Кафедра Прикладной математики
Института информационных технологий
РТУ МИРЭА

Дисциплина «Большие данные»

2022-2023 у.г.

Лекция 3. Технологии сбора информации и больших объемов данных

Часть 1. Структурированные и неструктурированные данные

Материалы

1. Объектно-ориентированный подход к хранению данных
2. Понятие структуры данных.
3. Структурированные данные
4. Пример структурированных данных
5. Неструктурированные данные
6. Пример неструктурированных данных
7. Методы структуризации данных
8. Примеры

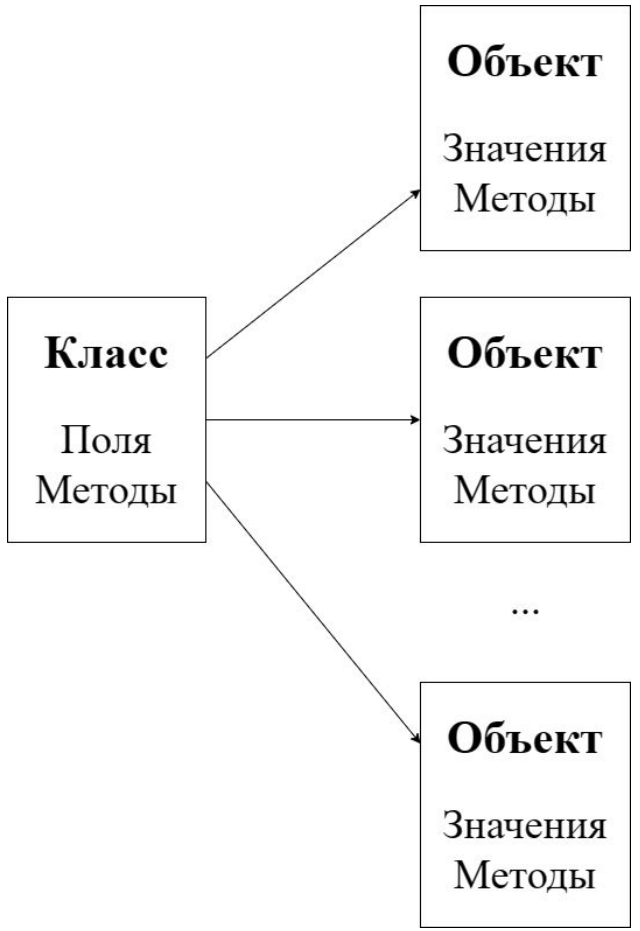
Объектно-ориентированный подход к хранению данных

В объектно-ориентированном подходе (ООП) все сущности формализуются набором **полей** и **методов.**

Поля характеризуют параметры сущности, а **методы** возможности воздействия на процесс и другие объекты.

В хранении данных принято хранить состояние об объекте или процессе в виде набора значений его характеристик, которые можно назвать **полем** объекта.

Объектный подход распространяется как на парадигму программирования, так и на системы хранения данных.



Понятие структуры данных

Структура данных в обработке и хранении данных это **перечень полей и их типов данных**, которыми представлена структурированная таблица данных.

На основе структуры данных можно проектировать сценарии обработки данных без наличия непосредственно записей в таблице (выборок).

Структура данных является отражением объектно-ориентированного подхода в обработке данных, и связано с понятием **структурированных данных**

Выходные	Имя	Вид данных
12 Id позиций в чеке	order_details_id	Непрерывный
12 Id чека	order_id	Непрерывный
ab Id Пиццы	pizza_id	Дискретный
12 Количество	quantity	Непрерывный
31 Дата заказа	order_date	Непрерывный
31 Время заказа	order_time	Непрерывный
9,0 Цена за штуку	unit_price	Непрерывный
9,0 Полная стоимость	total_price	Непрерывный
ab Размер пиццы	pizza_size	Дискретный
ab Категория пиццы	pizza_category	Дискретный
ab Ингридиенты	pizza_ingredients	Дискретный
ab Название пиццы	pizza_name	Дискретный

Рисунок. Структура данных заказов в пиццерии

Структурированные данные

Структурированными называются данные, отражающие отдельные факты предметной области и упорядоченные определенным образом с целью обеспечения возможности применения к ним различных методов обработки.

В случае таблиц данных подразумевается, что данные упорядочены по вертикали в типизированные столбцы, называемые **полями**, а по горизонтали — в строки, называемые **записями**.

ID Клиента	Фамилия	Имя	Отчество	Пол
1	Иванов	Иван	Иванович	М
2	Иванова	Людмила	Андреевна	Ж
3	Сидоров	Андрей	Анатольевич	М
4	Сидорова	Юлия	Ивановна	Ж
5	Петров	Аркадий	Алексеевич	М
6	Петрова	Анна	Александровна	Ж

Пример структурированных данных

#	ab Код страны	ab Направление торговли	31 Месяц и г...	12 Код това...	ab Единица измерения	12 Суммарный объем	12 Масса нетто	12 Количество товаров	12 Код ОКАТО России регио...
1	MY	ИМ	01.01.2016	8482109008	ШТ	443	30	72	40000
2	IT	ИМ	01.01.2016	6204695000	ШТ	131	1	7	46000
3	CN	ИМ	01.01.2016	9001900009	1	112750	18	0	46000
4	BY	ИМ	01.01.2016	8414302004	ШТ	392	57	8	50000
5	US	ИМ	01.01.2016	9018509000	1	54349	179	0	40000
6	EE	ИМ	01.01.2016	9021101000	1	17304	372	0	46000
7	FR	ИМ	01.01.2016	3816000000	1	323488	253600	0	40000
8	MX	ИМ	01.01.2016	8523519300	ШТ	1611	0	4	40000
9	JP	ИМ	01.01.2016	6204520000	ШТ	29	1	2	46000
10	KR	ИМ	01.01.2016	6110209100	ШТ	815	2	5	46000
11	KG	ИМ	01.01.2016	8527139900	ШТ	11868	2127	2630	46000
12	ZA	ИМ	01.01.2016	8421230000	ШТ	12686	1785	3451	45000
13	CN	ИМ	01.01.2016	8518109500	ШТ	12	0	10	65000
14	TR	ИМ	01.01.2016	8417900000	ШТ	206453	17297	1	92000
15	IT	ИМ	01.01.2016	3906100000	1	4492	1075	0	45000
16	CZ	ИМ	01.01.2016	8708409909	1	41	2	0	46000
17	ES	ИМ	01.01.2016	6404191000	ПАР	11822	346	760	45000
18	IT	ИМ	01.01.2016	9404909000	1	6801	485	0	46000
19	UA	ИМ	01.01.2016	8207801900	1	35793	1020	0	14000
20	CN	ИМ	01.01.2016	3304100000	1	59678	10829	0	46000
21	SI	ИМ	01.01.2016	6104440000	ШТ	1470	13	15	45000
22	CZ	ИМ	01.01.2016	6402121000	ПАР	5375	263	63	45000
23	CN	ИМ	01.01.2016	8532250000	1	5092	1953	0	5000
24	CN	ИМ	01.01.2016	7318220009	1	42770	55628	0	22000
25	FR	ИМ	01.01.2016	2204214400	Л	191	26	18	86000

Временные ряды

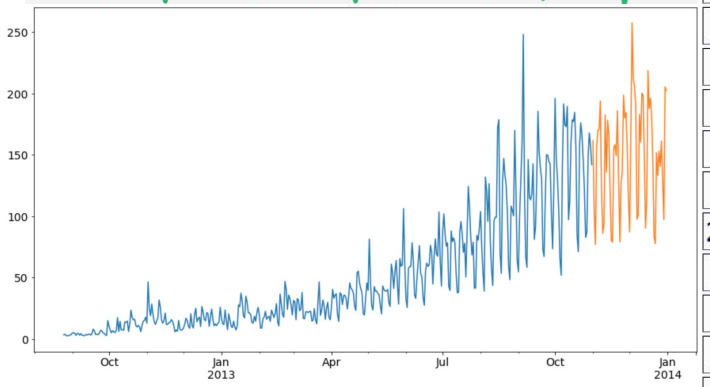
Измерения показателя во времени для одного обособленного объекта.
Содержат зависимые от времени и последовательности измерений данные о показателе выбранного объекта.

Таблица хранит:

- Временные метки (дата/время)
- Значения показателей (целочисленные или вещественные)

Применение:

- Экономика
- Геология
- Прикладная физика
- Химия
- ...



T	IP2_EA_M	IP2_EA_M_SA
	2014.1=100	2014.1=100
2014 1	100,0	106,1
2	98,7	106,5
3	107,2	106,7
4	104,7	106,9
5	104,0	107
6	104,5	106,9
7	105,7	106,7
8	105,7	106,3
9	107,0	106,1
10	112,9	106
11	111,7	106,1
12	122,6	106,3
2015 1	101,3	106,6
2	98,5	106,8
3	109,2	106,9
4	104,1	106,9
5	102,9	106,9
6	104,6	106,9
7	105,9	106,9
8	106,5	106,9
9	107,5	106,8

Данные транзакций

Транзакционные данные — это любая информация, которая связана с транзакциями.
Ключевое отличие транзакционных данных от других типов — это фиксация даты и времени.

