МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**ОТЧЁТ**

**по лабораторной работе №2**

«Проектирование базы данных**»**

**по дисциплине: «***Технологии и методы программирования***»**

|  |  |
| --- | --- |
| Выполнил(а):  Студенты гр. «АБс-123», «АВТФ»  Номоконов Данила Сергеевич  Сашенко Алексей Александрович  Оверченко Егор Евгеньевич  Козлов Александр Дмитриевич  «29» мая 2023г | Проверил: ассистент кафедры ЗИ Медведев М.А.  «\_\_\_» \_\_\_\_\_\_ 2023 г.  *(оценка, подпись)* |

Новосибирск 2023

**Цель:**

Спроектировать базу данных согласно заданному набору атрибутов и сущностей

**Задачи:**

* Составить концептуальную схему;
* Провести логическое проектирование;
* Провести физическое проектирование;

**Ход работы:**

Концептуальное проектирование

Концептуальное проектирование базы данных - это процесс создания модели информации, не зависящей от любых физических аспектов ее представления.

Этапы концептуального проектирования:

1. Определение сущностей

|  |  |
| --- | --- |
| **Сущность** | **Определение** |
| Призывник | Содержит информацию о призывнике |
| Персональные данные | Хранит информацию о персональных данных |
| Родственник | Хранит сведения о родственнике призывника |
| Личное дело | Содержит сведения о составе личного дела призывника |
| Медицинское заключение | Хранит информацию о здоровье призывника |
| Тип семьи | Содержит данные о типе семьи призывника |
| Призыв | Хранит информацию о дате и месте призыва |
| Военная часть | Содержит сведения о военной части призывника |
| Место работы | Содержит информацию о месте работы призывника |
| Место учёбы | Содержит информацию о месте учебы призывника |
| Родство | Хранит данные о родственниках призывника |

1. Определение связей между сущностями

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сущность 1** | **Кратность** | **Тип связи** | **Кратность** | **Сущность** |
| Место регистрации | 1..1 | Содержит | 1..1 | Персональные данные |
| Место работы | 1..\* | Содержит | 1..\* | Персональные данные |
| Место учёбы | 1..\* | Содержит | 1..\* | Персональные данные |
| Персональные данные | 1..1 | Содержит | 1..1 | Личное дело |
| Персональные данные | 1..1 | Содержит | 1..1 | Родственник |
| Призывник | 1..1 | Содержит | 1..\* | Родство |
| Родство | 1..1 | Содержит | 1..\* | Родственник |
| Призывник | 1..1 | Содержит | 1..1 | Лично дело |
| Личное дело | 1..\* | Содержит | 1..1 | Призыв |
| Личное дело | 1..\* | Содержит | 1..1 | Медицинское заключение |
| Личное дело | 1..\* | Содержит | 1..1 | Военная часть |

1. Определение атрибутов и их характеристики

**Призывник**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| Призывник ID | Integer | Целое число, больше нуля | Да | Нет | Да |
| Личное дело ID | Integer | Целое число, больше нуля | Нет | Нет | Да |

**Родство**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| Родство ID | Integer | Целое число, больше нуля | Да | Нет | Да |
| Родственник ID | Integer | Целое число, больше нуля | Нет | Нет | Да |
| Призывник ID | Integer | Целое число, больше нуля | Нет | Нет | Да |
| Степень родства | Varchar | Строка из букв кириллицы | Нет | Нет | Да |

**Персональные данные**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID персональных данных | Integer | Целое число, больше нуля | Да | Нет | Да |
| Фамилия | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Имя | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Отчество | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Дата рождения | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | Нет | Нет | Да |
| Место рождения | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| ID места регистрации | Integer | Целое число, больше нуля | Нет | Нет | Да |
| Судимость | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Национальность | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| ID типа семьи | Integer | Целое число, больше нуля | Нет | Нет | Да |
| Паспортные данные | Varchar | Строка из цифр | Нет | Нет | Да |
| Пол | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Номер телефона | Varchar | Строка из букв кириллицы | Нет | Нет | Да |

**Место регистрации**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID места регистрации | Integer | Целые числа, больше нуля | Да | Нет | Да |
| Федеральный округ | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Область | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Город | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Улица | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Номер дома | Integer | Целые числа, больше нуля | Нет | Нет | Да |
| Дробь | Integer | Целые числа, больше нуля | Нет | Нет | Нет |
| Квартира | Integer | Целые числа, больше нуля | Нет | Нет | Нет |

**Родственник**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID родственника | Integer | Целые числа, больше нуля | Да | Нет | Да |
| ID персональных данных | Integer | Целые числа, больше нуля | Нет | Нет | Да |

**Личное дело**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID личного дела | Integer | Целое число, больше нуля | Да | Нет | Да |
| ID медицинского заключения | Integer | Целое число, больше нуля | Нет | Нет | Да |
| ID сезона призыва | Integer | Целое число, больше нуля | Нет | Нет | Да |
| Проходит военную службу | Boolean | True, false | Нет | Нет | Да |
| Военный билет | Boolean | True, false | Нет | Нет | Нет |
| Звание | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Дата постановки на учет | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | Нет | Нет | Нет |
| Дата снятия с учета | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | Нет | Нет | Нет |
| ID персональных данных | Integer | Целое число, больше нуля | Нет | Нет | Да |
| ID военной части | Integer | Целое число, больше нуля | Нет | Нет | Да |

**Медицинское заключение**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID медицинского заключениея | Integer | Целое число, больше нуля | Да | Нет | Да |
| Статус годности | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Врач | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Диагноз | Varchar | Строка из букв кириллицы | Нет | Нет | Да |

**Место работы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID места работы | Integer | Целое число, больше нуля | Да | Нет | Да |
| Название организации | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Адрес | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Сфера | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Гос аккредитация | Boolean | True, false | Нет | Нет | Нет |
| ID персональных данных | Integer | Целое число, больше нуля | Да | Нет | Да |

**Место учебы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID места учебы | Integer | Целое число, больше нуля | Да | Нет | Да |
| Название | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Адрес | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Уровень образования | Varchar | Строка из букв кириллицы | Нет | Нет | Нет |
| Окончено | Boolean | True, false | Нет | Нет | Нет |
| ID персональных данных | Integer | Целое число, больше нуля | Да | Нет | Да |

**Военная часть**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID военной части | Integer | Целое число, больше нуля | Да | Нет | Да |
| Название военной части | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Город | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Вид войск | Varchar | Строка из букв кириллицы | Нет | Нет | Да |

**Призыв**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Домен** | **Первичный ключ** | **Альтернативный ключ** | **Обязательно к заполнению** |
| ID призыва | Integer | Целое число, больше нуля | Да | Нет | Да |
| Дата призыва | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | Нет | Нет | Да |
| Место призыва | Varchar | Строка из букв кириллицы | Нет | Нет | Да |
| Сезон призыва | Varchar | Строка из букв кириллицы | Нет | Нет | Да |

**Призывник / conscript**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID призывника | conscript \_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| ID личного дела | personal\_file\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |

**Родственник / relative**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID родственника | relatives\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| ID персональных данных | personal\_data\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |

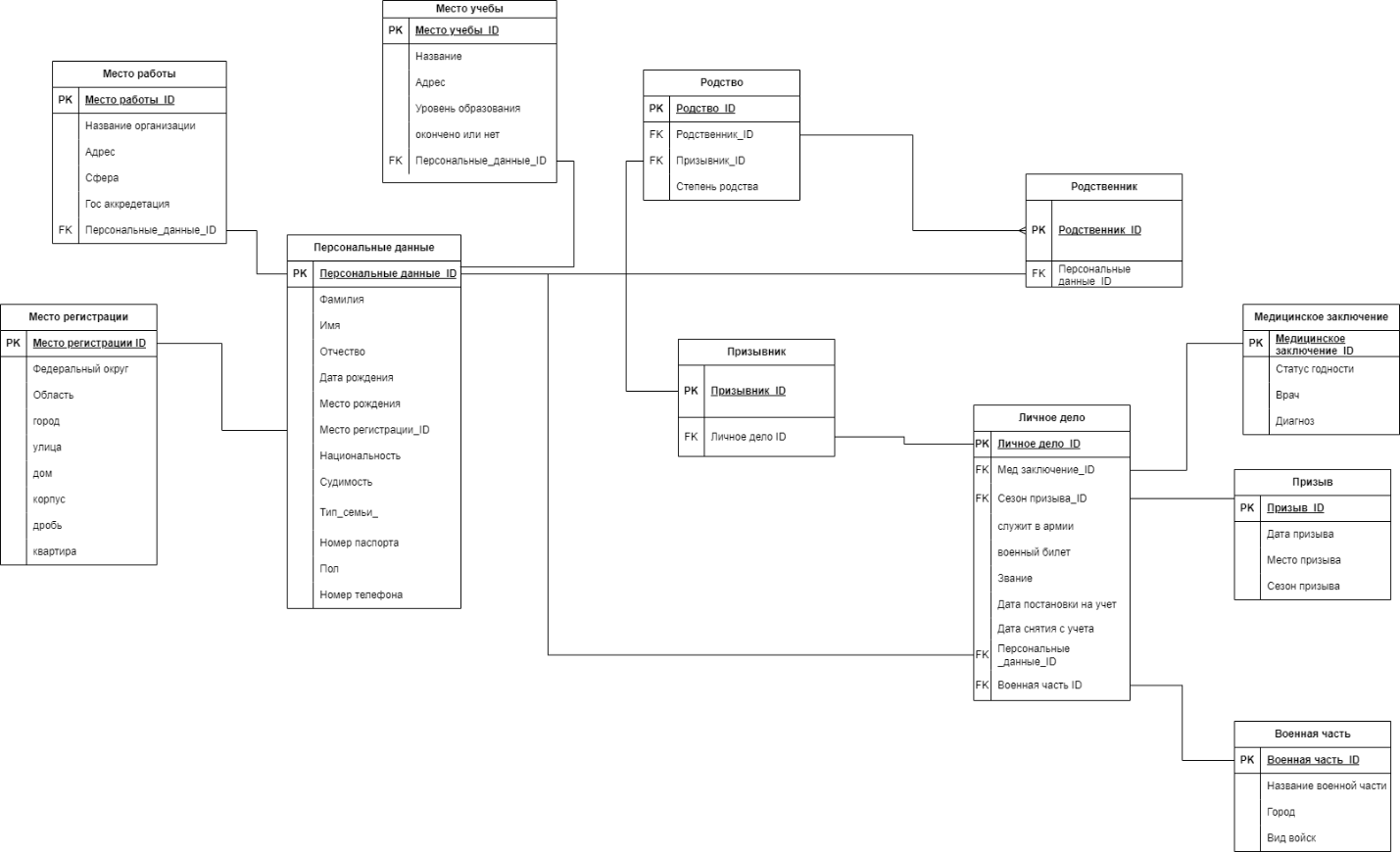


Рисунок 1.2 – ER-диаграмма

**Персональные данные / personal\_data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID персональных данных | personal\_data\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Фамилия | surname | Простой однозначный | Varchar [40] | Строка из букв кириллицы | - | Да |
| Имя | name | Простой однозначный | Varchar [15] | Строка из букв кириллицы | - | Да |
| Отчество | patronymic | Простой однозначный | Varchar [40] | Строка из букв кириллицы | - | Нет |
| Дата рождения | date\_of\_birth | Простой составной | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | - | Да |
| Место рождения | place\_of\_birth | Простой однозначный | Varchar [255] | Строка из букв кириллицы | - | Да |
| ID места регистрации | place\_of\_registration\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| Национальность | nationality | Простой однозначный | Varchar [50] | Строка из букв кириллицы | - | Да |
| Судимость | criminal\_record | Простой однозначный | Varchar [4] | Строка из букв кириллицы | - | Да |
| Тип семьи | family\_type | Простой однозначный | Varchar [50] | Целое число, больше нуля | FK | Да |
| Паспортные данные | passport\_data | Простой однозначный | Varchar [10] | Строка из букв кириллицы | - | Да |
| Пол | gender | Простой однозначный | Varchar [7] | Строка из буквы | - | Да |
| Номер телефона | phone\_number | Простой однозначный | Varchar [11] | Строка из буквы | - | Нет |

