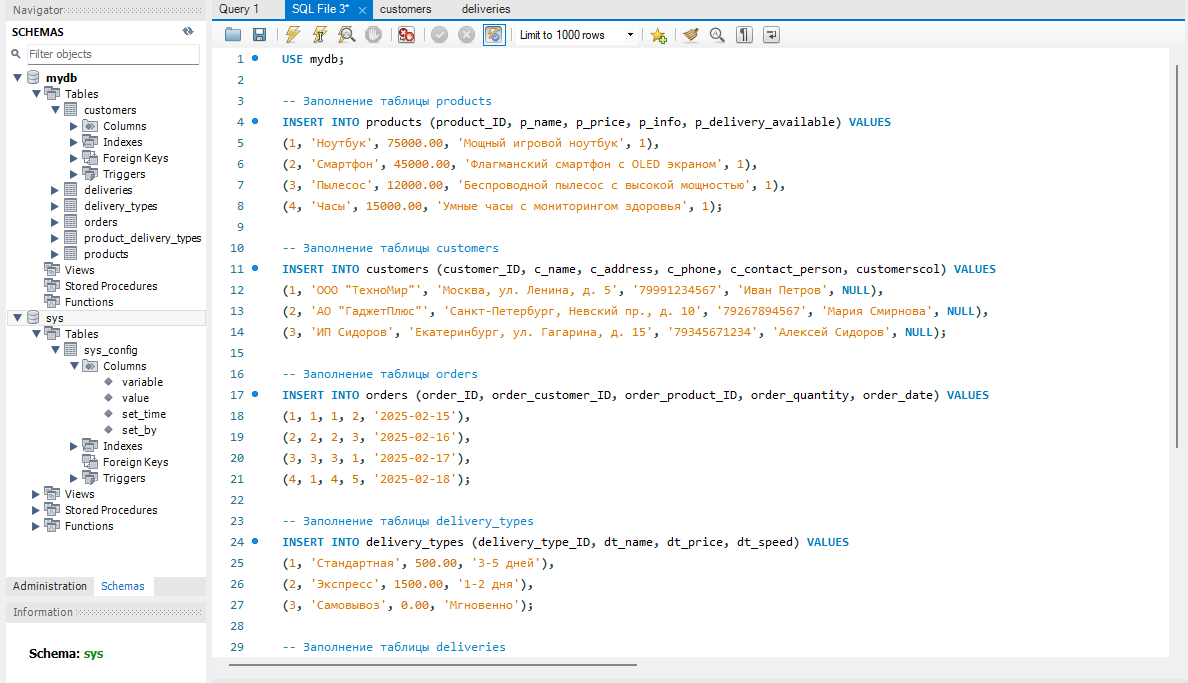
ENGINE = InnoDB;

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

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

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

1. Заполняем таблицу значениями

****

**Листинг:**

USE mydb;

-- Заполнение таблицы products

INSERT INTO products (product\_ID, p\_name, p\_price, p\_info, p\_delivery\_available) VALUES

(1, 'Ноутбук', 75000.00, 'Мощный игровой ноутбук', 1),

(2, 'Смартфон', 45000.00, 'Флагманский смартфон с OLED экраном', 1),

(3, 'Пылесос', 12000.00, 'Беспроводной пылесос с высокой мощностью', 1),

(4, 'Часы', 15000.00, 'Умные часы с мониторингом здоровья', 1);

-- Заполнение таблицы customers

INSERT INTO customers (customer\_ID, c\_name, c\_address, c\_phone, c\_contact\_person, customerscol) VALUES

(1, 'ООО "ТехноМир"', 'Москва, ул. Ленина, д. 5', '79991234567', 'Иван Петров', NULL),

(2, 'АО "ГаджетПлюс"', 'Санкт-Петербург, Невский пр., д. 10', '79267894567', 'Мария Смирнова', NULL),

(3, 'ИП Сидоров', 'Екатеринбург, ул. Гагарина, д. 15', '79345671234', 'Алексей Сидоров', NULL);

-- Заполнение таблицы orders

INSERT INTO orders (order\_ID, order\_customer\_ID, order\_product\_ID, order\_quantity, order\_date) VALUES