Показатели не зависят друг от друга в последовательности
Также сохраняется

- вид платежа,
- продукт,
- количество покупок,
- применяемые скидки и промокоды,

Учитывается поведение клиентов до и после конверсии.

Дата транзакц...	Магаз...	Чек	Клие...	Товар	Кол-...	Сумма продажи
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006110	1,00	57,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006114	1,00	49,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006115	2,00	100,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006130	1,00	38,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006131	1,00	86,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006134	2,00	155,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006144	1,00	51,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku006148	3,00	125,00
01.04.2017, 00:00	Store01	code00350356	cl100636	sku027910	1,00	11,00
01.04.2017, 00:00	Store01	code00749858	cl100636	sku025594	1,00	549,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku030449	1,00	355,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku013453	1,00	518,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku007369	2,00	387,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku012298	1,00	2 241,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku029092	1,00	258,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku011035	2,00	134,00
01.04.2017, 00:00	Store01	code00174677	cl089338	sku011186	1,00	1 846,00

Данные объектов

Таблицы данных объектов используются для хранения и извлечения больших двоичных объектов, например изображений, текстовых файлов, видео- и аудиопотоков, объектов данных и документов приложений большого размера.

Объект состоит из сохраненных данных, метаданных и уникального идентификатора доступа к объекту. Хранилища объектов поддерживают отдельные большие файлы, а также позволяют управлять всеми файлами.

path	blob	metadata
/delays/2017/06/01/flights.cscv	0XAABBCCDDEEF...	{created: 2017-06-02}
/delays/2017/06/02/flights.cscv	0XAADDCCDDEEF...	{created: 2017-06-03}
/delays/2017/06/03/flights.cscv	0XAEBBDEDDEEF...	{created: 2017-06-03}

Полуструктурированные данные

```
[
    {
        "id": 21034,
        "Имя": "Ярослав",
        "Возраст": 23,
        "Рост": 182,
        "Вес": 75.67
    },
    {
        "id": 84124,
        "Имя": "Лидия",
        "Возраст": 31,
        "Рост": 165,
        "Вес": 54.52
    },
    {
        "id": 38194,
        "Имя": "Петр",
        "Возраст": 44,
        "Рост": 192,
        "Вес": 94.31
    },
    {
        "id": 15152,
```

JSON-LD

	A	B	C	D	E
1	**ID**	**Имя**	**Возраст**	**Рост**	**Вес**
2	21034	Ярослав	23	182	75,67
3	84124	Лидия	31	165	54,52
4	38194	Петр	44	192	94,31

XLS

```
id, Имя, Возраст, Рост, Вес
21034, Ярослав, 23, 182, 75.67
84124, Лидия, 31, 165, 54.52
38194, Петр, 44, 192, 94.31
```

CSV

```
<DATA>
    <People>
        <Person
        id="21034"
        Имя="Ярослав"
        Возраст="23"
        Рост="182"
        Вес="75.67"
        />
        <Person
        id="84124"
        Имя="Лидия"
        Возраст="31"
        Рост="165"
        Вес="54.52"
        />
        <Person
        id="38194"
        Имя="Петр"
        Возраст="44"
        Рост="192"
        Вес="94.31"
        />
    </People>
</DATA>
```

XML

Данные документов

Таблица данных документов управляет набором значений-документов.
Обычно данные в этих хранилищах содержатся в виде документов JSON.

Каждое значение поля документа может представлять собой скалярный элемент, например число, или сложный объект, например список или коллекция типа "родитель — потомок".

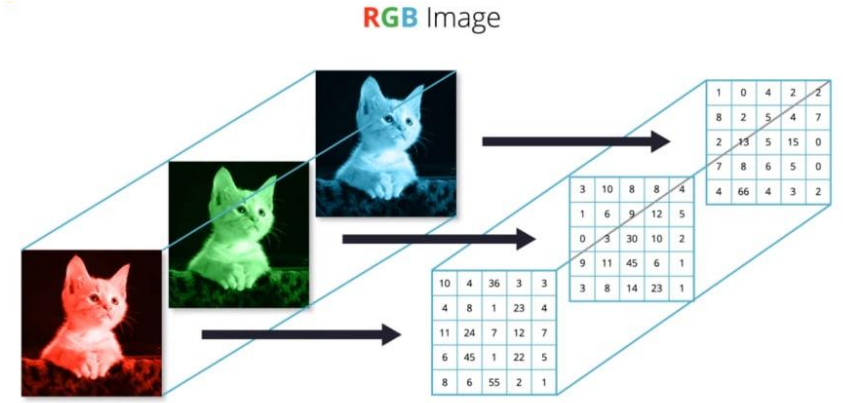
Данные в полях документа можно закодировать разными способами, например в формате XML, YAML, JSON, или хранить в виде обычного текста.
Приложение может получать документы по ключу документа.

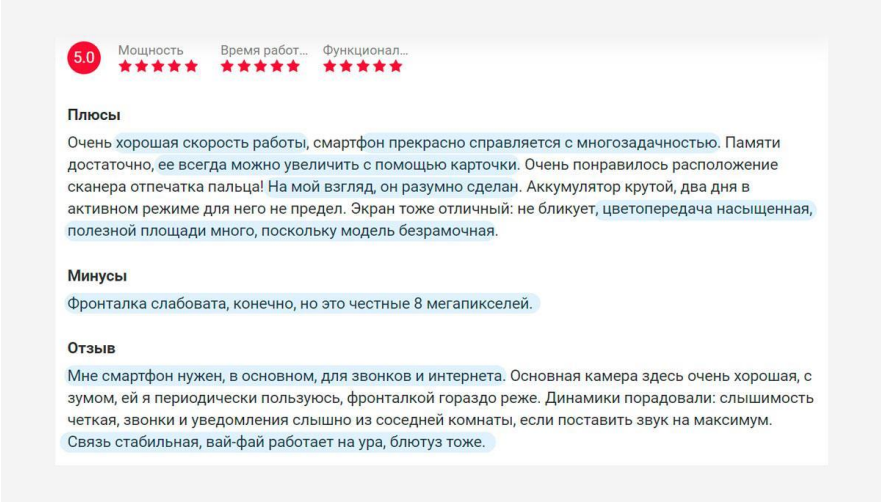
Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

Неструктурированные данные

Неструктурированные данные — данные, которые не соответствуют заранее определённой модели данных, и, как правило, представлены в форме текста с датами, цифрами, фактами, расположенными в нём в произвольной форме.

Такие техники, как интеллектуальный анализ данных (data mining), обработка естественного языка (Natural Language Processing) и интеллектуальный анализ текста, предоставляют методы поиска закономерностей с целью так или иначе интерпретировать неструктурированную информацию.





Неструктурированные данные

Структурированные данные	Неструктурированные данные
Организованная, типизированная информация, относящаяся к одной сущности	Не имеет предопределенной организации и имеет множество форм
Количественная	Качественная
Долговременное хранилище данных, реляционные базы данных	Озёра данных, файловые базы данных
Несколько предопределенных форматов	Большое количество различных форматов данных

Пример неструктурированных данных

Пассивный вис

Повисните на турнике (или дверном косяке, или ветке дерева). Вот и все. Это расслабленный вис, в отличие от обычного виса из главы про подтягивания. Не напрягайте плечи и отводите назад лопатки. Идея заключается в том, чтобы ваш позвоночник максимально вытянулся, а широчайшие мышцы спины и плечи были расслаблены. Даже ваши вертлужные впадины почувствуют это. Ах-х-х-х-х.

Текстовая информация

Фото и видео

Методы структуризации данных

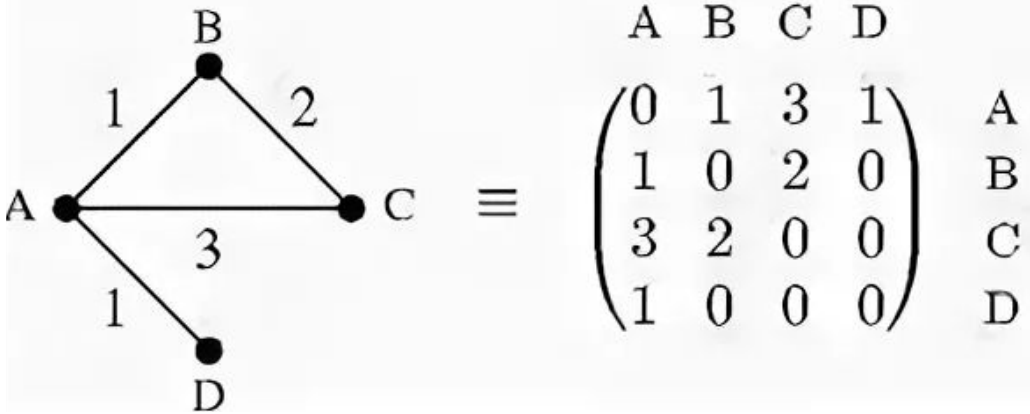
Структуризация данных рассматривается как отдельный механизм преобразования неструктурированных данных в удобный для обработки данных вид информации.

