Базы данных в информационных системах, хранилища данных

Соловьев А.В.

Базы данных

Международные стандарты:

- **База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных (ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными)
- **База данных** — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причём такое собрание данных, которое поддерживает одну или более областей применения (ISO/IEC 2382-1:1993. Information technology — Vocabulary — Part 1: Fundamental terms)

Базы данных

Другие определения:

- **База данных** — некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия (**Дейт**)
- **База данных** — организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей
- **База данных** — совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации

Системы управления базами данных

Система управления базами данных (**СУБД**) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных (ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными)

Системы управления базами данных

Основные функции СУБД

- управление данными во внешней и оперативной памяти;
- журнализация изменений,
- резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными);
- разделение прав доступа и защита данных от НСД;
- разрешение конфликтов параллельной работы с данными.

Информационная система

Информационная система (**ИС**) — система обработки информации и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию (ISO/IEC 2382-1:1993 Information technology — Vocabulary — Part 1: Fundamental terms)

ИС предназначена для своевременного обеспечения определенных пользователей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области, при этом результатом функционирования информационных систем является **информационная продукция** — документы, информационные массивы, базы данных и информационные услуги

Базы данных

Использование в ИС:

- **Централизованная (ЦБД)** (*centralized database*): БД, полностью поддерживаемая на одном компьютере.
- **Распределённая (РБД)** (*distributed database*): БД, составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.
  - Неоднородная (*heterogeneous distributed database*): фрагменты распределённой БД в разных узлах сети поддерживаются средствами более одной СУБД
  - Однородная (*homogeneous distributed database*): фрагменты распределённой БД в разных узлах сети поддерживаются средствами одной и той же СУБД.
  - Фрагментированная, или секционированная (*partitioned database*): методом распределения данных является фрагментирование (партиционирование, секционирование), вертикальное или горизонтальное.
  - Тиражированная (*replicated database*): методом распределения данных является тиражирование (репликация).

Логическая модель РБД

Логическая модель РБД строится на 3-х уровнях абстракции данных: представления информации, обработки (бизнес-логики) и хранения.

Уровни образуют строгую иерархию: уровень бизнес-логики взаимодействует с уровнями хранения и представления. Физически, уровни могут входить в состав одного программного модуля, или же распределяться на нескольких параллельных процессах в одном или нескольких узлах сети.

Уровень представления информации

- Обеспечивает интерфейс с пользователем.

Уровень бизнес-логики

- Определяет функциональность и работоспособность системы в целом. Блоки программного кода распределены по сети и могут использоваться многократно для создания сложных распределенных приложений.

Уровень хранения данных

- Обеспечивает физическое хранение, добавление, модификацию и выборку данных. На данный уровень также возлагается проверка целостности и непротиворечивости данных, а также реализация транзакций

Логическая модель РБД

Архитектура / Уровень	**Файл-сервер**	**Клиент-сервер (Бизнес-логика на клиенте)**	**Клиент-сервер (бизнес-логика на сервере)**	**N-уровневая архитектура**
Представления	Клиент	Клиент	Клиент	Клиент
Бизнес-логики	Клиент	Клиент	Сервер БД	Сервер приложений
Хранения	Файл-сервер (или клиент)	Сервер БД	Сервер БД	Сервер БД

Требования к РБД

- Локальные и глобальные (распределенные) средства доступа к данным (СУБД)
- Единообразная логика прикладных программ во всех АРМ сети.
- Малое временя реакции на запросы пользователей
- Надежность, исключающая разрушения целостности системы в случае выхода из строя ее отдельных компонент (узлов)
- Открытость, позволяющая наращивать объем локальных БД и добавлять новые АРМ
- Развитая система резервного копирования и восстановления данных на случай отказов оборудование и ПО
- Система безопасности информации, следящая за соблюдением привилегий доступа к данным
- Высокая эффективность, за счет выбора оптимальных алгоритмов использования сетевых ресурсов
- Развитые репликационные механизмы, позволяющие размещать обновленные копии данных в сети оптимальным образом

Принципы построения РБД

- Минимизация интенсивности обмена данными (сетевого трафика)
- Оптимальное размещение серверных и клиентских приложений в сети
- Декомпозиция данных на часто и редко используемые сегменты (например, для настройки репликации - размещение наиболее часто используемых данных на АРМ конечных пользователей)
- Периодическое сохранение копий данных и выполнение действий по поддержке целостности распределенной информационной системы

Критерии построения РБД

- Всесторонний анализ информационных потребностей предметной области с выявлением объемов хранимых данных их сложности, достоверности, взаимосвязанности.
- Моделирование предполагаемого сетевого трафика при работе РБД с различными моделями репликации данных.
- Кластеризация элементов данных и программ их обработки. Цель - добиться максимальной автономности и низкой связанности кластеров.
- Привязка кластеров данных к вероятным пользователям или АРМ.
- Поддержка эталонной копии данных и ограничение репликационного механизма
- Разработка и реализация правил приведения локальных и центральной БД в непротиворечивое

Свойства РБД (по К.Дейту)

1. Локальная автономия

- Управление данными на каждом из узлов распределенной системы выполняется локально. База данных, расположенная на одном из узлов, является неотъемлемым компонентом распределенной системы. Будучи фрагментом общего пространства данных, она, в то же время функционирует как полноценная локальная база данных; управление ею выполняется локально и независимо от других узлов системы

2. Независимость узлов

- В идеальной системе все узлы равноправны и независимы, а расположенные на них базы являются равноправными поставщиками данных в общее пространство данных. База данных на каждом из узлов самодостаточна - она включает полный собственный набор данных и полностью защищена от несанкционированного доступа

3. Непрерывные операции

- Возможность непрерывного доступа к данным (известное "24 х 7") в рамках РДБ вне зависимости от их расположения и вне зависимости от операций, выполняемых на локальных узлах. Это качество можно выразить лозунгом "данные доступны всегда, а операции над ними выполняются непрерывно"

Свойства РБД (по К.Дейту)

4. Прозрачность расположения

- Полная прозрачность расположения данных. Пользователь, обращающийся к РДБ, ничего не должен знать о реальном, физическом размещении данных в узлах информационной системы. Все операции над данными выполняются без учета их местонахождения. Транспортировка запросов к базам данных осуществляется встроенными системными средствами

5. Прозрачная фрагментация

- Возможность распределенного (то есть на различных узлах) размещения данных, логически представляющих собой единое целое. Существует фрагментация двух типов: горизонтальная и вертикальная. Первая означает хранение строк одной таблицы на различных узлах (фактически, хранение строк одной логической таблицы в нескольких идентичных физических таблицах на различных узлах). Вторая означает распределение столбцов логической таблицы по нескольким узлам

6. Прозрачное тиражирование

- Тиражирование данных - это асинхронный (в общем случае) процесс переноса изменений объектов исходной базы данных в базы, расположенные на других узлах распределенной системы. Прозрачность тиражирования означает возможность переноса изменений между базами данных внутрисистемными средствами, невидимыми пользователю распределенной системы

Свойства РБД (по К.Дейту)

7. Обработка распределенных запросов

- Возможность выполнения операций выборки над распределенной базой данных, сформулированных в рамках обычного запроса на языке SQL. То есть операцию выборки из РДБ можно сформулировать с помощью тех же языковых средств, что и операцию над локальной базой данных

