

СОДЕРЖАНИЕ

[1. Описание предметной области 2](#_Toc125795162)

[2. Логическая модель и физическая модель 3](#_Toc125795163)

[3. Создание БД SQL и реаливашя SQL-запросов в БД 11](#_Toc125795165)

[3.1 Работа в MySQL Command line 11](#_Toc125795166)

[3.2. Работа в MySQL Workbench 32](#_Toc125795167)

[ВЫВОДЫ 43](#_Toc125795168)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 44](#_Toc125795169)

1. **Описание предметной области**

Согласно статистическим данным, Российская Федерация находится на втором месте после США по количеству домашних животных, которое приходятся на численность населения. Это значит, что домашний питомец есть в каждой второй семье: и речь идет не только о привычных всем собаках и кошках, но и об экзотических животных – горностаях, ящерицах, шиншиллах. Вот почему вам стоит изучить бизнес-план открытия зоомагазина с расчетами. Данный инвестиционный проект будет актуален для крупных и маленьких городов.

Таким образом, магазины кормов для животных были построены и созданы для обеспечения кормом для домашних животных, и этот рынок будет расти в больших масштабах.

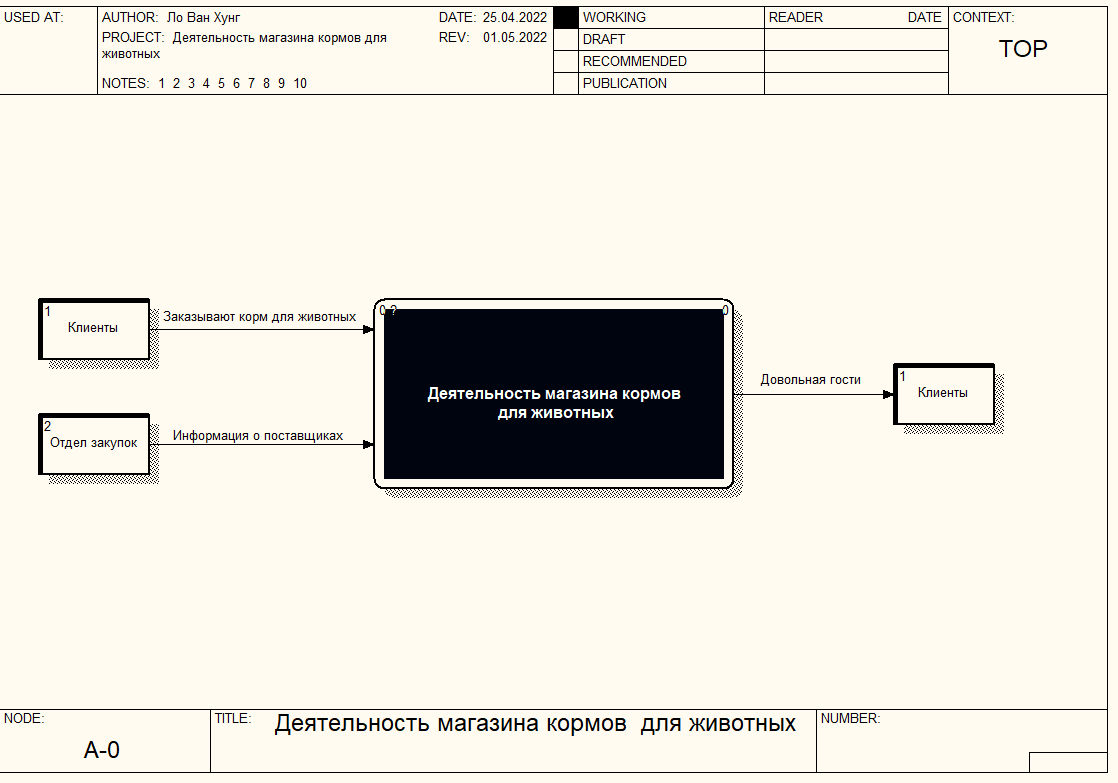
В настоящее время успешное функционирование различных фирм, организаций и предприятий просто не возможно без развитой информационной системы, которая позволяет автоматизировать сбор и обработку данных. Обычно для хранения и доступа к данным, содержащим сведения о некоторой предметной области, создается база данных.

База данных (БД) -- именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

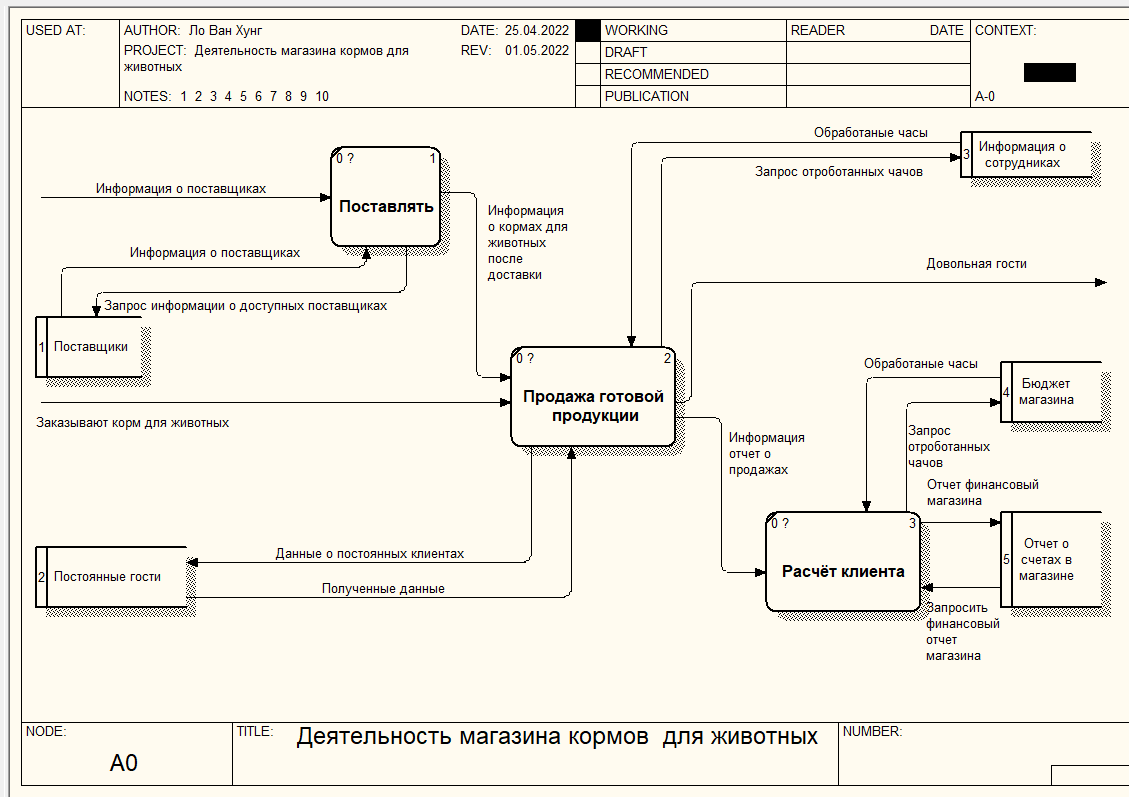
Предлагаемая база данных будет поддерживать работу магазина кормов для домашних животных по учету продаж и доставок различных товаров, база данных будет иметь тип базы данных управления бизнесом, так как будет обслуживать конкретный магазин, а не систему компаний в целом.

1. **Логическая модель и физическая модель**

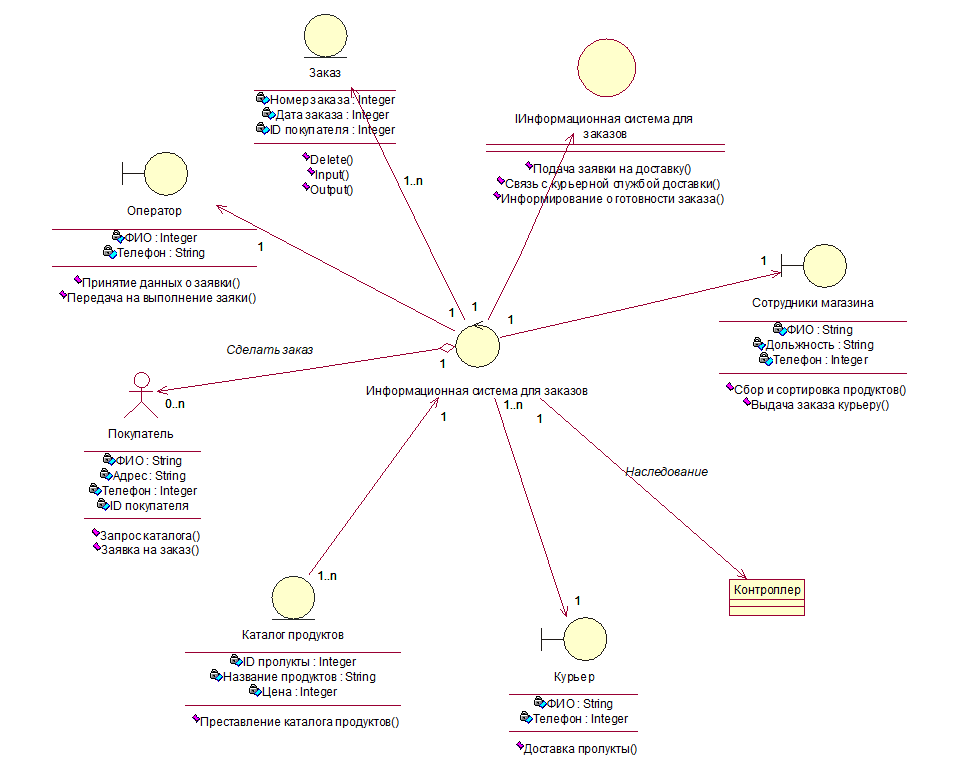
*Модель DFD по предметной области «Деятельность магазина кормов для животных».*



**Рисунок 1 – Контекстная диаграмма DFD**

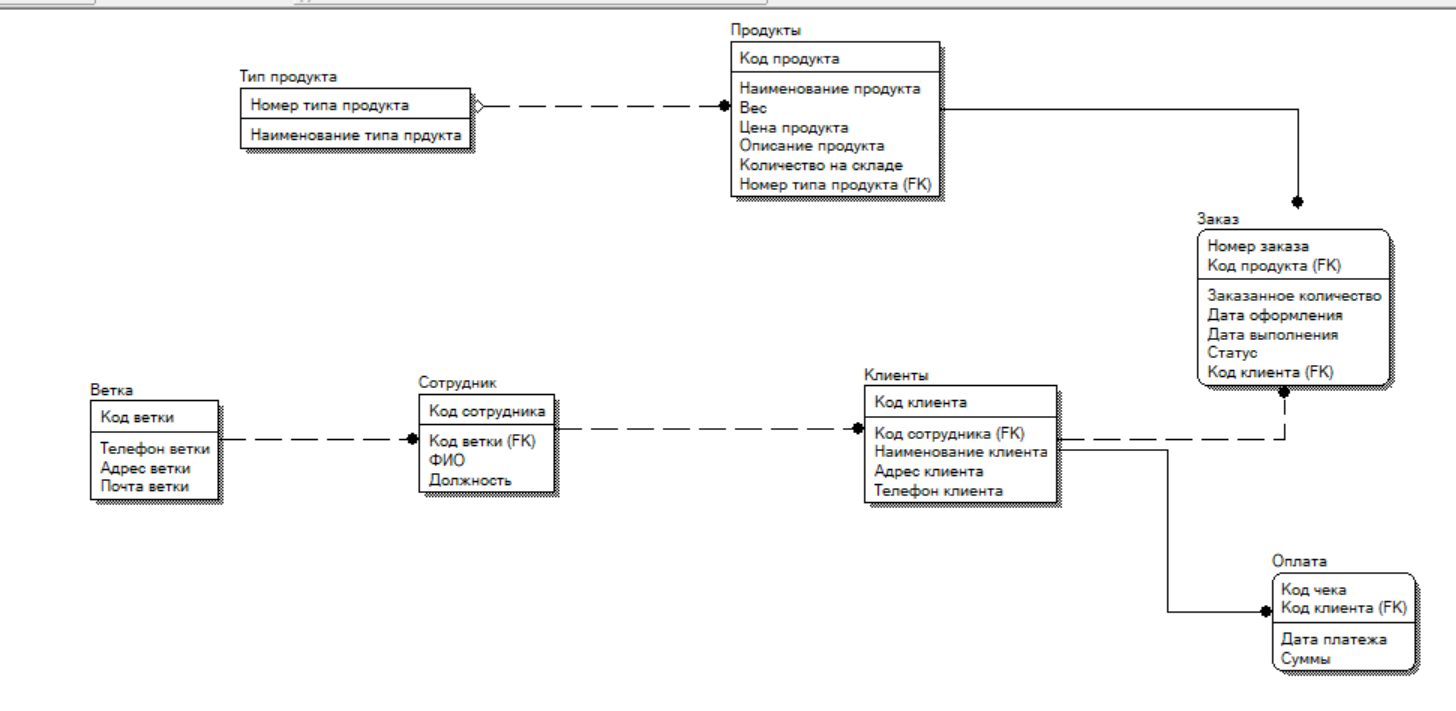


**Рисунок 2 - Декомпозиция контекстной диаграммы**

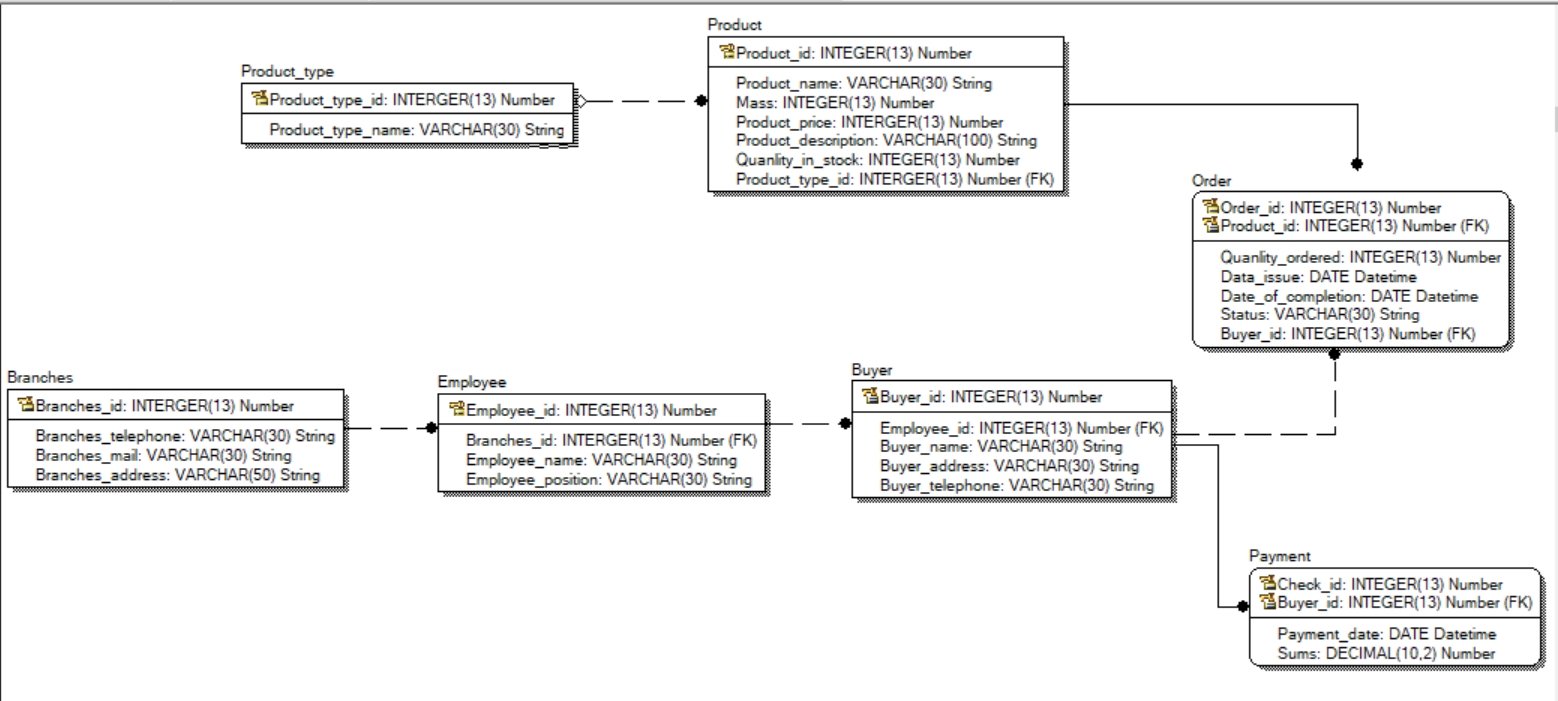


**Рисунок 3 – Диаграмма классов**

Из моделей, использованных и указанных в предыдущем семестре, Я создаю логических и физических моделей



