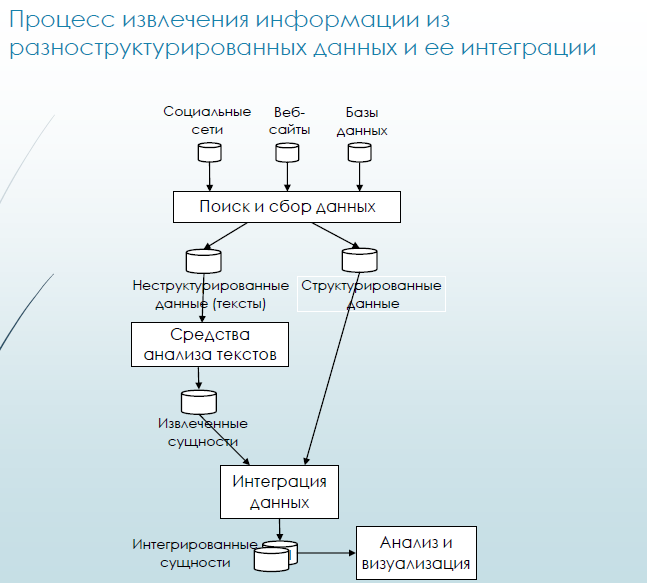
Вопросы

## Основные этапы извлечения и интеграции информации

**Four stages in data integration**: Data models unification; Schema

mapping; Entity resolution including entity matching; **Data fusion**





Для обеспечения масштабирования рассматриваемого процесса по объему извлекаемых и интегрируемых данных, его необходимо реализовать на основе некоторой платформы распределенного хранения и обработки больших объемов данных.

В данной работе в качестве такой платформы рассматривается ApacheHadoop.

Для обеспечения реализации методов интеграции данных над Hadoop(включая разрешение и слияние сущностей), в данной работе используется декларативный язык HIL

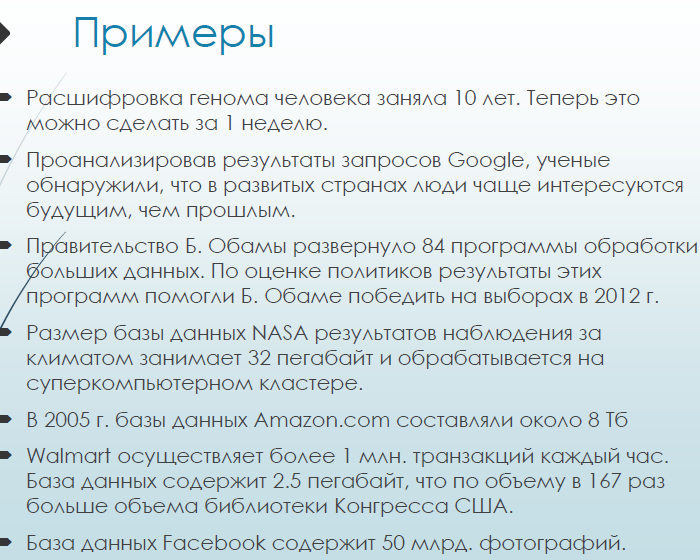
Выбран дистрибутив IBM BigInsights

В рамках архитектуры предполагается, что коллекции разноструктурированныхданных, релевантных задаче, уже отобраны экспертами.

Также предполагается, что под задачу создана целевая схема (в реляционной модели данных), в которую должны отображаться исходные коллекции.

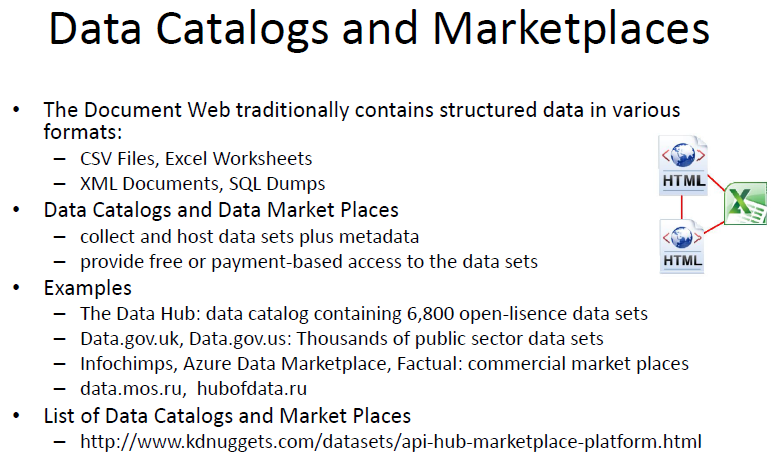
## Примеры приложений и проектов для интеграции информации