8. Обработка распределенных транзакций

- Возможность выполнения операций обновления распределенной базы данных (**INSERT, UPDATE, DELETE**), не разрушающее целостность и согласованность данных. Эта цель достигается применением двухфазового или двухфазного протокола фиксации транзакций (two-phase commit protocol). Его применение гарантирует согласованное изменение данных на нескольких узлах в рамках распределенной (или, как ее еще называют, глобальной) транзакции

9. Прозрачность сети

- В распределенной системе возможны любые сетевые протоколы. Доступ к любым базам данных может осуществляться по сети. Спектр поддерживаемых конкретной СУБД сетевых протоколов не должен быть ограничением системы с распределенными базами данных

Свойства РБД (по К.Дейту)

10. Независимость от оборудования

- В качестве узлов распределенной системы могут выступать компьютеры любых моделей и производителей - от мэйнфреймов до "персоналок"

11. Независимость от операционных систем

- Это качество вытекает из предыдущего и означает многообразие операционных систем, управляющих узлами распределенной системы

12. Независимость от систем управления

- В распределенной системе могут мирно сосуществовать СУБД различных производителей, и возможны операции поиска и обновления в базах данных различных моделей и форматов

Жизненный цикл БД

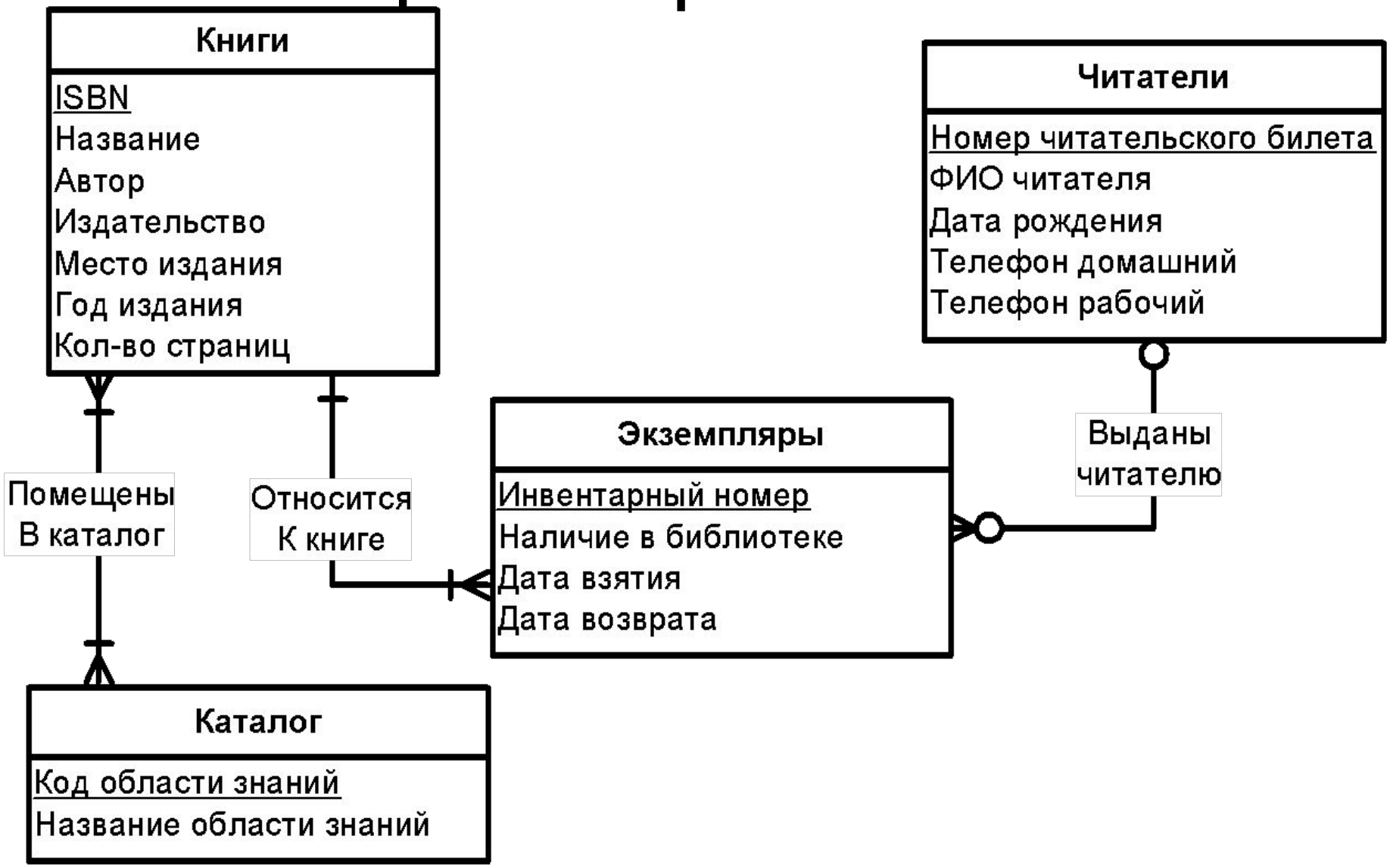
Проектирование БД

Системный анализ предметной области

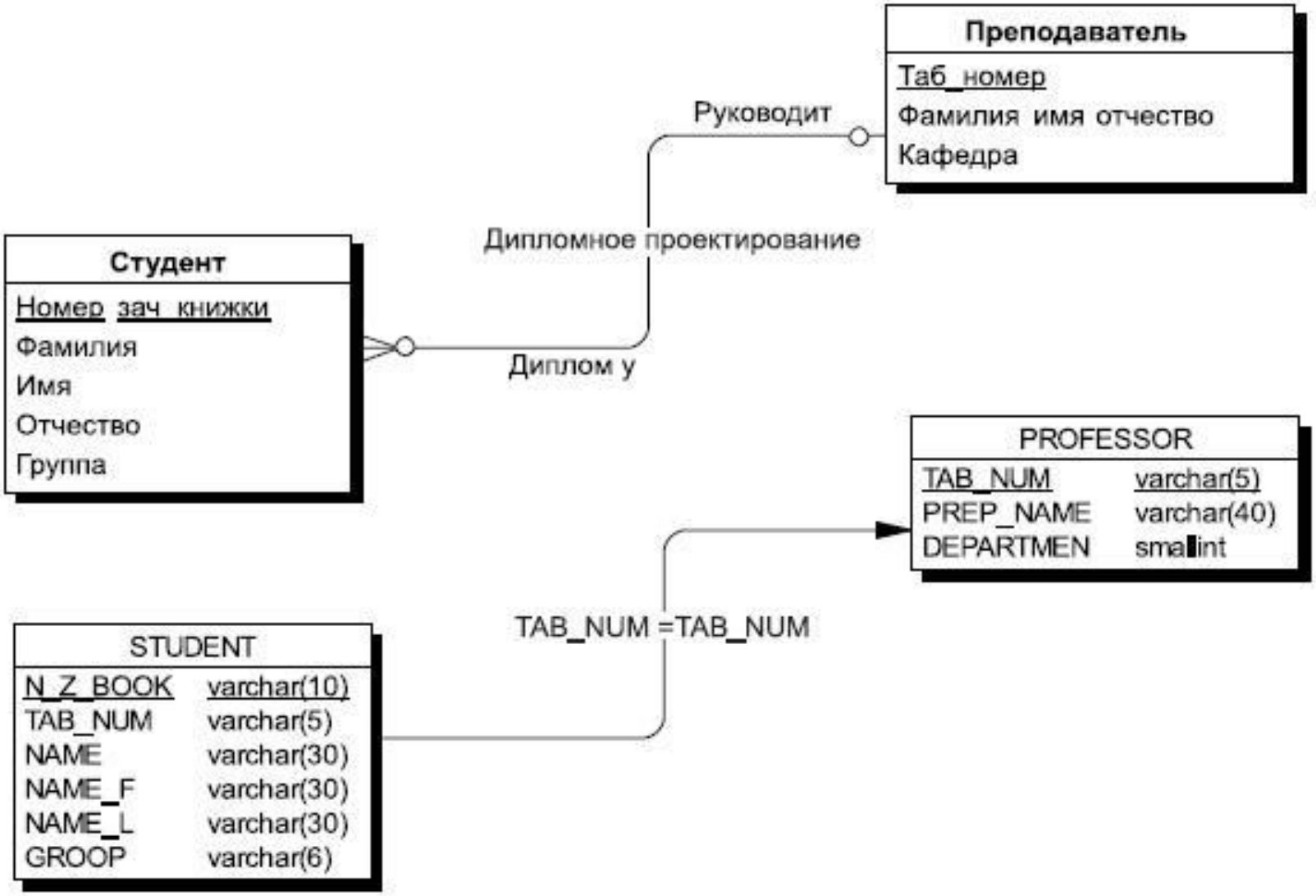
- **Функциональный подход** — он реализует принцип движения "от задач" и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая БД. В этом случае мы можем четко выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны
- **Предметный подход** — когда информационные потребности будущих пользователей БД жестко не фиксируются. Мы не можем точно выделить минимальный набор объектов предметной области, которые необходимо описывать. В описание предметной области в этом случае включаются такие объекты и взаимосвязи, которые наиболее характерны и наиболее существенны для нее (принцип построения диаграмм прецедентов – UseCase). БД, конструируемая при этом, называется предметной, то есть она может быть использована при решении множества разнообразных, заранее не определенных задач