**Рисунок 4 - Логическая модель база даных система управления отелями**



**Рисунок 5 - Физическая модель база даных система управления отелями**

2.1 Описание таблиц

Таблица 1 -Описание таблиц**ы** **branches**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| branches\_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация ветки |
| branches\_telephone | varchar(30) | - | - | - |  | номер телефона ветки |
| branches \_mail | varchar(30) | - | - | - |  | электонная почта ветки |
| branches\_address | varchar(50) | - | - | - |  | адрес ветки |

Таблица 2 -Описание таблиц**ы** **employee**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| employee \_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация  сотрудника |
| еmployee\_name | varchar(30) | - | - | - |  | имя сотрудника |
| еmployee**\_**position | varchar(30) | - | - | - |  | должность сотрудника |
| branches \_id | int(13) | + | - | - |  | идентификация  ветки |

Таблица 3 -Описание таблиц**ы** **buyer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| buyer \_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация  сотрудника |
| buyer\_name | varchar(30) | - | - | - |  | имя клиента |
| buyer\_address | varchar(30) | - | - | - |  | адрес клиента |
| buyer\_telephone | varchar(30) | - | - | - |  | номер телефона клиента |
| employee\_id | int(13) | + | - | - |  | идентификация  сотрудника |

Таблица 4 -Описание таблиц**ы** **product\_type**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| Product\_type \_id | int(13) | - | + | - | AUTO\_INCREMENT | Идентификация типа прдукта |
| Product\_type\_name | varchar(30) | - | - | - |  | Имя типа продукта |

Таблица 5 -Описание таблиц**ы** **product**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| product\_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация  продукта |
| product\_name | varchar(30) | - | - | - |  | имя продукта |
| producе\_mass | int(13) | - | - | - |  | вес продукта |
| product\_price | int(13) | - | - | - |  | цена продукта |
| product\_description | int(13) | - | - | - |  | описание продукта |
| quanlity\_in\_stock | int(13) | - | - | - |  | количество на складе |
| product\_type\_id | int(13) | + | - | - |  | идентификация типа прдукта |

Таблица 6 -Описание таблиц**ы** **orders**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| order\_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация  заказа |
| product\_id | int(13) | - | - | - |  | идентификация  продукта |
| quanlity\_ordered | int(13) | - | - | - |  | заказанное количество |
| data\_issue | date | - | - | - |  | дата оформления |
| date\_of\_completion | date | - | - | - |  | дата выполнения |
| status | varchar(30) | - | - | - |  | оисание |
| buyer\_id | int(13) | + | - | - |  | идентификация  сотрудника |

Таблица 7-Описание таблиц**ы** **payment**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип атрибута** | **FK** | **PK** | **NULL** | **DEFAULT** | **Оисание атрибута** |
| check\_id | int(13) | - | + | - | AUTO\_INCREMENT | идентификация  чека |
| payment\_date | date | - | - | - |  | дата оплаты |
| sums | decimal(10,2) | - | - | - |  | суммы |
| buyer\_id | int(13) | + | - | - |  | идентификация клиенты |

* 1. Спроектируйте таблицы базы данных

**Таблица 8 – Данные таблицы branches**

|  |  |  |  |
| --- | --- | --- | --- |
| **Branches\_id** | **Branches\_telephone** | **Branches \_mail** | **Branches\_address** |
| 1 | 89 777-910-001 | Petshopru01@gmail.com | Moscow, Ryazansky prospect, 16 |
| 2 | 89 651-567-529 | Petshopru02@gmail.com | Moscow, Belorechenskaya street, 38 |

**Таблица 9 – Данные таблицы employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee\_id** | **Employee\_name** | **Employee\_position** | **Branches\_id** |
| 1 | Hung | Manager | 1 |
| 2 | Leon | Manager | 2 |
| 3 | Takina | VP Marketer | 2 |
| 4 | Fugasi | Vp Sales | 1 |
| 5 | Andrey | Sales Rep | 1 |
| 6 | Anna | Sales Rep | 2 |
| 7 | Viktoria | Sales Rep | 1 |
| 8 | Sofia | Sales Rep | 2 |

**Таблица 10 – Данные таблицы buyer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Buyer\_id** | **Buyer\_name** | **Buyer\_telephone** | **Buyer\_address** | **Employee\_id** |
| 1 | Sancho | 89163326578 | Moscow | 5 |
| 2 | Elanga | 89124665814 | Moscow | 6 |
| 3 | Kiesa | 89667681421 | St.Petersburg | 7 |
| 4 | Ryad | 89641657549 | St.Petersburg | 5 |
| 5 | Foden | 89776543331 | Moscow | 7 |
| 6 | Kevin | 89461432871 | Kazan | 8 |
| 7 | Mason | 89251865621 | Kazan | 6 |
| 8 | Tenhag | 89016543210 | Tula | 5 |
| 9 | Furguson | 89019876587 | Ufa | 5 |
| 10 | Beckham | 89018964532 | Ufa | 8 |

**Таблица 11 – Данные таблицы product\_type**

|  |  |
| --- | --- |
| **Product\_type\_id** | **Product\_type\_name** |
| 1 | DRY PET FOODS |
| 2 | WET PET FOODS |

**Таблица 12 – Данные таблицы product**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Product\_id** | **Product\_name** | **Product\_mass(kg)** | **Product\_price($)** | **Product\_description** | **Quanlity\_in\_stock** | **Product\_type\_id** |
| 1 | Pedigree | 4 | 28 | Complete Nutrition Grilled Steak & Vegetable Flavor Dog Kibble Adult Dry Dog Food | 20 | 1 |
| 2 | Purina Pro Plan | 5 | 40 | Adult Sensitive Skin & Stomach Salmon & Rice Formula Dry Dog Food | 18 | 1 |
| 3 | Purina ONE | 5 | 42 | Natural SmartBlend Chicken & Rice Formula Dry Dog Food | 15 | 1 |
| 4 | Hill's Science Diet | 4 | 51 | Adult Perfect Weight Chicken Recipe Dry Dog Food | 15 | 1 |
| 5 | Kibble 'n Bits | 3 | 12 | Original Savory Beef & Chicken Flavors Dry Dog Food | 11 | 1 |
| 6 | Cesar | 2 | 22 | Classic Loaf in Sauce Beef Recipe, Filet Mignon, Grilled Chicken, Pack Dog Food | 31 | 2 |
| 7 | Blue Buffalo | 2 | 25 | Homestyle Recipes Adult Variety Pack Chicken & Beef Dinner Canned Dog Food | 28 | 2 |
| 8 | Hill's Prescription Diet | 3 | 51 | Hill's Prescription Diet | 14 | 2 |
| 9 | Rachael Ray Nutrish | 2 | 18 | Natural Variety Pack Wet Dog Food | 10 | 2 |
| 10 | Iams | 3 | 22 | ProActive Health Classic Ground with Chicken & Whole Grain Rice Adult Wet Dog Food | 11 | 2 |

**Таблица 13– Данные таблицы orders**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Order\_id** | **Product\_id** | **Quantity ordered** | **Data\_issue** | **Date\_of\_completion** | **Statuc** | **Buyer\_id** |
| 1 | 1 | 1 | 2022-12-10 | 2022-12-11 | Performance | 1 |
| 2 | 3 | 2 | 2022-12-11 | 2022-12-11 | Performance | 3 |
| 3 | 5 | 3 | 2022-12-11 | 2022-12-11 | Performance | 2 |
| 4 | 6 | 1 | 2022-12-12 | 2022-12-13 | Performance | 4 |
| 5 | 6 | 1 | 2022-12-12 | 2022-12-13 | Performance | 8 |
| 6 | 1 | 2 | 2022-12-13 | 2022-12-13 | Performance | 9 |
| 7 | 3 | 3 | 2022-12-13 | 2022-12-14 | Performance | 10 |
| 8 | 2 | 1 | 2022-12-13 | 2022-12-14 | Performance | 7 |
| 9 | 8 | 3 | 2022-12-14 | 2022-12-15 | Performance | 5 |
| 10 | 9 | 2 | 2022-12-15 | 2022-12-16 | Performance | 6 |

**Таблица 14 – Данные таблицы payment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Check\_id** | **Buyer\_id** | **Payment\_date** | **Sums** |
| 1 | 1 | 2022-12-10 | 28 |
| 2 | 3 | 2022-12-11 | 84 |
| 3 | 2 | 2022-12-11 | 36 |
| 4 | 4 | 2022-12-12 | 22 |
| 5 | 8 | 2022-12-12 | 11 |
| 6 | 9 | 2022-12-13 | 56 |
| 7 | 10 | 2022-12-13 | 126 |
| 8 | 7 | 2022-12-13 | 40 |
| 9 | 5 | 2022-12-14 | 153 |
| 10 | 6 | 2022-12-15 | 36 |

1. Создание БД SQL и реаливашя SQL-запросов в БД
   1. Работа в MySQL Command line

3.1.1 Снимки экрана (скриншоты) процесса разработки

Cоздание new database (рисунок 6)

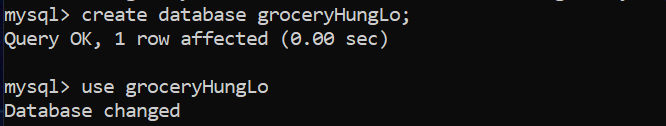


Рисунок 6 - Cоздание новую базу данных

Создание таблицы: branches, employee, product\_type, product, buyer, orders, payment (рисунок 7 - 13).

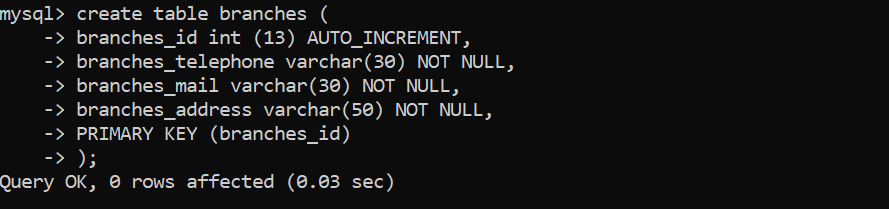


Рисунок 7 - Cоздание таблицу «branches»

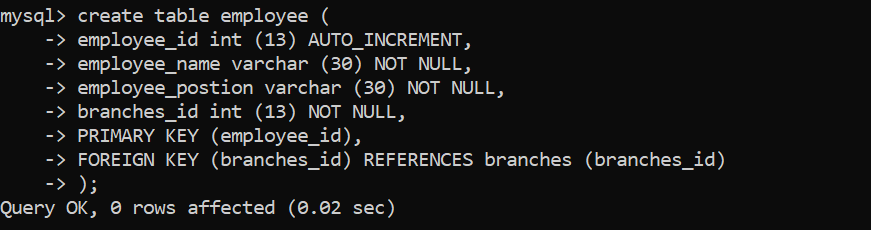


Рисунок 8 - Cоздание таблицу «employee»

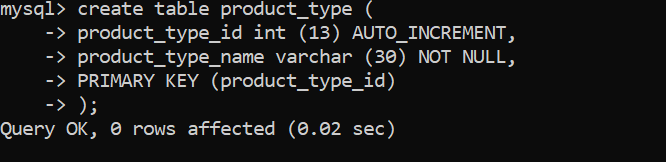


Рисунок 9 - Cоздание таблицу «product\_type»



Рисунок 10 - Cоздание таблицу «product»

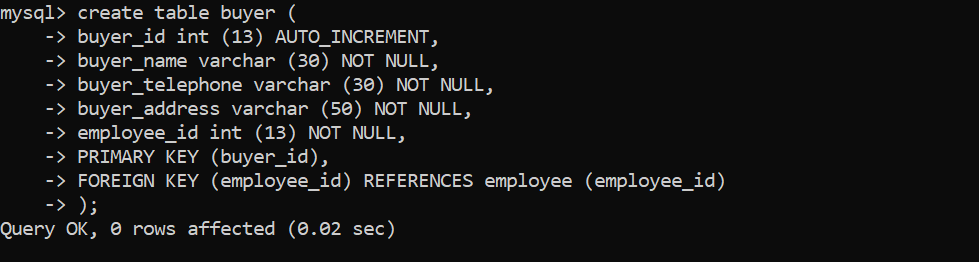


Рисунок 11 - Cоздание таблицу «buyer»



Рисунок 12 - Cоздание таблицу «orders»

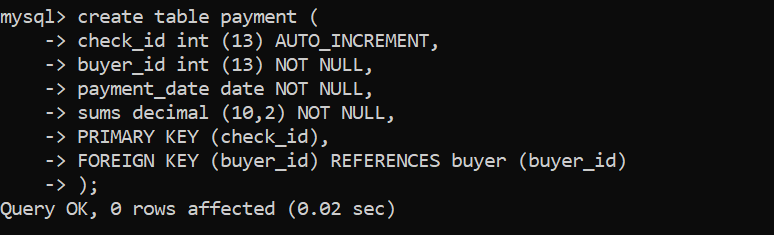


Рисунок 13 - Cоздание таблицу «payment»