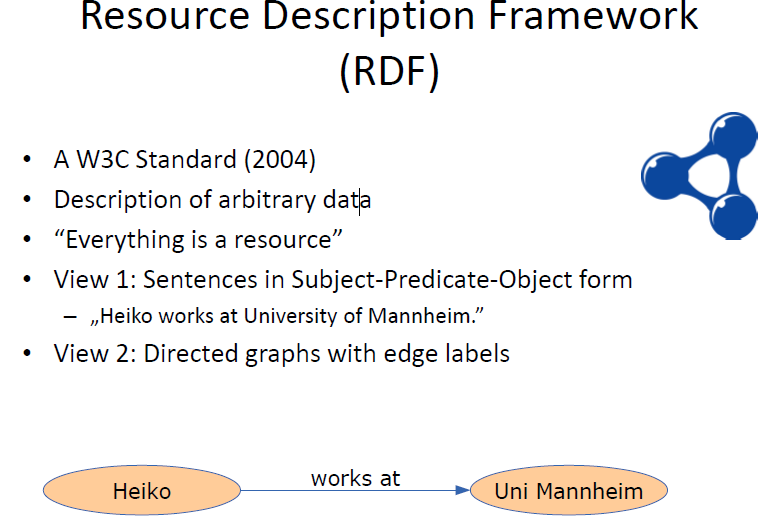


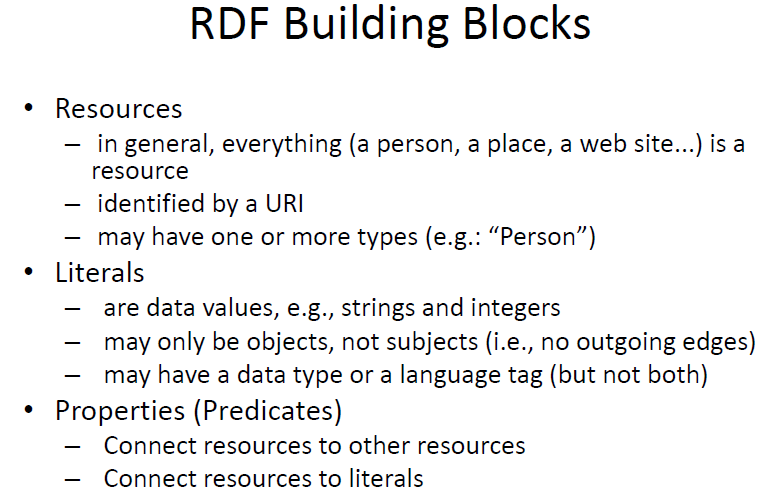


## Специфика извлечения и интеграции информации из больших данных

## Виды источников данных







Связанные данные:

Расширение интернета в виде единого глобального графа данных

–При использовании RDF для публикации структурированных данных в Интернете

- путем установки связей между элементами данных в разных источниках данных.

Сущности идентифицирутся по URI

Веб-форматы данных – JSON, xml, csv, html, …

CSV

• Не особенно формат веб-данных  
• Но довольно широко используется (также в Интернете)  
• Данные, экспортированные из RDBM и электронных таблиц  
• Файл CSV (значения, разделенные запятыми) кодирует таблицу  
• Первая строка часто используется в качестве заголовка

JSON

• JavaScript: популярный язык программирования в сети  
• Встроенный в HTML  
• Первоначально:  
- Простые взаимодействия (например, обмен изображениями при наведении мыши)  
•В наше время:  
–Также для сложных приложений  
–Ajax (асинхронный JavaScript и XML)

–Объекты, как они отмечены в JavaScript

–Объекты заключены в фигурные скобки {…}

–Данные организованы в пары ключ-значение

-Возможны вложенные объекты

## Задачи курирования данных -L3



**Применимость данных в решении задач**

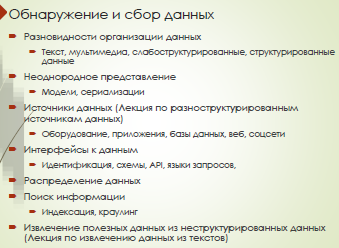
1. \* Понимание состава данных
2. \* Понимание структуры данных
3. \* Данные должны быть некоторым образом каталогизированы/классифицированы в предметных областях
4. \* Сведения об источниках данных
5. \* Приемлемое качество данных
6. \* Наличие методов обработки
7. \* Возможность использования результатов решения других задач

На каждом этапе своего жизненного цикла данные требуют сопровождения для обеспечения их применимости в решении задач.