Пример описания предметной области

Инфологическое проектирование



Даталогическое проектирование



Даталогическое проектирование

Выбор СУБД

Критерии выбора СУБД:

- Архитектурные возможности СУБД (1)
- Коммуникационные возможности СУБД (2)
- Функциональные возможности СУБД (3)
- Средства разработки БД (4)
- Эффективность работы СУБД (5)
- Надежность работы СУБД (6)
- Требования к рабочей среде (7)
- Особенности разработки приложений (8)
- Финансовые критерии (9)
- Социальные критерии (10)

Архитектурные возможности СУБД

Масштабируемость (1.1). Необходимо учитывать, возможность увеличения числа пользователей, объема хранимых данных и объема, а также формы обрабатываемой информации. По данному критерию нет необходимости стремиться к самому максимальному значению из всех возможных в рассматриваемом классе СУБД. Этот критерий должен быть оценен по максимуму из ТРЕБУЕМЫХ значений (с учетом стратегии развития плюс учтенный запас).

Распределенность (1.2). В случае не централизованного хранения и обработки данных на одном сервере, различные СУБД имеют разные инструментальные возможности управления распределенными БД. Необходимо рассмотреть СУБД на максимизацию производительности

Коммуникационные возможности СУБД

Сетевые возможности (2.1)

Необходимо выбрать оптимальные для обеспечения решаемых задач набор сетевых протоколов и служб для работы и администрирования

Поддержка стандартных интерфейсов связи с БД (2.2)

Этот критерий имеет большую важность при оценке

Использование БД других форматов (2.3)

Наибольший балл по данному
критерию имеют СУБД, которые способны проводит операции над БД других форматов
без импортирования или преобразования

Импорт и экспорт данных из БД (2.4)

Функциональные возможности СУБД

Внутренний язык СУБД (3.1). В различных СУБД для реализации триггеров, хранимых процедур, генерации ключей, обеспечения целостности, управления транзакциями и т. п. используются неодинаковые языки реализации

Типы данных. Здесь следует рассмотреть базовые и основные типы данных; также наличие возможностей их расширения до необходимых в данной реализации, и ограничения на операции над данными

Соответствие стандартам языка запросов SQL (3.2). Все современные системы совместимы с базовым стандартом языка запросов SQL-92, однако многие из них реализуют различные расширения данного стандарта. Кроме того, наличие в СУБД реализации новых требований, которые заложены в последних стандартах SQL:2008 является преимуществом

Средства разработки БД

Средства проектировании БД (4.1)
Некоторые СУБД имеют свои средства проектирования БД, которые инструментарием существенно различаются
Средства для оптимизации запросов (4.2)
Возможности инструментального проведения анализа оптимальности выполнения запросов
Основные и дополнительные средства поиска (4.3)
Некоторые современные системы имеют дополнительные средства для поиска, в частности средства обеспечивающий поиск близкий к контекстному

Эффективность работы СУБД

Контроль использования ресурсов сервера (5.1)
Система может иметь возможность управления использованием как оперативной памяти, так и дискового пространства. Необходимо оценить наличие, гибкость и автоматизированность данных настроек
Настройка производительности (5.2)
Рейтинг TPC (Transactions per Cent) (5.3)
Параллельная обработка (5.4)
Оптимизирование запросов (5.5)
Оценка производительности (5.6)
Один из возможных методов оценки производительности – это проведение тестирования с помощью эталонных тестов из набора AS3AP (ANSI SQL Standard Scalable and Portable), корторый контролирует широкий спектр часто встречающихся операций БД и моделируют в том числе однопользовательские и многопользовательские среды

Надежность работы СУБД

Восстановление после сбоев (6.1). Эффективные механизмы восстановления как после мягких, так и после жестких сбоев
Резервное копирование (6.2). Существует несколько механизмов резервирования данных: хранение одной или более копий всей базы данных, хранение копии ее части, копирование логической структуры и т. д. В данном случае должна быть прямая зависимость оценки по критерию от количества механизмов
Механизм управления транзакциями (6.3). Особе внимание необходимо уделить механизму отката транзакций, который может иметь различное быстродействие и эффективность. При сравнении СУБД по данному критерию не стоит отказываться по возможности экспериментальной практики
Информационная безопасность (6.4). Существуют несколько различных механизмов защиты данных: дискреционное управление доступом, мандатное управление доступом, шифрование информации. Расчет значения по данному критерию лучше производить в сочетании с коэффициентом секретности данных в разрабатываемой БД, т.е. возможно нет необходимости в наличии всех известны на сегодняшний момент механизмов защиты данных в СУБД

Требования к рабочей среде

Мобильность (7.1). Необходимо предусмотреть максимальную независимость БД, как от аппаратных средства, так и от программного обеспечения, в частности от операционной системы (хотя бы гарантировать неизменность предустановленной до или в процессе разработки операционной системы)

Минимальные требования по оборудованию и ПО (7.2). В данном критерии требуется оценить минимальность необходимости наличия узко специализированного (не традиционного) оборудования и ПО для полнофункциональной и качественной работы БД

Особенности разработки приложений
Средства разработки приложений в архитектурах типа клиент-сервер (8.1)
Наличие таких средств позволяет наилучшим образом реализовать все возможности СУБД и даже производить автоматического проектирования приложений. Данный критерий должен иметь переключаемый весовой коэффициент, зависящий от необходимости разработок приложений такого характера
Разработка Web-приложений (8.2)
Наличие набора инструментов для построения приложений под Web. Данный критерий должен иметь переключаемый весовой коэффициент, зависящий от необходимости разработок приложений такого характера
Поддерживаемые языки программирования (8.3)
Широкий спектр используемых языков программирования влияет на быстродействие и функциональность приложений

Финансовые критерии

Стоимость базового комплекта (9.1). В эту оценку обязательно включать не только приобретение самой СУБД, но также приобретение аппаратных средств, установочные и наладочные работы, обучение персонала, эксплуатационные расходы, техническую поддержку. А также дополнительную стоимость (например стоимость дополнительного лицензирования пользовательских мест) согласованную с планом стратегического развития.

