1. Используя команду cat в терминале операционной системы Linux, создать два файла Домашние животные (заполнив файл собаками, кошками, хомяками) и Вьючные животными заполнив файл Лошадьми, верблюдами и ослы), а затем объединить их. Просмотреть содержимое созданного файла. Переименовать файл, дав ему новое имя (Друзья человека).

Создание файлов и заполнение их:

cat > 'Домашние животные'

cat > 'Вьючные животные'

Объединение в один файл:

cat Домашние\ животные Вьючные\ животные > 'Животные'

cat Животные

Переименование файла:

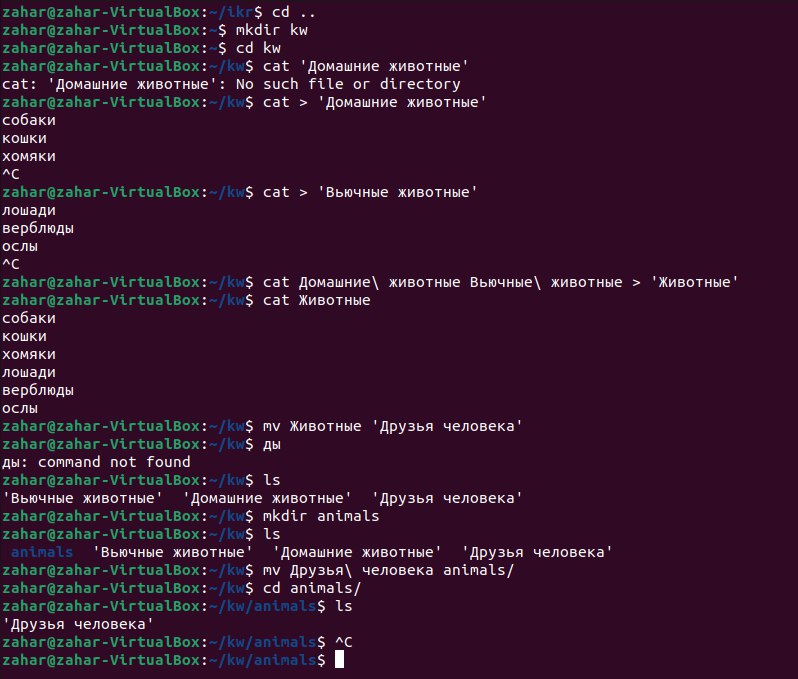
mv Животные 'Друзья человека'

1. Создать директорию, переместить файл туда.

mkdir animals

mv Друзья\ человека animals/

cd animals/



1. Подключить дополнительный репозиторий MySQL. Установить любой пакет из этого репозитория.

Скачивание файла:

