Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Факультет прикладной математики и механики

кафедра «Математическое моделирование систем и процессов»

направление (профиль) программы бакалавриата: 01.03.02 – «Прикладная математика и информатика», «Математическое моделирование»

**О Т Ч Е Т**

**По лабораторной работе по предмету**

**«Базы данных и экспертные системы»**

**Тема: «Проектирование базы данных средневековой деревни»**

Выполнил студент гр.

ММ-19-2б

Мельников Демид Леонидович

(Фамилия, имя, отчество)

**Проверил:**

доцент каф. ММСП, Янц Антон Юрьевич

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

(оценка) (подпись)

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

(дата)

**Пермь, 2022**

Оглавление

[Постановка задачи 3](#_Toc104199876)

[Требования к базе 3](#_Toc104199877)

[Логическое проектирование 3](#_Toc104199878)

[Физическое проектирование 5](#_Toc104199879)

[Составление SQL-запросов 7](#_Toc104199880)

[Выводы 13](#_Toc104199881)

# Постановка задачи

Необходимо построить базу данных деревни, чтобы помочь феодалу в управлении.

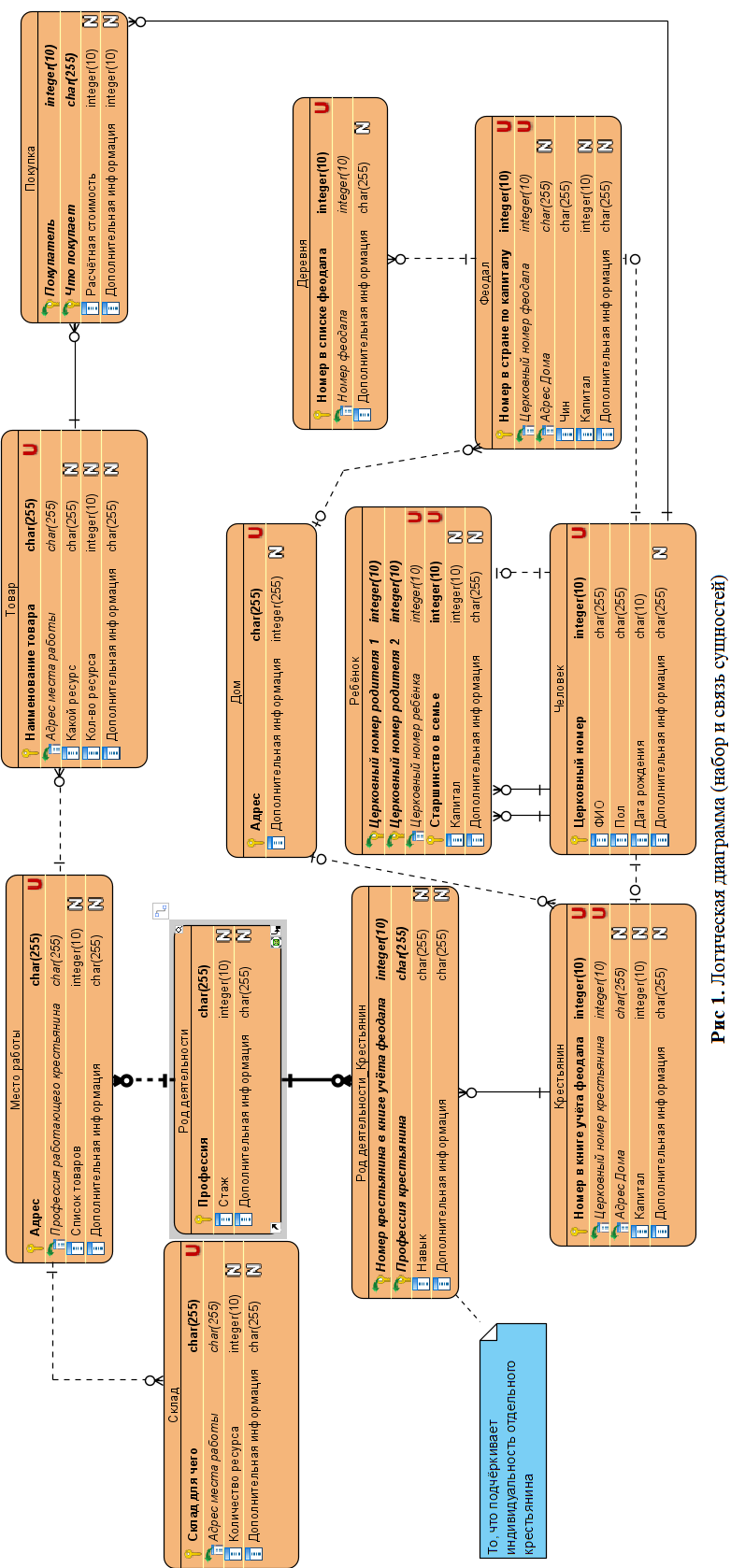
# Требования к базе

* База данных должна хранить всех людей, живущих в деревне. В ней должны храниться их личные данные: имя, пол, дата рождения, профессиональные навыки и т.д.;
