



ВИДЫ БАЗ ДАННЫХ

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

АЛЕКСЕЙ ЛАВРОВ
УНИВЕРСИТЕТ ИТМО

11.02.2025

ЧТО ТАКОЕ БАЗА ДАННЫХ ?

- **База данных (БД, DataBase, DB)** – организованная в соответствии с определёнными правилами и хранящаяся в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

- **СУБД** (*DataBase Management System, DBMS*) – это совокупность программных средств для создания и использования баз данных.

Основное назначение СУБД:

- хранение данных;
- обеспечение доступа к данным (и контроль прав доступа);
- контроль целостности данных (проверка соблюдения правил) при добавлении, изменении или удалении данных.

ФОРМАТЫ ДАННЫХ И МОДЕЛИ ДАННЫХ

Формат данных определяет, как «физически» представлены данные в БД, то есть как информация преобразуется в двоичный код, хранящийся в памяти.

Модель данных определяет, как на логическом уровне рассматривается структура данных в БД, как данные логически взаимосвязаны.

Физически база данных хранится на компьютере в виде файла или набора файлов.

Движок базы данных (*Database Engine*) – компонент СУБД, отвечающий за хранение и изменение данных. Именно движок определяет формат данных.

Примеры движков: *InnoDB*, *MyISAM*, *SQLite*, *MS Jet*, и др.

ОСНОВНЫЕ ФОРМАТЫ ФАЙЛОВ ДЛЯ СУБД

бинарные

текстовые

табличные

- CSV
- TSV
- и т.п.

древовидные

- XML
- JSON
- и т.п.

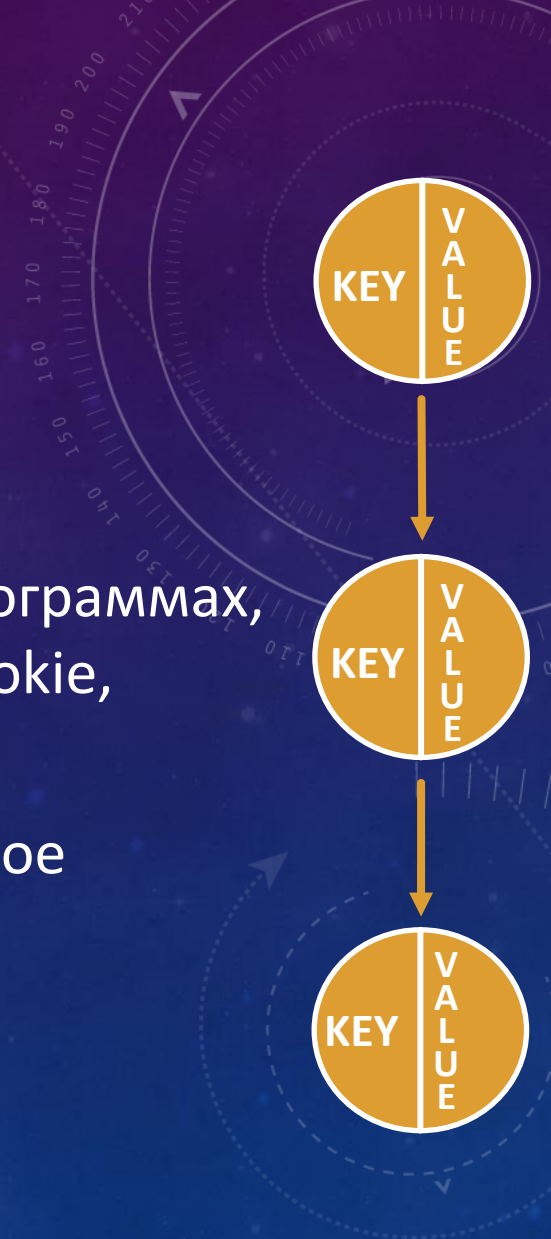
и другие...

МОДЕЛИ ДАННЫХ

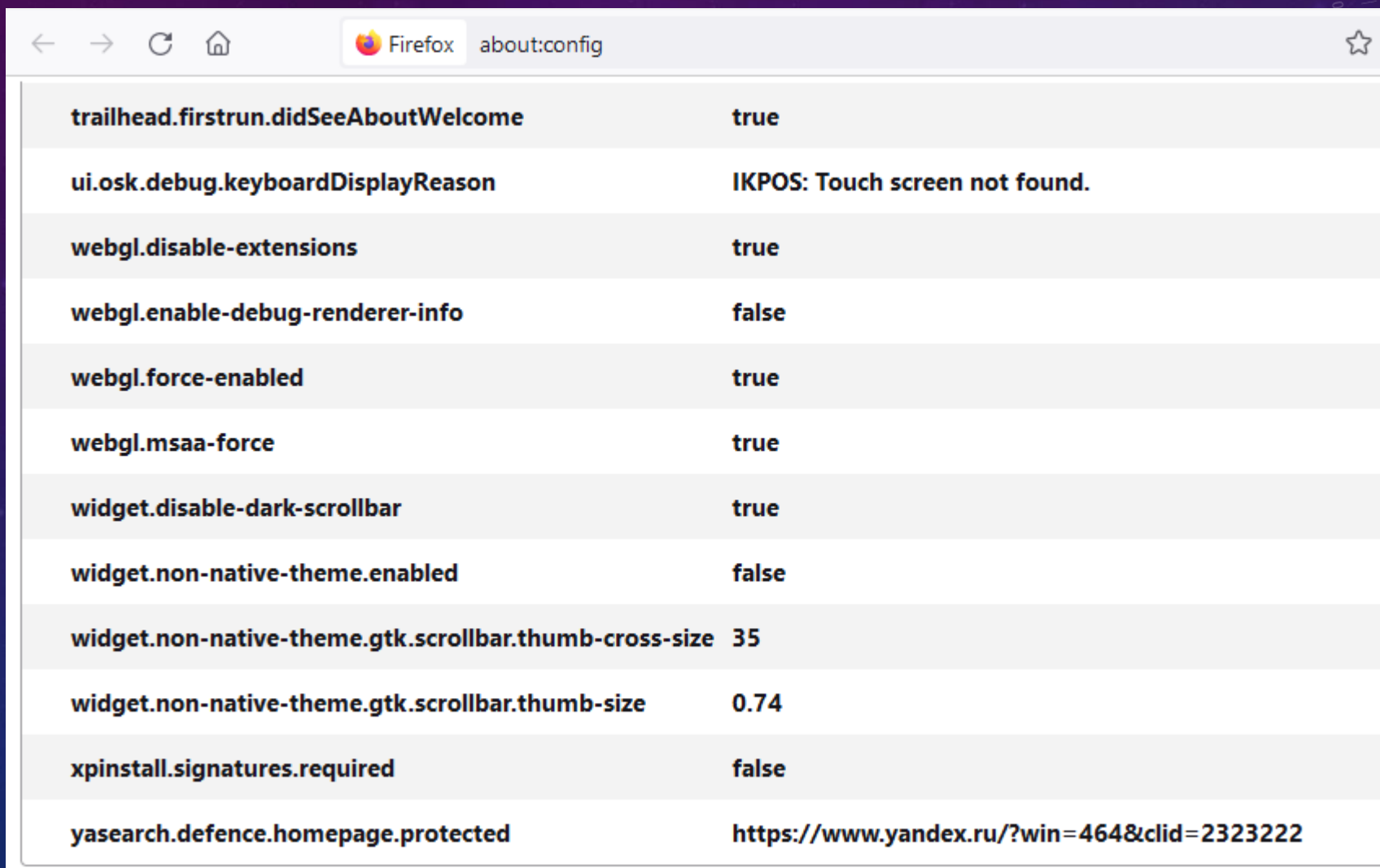
- **Линейные** (хранилища «ключ-значение»). Примеры СУБД: *MemcacheDB, Redis*.
- **Иерархические** («документо-ориентированные БД»).
Примеры СУБД: *MongoDB, Sedna, eXist*.
- **Сетевые** (СУБД не распространены).
- **Реляционные**. Примеры СУБД: *MS SQL, MySQL, MariaDB, Firebird, MS Access*.
- **Объектно-ориентированные** (СУБД не распространены).
- **Объектно-реляционные**. Примеры СУБД: *Oracle Database, PostgreSQL*.