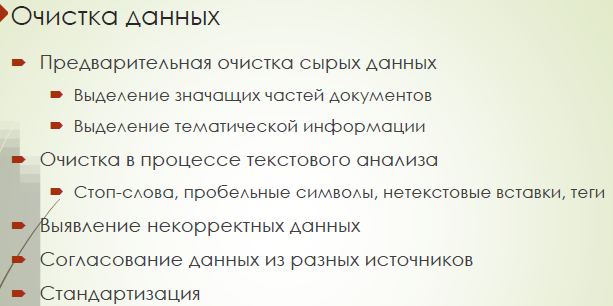
**Курирование данных –** это постоянное активное управление данными в течение всего их полезного жизненного цикла. Цели курирования – полезное использование данных, доступность данных, обеспечение качества данных.

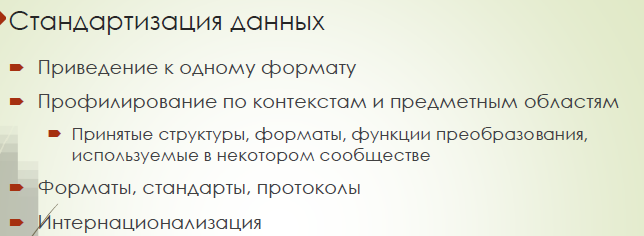
**Основными задачами КУРИРОВАНИЯ ДАННЫХ являются**