Структуризация данных доступна для таких данных как **текстовые** данные и **графовые** данные.

Данные структуризации не обладают достаточной эффективностью хранения и обработки.

	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9



Часть 2. Шкалы данных. Обработка шкал данных. Вид данных

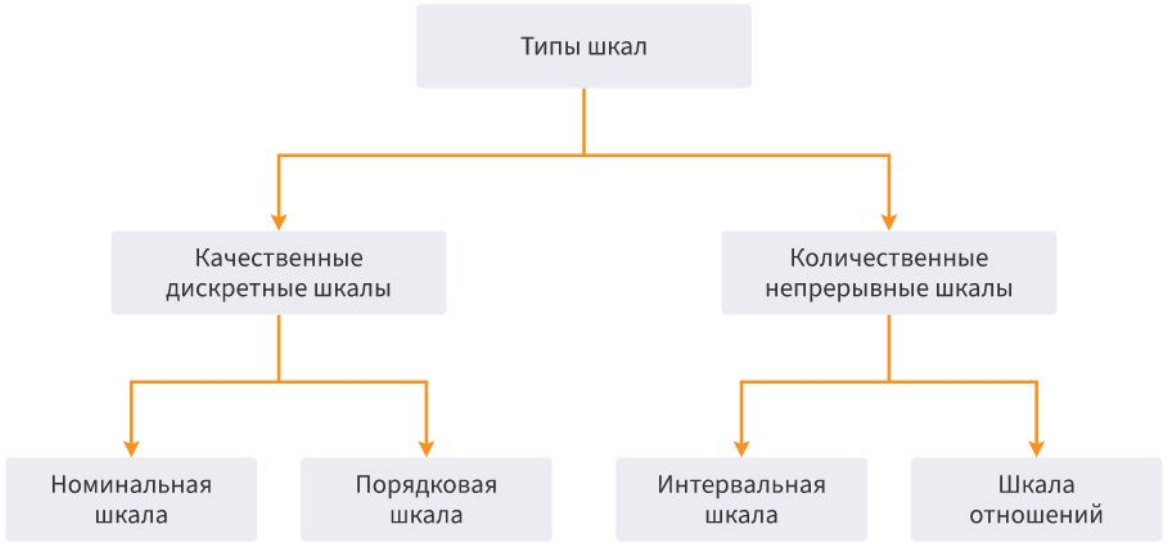
Понятие шкал структурированных данных

Шкала измерения в статистике — это способ представления переменных и их группировки в различные категории.

Она определяет характер значений, присвоенных переменным в наборе данных.

Номинальная и **порядковая**х шкалы — измерение качественных данных (категории).

Интервальная и шкала **отношений** — измерение количественных данных.



Шкалы измерений, классификация

Основными свойствами шкал измерений являются:

1. Идентифицируемость
2. Величина
3. Равенство интервалов
4. Абсолютный ноль

Уровни измерений данных

1. Номинальная шкала (категориальная, наименований)
2. Порядковая шкала (ординальная, ранговая)
3. Интервальная шкала (разностей)
4. Шкала отношений (абсолютная)

Свойства \ Тип шкалы	Номинальная	Порядковая	Интервальная	Отношений
Идентифицируемость	●	●	●	●
Величина (магнитуда)		●	●	●
Равенство интервалов			●	●
Абсолютный ноль				●

Номинальная шкала

Номинальная шкала: описание групп статистик, подписи визуализации.

Отражают те или иные свойства объекта, выраженные словесно.
Их элементы могут только совпадать или не совпадать друг другом, Их нельзя сопоставлять по принципу «больше-меньше».
Недопустимы также и арифметические действия.

Характерным примером может служить группа крови.

Мерой среднего может служить **мода**.

Номинальная шкала

Порядковая шкала

Порядковая шкала: то же, что и номинальная шкала и расчет квантилей, исследование градации оценки качества.

По ней можно ранжировать и сравнивать объекты, по какому – либо признаку.

Мерой среднего может служить **медиана**.

Порядковая шкала

Интервальная шкала

Интервальная шкала: сравнение с эталоном, линейное преобразование (сдвиг), сложение и вычитание.

Является метрической шкалой.

Мерой среднего может являться среднее арифметическое.

Пример: шкала Цельсия, измерение времени, широта и долгота.

Шкала отношений

Шкала отношений: присутствует дополнительное свойство — естественное и однозначное присутствие нулевой точки

Точкой начала отсчета является точка, в которой значение параметра равно нулю. Появляется возможность отсчитывать от нее абсолютное значение параметра, определять разницы значений и во сколько раз одно больше другого.

Присутствуют операции сложения, вычитания, умножения, деления и наличие абсолютного нуля.

Дискретные данные

По характеру варьирования переменные делятся на дискретные и нерперывные.

Дискретные данные являются значениями признака, общее число которых конечной или бесконечно, но может быть подсчитано при помощи натруральных чисел.

С дискретными данннными не могут быть произведены никакие арифметические действия, либо они не имеют смысла.

Дискретными данными являются все данные строкового и бинарного типа. Примеры: код товара, образование, город, тип скидки, пол, категория.

Непрерывные данные

Непрерывные данные – это данные, которые могут принимать любые значения в некотором интервале. Над непрерывными данными можно производить арифметические операции: сложение, вычитание, умножение и деление, и они имеют смысл.

Примеры: возраст, рост, стоимость, количество.

Тип данных	Вид данных	
	Непрерывный	Дискретный
Числовой	+	+
Строковый		+
Логический		+
Дата/время	+	+

Часть 3. Хранение информации в виде структурированных данных. Реляционная модель данных

Материалы

1. Структура данных как шаблон
2. Поля данных, домены, записи
3. Записи как экземпляры класса
4. Уникальность записи в таблице
5. Реляционная алгебра
6. **Хранение информации в виде таблиц**
7. **Реляционные базы данных**
8. **Системы управления базами данных**
9. **Понятие схемы данных**
10. Нормальные формы базы данных
11. **Доступ к данным в реляционных СУБД**
12. **Схема на чтение, схема на запись**

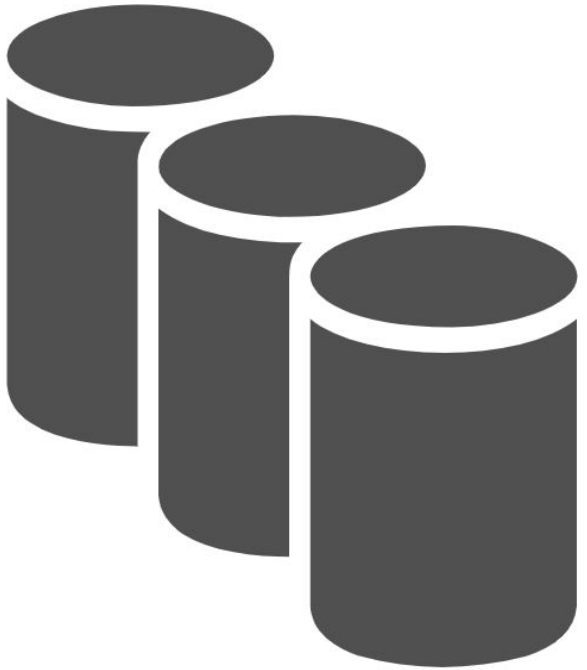
Базы данных

База данных (БД) – это совокупность данных, хранящихся и упорядоченных в соответствии с определенной **структурой.**