**Место регистрации / place\_of\_registration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID места регистрации | place\_of\_registration\_id | Простой однозначный | Integer | Целые числа, больше нуля | PK | Да |
| Федеральный округ | federal\_district | Простой однозначный | Varchar  [50] | Строка из букв кириллицы | - | Да |
| Область | area | Простой однозначный | Varchar  [50] | Строка из букв кириллицы | - | Да |
| Город | city | Простой однозначный | Varchar  [50] | Строка из букв кириллицы | - | Да |
| Улица | street | Простой однозначный | Varchar  [50] | Строка из букв кириллицы | - | Да |
| Номер дома | house\_number | Простой однозначный | Varchar  [5] | Целые числа, больше нуля | - | Да |
| Дробь | fraction | Простой однозначный | Integer | Целые числа, больше нуля | - | Нет |
| Квартира | apartment\_number | Простой однозначный | Integer | Целые числа, больше нуля | - | Нет |

**Личное дело / personal\_file**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID личного дела | personal\_file\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| ID мед заключения | medical\_report\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| ID сезона призыва | call\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| Проходит военную службу | military\_service | Простой однозначный | Varchar  [4] | Строка из букв кириллицы | - | Да |
| Военный билет | military\_ticket | Простой однозначный | Varchar  [20] | Строка из букв кириллицы | - | Нет |
| Звание | rank | Простой однозначный | Varchar  [25] | Строка из букв кириллицы | - | Нет |
| Дата постановки на учет | date\_of\_registration | Простой однозначный | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | - | Нет |
| Дата снятия с учета | date\_of\_de-registration | Простой однозначный | Date | Строка вида ХХ:ХХ:ХХХХ, где Х – целое число | - | Нет |
| ID персональных данных | personal\_data\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| ID военной части | military\_unit\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |

**Мед заключение / medical\_report**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID медицинского заключения | medical\_report\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Статус годности | shelf\_life\_category | Простой однозначный | Varchar  [20] | Строка из букв кириллицы | - | Да |
| Врач | doctor | Простой однозначный | Varchar [50] | Строка из букв кириллицы | - | Да |
| Диагноз | diagnosis | Простой однозначный | Varchar [50] | Строка из букв кириллицы | - | Да |

**Место работы / work\_study**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID места работы | place\_work\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Название организации | name\_organization | Простой однозначный | Varchar  [255] | Строка из букв кириллицы | - | Да |
| Адрес | address | Простой однозначный | Varchar  [255] | Строка из букв кириллицы | - | Да |
| Сфера | field\_activity | Простой однозначный | Varchar  [100] | Строка из букв кириллицы | - | Да |
| Гос аккредитация | state\_accreditation | Простой однозначный | Varchar  [20] | Строка из букв кириллицы | - | Да |
| ID персональных данных | personal\_data\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |

**Место учебы / place\_study**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID места учебы | place\_study\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Название организации | name\_organization | Простой однозначный | Varchar [255] | Строка из букв кириллицы | - | Да |
| Адрес | address | Простой однозначный | Varchar [255] | Строка из букв кириллицы | - | Да |
| Уровень образования | education\_level | Простой однозначный | Varchar  [50] | Строка из букв кириллицы | - | Да |
| Статус окончания | end\_status | Простой однозначный | Varchar  [15] | Строка из букв кириллицы | - | Да |
| ID персональных данных | personal\_data\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |

**Родство / kinship**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID родства | kinship\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| ID родственника | relative\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| ID призывника | conscript\_id | Простой однозначный | Integer | Целое число, больше нуля | FK | Да |
| Степень родства | degree\_kinship | Простой однозначный | Varchar [20] | Строка из букв кириллицы | - | Да |

**Призыв / call**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID призыва | call\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Дата призыва | draft\_date | Простой однозначный | Date | Строка из цифр, разделенных точкой | - | Да |
| Место призыва | place\_of\_call | Простой однозначный | Varchar [255] | Строка из букв кириллицы | - | Да |
| Сезон призыва | season\_name | Простой однозначный | Varchar [10] | Строка из букв кириллицы | - | Да |

**Военная часть / military\_unit**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Атрибут** | **Наименование** | **Вид** | **Тип** | **Домен** | **Ключ** | **Обязательно к заполнению** |
| ID военной части | military\_unit\_id | Простой однозначный | Integer | Целое число, больше нуля | PK | Да |
| Название военной части | name\_military\_unit | Простой однозначный | Varchar [255] | Строка из букв кириллицы | - | Да |
| Город | city | Простой однозначный | Varchar [50] | Строка из букв кириллицы | - | Да |
| Вид войск | type\_troops | Простой однозначный | Varchar [100] | Строка из букв кириллицы | - | Да |

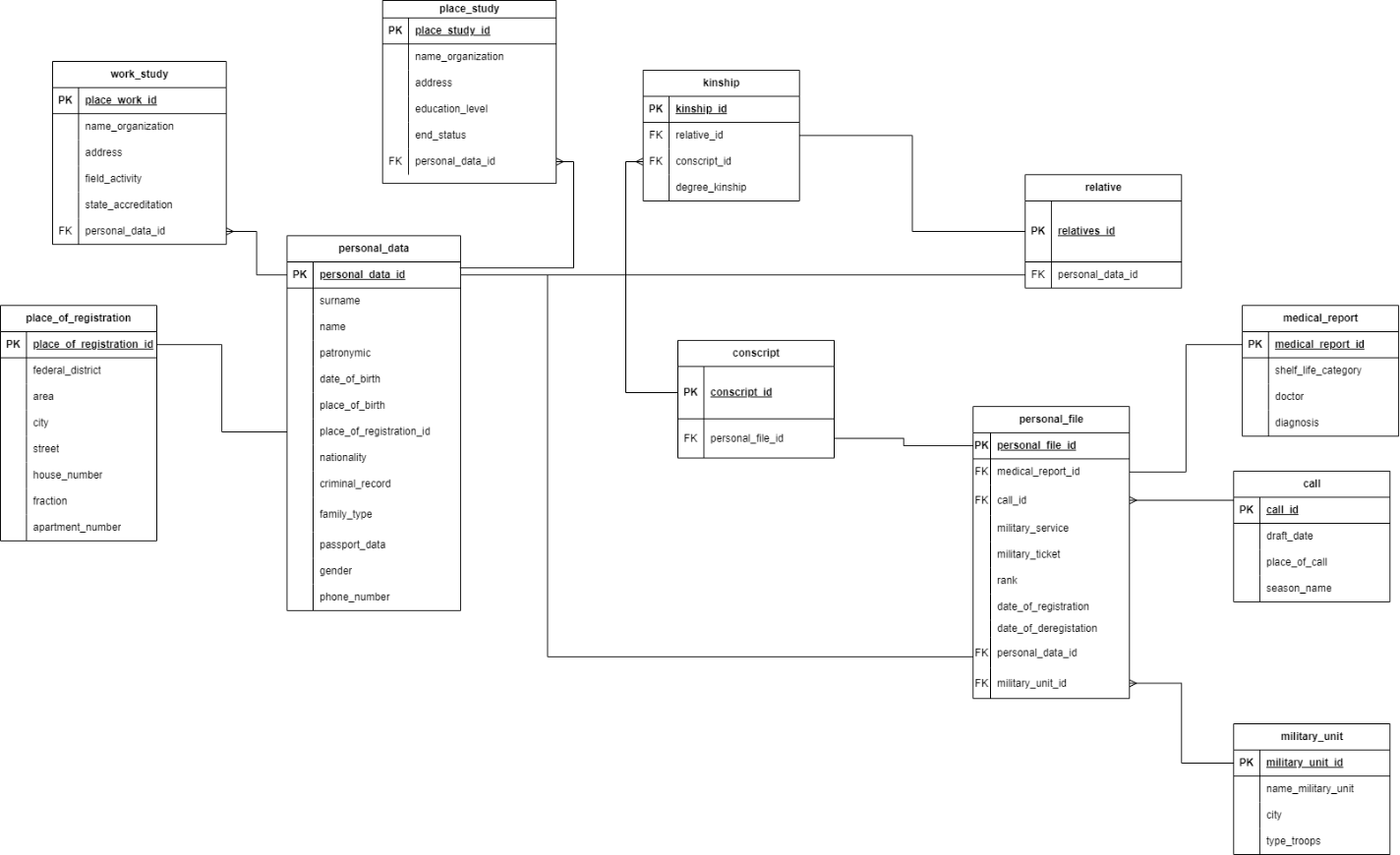


Рисунок 1.2 – ER-диаграмма после логического проектирования

***Первое правило нормализации*:** переменная отношения находится в 1НФ тогда и только тогда, когда в любом допустимом значении отношений каждый его кортеж содержит только одно значение для каждого из атрибутов.

Таблица *Место регистрации* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Место работы* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Персональные данные* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Место учебы* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Родство* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Призывник* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Личное дело* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Родственник* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Медицинское заключение* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Призыв* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Таблица *Военная часть* находится в 1 нормальной форме, так как ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

***Второе правило нормализации*:** переменная отношения находится во 2НФ тогда и только тогда, когда она находится в 1НФ и каждый не ключевой атрибут функционально полно зависит от ее потенциального ключа.

Таблица *Место регистрации* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Место работы* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Место учебы* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Персональные данные* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Медицинское заключение* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Родство* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Призывник* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Личное дело* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Родственник* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Призыв* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

Таблица *Военная часть* находится во 2 НФ, так как она имеет простой первичный ключ и все атрибуты зависят от этого ключа.

***Третье правило нормализации*:** переменная отношения находится в 3НФ тогда и только тогда, когда она находится во 2НФ и отсутствуют транзитивные зависимости не ключевых атрибутов от ключевых.

В связи с вариантом предметной области не таблицы удалось нормализовать до третьей формы, иначе это навредило бы работе базы данных.

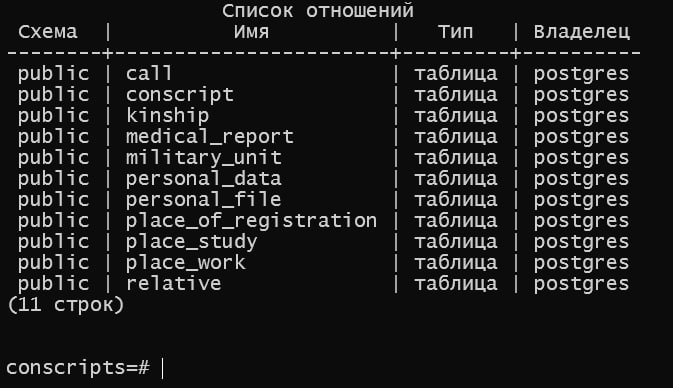
1. **Проверка соответствия отношений требованиям пользовательских транзакций.**

Все отношения соответствуют требованиям пользовательских транзакций.

Определение требований поддержки бизнес правил и целостности данных.