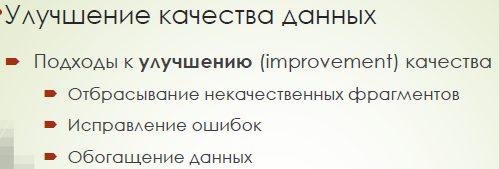
* Обнаружение и сбор данных



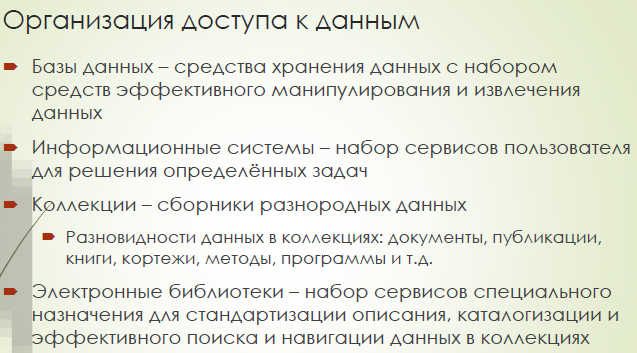
* Подготовка и очистка данных

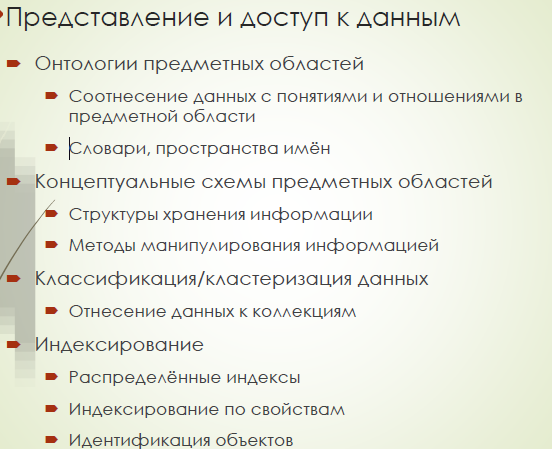




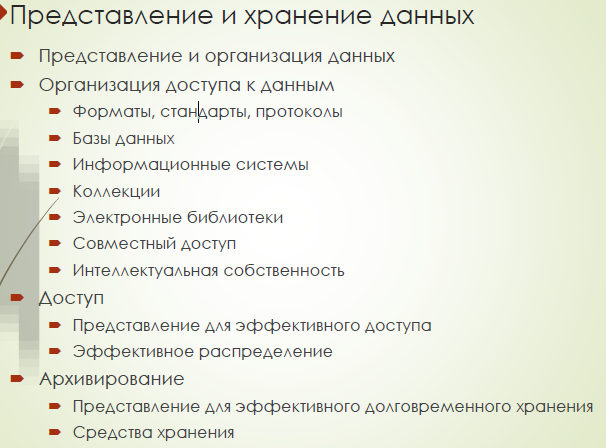


* Представление данных
* Организация доступа к данным

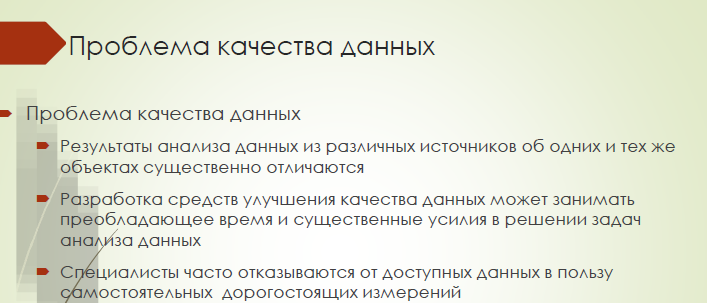


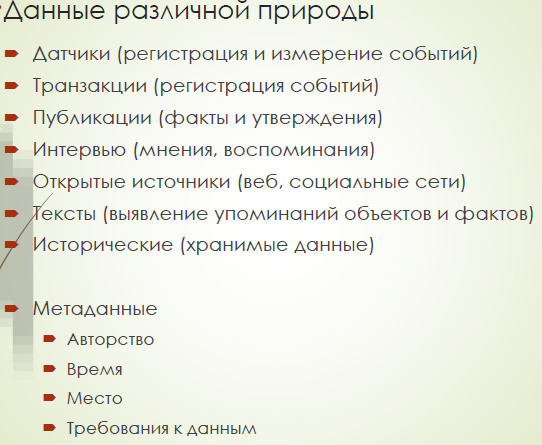


* Обеспечение долговременного хранения и архивирование



* Обеспечение повторного использования данных и методов
* Сопровождение данных дополнительными ресурсами
* Оценка и поддержание качества данных

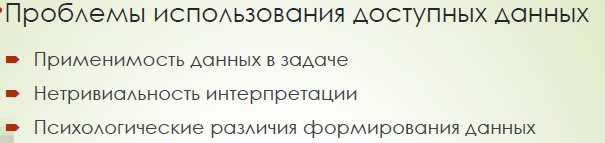




*Дополнительно*



## Проблемы использования открытых данных и подходы к их решению – L3



## Что такое извлечение информации из текстовых данных – L5

## Виды представления извлеченных данных

## Примеры приложений и проектов, использующих извлечение информации из текстов

## Способы извлечения информации из текстов

## Что такое неоднозначность понятий и подходы к разрешению неоднозначностей

## Процесс интеграции информации. Сопоставление схем. Трансформация данных

## Подходы к разрешению конфликтов схем

## Разрешение сущностей (Entity Resolution, ER). Выявление дубликатов, удаление дубликатов, установление связей между сущностями из разных исходных коллекций

## Алгоритмы ER. Data Preparation and Match Features

## Алгоритмы ER. Pairwise ER

## Алгоритмы ER. Constraints in ER

## Алгоритмы ER. Record linkage, Deduplication, Collective

## Масштабирование ER для больших данных. Blocking. Distributed ER

## Relational ER. Sorted Neighborhood Method

## Hierarchical ER. Containment Metric. Top-Down & Bottom-Up Algorithms

## Graph ER. Graph Entity Resolution Framework

## Что такое Слияние данных (Data Fusion, DF)

* Фокус на слияние данных
* Адреса второй вызов (противоречивые данные)
* Объединение записей одного и того же объекта реального мира в одну запись
* Разрешение возможных конфликтов из разных источников данных
* Обнаружение и удаление грязных данных

Преобразуйте извлеченные (и, возможно, связанные) записи в агрегированные сущности (чтобы объединить и отобразить исходные данные в целевую схему: группировать, агрегировать и объединять данные в соответствующие, потенциально вложенные выходные структуры).  
♦ Поскольку данные собираются из нескольких источников, повторяющиеся значения для определенных полей неизбежны и должны быть решены, чтобы определить, какое значение сохраняется.  
♦ Одна из серьезных проблем состоит в том, как удовлетворить целевые зависимости, ограничения (например, key, tgd, egd), рассматривая их как ограничения ETL, наложенные на созданные коллекции (возможно, как часть процесса разрешения сущности).  
Problem Еще одной проблемой является разрешение конфликтов личности. Используя методы обнаружения дубликатов, можно найти несколько, возможно, несовместимых представлений одних и тех же объектов реального мира и назначить идентификатор объекта (может быть предоставлена ​​надлежащая ссылка)  
♦ Очевидным следующим шагом после обнаружения дубликатов является их объединение или объединение и, таким образом, создание единого, возможно, более полного представления этого объекта реального мира. На этом этапе возможные конфликты данных между несколькими представлениями должны быть каким-то образом разрешены. Этот второй шаг называется объединением данных

Слияние данных: разрешение конфликтов и повышение точности данных в процессе интеграции данных