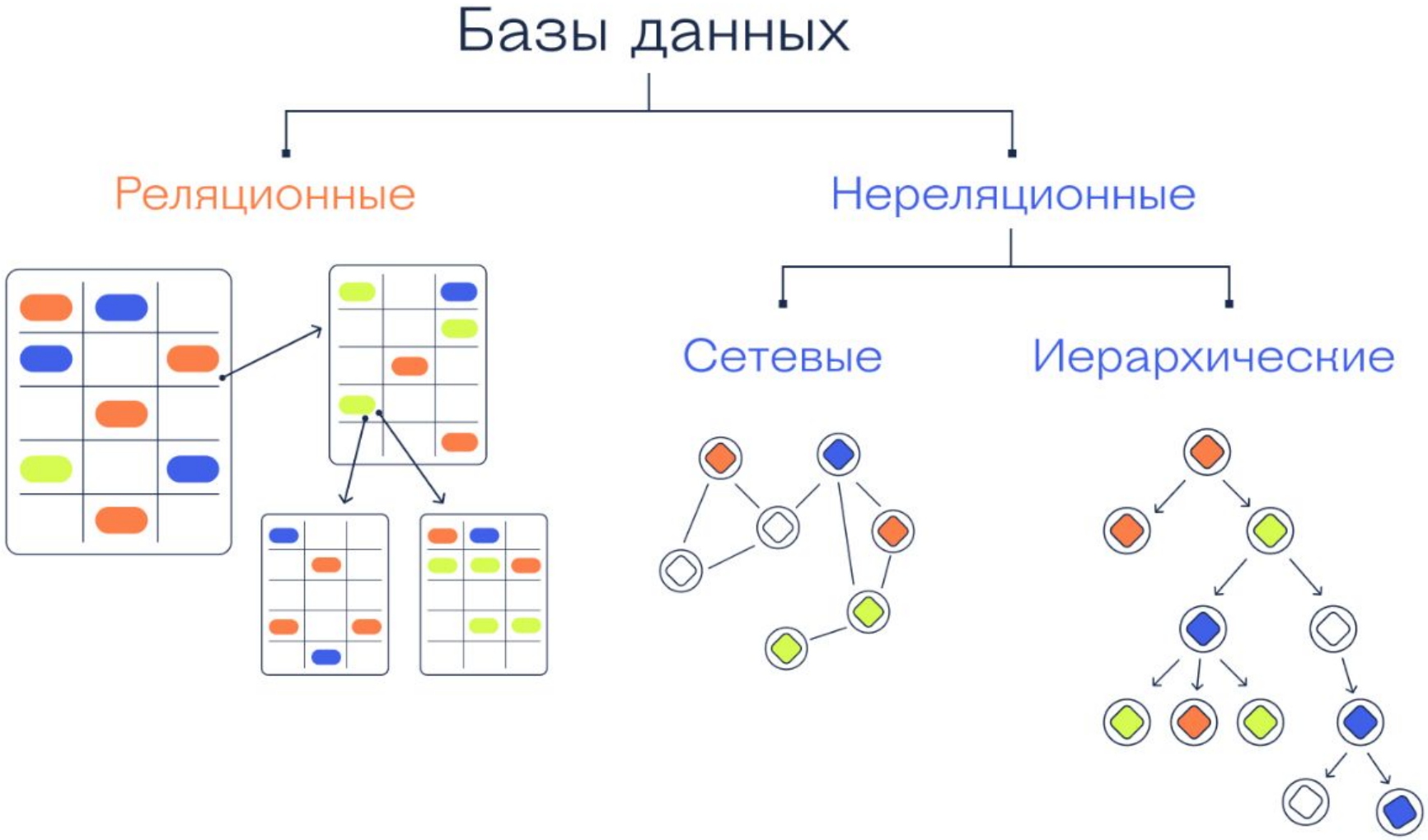
Модель данных определяет то, как и каким образом данные будут **располагаться** в БД и как к ним будет **предоставляться доступ.**

Если проще, то БД это просто информация с которой мы работаем.

С базой данных нельзя полноценно взаимодействовать не используя систему управления базами данных.



Модели данных

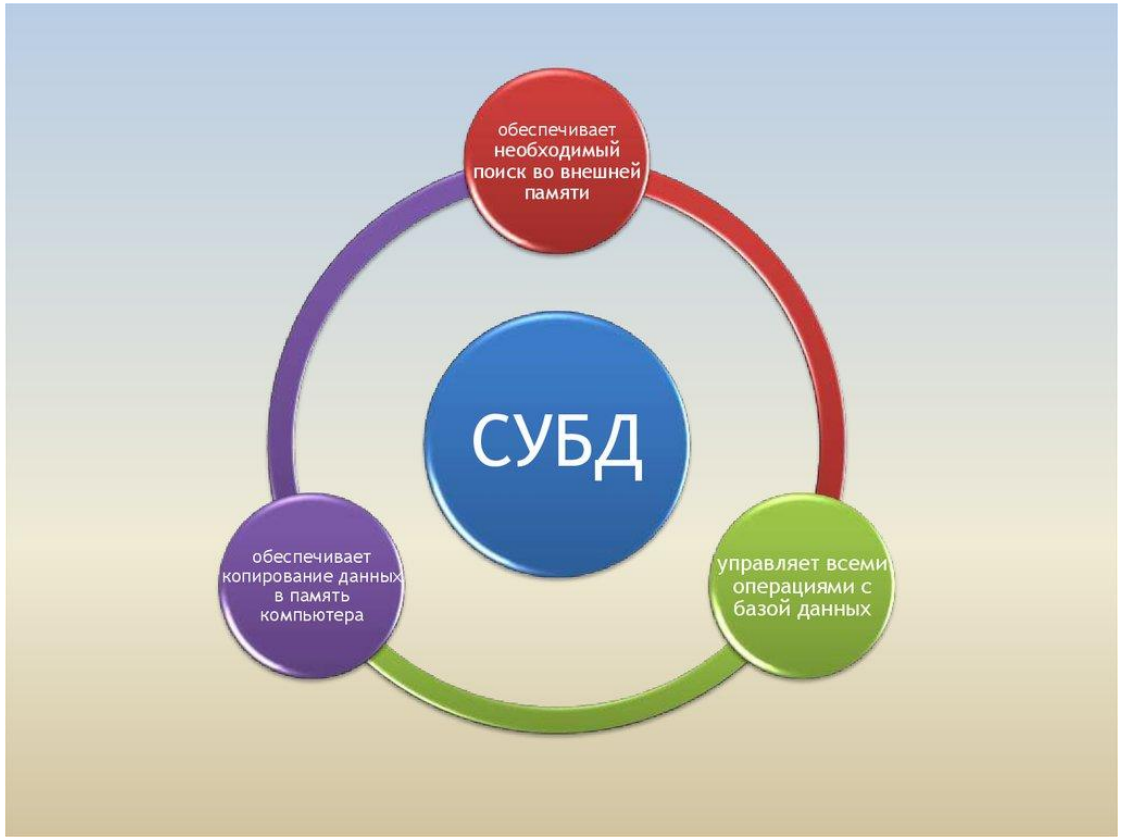


Системы управления базами данных

Базу данных невозможно было бы изменить или заполнить не будь **системы для её управления**

Система управления базами данных (СУБД) представляет из себя совокупность программных и языковых средств для создания, удаления, изменения и любых других манипуляций с данными в БД.

СУБД работает в соответствии со **структурой**, на которой строится база данных



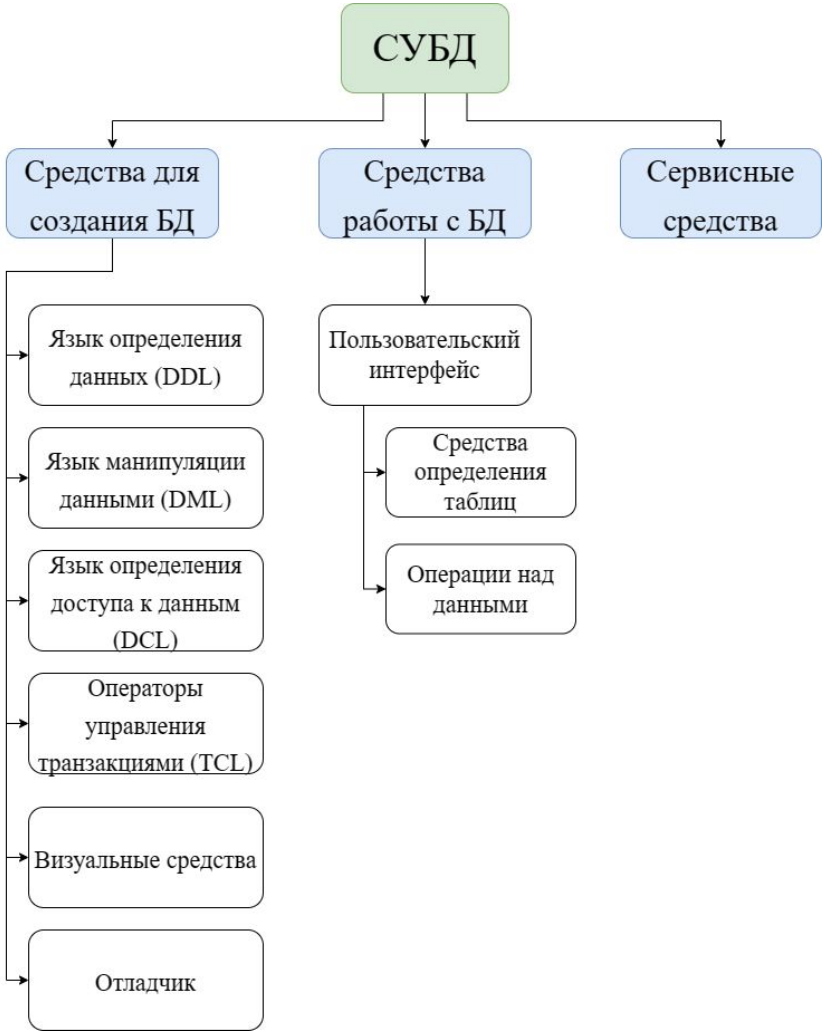
Функции СУБД

Все манипуляции с базой данных и с данными происходят через СУБД

Основными функциями СУБД являются:

- Управление данными во внешней памяти
- Управление буферами оперативной памяти
- Поддержка языков базы данных
- Журнализация и резервное копирование базы данных

Для манипуляций над данными в реляционных СУБД используют декларативный язык запросов SQL.