**Часть 2**

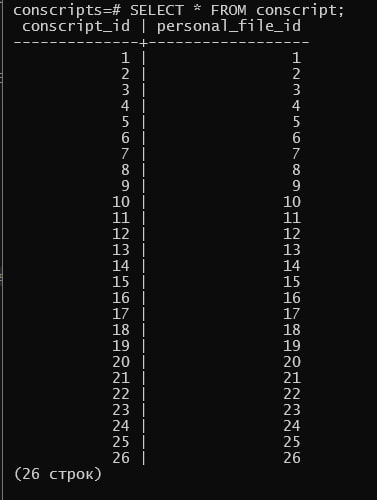
2.1 Создание таблиц:

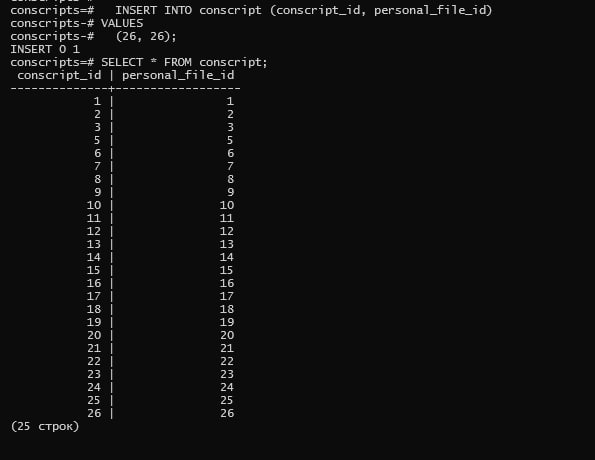


**2.2 Триггеры:**

1. Если прошло условно n лет с момента снятия с учета то, при следующем добавлении, удаляется данные об этом призывнике

CREATE OR REPLACE FUNCTION delete\_conscript\_trigger() RETURNS TRIGGER AS $$ DECLARE deregistration\_date DATE; BEGIN SELECT date\_of\_deregistration INTO deregistration\_date FROM personal\_file JOIN conscript ON personal\_file.personal\_file\_id = conscript.personal\_file\_id; IF (deregistration\_date IS NOT NULL AND (deregistration\_date + INTERVAL '5 years') <= current\_date) THEN DELETE FROM conscript WHERE conscript\_id = NEW.conscript\_id; END IF; RETURN NEW; END; $$ LANGUAGE 'plpgsql'; CREATE TRIGGER delete\_conscript BEFORE INSERT ON personal\_file FOR EACH ROW EXECUTE FUNCTION delete\_conscript\_trigger();





2. Ограничение на то, что дата постановки на учет не может быть ≧ дата призыва

CREATE OR REPLACE FUNCTION check\_registration\_date()

RETURNS TRIGGER AS $$

DECLARE

draft\_date1 DATE;

BEGIN

SELECT draft\_date INTO draft\_date1

FROM call

WHERE call\_id = NEW.personal\_file\_id;

IF NEW.date\_of\_registration >= draft\_date1 THEN

RAISE EXCEPTION 'The date of registration cannot be greater than or equal to the date of conscription';

END IF;

RETURN NEW;

END;

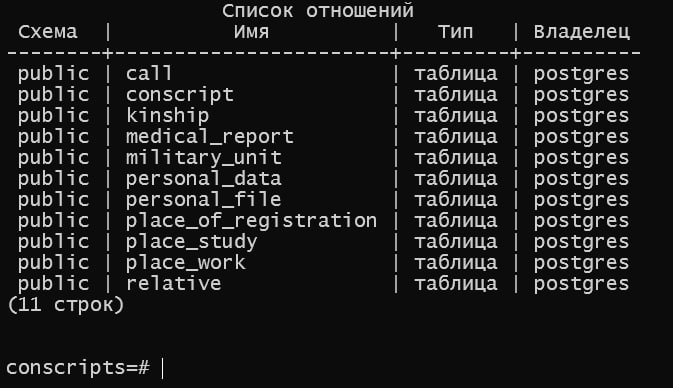
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER registration\_date\_check

BEFORE INSERT OR UPDATE ON personal\_file

FOR EACH ROW

EXECUTE FUNCTION check\_registration\_date();



3. Автоматическое заполнение поля «Тип семьи» - после ввода информации о Родственниках и их родственных связях в таблицы Родство и Родственник, триггер автоматически заполняет поле "Тип семьи" в таблице Призывник (например, "одиночка", "неполная семья", "полная семья" и т.д.).

CREATE OR REPLACE FUNCTION update\_family\_type()

RETURNS TRIGGER AS $$

BEGIN

-- Проверяем наличие матери и отца у призывника

IF EXISTS (

SELECT 1

FROM kinship k

WHERE k.conscript\_id = NEW.conscript\_id

AND k.degree\_kinship = 'Мать'

) AND EXISTS (

SELECT 1

FROM kinship k

WHERE k.conscript\_id = NEW.conscript\_id

AND k.degree\_kinship = 'Отец'

) THEN

-- Если есть и мать, и отец, устанавливаем значение "Полная"

UPDATE personal\_data

SET family\_type = 'Полная'

WHERE personal\_data\_id = NEW.conscript\_id;

ELSIF EXISTS (

SELECT 1

FROM kinship k

WHERE k.conscript\_id = NEW.conscript\_id

AND (k.degree\_kinship = 'Мать' OR k.degree\_kinship = 'Отец')

) THEN

-- Если есть только мать или только отец, устанавливаем значение "Неполная"

UPDATE personal\_data

SET family\_type = 'Не полная'

WHERE personal\_data\_id = NEW.conscript\_id;

ELSE

-- Если нет ни матери, ни отца, устанавливаем значение "Одиночка"

UPDATE personal\_data

SET family\_type = 'Одиночка'

WHERE personal\_data\_id = NEW.conscript\_id;

END IF;

RETURN NEW;

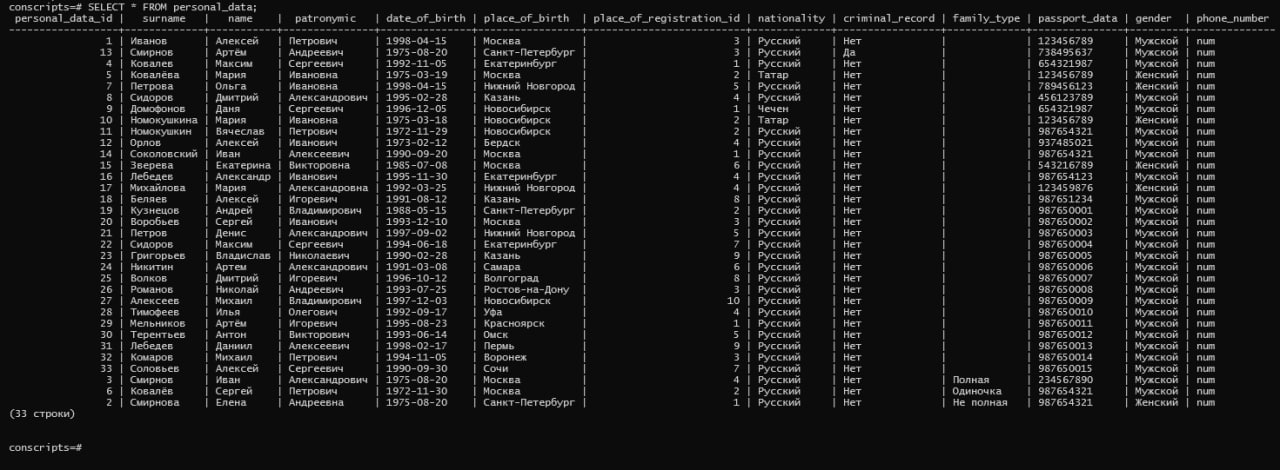
END;

$$ LANGUAGE 'plpgsql';

CREATE TRIGGER update\_family\_type\_trigger

AFTER INSERT OR UPDATE ON kinship

FOR EACH ROW

EXECUTE FUNCTION update\_family\_type();

4. «Проверка даты призыва» - перед введением информации о Призыве, триггер проверяет, что указанная дата призыва находится в допустимых пределах (например, не раньше 18-летнего возраста призывника и не позднее законной возрастной границы).

CREATE OR REPLACE FUNCTION check\_draft\_date()

RETURNS TRIGGER AS $$

BEGIN

-- Получение даты рождения призывника

DECLARE

birth\_date DATE;

min\_draft\_date DATE;

max\_draft\_date DATE;

BEGIN

SELECT date\_of\_birth INTO birth\_date

FROM personal\_data

WHERE personal\_data\_id = NEW.call\_id;

-- Вычисление минимальной и максимальной даты призыва на основе даты рождения

min\_draft\_date := birth\_date + INTERVAL '18 years';

max\_draft\_date := birth\_date + INTERVAL '27 years';

IF NEW.draft\_date < min\_draft\_date OR NEW.draft\_date > max\_draft\_date THEN

RAISE EXCEPTION 'Invalid draft date';

END IF;

RETURN NEW;

END;

END;

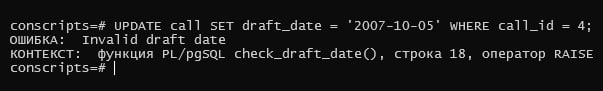
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER draft\_date\_check

BEFORE INSERT OR UPDATE ON call

FOR EACH ROW

EXECUTE FUNCTION check\_draft\_date();



5. Обновление статуса годности к военной службе" - при изменении информации в таблице Медицинское заключение,

триггер автоматически обновляет статус годности к военной службе в таблице Призывник на основе новой информации (например, "годен", "ограниченно годен", "негоден").

CREATE OR REPLACE FUNCTION update\_shelf\_life\_category()

RETURNS TRIGGER AS $$

BEGIN

CASE NEW.diagnosis

WHEN 'Здоров' THEN

NEW.shelf\_life\_category := 'Годен';

WHEN 'Психическое расстройство' THEN

NEW.shelf\_life\_category := 'Негоден';

WHEN 'Астма' THEN

NEW.shelf\_life\_category := 'Ограниченно годен';

ELSE

NULL;

END CASE;

RETURN NEW;

END;

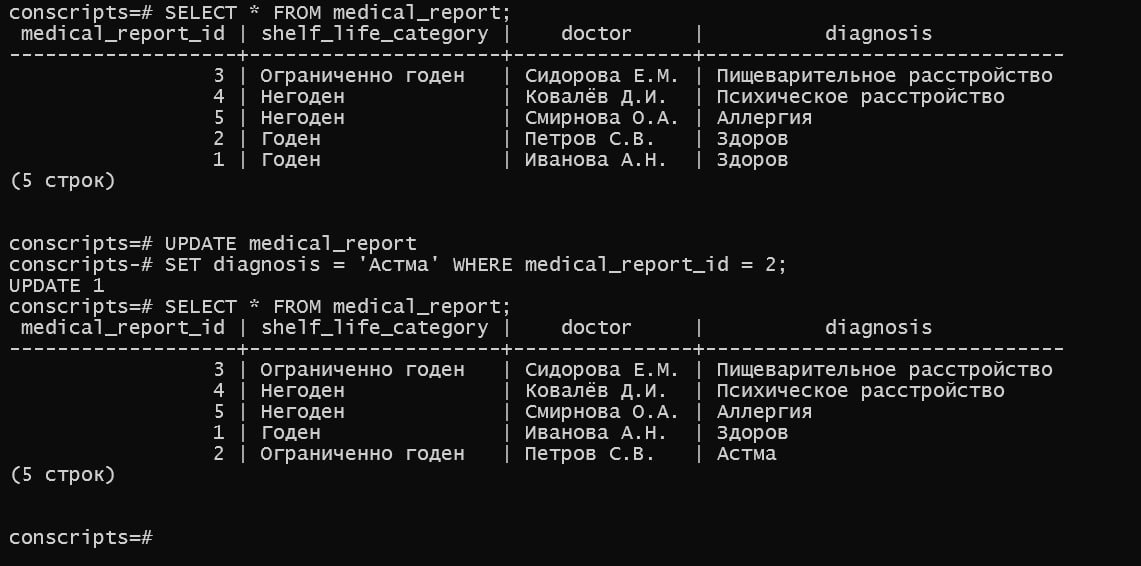
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER update\_shelf\_life\_category\_trigger

BEFORE UPDATE ON medical\_report

FOR EACH ROW

EXECUTE FUNCTION update\_shelf\_life\_category();



**Функции:**

* 1. Установить порядок призыва призывников в зависимости от их медицинского здоровья и судимостей их родственников.