* База данных должна хранить адреса всех домов и улиц в деревне;
* База данных должна, отдельно от людей, содержать в себе детей;
* База данных должна хранить товары, которыми крестьяне могут торговать между собой, а также должна фиксировать все покупки;

# Логическое проектирование

Произведём анализ данной системы. Выделим сущности, присутствующие в системе и их атрибуты, диаграмма сущностей приведена на рис. 1. У всех сущностей в последней строке идёт дополнительная информация, которая не обязательна:

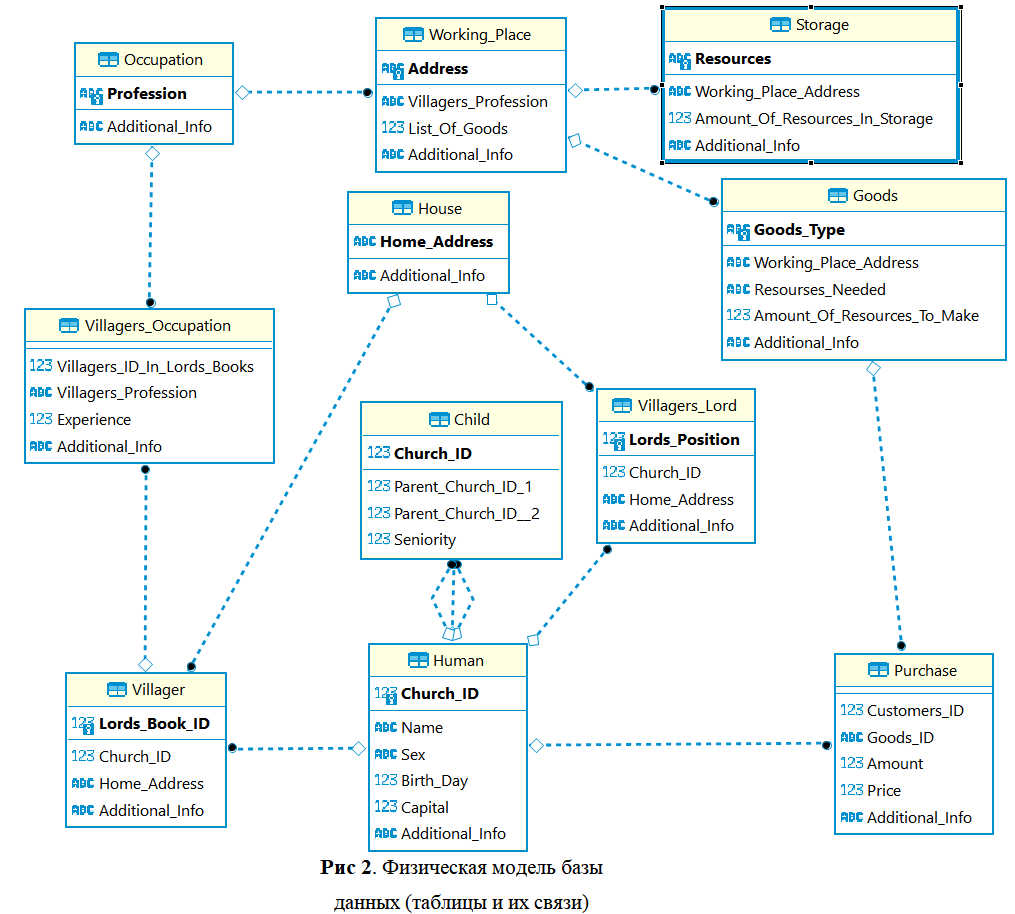
* Человек (номер в церковной книге, имя, пол, дата рождения, личный капитал);
* Ребёнок (церковный номер первого родителя, церковный номер второго родителя, старшинство в семье);
* Крестьянин (прописка, церковный номер человека, адрес дома);
* Феодал (главенство, церковный номер человека, адрес дома);
* Дом (адрес);
* Род деятельности (профессия);
* Род деятельности крестьянина (прописка крестьянина, профессия крестьянина, опыт работы);
* Место работы крестьянина (адрес, профессия крестьянина, список товаров);
* Склад (ресурс на хранении, адрес места работы, количество ресурсов);
* Товар (наименование товара, адрес места работы, необходимый ресурс для создания, сколько этого ресурса нужно для создания);
* Покупка (церковный номер покупателя, наименование товара, количество товара, цена);