ЛИНЕЙНАЯ МОДЕЛЬ ДАННЫХ

- Используется в простейших БД, называемых “key-value storage” (хранилище «ключ-значение»). Например, *MemcacheDB*, *Redis*.
- Также по этой модели устроено хранение настроек во многих программах, например: реестр Windows, конфигурация Firefox, хранилище Cookie, хранилища LocalStorage и sessionStorage в браузерах, и т.п.
- Ключ представляет собой некоторое название (строку), уникальное в этом хранилище.
- Каждому ключу сопоставлено значение, которое может быть некоторого простого типа (число, строка, булево значение, и т.п.)
- Линейная модель очень примитивна и непригодна для хранения сложно взаимосвязанных данных.



ПРИМЕР ЛИНЕЙНОЙ МОДЕЛИ ДАННЫХ: FIREFOX config



The image shows a screenshot of the Firefox browser's `about:config` page. The browser's address bar at the top displays the Firefox logo and the text `about:config`. Below the address bar, a table lists various configuration settings. Each row consists of a configuration name on the left and its current value on the right. The rows are separated by alternating light gray and white background colors. The settings include boolean values (`true`, `false`), a string, and numeric values.

<code>trailhead.firstrun.didSeeAboutWelcome</code>	<code>true</code>
<code>ui.osk.debug.keyboardDisplayReason</code>	<code>IKPOS: Touch screen not found.</code>
<code>webgl.disable-extensions</code>	<code>true</code>
<code>webgl.enable-debug-renderer-info</code>	<code>false</code>
<code>webgl.force-enabled</code>	<code>true</code>
<code>webgl.msaa-force</code>	<code>true</code>
<code>widget.disable-dark-scrollbar</code>	<code>true</code>
<code>widget.non-native-theme.enabled</code>	<code>false</code>
<code>widget.non-native-theme.gtk.scrollbar.thumb-cross-size</code>	<code>35</code>
<code>widget.non-native-theme.gtk.scrollbar.thumb-size</code>	<code>0.74</code>
<code>xpinstall.signatures.required</code>	<code>false</code>
<code>yasearch.defence.homepage.protected</code>	<code>https://www.yandex.ru/?win=464&clid=2323222</code>

ПРИМЕР ЛИНЕЙНОЙ МОДЕЛИ ДАННЫХ: РЕЕСТР WINDOWS

Редактор реестра

Файл Правка Вид Избранное Справка

Компьютер\HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0

Имя	Тип	Значение
(По умолчанию)	REG_SZ	(значение не присвоено)
~MHz	REG_DWORD	0x00000b4d (2893)
Component Information	REG_BINARY	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Configuration Data	REG_FULL_RESOURCE_DESCRIPTOR	ff ff ff ff ff ff ff 00 00 00 00 00 00 00 00
FeatureSet	REG_DWORD	0x3d1b3fff (1025196031)
Identifier	REG_SZ	Intel64 Family 6 Model 60 Stepping 3
Platform Specific Field 1	REG_DWORD	0x00000020 (32)
Previous Update Revision	REG_BINARY	00 00 00 00 1d 00 00 00
ProcessorNameString	REG_SZ	Intel(R) Core(TM) i5-4210H CPU @ 2.90GHz
Update Revision	REG_BINARY	00 00 00 00 25 00 00 00
Update Status	REG_DWORD	0x00000000 (0)
VendorIdentifier	REG_SZ	GenuineIntel

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

- Данные представлены как древовидные структуры.
- Каждый элемент (кроме корневого) имеет ровно одного родителя и может иметь или не иметь любое количество потомков.
- Многие современные иерархические СУБД основаны на древовидных форматах XML или JSON. База данных в таких СУБД представляет собой набор XML-документов или JSON-документов, их называют «документо-ориентированными базами данных».
- Примеры: *MongoDB*, *Sedna*, *eXist*.



Объекты restaurants @info_db (Mongo...

Начать транзакцию Фильтр Сортировка Импорт Экспорт Анализ

```
// 2
{
  "_id": objectId("55f141312c7447c3da7051b28"),
  "URL": "http://www.just-eat.co.uk/restaurants-atthairestaurant/menu",
  "address": "30 Greyhound Road Hammersmith",
  "address line 2": "London",
  "name": "@ Thai Restaurant",
  "outcode": "W6",
  "postcode": "8NX",
  "rating": 4.5,
  "type_of_food": "Thai",
  "location": {
    "coordinates": [
      -74.005289,
      40.628886
    ],
    "type": "Point"
  }
}
```

// 3

db.getCollection("restaurants").update({_id: ObjectId("55f141312c7447c3da7051b28")}, {

Отображение JSON

Объекты restaurants @info_db (Mongo...

Выполнить Совершить откат Текст Фильтр

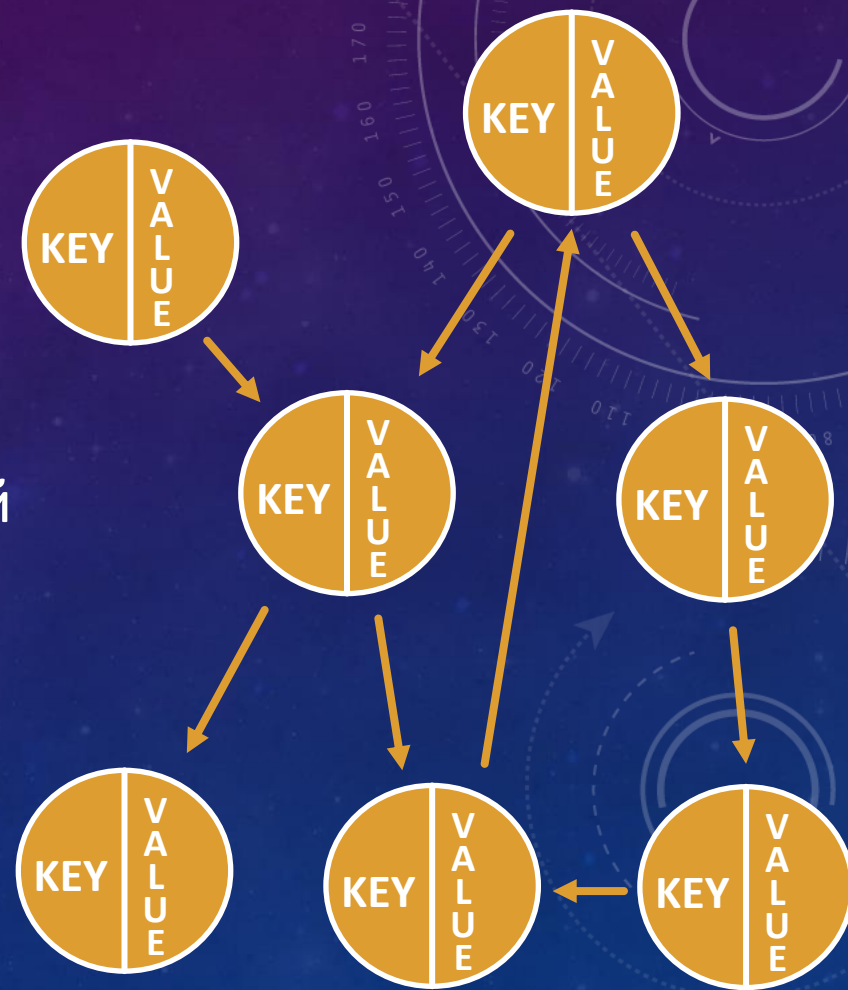
_id: 55f141312c7447c3da7051b28
URL: http://www.just-eat.co.uk/restaurants-atthairestaurant/menu
address: 30 Greyhound Road Hammersmith
address line 2: London
name: @ Thai Restaurant
outcode: W6
postcode: 8NX
rating: 4.5
type_of_food: Thai
location: (Document) 2 items
 coordinates: (Array) 2 items
 0: -74.005289
 1: 40.628886
 type: Point
30 Greyhound Road Hammersmith