CREATE OR REPLACE FUNCTION sort\_conscripts\_by\_health\_and\_criminal()

RETURNS TABLE(surname VARCHAR(255), name VARCHAR(255), patronymic VARCHAR(255), conscript\_id INT, health\_status VARCHAR(255), relative\_criminal\_record BOOLEAN)

AS $$

BEGIN

RETURN QUERY

SELECT pd.surname, pd.name, pd.patronymic, c.conscript\_id, mrd.shelf\_life\_category AS health\_status,

CASE WHEN pd\_rel.criminal\_record = 'Да' THEN TRUE ELSE FALSE END AS relative\_criminal\_record

FROM conscript c

JOIN personal\_file pf ON c.personal\_file\_id = pf.personal\_file\_id

JOIN medical\_report mrd ON pf.medical\_report\_id = mrd.medical\_report\_id

LEFT JOIN personal\_data pd ON pd.personal\_data\_id = pf.personal\_data\_id

LEFT JOIN kinship k ON k.conscript\_id = c.conscript\_id

LEFT JOIN relative r ON r.relative\_id = k.relative\_id

LEFT JOIN personal\_data pd\_rel ON pd\_rel.personal\_data\_id = r.personal\_data\_id

ORDER BY CASE

WHEN mrd.shelf\_life\_category = 'Годен' THEN 1

WHEN mrd.shelf\_life\_category = 'Ограниченно годен' THEN 2

WHEN mrd.shelf\_life\_category = 'Негоден' THEN 3

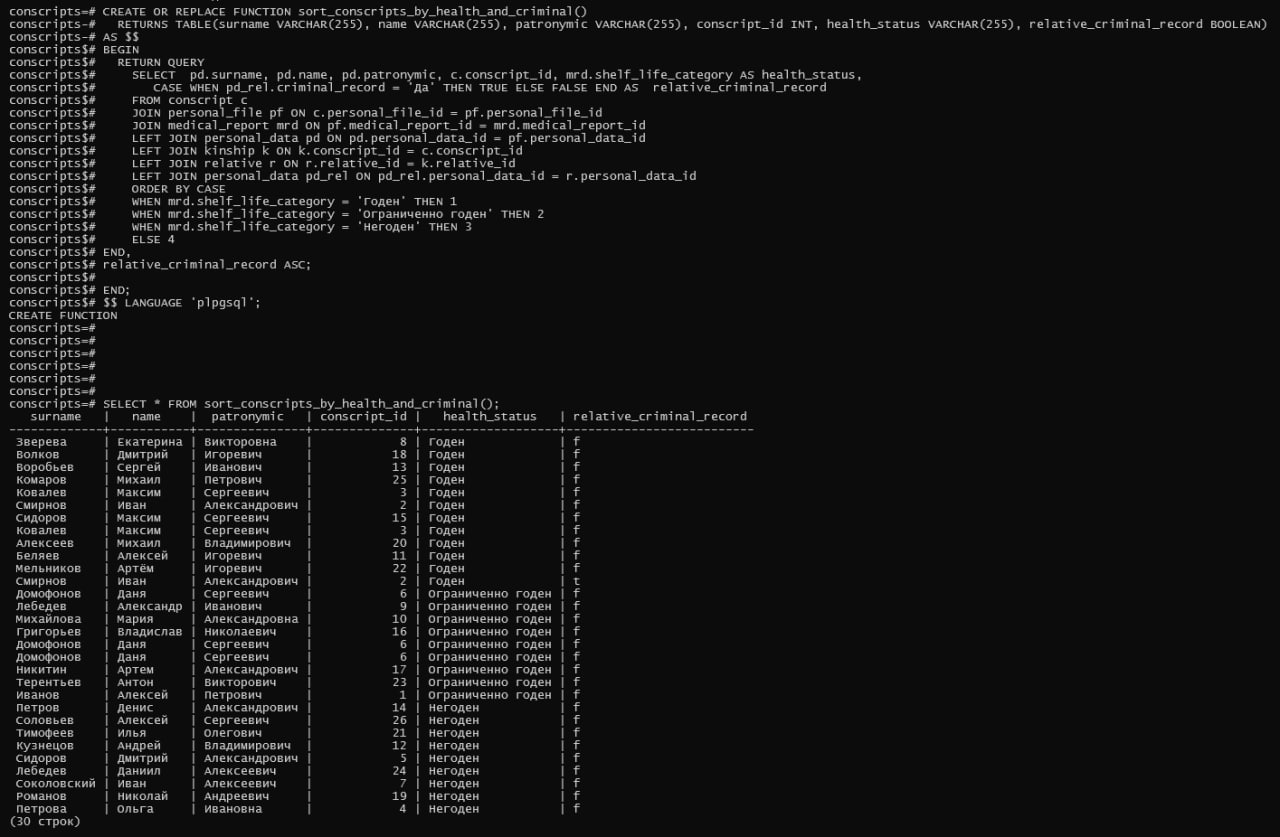
ELSE 4

END,

relative\_criminal\_record ASC;

END;

$$ LANGUAGE 'plpgsql';



* 1. Рассчитайть процент полностью здоровых призывников для каждого региона.

CREATE OR REPLACE FUNCTION calculate\_healthy\_percentage()

RETURNS TABLE(region VARCHAR(255), healthy\_percentage NUMERIC)

AS $$

BEGIN

RETURN QUERY

SELECT pr.area AS region,

CAST((CAST(COUNT(\*) FILTER (WHERE mrd.shelf\_life\_category = 'Годен') AS NUMERIC) / CAST(COUNT(\*) AS NUMERIC)) \* 100 AS NUMERIC(10, 2)) AS healthy\_percentage

FROM place\_of\_registration pr

JOIN personal\_data pd ON pd.place\_of\_registration\_id = pr.place\_of\_registration\_id

JOIN personal\_file pf ON pd.personal\_data\_id = pf.personal\_data\_id

JOIN medical\_report mrd ON pf.medical\_report\_id = mrd.medical\_report\_id

JOIN conscript c ON pf.personal\_file\_id = c.personal\_file\_id

JOIN military\_unit mu ON pf.military\_unit\_id = mu.military\_unit\_id

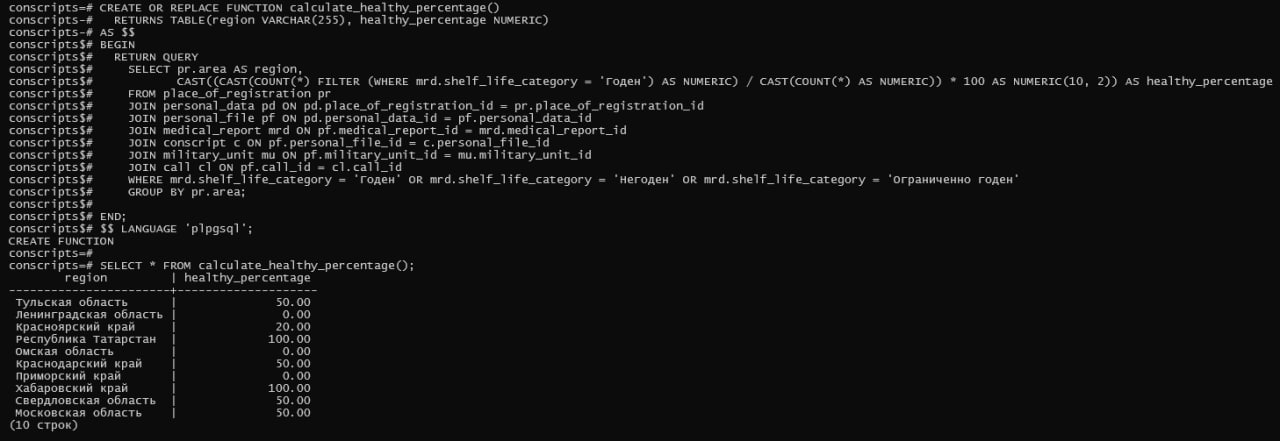
JOIN call cl ON pf.call\_id = cl.call\_id

WHERE mrd.shelf\_life\_category = 'Годен' OR mrd.shelf\_life\_category = 'Негоден' OR mrd.shelf\_life\_category = 'Ограниченно годен'

GROUP BY pr.area;

END;

$$ LANGUAGE 'plpgsql';



3. Установить порядок призыва призывников в зависимости от их медицинского здоровья и судимостей их родственников.

CREATE OR REPLACE FUNCTION sort\_conscripts\_by\_health\_and\_criminal()

RETURNS TABLE(surname VARCHAR(255), name VARCHAR(255), patronymic VARCHAR(255), conscript\_id INT, health\_status VARCHAR(255), relative\_criminal\_record BOOLEAN)

AS $$

BEGIN

RETURN QUERY

SELECT pd.surname, pd.name, pd.patronymic, c.conscript\_id, mrd.shelf\_life\_category AS health\_status,

CASE WHEN pd\_rel.criminal\_record = 'Да' THEN TRUE ELSE FALSE END AS relative\_criminal\_record

FROM conscript c

JOIN personal\_file pf ON c.personal\_file\_id = pf.personal\_file\_id

JOIN medical\_report mrd ON pf.medical\_report\_id = mrd.medical\_report\_id

LEFT JOIN personal\_data pd ON pd.personal\_data\_id = pf.personal\_data\_id

LEFT JOIN kinship k ON k.conscript\_id = c.conscript\_id

LEFT JOIN relative r ON r.relative\_id = k.relative\_id

LEFT JOIN personal\_data pd\_rel ON pd\_rel.personal\_data\_id = r.personal\_data\_id

ORDER BY CASE

WHEN mrd.shelf\_life\_category = 'Годен' THEN 1

WHEN mrd.shelf\_life\_category = 'Ограниченно годен' THEN 2

WHEN mrd.shelf\_life\_category = 'Негоден' THEN 3

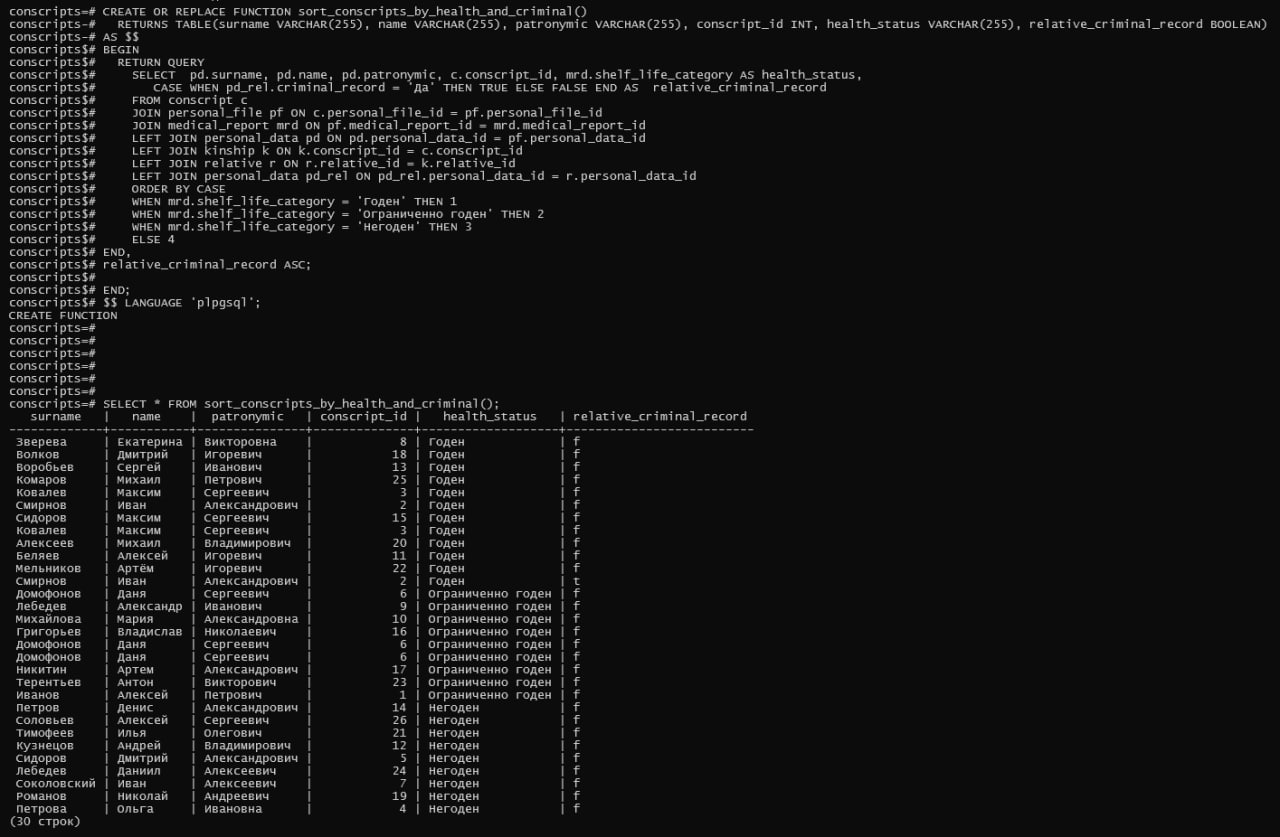
ELSE 4

END,

relative\_criminal\_record ASC;