Физическое проектирование

Проведем более глубокий и детальный анализ для построения конечной базы данных. Выделим необходимые таблицы, а также их взаимосвязи, обеспечивающие достойную функциональность системы. Физическая модель приведена на рис. 2:

* *Human* (PK – Church\_ID) – таблица, хранящая в себе информацию о Человеке. Имеет связь с таблицами Child,Villagers\_Lord, Villager, Goods ( через Purchase);
* *Child* (PK - Church\_ID, Parent\_Church\_ID\_1, Parent\_Church\_ID\_2) – таблица, хранящая в себе информацию о Ребёнке. Имеет связь с таблицей Human;
* *Villagers\_Lord* (PK – Lords\_Position) – таблица, хранящая в себе информацию о Феодале. Имеет связь с таблицами Human, House;
* *Villager* (PK – Lords\_Book\_ID) – таблица, хранящая в себе информацию о Крестьянине. Имеет связь с таблицами Human, House, Occupation (через Villagers\_Occupation);
* *House* (PK – Home\_Address) – таблица, хранящая в себе информацию о Доме. Имеет связь с таблицами Villager, Villagers\_Lord;
* *Occupation* (PK – Profession) – таблица, хранящая в себе информацию о Роде деятельности. Имеет связь с таблицами Working\_Place, Villager (через Villagers\_Occupation);
* *Villagers\_Occupation* (PK – Villagers\_ID\_In\_Lords\_Book, Villagers\_Profession) – таблица, хранящая в себе информацию о Роде деятельности крестьянина. Имеет связь с таблицами Villager, Occupation;
* *Working\_Place* (PK – Address) – таблица, хранящая в себе информацию о Месте работе. Имеет связь с таблицами Storage, Occupation;
* *Storage* (PK – Resources) – таблица, хранящая в себе информацию о Складе. Имеет связь с таблицей Working\_Place;
* *Goods* (PK – Goods\_Type) – таблица, хранящая в себе информацию о Товаре. Имеет связь с таблицами Working\_Place, Human (через Purchase);
* *Purchase* (PK – Customers\_ID, Goods\_ID) – таблица, хранящая в себе информацию о Покупках. Имеет связь с таблицами Human, Goods;



# Составление SQL-запросов

1. Вывод женщин, родившихся после 1250 года, упорядоченных по возрастанию капитала:

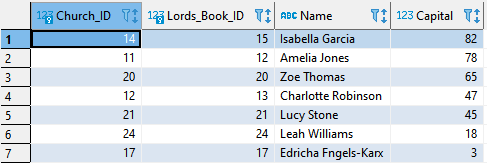
**Select** h.Church\_ID, v.Lords\_Book\_ID , h.Name, h.Capital **From** Human h

**InnerJoin** Villager v **On** v.Church\_ID = h.Church\_ID

**Where** Sex = "F"**And** Birth\_Day>= "1250-01-01"

**Order By** Capital **Desc**;

**Результат:**



1. Вывод мужчин, у которых более одного ребёнка:

**Select Count**(h.Church\_ID), h.Name**From** Human h

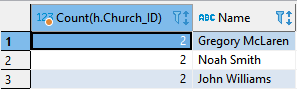
**Inner Join** Child c **On** c. Parent\_Church\_ID\_1 = h.Church\_ID **Or** c.Parent\_Church\_ID\_\_2 = h.Church\_ID

**Where** Sex = "M"

**Group** By h.Church\_ID

**Having Count**(h.Church\_ID) > 1;

**Результат:**



**Комментарии:**

* Если не использовать Having, а добавить условие в where, то выдаёт ошибку.