Отображение в виде дерева

ПРИМЕР
ИЕРАРХИЧЕСКОЙ
СУБД:
MongoDB

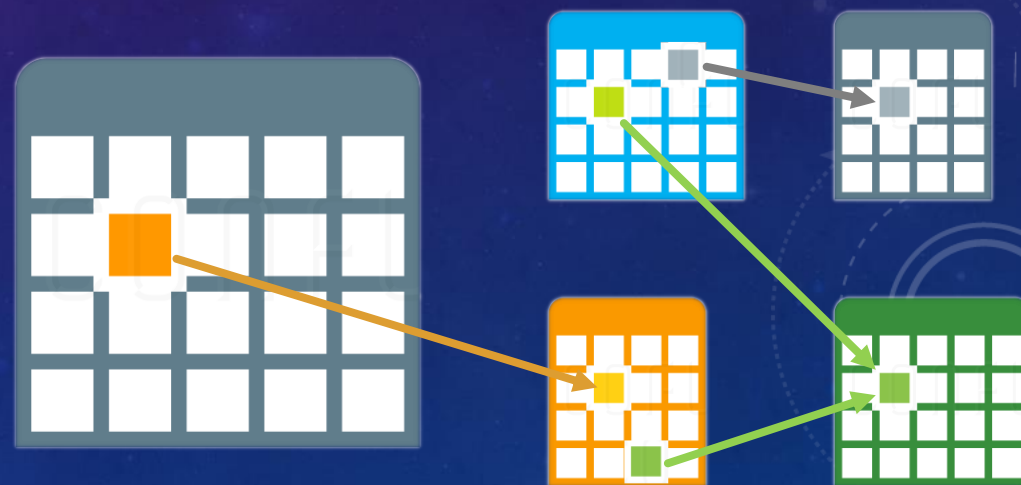
СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

- В сетевой модели каждый элемент может быть напрямую связан с любым другим элементом.
- Сетевая модель универсальна (позволяет хранить любые связи между данными, в отличие от линейной и иерархической), но более затратна в реализации.
- На практике сетевая модель не получила распространения, потому что в то же время была придумана реляционная модель, которая так же универсальна, но проще в реализации.



РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

- Реляционная база данных состоит из набора так называемых «**отношений**» (**relation**), каждое из которых хранится в виде **таблицы** значений.
- Связь между данными происходит за счёт хранения совпадающих значений в разных таблицах.
- Каждый столбец таблицы имеет конкретный тип и смысловое значение. Для каждой таблицы строго задано, из каких столбцов она состоит. А количество строк может быть любым.
- Примеры реляционных СУБД:
MS SQL, MySQL, MariaDB, Firebird, MS Access.



ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ МОДЕЛЬ ДАННЫХ

- Объектно-ориентированная база данных описывается как набор **классов**, данные хранятся как объекты (экземпляры классов).
- Каждый класс состоит из описания свойств (полей) и методов (функций).
- Таким образом, в ООБД реализуется инкапсуляция (привязка действий над данными к самим данным). Также может быть реализовано наследование.

ОБЪЕКТНО-РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

- Данные хранятся в отношениях (таблицах), но к этим таблицам привязаны методы для обработки и вывода этих данных.
- Примеры объектно-реляционных СУБД: *Oracle Database, PostgreSQL*.



МАТЕМАТИЧЕСКАЯ ТЕОРИЯ ОТНОШЕНИЙ

ОСНОВНЫЕ ПОНЯТИЯ

- Математическое понятие **«отношение»** основывается на понятиях **«множество»**, **«подмножество»** и **«декартово произведение»**.
- **Множество (set)** – это несколько неупорядоченных неповторяющихся элементов.
- **Подмножество (subset)** – это множество, составленное из элементов другого множества.
- **Декартово произведение (Cartesian product)** – это операция над множествами, в результате которой получается множество всех возможных комбинаций, содержащих по одному элементу из каждого множества.

ПРИМЕРЫ ПОДМНОЖЕСТВ

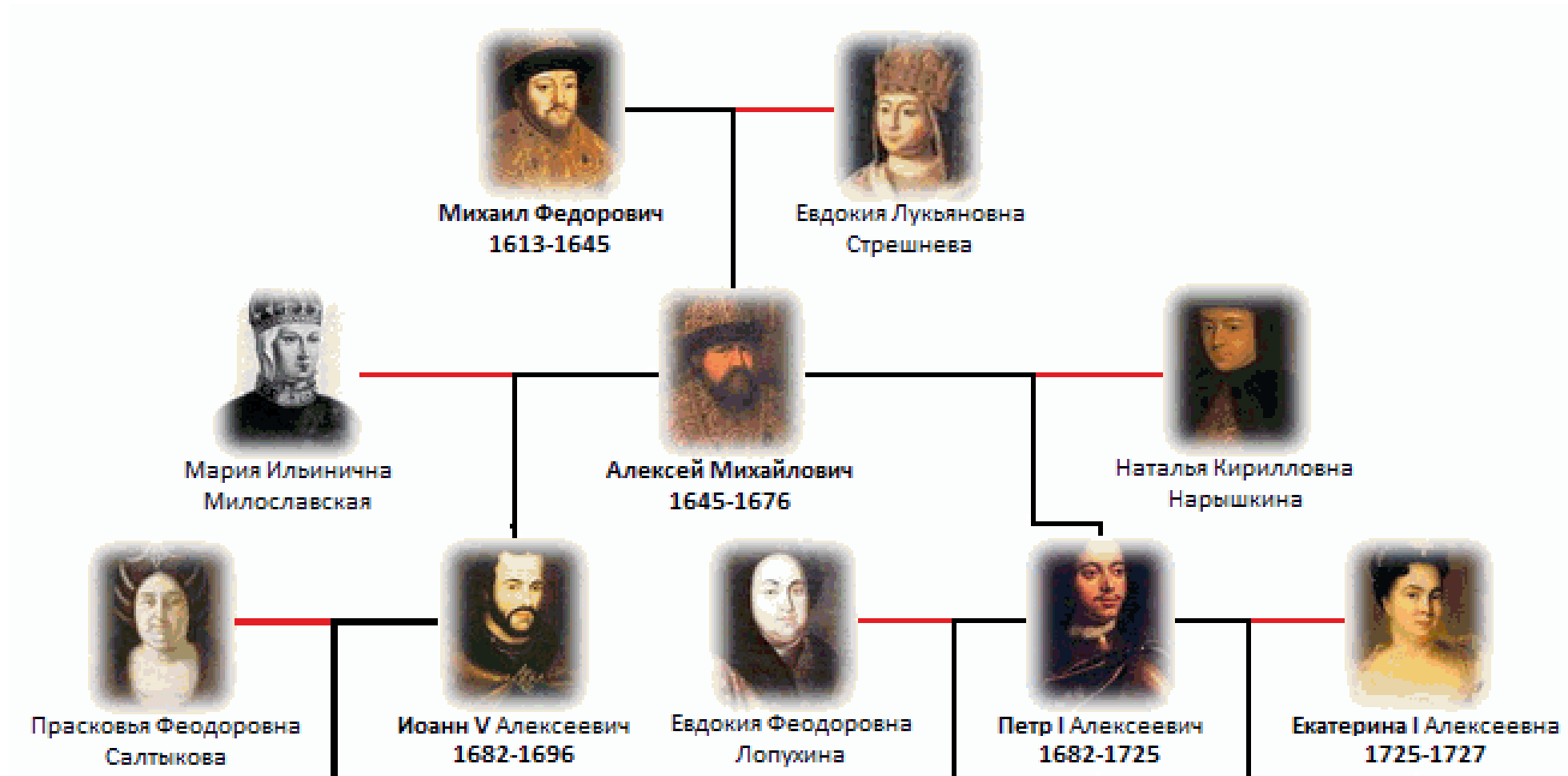
- Пусть множество A состоит из элементов 1, 2 и 3:
 $A = \{1, 2, 3\}$
- **Подмножествами** множества A будут являться множества:
 - $\{1, 2\}$
 - $\{2, 3\}$
 - $\{1, 3\}$
 - $\{1\}$
 - $\{2\}$
 - $\{3\}$
 - $\{1, 2, 3\}$
 - $\{\} = \emptyset$
- Порядок элементов в множестве не играет роли: $\{1, 2, 3\} = \{3, 2, 1\} = \{3, 1, 2\}$
- Вхождение одного множества в другое в качестве подмножества обозначается знаком \subset . Например: если $X = \{2, 1\}$, то $X \subset A$.

