1. Создание файлов, их объединение и переименование:

Создание файла "Домашние животные" и его заполнение

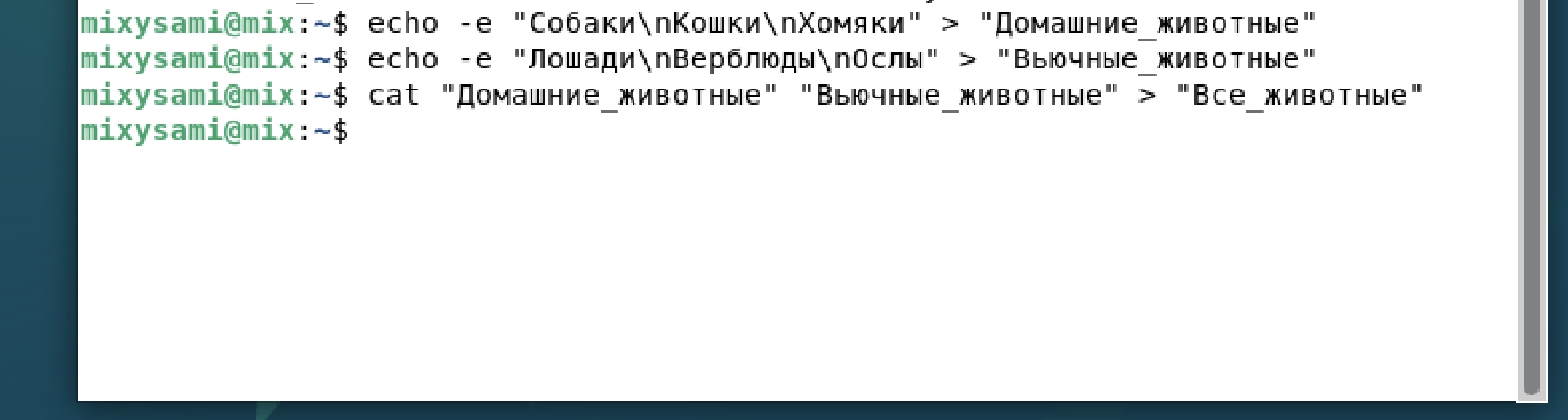
echo -e "Собаки\nКошки\nХомяки" > "Домашние животные"

Создание файла "Вьючные животные" и его заполнение

echo -e "Лошади\nВерблюды\nОслы" > "Вьючные животные"

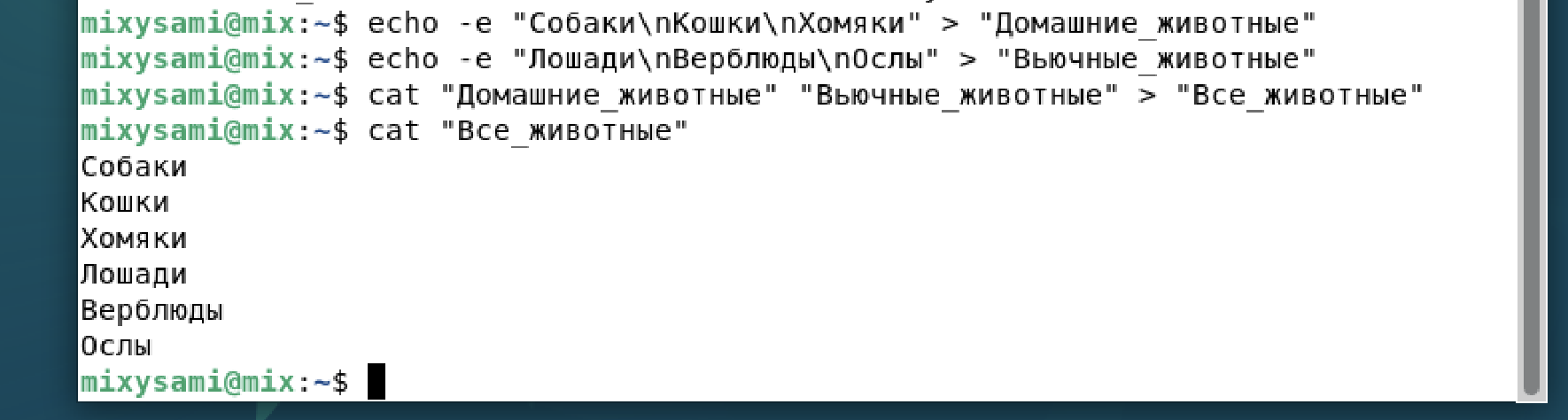
Объединение файлов в один

cat "Домашние животные" "Вьючные животные" > "Все животные"



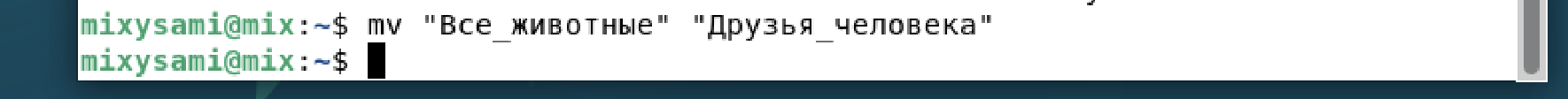
Просмотр содержимого объединенного файла

cat "Все животные"



Переименование файла

mv "Все животные" "Друзья человека"



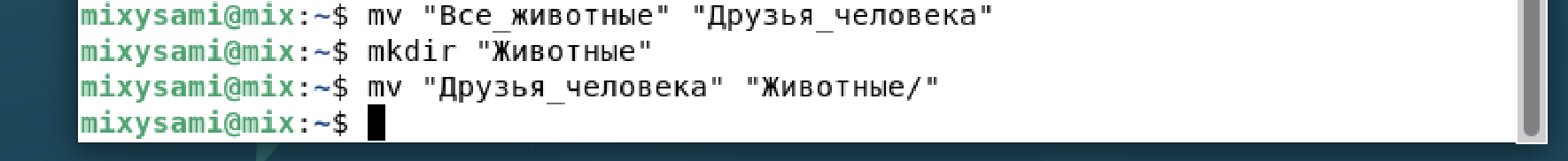
2. Создание директории и перемещение файла:

Создание директории

mkdir "Животные"