**Where** Sex = "M" And Count(h.Church\_ID) > 1

1. Вывод случайного крестьянина

**Select** h.Church\_ID, v.Lords\_Book\_ID, h.Name, h.Sex,h.Birth\_Day, v.Home\_Address, vo.Villagers\_Profession,vo.Experience, h.Capital, h.Additional\_Info **From** Human h, Villagers\_Occupation vo

**InnerJoin** Villager v **On** v.Church\_ID = h.Church\_ID

**Where** Birth\_Day <= "1285-01-01"

**Order By** Random ()

**Limit** 1;

**Результат:**





1. Вывод всех работающих крестьян

**Select** h.Church\_ID, v.Lords\_Book\_ID, h.Name, h.Sex, h.Birth\_Day, vo.Villagers\_Profession, vo.Experience **From** Human h

**Inner Join** Villagers\_Occupation vo, Villager v On v.Lords\_Book\_ID = vo.Villagers\_ID\_In\_Lords\_Books **And** v.Church\_ID = h.Church\_ID

**Group by** h.Church\_ID

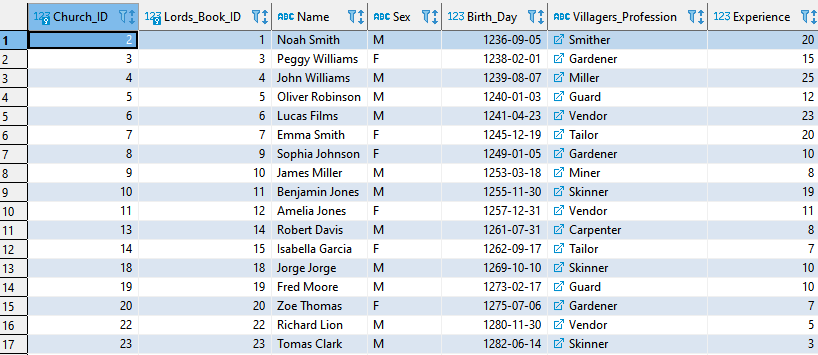
**Except**

**Select** h.Church\_ID, v.Lords\_Book\_ID, h.Name, h.Sex, h.Birth\_Day, vo.Villagers\_Profession, vo.Experience **From** Human h

**Inner Join** Villagers\_Occupationvo, Villager v On v.Lords\_Book\_ID = vo.Villagers\_ID\_In\_Lords\_Books **And** v.Church\_ID = h.Church\_ID

**Where** vo.Villagers\_Profession = "Unemployed";

**Результат:**



1. Вывод всех товаров, для изготовления которых необходимы ресурсы, вместе с адресом изготовления и профессией жителя

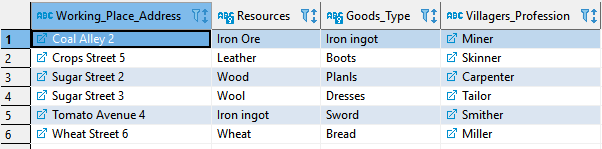
**Select** g.Working\_Place\_Address, s.Resources, g.Goods\_Type, wp.Villagers\_Profession **From** Goods g , Working\_Place wp

**Left Outer Join** Storage s On s.Working\_Place\_Address = g.Working\_Place\_Address

**Where** wp.Address = s.Working\_Place\_Address

**Group by** s.Working\_Place\_Address;

**Результат:**



**Комментарии:**

* Если использовать Outer Join или Right Join, появится ошибка: "RIGHT and FULL OUTER JOINs are not currently supported", видимо в этой версии их нет.

