

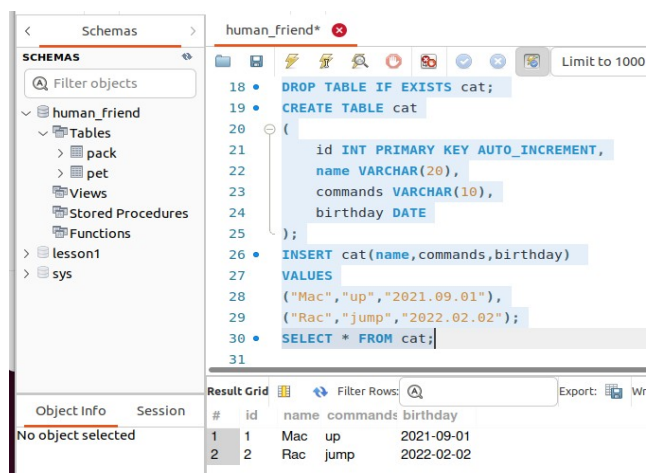
7. В подключенном MySQL репозитории создать базу данных "Друзья человека"

```
DROP DATABASE IF EXISTS human_friend;
CREATE DATABASE IF NOT EXISTS human_friend;
USE human_friend;
```

8. Создать таблицы с иерархией из диаграммы в БД

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

```
DROP TABLE IF EXISTS cat;
CREATE TABLE cat
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    commands VARCHAR(10),
    birthday DATE
);
INSERT cat(name,commands,birthday)
VALUES
("Mac","up","2021.09.01"),
("Rac","jump","2022.02.02");
SELECT * FROM cat;
```



```
CREATE TABLE IF NOT EXISTS dog
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    commands VARCHAR(10),
    birthday DATE
);
INSERT dog(name,commands,birthday)
VALUES
("Mad","up","2023.01.01"),
("Rad","jump","2016.02.02.");
SELECT * FROM dog;
```

```
CREATE TABLE IF NOT EXISTS namster
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    commands VARCHAR(10),
    birthday DATE
);
INSERT namster(name,commands,birthday)
VALUES
("Man","up","2020.01.01."),
("Dan","jump","2022.02.02.");
SELECT * FROM namster;
```

```
CREATE TABLE IF NOT EXISTS horse
(
```

```

        id INT PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(20),
        commands VARCHAR(10),
        birthday DATE
    );
INSERT horse(name,commands,birthday)
VALUES
("Mah", "up", "2018.01.01"),
("Rah", "jump", "2021.02.02.");
SELECT * FROM horse;

```

```

CREATE TABLE IF NOT EXISTS camel
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    commands VARCHAR(10),
    birthday DATE
);
INSERT camel(name,commands,birthday)
VALUES
("Mac", "up", "2012.01.01."),
("Dac", "jump", "2021.02.02");
SELECT * FROM camel;

```

```

CREATE TABLE IF NOT EXISTS donkey
(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(20),
    commands VARCHAR(10),
    birthday DATE
);
INSERT donkey(name,commands,birthday)
VALUES
("Mad", "up", "2022.01.01"),
("Gad", "jump", "2015.02.02");
SELECT * FROM donkey;

```

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

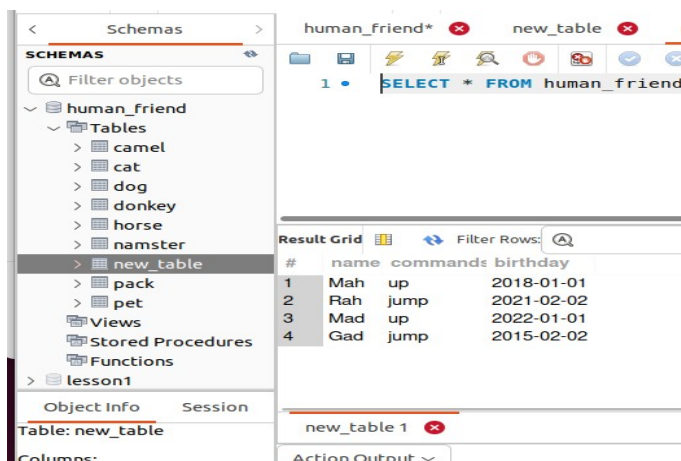
DROP TABLE camel;

Объединить таблицы лошади, и ослы в одну таблицу.

```

CREATE TABLE new_table
SELECT horse.name,horse.commands,horse.birthday
FROM horse
UNION
SELECT donkey.name,donkey.commands,donkey.birthday
FROM donkey;