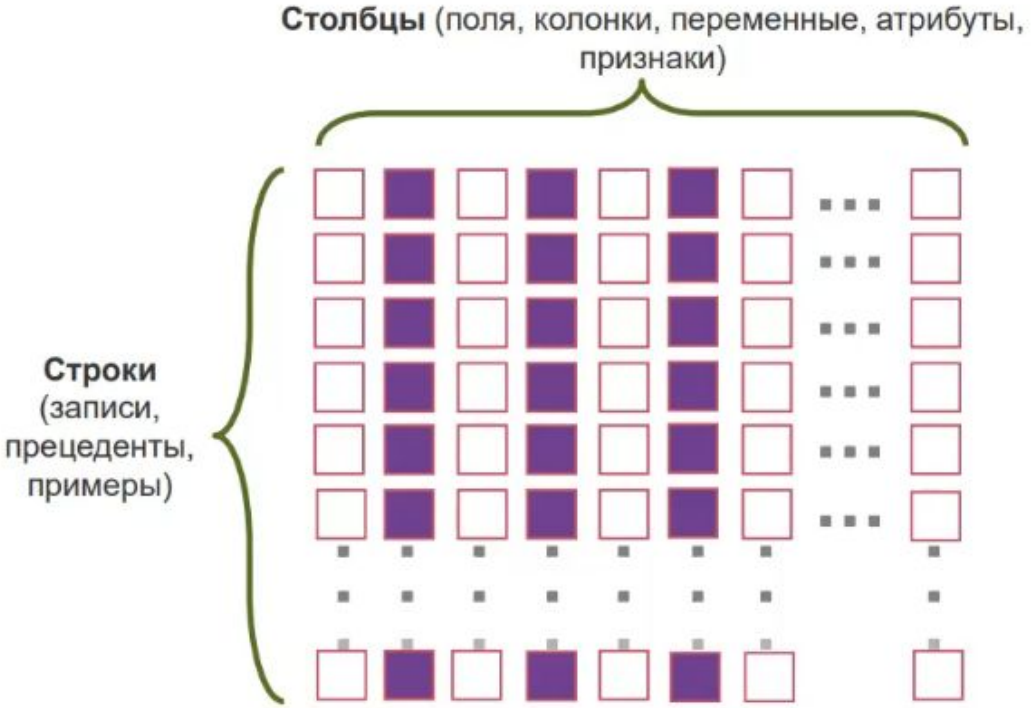
Реляционная база данных

В реляционной БД вся информация хранится в **таблицах**, состоящих из **столбцов** и **строк.**

Столбцы – это атрибуты или характеристики объекта

Каждая **строка** хранит данные об отдельном объекте.

Все строки одной таблицы имеют одинаковую структуру и состоят из ячеек, содержащих описание того или иного атрибута объекта.



Пример таблицы данных

Таблица, хранящая данные об автолюбителях, имеет следующие **атрибуты** (столбцы):

имя: строковый тип,
фамилия: строковый тип,
возраст: числовой тип,
профессия: строковый тип,
дата покупки: дата,
автомобиль: строковый тип

А также добавим в нее данные.

FirstName	SecondName	Age	Profession	DateStart	Car
Виктор	Межневский	23	Прораб	12-01-2020	BMW X3
Мира	Лирина	27	Врач	04-04-2019	Renault Captur
Игорь	Свирин	32	Слесарь	21-11-2013	Lada Vesta

Ключи первичный и внешний

Первичный ключ (PRIMARY key) – уникальный атрибут, идентифицирующий отдельную запись таблицы данных.

Первичные ключи нельзя менять. Первичным ключом может выступать как **число** так и **строка**.

Вторичный ключ (FOREIGN key) – уникальный атрибут внешней таблицы, создающий связь с данной по совпадающим значениям в столбце.

ID	FirstName	SecondName	Age	Profession	DateStart	Car
1	Виктор	Межневский	23	Прораб	12-01-2020	BMW X3
2	Мира	Лирина	27	Врач	04-04-2019	Renault Captur
3	Игорь	Свирин	32	Слесарь	21-11-2013	Lada Vesta

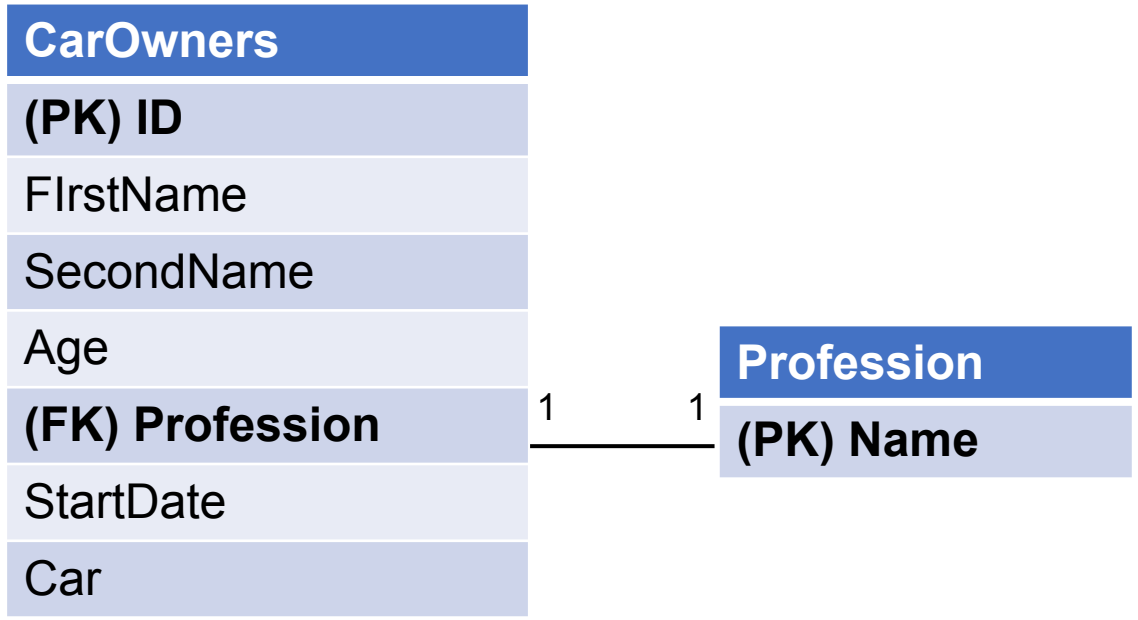
Car	Length	Width	Weight
BMW X3	4708	1891	1900
Renault Captur	4122	1778	1350
Lada Granta	3926	1700	1120
Lada Vesta	4410	1764	1300

Связь один к одному

Связи между таблицами бывают следующих видов:

- один к одному,
- один ко многим,
- многие ко многим.

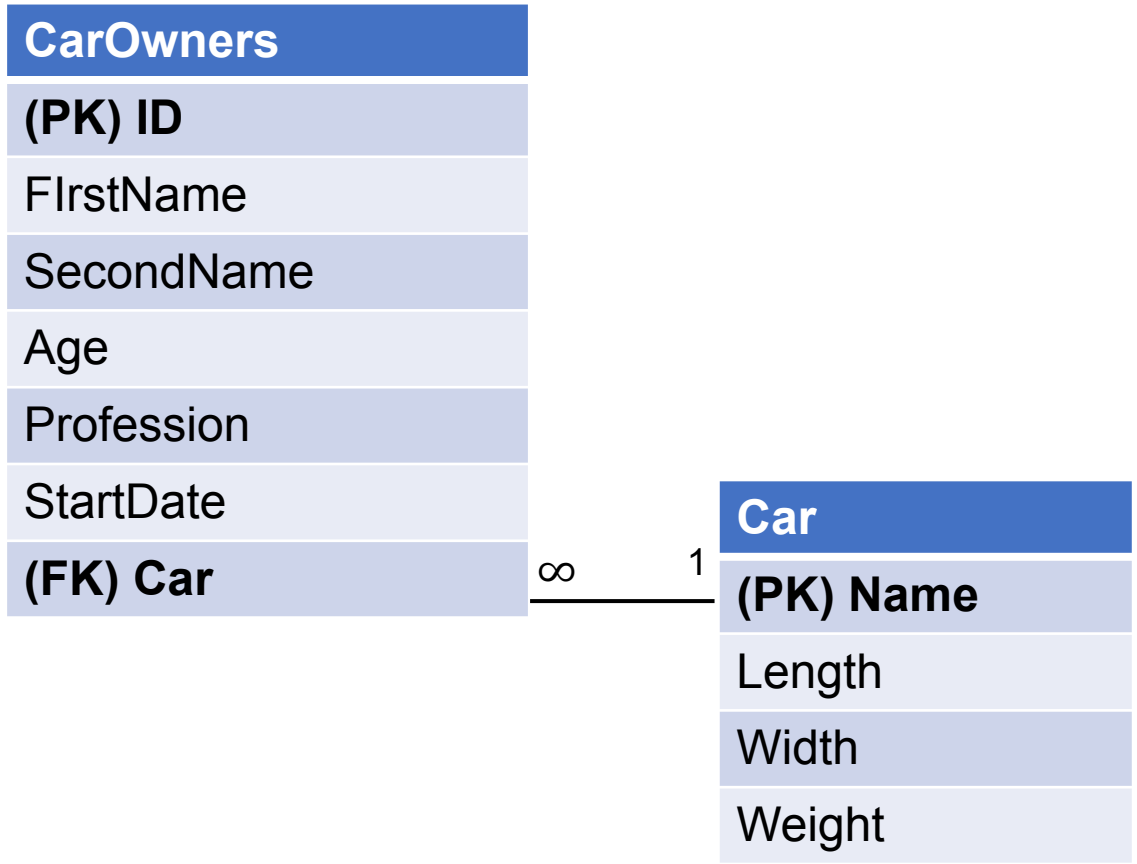
Связь **один к одному** подразумевает, что **один** объект (строка) первой таблицы зависит от **одного** объекта второй таблицы и наоборот.