(1, 1, 1, 2, '2025-02-15'),

(2, 2, 2, 3, '2025-02-16'),

(3, 3, 3, 1, '2025-02-17'),

(4, 1, 4, 5, '2025-02-18');

-- Заполнение таблицы delivery\_types

INSERT INTO delivery\_types (delivery\_type\_ID, dt\_name, dt\_price, dt\_speed) VALUES

(1, 'Стандартная', 500.00, '3-5 дней'),

(2, 'Экспресс', 1500.00, '1-2 дня'),

(3, 'Самовывоз', 0.00, 'Мгновенно');

-- Заполнение таблицы deliveries

INSERT INTO deliveries (deliveries, delivery\_order\_ID, delivery\_type\_ID, delivery\_cost) VALUES

(1, 1, 1, 500.00),

(2, 2, 2, 1500.00),

(3, 3, 3, 0.00),

(4, 4, 1, 500.00);

-- Заполнение таблицы product\_delivery\_types

INSERT INTO product\_delivery\_types (product\_ID, delivery\_type\_ID) VALUES

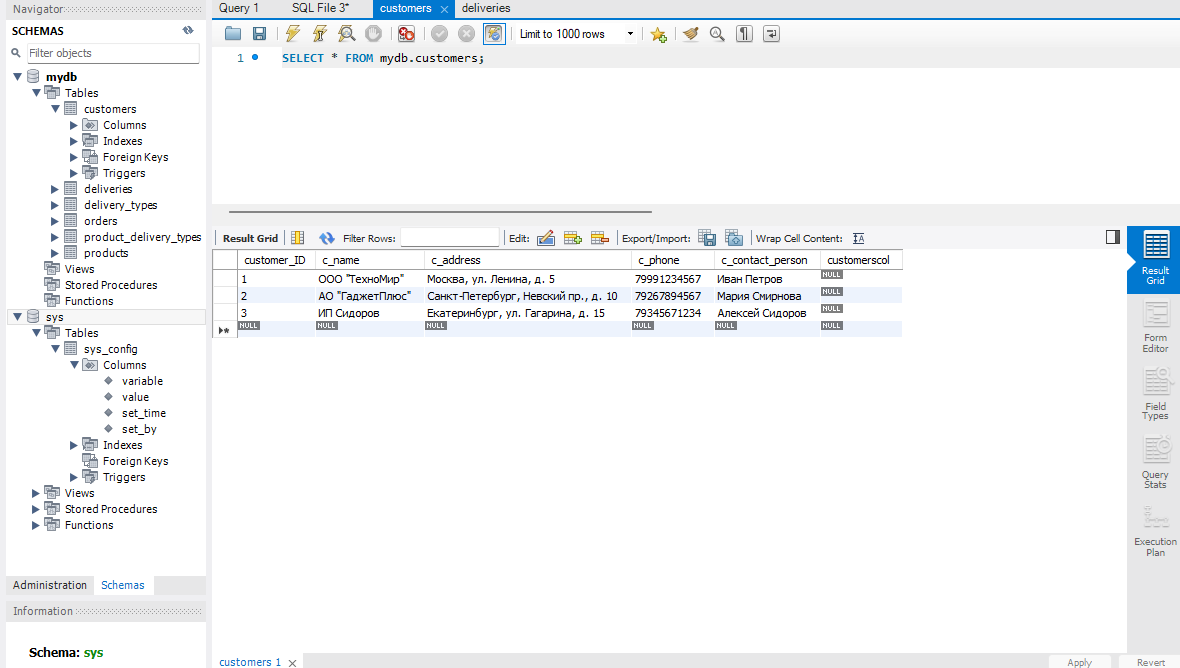
(1, 1), (1, 2),

(2, 1), (2, 2), (2, 3),

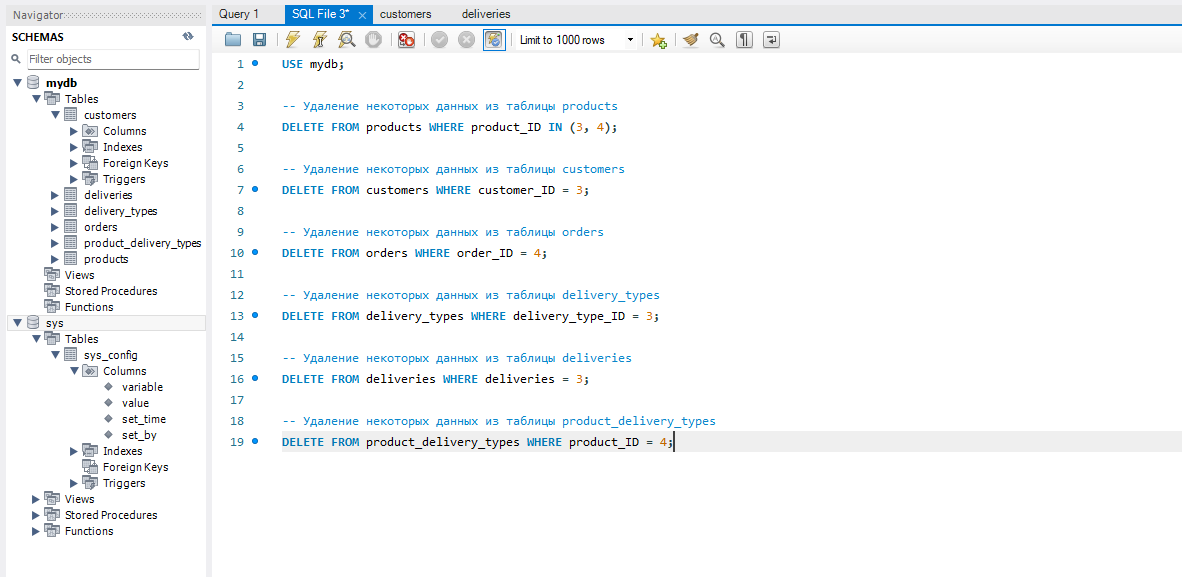
(3, 1),

(4, 2), (4, 3);

Результат: таблицы заполнились



Написали скрипт удаления данных



Листинг:

USE mydb;

-- Удаление некоторых данных из таблицы products

DELETE FROM products WHERE product\_ID IN (3, 4);

-- Удаление некоторых данных из таблицы customers

DELETE FROM customers WHERE customer\_ID = 3;

-- Удаление некоторых данных из таблицы orders

DELETE FROM orders WHERE order\_ID = 4;

-- Удаление некоторых данных из таблицы delivery\_types

DELETE FROM delivery\_types WHERE delivery\_type\_ID = 3;

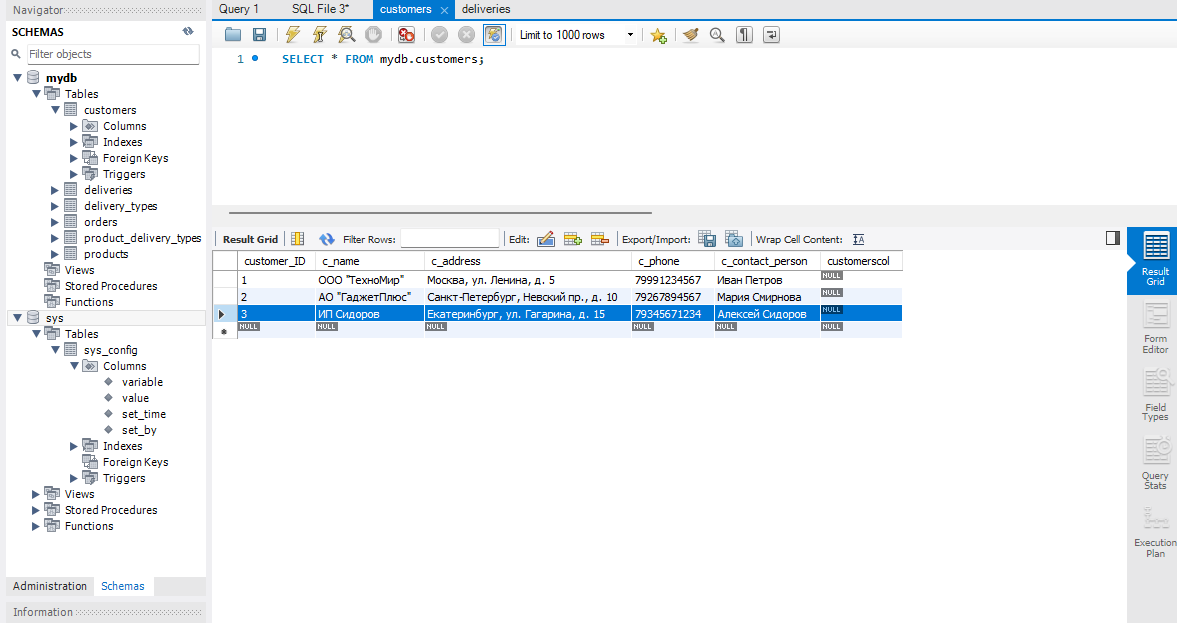
-- Удаление некоторых данных из таблицы deliveries