## Слияние данных. Completeness, Conciseness, and Correctness

Слияние данных: разрешение конфликтов и повышение точности данных в процессе интеграции данных

Полнота и согласованность

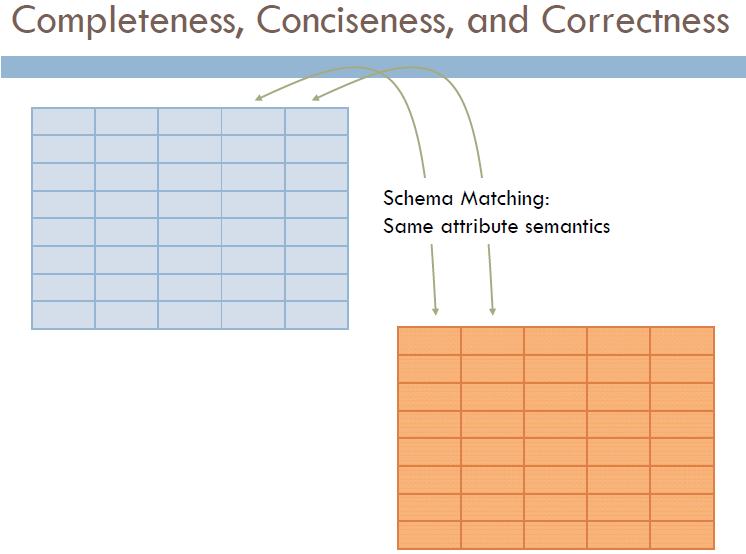
Полнота (Completeness) набора данных, такого как результат запроса или исходная таблица, измеряет объем данных в этом наборе. Количество измеряется относительно всех данных, присутствующих во вселенной дискурса.

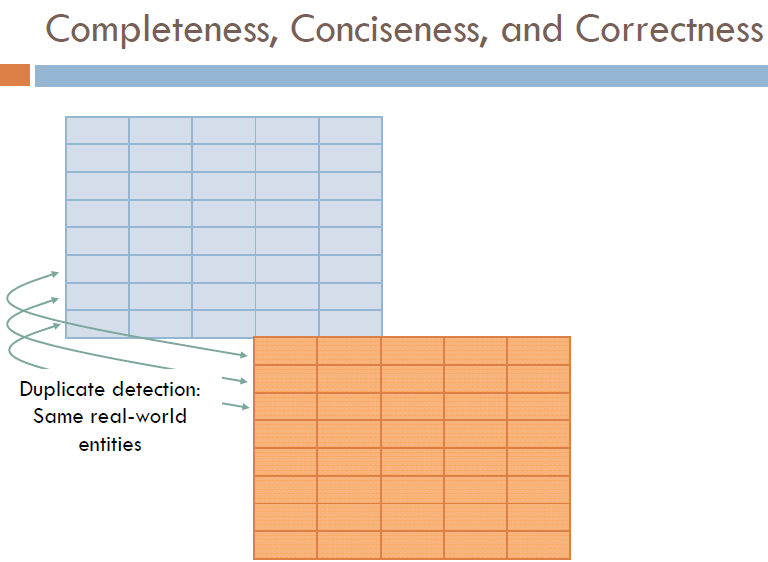
Полнота экстенсивности - это число уникальных представлений объектов в наборе данных по отношению к общему количеству уникальных объектов в реальном мире, например, во всех источниках интегрированной системы.

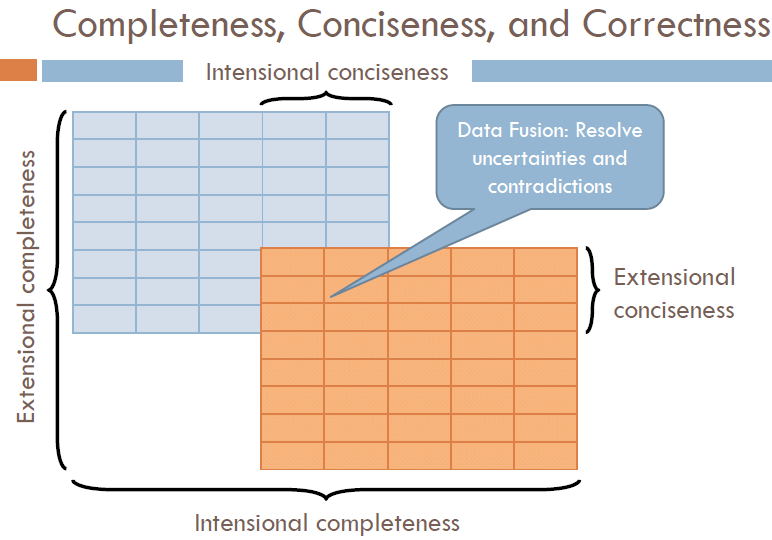
Интенсивная полнота - это число уникальных атрибутов в наборе данных по отношению к общему количеству доступных уникальных атрибутов. По аналогии с точностью, краткость измеряет уникальность представлений объектов в наборе данных.