* + 1. Снимок экрана завершенной базы данных

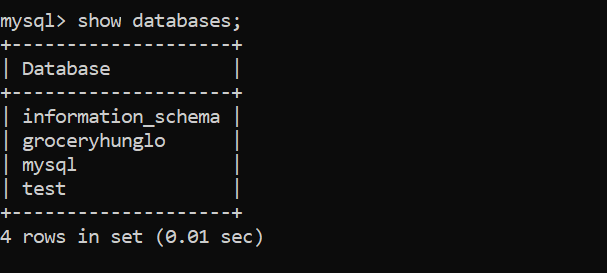


Рисунок 14 - Показать созданные базы данных

* + 1. Снимок экрана (скриншоты) списка созданных таблиц

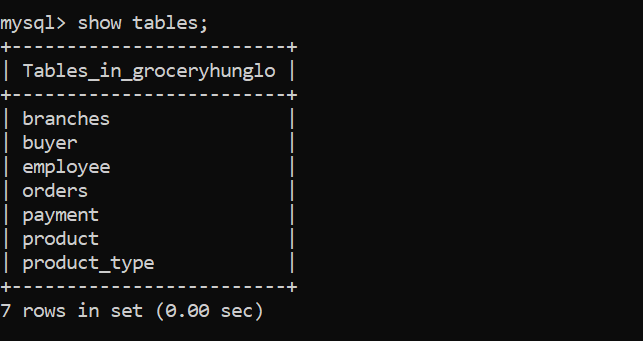
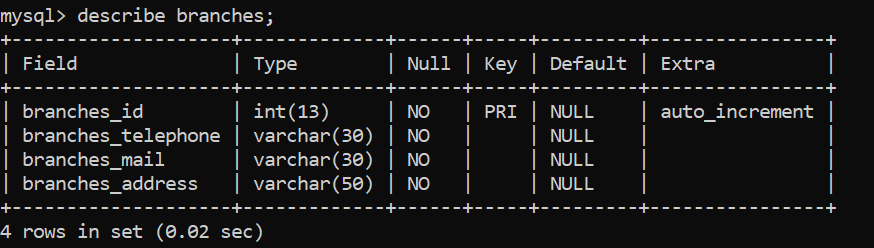
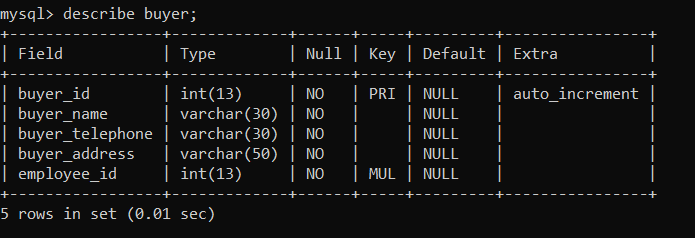


Рисунок 15 - Показать таблицы

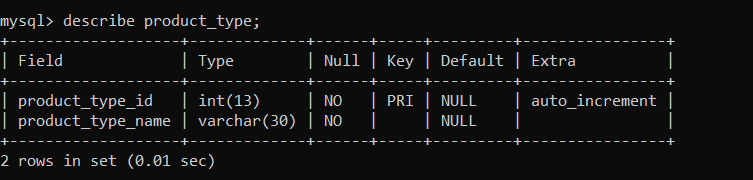
* + 1. Снимок экрана (скриншоты) структуры созданных таблиц **Рисунок 16 - Описание таблицы «branches»**



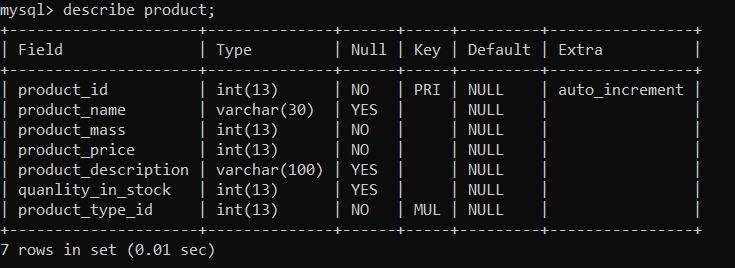
**Рисунок 17 - Описание таблицы «employee»**



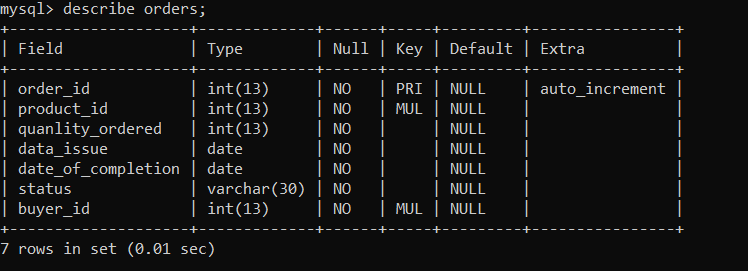
**Рисунок 18 - Описание таблицы «buyer»**



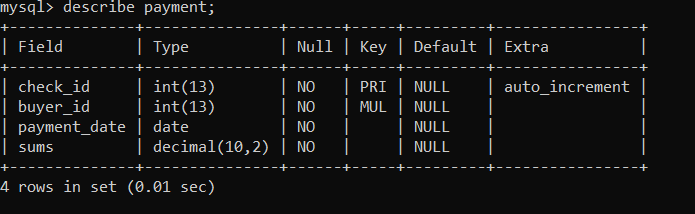
**Рисунок 19 - Описание таблицы «product\_type»**



**Рисунок 20-Описание таблицы «product»**



**Рисунок 21-Описание таблицы «orders»**



**Рисунок 22-Описание таблицы «payment»**

* + 1. Снимки экрана (скриншоты) содержания созданных таблиц

Создать исходные данные для таблиц: branches, employee, product\_type, product, buyer, orders, payment (рисунок 23 - 29).



Рисунок 23 – Создание данные в таблице «branches»

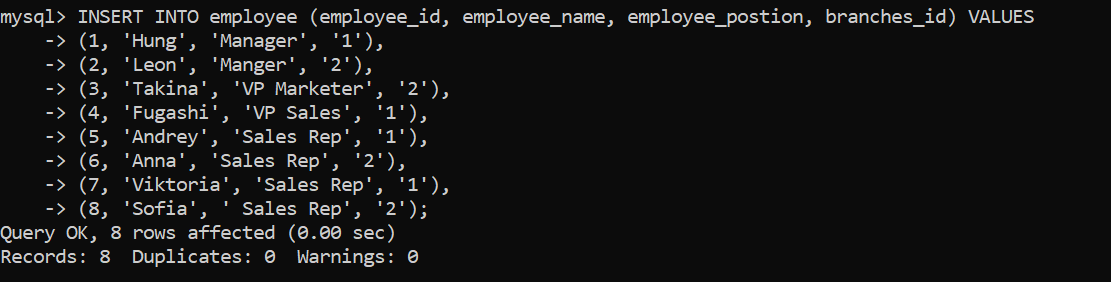


Рисунок 24- Создание данные в таблице «employee»

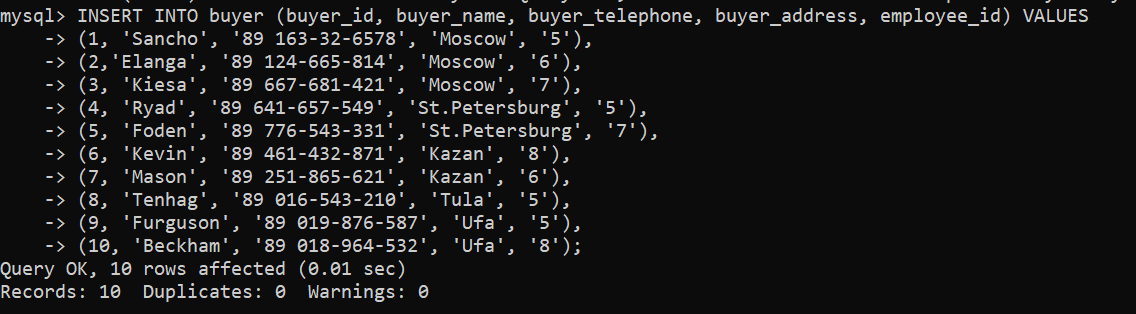


Рисунок 25 - Создание данные в таблице «buyer»

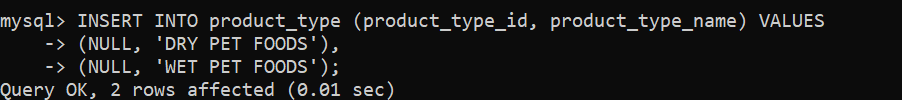


Рисунок 26 - Создание данные в таблице «product\_type»

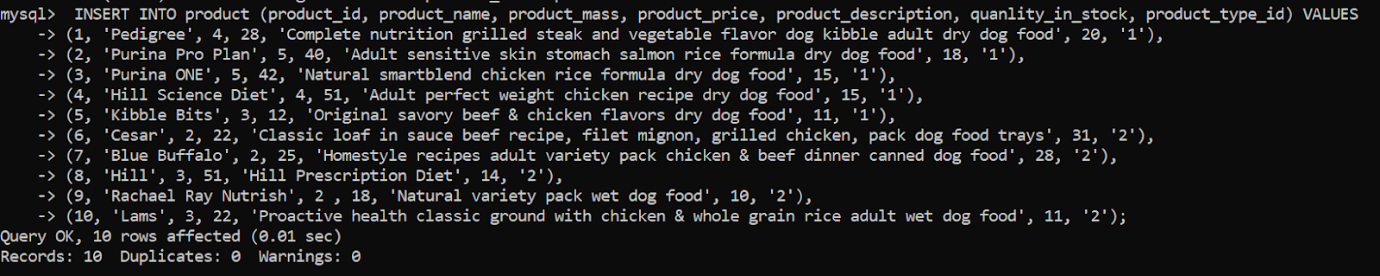


Рисунок 27 - Создание данные в таблице «product»

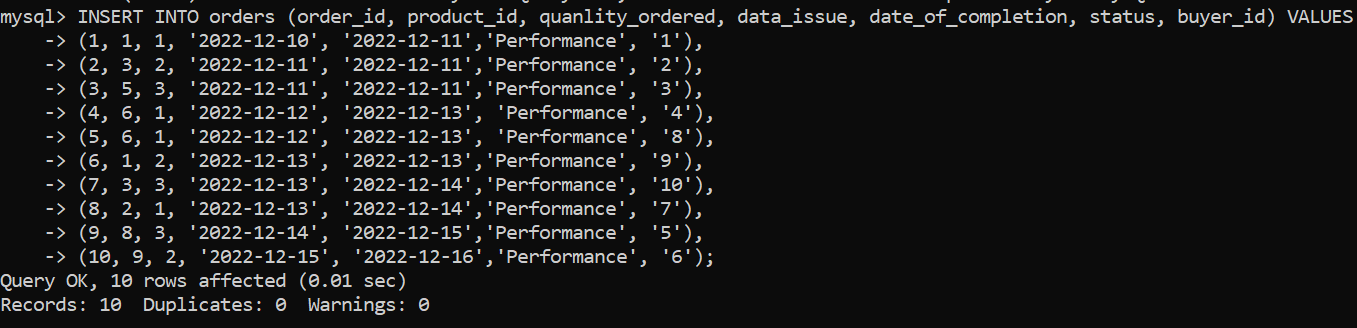


Рисунок 28- Создание данные в таблице «orders»

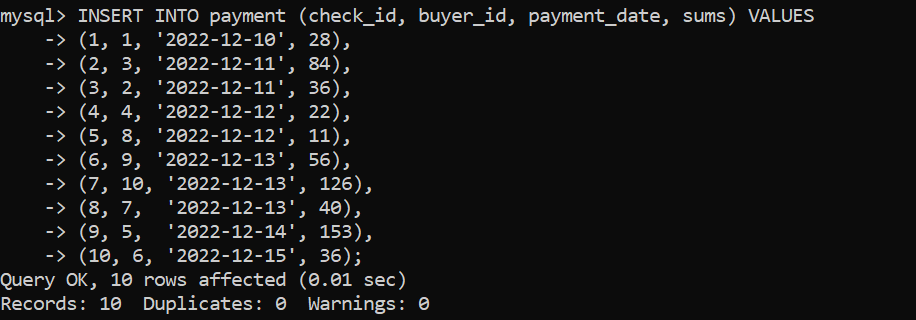


Рисунок 29 - Создание данные в таблице «payment»

Отображение информации таблицы: branches, employee, product\_type, product, buyer, orders, payment (рисунок 30 - 36).

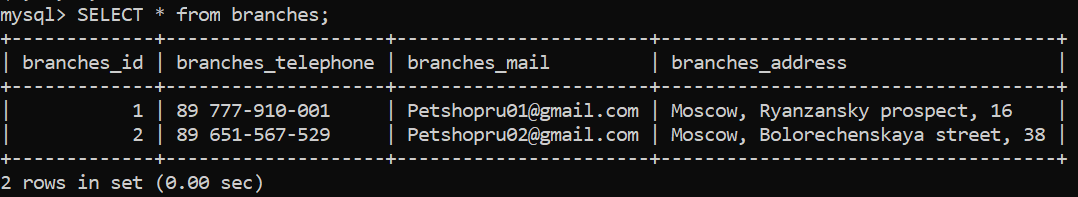


Рисунок 30- Отображение данные в таблице «branches»

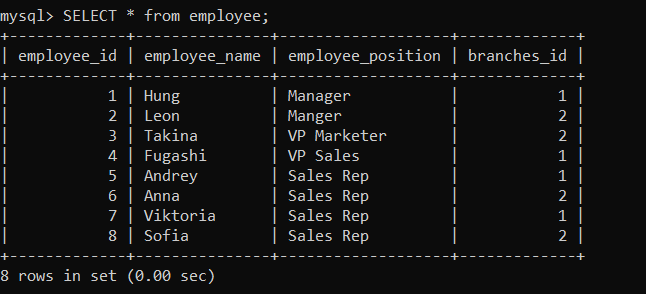


Рисунок 31- Отображение данные в таблице «employee»

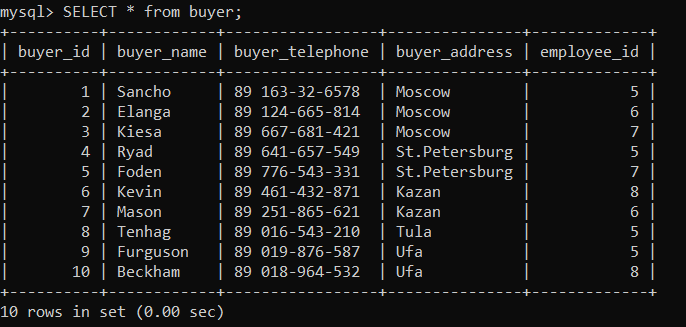


Рисунок 32- Отображение данные в таблице «buyer»

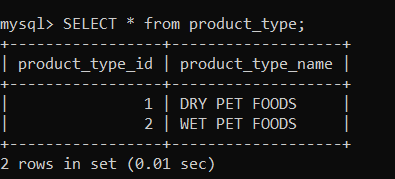


Рисунок 33- Отображение данные в таблице «product\_type»

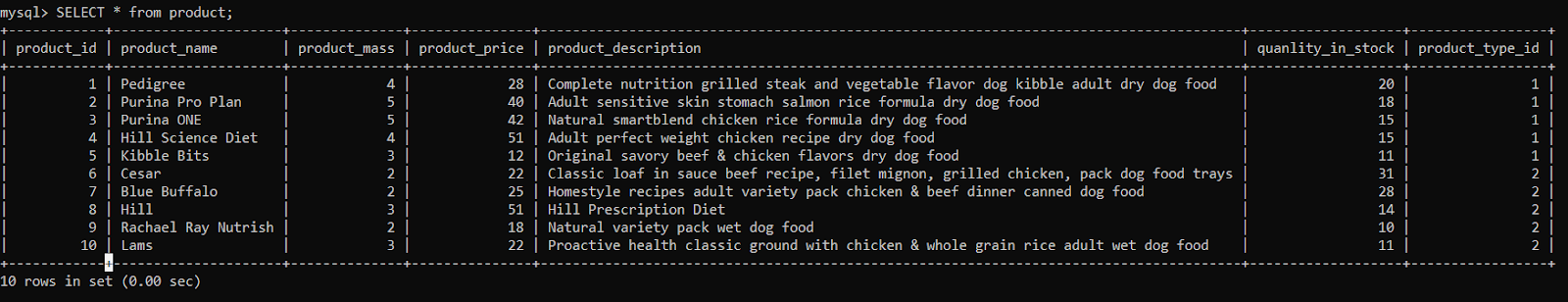


Рисунок 34- Отображение данные в таблице «product»