END;

$$ LANGUAGE 'plpgsql';



1. Сформируйте медицинское заключение для хранения информации о состоянии здоровья призывника и принятия решения о годности его к военной службе. (пояснение: т.е. медицинское заключение должно формироваться автоматически, на основании других таблиц)

CREATE TABLE medical\_report\_history (

report\_id SERIAL PRIMARY KEY,

conscript\_id\_sel INT,

surname VARCHAR(255),

name VARCHAR(255),

patronymic VARCHAR(255),

health\_status VARCHAR(255),

diagnosis\_status VARCHAR(255),

doctor\_name VARCHAR(255)

);

CREATE OR REPLACE FUNCTION generate\_medical\_report(conscript\_id\_sel INT)

RETURNS VOID

AS $$

DECLARE

medical\_info RECORD;

BEGIN

-- Получение информации о призывнике

SELECT

pd.surname,

pd.name,

pd.patronymic,

mrd.shelf\_life\_category AS health\_status,

mrd.diagnosis AS diagnosis\_status,

mrd.doctor AS doctor\_name

INTO

medical\_info

FROM

conscript c

JOIN personal\_file pf ON c.personal\_file\_id = pf.personal\_file\_id

JOIN personal\_data pd ON pd.personal\_data\_id = pf.personal\_data\_id

JOIN medical\_report mrd ON pf.medical\_report\_id = mrd.medical\_report\_id

WHERE

c.conscript\_id = conscript\_id\_sel;

-- Добавление записи в отдельную таблицу с медицинскими заключениями

INSERT INTO medical\_report\_history (conscript\_id\_sel, surname, name, patronymic, health\_status, diagnosis\_status, doctor\_name)

VALUES (conscript\_id\_sel, medical\_info.surname, medical\_info.name, medical\_info.patronymic, medical\_info.health\_status, medical\_info.diagnosis\_status, medical\_info.doctor\_name);

-- Дополнительные действия по формированию медицинского заключения

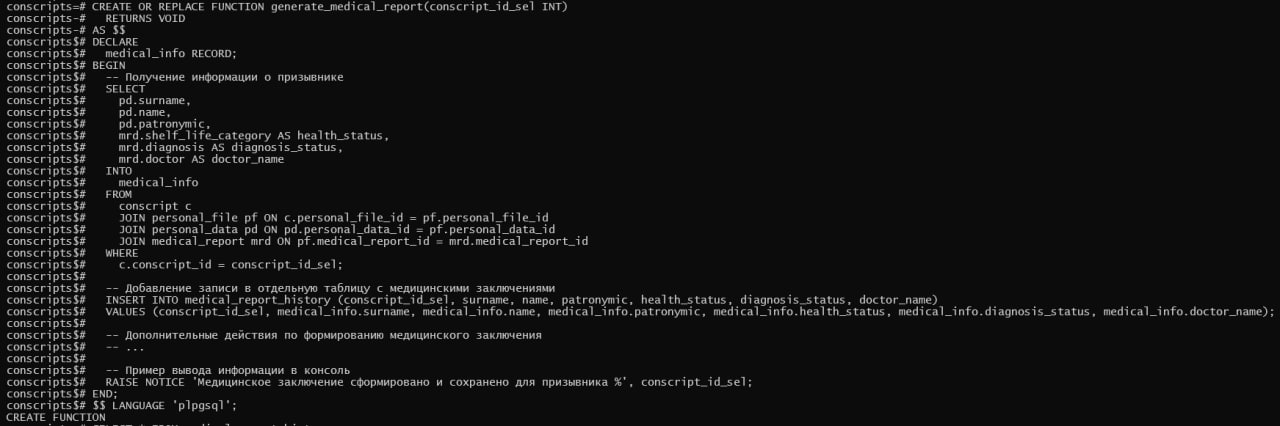
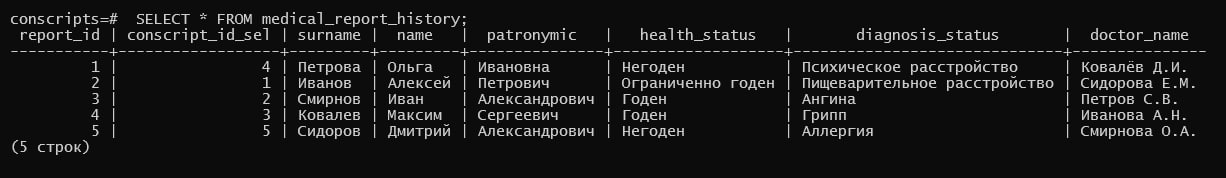
-- ...

-- Пример вывода информации в консоль

RAISE NOTICE 'Медицинское заключение сформировано и сохранено для призывника %', conscript\_id\_sel;

END;

$$ LANGUAGE 'plpgsql';



1. Рассчитать скользящее значение полностью прошедших военную службу призывников в мотострелковых войсках к 2021 году.

CREATE OR REPLACE FUNCTION calculate\_service\_years()

RETURNS TABLE (year INT, average\_service\_years BIGINT) AS $$

DECLARE

start\_year INT := 2017;

end\_year INT := 2021;

current\_year INT;

BEGIN

FOR current\_year IN start\_year..end\_year LOOP

RETURN QUERY

SELECT current\_year, COUNT(EXTRACT(YEAR FROM age(pf.date\_of\_registration, pf.date\_of\_deregistration)))

FROM personal\_file pf

JOIN military\_unit mu ON mu.military\_unit\_id = pf.military\_unit\_id

WHERE mu.type\_troops = 'Мотострелковые войска'

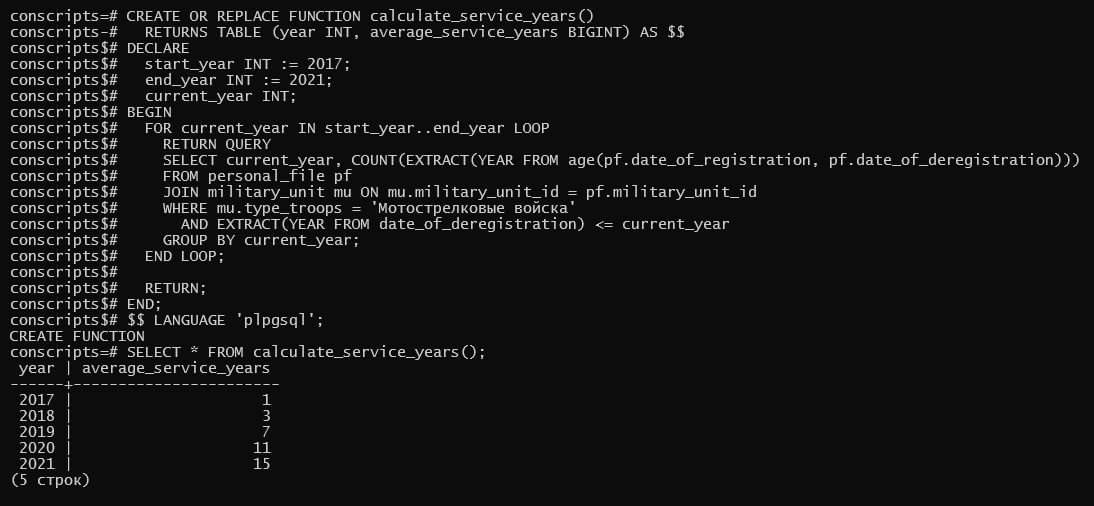
AND EXTRACT(YEAR FROM date\_of\_deregistration) <= current\_year

GROUP BY current\_year;

END LOOP;

RETURN;

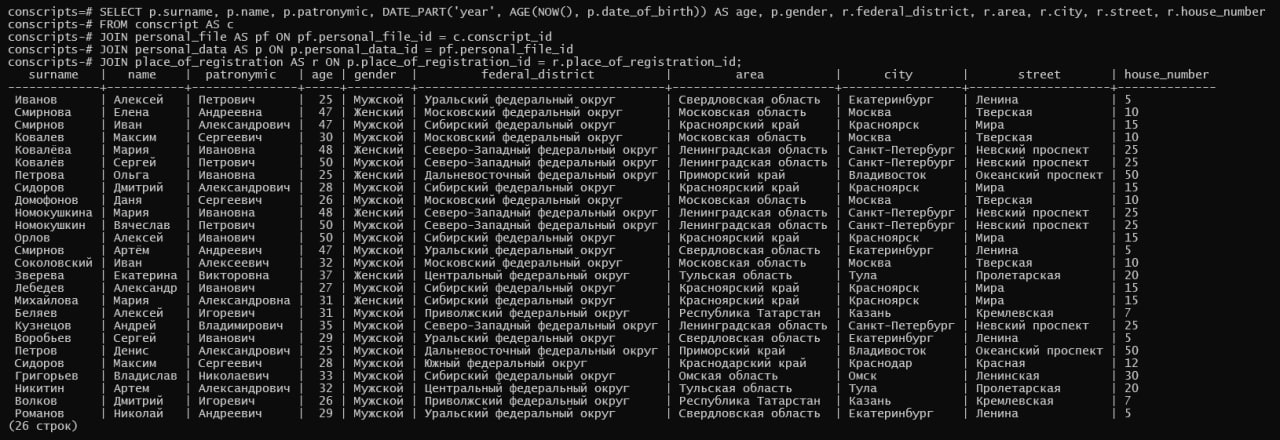
END;

$$ LANGUAGE 'plpgsql';

**Запросы:**

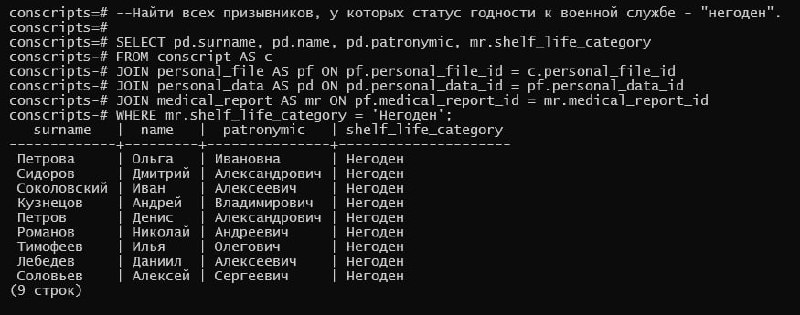
* 1. Получить список всех призывников с указанием их ФИО, возраста, пола и адреса.