Четыре степени интеграции, объединяющие две таблицы (1: нет сопоставления атрибутов, не используются ключи сущности; 2: сопоставление атрибутов используется, но не используется ключ сущности: 3: используется частичное сопоставление атрибутов и ключи сущностей; 4: полное сопоставление атрибутов и ключи сущности используются). 4-й вариант обеспечивает наиболее интегрированный случай.

Большинство методов интеграции и интегрированных информационных систем пытаются увеличить как экстенсионную, так и интенциональную полноту

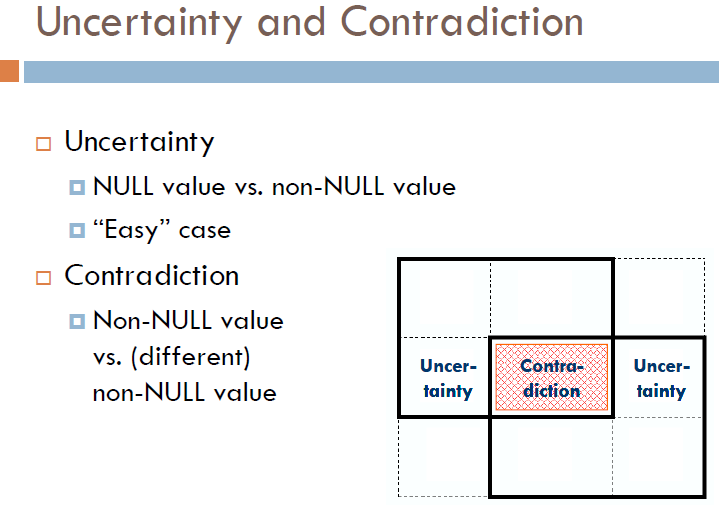


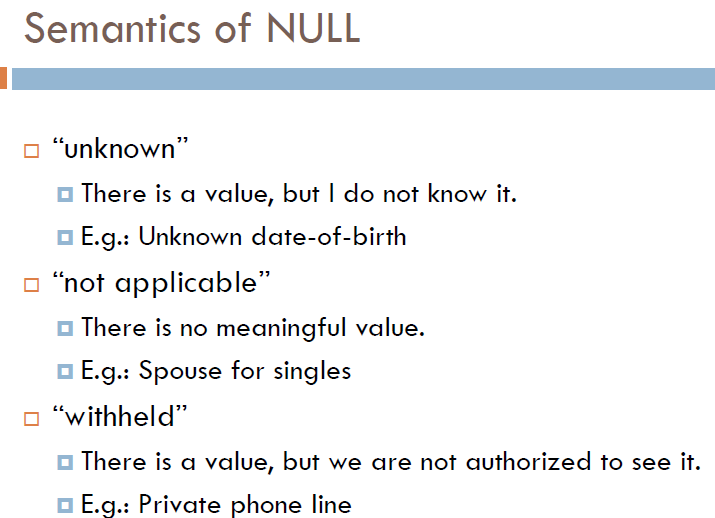




## Слияние данных. Стратегии разрешения конфликтов. Функции разрешения конфликтов (ГОС – ДОП-4)

Слияние данных: разрешение конфликтов и повышение точности данных в процессе интеграции данных





Конфликты данных:

пропущенные данные (значение неизвестно или неприменимо, или скрыто),

неопределенность (значения NULL и NULL),

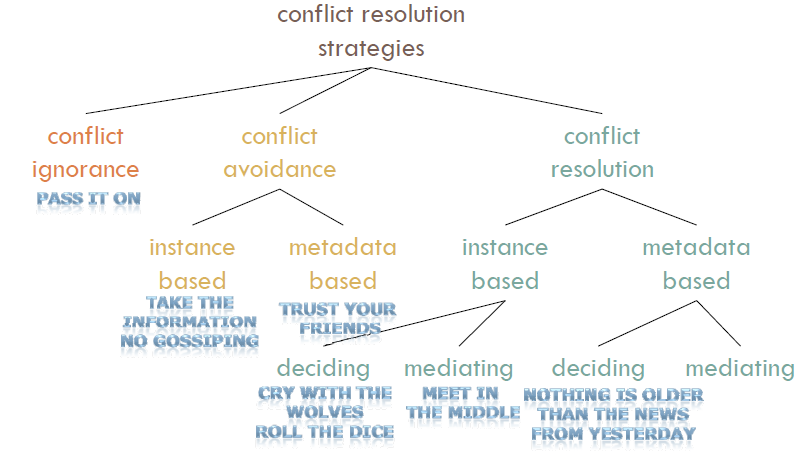
противоречие (как минимум, два различных значения, отличных от NULL)