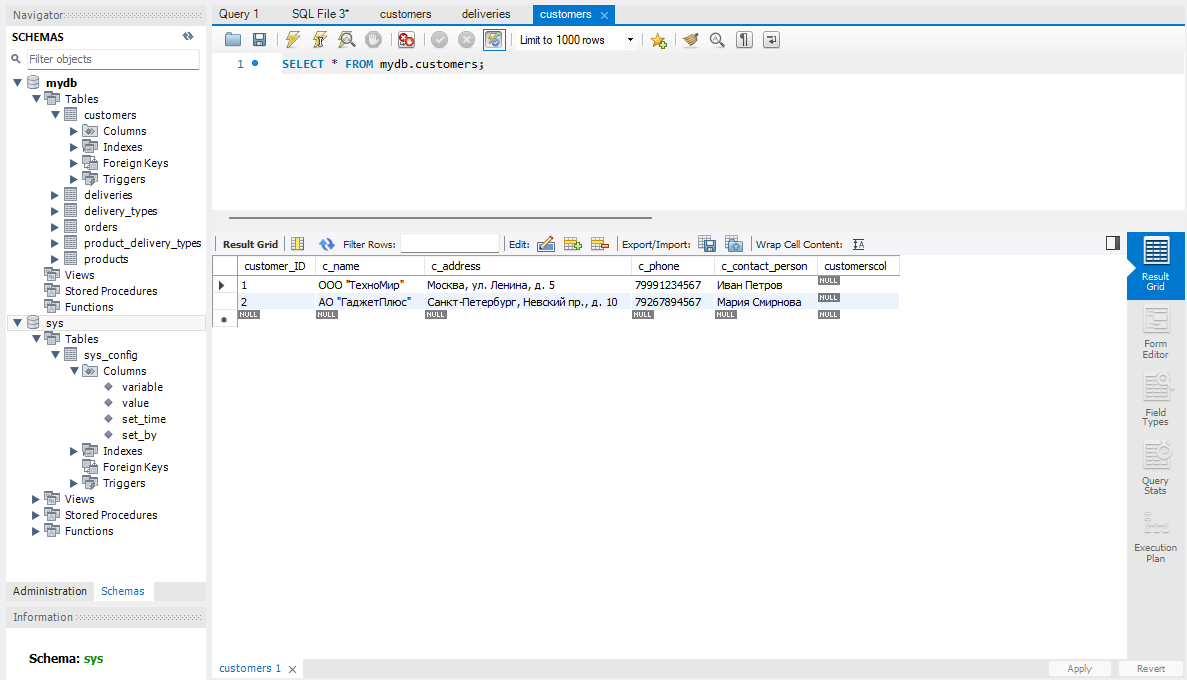
DELETE FROM deliveries WHERE deliveries = 3;