SELECT p.surname, p.name, p.patronymic, DATE\_PART('year', AGE(NOW(), p.date\_of\_birth)) AS age, p.gender, r.federal\_district, r.area, r.city, r.street, r.house\_number FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.conscript\_id JOIN personal\_data AS p ON p.personal\_data\_id = pf.personal\_file\_id JOIN place\_of\_registration AS r ON p.place\_of\_registration\_id = r.place\_of\_registration\_id;



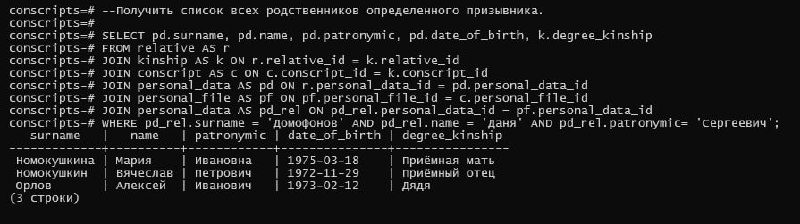
* 1. Найти всех призывников, у которых статус годности к военной службе - "негоден".

SELECT pd.surname, pd.name, pd.patronymic, mr.shelf\_life\_category FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN medical\_report AS mr ON pf.medical\_report\_id = mr.medical\_report\_id WHERE mr.shelf\_life\_category = 'Негоден';

****

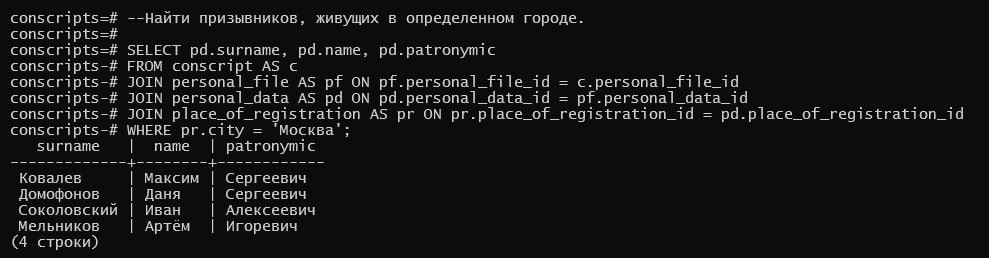
* 1. Получить список всех родственников определенного призывника.

SELECT pd.surname, pd.name, pd.patronymic, pd.date\_of\_birth, k.degree\_kinship FROM relative AS r JOIN kinship AS k ON r.relative\_id = k.relative\_id JOIN conscript AS c ON c.conscript\_id = k.conscript\_id JOIN personal\_data AS pd ON r.personal\_data\_id = pd.personal\_data\_id JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd\_rel ON pd\_rel.personal\_data\_id = pf.personal\_data\_id WHERE pd\_rel.surname = 'Домофонов' AND pd\_rel.name = 'Даня' AND pd\_rel.patronymic= 'Сергеевич';

****

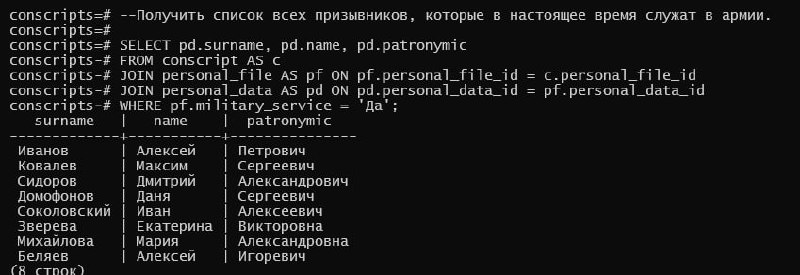
1. Найти призывников, живущих в определенном городе.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN place\_of\_registration AS pr ON pr.place\_of\_registration\_id = pd.place\_of\_registration\_id WHERE pr.city = 'Москва';

****

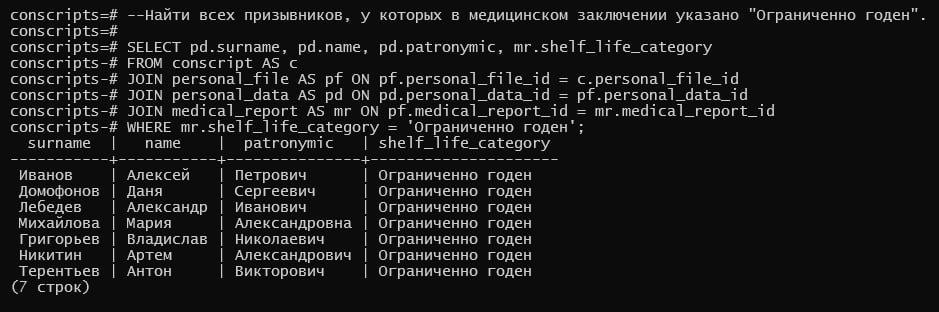
1. Получить список всех призывников, которые в настоящее время служат в армии.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id WHERE pf.military\_service = 'Да';

****

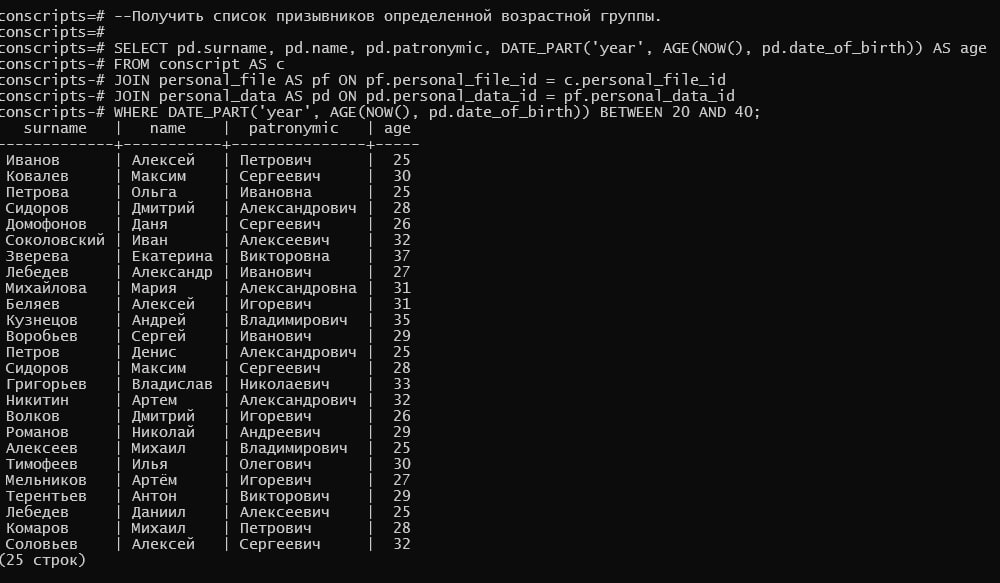
1. Найти всех призывников, у которых в медицинском заключении указано "Ограниченно годен".

SELECT pd.surname, pd.name, pd.patronymic, mr.shelf\_life\_category FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN medical\_report AS mr ON pf.medical\_report\_id = mr.medical\_report\_id WHERE mr.shelf\_life\_category = 'Ограниченно годен';

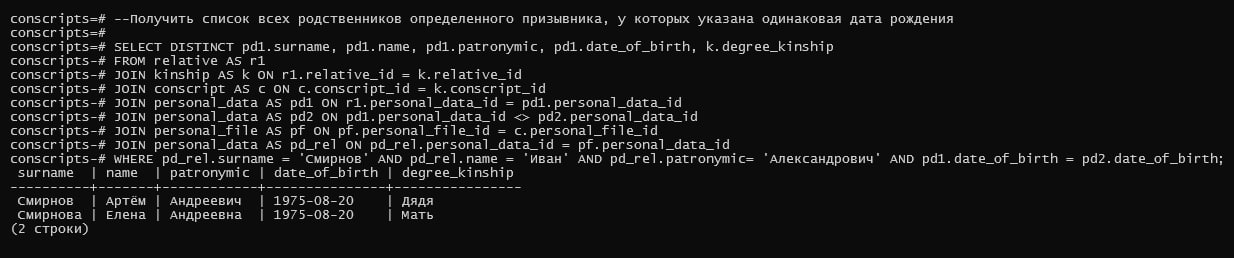


1. Получить список призывников определенной возрастной группы.

SELECT pd.surname, pd.name, pd.patronymic, DATE\_PART('year', AGE(NOW(), pd.date\_of\_birth)) AS age FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id WHERE DATE\_PART('year', AGE(NOW(), pd.date\_of\_birth)) BETWEEN 20 AND 40;

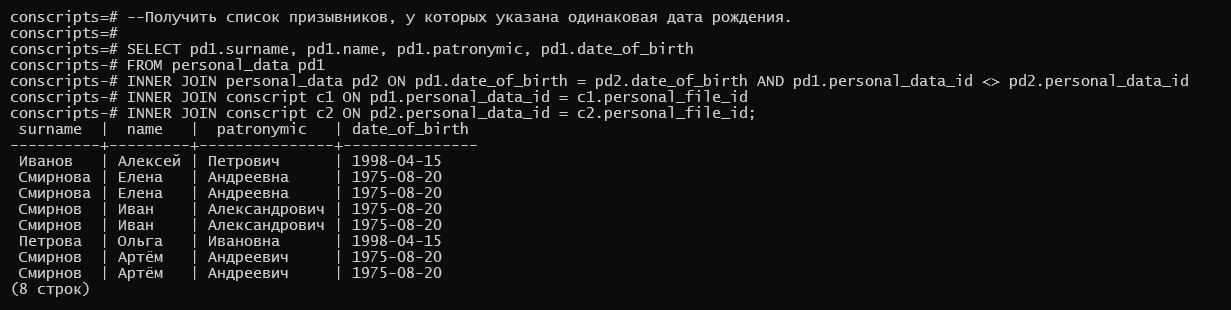


1. Найти всех родственников определенного призывника, которые являются его родителями.

SELECT pd.surname, pd.name, pd.patronymic, k.degree\_kinship FROM relative AS r JOIN kinship AS k ON r.relative\_id = k.relative\_id JOIN conscript AS c ON c.conscript\_id = k.conscript\_id JOIN personal\_data AS pd ON r.personal\_data\_id = pd.personal\_data\_id JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd\_rel ON pd\_rel.personal\_data\_id = pf.personal\_data\_id WHERE pd\_rel.surname = 'Смирнов' AND pd\_rel.name = 'Иван' AND pd\_rel.patronymic= 'Александрович' AND (k.degree\_kinship = 'Мать' OR k.degree\_kinship = 'Отец');

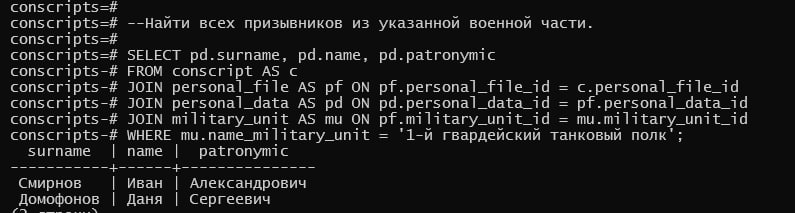
1. Получить список призывников, у которых указана одинаковая дата рождения.

