

Задание

1. Есть три таблицы:

```
CREATE TABLE Patients (  
    PRIMARY KEY (id),  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL  
)  
COMMENT "Пациенты";  
  
CREATE TABLE Events (  
    PRIMARY KEY (id),  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    create_date DATETIME DEFAULT CURRENT_DATE() NOT NULL,  
    patient_id INT NOT NULL,  
    FOREIGN KEY (patient_id) REFERENCES Patients(id)  
)  
COMMENT "Обращения пациентов";  
  
CREATE TABLE Actions (  
    PRIMARY KEY (id),  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    event_id INT NOT NULL,  
    FOREIGN KEY (event_id) REFERENCES Events(id)  
)  
COMMENT "Действия при обращениях (анализы и т.п.)";
```

Напишите запрос, результатом которого будут следующие столбцы:

- "ФИО пациента" (фамилия и имя пациента через пробел в одном столбце);
- "Дата обращения";
- "Наименование действия".

В кавычках указано название столбца в результате выборки. Сделать вышеописанную выборку для пациентов с фамилией Иванов, именем Иван.

2. Есть таблица:

```
CREATE TABLE Patients (  
    PRIMARY KEY (id),  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    last_name VARCHAR(255) NOT NULL,  
    first_name VARCHAR(255) NOT NULL,  
    patronymic VARCHAR(255) DEFAULT '' NOT NULL,  
    deleted TINYINT(1) DEFAULT 0 NOT NULL  
)  
COMMENT "Пациенты";
```

Напишите запрос, результатом которого будет:

- Добавление колонки sex ("Пол"): 0-неопределено, 1-М, 2-Ж,
- Проставление значения колонки sex в соответствии с данными таблицы (last_name, first_name, patronymic и т.п.). Помните, что данных может больше 10000!

Решение

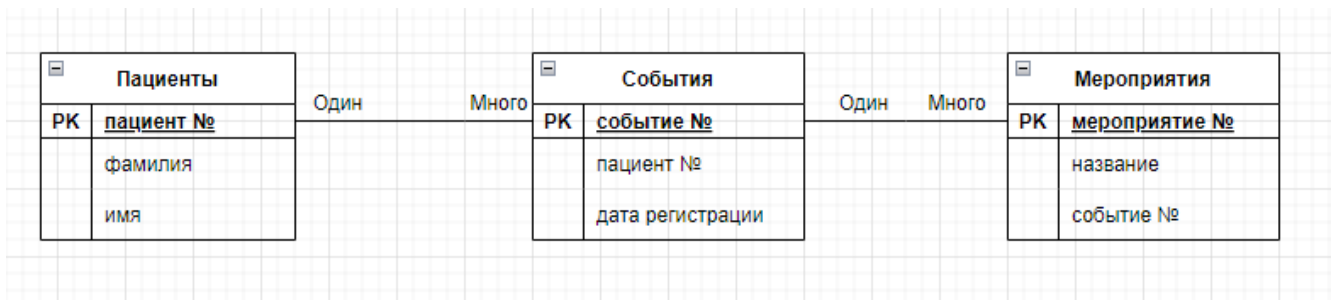


Рисунок 1 – Инфологическое представление

```
use db_1;
select p.id, concat(last_name, ' ', first_name), create_date, name
from patients p
left join events ev
on ev.id=p.id
left join actions ac
on ev.id=ac.event_id
where p.last_name='Ivanov' and p.first_name='Ivan';
```

```
1 • use db_1;
2 • select p.id, concat(last_name, ' ', first_name), create_date, name
3   from patients p
4   left join events ev
5   on ev.id=p.id
6   left join actions ac
7   on ev.id=ac.event_id
8   where p.last_name='Ivanov' and p.first_name='Ivan';
```

MySQL Workbench interface showing a query execution for the 'patients_2' table. The query filters for patients with last_name 'Ivanov' and first_name 'Ivan'. The result grid shows one row: id 1, name 'Ivanov Ivan', create_date '2022-11-22 00:30:01', and name 'dentist job'.

Query:

```

1 • use db_1;
2 • select p.id, concat(last_name, ' ', first_name), create_date, name
3   from patients p
4  left join events ev
5    on ev.id=p.id
6  left join actions ac
7    on ev.id=ac.event_id
8  where p.last_name='Ivanov' and p.first_name='Ivan';
9

```

Result Grid:

#	id	concat(last_name, ' ', first_name)	create_date	name
1	1	Ivanov Ivan	2022-11-22 00:30:01	dentist job

Table: patients_2

Columns:

- id: int AI PK
- last_name: varchar(255)
- first_name: varchar(255)
- patronymic: varchar(255)
- deleted: tinyint(1)
- sex: varchar(1)

Action Output:

#	Time	Action	Message
2	03:15:58	select p.id, concat(last_name, ' ', first_name), create_date, name from patients p left joi...	1 row(s) returned
3	03:16:26	SELECT * FROM db_1.patients_2 LIMIT 0, 1000	2 row(s) returned
4	03:16:27	SELECT * FROM db_1.patients_2 LIMIT 0, 1000	2 row(s) returned

```
from django.db import models
```

```

class Patients(models.Model):
    UNDEF='0'
    MALE='1'
    FEMALE='2'
    SEX_CHOICES=[
        (UNDEF, 'undef'),
        (MALE, 'male'),
        (FEMALE, 'female'),
    ]
    firstname=models.CharField(max_length=255)
    lastname = models.CharField(max_length=255)
    patronymic= models.CharField(max_length=255,default='', null=False)
    sex=models.CharField(
        max_length=1,
        choices=SEX_CHOICES,
        default=UNDEF,
    )

    def __str__(self):
        return "%s %s" % (self.firstname, self.lastname )

```

```
class Events(models.Model):
```

```
create_date=models.DateField()
patient_id=models.ForeignKey(Patients, on_delete=models.CASCADE, null=True)
```

```
class Actions(models.Model):
    name=models.CharField(max_length=255, null=True)
    event_id=models.ForeignKey(Events, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return self.name
```

Django administration

WELCOME, TALANT_RAHIMBERDIEV. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home » Analitic » Patientss » Sergey Sergeev

Start typing to filter...

ANALITIC

Actionss [+ Add](#)

Eventss [+ Add](#)

Patientss [+ Add](#)

APP

Tasks [+ Add](#)

Users [+ Add](#)

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

Change patients

HISTORY

Firstname:

Lastname:

Patronymic:

Sex:

Delete

Save and add another

Save and continue editing

SAVE