ПРИМЕРЫ ДЕКАРТОВЫХ ПРОИЗВЕДЕНИЙ

- Пусть множество $A = \{1, 2, 3\}$; множество $B = \{7, 8\}$
- **Декартовым произведением** $A \times B$ будет множество пар:
 $\{ (1, 7), (2, 7), (3, 7), (1, 8), (2, 8), (3, 8) \}$
- **Декартовым произведением** $B \times A$ будет множество пар:
 $\{ (7, 1), (7, 2), (7, 3), (8, 1), (8, 2), (8, 3) \}$
- **Декартовым произведением** $B \times B$ будет множество пар:
 $\{ (7, 8), (8, 7), (7, 7), (8, 8) \}$
- **Декартовым произведением** $B \times B \times A$ будет множество троек:
 $\{ (7, 8, 1), (7, 8, 2), (7, 8, 3), (8, 7, 1), (8, 7, 2), (8, 7, 3), (7, 7, 1), (7, 7, 2), (7, 7, 3), (8, 8, 1), (8, 8, 2), (8, 8, 3) \}$
- Каждая получаемая пара, тройка и т.д. называется **кортеж (tuple)**.

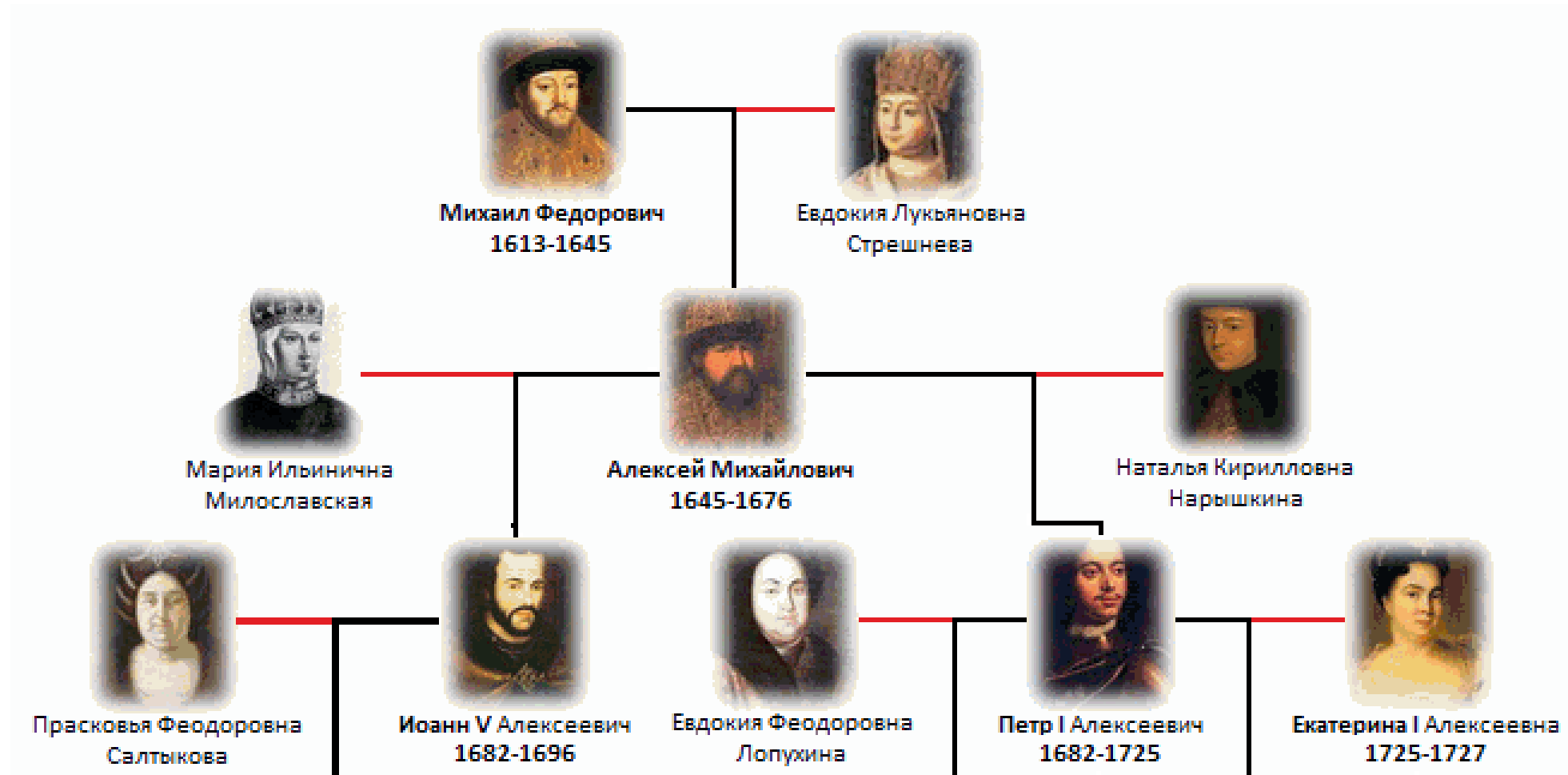
РАЗЛИЧИЯ МНОЖЕСТВ И КОРТЕЖЕЙ

- Элементы множества принято перечислять в фигурных скобках, а элементы кортежей – в круглых или в треугольных скобках.
- В отличие от множеств, в кортежах порядок элементов имеет значение:
 $(1, 2, 3) \neq (3, 2, 1)$.
- Рассмотрим теперь более жизненный пример...



Пусть **M** – множество представителей династии мужского пола.