1. Вывод всех жителей вместе с их домами

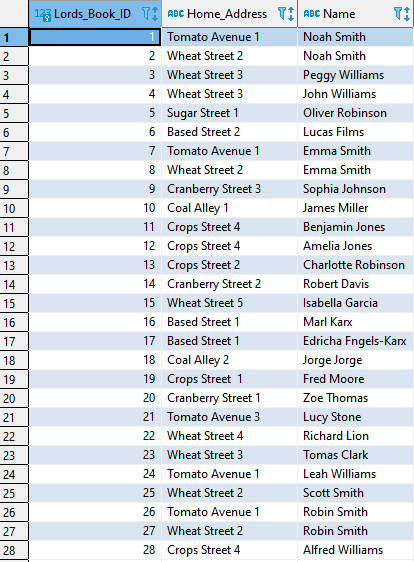
**Select** v.Lords\_Book\_ID , ho.Home\_Address, h.Name **From** Villager v , Human h

**Cross Join** House ho

**Where** v.Home\_Address = ho.Home\_Address **And** h.Church\_ID = v.Church\_ID

**Order by** v.Lords\_Book\_ID;

**Результат:**



**Комментарии:**

* Если у жителя больше одной прописки, то выведет все его дома.

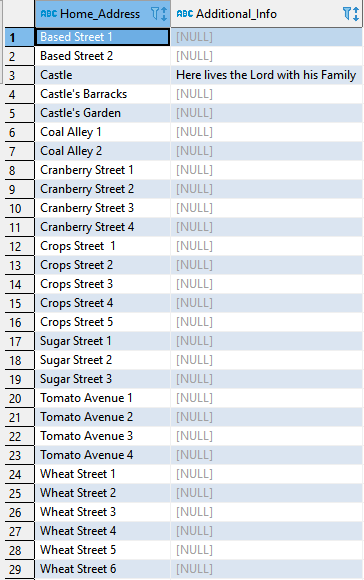
1. Список всех возможных строений

**Select** h.Home\_Address, h.Additional\_Info **From** House h

**Union**

**Select** wp.Address, wp.Additional\_Info **From**Working\_Place wp;

**Результат:**



**Комментарии:**

* Выводит улицы и постройки внутри замка, которые являются рабочими местами и не являются домом с конкретной улицей.

1. Вывод всех товаров и их принадлежности

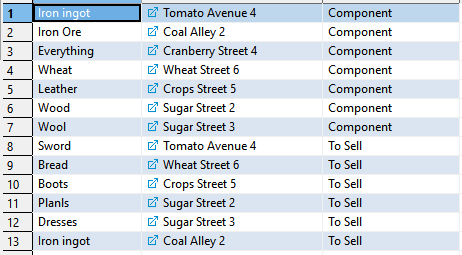
**Select** s.Resources, s.Working\_Place\_Address, s.Additional\_Info **From** Storage s

**UnionAll**

**Select** g.Goods\_Type, g.Working\_Place\_Address, g.Additional\_Info **From** Goods g

**Order by** Additional\_Info;

**Результат:**



**Комментарии:**

* Выводит полноценный товар и выведет его же, если он является компонент для создания другого

1. Вывод вещей, которые являются и компонентом, и полноценным товаром

**Select** s.Resources **From** Storage s

**Intersect**

**Select** g.Goods\_Type **From** Goods g;

**Результат:**



1. Представление из пенсионеров

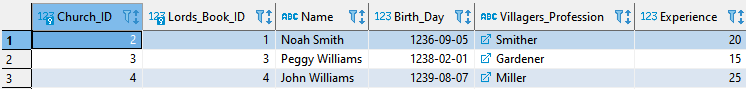
**Create View If Not Exists** Age **As**

**Select** h.Church\_ID, h.Name,v.Lords\_Book\_ID , h.Birth\_Day,vo.Villagers\_Profession, vo.Experience **From** Human h, Villager v

**Inner Join** Villagers\_Occupation vo **On** vo.Villagers\_Profession != "Unemployed" **And** v.Church\_ID = h.Church\_ID **And** vo.Villagers\_ID\_In\_Lords\_Books = v.Lords\_Book\_ID

**Where** h.Birth\_Day <= **Date**("1240-01-01");

**Результат:**



1. Сколько денег было бы у каждого, если бы настал коммунизм

**Select Sum**(h.Capital) / 30 **As** Allmoney **From** Human h ;

**Результат:**



# Выводы

Спроектированы физическая и логические модели базы данных, так же создана база данных «Средневековая деревня». Данный проект позволяет феодалу облегчить свои обязанности. Потенциал применения в видео игровой индустрии.