Качество модели общей стоимости владения (TCO) (9.2). Общая стоимость владения (от англ. Total Cost of Ownership - TCO) - это экономическая модель-методика, предназначенная для определения затрат на информационные системы (и не только), рассчитывающихся на всех этапах жизненного цикла системы. TCO позволяет понять и определить структуру затрат на информационные технологии. Все затраты разделяются на прямые и косвенные. Прямые затраты (явные) – составляют затраты, проходящие через бухгалтерию (заработная плата сотрудников, закупки оборудования и ПО и др.). Непрямые затраты (неявные) – затраты на устранение сбоев или проблем на компьютерах, простои рабочего времени, командировочные, затраты на предотвращение рисков и затраты на устранение их последствий, затраты на обучение персонала и другие подобные затраты и др.

Социальные критерии

Фирма-производитель (10.1). По данному критерию выигрывают СУБД, производители которых представляют свою высококачественную продукцию на протяжении нескольких лет на рынке с соблюдением правил наследования версионности своих продуктов. А также твердое финансовое положение производителя, годовой оборот, численность состава, объем продаж, наличие консультаций и т.д.

Распространенность СУБД (10.2). При проставлении значений необходимо учитывать и негативное влияние большой распространенности СУБД, в частности общих и известных слабых мест защиты от утечки информации.

Многоязыковая поддержка (10.3). Основным фактором при оценке по данному критерию должно являться, прежде всего, возможность использования русского языка (поддержка кириллических кодировок для символьных и строковых типов данных, возможность создания индексов для таких типов), как стандартный функционал СУБД.

Наличие документации на русском языке (10.4). Необходима отдельная оценка наличия качественной и полной, а самое главное доступной документации на русском языке.

Поэтапное проведение процесса критериальной оценки

Из выше перечисленных критериев, объединенных в 10 групп видно, что некоторые из критериев являются составными при формирования значений, что еще более усложняет задачи выбора на основе критериальной оценки. Поэтому для более простого проведения критериальной оценки при выборе СУБД необходимо сужения числа альтернативных решений. То есть, стоит проводить саму процедуру выбора в несколько этапов. В частности определить на первом этапе из всего множества СУБД только те, которые являются пригодными для решения поставленной задачи.

На качественном уровне, достаточно сравнить СУБД по следующим группам показателей:

модель данных (к1)

удобство и простота использования (к2)

- понятные процедуры установки программных продуктов,
- удобный и унифицированный интерфейс конечного пользователя,
- простота выполнения обычных операций: создания БД, модификации, подготовки данных, выполнения запросов и отчетов;
- наличие интеллектуальных подсистем подсказок, помощи в процессе
работы и обучения, включая примеры;

качество средств разработки (к3)

- возможности создания пользовательских интерфейсов,
- мощность языка создания программ,
- автоматизация разработки различных объектов: экранных форм, отчетов, запросов;

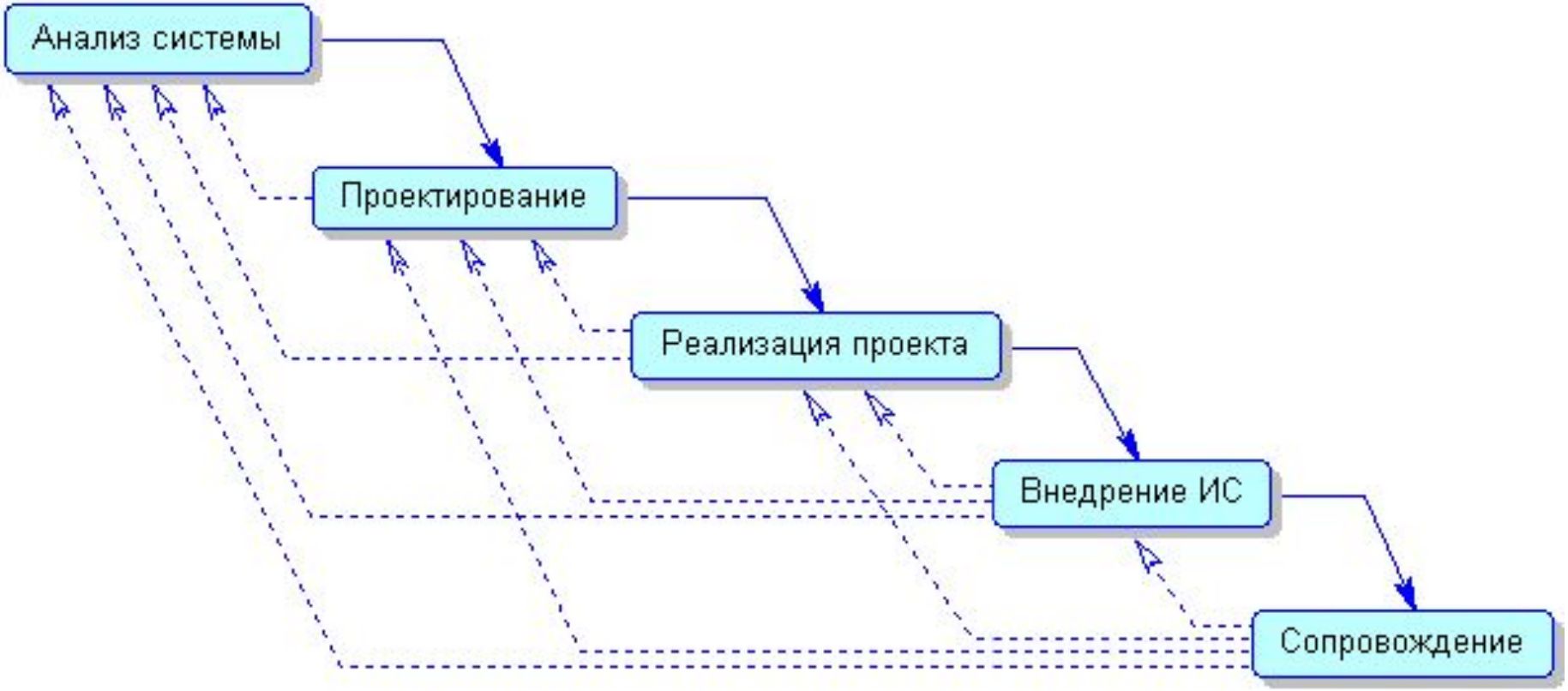
качество средств защиты БД (к4)
- доступ к функциям защиты на уровне средств разработки
- доступ к функциям защиты на уровне пользователя.

качество средств контроля корректности БД (к5)
- обеспечение уникальности записей БД по первичному ключу,
- автоматический контроль целостности связей между таблицами во время выполнения операций обновления, вставки и удаления записей,
- проверка корректности значений в БД;

качество коммуникационных средств (к6)
- поддержку сетевых протоколов,
- поддержку стандартных интерфейсов с БД,
- наличие средств групповой работы с информацией БД,
- способность использовать и модифицировать БД других форматов без импортирования или преобразования;

По вышеперечисленным критериям достаточно провести оценку по шкале: да, нет. На следующем этапе уже можно провести количественную сравнительную оценку по всем 10-и группам критериев, описанных ранее, но только для выделенных на первом этапе СУБД.