M = {Михаил Фёдорович, Алексей Михайлович, Иоанн Алексеевич, Пётр Алексеевич}



F – множество представителей династии женского пола.

F = {Евдокия Стрешнева, Мария Милославская, Наталья Нарышкина, Прасковья Салтыкова, Евдокия Лопухина, Екатерина Алексеевна}

M = {Михаил Фёдорович, Алексей Михайлович, Иоанн Алексеевич, Пётр Алексеевич}

F = {Евдокия Стрешнева, Мария Милославская, Наталья Нарышкина, Прасковья Салтыкова, Евдокия Лопухина, Екатерина Алексеевна}

Декартово произведение:

Mx F = { (Михаил Фёдорович, Евдокия Стрешнева), (Михаил Фёдорович, Мария Милославская), (Михаил Фёдорович, Наталья Нарышкина), (Михаил Фёдорович, Прасковья Салтыкова), (Михаил Фёдорович, Евдокия Лопухина), (Михаил Фёдорович, Екатерина Алексеевна), (Алексей Михайлович, Евдокия Стрешнева), (Алексей Михайлович, Мария Милославская), (Алексей Михайлович, Наталья Нарышкина), (Алексей Михайлович, Прасковья Салтыкова), (Алексей Михайлович, Евдокия Лопухина), (Алексей Михайлович, Екатерина Алексеевна), (Иоанн Алексеевич, Евдокия Стрешнева), (Иоанн Алексеевич, Мария Милославская), (Иоанн Алексеевич, Наталья Нарышкина), (Иоанн Алексеевич, Прасковья Салтыкова), (Иоанн Алексеевич, Евдокия Лопухина), (Иоанн Алексеевич, Екатерина Алексеевна), (Пётр Алексеевич, Евдокия Стрешнева), (Пётр Алексеевич, Мария Милославская), (Пётр Алексеевич, Наталья Нарышкина), (Пётр Алексеевич, Прасковья Салтыкова), (Пётр Алексеевич, Евдокия Лопухина), (Пётр Алексеевич, Екатерина Алексеевна) }

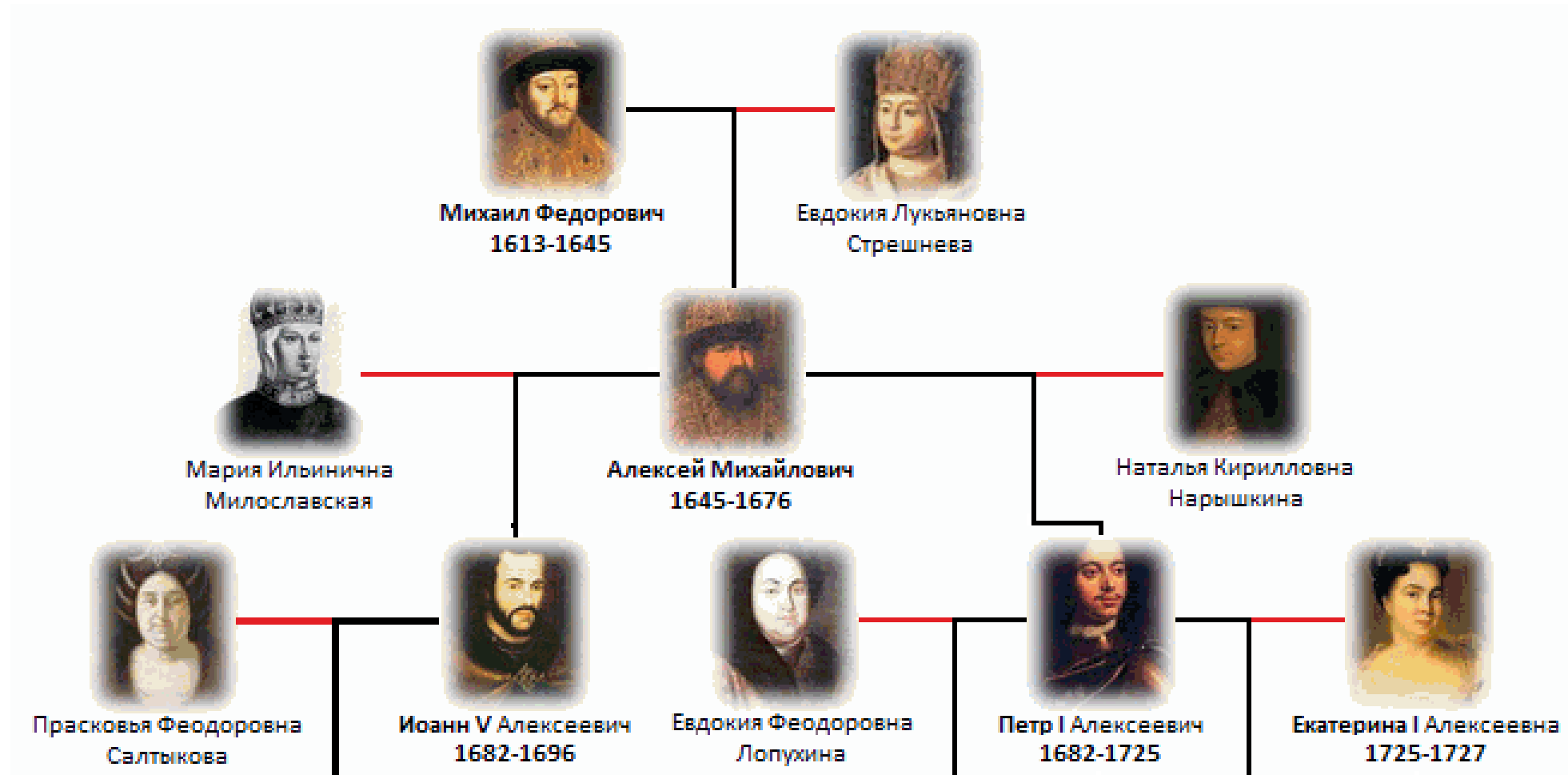
Выделим те пары в декартовом произведении, которые действительно являются парам в реальной жизни (супружескими парами):

$M \times F$ = { (Михаил Фёдорович, Евдокия Стрешнева), (Михаил Фёдорович, Мария Милославская), (Михаил Фёдорович, Наталья Нарышкина), (Михаил Фёдорович, Прасковья Салтыкова), (Михаил Фёдорович, Евдокия Лопухина), (Михаил Фёдорович, Екатерина Алексеевна), (Алексей Михайлович, Евдокия Стрешнева), (Алексей Михайлович, Мария Милославская), (Алексей Михайлович, Наталья Нарышкина), (Алексей Михайлович, Прасковья Салтыкова), (Алексей Михайлович, Евдокия Лопухина), (Алексей Михайлович, Екатерина Алексеевна), (Иоанн Алексеевич, Евдокия Стрешнева), (Иоанн Алексеевич, Мария Милославская), (Иоанн Алексеевич, Наталья Нарышкина), (Иоанн Алексеевич, Прасковья Салтыкова), (Иоанн Алексеевич, Евдокия Лопухина), (Иоанн Алексеевич, Екатерина Алексеевна), (Пётр Алексеевич, Евдокия Стрешнева), (Пётр Алексеевич, Мария Милославская), (Пётр Алексеевич, Наталья Нарышкина), (Пётр Алексеевич, Прасковья Салтыкова), (Пётр Алексеевич, Евдокия Лопухина), (Пётр Алексеевич, Екатерина Алексеевна) }

Выпишем множество этих пар, обозначив его как **R**:

R = { (Михаил Фёдорович, Евдокия Стрешнева), (Алексей Михайлович, Мария Милославская), (Алексей Михайлович, Наталья Нарышкина), (Иоанн Алексеевич, Прасковья Салтыкова), (Пётр Алексеевич, Евдокия Лопухина), (Пётр Алексеевич, Екатерина Алексеевна) }

Множество **R** является подмножеством декартова произведения множеств **M** и **F**: **R** \subset **M** \times **F**.