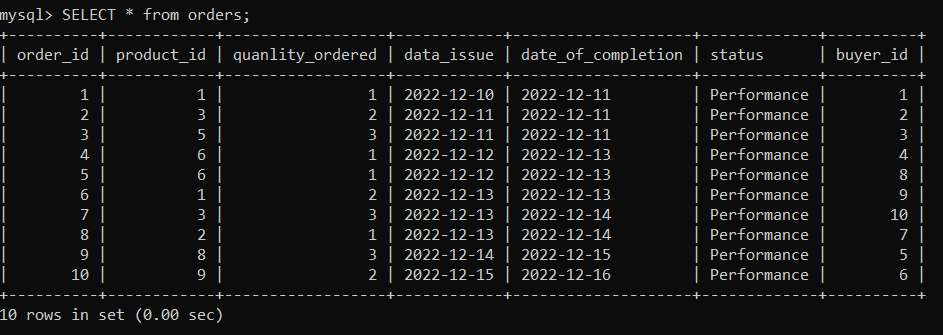


Рисунок 35- Отображение данные в таблице «orders»

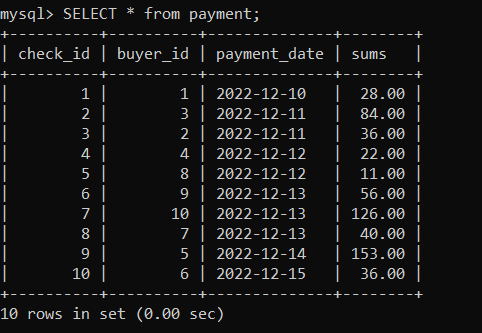
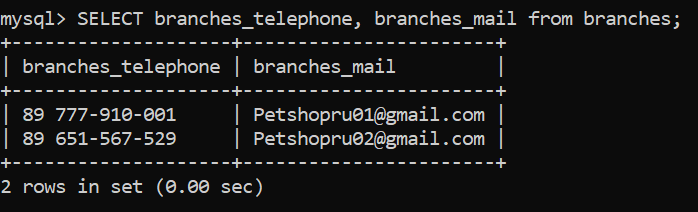


Рисунок 36- Отображение данные в таблице «payment»

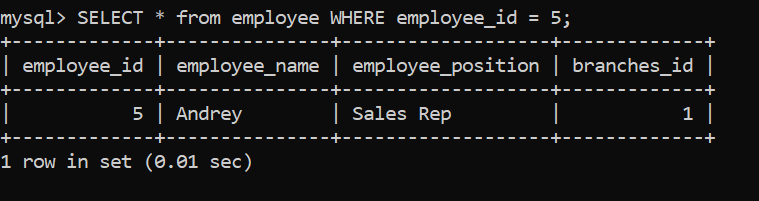
* + 1. Снимки экрана (скриншоты) выборки данных по различным параметрам (по каждому оператору)

Запросы с использованием этих операторов: =, >, <, >=, =<, !=, IS NOT NULL, IS NULL, BETWEEN, IN, NOT IN, LIKE, NOT LIKE.



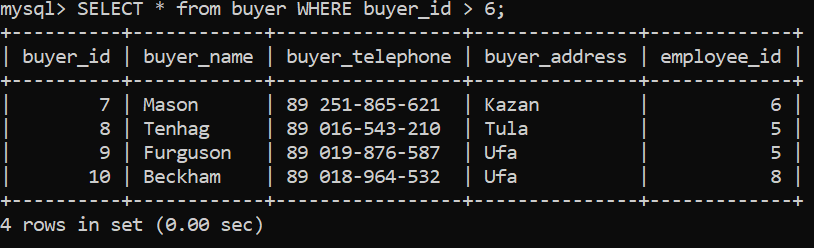
**Рисунок 37 - Распечатать номер телефона и электронной почты всех ветков**

Оператор: = (равно)



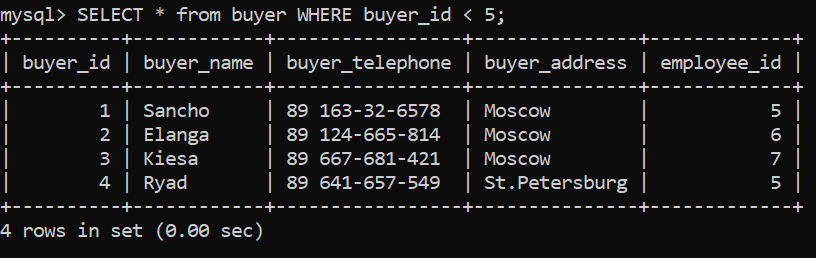
**Рисунок 38 - Распечатать информацию о сотруднике с идентификатором равно 5**

Оператор: > (больше)



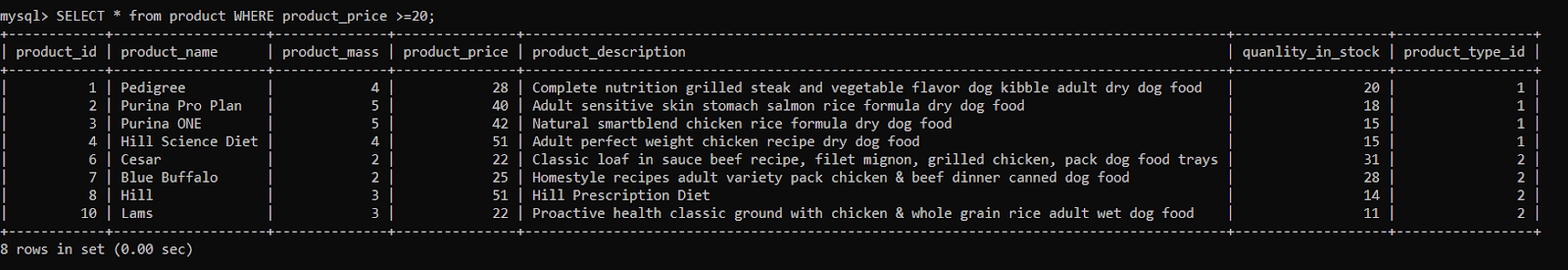
**Рисунок 39 - Распечатать информацию о клиенте с идентификатором больше 6**

Оператор: < (меньше)



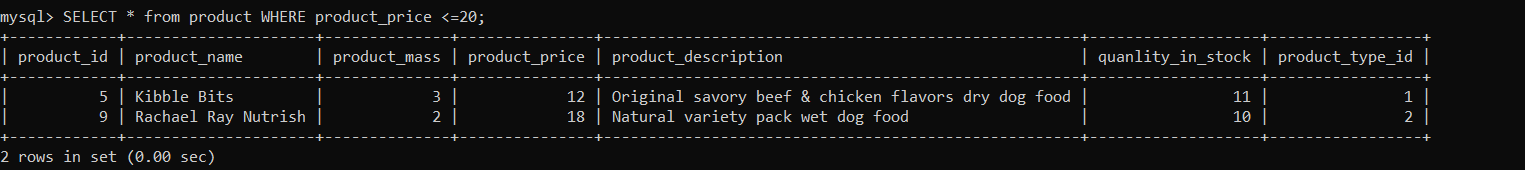
**Рисунок 40 - Распечатать информацию о клиенте с идентификатором меньше 5**

Оператор: >= (больше или равно)



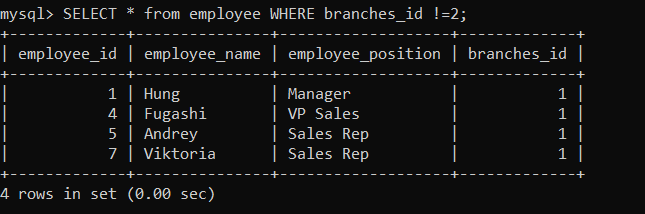
**Рисунок 41 - Распечатать информацию о продукте с product\_price больше или равно**  **20**

Оператор: <= (меньше или равно)



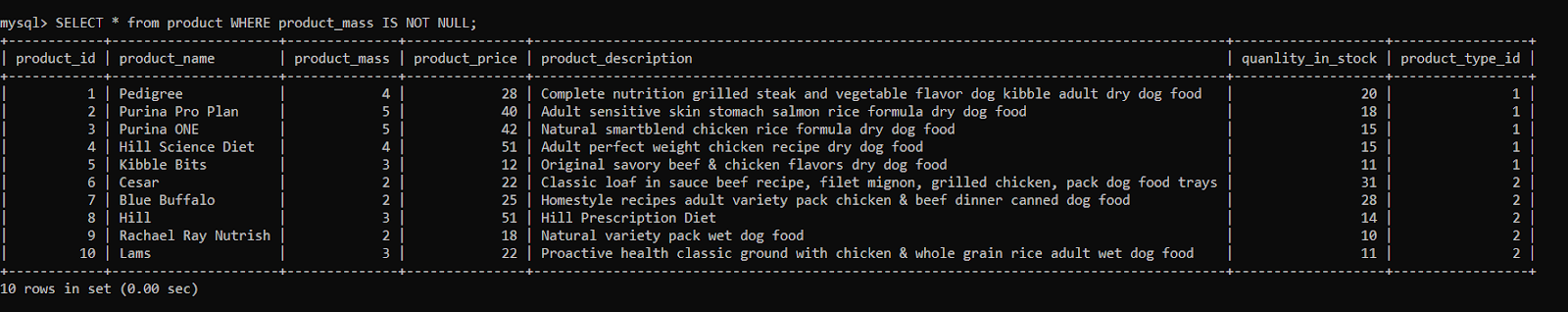
**Рисунок 41 - Распечатать информацию о продукте с product\_price меньше или равно**  **20**

Оператор: != (не равно)



**Рисунок 42 - Распечатать информацию о сотруднике с branches\_id кроме 2**

Оператор: IS NOT NULL



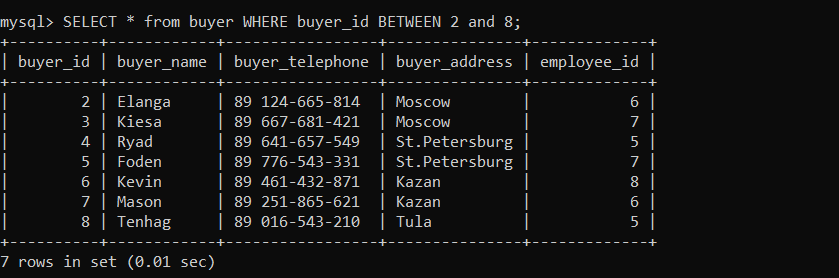
**Рисунок 43 - Распечатать информацию о продукте с products\_mass номера не NULL**

Оператор: IS NULL



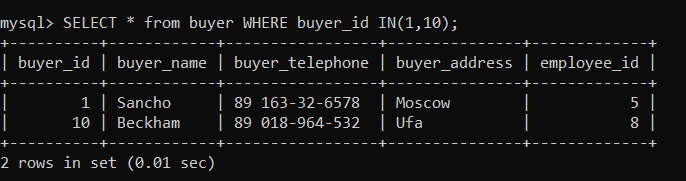
**Рисунок 44 - Распечатать информацию products\_price NULL**

Оператор: BETWEEN(между)

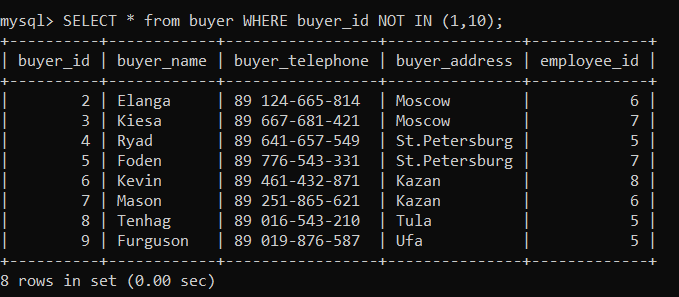


**Рисунок 45 - Распечатать информацию о клиенте с buyer\_id от 2 до 8**

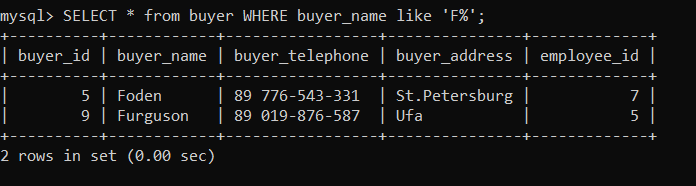
Оператор: IN (значение содержится)



**Рисунок 46 - Распечатать информацию о клиенте с buyer\_id -** имеет значение 1, 10

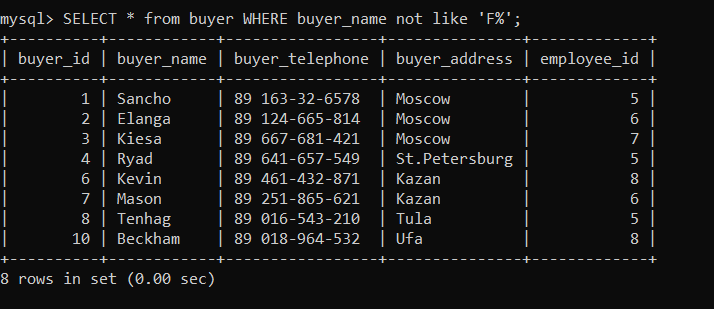


**Рисунок 47 - Распечатать информацию о клиенте с buyer\_id - другое** значение 1, 10

Оператор: LIKE (соответствие) 

**Рисунок 48 - Распечатать информацию о клиенте с buyer\_name – как F**

Оператор: NOT LIKE (не соответствие)



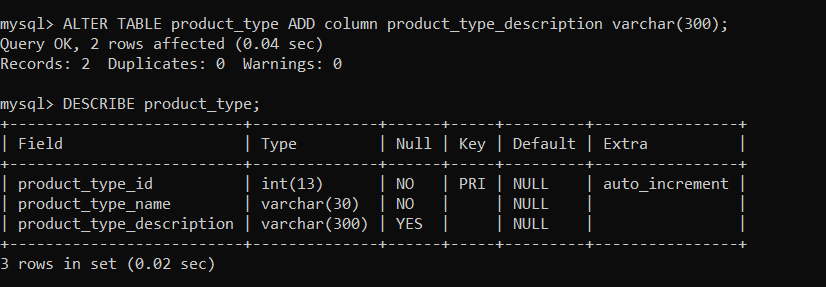
**Рисунок 49 - Распечатать информацию о клиенте с buyer\_name – не как F**

3.1.7 Cнимок экрана (скриншоты) сортировки данных



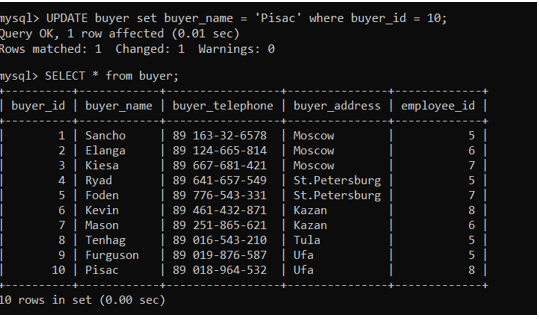
**Рисунок 50 - Отсортируйте данные в таблице «products» по имени и** в обратном порядке**.**

3.1.8 Снимки экрана (скриншоты) применения операторов изменения данных



**Рисунок 51 - Добавьте столбец product\_type\_description в таблицу product\_type.**

Оператор: Оператор UPDATE



**Рисунок 52 - Обновите данные столбца «admin\_address» для клиента с идентификаторами 1.**

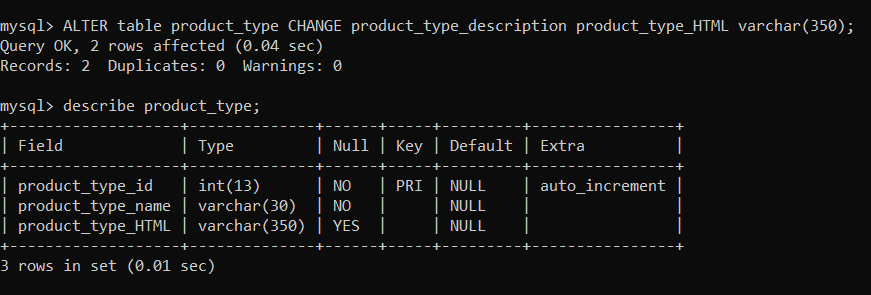
Оператор: CHANGE 

Рисунок 53 - Переименуйте столбец product\_type\_description в product\_type\_HTML.