SELECT pd1.surname, pd1.name, pd1.patronymic, pd1.date\_of\_birth FROM personal\_data pd1 INNER JOIN personal\_data pd2 ON pd1.date\_of\_birth = pd2.date\_of\_birth AND pd1.personal\_data\_id <> pd2.personal\_data\_id INNER JOIN conscript c1 ON pd1.personal\_data\_id = c1.personal\_file\_id INNER JOIN conscript c2 ON pd2.personal\_data\_id = c2.personal\_file\_id;

****

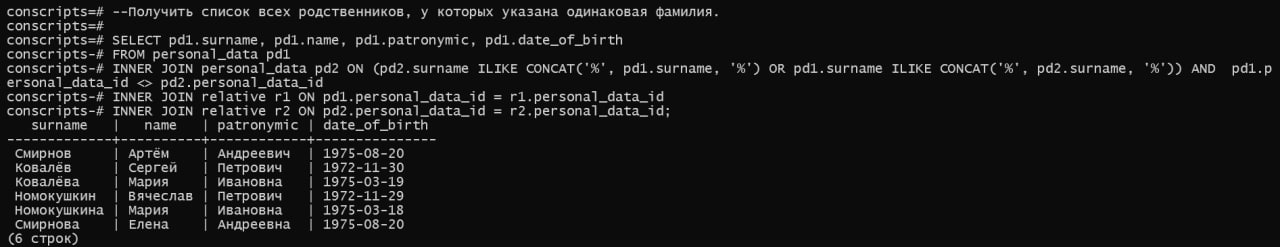
1. Найти всех призывников из указанной военной части.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN military\_unit AS mu ON pf.military\_unit\_id = mu.military\_unit\_id WHERE mu.name\_military\_unit = '1-й гвардейский танковый полк';

****

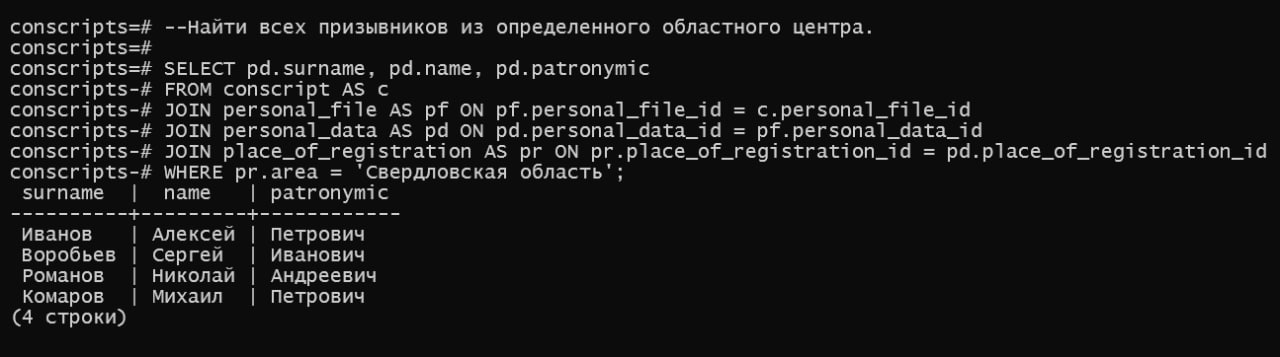
1. Получить список всех родственников, у которых указана одинаковая фамилия.

SELECT pd1.surname, pd1.name, pd1.patronymic, pd1.date\_of\_birth FROM personal\_data pd1 INNER JOIN personal\_data pd2 ON (pd2.surname ILIKE CONCAT('%', pd1.surname, '%') OR pd1.surname ILIKE CONCAT('%', pd2.surname, '%')) AND pd1.personal\_data\_id <> pd2.personal\_data\_id INNER JOIN relative r1 ON pd1.personal\_data\_id = r1.personal\_data\_id INNER JOIN relative r2 ON pd2.personal\_data\_id = r2.personal\_data\_id;



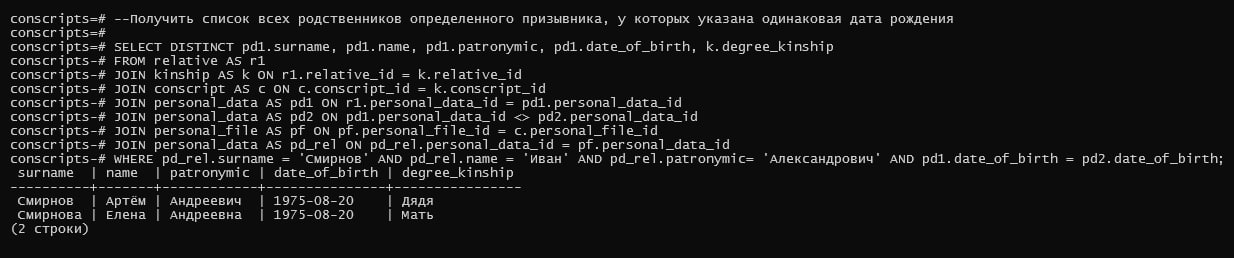
1. Найти всех призывников из определенного областного центра.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN place\_of\_registration AS pr ON pr.place\_of\_registration\_id = pd.place\_of\_registration\_id WHERE pr.area = 'Свердловская область';

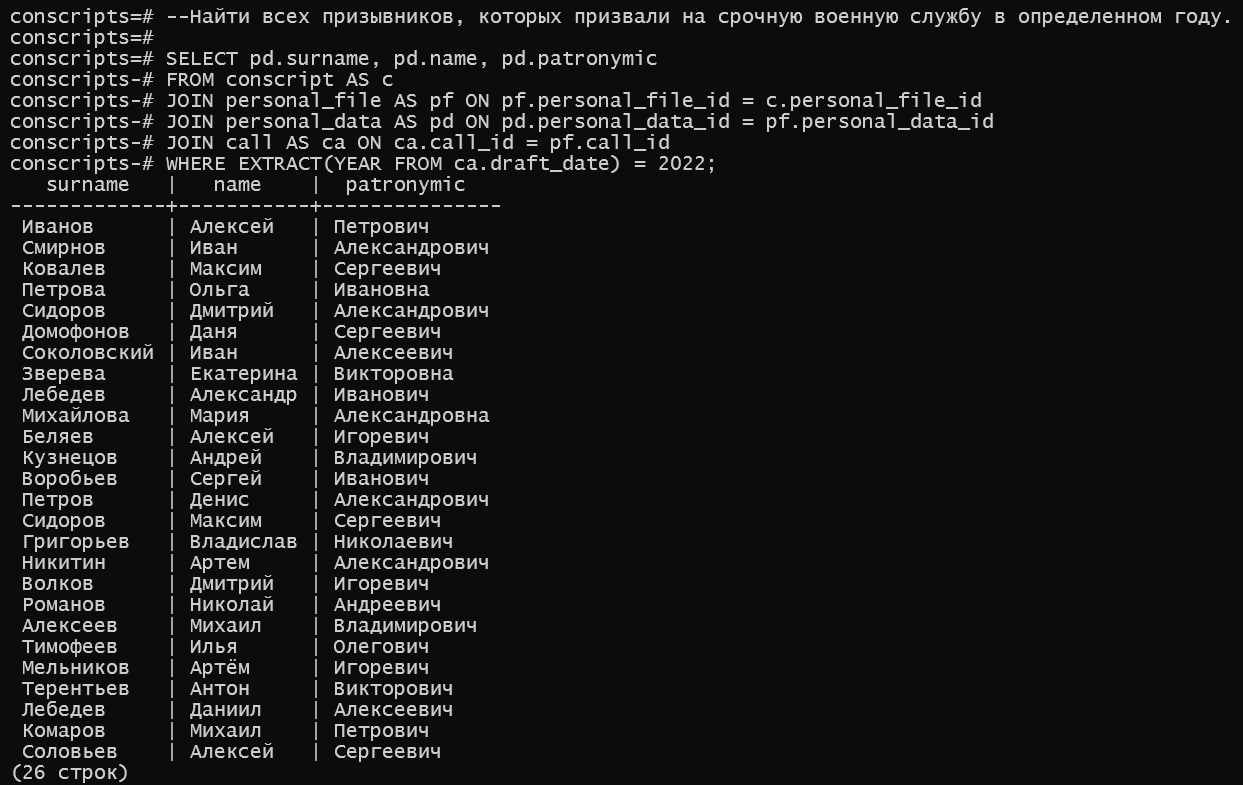


1. Получить список всех родственников определенного призывника, у которых указана одинаковая дата рождения

SELECT DISTINCT pd1.surname, pd1.name, pd1.patronymic, pd1.date\_of\_birth, k.degree\_kinship FROM relative AS r1 JOIN kinship AS k ON r1.relative\_id = k.relative\_id JOIN conscript AS c ON c.conscript\_id = k.conscript\_id JOIN personal\_data AS pd1 ON r1.personal\_data\_id = pd1.personal\_data\_id JOIN personal\_data AS pd2 ON pd1.personal\_data\_id <> pd2.personal\_data\_id JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd\_rel ON pd\_rel.personal\_data\_id = pf.personal\_data\_id WHERE pd\_rel.surname = 'Смирнов' AND pd\_rel.name = 'Иван' AND pd\_rel.patronymic= 'Александрович' AND pd1.date\_of\_birth = pd2.date\_of\_birth;

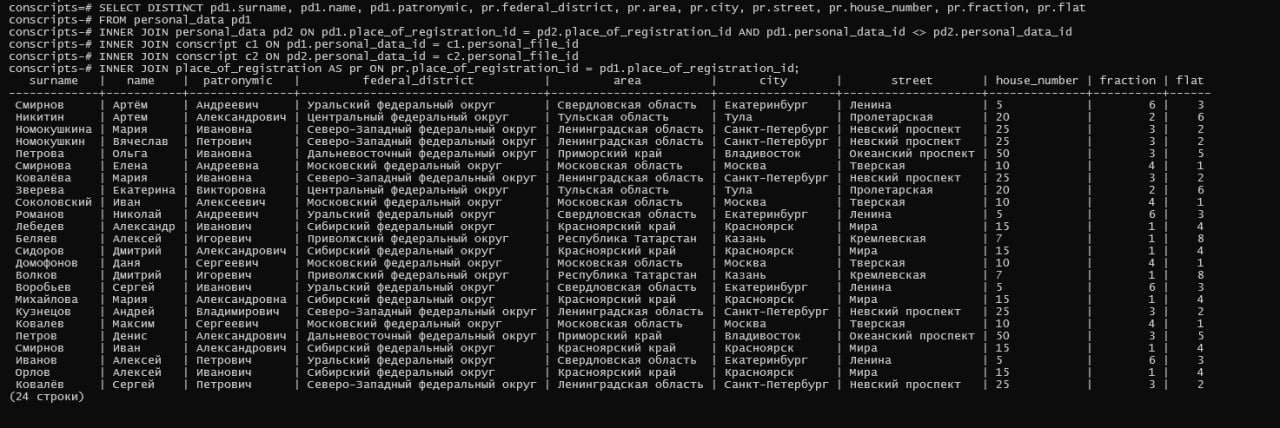
1. Найти всех призывников, которых призвали на срочную военную службу в определенном году.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN call AS ca ON ca.call\_id = pf.call\_id WHERE EXTRACT(YEAR FROM ca.draft\_date) = 2022;



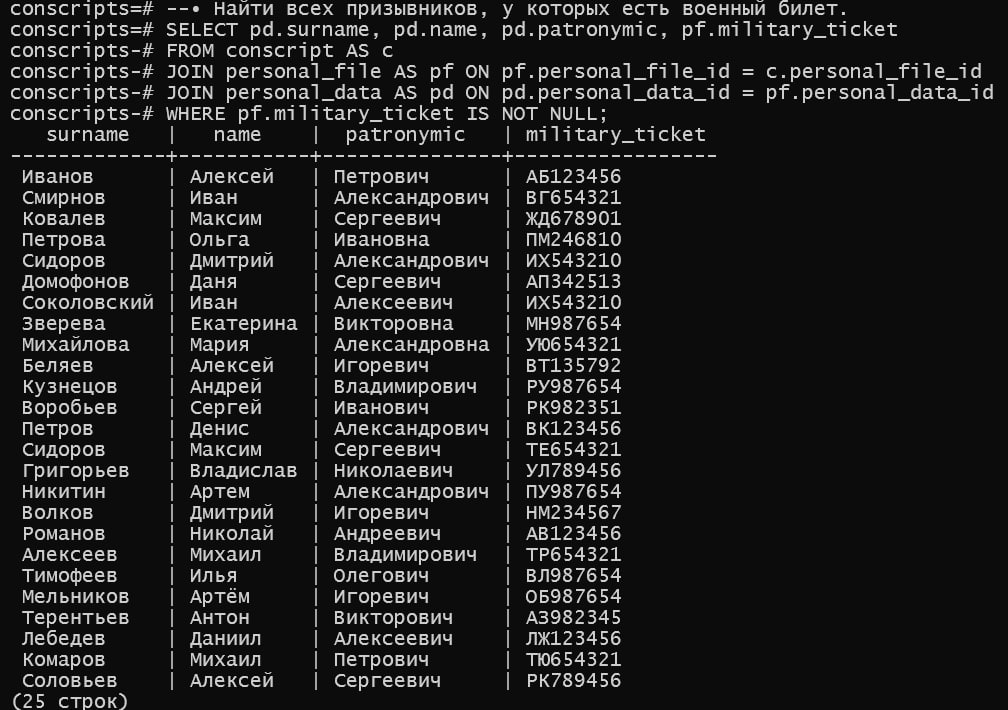
1. Получить список всех родственников, у которых указан одинаковый адрес проживания.