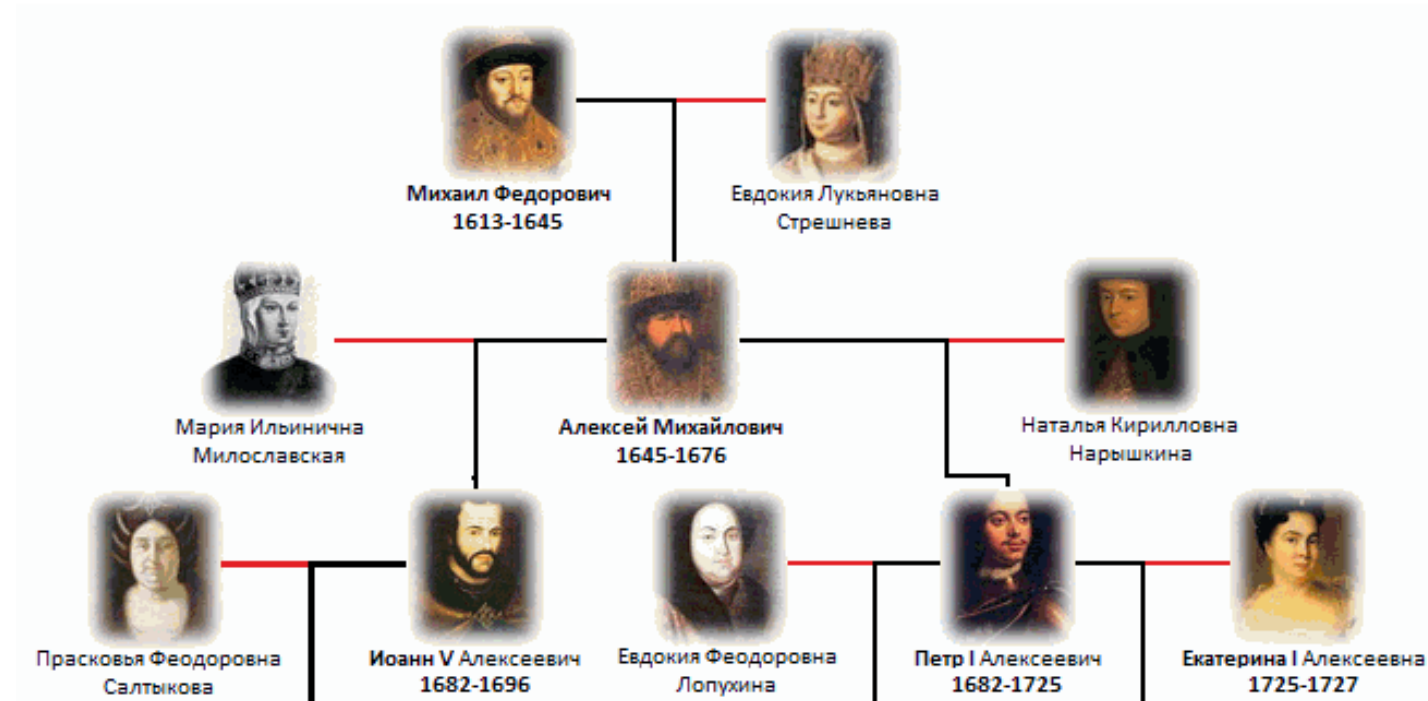


$R = \{ (\text{Михаил Фёдорович, Евдокия Стрешнева}), (\text{Алексей Михайлович, Мария Милославская}), (\text{Алексей Михайлович, Наталья Нарышкина}), (\text{Иоанн Алексеевич, Прасковья Салтыкова}), (\text{Пётр Алексеевич, Евдокия Лопухина}), (\text{Пётр Алексеевич, Екатерина Алексеевна}) \}$

R – это **отношение** на множествах M и F , показывающее супружеские связи между элементами этих множеств.

М = {Михаил Фёдорович, Алексей Михайлович, Иоанн Алексеевич, Пётр Алексеевич}

МхМ = { (Михаил Фёдорович, Михаил Фёдорович), (Михаил Фёдорович, Алексей Михайлович),
(Михаил Фёдорович, Иоанн Алексеевич), (Михаил Фёдорович, Пётр Алексеевич),
(Алексей Михайлович, Михаил Фёдорович), (Алексей Михайлович, Алексей Михайлович),
(Алексей Михайлович, Иоанн Алексеевич), (Алексей Михайлович, Пётр Алексеевич),
(Иоанн Алексеевич, Михаил Фёдорович), (Иоанн Алексеевич, Алексей Михайлович),
(Иоанн Алексеевич, Иоанн Алексеевич), (Иоанн Алексеевич, Пётр Алексеевич),
(Пётр Алексеевич, Михаил Фёдорович), (Пётр Алексеевич, Алексей Михайлович),
(Пётр Алексеевич, Иоанн Алексеевич), (Пётр Алексеевич, Пётр Алексеевич) }



Давайте выпишем те пары,
в которых элементы являются
братьями по отношению друг
к другу.

M = {Михаил Фёдорович, Алексей Михайлович, Иоанн Алексеевич, Пётр Алексеевич}

MxM = { (Михаил Фёдорович, Михаил Фёдорович), (Михаил Фёдорович, Алексей Михайлович),
(Михаил Фёдорович, Иоанн Алексеевич), (Михаил Фёдорович, Пётр Алексеевич),
(Алексей Михайлович, Михаил Фёдорович), (Алексей Михайлович, Алексей Михайлович),
(Алексей Михайлович, Иоанн Алексеевич), (Алексей Михайлович, Пётр Алексеевич),
(Иоанн Алексеевич, Михаил Фёдорович), (Иоанн Алексеевич, Алексей Михайлович),
(Иоанн Алексеевич, Иоанн Алексеевич), (Иоанн Алексеевич, Пётр Алексеевич),
(Пётр Алексеевич, Михаил Фёдорович), (Пётр Алексеевич, Алексей Михайлович),
(Пётр Алексеевич, Иоанн Алексеевич), (Пётр Алексеевич, Пётр Алексеевич) }

B = { (Иоанн Алексеевич, Пётр Алексеевич), (Пётр Алексеевич, Иоанн Алексеевич) }

$$\mathbf{B} \subset \mathbf{M} \times \mathbf{M}$$

B – это **отношение** братства на множестве **M**.

M = {Михаил Фёдорович, Алексей Михайлович, Иоанн Алексеевич, Пётр Алексеевич}

MxM = { (Михаил Фёдорович, Михаил Фёдорович), (Михаил Фёдорович, Алексей Михайлович),
(Михаил Фёдорович, Иоанн Алексеевич), (Михаил Фёдорович, Пётр Алексеевич),
(Алексей Михайлович, Михаил Фёдорович), (Алексей Михайлович, Алексей Михайлович),
(Алексей Михайлович, Иоанн Алексеевич), (Алексей Михайлович, Пётр Алексеевич),
(Иоанн Алексеевич, Михаил Фёдорович), (Иоанн Алексеевич, Алексей Михайлович),
(Иоанн Алексеевич, Иоанн Алексеевич), (Иоанн Алексеевич, Пётр Алексеевич),
(Пётр Алексеевич, Михаил Фёдорович), (Пётр Алексеевич, Алексей Михайлович),
(Пётр Алексеевич, Иоанн Алексеевич), (Пётр Алексеевич, Пётр Алексеевич) }

Перечислим элементы подмножества, соответствующего родственному отношению отцовства:

P = { (Михаил Фёдорович, Алексей Михайлович), (Алексей Михайлович, Иоанн Алексеевич),
(Алексей Михайлович, Пётр Алексеевич) }

ЧТО ТАКОЕ ОТНОШЕНИЕ ?