wget <https://dev.mysql.com/get/mysql-apt-config_0.8.24-1_all.deb>

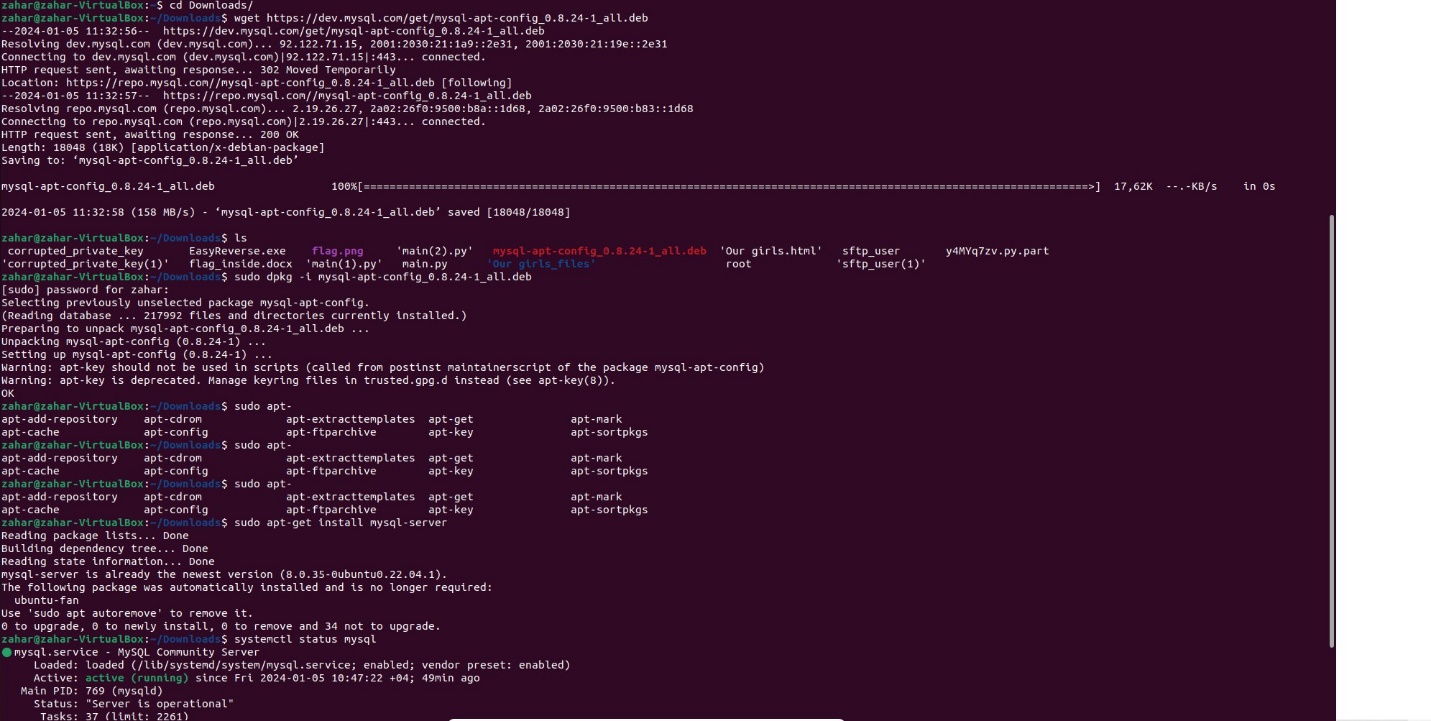
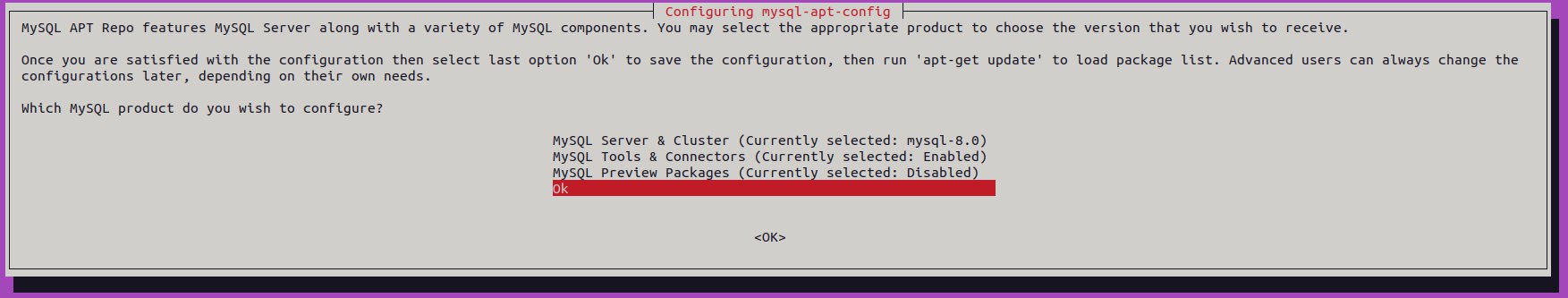
Установка файла:

sudo dpkg -i mysql-apt-config\_0.8.24-1\_all.deb

sudo apt-get install mysql-server

Проверка работоспособности:

systemctl status mysql



1. Установить и удалить deb-пакет с помощью dpkg:

Скачивание deb-пакета:

wget <http://ftp.de.debian.org/debian/pool/main/z/zip/zip_3.0-13_amd64.deb>

Установка пакета:

sudo dpkg -i zip\_3.0-13\_amd64.deb

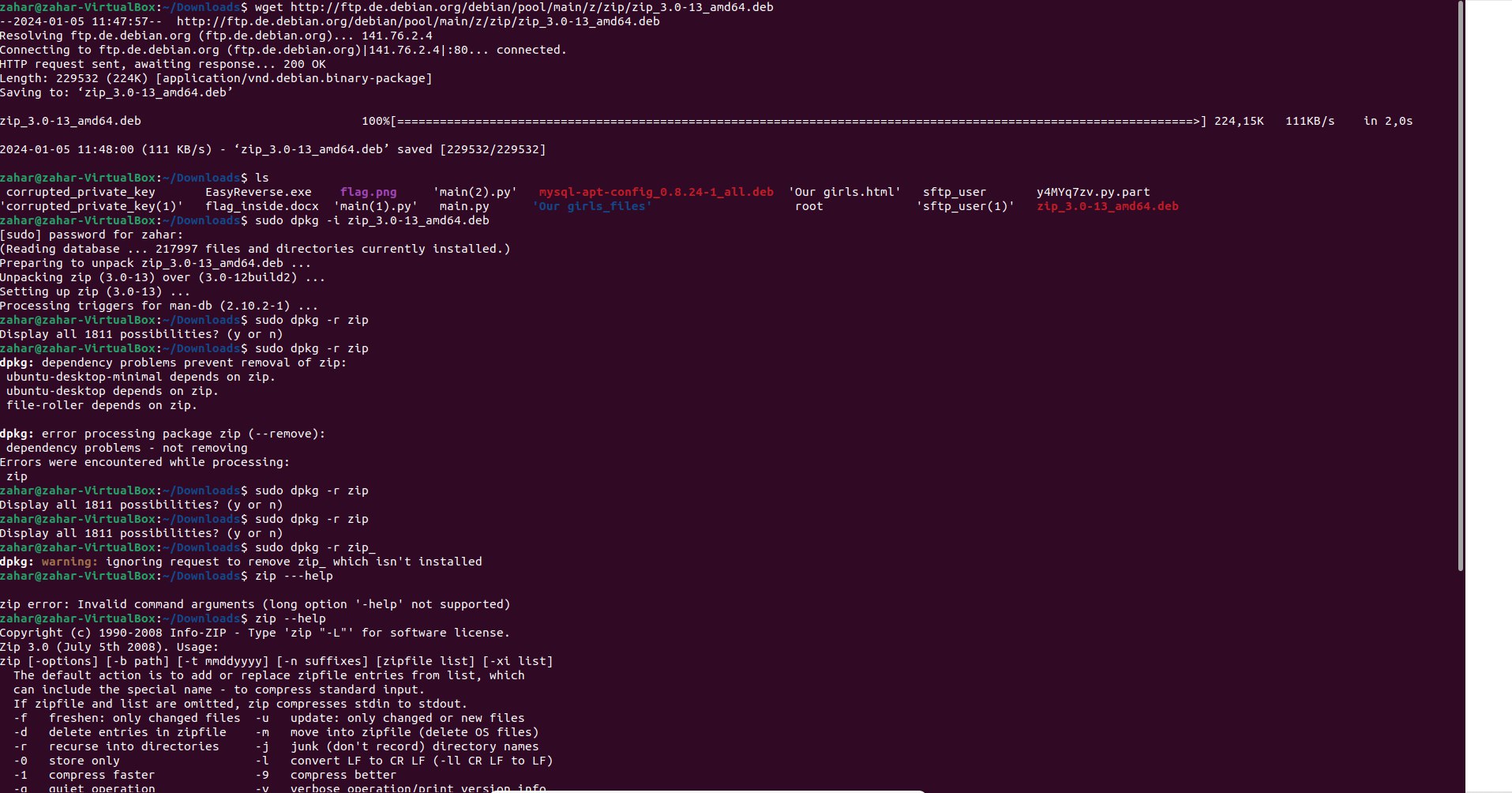
sudo dpkg -r zip

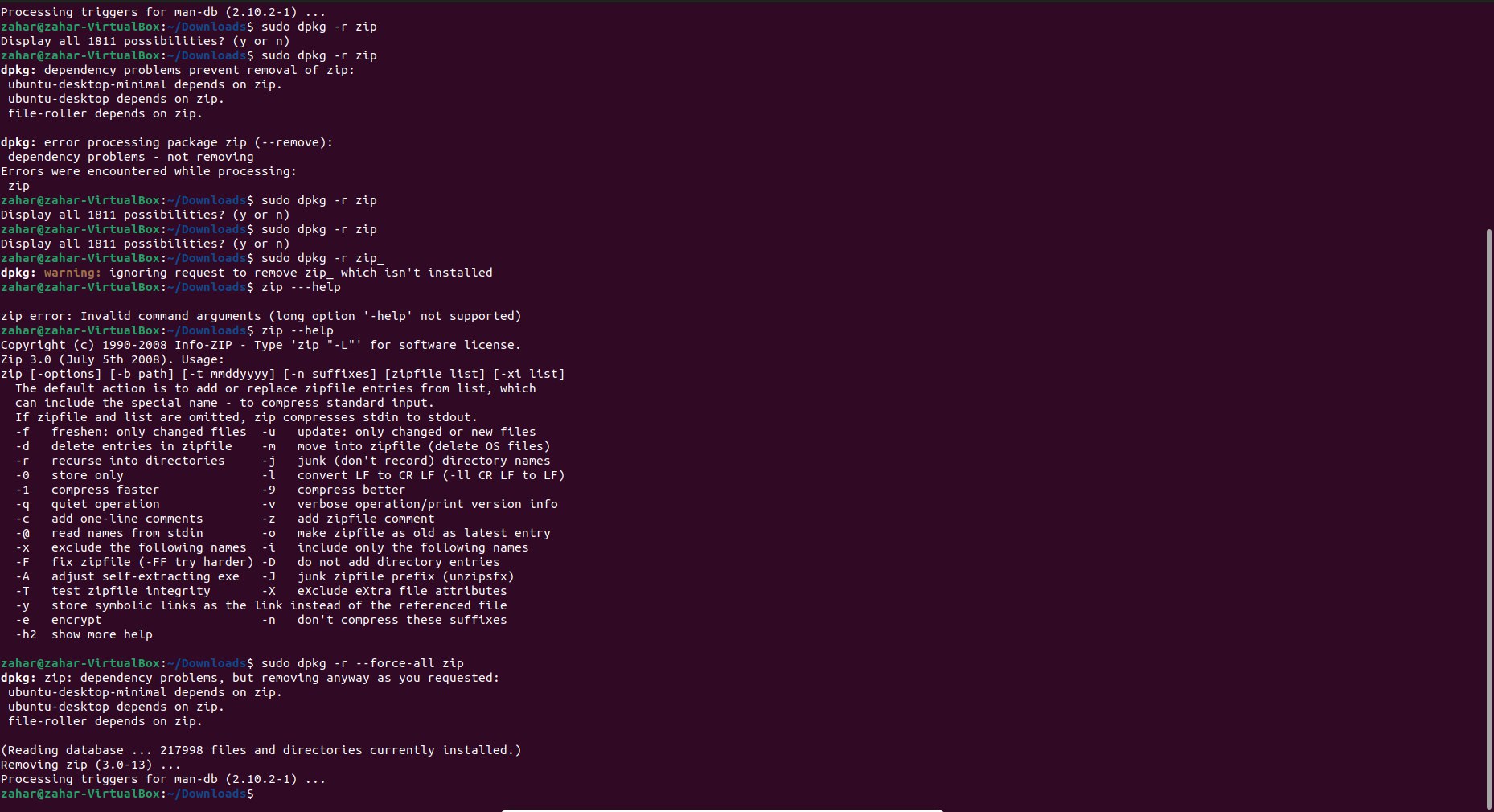
Проверка работоспособности:

zip –help

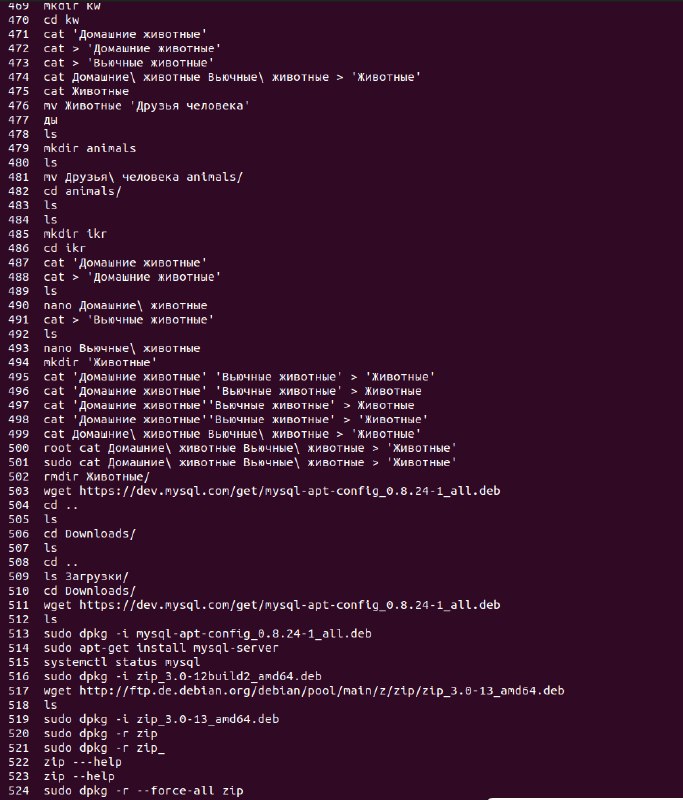
Удаление файла:

sudo dpkg -r --force-all zip

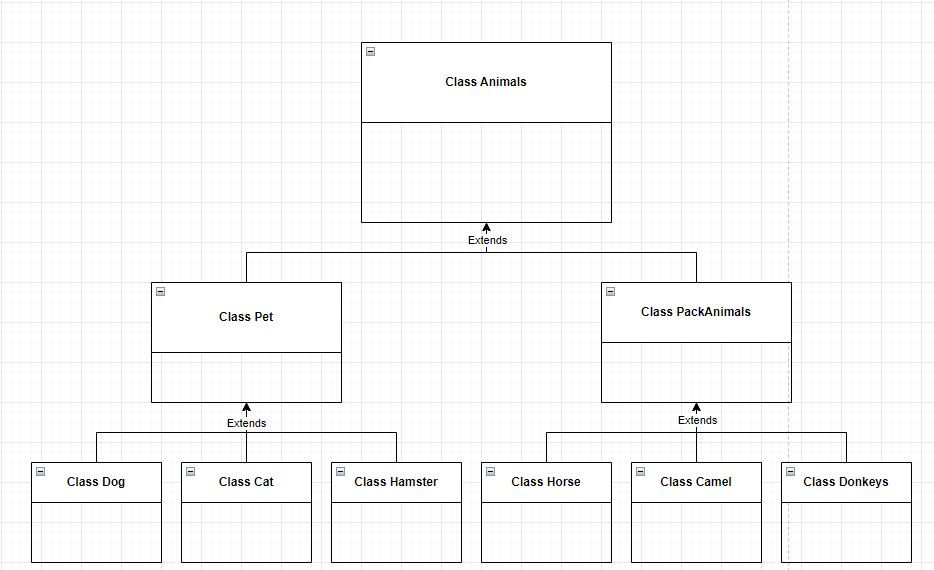




1. Выложить историю команд в терминале ubuntu



1. Нарисовать диаграмму, в которой есть класс родительский класс, домашние животные и вьючные животные, в составы которых в случае домашних животных войдут классы:

собаки, кошки, хомяки, а в класс вьючные животные войдут: Лошади, верблюды и ослы). 

1. В подключенном MySQL репозитории создать базу данных “Друзья человека”

create schema animals;

1. Создать таблицы с иерархией из диаграммы в БД

use animals;

create table dog(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

create table cat(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

create table hamster(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

create table horse(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

create table camel(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

create table donkey(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date

);

1. Заполнить низкоуровневые таблицы именами(животных), командами которые они выполняют и датами рождения

insert into dog (a\_name, commands, birthday) values

('dog1', 'gav', '2020-01-12'),

('dog2', 'gav', '2023-03-06'),

('dog3', 'gav', '2021-04-11'),

('dog4', 'gav', '2022-11-01'),

('dog5', 'gav', '2017-08-03');

insert into cat (a\_name, commands, birthday) values

('cat1', 'meow', '2018-02-01'),

('cat2', 'meow', '2023-06-06'),

('cat3', 'meow', '2022-05-12'),

('cat4', 'meow', '2023-01-04'),

('cat5', 'meow', '2015-03-02');

insert into hamster (a\_name, commands, birthday) values

('hamster1', 'pip', '2021-05-11'),

('hamster2', 'pip', '2020-07-12'),

('hamster3', 'pip', '2023-12-01'),

('hamster4', 'pip', '2015-10-06'),

('hamster5', 'pip', '2019-03-02');