Связь один ко многим

Связь один ко многим – связь при которой одна строка первой таблицы относится к нескольким строкам (нескольким объектам) второй таблицы, а одна строка второй таблицы относится к одной строке (одному объекту) первой.

Пример, использованный ранее.

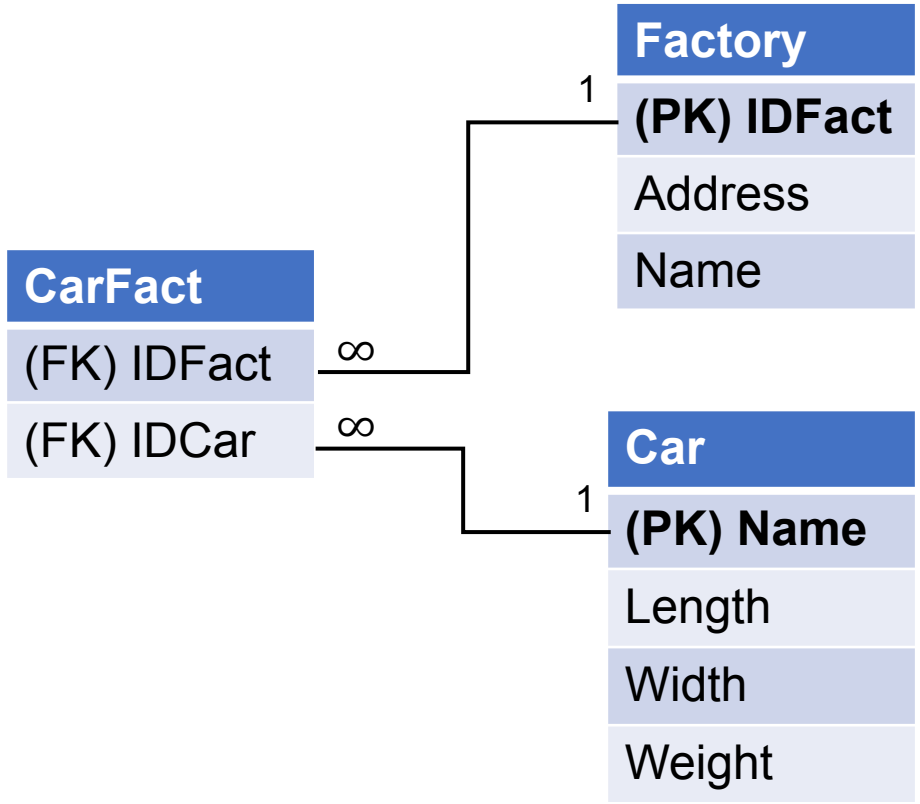


Связь многие ко многим

Связь многие ко многим. «**Один** объект первой таблицы зависит от **нескольких** объектов второй таблицы и **один** объект второй таблицы зависит от **нескольких** объектов первой таблицы».

Таблицы, участвующие в связи:

- Две основных
- Одна связующая, хранит два вторичных ключа



Понятие схемы данных

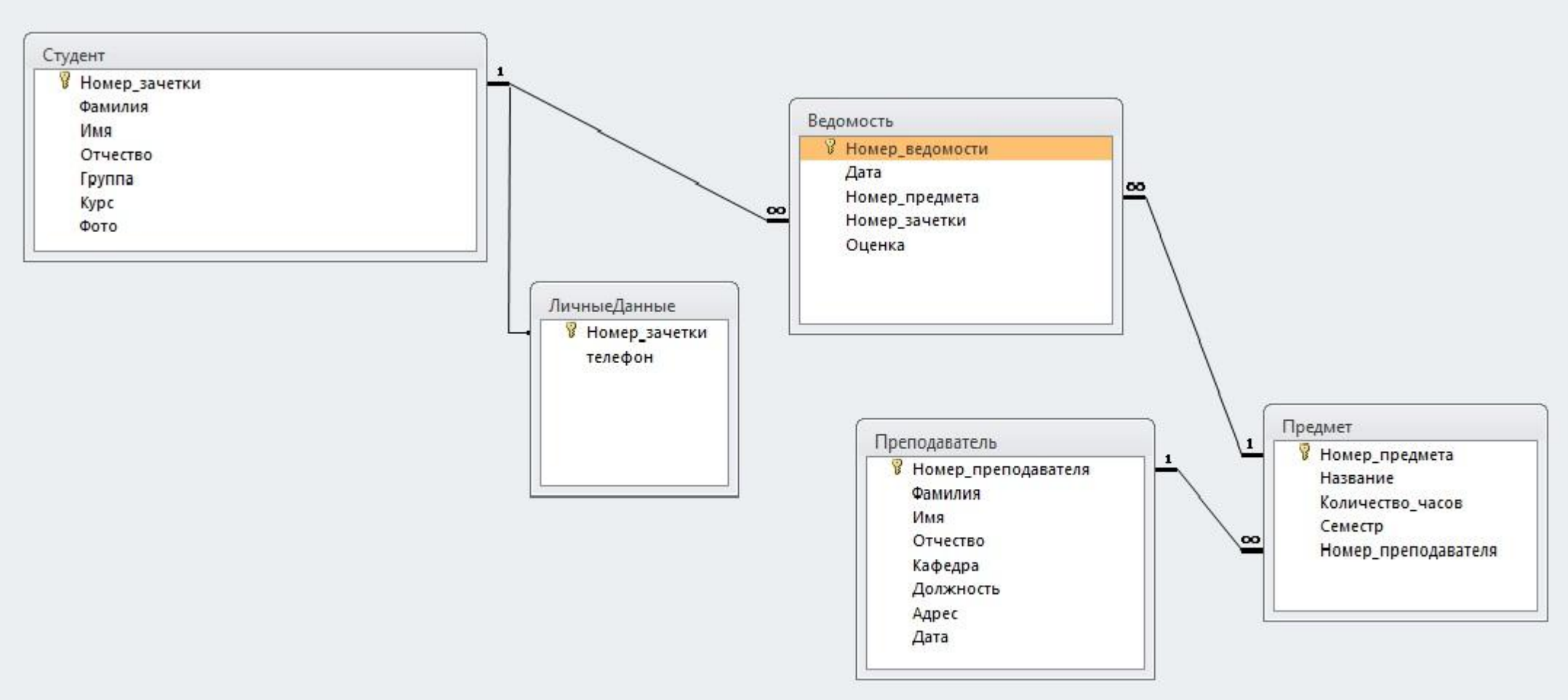
В использованных ранее рисунках с иллюстрациями связей таблиц мы использовали наглядный инструмент отображения схем таблиц.

Схема БД – список таблиц, их атрибутов, типов данных, ограничений, ключей и связей между таблицами, необходимый для корректной организации хранения данных в памяти вычислительного устройства и доступа к данным извне, как на запись, так и на чтение.

Схема БД является удобным инструментом унификации доступа к данным и помогает ускорять доступ к информации по сравнению с другими моделями хранения данных.

Также корректная схема и организация ограничений и связей таблиц помогают обеспечить высокую отказоустойчивость и целостность хранилища данных на автоматическом уровне.

Пример схемы РБД



Доступ к данным в реляционных СУБД

Доступ к данным в **РСУБД** классически осуществляется с помощью языка DML, подязыка SQL.

Функции языков DML определяются первым словом в предложении (часто называемом запросом), которое почти всегда является глаголом. В случае с SQL эти глаголы — «select» («выбрать»), «insert» («вставить»), «update» («обновить»), и «delete» («удалить»).