- В математике **отношением (relation)** между множествами называется любое подмножество декартова произведения этих множеств.

$$R \subset A_1 \times A_2 \times \dots \times A_n$$

- R - отношение на множествах A_1, A_2, \dots, A_n ;
- n - арность отношения R .

АРНОСТЬ ОТНОШЕНИЙ

- Подмножество декартова произведения 2 множеств - бинарное отношение.
- Подмножество декартова произведения 3 множеств - тернарное отношение.
- Подмножество декартова произведения n множеств - n -арное отношение.
- Если какое-то множество фигурирует в декартовом произведении несколько раз, то при подсчёте арности мы рассматриваем его не как одно множество, а как несколько одинаковых множеств.
- Отношение "супруги" ($R \subset M \times F$) - бинарное отношение.
- Отношение "братья" ($B \subset M \times M$) - тоже бинарное отношение.

ОПИСАНИЕ ОТНОШЕНИЙ С ПОМОЩЬЮ ПРЕДИКАТОВ

$R = \{ (x, y): x \text{ является мужем } y \}$ на множествах M, F

$B = \{ (x, y): x \text{ является братом } y \}$ на множестве M

Рассмотрим два множества и их декартово произведение:

$T = \{ \text{Миронов, Сысоева, Сопроненко, Христюлюбова} \}$

$D = \{ \text{Основы композиции, Представление данных, История искусств, Пластанатомия} \}$

$T \times D = \{ (\text{Миронов, Основы композиции}), (\text{Миронов, Представление данных}), (\text{Миронов, История искусств}), (\text{Миронов, Пластанатомия}), (\text{Сысоева, Основы композиции}), (\text{Сысоева, Представление данных}), (\text{Сысоева, История искусств}), (\text{Сысоева, Пластанатомия}), (\text{Сопроненко, Основы композиции}), (\text{Сопроненко, Представление данных}), (\text{Сопроненко, История искусств}), (\text{Сопроненко, Пластанатомия}), (\text{Христюлюбова, Основы композиции}), (\text{Христюлюбова, Пластанатомия}), (\text{Христюлюбова, Представление данных}), (\text{Христюлюбова, История искусств}) \}$

$\mathbf{T} = \{ \text{Миронов, Сысоева, Сопроненко, Христюлюбова} \}$

$\mathbf{D} = \{ \text{Основы композиции, Представление данных, История искусств, Пластанатомия} \}$

Рассмотрим отношение $\mathbf{L} = \{(t,d): t \text{ преподаёт } d\}$ на множествах \mathbf{T}, \mathbf{D} :

$\mathbf{L} = \{ (\text{Миронов, Представление данных}),$
 $(\text{Сысоева, Основы композиции}),$
 $(\text{Христюлюбова, История искусств}),$
 $(\text{Сопроненко, Основы композиции}),$
 $(\text{Сысоева, Пластанатомия}) \}$

Рассмотрим три множества и их декартово произведение:

T = { Миронов, Сысоева, Сопроненко, Христюлюбова }

D = {Основы композиции, Представление данных, История искусств, Пластанатомия}

G = {P33681, P33682}

T×D×G = { (Миронов, Основы композиции, P33681),
(Миронов, Основы композиции, P33682),
(Миронов, Представление данных, P33681),
(Миронов, Представление данных, P33682),
(Миронов, История искусств, P33681),
(Миронов, История искусств, P33682), ...и так далее... }

$\mathbf{T} = \{ \text{Миронов, Сысоева, Сопроненко, Христюлюбова} \}$

$\mathbf{D} = \{ \text{Основы композиции, Представление данных, История искусств, Пластанатомия} \}$

$\mathbf{G} = \{ \text{P33681, P33682} \}$

Рассмотрим отношение $\mathbf{S} = \{ (t,d,g): t \text{ преподаёт } d \text{ в группе } g \}$ на множествах $\mathbf{T}, \mathbf{D}, \mathbf{G}$:

$\mathbf{S} = \{$ (Миронов, Представление данных, P33682),
(Сысоева, Основы композиции, P33682),
(Сысоева, Пластанатомия, P33681),
(Сопроненко, Основы композиции, P33681),
(Христюлюбова, История искусств, P33681),
(Христюлюбова, История искусств, P33682) $\}$

Любое отношение, состоящее из ограниченного набора кортежей, можно представить в виде таблицы.

Преподаватель	Дисциплина	Группа
Миронов	Представление данных	P33682
Сысоева	Основы композиции	P33682
Сысоева	Пластанатомия	P33681
Сопроненко	Основы композиции	P33681
Христюлюбова	История искусств	P33681
Христюлюбова	История искусств	P33682

Но не любая таблица представляет собой отношение.

ОПЕРАЦИИ НАД ОТНОШЕНИЯМИ

1. Поскольку любое отношение является множеством кортежей, то к отношениям применимы все операции, которые могут применяться к множествам:

- пересечение,
- объединение,
- разность,
- симметрическая разность,
- декартово произведение.

2. Некоторые особые операции над отношениями:

- проекция,
- композиция,
- декомпозиция,
- соединение.

ПРОЕКЦИЯ...

Преподаватель	Дисциплина	Группа
Миронов	Представление данных	P33682
Сысоева	Основы композиции	P33682
Сысоева	Пластанатомия	P33681
Сопроненко	Основы композиции	P33681
Христюлюбова	История искусств	P33681
Христюлюбова	История искусств	P33682



Преподаватель	Группа
Миронов	P33682
Сысоева	P33682
Сысоева	P33681
Сопроненко	P33681
Христюлюбова	P33681
Христюлюбова	P33682

Проекция позволяет получить из одного отношения другое, меньшей арности: то есть это такое же отношение, но построенное на меньшем количестве множеств. Если представлять отношения как таблицы, то в результате проекции мы как-бы убираем некоторые столбцы из таблицы...

ПРОЕКЦИЯ...

Преподаватель	Дисциплина	Группа
Миронов	Представление данных	P33682
Сысоева	Основы композиции	P33682
Сысоева	Пластанатомия	P33681
Сопроненко	Основы композиции	P33681
Христолюбова	История искусств	P33681
Христолюбова	История искусств	P33682



Преподаватель	Дисциплина
Миронов	Представление данных
Сысоева	Основы композиции
Сысоева	Пластанатомия
Сопроненко	Основы композиции
Христолюбова	История искусств
Христолюбова	История искусств

Но результатом **проекции** должно быть **отношение**, то есть некоторое **множество кортежей**. А в множестве не может быть двух (или более) одинаковых элементов, так что в отношении не должно быть двух одинаковых кортежей. Если при выполнении проекции у нас получаются одинаковые строки в таблице, их следует «слить» в одну.

ПРОЕКЦИЯ

Преподаватель	Дисциплина	Группа
Миронов	Представление данных	P33682
Сысоева	Основы композиции	P33682
Сысоева	Пластанатомия	P33681
Сопроненко	Основы композиции	P33681
Христолюбова	История искусств	P33681
Христолюбова	История искусств	P33682



Преподаватель	Дисциплина
Миронов	Представление данных
Сысоева	Основы композиции
Сысоева	Пластанатомия
Сопроненко	Основы композиции
Христолюбова	История искусств

Таким образом, при выполнении проекции у нас может уменьшиться не только количество столбцов, но и количество строк таблицы.