SELECT DISTINCT pd1.surname, pd1.name, pd1.patronymic, pr.federal\_district, pr.area, pr.city, pr.street, pr.house\_number, pr.fraction, pr.flat FROM personal\_data pd1 INNER JOIN personal\_data pd2 ON pd1.place\_of\_registration\_id = pd2.place\_of\_registration\_id AND pd1.personal\_data\_id <> pd2.personal\_data\_id INNER JOIN conscript c1 ON pd1.personal\_data\_id = c1.personal\_file\_id INNER JOIN conscript c2 ON pd2.personal\_data\_id = c2.personal\_file\_id INNER JOIN place\_of\_registration AS pr ON pr.place\_of\_registration\_id = pd1.place\_of\_registration\_id;



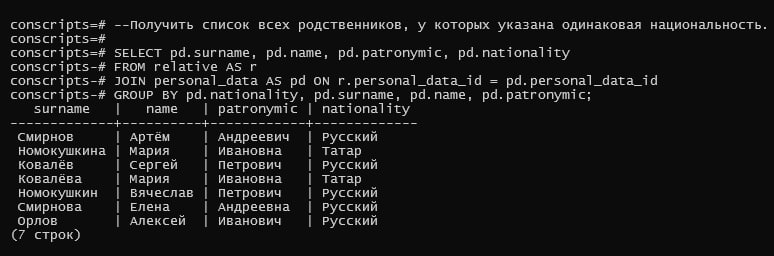
1. Найти всех призывников, у которых есть военный билет.

SELECT pd.surname, pd.name, pd.patronymic, pf.military\_ticket FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id WHERE pf.military\_ticket IS NOT NULL;



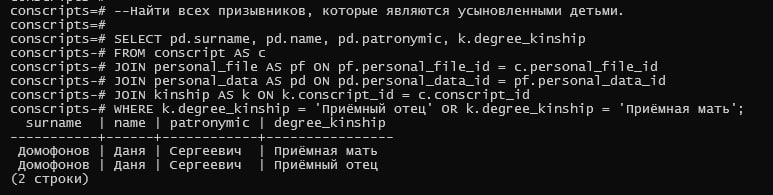
1. Получить список всех родственников, у которых указана одинаковая национальность.

SELECT pd.surname, pd.name, pd.patronymic, pd.nationality FROM relative AS r JOIN personal\_data AS pd ON r.personal\_data\_id = pd.personal\_data\_id GROUP BY pd.nationality, pd.surname, pd.name, pd.patronymic;



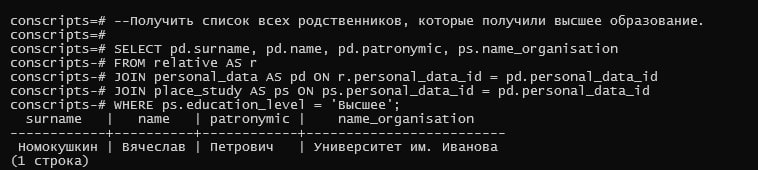
1. Найти всех призывников, которые являются усыновленными детьми.

SELECT pd.surname, pd.name, pd.patronymic, k.degree\_kinship FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN kinship AS k ON k.conscript\_id = c.conscript\_id WHERE k.degree\_kinship = 'Приёмный отец' OR k.degree\_kinship = 'Приёмная мать';



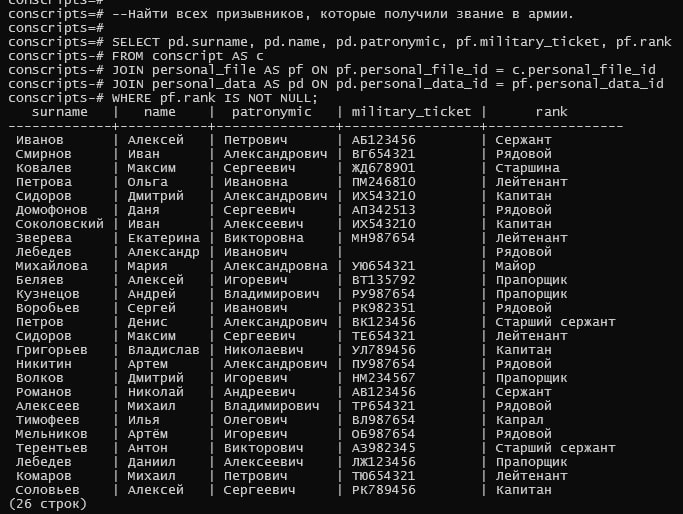
1. Получить список всех родственников, которые получили высшее образование.

SELECT pd.surname, pd.name, pd.patronymic, ps.name\_organisation FROM relative AS r JOIN personal\_data AS pd ON r.personal\_data\_id = pd.personal\_data\_id JOIN place\_study AS ps ON ps.personal\_data\_id = pd.personal\_data\_id WHERE ps.education\_level = 'Высшее';



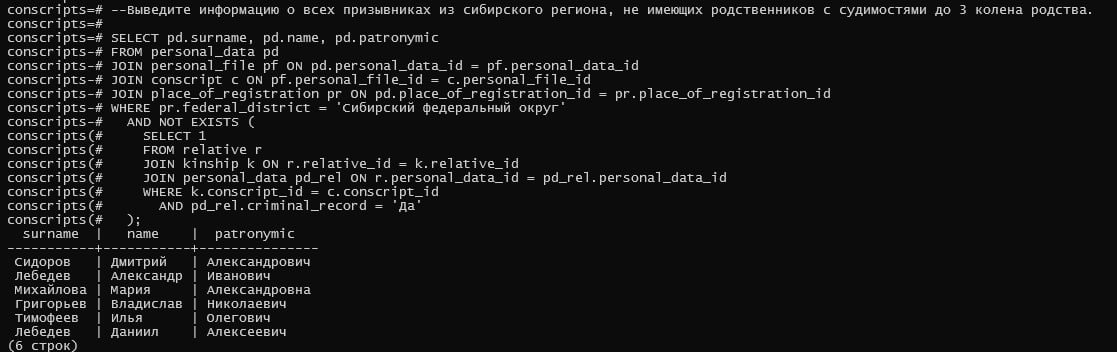
1. Найти всех призывников, которые получили звание в армии.

SELECT pd.surname, pd.name, pd.patronymic, pf.military\_ticket, pf.rank FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id WHERE pf.rank IS NOT NULL;



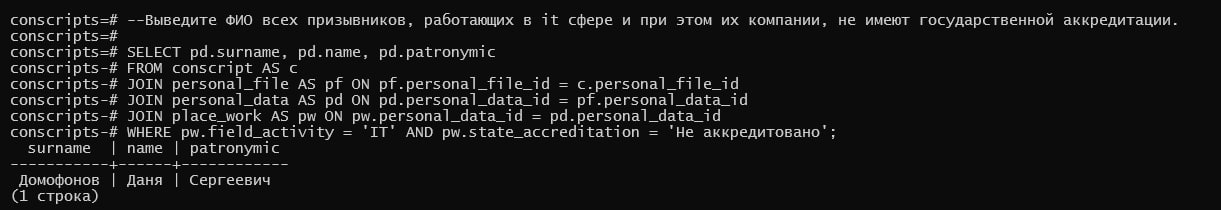
1. Выведите информацию о всех призывниках из сибирского региона, не имеющих родственников с судимостями до 3 колена родства.

SELECT pd.surname, pd.name, pd.patronymic FROM personal\_data pd JOIN personal\_file pf ON pd.personal\_data\_id = pf.personal\_data\_id JOIN conscript c ON pf.personal\_file\_id = c.personal\_file\_id JOIN place\_of\_registration pr ON pd.place\_of\_registration\_id = pr.place\_of\_registration\_id WHERE pr.federal\_district = 'Сибирский федеральный округ' AND NOT EXISTS ( SELECT 1 FROM relative r JOIN kinship k ON r.relative\_id = k.relative\_id JOIN personal\_data pd\_rel ON r.personal\_data\_id = pd\_rel.personal\_data\_id WHERE k.conscript\_id = c.conscript\_id AND pd\_rel.criminal\_record = 'Да' );



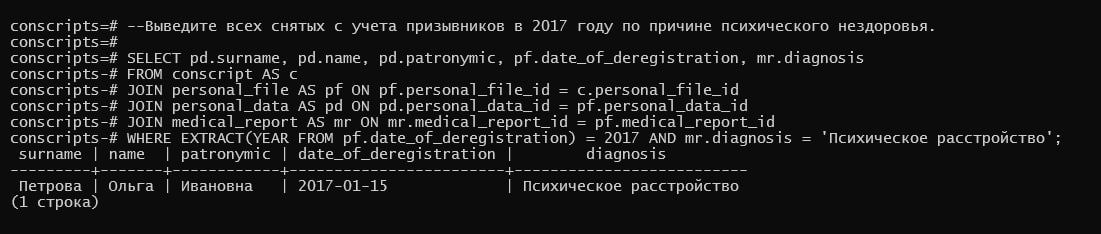
1. Выведите ФИО всех призывников, работающих в it сфере и при этом их компании, не имеют государственной аккредитации.

SELECT pd.surname, pd.name, pd.patronymic FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN place\_work AS pw ON pw.personal\_data\_id = pd.personal\_data\_id WHERE pw.field\_activity = 'IT' AND pw.state\_accreditation = 'Не аккредитовано';



1. Выведите всех снятых с учета призывников в 2017 году по причине психического нездоровья.

SELECT pd.surname, pd.name, pd.patronymic, pf.date\_of\_deregistration, mr.diagnosis FROM conscript AS c JOIN personal\_file AS pf ON pf.personal\_file\_id = c.personal\_file\_id JOIN personal\_data AS pd ON pd.personal\_data\_id = pf.personal\_data\_id JOIN medical\_report AS mr ON mr.medical\_report\_id = pf.medical\_report\_id WHERE EXTRACT(YEAR FROM pf.date\_of\_deregistration) = 2017 AND mr.diagnosis = 'Психическое расстройство';



**Вывод**

В ходе выполнения лабораторной работы было проведено концептуальное и физическое проектирование базы данных. Цель работы заключалась в проектировании базы данных для хранения информации призывниках.

В процессе работы были решены следующие задачи:

Были определены основные сущности: призывник, персональные данные, родственники, личное дело, медицинское заключение, тип семьи, призыв, военная часть, место работы, место учебы и место регистрации. Были определены атрибуты и связи между ними.

Были разработаны и выполнены различные запросы к базе данных, включая получение списков призывников с различными атрибутами, поиск призывников по определенным критериям (например, статус годности к военной службе или место проживания), а также получение информации о родственниках призывников.

Были рассчитаны проценты полностью здоровых призывников для каждого региона, скользящее значение прошедших военную службу призывников в определенных войсках к определенному году, а также установлен порядок призыва призывников в зависимости от их медицинского здоровья и судимостей родственников.

Были созданы триггеры, которые автоматически выполняют определенные действия в базе данных при наступлении определенных событий или условий. Например, триггер может обновлять связанные данные при изменении определенного поля или выполнять дополнительные действия при вставке новой записи.