-- Удаление некоторых данных из таблицы product\_delivery\_types

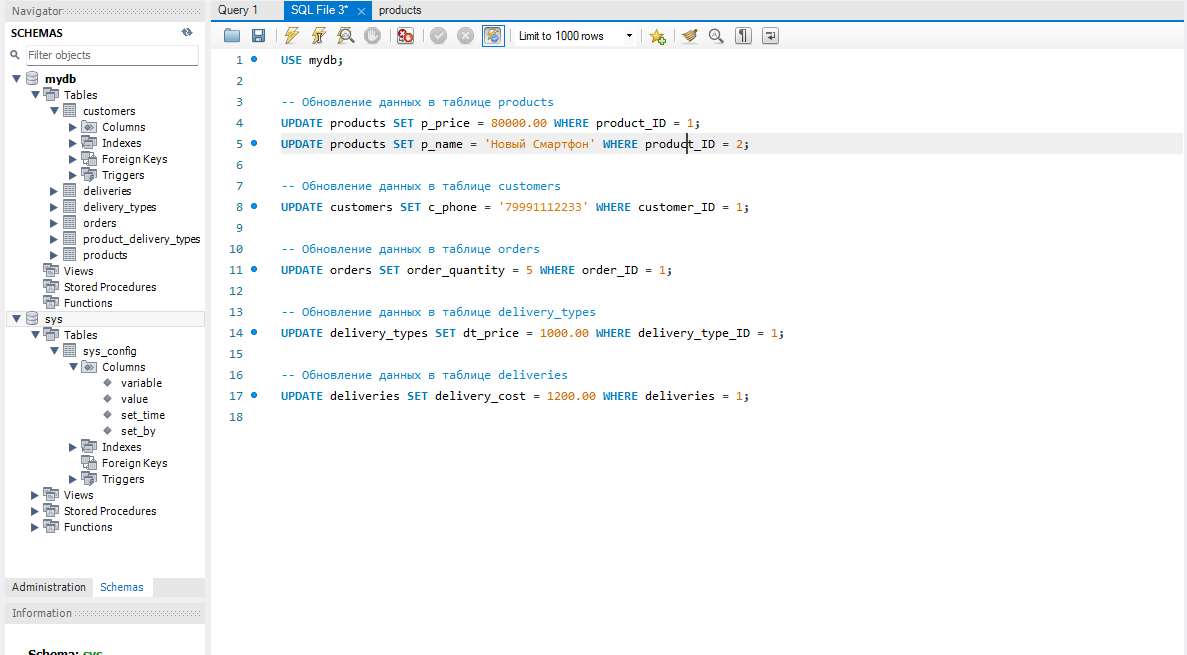
DELETE FROM product\_delivery\_types WHERE product\_ID = 4;

Пример работы: удалили из таблицы customers customer\_ID = 3





Написали скрипт обновления данных



Листинг:

USE mydb;

-- Обновление данных в таблице products

UPDATE products SET p\_price = 80000.00 WHERE product\_ID = 1;

UPDATE products SET p\_name = 'Новый Смартфон' WHERE product\_ID = 2;

-- Обновление данных в таблице customers

UPDATE customers SET c\_phone = '79991112233' WHERE customer\_ID = 1;

-- Обновление данных в таблице orders

UPDATE orders SET order\_quantity = 5 WHERE order\_ID = 1;

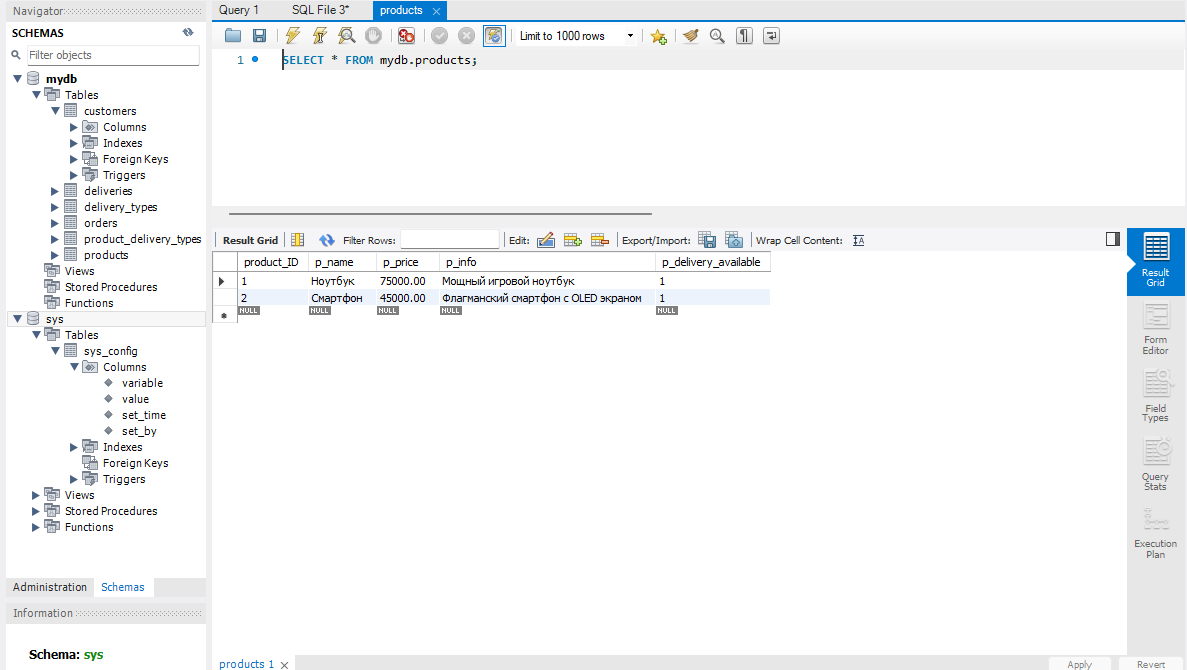
-- Обновление данных в таблице delivery\_types