Проектирование информационных систем и БД



Проектирование ИС и БД

Для решения задач проектирования сложных систем существуют специальные методологии и стандарты.

К таким стандартам относятся методологии семейства ***IDEF*** (ICAM DEFinition, ICAM - Integrated Computer-Aided Manufacturing). С их помощью можно эффективно проектировать, отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах.

IDEF1X **(*IDEF1*** **Extended) - Data Modeling** - методология проектирования реляционных *баз данных*. Заключается в построении моделей данных типа "сущность-связь" (ERD - Entity-Relationship Diagram) в нотации этого стандарта.

Проектирование ИС и БД

В России действующие ГОСТ-ы по разработке автоматизированных систем следующие:

- ГОСТ 34.003-90 "Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения";
- ГОСТ 34.201-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем";
- ГОСТ 34.601-90 "Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания";
- ГОСТ 34.602-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы".
- На разработку программной документации действуют стандарты класса ЕСПД (ГОСТ 19.101-77 "Единая система программной документации. Общие положения" и др.)

Проектирование ИС и БД

Важное место в моделировании информационных систем занимает методология и системы, использующие ***UML*** - унифицированный язык моделирования (**Unified Modeling Language**).

UML - язык для спецификации, визуализации, конструирования и документирования сложных информационно-насыщенных объектных систем. В настоящее время зарегистрирован как международный стандарт **ISO/IEC 19501:2005** "Information technology - Open Distributed Processing - Unified Modeling Language (*UML*)".

***UML*-модель** может включать в себя следующие аспекты;

- **Структурный аспект**: Use-Case-диаграммы, идентифицирующие бизнес-процессы и бизнес-транзакции, их взаимосвязь, соподчиненность и взаимодействие; Package-диаграммы, описывающие структуру предметной области и иерархическую структуру организации.
- **Динамический аспект**: Behavior-диаграммы (Activity, Statechart, Collaboration, Sequence), описывающие поведение (жизненный цикл) бизнес-процессов в их взаимодействии во времени и в пространстве с привязкой к используемым ресурсам и получаемым результатам.
- **Статический аспект**: Class-диаграммы, отражающие совокупность взаимосвязанных объектов, т.е. рассматривающие логическую структуру предметной области, ее внутренние концепции, иерархию объектов и статические связи между ними, **структуры данных и объектов**; Deployment-диаграммы, отражающие технологические ресурсы организации.

Проектирование ИС и БД

Инструментальные средства

На настоящий момент почти все СУБД поддерживают разработку физической модели схем баз данных с автоматической генерацией конечного кода - Microsoft Visual Studio, Oracle и т. д. Имеются также специальные модельные средства, поддерживающие кроме физической модели также и логическую. Одним из лидеров здесь является пакет Erwin компании Computer Associates. Концептуальные модели схем баз данных часто создаются в общих, универсальных UML-средах типа IBM Rational Rose.

Хранилища данных

Хранилище данных (*Data Warehouse*) — предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации. Строится на базе систем управления базами данных и систем поддержки принятия решений (СППР).

Данные, поступающие в хранилище данных, как правило, доступны только для чтения.

Системы поддержки принятия решений

Система поддержки принятия решений (СППР) (*Decision Support System, DSS*) — компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности. СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных

Принципы организации хранилища данных

- *Проблемно-предметная ориентация.* Данные объединяются в категории и хранятся в соответствии с областями, которые они описывают, а не с приложениями, которые они используют.
- *Интегрированность*. Данные объединены так, чтобы они удовлетворяли всем требованиям предприятия в целом, а не единственной функции бизнеса.
- *Некорректируемость*. Данные в хранилище данных не создаются: то есть поступают из внешних источников, не корректируются и не удаляются.
- *Зависимость от времени*. Данные в хранилище точны и корректны только в том случае, когда они

Хранилища данных

Достоинствами классического хранилища данных являются:

- общая семантика;
- централизованная, управляемая среда;
- согласованный набор процессов извлечения и бизнес-логики использования;
- непротиворечивость содержащейся информации;
- легко создаваемые по шаблонам и наполняемые витрины данных;
- единый репозиторий (хранилище) метаданных;
- многообразие механизмов обработки и представления данных.

К **недостаткам** можно отнести:

- большие затраты по реализации,
- высокую ресурсоемкость в масштабе всего предприятия,
- потребность в сложных сервисных системах,
- рискованный сценарий развития, когда все данные и метаданные находятся в одном репозитории и в неблагоприятном случае могут быть потеряны.

Кроме того, при фильтрации, агрегировании и рафинировании "сырых" данных для такого хранилища обычно теряется очень много информации, которая может быть чрезвычайно полезной при бизнес-анализе. В связи с этим возникло понимание того, что хранилище, помимо механизмов размещения и извлечения данных (OnLine Transactional Processing - OLTP), репозитория и витрин, должно иметь соответствующее пространство для организации "сырых" данных и их многомерного анализа в режиме реального времени (On LineAnalytical Processing - OLAP).

Хранилища данных

Источники данных для хранилища:

- Системы регистрации операций (БД)
- Отдельные документы
- Наборы данных

Источники данных классифицируются по:

- Территориальному и административному размещению
- Степени достоверности
- Частоте обновляемости
- СУБД

Хранилища данных

Операции с данными:

- Извлечение – перемещение информации от источников данных в отдельную БД, приведение их к единому формату.
- Преобразование – подготовка информации к хранению в оптимальной форме для реализации запроса, необходимого для принятия решений.
- Загрузка - помещение данных в хранилище, производится атомарно, путем добавления новых фактов или корректировкой существующих.
- Анализ - OLAP, Data Mining (интеллектуальный анализ данных), Reporting итд.
- Представление результатов анализа

Архитектура хранилищ данных

1. **Нормализованные хранилища**

- Данные находятся в предметно ориентированных таблицах третьей нормальной формы.
- **Недостатки**: большое количество таблиц как следствие нормализации, что приводит к ухудшению производительности системы.
- Для решения этой проблемы производительности используются денормализованные таблицы — **витрины данных**.
- При значительных объемах данных могут использовать несколько уровней «витрин»/ «хранилищ».

Архитектура хранилищ данных

1. Нормализованные хранилища

Витрины данных

1. **Нормализованные хранилища**

Витрина данных (*Data Mart*) — срез хранилища данных, представляющий собой массив тематической, узконаправленной информации, ориентированный, например, на пользователей одной рабочей группы или департамента.

Достоинства

- Аналитики работают только с теми данными, которые им реально нужны
- Целевая БД максимально приближена к конечному пользователю.
- Витрины данных обычно содержат тематические подмножества заранее агрегированных данных, их проще проектировать и настраивать.
- Для реализации витрин данных не требуется высокомощная вычислительная техника.

Недостатки

- мало контролируемая избыточность
- проблемы обеспечения целостности и непротиворечивости

Архитектура хранилищ данных

2. Хранилища с измерениями

- Данные копируются из систем **OLTP** и обрабатываются по технологии **OLAP**
- **Измерение (dimension)** - это множество объектов одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст числового показателя. Измерение принято визуализировать в виде ребра многомерного куба.

  Объекты, совокупность которых и образует измерение, называются членами измерений (members). Члены измерений визуализируют как точки или участи, откладываемые на осях гиперкуба.
- **Недостатки**: сложные процедуры подготовки и загрузки данных, а также управление и изменение измерений данных.