Оператор: DELETE

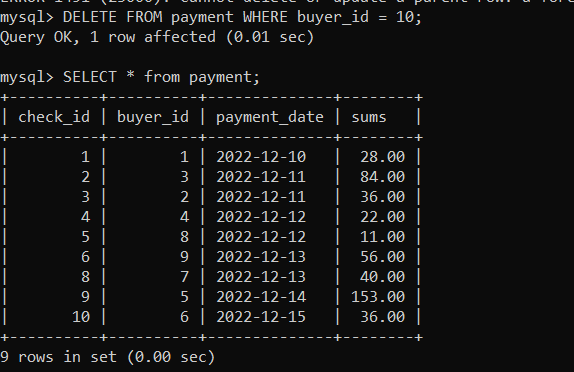


Рисунок 54 - Удалите payment с идентификаторой клиентом 10

Поддержка целостности БД

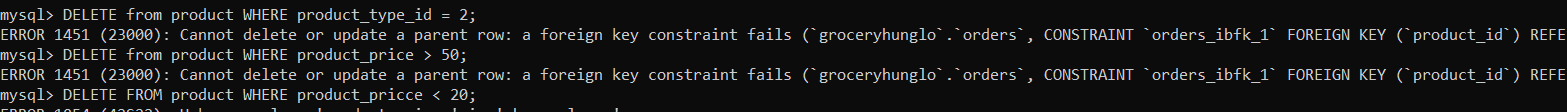


Рисунок 55 - Невозможно удалить информацию о product\_type\_id, которая содержит внешний ключ в другой таблице - целостность базы данных.

Итоговый скрипт БД

--

--SQL 1

--

creat database groceryHungLo;

use groceryHungLo;;

create table branches (

branches\_id int(13) AUTO\_INCREMENT,

branches\_telephone varchar(30) NOT NULL,

branches\_mail varchar(30) NOT NULL,

branches\_address varchar(50) NOT NULL,

PRIMARY KEY (branches\_id)

);

create table employee (

employee\_id int (13) AUTO\_INCREMENT,

employee\_name varchar (30) NOT NULL,

employee\_position varchar (30) NOT NULL,

branches\_id int (13) NOT NULL,

PRIMARY KEY (employee\_id),

FOREIGN KEY (branches\_id) REFERENCES branches (branches\_id)

);

create table product\_type (

product\_type\_id int (13) AUTO\_INCREMENT,

product\_type\_name varchar (30) NOT NULL,

PRIMARY KEY (product\_type\_id)

);

create table product (

product\_id int (13) AUTO\_INCREMENT,

product\_name varchar (30) NOT NULL,

product\_mass int (13) NOT NULL,

product\_price int (13) NOT NULL,

product\_description varchar (100) NOT NULL,

quanlity\_in\_stock int (13) NOT NULL,

product\_type\_id int (13) NOT NULL,

PRIMARY KEY (product\_id),

FOREIGN KEY (product\_type\_id) REFERENCES product\_type (product\_type\_id)

);

create table buyer (

buyer\_id int (13) AUTO\_INCREMENT,

buyer\_name varchar (30) NOT NULL,

buyer\_telephone varchar (30) NOT NULL,

buyer\_address varchar (30) NOT NULL,

employee\_id int (13) NOT NULL,

PRIMARY KEY (buyer\_id),

FOREIGN KEY (employee\_id) REFERENCES employee (employee\_id)

);

product\_id int (13) NOT NULL,

quanlity\_ordered int (13) NOT NULL,

data\_issue date NOT NULL,

date\_of\_completion date NOT NULL,

status varchar (30) NOT NULL,

buyer\_id int (13) NOT NULL,

PRIMARY KEY (order\_id),

FOREIGN KEY (product\_id) REFERENCES product (product\_id),

FOREIGN KEY (buyer\_id) REFERENCES buyer (buyer\_id)

);

reate table orders (

order\_id int (13) AUTO\_INCREMENT,

product\_id int (13) NOT NULL,

quanlity\_ordered int (13) NOT NULL,

data\_issue date NOT NULL,

date\_of\_completion date NOT NULL,

status varchar (30) NOT NULL,

buyer\_id int (13) NOT NULL,

PRIMARY KEY (order\_id),

FOREIGN KEY (product\_id) REFERENCES product (product\_id),

FOREIGN KEY (buyer\_id) REFERENCES buyer (buyer\_id)

);

create table payment (

check\_id int NOT NULL AUTO\_INCREMENT,

buyer\_id int NOT NULL,

payment\_date date NOT NULL,

sums decimal (10,2) NOT NULL,

PRIMARY KEY (`check\_id`),

FOREIGN KEY (`buyer\_id`) REFERENCES buyer (`buyer\_id`)

);

show database;

show tables;

describe branches;

describe employee;

describe buyer;

describe product\_type;

describe product;

describe orders;

describe payment;

INSERT INTO branches (branches\_id, branches\_telephone, branches\_mail, branches\_address) VALUES

(NULL, '89 777-910-001', 'Petshopru01@gmail.com', 'Moscow, Ryazansky prospect, 16'),

(NULL, '89 651-567-529', 'Petshopru02@gmail.com', 'Moscow, Belorechenskaya street, 38,);

INSERT INTO employee (employee\_id, employee\_name, employee\_position, branches\_id) VALUES

(1, ' Hung', 'Manager', '1'),

(2, ' Leon', 'Manager', '2'),

(3, 'Takina', 'VP Marketert', '2'),

(4, 'Fughashi', 'VP Sales', '1'),

(5, 'Andrey', 'Sales Rep', '1'),

(6, 'Anna', 'Sales Rep', '2'),

(7, 'Viktoria', 'Sales Rep', '1'),

(8, 'Sofia', 'Sales Rep', '2');

INSERT INTO buyer (buyer\_id, buyer\_name, buyer\_telephone, buyer\_address, employee\_id) VALUES

(1, 'Sancho', '89 163-32-6578', 'Moscow', '5'),

(2,'Elanga', '89 124-665-814', 'Moscow', '6'),

(3, 'Kiesa', '89 667-681-421', 'Moscow', '7'),

(4, 'Ryad', '89 641-657-549', 'St.Petersburg', '5'),

(5, 'Foden', '89 776-543-331', 'St.Petersburg', '7'),

(6, 'Kevin', '89 461-432-871', 'Kazan', '8'),

(7, 'Mason', '89 251-865-621', 'Kazan', '6'),

(8, 'Tenhag', '89 016-543-210', 'Tula', '5'),

(9, 'Furguson', '89 019-876-587', 'Ufa', '5'),

(10, 'Beckham', '89 018-964-532', 'Ufa', '8');

INSERT INTO product\_type (product\_type\_id, product\_type\_name) VALUES

(NULL, 'DRY PET FOODS'),

(NULL, 'WET PET FOODS');

INSERT INTO product (product\_id, product\_name, product\_mass, product\_price, product\_description, quanlity\_in\_stock, product\_type\_id) VALUES

(1, 'Pedigree', 4, 28, 'Complete nutrition grilled steak and vegetable flavor dog kibble adult dry dog food', 20, '1'),

(2, 'Purina Pro Plan', 5, 40, 'Adult sensitive skin stomach salmon rice formula dry dog food', 18, '1'),

(3, 'Purina ONE', 5, 42, 'Natural smartblend chicken rice formula dry dog food', 15, '1'),

(4, 'Hill Science Diet', 4, 51, 'Adult perfect weight chicken recipe dry dog food', 15, '1'),

(5, 'Kibble Bits', 3, 12, 'Original savory beef & chicken flavors dry dog food', 11, '1'),

(6, 'Cesar', 2, 22, 'Classic loaf in sauce beef recipe, filet mignon, grilled chicken, pack dog food trays', 31, '2'),

(7, 'Blue Buffalo', 2, 25, 'Homestyle recipes adult variety pack chicken & beef dinner canned dog food', 28, '2'),

(8, 'Hill', 3, 51, 'Hill Prescription Diet', 14, '2'),

(9, 'Rachael Ray Nutrish', 2 , 18, 'Natural variety pack wet dog food', 10, '2'),

(10, 'Lams', 3, 22, 'Proactive health classic ground with chicken & whole grain rice adult wet dog food', 11, '2');

INSERT INTO orders (order\_id, product\_id, quanlity\_ordered, data\_issue, date\_of\_completion, status, buyer\_id) VALUES

(1, 1, 1, '2022-12-10', '2022-12-11','Performance', '1'),

(2, 3, 2, '2022-12-11', '2022-12-11','Performance', '2'),

(3, 5, 3, '2022-12-11', '2022-12-11','Performance', '3'),

(4, 6, 1, '2022-12-12', '2022-12-13', 'Performance', '4'),

(5, 6, 1, '2022-12-12', '2022-12-13', 'Performance', '8'),

(6, 1, 2, '2022-12-13', '2022-12-13','Performance', '9'),

(7, 3, 3, '2022-12-13', '2022-12-14','Performance', '10'),

(8, 2, 1, '2022-12-13', '2022-12-14','Performance', '7'),

(9, 8, 3, '2022-12-14', '2022-12-15','Performance', '5'),

(10, 9, 2, '2022-12-15', '2022-12-16','Performance', '6');

INSERT INTO payment (check\_id, buyer\_id, payment\_date, sums) VALUES

(1, 1, '2022-12-10', 28),

(2, 3, '2022-12-11', 84),

(3, 2, '2022-12-11', 36),

(4, 4, '2022-12-12', 22),

(5, 8, '2022-12-12', 11),

(6, 9, '2022-12-13', 56),

(7, 10, '2022-12-13', 126),

(8, 7, '2022-12-13', 40),

(9, 5, '2022-12-14', 153),

(10, 6, '2022-12-15', 36);

SELECT \* from branches;

SELECT \* from employee;

SELECT \* from buyer;

SELECT \* from product\_type;

SELECT \* from product;

SELECT \* from orders;

SELECT \* from payment;

--

-- SQL2

--

SELECT \* from branches;

SELECT branches\_telephone, branches\_mail from branches;

SELECT \* from employee;

SELECT \* from employee WHERE employee\_id = 5;

SELECT \* from buyer;

SELECT \* from buyer WHERE buyer\_id > 6;

SELECT \* from buyer WHERE buyer\_id < 5;

SELECT \* from product;

SELECT \* from product WHERE product\_price >=20;

SELECT \* from product WHERE product\_price <=20;

SELECT \* from employee;

SELECT \* from employee WHERE branches\_id !=2;

SELECT \* from product

SELECT \* from product WHERE product\_mass IS NOT NULL;

SELECT \* from product WHERE quanlity\_in\_stock IS NULL;

SELECT \* from buyer

SELECT \* from buyer WHERE buyer\_id IN(1,10);

SELECT \* from buyer WHERE buyer\_id NOT IN (1,10);

SELECT \* from buyer WHERE buyer\_name like 'F%';

SELECT \* from buyer WHERE buyer\_name not like 'F%';

SELECT \* from product;

SELECT \* from product order by product\_price;

SELECT \* from product order by product\_price desc;

DESCRIBE product\_type;

ALTER TABLE product\_type ADD column product\_type\_description varchar(300);

DESCRIBE product\_type;

describe buyer;

UPDATE buyer set buyer\_name = 'Pisac' where buyer\_id = 10;

describe buyer;

describe product\_type;

ALTER table product\_type CHANGE product\_type\_description product\_type\_HTML varchar(350);

describe product\_type;

SELECT \* from payment;

DELETE FROM payment WHERE buyer\_id = 10;

SELECT \* from payment;

3.2. Работа в MySQL Workbench

Физическая модель разрабатываемой базы данных представлена на Рисунке 56.