Перемещение файла в новую директорию

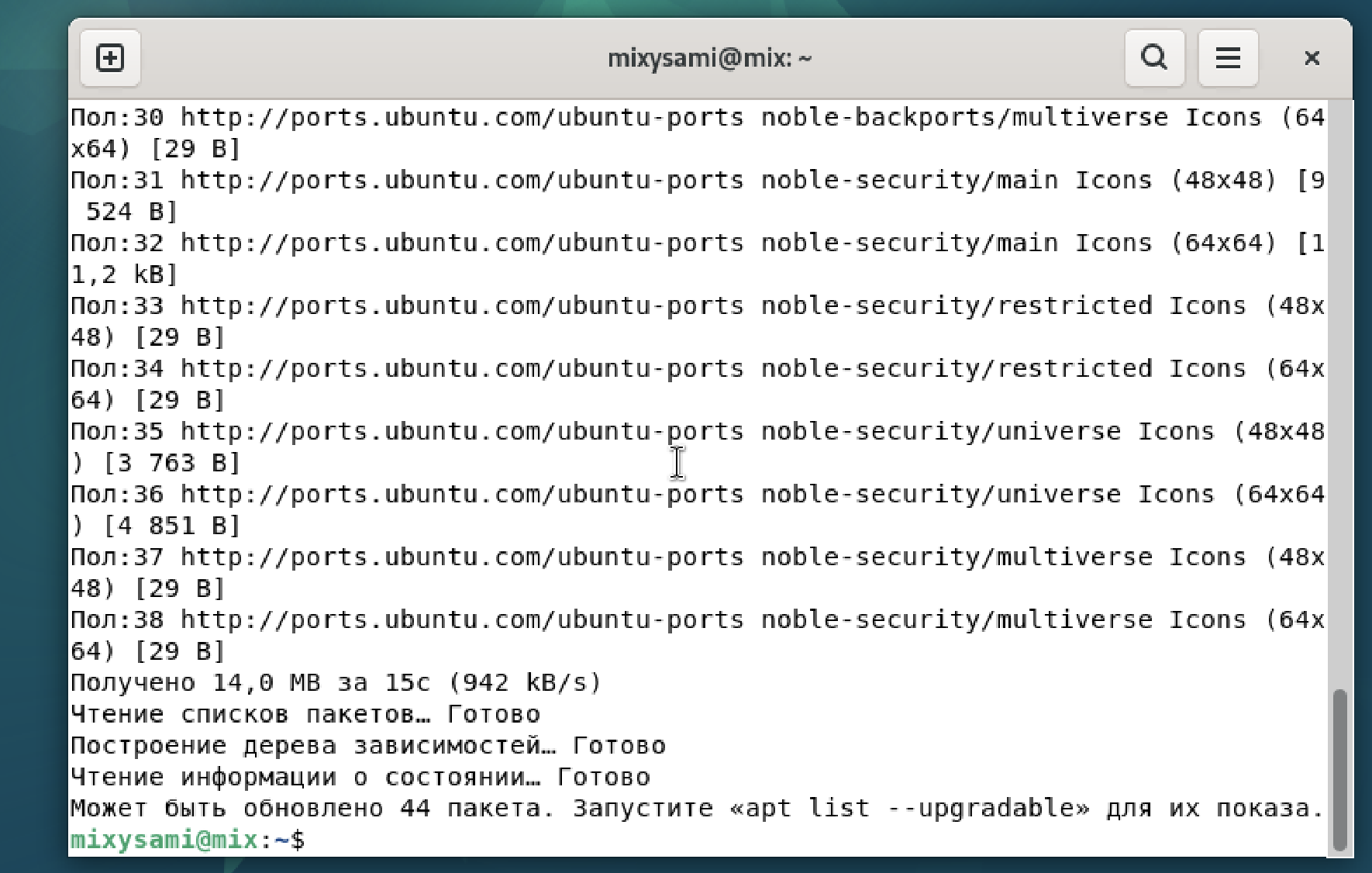
mv "Друзья человека" "Животные/"



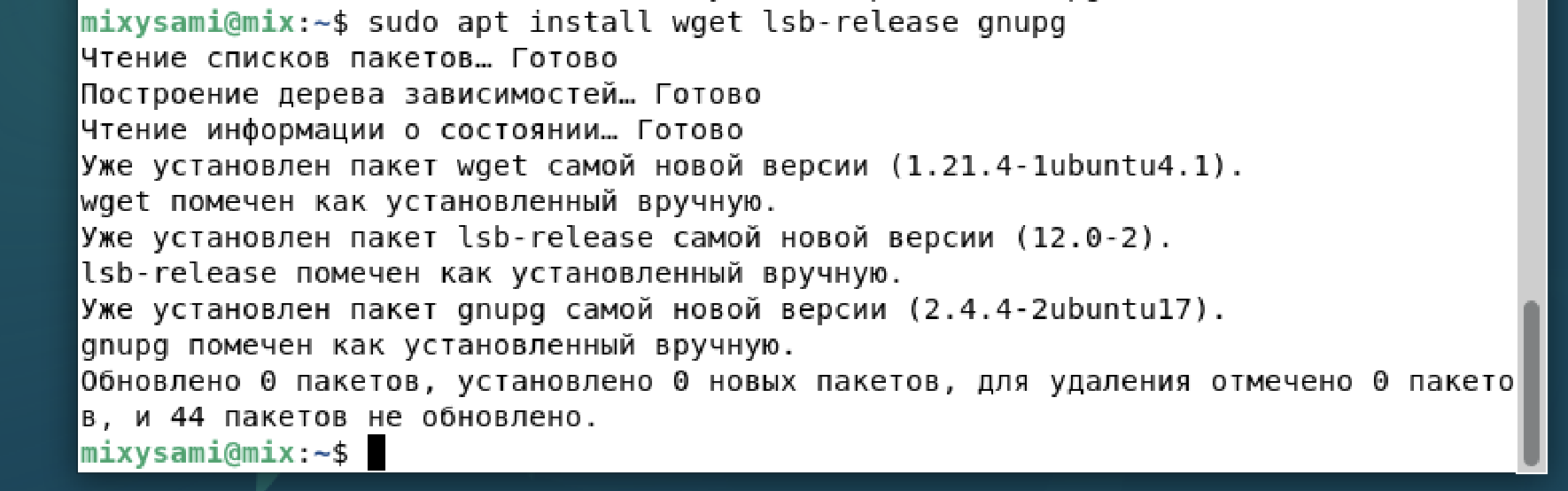
3. Подключение дополнительного репозитория MySQL и установка пакета:

Подключение репозитория MySQL

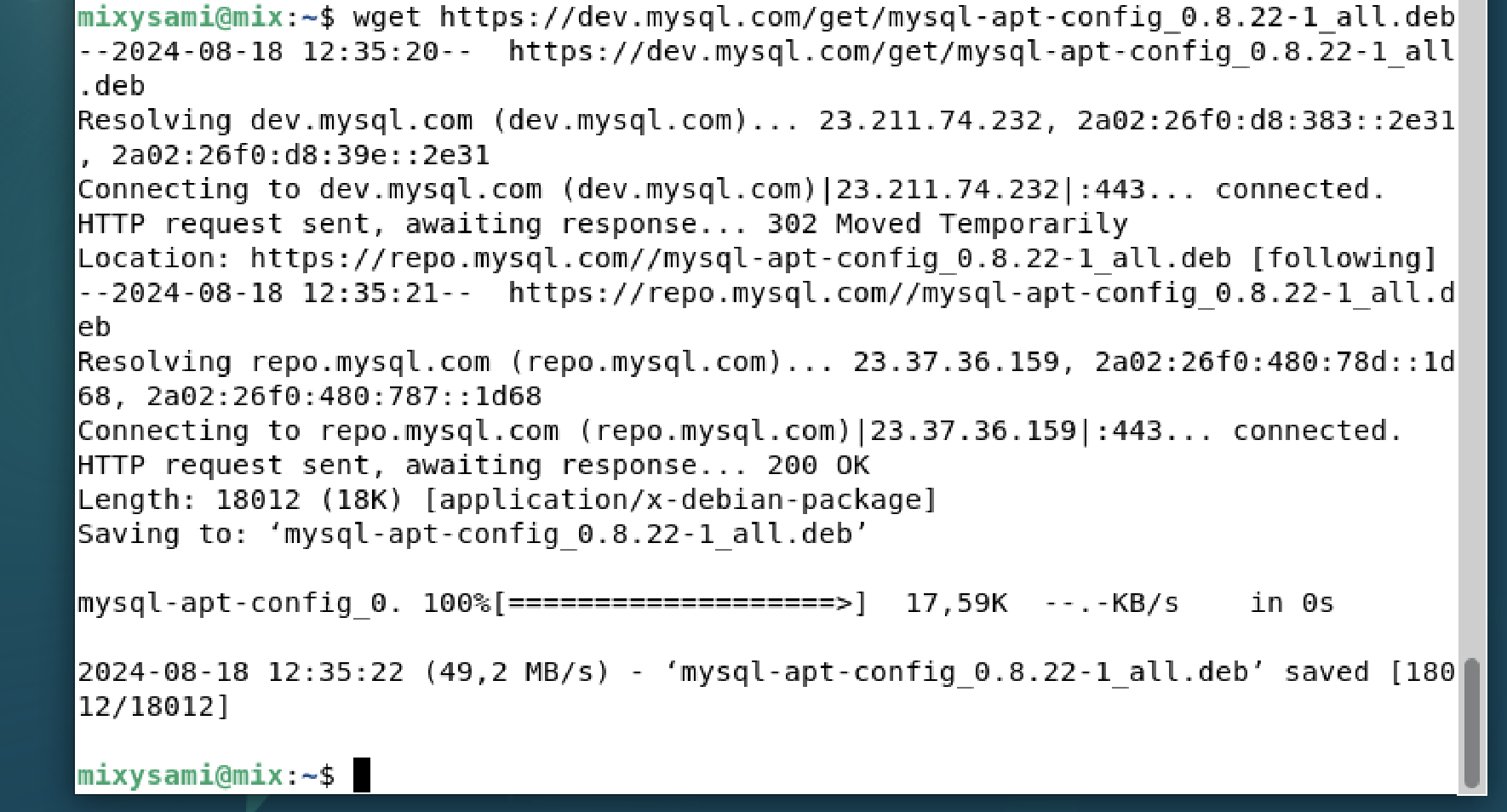
sudo apt update



sudo apt install wget lsb-release gnupg



wget https://dev.mysql.com/get/mysql-apt-config\_0.8.22-1\_all.deb

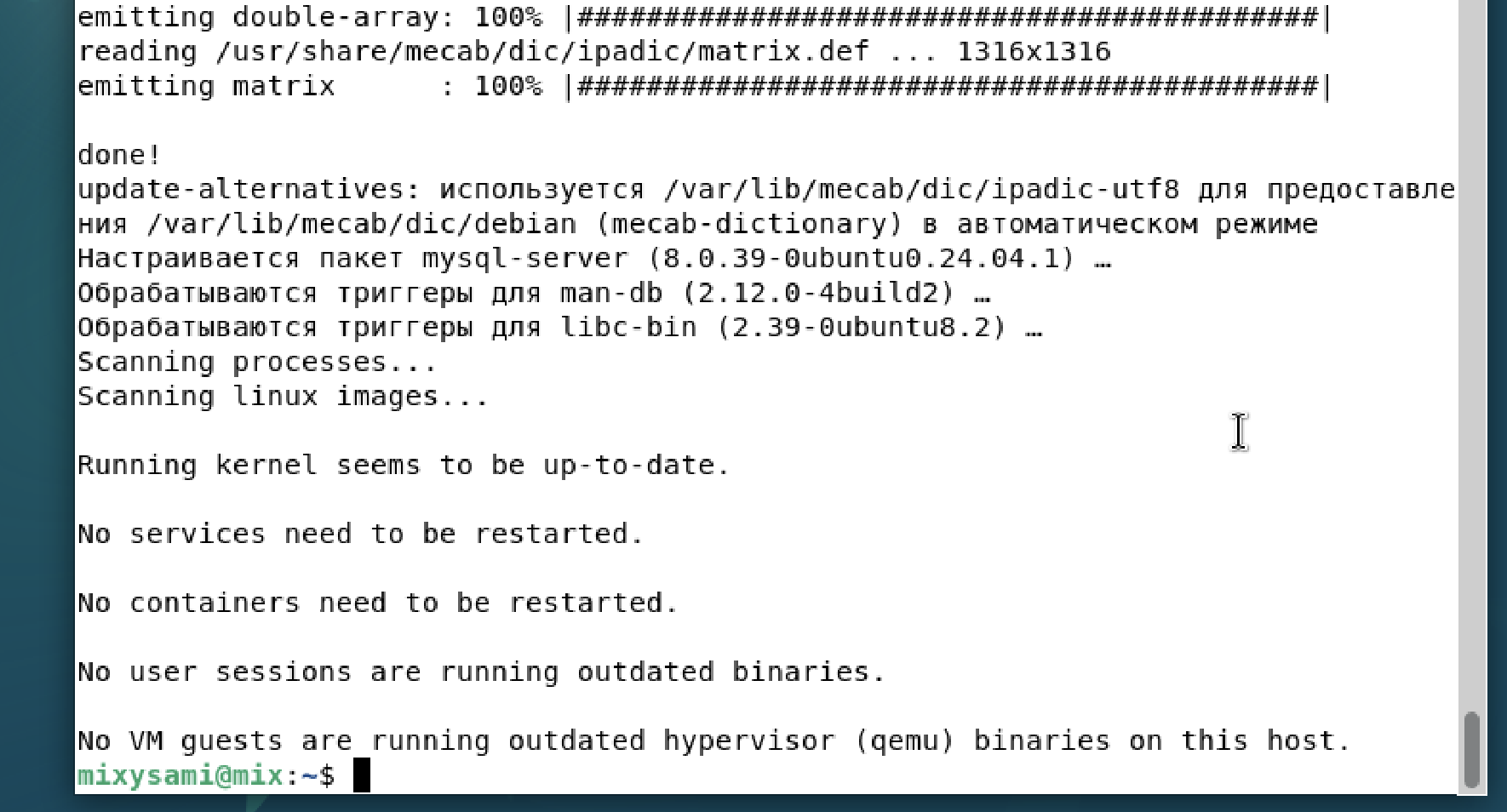


sudo dpkg -i mysql-apt-config\_0.8.22-1\_all.deb

Обновление списка пакетов и установка MySQL

sudo apt update

sudo apt install mysql-server



4. Установка и удаление deb-пакета с помощью `dpkg`:

Скачивание deb-пакета

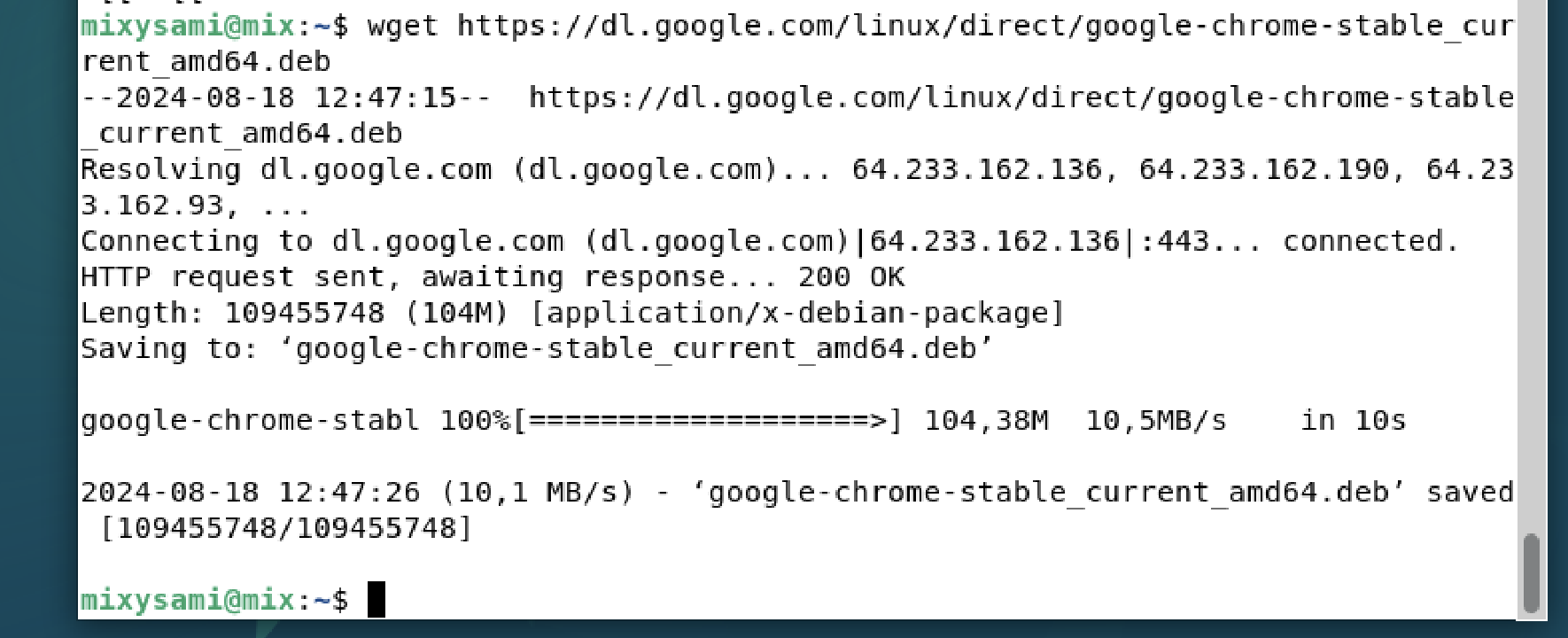
wget <https://example.com/package.deb>

Установка пакета

sudo dpkg -i package.deb

Удаление пакета

sudo dpkg -r package

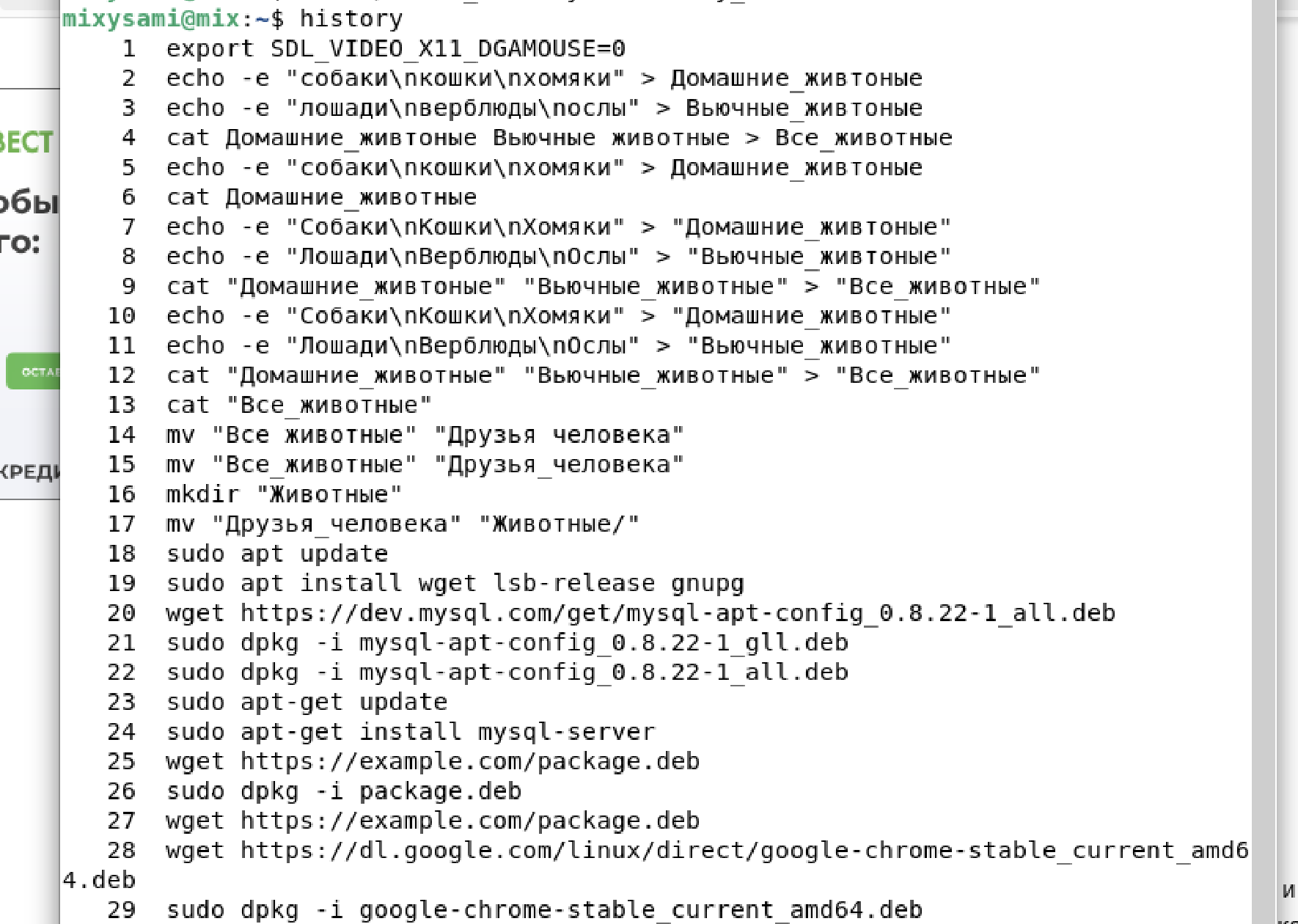


5. Вывод истории команд в терминале:

Вся история команд сохраняется в файле .bash\_history

cat ~/.bash\_history



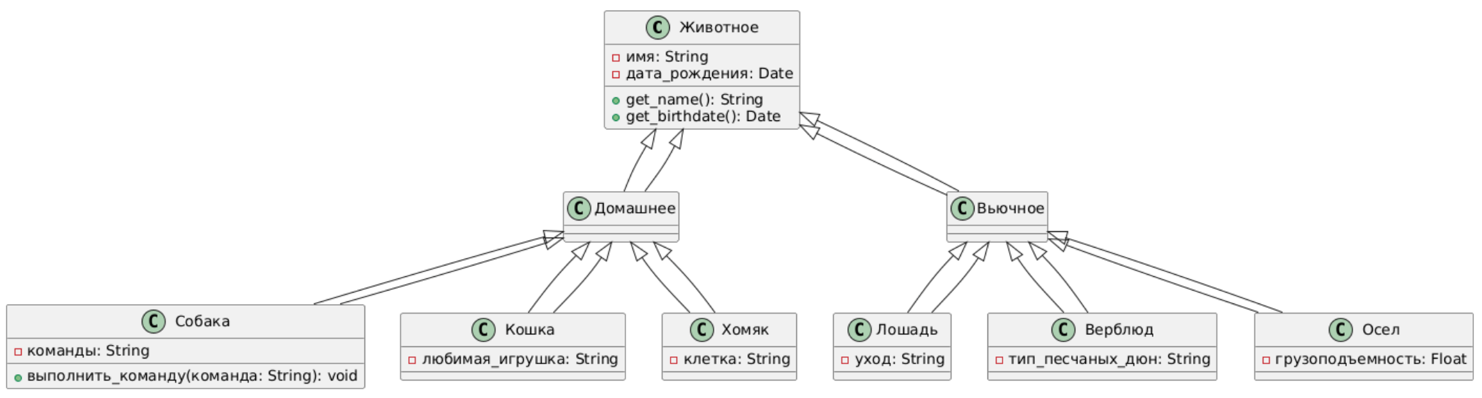




6. Диаграмма классов

Вот пример UML-диаграммы классов:

`



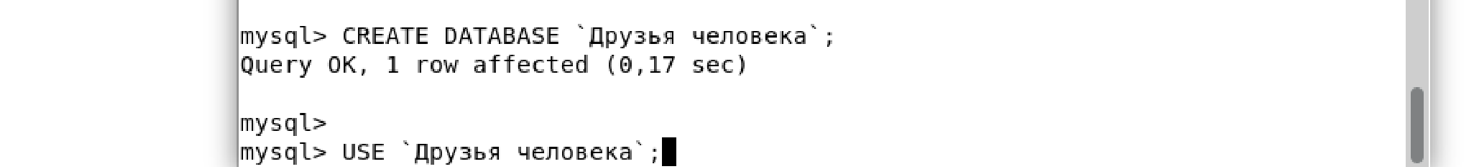
7. Создание базы данных “Друзья человека” в MySQL:

Запуск MySQL консоли

sudo mysql -u root -p

Создание базы данных

CREATE DATABASE `Друзья человека`;



8. Создание таблиц с иерархией в базе данных:

sql

USE `Друзья человека`;

CREATE TABLE Домашние (

id INT PRIMARY KEY AUTO\_INCREMENT,

вид VARCHAR(255)

);

CREATE TABLE Вьючные (

id INT PRIMARY KEY AUTO\_INCREMENT,

вид VARCHAR(255)

);

CREATE TABLE Собаки (

id INT PRIMARY KEY AUTO\_INCREMENT,

имя VARCHAR(255),

команды VARCHAR(255),

дата\_рождения DATE

);