Принцип функционирования хранилищ данных

Данные из **OLTP**-системы копируются в хранилище данных таким образом, чтобы построение отчётов и **OLAP**-анализ не использовал ресурсы транзакционной системы и не нарушал её стабильность. Как правило, данные загружаются в хранилище с определённой периодичностью, поэтому актуальность данных может несколько отставать от **OLTP**-системы.

В **OLAP** применяется многомерное представление агрегированных данных для обеспечения быстрого доступа к стратегически важной информации в целях углубленного анализа.

Приложения **OLAP** должны обладать следующими основными свойствами:

- многомерное представление данных;
- поддержка сложных расчетов;
- правильный учет фактора времени.

Основные элементы OLAP

В основе **OLAP** лежит понятие гиперкуба, или многомерного куба данных, в ячейках которого хранятся анализируемые данные.

- **Факт** - это измеряемая (числовая) величина, которая располагается в ячейках гиперкуба. Один **OLAP**-куб может обладать одним или несколькими показателями.
- **Измерение (dimension)** - это множество объектов одного или нескольких типов, организованных в виде иерархической структуры и обеспечивающих информационный контекст числового показателя. Измерение принято визуализировать в виде ребра многомерного куба.
- **Ячейка (cell)** - атомарная структура куба, соответствующая полному набору конкретный значений измерений.
- **Иерархия** - группировка объектов одного измерения в объекты более высокого уровня. Например - день-месяц-год. Иерархии в измерениях необходимы для возможности агрегации и детализации значений показателей согласно их иерархической структуре. Иерархия целиком основывается на одном измерении и состоит из уровней.

Основные операции OLAP

В **OLAP**-системах поддерживаются следующие базовые операции:

- проекция. При проекции значения в ячейках, лежащих на оси проекции, суммируются по некоторому предопределенному закону;
- раскрытие (**drill-down**). Одно из значений измерения заменяется совокупностью значений из более низкого (следующего по иерархии) уровня иерархии измерения;
- свертка (**roll-up/drill-up**). Операция, обратная раскрытию;
- сечение (**slice-and-dice**).

Основные типы OLAP

MOLAP (Multidimensional OLAP)

- Детальные и агрегированные данные хранятся в многомерной базе данных. Хранение данных в многомерных структурах позволяет манипулировать данными как многомерным массивом, благодаря чему скорость вычисления агрегатных значений одинакова для любого из измерений. Однако в этом случае многомерная база данных оказывается избыточной, так как многомерные данные полностью содержат детальные реляционные данные.

Преимущества MOLAP

- Высокая производительность
- Структура и интерфейсы наилучшим образом соответствуют структуре аналитических запросов.
- Многомерные СУБД легко справляются с задачами включения в информационную модель разнообразных встроенных функций.

Недостатки MOLAP

- MOLAP могут работать только со своими собственными многомерными БД и основываются на патентованных технологиях для многомерных СУБД, поэтому являются наиболее дорогими
- Отсутствуют единые стандарты на интерфейс, языки описания и манипулирования данными.

Основные типы OLAP

ROLAP (Relational OLAP)

- ROLAP-системы позволяют представлять данные, хранимые в классической реляционной базе, в многомерной форме или в плоских локальных таблицах на файл-сервере, обеспечивая преобразование информации в многомерную модель через промежуточный слой метаданных. Агрегаты хранятся тоже в реляционной БД в специально созданных таблицах. В этом случае гиперкуб эмулируется СУБД на логическом уровне.

Преимущества ROLAP

- При оперативной аналитической обработке содержимого хранилища данных инструменты ROLAP позволяют производить анализ непосредственно над хранилищем.
- Реляционные СУБД обеспечивают высокий уровень защиты данных и хорошие возможности разграничения прав доступа.

Недостатки ROLAP

- Меньшая производительность, чем у MOLAP

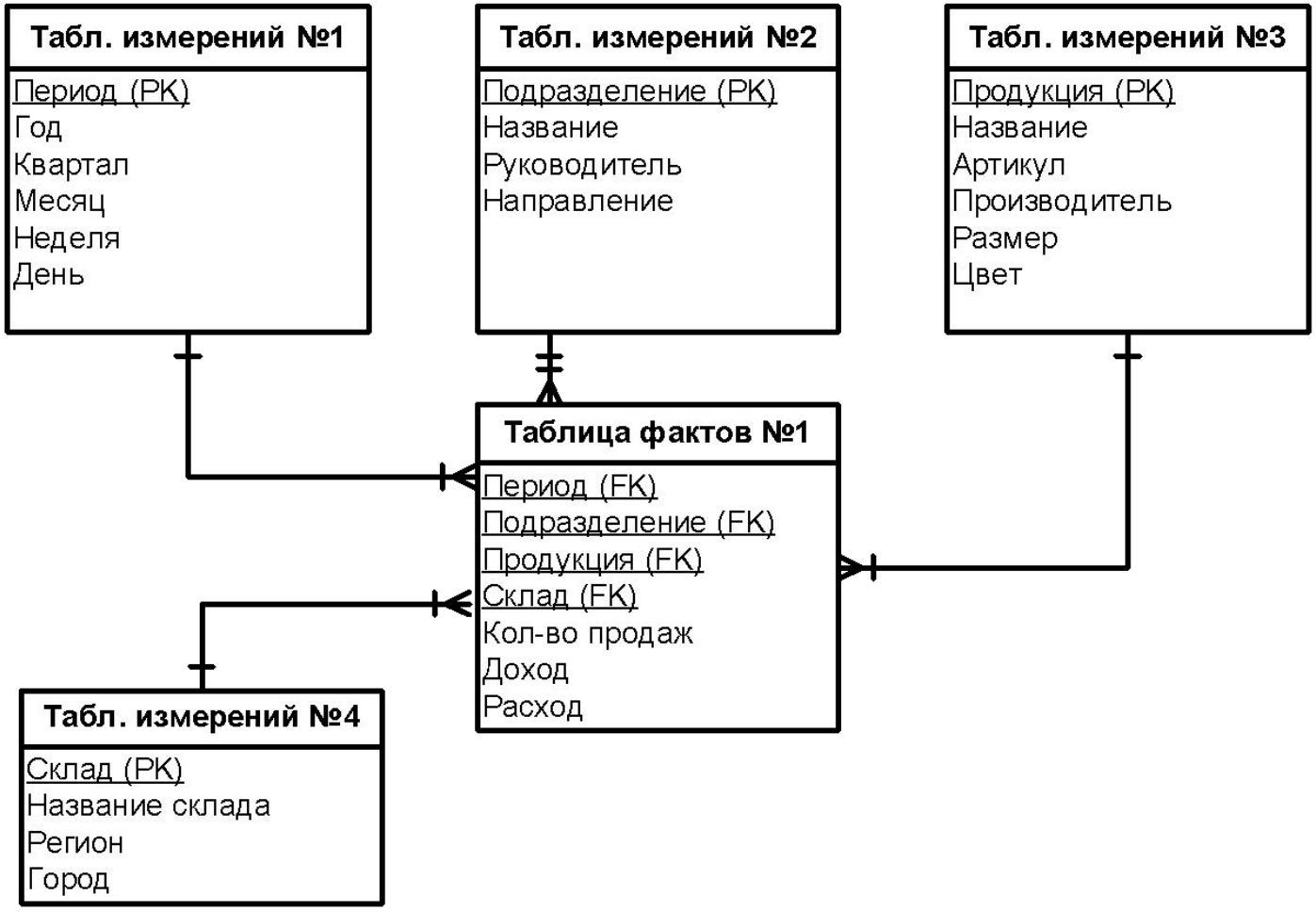
Основные типы OLAP

HOLAP (Hybrid OLAP)

- Детальные данные остаются в той же реляционной базе данных, где они изначально находились, а агрегатные данные хранятся в многомерной базе данных

Архитектура хранилищ данных

Хранилище с измерениями ROLAP (схема звезда)



Архитектура хранилищ данных

Хранилище с измерениями ROLAP (схема звезда)

Схема типа «звезды» - схема реляционной базы данных, служащая для поддержки многомерного представления содержащихся в ней данных

Преимущества

- Благодаря денормализации таблиц измерений упрощается восприятие структуры данных пользователем и формулировка запросов, уменьшается количество операций соединения таблиц при обработке запросов.