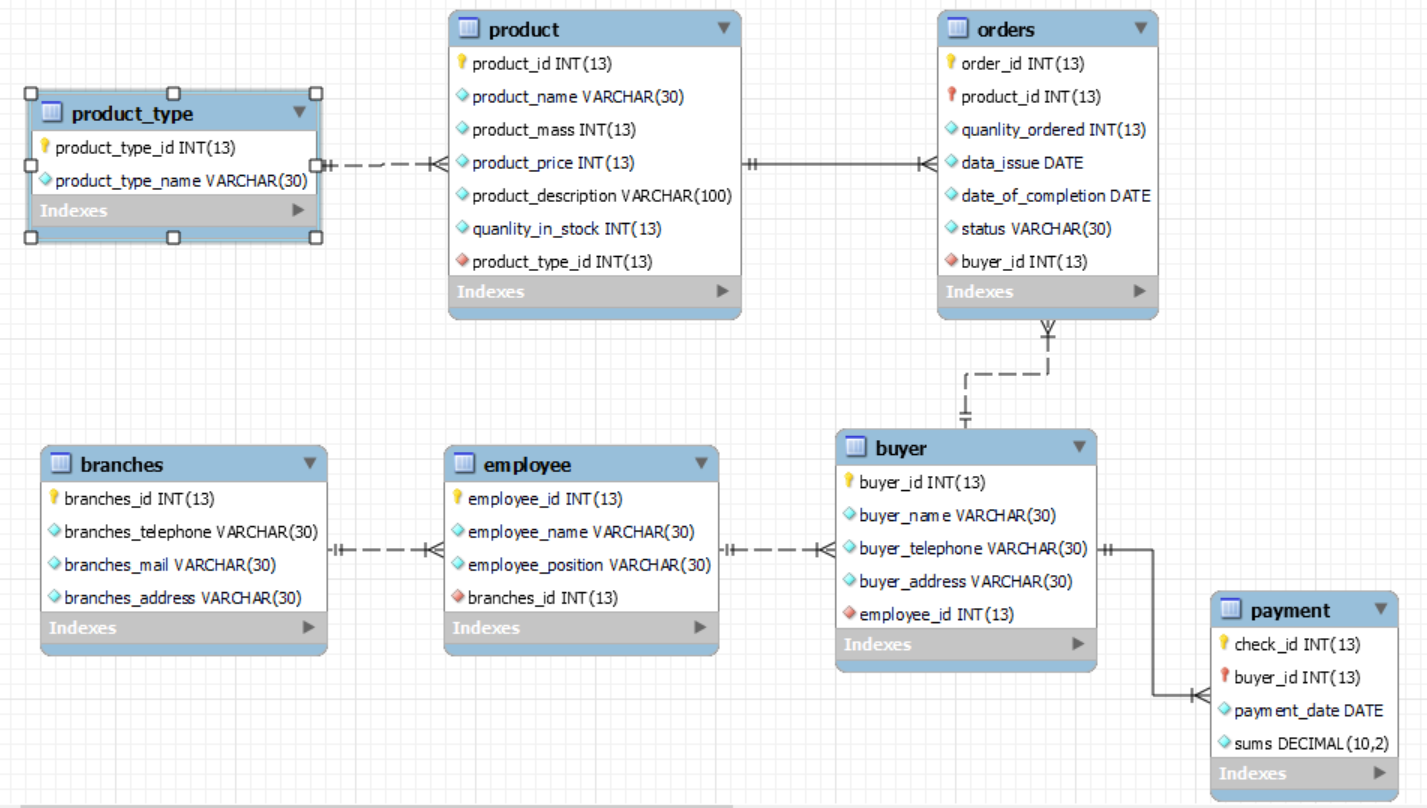
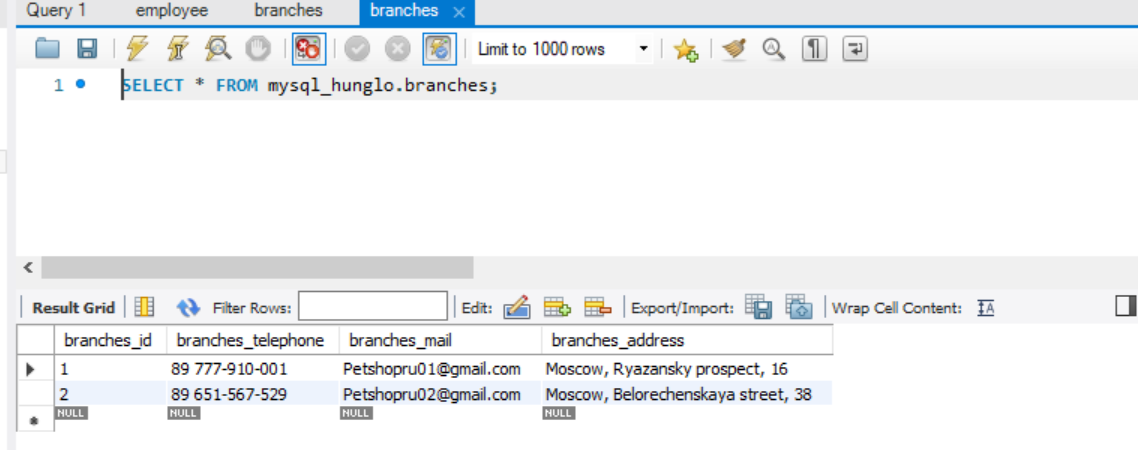
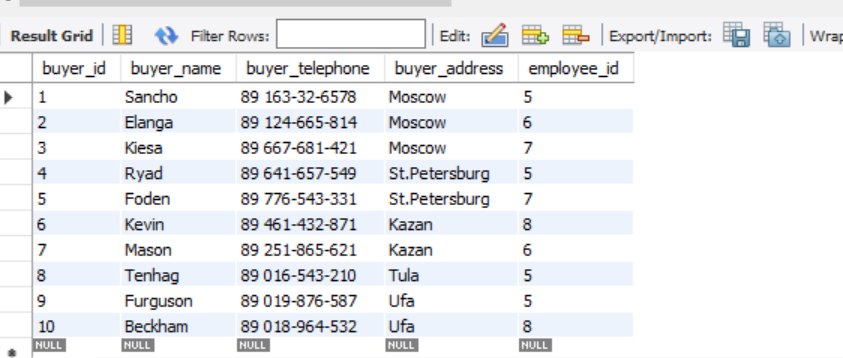


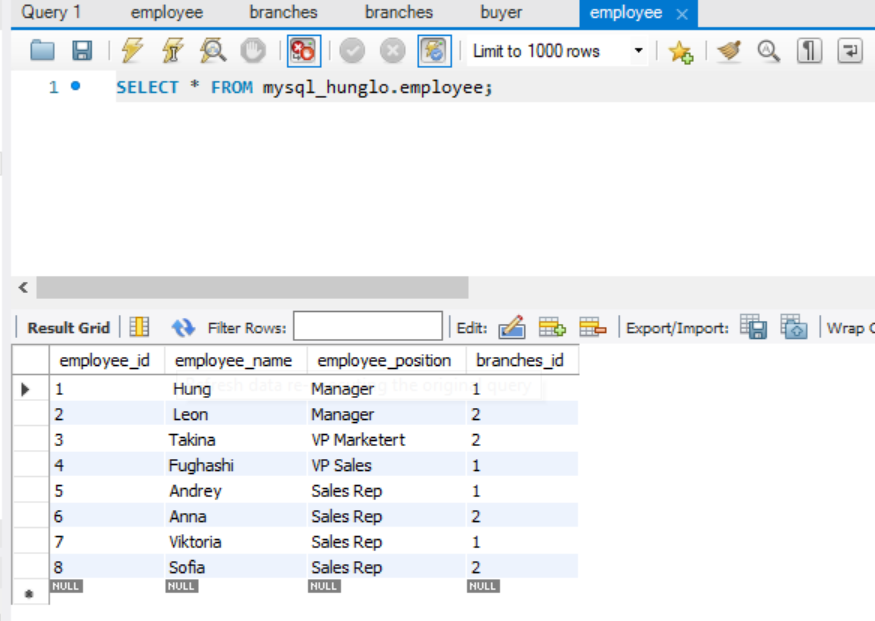
Рисунок 56 - Физическая модель базы данных в MySQL Workbench.



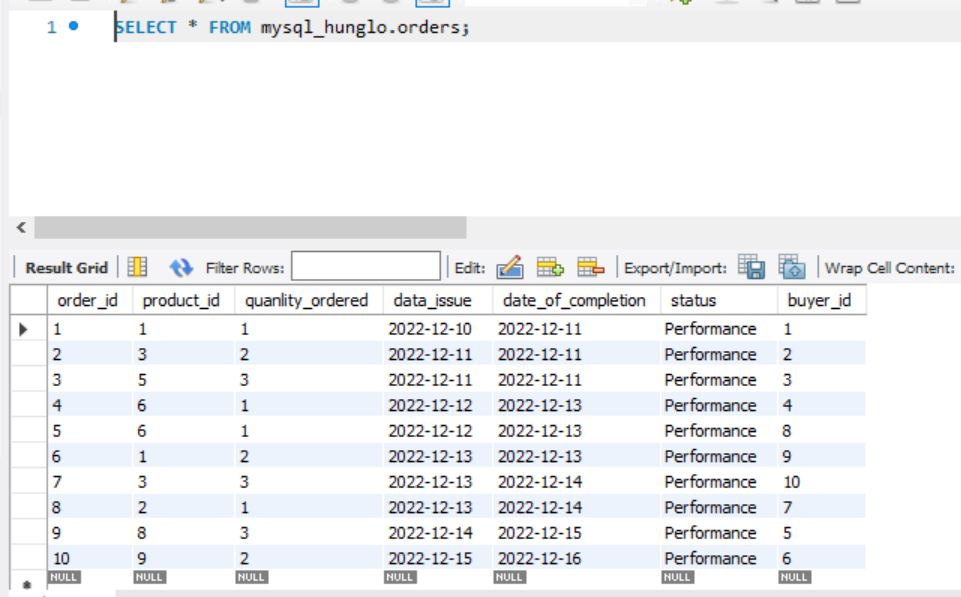
**Рисунок 57 - Таблица “branches”**



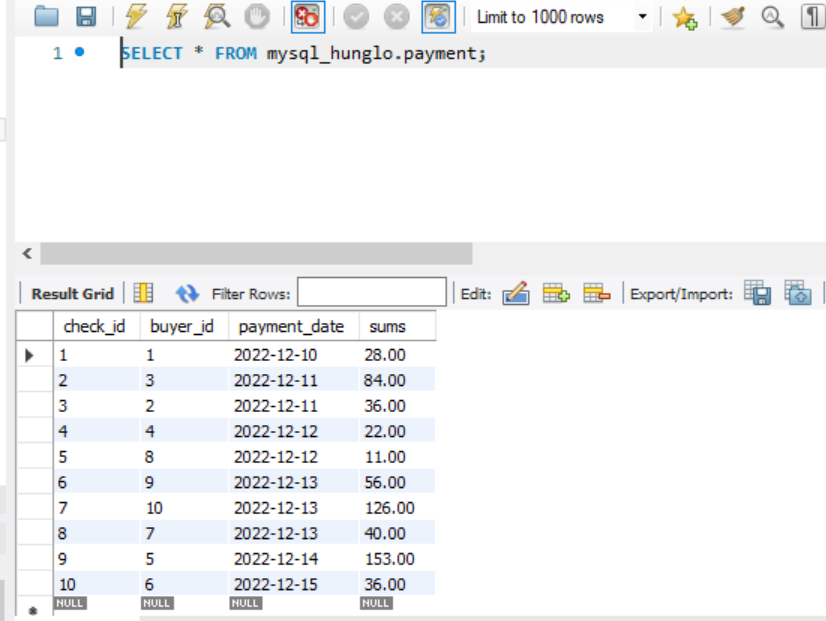
**Рисунок 58 - Таблица “buyer”**



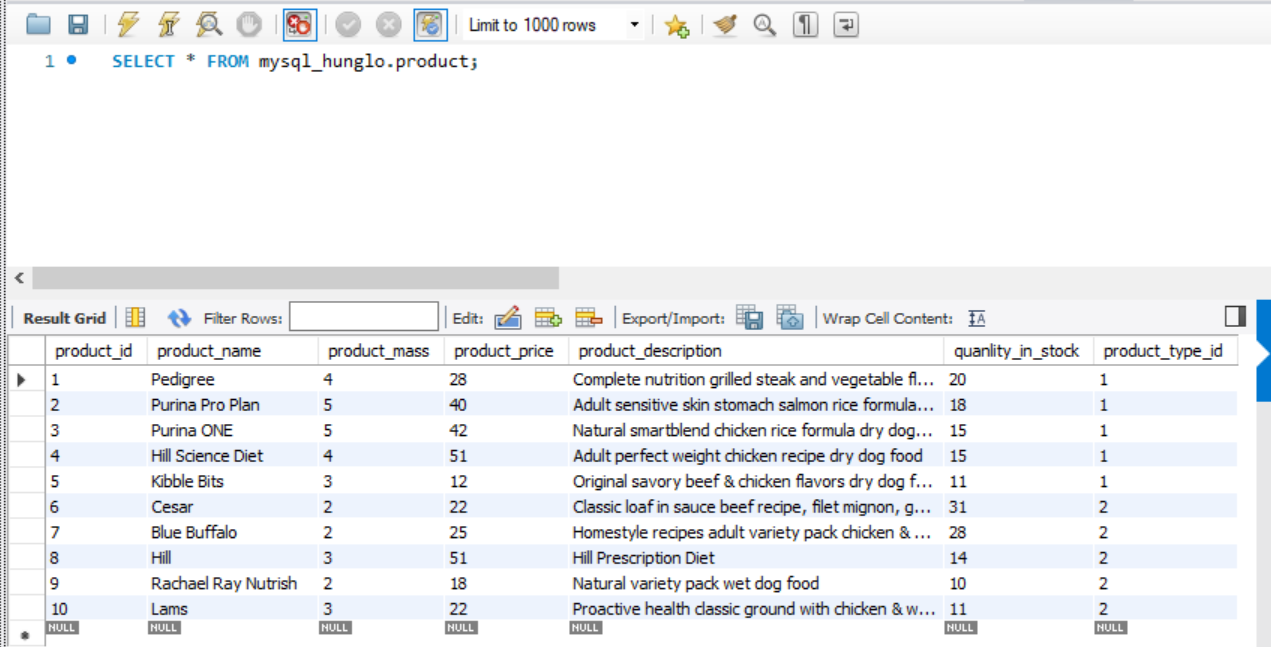
**Рисунок 59 - Таблица “employee”**



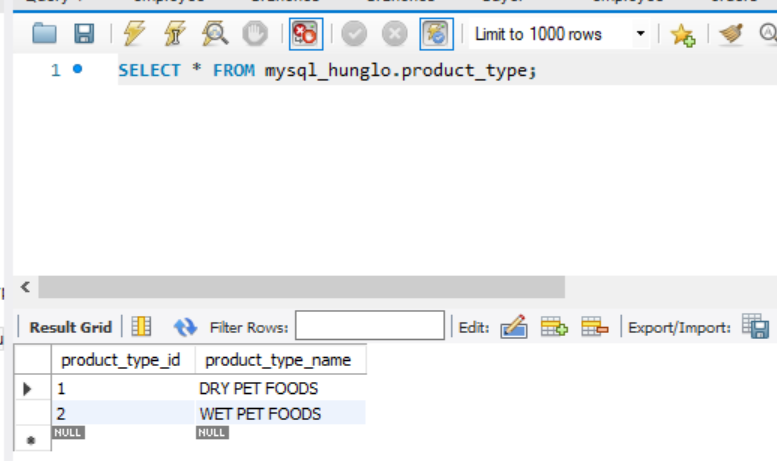
**Рисунок 60 - Таблица “orders”**



**Рисунок 61 - Таблица “payment”**



**Рисунок 62 - Таблица “product”**



**Рисунок 63 - Таблица “payment”**

Запросы к построенной БД

Запрос 1. Операция проекции. Осуществляется выбор только части полей таблицы, т.е. производится вертикальная выборка данных.

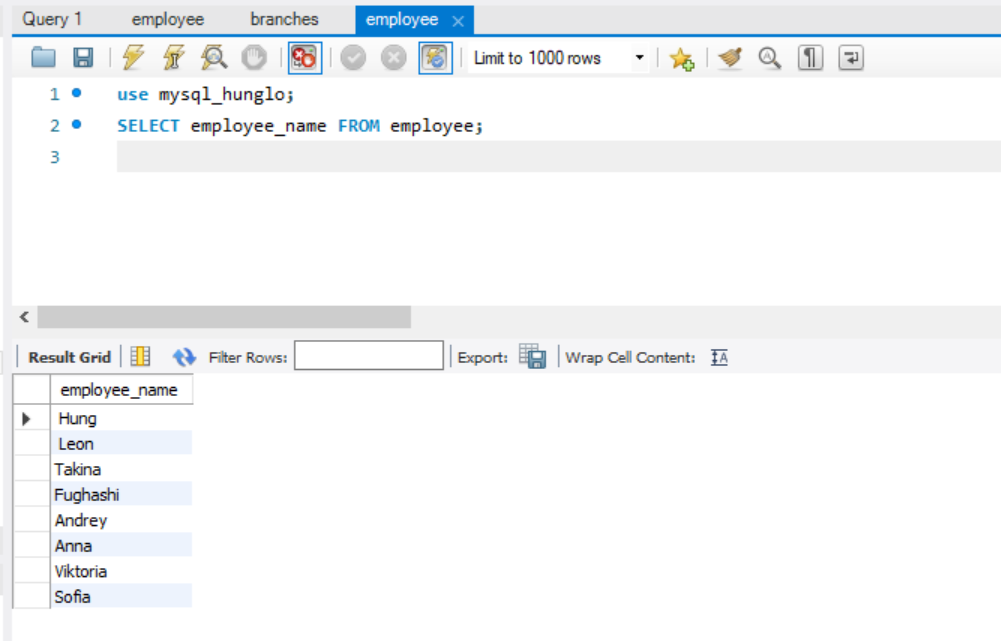


Рисунок 64 - **Результат выполнения запроса**

Запрос 2. Операция селекции. Осуществляется горизонтальная выборка – в результат попадают только записи, удовлетворяющие условию.