insert into horse (a\_name, commands, birthday) values

('horse1', 'neigh', '2017-05-01'),

('horse2', 'neigh', '2019-02-11'),

('horse3', 'neigh', '2014-03-12'),

('horse4', 'neigh', '2015-01-11'),

('horse5', 'neigh', '2023-01-11');

insert into camel (a\_name, commands, birthday) values

('camel1', 'ooow', '2022-05-11'),

('camel2', 'ooow', '2017-05-11'),

('camel3', 'ooow', '2020-03-10'),

('camel4', 'ooow', '2019-11-03'),

('camel5', 'ooow', '2017-06-09');

insert into donkey (a\_name, commands, birthday) values

('donkey1', 'ihaw', '2020-01-12'),

('donkey2', 'ihaw', '2023-03-06'),

('donkey3', 'ihaw', '2021-04-11'),

('donkey4', 'ihaw', '2022-11-01'),

('donkey5', 'ihaw', '2017-08-03');

1. Удалив из таблицы верблюдов, т.к. верблюдов решили перевезти в другой питомник на зимовку. Объединить таблицы лошади, и ослы в одну таблицу.

truncate camel;

insert into horse (a\_name, commands, birthday)

select a\_name, commands, birthday

from donkey;

drop table donkey;

rename table horse to horse\_and\_donkey;

1. Создать новую таблицу “молодые животные” в которую попадут все животные старше 1 года, но младше 3 лет и в отдельном столбце с точностью до месяца подсчитать возраст животных в новой таблице

create table young\_animal(

id int primary key auto\_increment,

a\_name char(30),

commands text,

birthday date,

age text

);

DELIMITER //

create function age(b\_date date) returns text

deterministic

begin

declare curr\_date date;

declare age\_years int;

declare age\_months int;

set curr\_date = curdate();

set age\_years = year(curr\_date) - year(b\_date);

set age\_months = month(curr\_date) - month(b\_date);

if day(curr\_date) < day(b\_date) then

set age\_months = age\_months - 1;

end if;

if age\_months < 0 then

set age\_years = age\_years - 1;

set age\_months = age\_months + 12;

end if;

return concat(age\_years, ' years ', age\_months, 'months');

end //

DELIMITER ;

insert into young\_animal(a\_name, commands, birthday, age)

select a\_name, commands, birthday, age(birthday)

from dog

where timestampdiff(year, birthday, curdate()) between 1 and 2

union all

select a\_name, commands, birthday, age(birthday)

from cat

where timestampdiff(year, birthday, curdate()) between 1 and 2

union all

select a\_name, commands, birthday, age(birthday)

from hamster

where timestampdiff(year, birthday, curdate()) between 1 and 2

union all

select a\_name, commands, birthday, age(birthday)

from horse\_and\_donkey

where timestampdiff(year, birthday, curdate()) between 1 and 2;

1. Объединить все таблицы в одну, при этом сохраняя поля, указывающие на прошлую принадлежность к старым таблицам.

set SQL\_SAFE\_UPDATES = 0;

delete from dog

where timestampdiff(year, birthday, curdate()) between 1 and 2;

delete from cat

where timestampdiff(year, birthday, curdate()) between 1 and 2;

delete from hamster

where timestampdiff(year, birthday, curdate()) between 1 and 2;

delete from horse\_and\_donkey

where timestampdiff(year, birthday, curdate()) between 1 and 2;

1. Создать класс с Инкапсуляцией методов и наследованием по диаграмме.

в gb/JavaApp/Animals

1. Написать программу, имитирующую работу реестра домашних животных. В программе должен быть реализован следующий функционал:
   1. Завести новое животное
   2. определять животное в правильный класс
   3. увидеть список команд, которое выполняет животное
   4. обучить животное новым командам
   5. Реализовать навигацию по меню
2. Создайте класс Счетчик, у которого есть метод add(), увеличивающий̆ значение внутренней̆ int переменной̆на 1 при нажатие “Завести новое животное” Сделайте так, чтобы с объектом такого типа можно было работать в блоке try-with-resources. Нужно бросить исключение, если работа с объектом типа счетчик была не в ресурсном try и/или ресурс остался открыт. Значение считать в ресурсе try, если при заведения животного заполнены все поля.

В gb/JavaApp/Animals/Counter