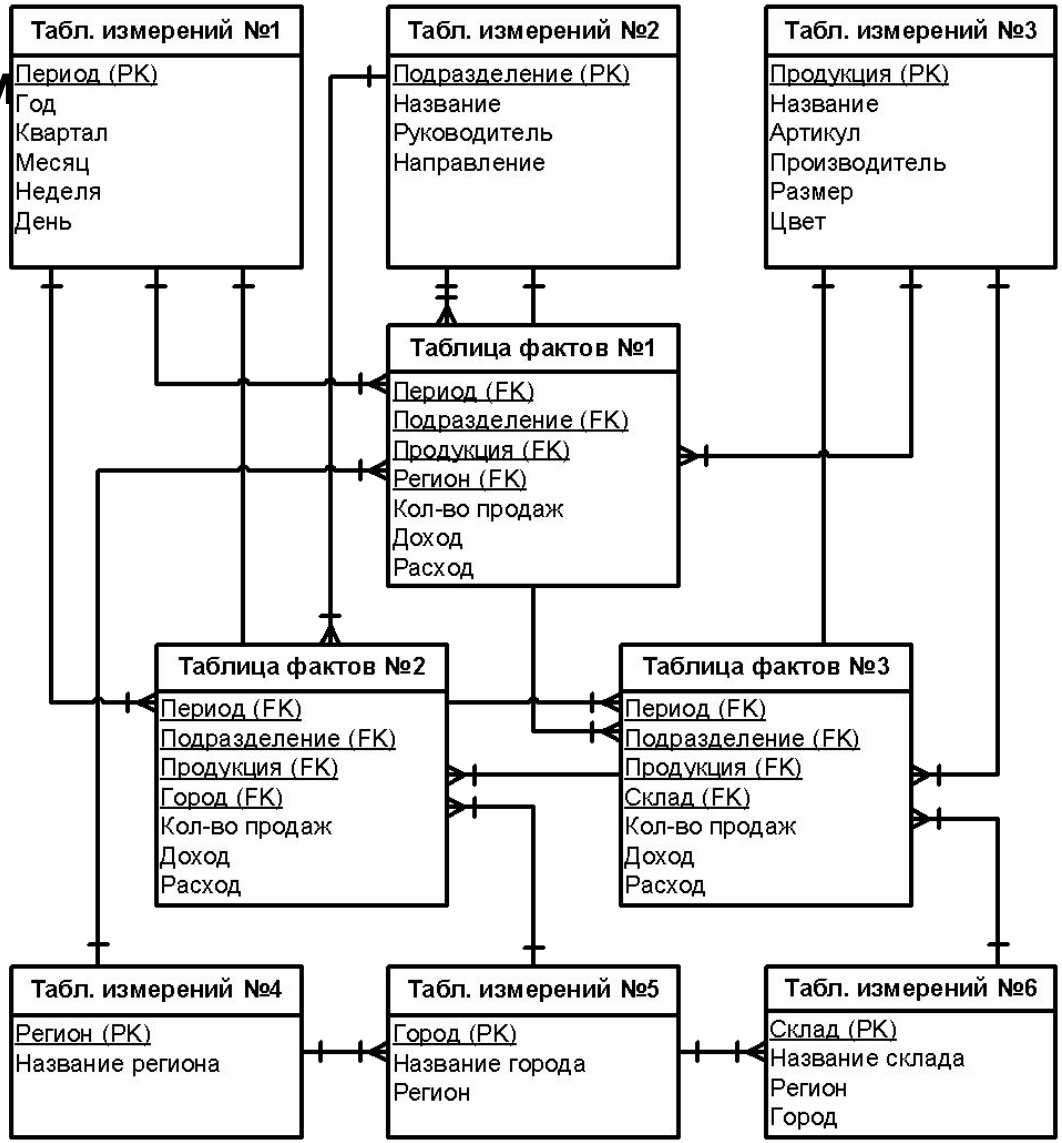
Недостатки

- Денормализация таблиц измерений вносит избыточность данных, возрастает требуемый для их хранения объем памяти. Если агрегаты хранятся совместно с исходными данными, то в измерениях необходимо использовать дополнительный параметр - уровень иерархии.

Архитектура хранилищ данных

Хранилище с измерениям

ROLAP (схема снежинка)



Архитектура хранилищ данных

Хранилище с измерениями ROLAP (схема снежинка)

Схема типа «снежинки» - схема реляционной базы данных, служащая для поддержки многомерного представления содержащихся в ней данных, является разновидностью схемы типа «звезда»

Преимущества

- Нормализация части таблиц измерений в отличие от схемы «звезда» позволяет минимизировать избыточность данных и более эффективно выполнять запросы.

Недостатки

- За нормализацию таблиц измерений иногда приходится платить временем выполнения запросов.

Big Data

Большие данные (*big data*) — серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети

Характеристики Big Data – VVVVV

- объём (*Volume)* в смысле величины физического объёма БД
- скорость (*Velocity)* в смыслах как скорости прироста, так и необходимости высокоскоростной обработки и получения результатов
- многообразие (*Variety)* в смысле возможности одновременной обработки различных типов структурированных и полуструктурированных данных

Характеристики Big Data – VVVVV

- достоверность данных (*Veracity*) в смысле возможности проведения достоверного анализа (например, разделение действий, проводимых роботом и человеком и др.)
- ценность накопленной информации (*Value*) Большие Данные должны быть полезны компании и приносить определенную ценность для нее (например, помогать в усовершенствовании бизнес-процессов, оптимизации расходов и т.д.)

Источники появления Big Data

- непрерывно поступающие данные с измерительных устройств
- события от радиочастотных идентификаторов
- потоки сообщений из социальных сетей
- метеорологические данные
- данные дистанционного зондирования Земли
- потоки данных о местонахождении абонентов сетей сотовой связи
- потоки данных устройств аудио- и видеорегистрации

Методы анализа Big Data

- ***Data Mining***: обучение ассоциативным правилам, классификация (методы категоризации новых данных на основе принципов, ранее применённых к уже наличествующим данным), кластерный анализ, регрессионный анализ;
- ***краудсорсинг*** — категоризация и обогащение данных силами широкого, неопределённого круга лиц;
- ***смешение и интеграция данных*** — набор техник, позволяющих интегрировать разнородные данные из разнообразных источников для возможности глубинного анализа (цифровая обработка сигналов и обработка естественного языка);
- ***машинное обучение***, включая обучение с учителем и без учителя;
- ***методы искусственного интеллекта: искусственные нейронные сети, сетевой анализ, оптимизация,*** **в том числе** ***генетические алгоритмы***;
- ***распознавание образов***;
- ***прогнозная аналитика***;
- ***имитационное моделирование***;
- ***пространственный анализ*** — класс методов, использующих топологическую, геометрическую и географическую информацию в данных;
- ***статистический анализ***;
- ***визуализация аналитических данных*** — представление информации в виде рисунков, диаграмм, с использованием интерактивных возможностей и анимации как для получения результатов, так и для использования в качестве исходных данных для дальнейшего анализа.