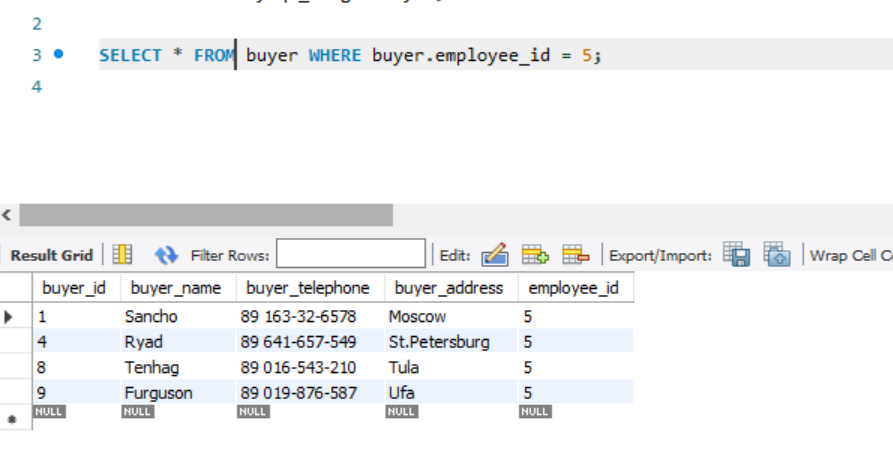


Рисунок 65 - **Результат выполнения запроса**

**Запрос 3. Операции соединения**. Здесь следует выделить декартово произведение и на его основе соединение по условию, а также естественное соединение (по одноименным полям или равенству полей с одинаковым смыслом).

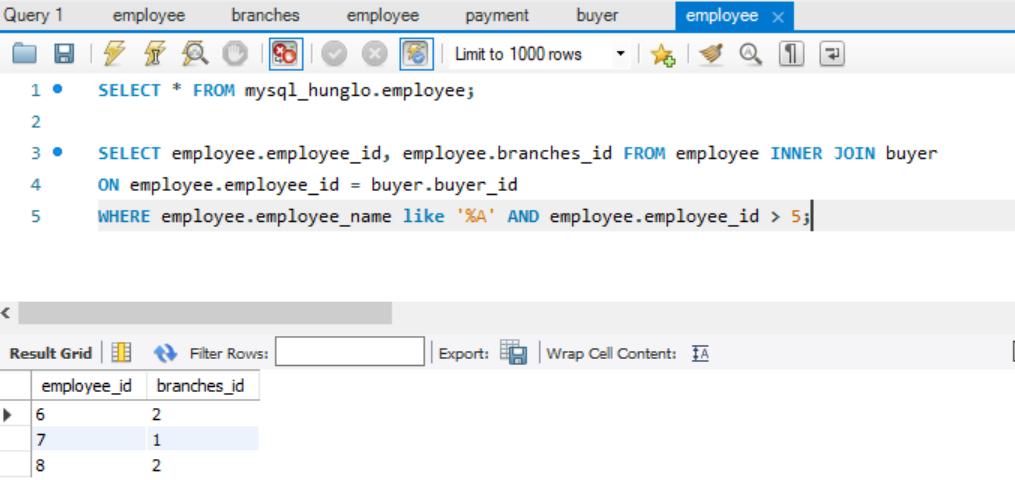


Рисунок 66 - **Результат выполнения запроса**

Запрос 4. Операция объединения. Теоретико-множественные операции часто можно записать с помощью логических операций, примененных в конструкции WHERE запроса.

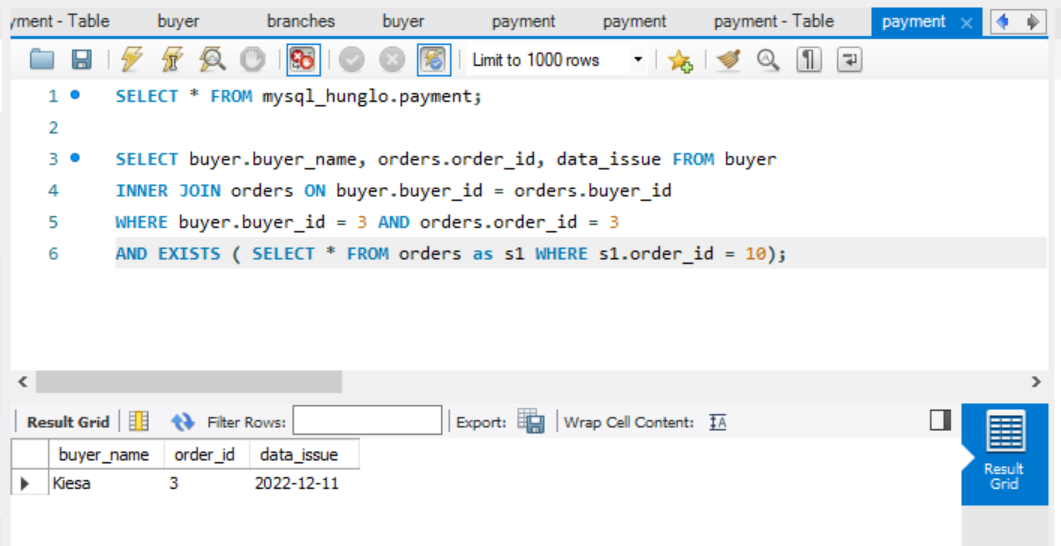


Рисунок 67 - **Результат выполнения запроса**

**Запрос 5. Операция пересечения**. В простых случаях эту операцию  
можно описать с помощью логической операции AND. В более сложных случаях эта операция определяется чаще всего с помощью подзапроса и ключевого слова EXISTS, которое показывает наличие похожего элемента во множестве,которое задается подзапросом.

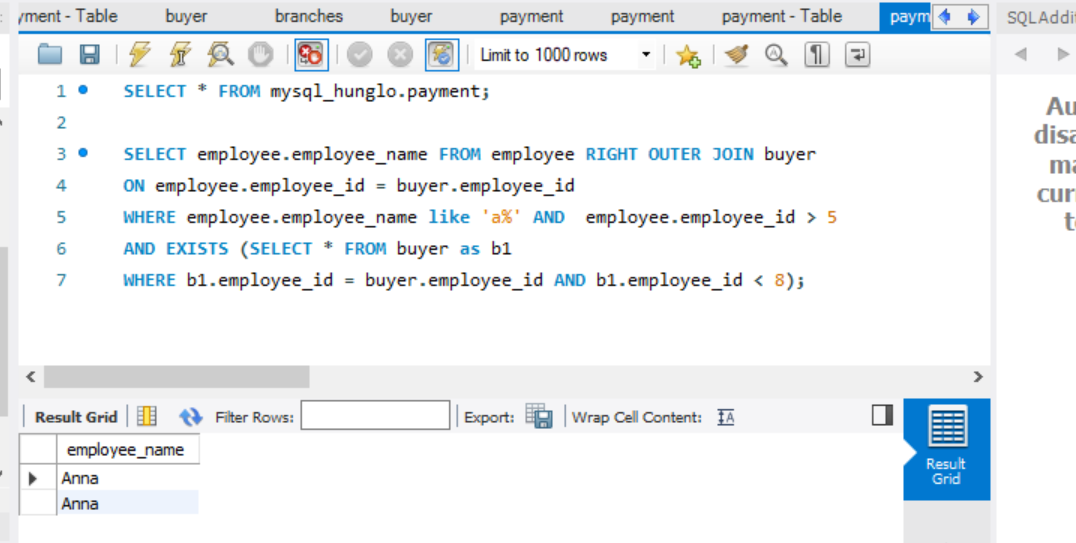


Рисунок 68 - **Результат выполнения запроса**

Запрос 6. Операция разности. Эта операция также определяется часто с помощью подзапроса с ключевым словом NOT EXISTS, которое показывает отсутствие элемента во множестве, задаваемом подзапросом. Приведем аналогичный предыдущему пример.

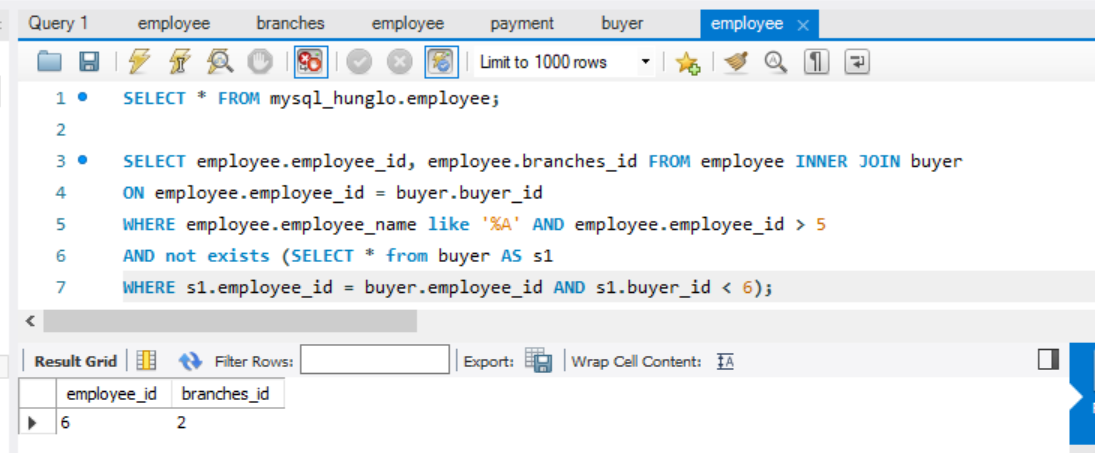


Рисунок 69 - **Результат выполнения запроса**

Запрос 7. Операция группировки. Эта операция связана со своеобразной сверткой таблицы по полям группировки. Помимо полей группировки результат запроса может содержать итоговые агрегирующие функции по группам (COUNT, SUM, AVG, MAX, MIN).

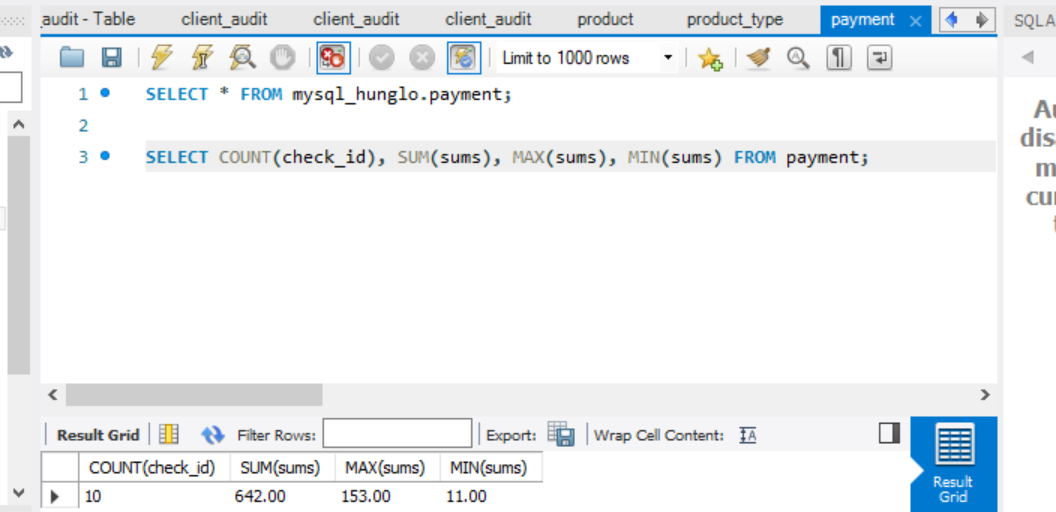


Рисунок 70 - **Результат выполнения запроса**

Запрос 8. Операция сортировки.

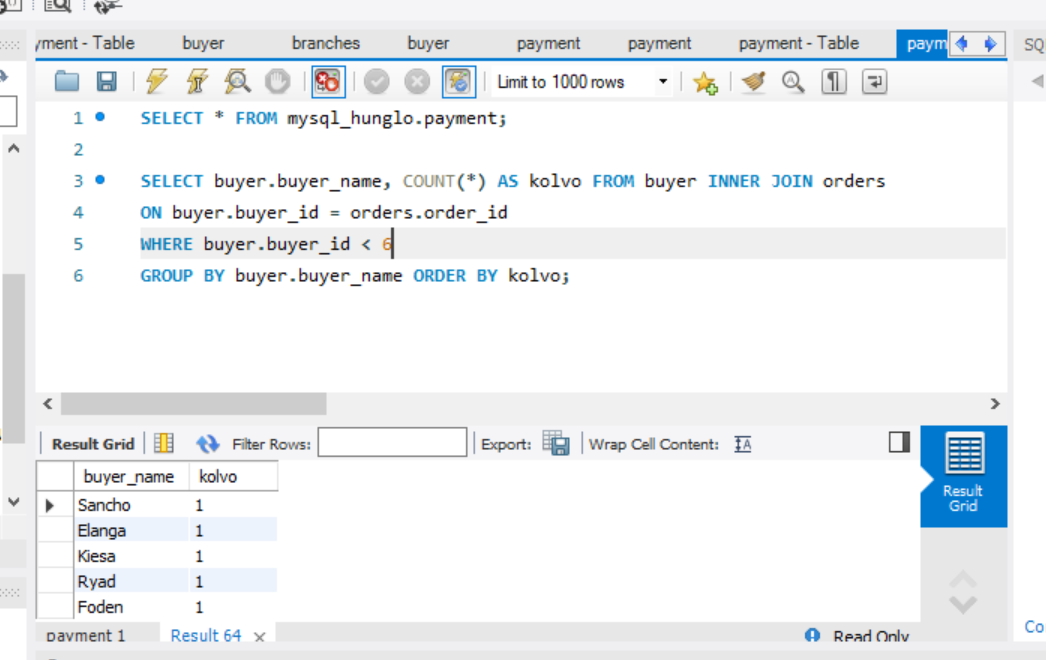


Рисунок 71 - **Результат выполнения запроса**

Запрос 9. Операция деления. Это самая нетривиальная операция реляционной алгебры, которая обычно применяется тогда, когда требуется найти все записи первой таблицы, которые соединяются естественным образом со всеми записями второй таблицы.