Разрешение конфликтов - классы:

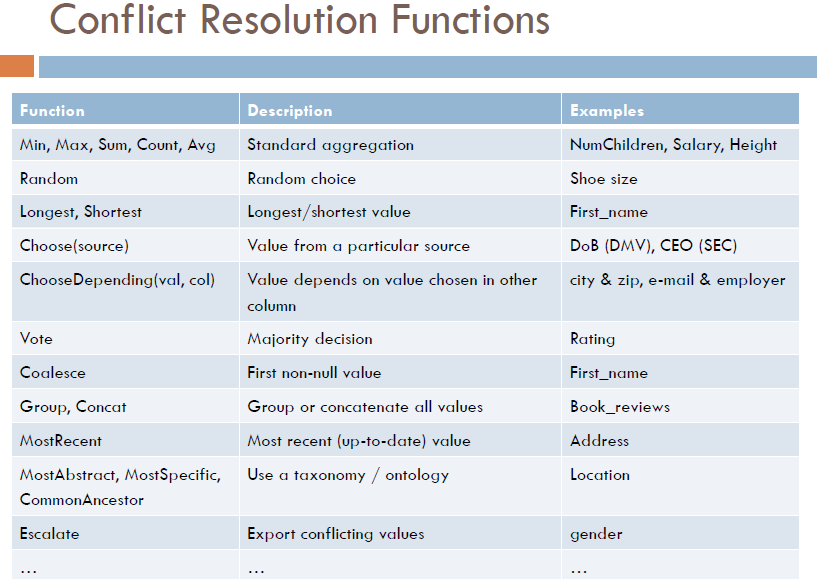
Игнорирование конфликтов – Оставлять как есть (рассматриваются все возможности)

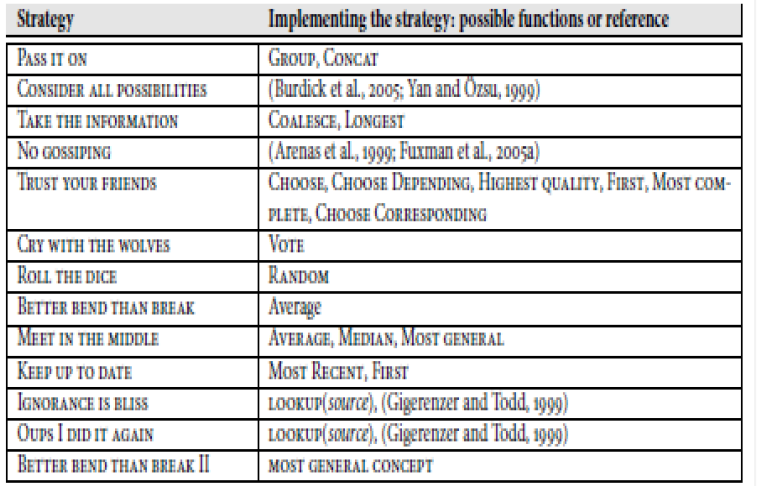
Предотвращение конфликтов – все конфликтные данные – убираются, остается только то, что верно (например предпочтение данных одного источника данным другого, большее доверие тем или иным данным). По сути, убираются все противоречия

Разрешение конфликтов – стратегия принятия решений и посредничества – в зависимости от класса выбор значения зависит только от значений данных (на основе экземпляров) или также учитывает некоторую другую информацию на основе метаданных. С другой стороны, стратегии посредничества могут выбирать значение, которое необязательно существует среди конфликтующих значений. Такие стратегии могут принять решение о новой ценности, которой не было среди конфликтующих ценностей.









## Слияние данных. Union-based Approaches. Union, Outer Union, Minimum Union, Complement Union (ГОС – ДОП-4)

Слияние данных: разрешение конфликтов и повышение точности данных в процессе интеграции данных

♦ Объединение подходов: вся информация из исходных таблиц сохраняется и может быть легко достигнута как объемная, так и полная интенсификация  
♦ Удаляется только ненужная информация (т. Е. Дубликаты или подчиненные кортежи), что увеличивает краткость.  
♦ Однако в большинстве случаев сохраняется слишком много кортежей и, следовательно, слишком много информации, что требует дополнительного шага к более удовлетворительному уровню краткости. В конечном итоге это приводит к одному кортежу на объект реального мира.  
♦ Интуитивно понятный способ увеличения краткости вместе с объединенным подходом заключается в применении группирования и агрегирования.