CREATE TABLE Лошади (

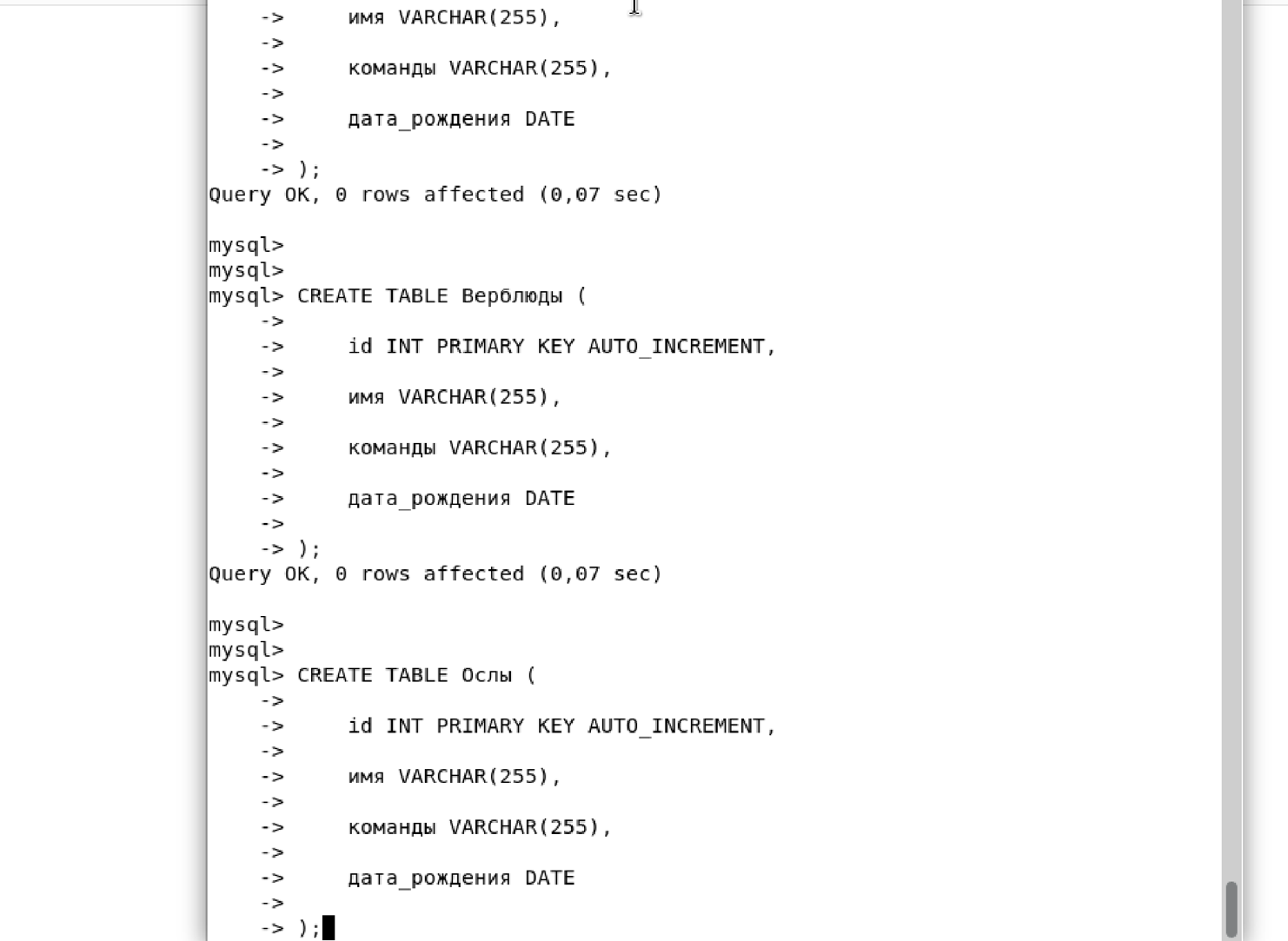
id INT PRIMARY KEY AUTO\_INCREMENT,

имя VARCHAR(255),

команды VARCHAR(255),

дата\_рождения DATE

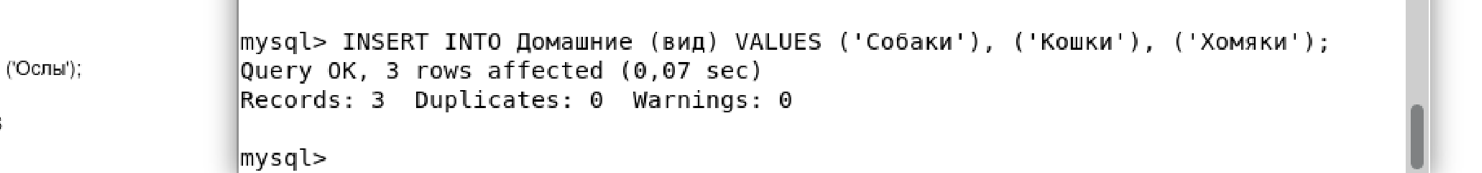
);