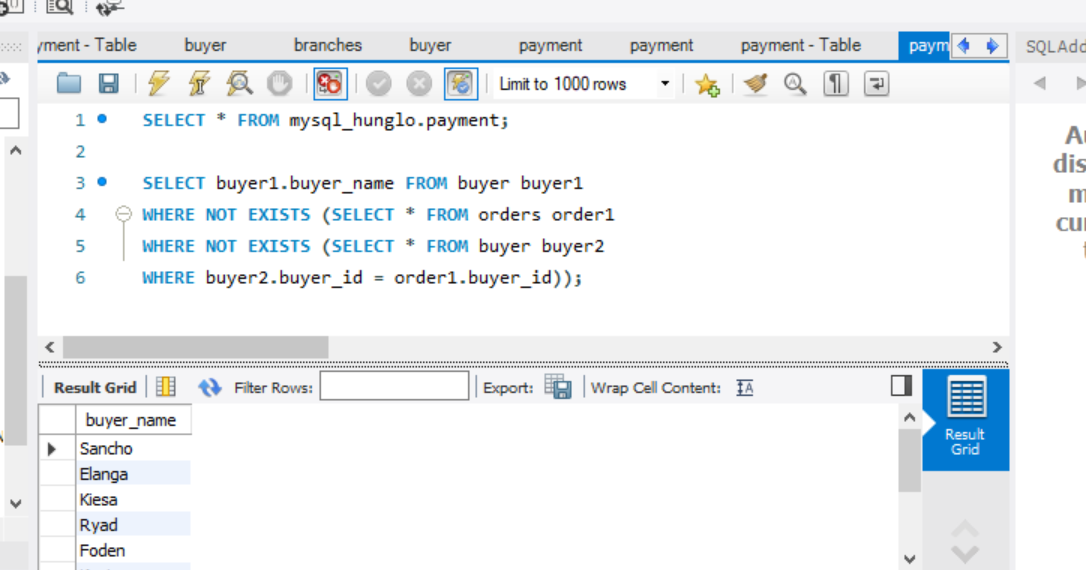


Рисунок 71 - **Результат выполнения запроса**

**Хранимые процедуры, функции и триггеры**

Хранимые процедуры CREATE PROCEDURE: Подсчитайте каждый тип продукта

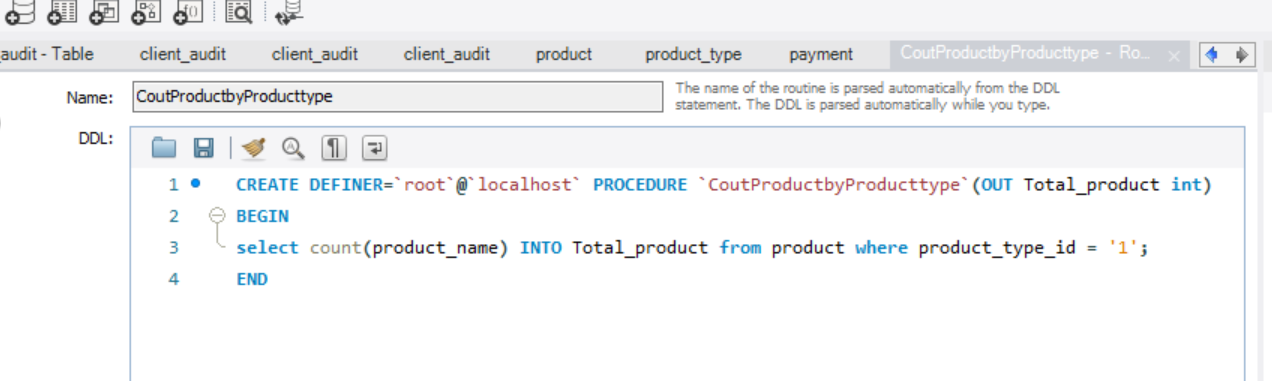
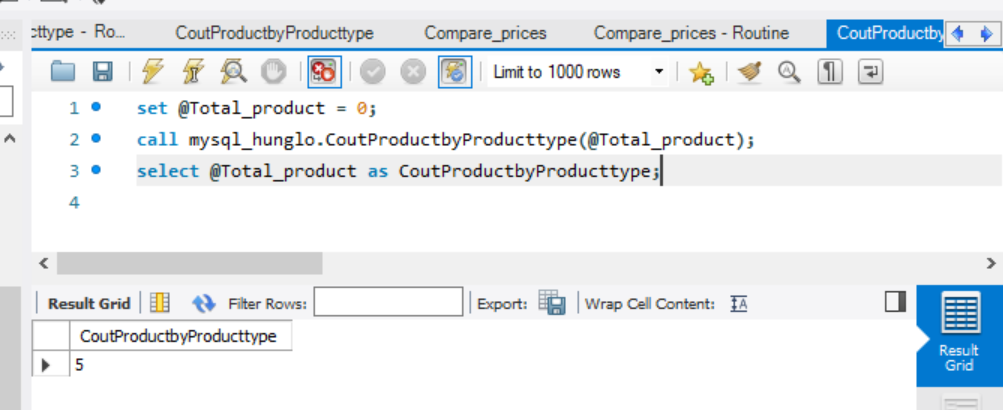
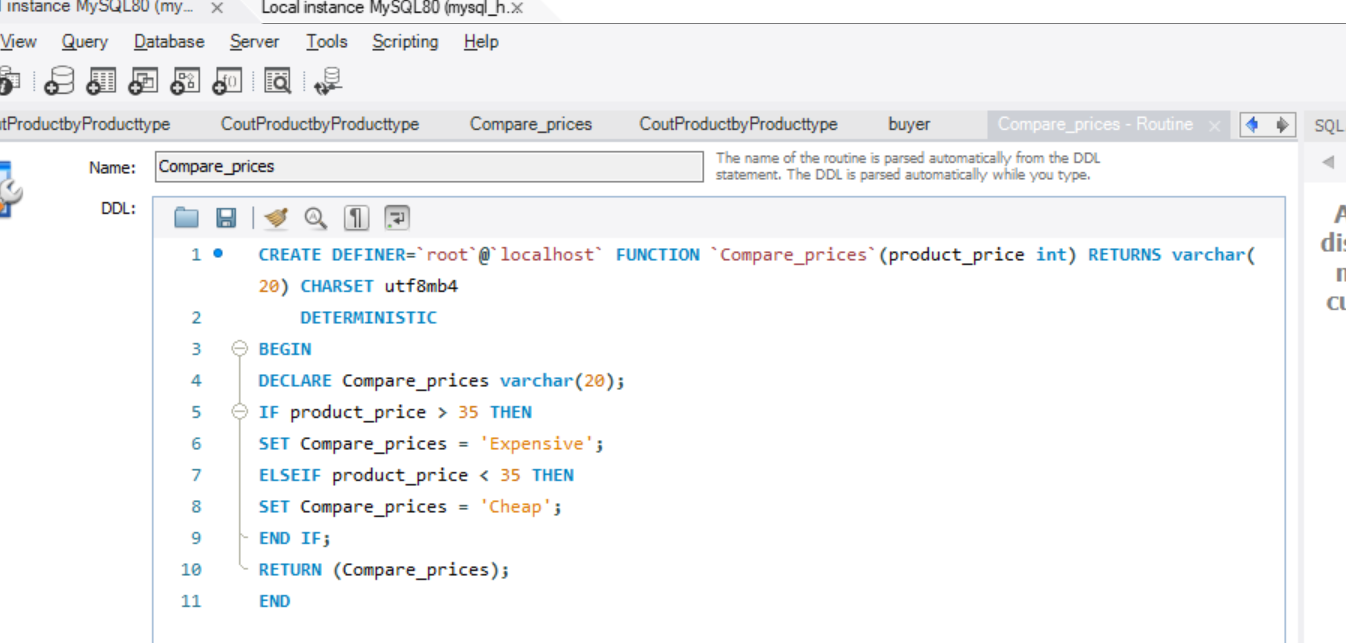


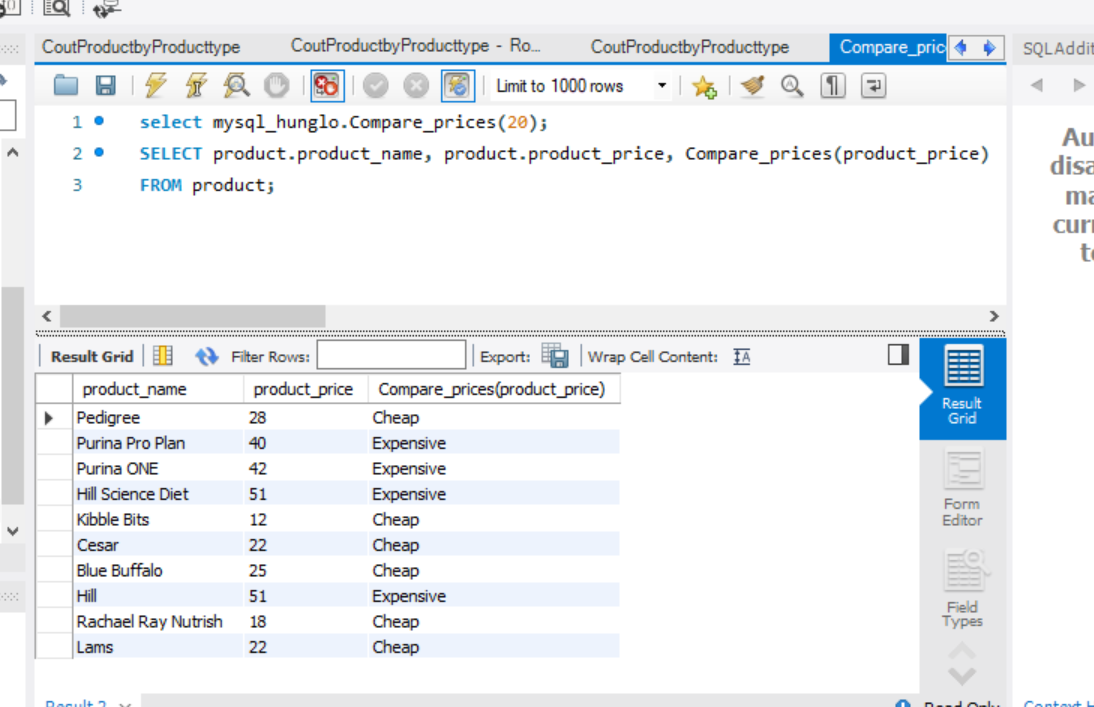
Рисунок 71 - CREATE PROCEDURE 

**Рисунок 72 - Показаны результаты**

Создайте функцию, используя CREATE FUNCTION: сравнить цены на товары.

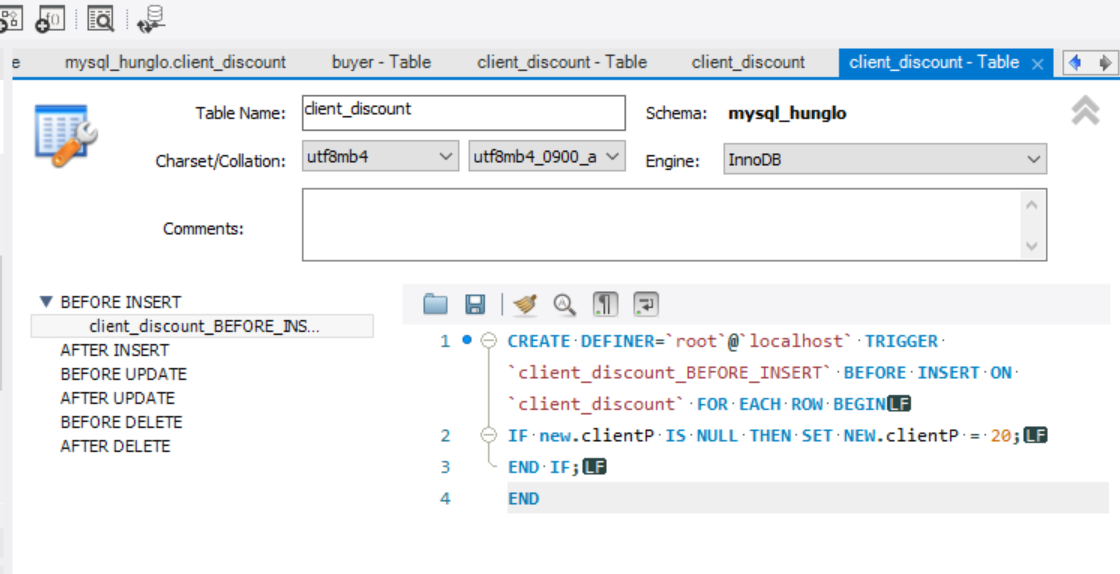


**Рисунок 73 -** CREATE FUNCTION

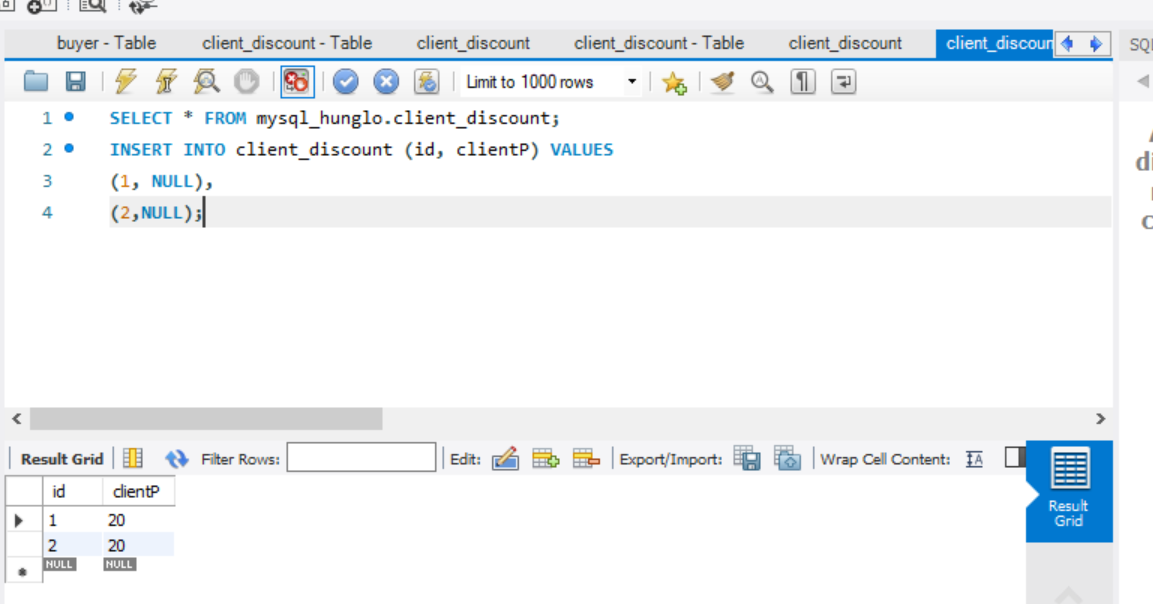


**Рисунок 74 - Показаны результаты**

Создайте триггер TRIGGER: Возвращает значение NULL к фактическому значению



**Рисунок 75 -** CREATE **TRIGGER**



**Рисунок 76 - Показаны результаты**

ВЫВОДЫ

В ходе работы мы построили логическую и физическую модели предметной области “Деятелности магазин кормов для животных” и создали к ней базу данных, а также реализовали её с использованием SQL и создали приложение для работы с ней.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Практика\_SQL\_1-2-3 - Разработка баз данных\_1/1\_57 / Материалы по дисциплине /Материалы к практическим занятиям / <https://online-edu.mirea.ru/>
2. <https://viblo.asia/p/su-dung-trigger-trong-sql-qua-vi-du-co-ban-aWj538APK6m>
3. <https://www.sql.edu.vn/microsoft-sql-server/create-function/>
4. <https://www.w3schools.com/sql/default.asp>