1) ***Union, Outer Union, and Minimum Union***. Оператор ***Union*** (с заданной семантикой) объединяет кортежи двух союзно-совместимых отношений и удаляет точные дубликаты (кортежи, которые совпадают во всех значениях атрибутов). ***Outer Union*** преодолевает одно ограничение объединения и объединяет два несовместимых отношения, используя атрибуты первого дополнения, которые не являются общими с нулевыми значениями. ***Union*** и ***Outer Union*** - это не ценность, а сохранение объекта. Они не сохраняют уникальность, а конфликты игнорируются. Оператор ***Minimum Union*** определяется как результат ***Outer Union***, из которого удалены удаленные кортежи.  
2) ***Merge,***, ***Prioritized Merge***. Оператор ***Merge*** можно описать как ***Union*** двух ***Outer Union***. Существует вариант, а именно слияние по приоритетам ***Prioritized Merge***, которое используется для определения приоритетов между источниками и предоставления значений предпочтений из одного источника. ***Merge*** хорошо подходит в сценариях, где многие нулевые значения могут быть дополнены, например, когда источники данных заполнены редко или только слегка перекрываются в своих схемах.

(Нормальный) союз: исключить точные дубликаты  
Минимальный союз: исключить вложенные кортежи  
Кортеж t1 включает в себя кортеж t2  
- если он использует ту же схему  
- имеет меньшие значения NULL  
- совпадает во всех ненулевых значениях  
Два шага  
- Внешний союз  
--Subsumption

## Слияние данных. Join-based Approaches. Full Disjunction, Match Join, Grouping And Aggregation (ГОС – ДОП-4)

Слияние данных: разрешение конфликтов и повышение точности данных в процессе интеграции данных

Join-based Approaches: подходы объединения обычно сохраняют информацию об одних и тех же объектах реального мира в разных столбцах (не в разных строках). Столбцы с одинаковыми семантическими атрибутами автоматически не совпадают, как в подходах объединения.  
Их нужно явно комбинировать.  
♦ Комплектность приходит более или менее бесплатно; но краткость должна быть достигнута путем дополнительной обработки результата соединения.  
♦ Если в таблицах нет дубликатов внутри источника, достаточно простой комбинации двух строк с использованием определенной пользователем функции. Тем не менее, если есть дубликаты внутри источника, группировка и агрегация, или требуется какой-то расширенный оператор соединения.

1) ***Standard Joins***. Равное объединение объединяет кортежи из двух отношений, если условие объединения состоит из условий равенства между столбцами. Key-join - это сохранение уникальности, но не обязательно сохранение значения / объекта. То же самое верно для естественного соединения. Полное внешнее соединение расширяет результат стандартной операции соединения, добавляя кортежи, которые включены только в одно из исходных отношений.  
2) ***Full Disjunction*** Оператор полной дизъюнкции определяется как комбинация двух или более таблиц, где все совпадающие кортежи объединяются в один кортеж. Полные дизъюнкции - это сохранение уникальности и сохранение значения / объекта, как полных внешних объединений. Полная дизъюнкция, как правило, не может быть рассчитана с помощью комбинации только внешних соединений.

3) ***Match Join and Conflict Tolerant Queries***. На первом этапе соответствующие значения атрибутов из всех источников отдельно проецируются и объединяются объединением, расширяемым их соответствующим идентификатором реального мира. Затем идентификаторы реального мира используются для соединения, в результате чего получается одна большая таблица. Операция по созданию этой таблицы называется match join. На втором шаге, чтобы увеличить краткость, кортежи выбираются из этой таблицы в соответствии с сопровождающей конфликтоустойчивой моделью запросов. При выборе кортежей он использует предикат выбора и наблюдает за параметром оценки операции.  
Этот параметр может быть:  
   высокая достоверность: предикат выбора должен быть истинным для всех кортежей для каждого идентификатора реального мира;  
   случайное доказательство: проверяет предикат для случайного кортежа по реальному идентификатору;  
   возможно вообще: предикат выбора должен быть истинным как минимум для одного кортежа для каждого реального идентификатора.  
В конце остается только одно представление для каждого идентификатора реального мира, таким образом, оператор обеспечивает выполнение уникальности, создавая бесконфликтный результат, но, в зависимости от параметризации, сохраняет значение / объект.