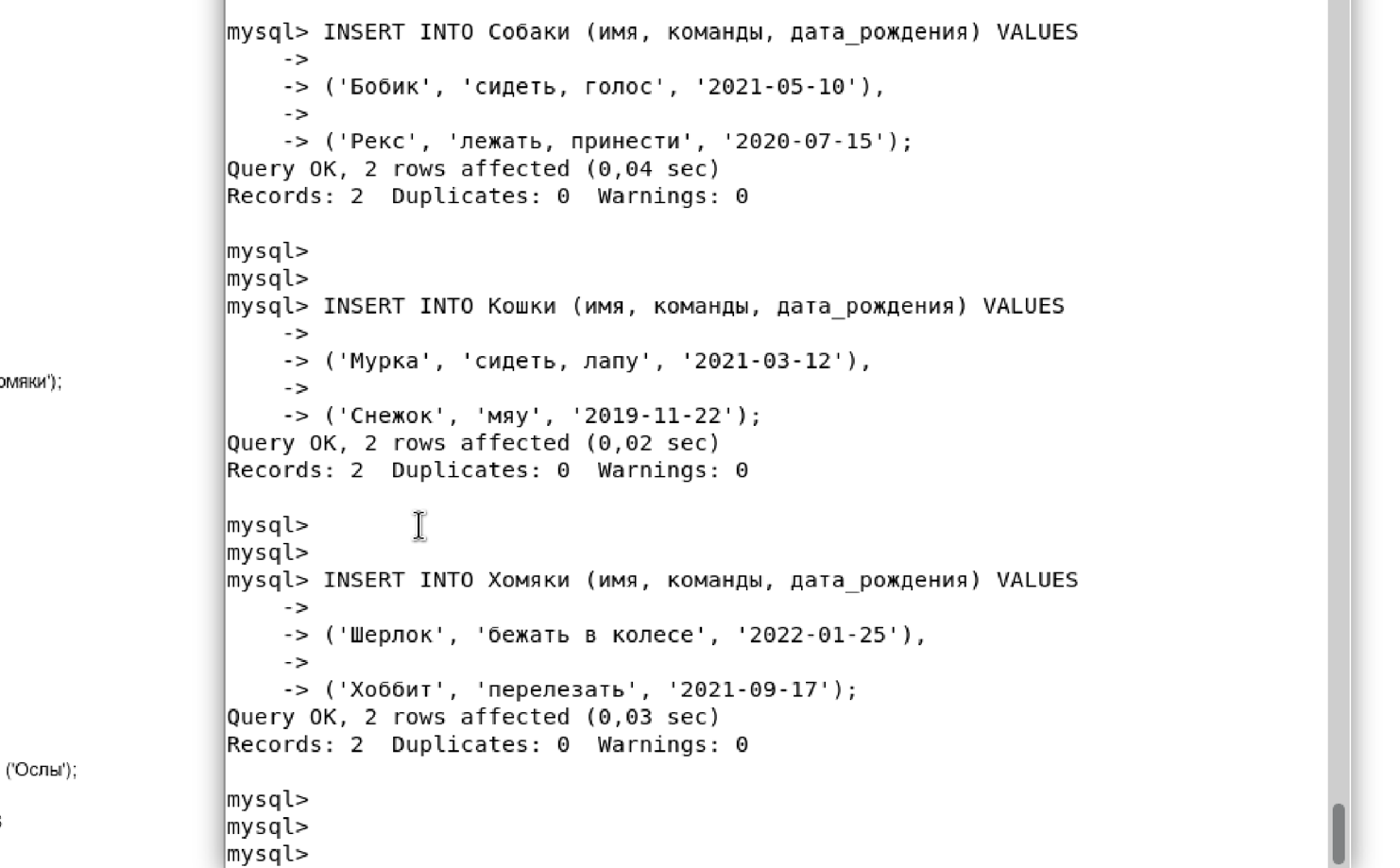


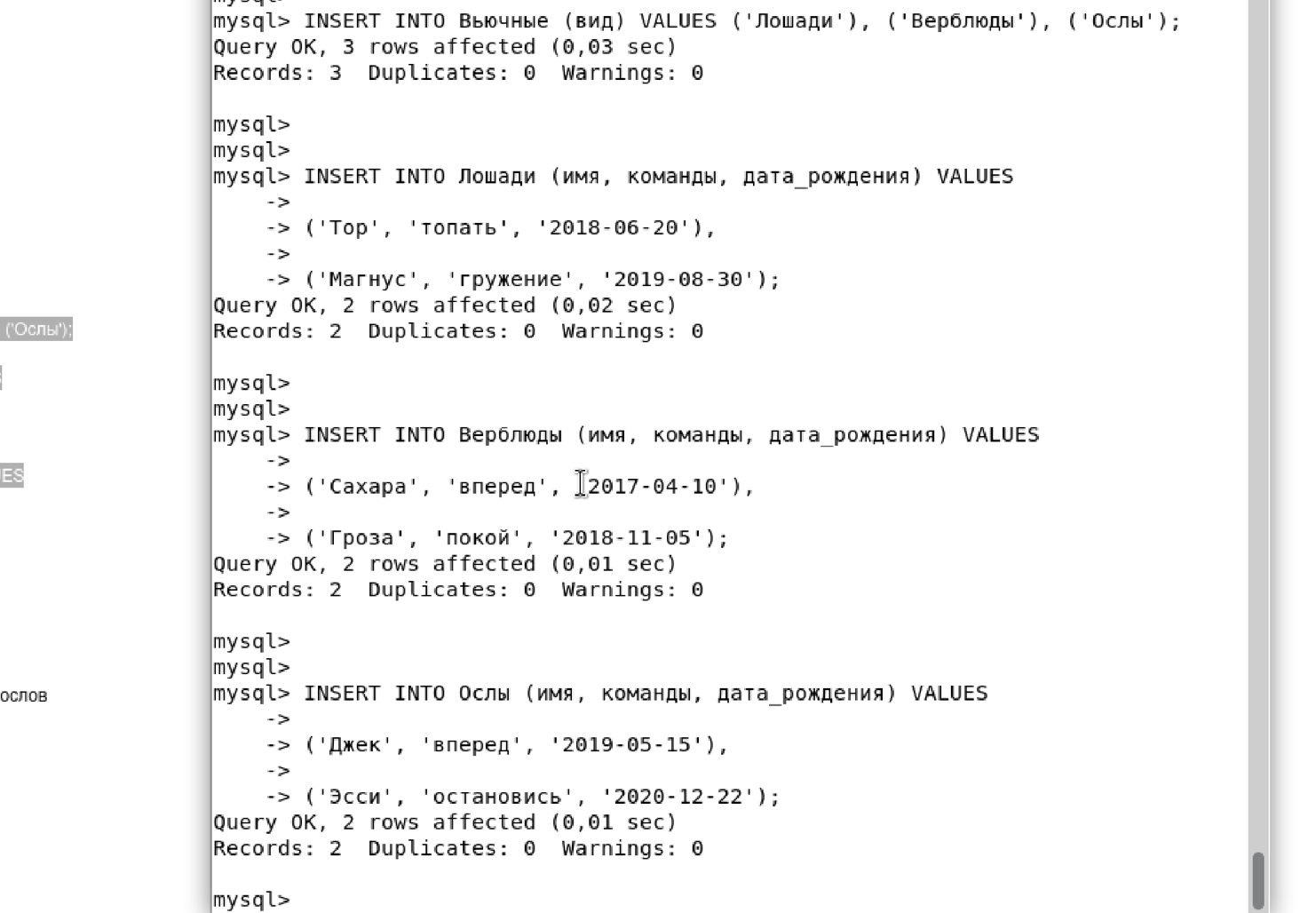
9. Заполнение низкоуровневых таблиц:

sql

INSERT INTO Собаки (имя, команды, дата\_рождения) VALUES ('Бобик', 'сидеть, голос', '2021-05-10');



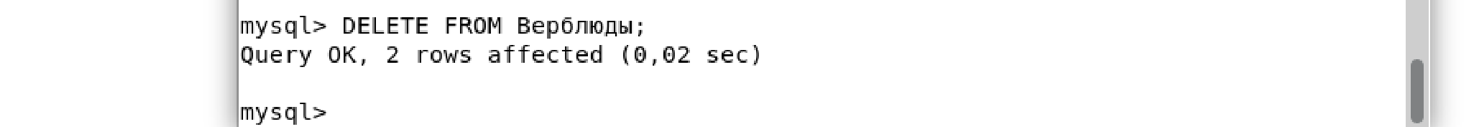




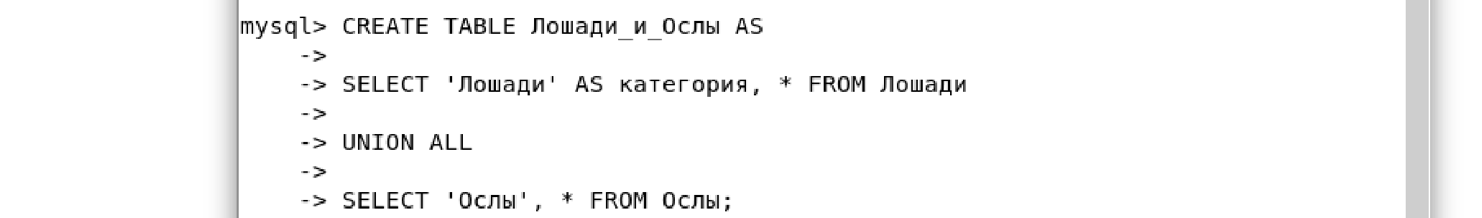
10. Удаление верблюдов и объединение таблиц "Лошади" и "Ослы":

sql

DELETE FROM Вьючные WHERE вид = 'Верблюды';



Объединение таблиц



11. Создание таблицы "Молодые животные" с расчетом возраста:

sql

CREATE TABLE Молодые\_животные AS

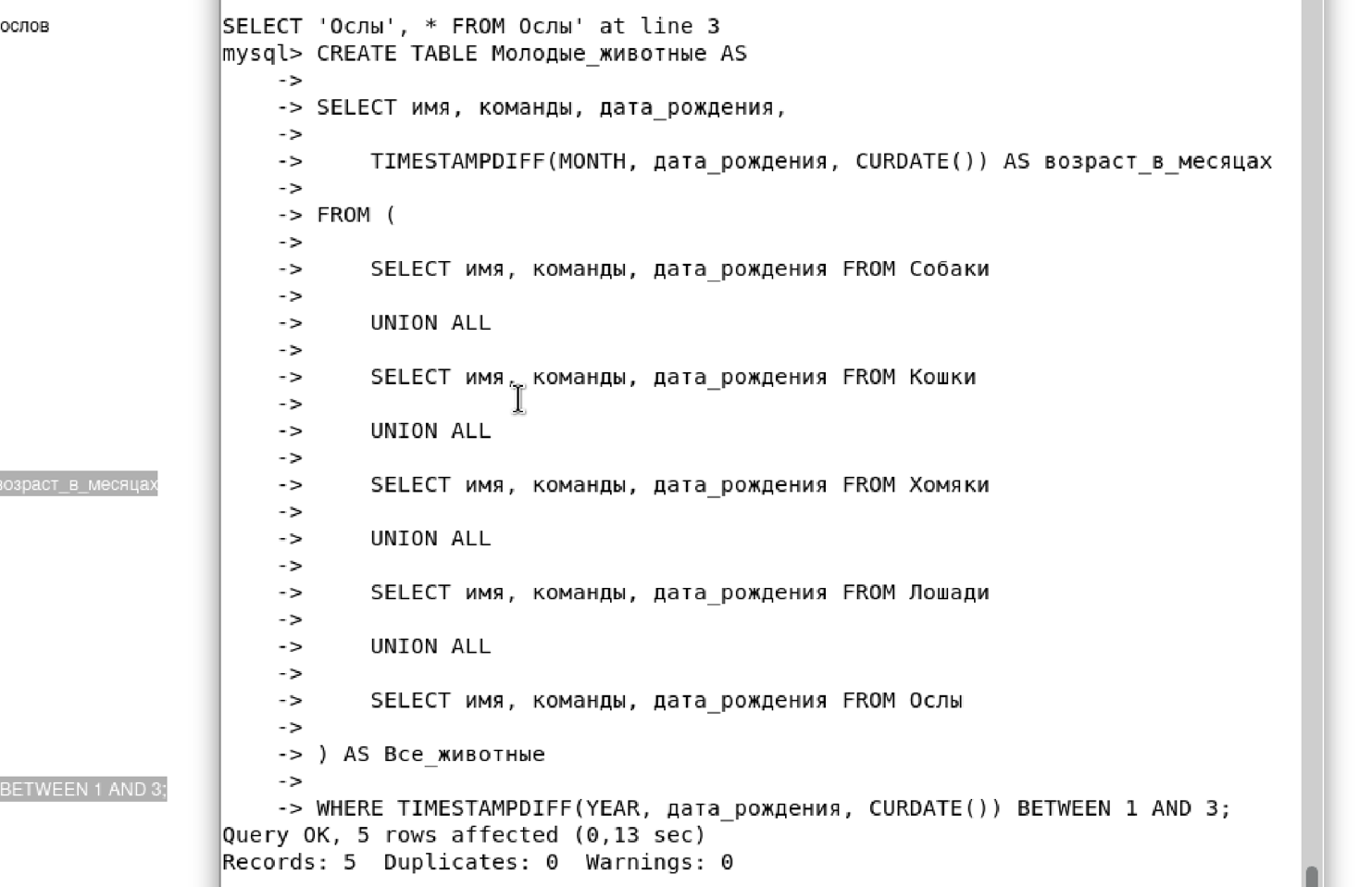
SELECT \*,

TIMESTAMPDIFF(MONTH, дата\_рождения, CURDATE()) AS возраст\_в\_месяцах

FROM Собаки

WHERE TIMESTAMPDIFF(YEAR, дата\_рождения, CURDATE()) BETWEEN 1 AND 3;

-- Аналогично добавьте данные из других таблиц



12. Объединение всех таблиц в одну с сохранением информации о прошлой принадлежности:

sql

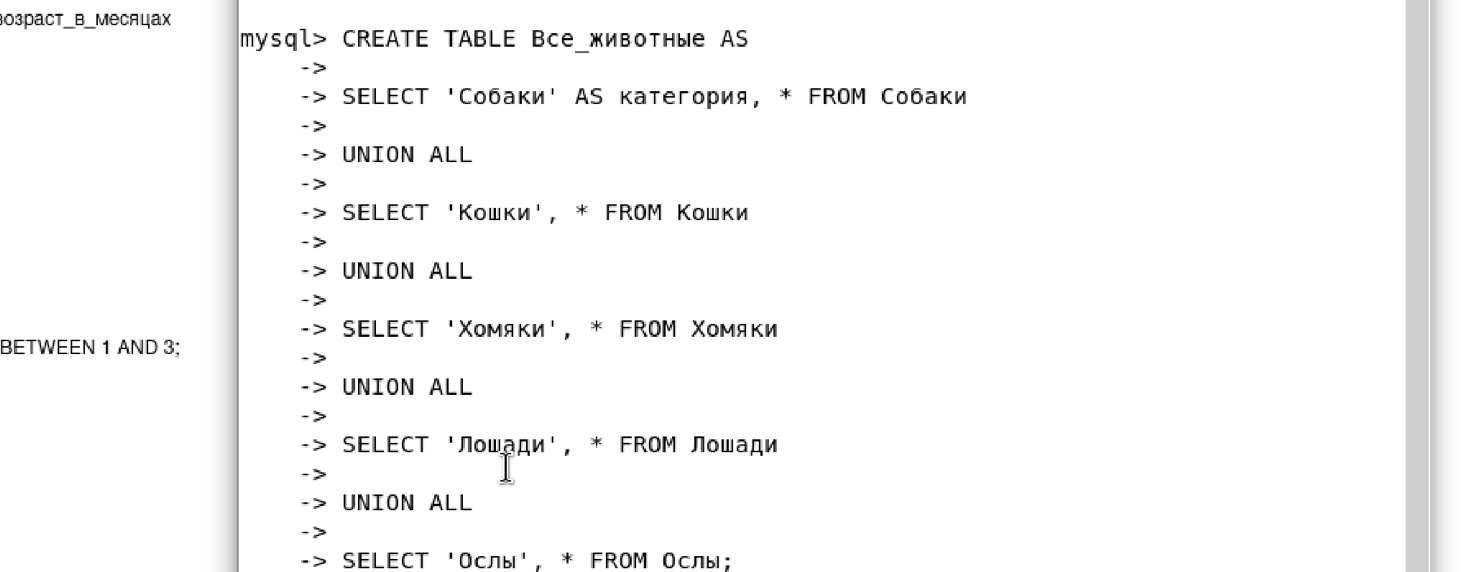
CREATE TABLE Все\_животные AS

SELECT 'Собаки' AS категория, \* FROM Собаки

UNION ALL

SELECT 'Кошки', \* FROM Кошки

-- Продолжайте добавлять объединения для всех таблиц



13. Пример класса с инкапсуляцией и наследованием:

python

class Животное:

def \_\_init\_\_(self, имя, дата\_рождения):

self.\_\_имя = имя

self.\_\_дата\_рождения = дата\_рождения

def get\_name(self):

return self.\_\_имя

def get\_birthdate(self):

return self.\_\_дата\_рождения

class Домашнее(Животное):

pass

class Вьючное(Животное):

pass

class Собака(Домашнее):

def выполнить\_команду(self, команда):

print(f"{self.get\_name()} выполняет команду {команда}")

Пример использования:

собака = Собака("Бобик", "2021-05-10")

собака.выполнить\_команду("сидеть")