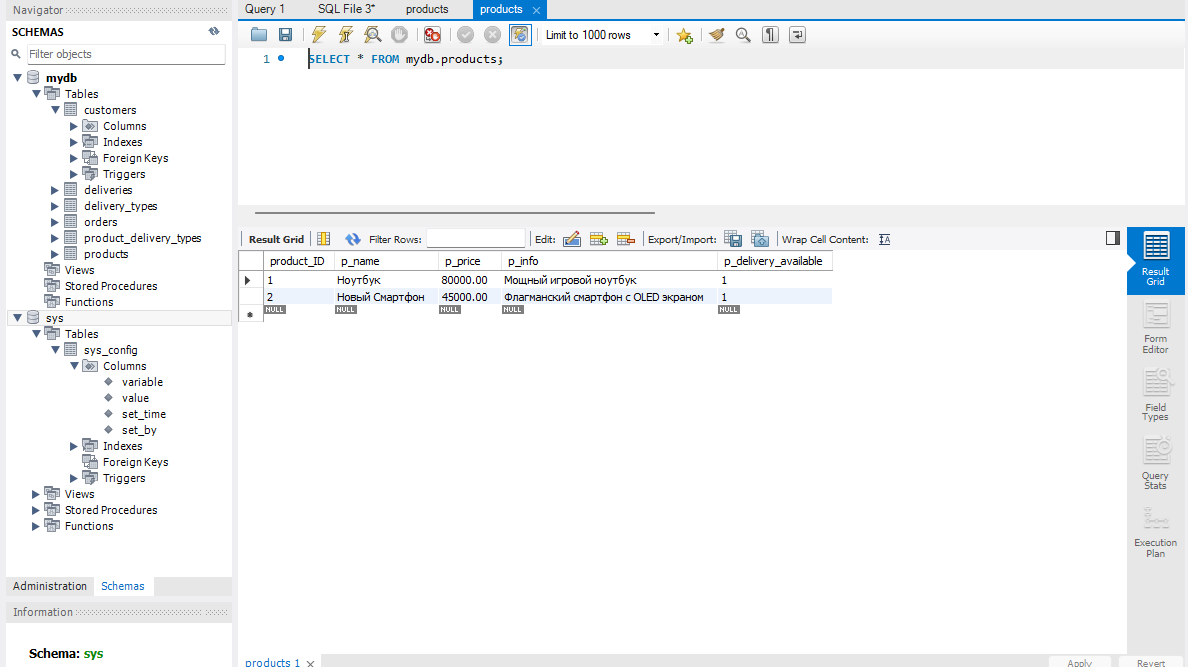
UPDATE delivery\_types SET dt\_price = 1000.00 WHERE delivery\_type\_ID = 1;

-- Обновление данных в таблице deliveries

UPDATE deliveries SET delivery\_cost = 1200.00 WHERE deliveries = 1;

Пример работы: обновили информацию в таблице products p\_price = 80000.00, p\_name = 'Новый Смартфон'.





**Вывод:**

Мы научились создавать таблицу в MySQL, добавлять и изменять данные в таблице, а также заполнять таблицу значениями