```

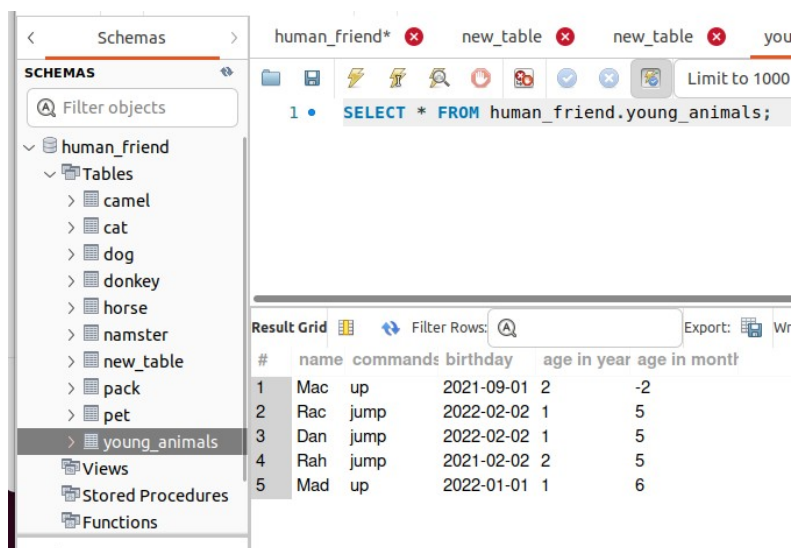


The screenshot shows a database management interface. On the left, the 'Schemas' panel lists various tables, including 'camel', 'cat', 'dog', 'donkey', 'horse', 'namster', 'new_table', 'pack', 'pet', 'Views', 'Stored Procedures', and 'Functions'. The 'new_table' is selected. In the center, the SQL query 'SELECT * FROM human_friend' is entered. On the right, the 'Result Grid' displays the data from the query, showing four rows of data with columns: #, name, commands, and birthday.

#	name	commands	birthday
1	Mah	up	2018-01-01
2	Rah	jump	2021-02-02
3	Mad	up	2022-01-01
4	Gad	jump	2015-02-02

11. 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_animals
SELECT name, commands, birthday, year(CURRENT_DATE())-year(birthday) AS "age in year", month(CURRENT_DATE())-month(birthday) AS "age in month"
FROM cat
WHERE ((year(CURRENT_DATE())-year(birthday))>=1)&((year(CURRENT_DATE())-year(birthday))<3)
UNION
SELECT name, commands, birthday, year(CURRENT_DATE())-year(birthday) AS "age in year", month(CURRENT_DATE())-month(birthday) AS "age in month"
FROM dog
WHERE ((year(CURRENT_DATE())-year(birthday))>=1)&((year(CURRENT_DATE())-year(birthday))<3)
UNION
SELECT name, commands, birthday, year(CURRENT_DATE())-year(birthday) AS "age in year", month(CURRENT_DATE())-month(birthday) AS "age in month"
FROM namster
WHERE ((year(CURRENT_DATE())-year(birthday))>=1)&((year(CURRENT_DATE())-year(birthday))<3)
UNION
SELECT name, commands, birthday, year(CURRENT_DATE())-year(birthday) AS "age in year", month(CURRENT_DATE())-month(birthday) AS "age in month"
FROM horse
WHERE ((year(CURRENT_DATE())-year(birthday))>=1)&((year(CURRENT_DATE())-year(birthday))<3)
UNION
SELECT name, commands, birthday, year(CURRENT_DATE())-year(birthday) AS "age in year", month(CURRENT_DATE())-month(birthday) AS "age in month"
FROM donkey
WHERE ((year(CURRENT_DATE())-year(birthday))>=1)&((year(CURRENT_DATE())-year(birthday))<3);
```



The screenshot shows a database management interface. On the left, a tree view displays the 'human_friend' schema with tables: camel, cat, dog, donkey, horse, namster, new_table, pack, pet, and young_animals. The 'young_animals' table is selected. The main window shows the SQL query: `SELECT * FROM human_friend.young_animals;` and the resulting data grid.

#	name	commands	birthday	age in year	age in month
1	Mac	up	2021-09-01	2	-2
2	Rac	jump	2022-02-02	1	5
3	Dan	jump	2022-02-02	1	5
4	Rah	jump	2021-02-02	2	5
5	Mad	up	2022-01-01	1	6

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