В данном примере получившиеся отношения несут в себе информацию о том, у каких групп преподают те или иные преподаватели, и какие дисциплины ведут какие преподаватели.

Преподаватель	Группа
Миронов	P33682
Сысоева	P33682
Сысоева	P33681
Сопроненко	P33681
Христолюбова	P33681
Христолюбова	P33682

Группа	Студент
P33681	Зуев
P33681	Маслова
P33681	Родькин
P33682	Вохмин
P33682	Головин
P33682	Махмудов
P33682	Терехов



КОМПОЗИЦИЯ

Преподаватель	Студент
Миронов	Вохмин
Миронов	Головин
Миронов	Махмудов
Миронов	Терехов
Сысоева	Зуев
Сысоева	Маслова
Сысоева	Родькин
Сысоева	Вохмин
Сысоева	Головин
Сысоева	Махмудов
Сысоева	Терехов
Сопроненко	Зуев
Сопроненко	Маслова
Сопроненко	Родькин
Христолюбова	Зуев

Композиция позволяет из двух отношений получить третье, используя общее множество значений. Для каждого общего значения берутся соответствующие ему значения из первого отношения и из второго, и из них собираются corteжи нового отношения. Например, значению **P33681** соответствуют значения **Сысоева**, **Сопроненко**, **Христолюбова** в первом отношении, и значения **Зуев**, **Маслова**, **Родькин** во втором отношении. Все возможные их сочетания образуют corteжи нового отношения. Так же и со значением **P33682**.

ЕСТЕСТВЕННОЕ СОЕДИНЕНИЕ

Преподаватель	Группа
Миронов	P33682
Сысоева	P33682
Сысоева	P33681
Сопроненко	P33681
Христолюбова	P33681
Христолюбова	P33682

Группа	Студент
P33681	Зуев
P33681	Маслова
P33681	Родькин
P33682	Вохмин
P33682	Головин
P33682	Махмудов
P33682	Терехов



Производится также, как и композиция, но в получившемся отношении также остаётся и столбец общих значений, по которым производилось соединение.

Преподаватель	Группа	Студент
Миронов	P33682	Вохмин
Миронов	P33682	Головин
Миронов	P33682	Махмудов
Миронов	P33682	Терехов
Сысоева	P33681	Зуев
Сысоева	P33681	Маслова
Сысоева	P33681	Родькин
Сысоева	P33682	Вохмин
Сысоева	P33682	Головин
Сысоева	P33682	Махмудов
Сысоева	P33682	Терехов
Сопроненко	P33681	Зуев
Сопроненко	P33681	Маслова
Сопроненко	P33681	Родькин

ДЕКОМПОЗИЦИЯ

Преподаватель	Группа	Студент
Миронов	P33682	Вохмин
Миронов	P33682	Головин
Миронов	P33682	Махмудов
Миронов	P33682	Терехов
Сысоева	P33681	Зуев
Сысоева	P33681	Маслова
Сысоева	P33681	Родькин
Сысоева	P33682	Вохмин
Сысоева	P33682	Головин
Сысоева	P33682	Махмудов
Сысоева	P33682	Терехов
Сопроненко	P33681	Зуев
Сопроненко	P33681	Маслова
Сопроненко	P33681	Родькин

Операция, обратная естественному соединению. Фактически декомпозиция – это выполнение двух проекций.



Декомпозиция бывает с потерей информации, либо без потерь. Правила выполнения декомпозиции без потерь лежат в основе проектирования реляционных БД.

Преподаватель	Группа
Миронов	P33682
Сысоева	P33682
Сысоева	P33681
Сопроненко	P33681
Христолюбова	P33681
Христолюбова	P33682

Группа	Студент
P33681	Зуев
P33681	Маслова
P33681	Родькин
P33682	Вохмин
P33682	Головин
P33682	Махмудов
P33682	Терехов

УНИВЕРСАЛЬНОЕ ОТНОШЕНИЕ

Универсальное отношение – это отношение, которое построено на декартовом произведении всех множеств, рассматриваемых в предметной области. Из этого декартова произведения в отношении оставляются только те кортежи, значения которых содержат в себе корректную по смыслу информацию (то есть то, что действительно есть или может быть в предметной области).

Таким образом, универсальное отношение описывает всю информацию предметной области, которую нужно хранить.

Для наглядности универсальное отношение лучше представлять в виде **таблицы** (реальной или гипотетической). Набор столбцов этой таблицы определяет, какие виды информации нужно хранить. А примеры строк этой таблицы показывают, как значения из разных столбцов могут быть взаимосвязаны.

ПРИМЕР ПРЕДМЕТНОЙ ОБЛАСТИ

Для примера рассмотрим, какая информация могла бы храниться в базе данных для детского сада:

- Фамилия ребёнка
- Имя ребёнка
- Год рождения ребёнка
- Номер группы
- Фамилия воспитателя
- Имя воспитателя
- Фамилия родителя
- Имя родителя
- Телефон родителя

ПРИМЕР УНИВЕРСАЛЬНОГО ОТНОШЕНИЯ

Фамилия ребёнка	Имя ребёнка	Дата рождения ребёнка	Фамилия воспитателя	Имя воспитателя	Номер группы	Фамилия родителя	Имя родителя	Телефон родителя
Петров	Коля	2021-06-08	Прокофьева	Инна	4	Петров	Аркадий	+70127773344
Петров	Коля	2021-06-08	Прокофьева	Инна	4	Петрова	Елена	+79210140307
Иванова	Таня	2021-05-12	Прокофьева	Татьяна	3	Иванов	Пётр	+79005551122
Иванов	Коля	2019-11-25	Игнатенко	Алина	6	Иванов	Пётр	+79005551122
Смирнова	Юля	2019-09-15	Игнатенко	Алина	6	Смирнов	Иван	+79890010123
Смирнова	Юля	2021-06-08	Прокофьева	Инна	4	Смирнова	Елена	+79210140307

В данном примере показано, что у одного ребёнка (Петров Коля) может быть двое родителей, информацию про которых нужно хранить, у родителя (Иванов Пётр) может быть двое детей, кроме того могут быть разные дети с одинаковыми именами (Смирнова Юля), и т.п.

ДОМАШНЕЕ ЗАДАНИЕ

- Выберите предметную область, в которой есть около 10 различных множеств информации (10 свойств каких-то объектов). Это не должны быть свойства одного информационного объекта, иначе база данных получится слишком примитивной. Лучше всего, чтобы это были свойства 3-4 разных объектов (как в примере с детским садом, где есть 4 информационных объекта: ребёнок, родитель, воспитатель, группа).
- Составьте **универсальное отношение** по той предметной области, которую вы выбрали для проектирования учебной базы данных.
- Изобразите универсальное отношение в виде **таблицы** в документе Google Docs ([doc.new](https://docs.google.com/document)), написав **названия столбцов** таблицы так, чтобы по этим названиям было понятно, что в них хранится. **Важно:** это должна быть одна таблица, а не несколько таблиц!
- В случае, если смысл некоторых значений сложно уместить в коротком названии столбца, добавьте примечания снаружи таблицы, описав специфические особенности информации.
- Заполните отношение **примерами** кортежей (примеры **строк** таблицы), чтобы по этим примерам было понятнее, какая информация может встречаться, и как могут сочетаться значения различных столбцов.