Языки DML могут несущественно различаться у различных производителей СУБД.

```sql
SELECT * FROM table_name;

Select * from Apps

select count(*) from Apps

select * from Apps where AppName = 'MoneyControl'

select AppName, AppCategory from Apps

select distinct AppName from Apps

select AppName from Apps where AppPrice > 60
```

Доступ к данным в реляционных СУБД

Доступ к данным в **РСУБД** также может осуществляться посредством ODBC (контроллер базы данных) или API (прикладной интерфейс программы).

В прикладных пакетах анализа данных существуют возможности быстрого доступа к данным таблиц базы данных за счет разработанных библиотек, компонентов и утилит.

```python
# Импорт модулей для работы с базами и таблицами
import sqlite3
import pandas as pd
# Создание подключения к sqlite3 базе данных
cnx = sqlite3.connect('file.db')
# В df находятся данные от запроса к базе
df = pd.read_sql_query("SELECT * FROM table_name", cnx)
```

Рисунок. Подключение к базе данных в Python

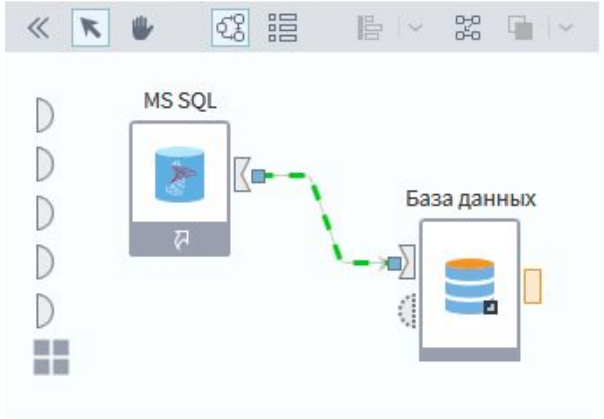


Рисунок. Подключение к базе данных в Loginom

Доступ к данным в реляционных СУБД

Пример выборки таблицы данных на языке DML для приведенной таблицы **car_users**.

```
SELECT FirstName,
       SecondName,
       Age,
       Profession,
       DateStart,
       Car
FROM car_users;
```

FirstName	SecondName	Age	Profession	DateStart	Car
Виктор	Межневский	23	Прораб	12-01-2020	BMW X3
Мира	Лирина	27	Врач	04-04-2019	Renault Captur
Игорь	Свирин	32	Слесарь	21-11-2013	Lada Vesta

Обратите внимание на то, что выборка данных происходит поколоночно

Доступ к данным в реляционных СУБД

Оператор SELECT состоит из нескольких предложений (разделов):

- **SELECT** определяет список возвращаемых столбцов (как существующих, так и вычисляемых), их имена, ограничения на уникальность строк в возвращаемом наборе, ограничения на количество строк в возвращаемом наборе;
- **FROM** задаёт табличное выражение, которое определяет базовый набор данных для применения операций, определяемых в других предложениях оператора;
- **WHERE** задает ограничение на строки табличного выражения из предложения FROM;
- **GROUP BY** объединяет ряды, имеющие одинаковое свойство с применением агрегатных функций
- **HAVING** выбирает среди групп, определённых параметром GROUP BY
- **ORDER BY** задает критерии сортировки строк; отсортированные строки передаются в точку вызова.

Доступ к данным в реляционных СУБД

Оператор SELECT имеет следующую структуру:

```
SELECT
      [DISTINCT | DISTINCTROW | ALL]
      select_expression,...
FROM    table_references
[WHERE      where_definition]
[GROUP BY  {unsigned_integer | col_name | formula}]
[HAVING     where_definition]
[ORDER BY  {unsigned_integer | col_name | formula} [ASC | DESC], ...]
```

Часть 4. Внесение данных в РБД. Транзакции в РБД

Добавление информации в базу данных

Операторы, отвечающие за внесение изменений в наполнение реляционной базы данных находятся в языке DML.
Операторы манипуляции данными:

- INSERT добавляет новые данные,
- UPDATE изменяет существующие данные,
- DELETE удаляет данные;

Данные операторы влияют на хранящиеся экземпляры объектов в РБД, собственно данные в базе данных.

```
INSERT INTO Persons(ID, FirstName, LastName, Department)
VALUES (1, 'Anna', 'Klimenok', 'QA'),
       (2, 'Olga', 'Chekan', 'QA'),
       (3, 'Olga', 'Naumik', 'QA'),
       (4, 'Alexey', NULL, 'TC'),
       (5, 'Oleg', NULL, 'TC'),
       (6, 'Sergey', 'Pavlov', 'DV');
```

```
DELETE FROM table_name
 WHERE column_name = some_value ;
```

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

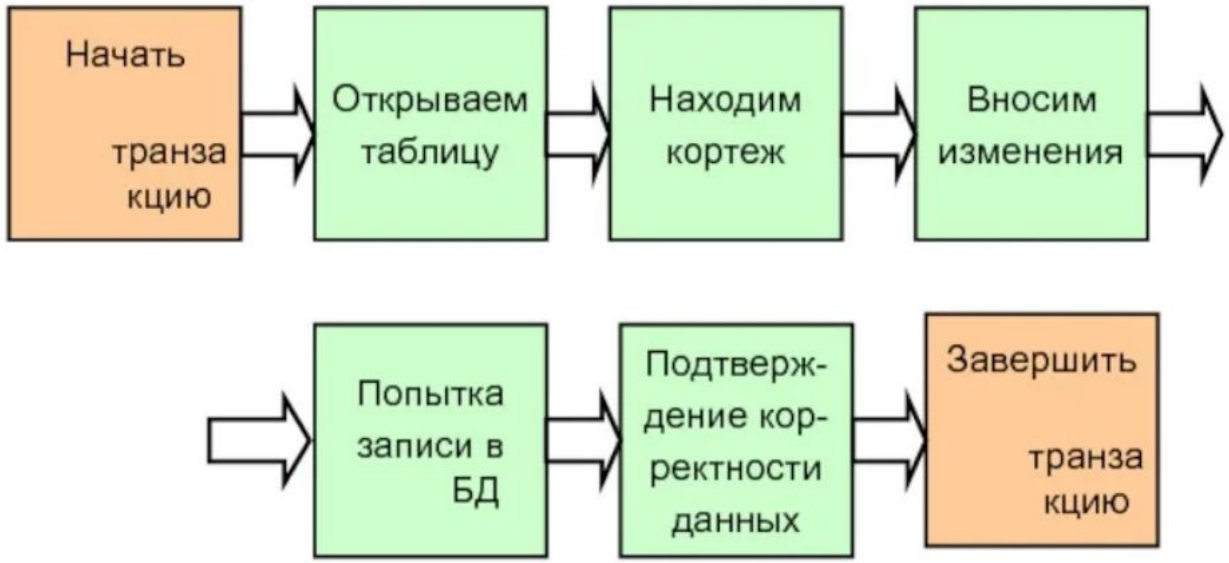
Транзакции в базу данных

Изменения в базе данных, переводящие её из одного согласованного состояния в другое производятся с использованием механизма **транзакций.**

Транзакция – группа операторов определения, манипуляции данных, переводящих базу данных из одного согласованного состояния в другое согласованное состояние.

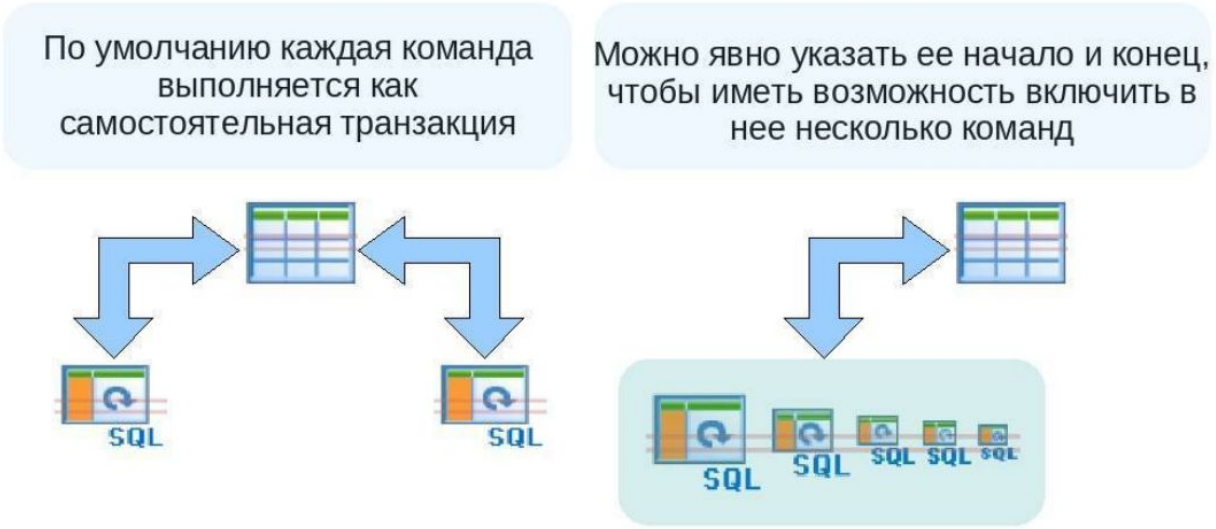
Транзакции сопровождают:

- Создание таблиц
- Изменение таблиц
- Удаление таблиц
- Вставку наблюдений (строк)
- Изменение наблюдений
- Удаление наблюдений



Функции транзакций

Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта.
Транзакции обрабатываются **транзакционными системами**, в процессе работы которых создаётся история транзакций.
Необходимы для поддержки целостности данных, журналирования запросов, восстановления РБД и т.д.



Часть 5. Очистка данных

Материалы

1. Грязные данные, пропуски в данных, невалидные данные
2. Понятие чистых данных
3. Пропуски в строковых данных
4. Пропуски в целых и вещественных числах
5. Пропуски в категориях
6. Ограничения на применения алгоритмов заполнения пропусков в данных (количество пропусков по сравнению с числом значений в таблице)

Грязные данные

Грязные данные – это неверные, недостаточные, не несущие никакой пользы. К таковым относится информация, представленная в некорректном формате или несоответствующая критериям. Они появились вместе с системой ввода данных.

Причиной их появления может быть что угодно:

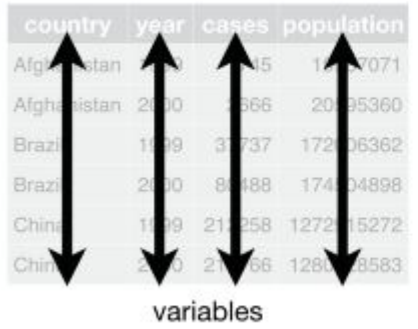
- ошибка во время ввода;
- противоречие критериям;
- отсутствие оперативного обновления;
- неправильное обновление копий данных;

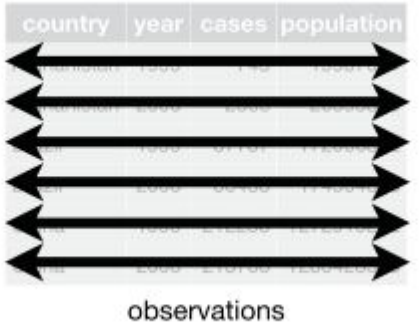
Понятие чистых данных

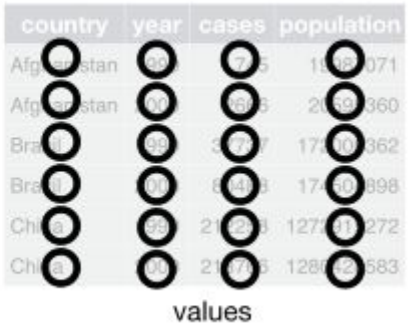
Чистые данные представляют собой табличный набор наблюдений в котором каждой строке данных соответствует полный перечень атрибутов с адекватными значениями.

Пропуски в данных и невалидные данные не являются допустимым сценарием для качественной обработки данных.

Грязные данные же являются антиподом чистых данных. Грязные данные – табличный набор наблюдений, подверженный пропускам и искаженям. Адекватность данных измеряется шкалами измерений.





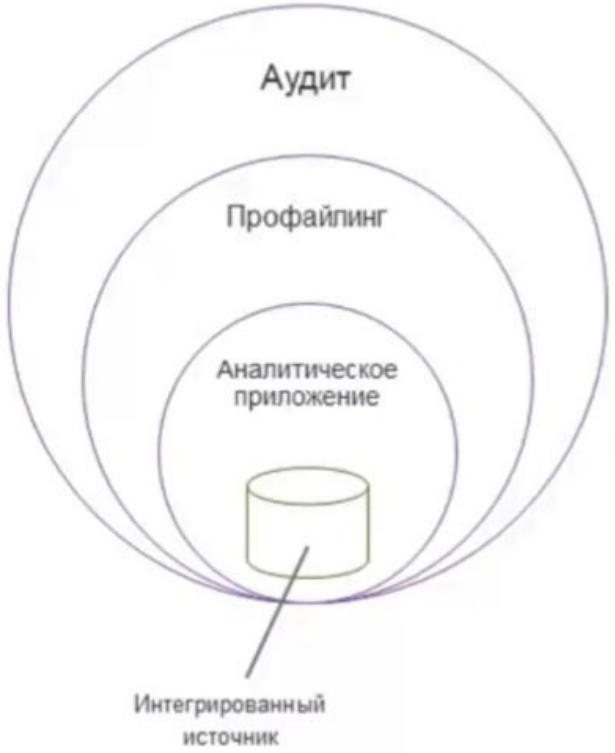


Профайлинг данных

Профайлинг данных – процесс изучения данных с целью достижения понимания их структуры, содержимого и оценки качества.

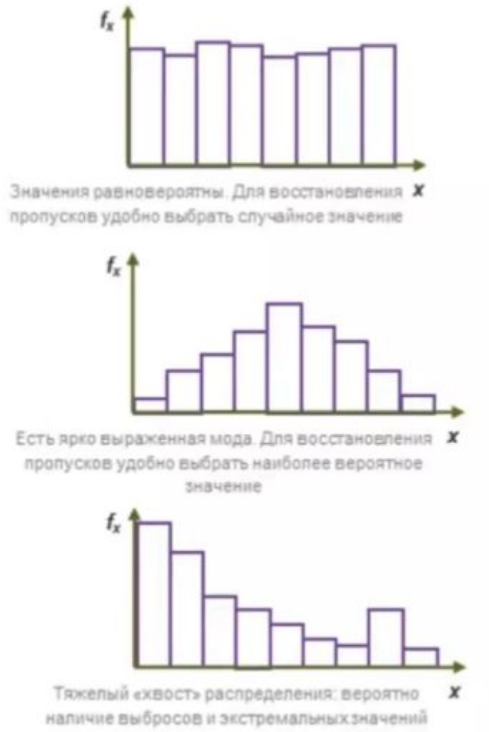
Профайлинг данных включает в себя следующие этапы:

- Подведение общих описательных статистик по выборке.
- Обнаружение пропусков.
- Обнаружение выбросов и экстремальных значений.
- Обнаружение дубликатов и противоречий.
- Сложные проверки.



Результат профайлинга данных

Пропуски в данных

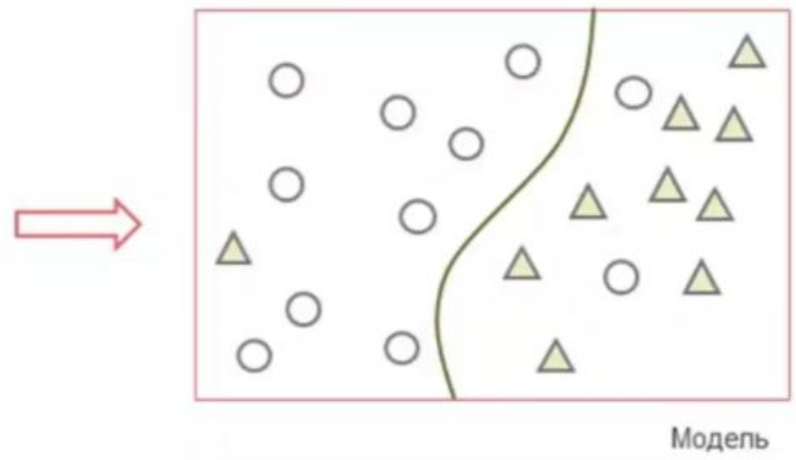


№	Возраст	Стаж
1	32	9
2		7
3	45	
4	25	1
5		2
6	22	4
7	30	12
8	46	23
9		1
10	59	
11	19	1
12		7

ФИО	Пол	Доход	Возраст
Иванова Н.А.	Жен	17000	39
		~~15000~~	~~56~~
Семенов Л.И.	Муж	41000	45

Признак 1
84
85
?
45

Признак 2
32
?
17
?



Моделирование

Стратегии борьбы с пропусками

Число пропусков:

- **Очень малое** (до 0.5 – 1%) – можно удалить примеры
- **Незначительное** (1 – 1.5%) – рекомендуется восстановление пропусков
- **Среднее** (15–30%) и **большое** (30–50%) – пропуски необходимо восстановить, результаты могут быть неадекватны
- **Очень большое** (50% и выше) – лучше отказаться от анализа набора данных

Выбросы и экстремальные значения

Значение является выбросом, если оно отличается от остальных наблюдений настолько, что у исследователя возникает подозрение, что оно сформировано под влиянием иных механизмов и факторов, чем большинство других данных в наборе. (3 сигмы)

Выбросы не сильно влияет на логику обработки данных, ибо не являются физически неадекватными. Экстремальные значения являются критичными при обработке данных и сильно влияют на обработку данных. (5 сигм)