Технологии Big Data

- **NoSQL**
- **MapReduce**
- **Hadoop**
- **In-memory**
- **Columnar (колоночное сжатие)**
- **Log-файл аналитика**

Технологии Big Data

NoSQL свойства (BASE):

- базовая доступность (*basic availability*) — каждый запрос гарантированно завершается (успешно или безуспешно).
- гибкое состояние (*soft state*) — состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных.
- согласованность в конечном счёте (*eventual consistency*) — данные могут быть некоторое время рассогласованы, но приходят к согласованию через некоторое время.

Технологии Big Data

NoSQL типы хранилищ:

1. Хранилище «ключ-значение» - простейшее хранилище данных, использующее ключ для доступа к значению. Используются для хранения изображений, в качестве кэшей для объектов, а также в системах, спроектированных с прицелом на масштабируемость.
Примеры: Berkeley DB, MemcacheDB, Redis, Riak, Amazon DynamoDB.

Технологии Big Data

NoSQL типы хранилищ:

2. Хранилище семейств колонок - данные хранятся в виде разреженной матрицы, строки и столбцы которой используются как ключи

Примеры: Apache HBase, Apache Cassandra, Apache Accumulo, Hypertable, SimpleDB (Amazon.com)

Сценарии использования: системы управления содержимым, блоги, регистрация событий. Использование отметок времени (timestamp) позволяет использовать этот вид хранилища для организации счётчиков, а также регистрации и обработки различных данных во времени.

Технологии Big Data

NoSQL типы хранилищ:

3. Документо-ориентированная СУБД - служат для хранения иерархических структур данных. Находят своё применение в системах управления содержимым, издательском деле, документальном поиске и т. п. Примеры: CouchDB, MarkLogic, MongoDB, eXist, Berkeley DB XML.

Технологии Big Data

NoSQL типы хранилищ:

4. Базы данных на основе графов - применяются для задач, в которых данные имеют большое количество связей, например, социальные сети.

Примеры: Neo4j, OrientDB, AllegroGraph, Blazegraph, InfiniteGraph, FlockDB, Titan.

Так как рёбра графа являются хранимыми, обход графа не требует дополнительных вычислений (как JOIN), но для нахождения начальной вершины обхода требуется наличие индексов. Графовые базы данных имеют различные языки запросов: Gremlin, Cypher (Neo4j).

Технологии Big Data

NoSQL преимущества:

- Применение различных типов хранилищ.
- Возможность разработки базы данных без задания схемы.
- Линейная масштабируемость (добавление процессоров увеличивает производительность).
- Сокращение времени разработки
- Увеличение скорости обработки запросов

Технологии Big Data

MapReduce — модель распределённых вычислений, представленная компанией Google, используемая для параллельных вычислений над очень большими, несколько петабайт, наборами данных в компьютерных кластерах.

На **Map-шаге** происходит предварительная обработка данных. Для этого один из компьютеров (master node) получает входные данные задачи, разделяет их на части и передает другим компьютерам (worker node) для предварительной обработки.

На **Reduce-шаге** происходит свёртка предварительно обработанных данных. Главный узел получает ответы от рабочих узлов и на их основе формирует результат — решение задачи, которая изначально формулировалась.

Технологии Big Data

MapReduce преимущества:

- позволяет распределенно производить операции предварительной обработки и свертки. Операции предварительной обработки работают независимо друг от друга и могут производиться параллельно
- параллелизм также дает возможности восстановления после частичных сбоев серверов: если в рабочем узле, производящем операцию предварительной обработки или свертки, возникает сбой, то его работа может быть передана другому рабочему узлу (при условии, что входные данные для проводимой операции доступны)

Технологии Big Data

Hadoop — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов.

Используется для реализации поисковых и контекстных механизмов многих высоконагруженных веб-сайтов, в том числе, для Yahoo! и Facebook. Разработан на Java в рамках вычислительной парадигмы **MapReduce**, согласно которой приложение разделяется на большое количество одинаковых элементарных заданий, выполнимых на узлах кластера с интеграцией частных результатов в конечный результат.

Технологии Big Data

Hadoop

Состоит из четырёх модулей (по состоянию на 2014 г.):

- Hadoop Common (связующее программное обеспечение — набор инфраструктурных программных библиотек и утилит, используемых для других модулей и родственных проектов),
- HDFS (распределённая файловая система),
- YARN (система для планирования заданий и управления кластером)
- Hadoop MapReduce (платформа программирования и выполнения распределённых MapReduce-вычислений)

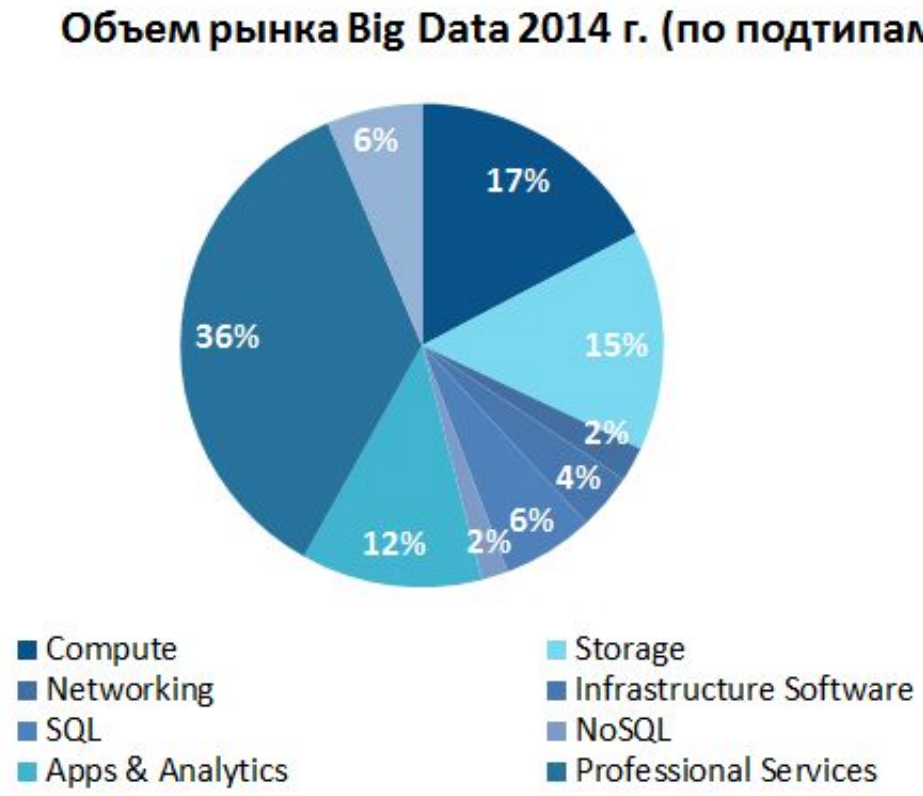
Технологии Big Data

Hadoop

Эксперты Allied Market Research обещают, что рынок решений для Hadoop в долгосрочной перспективе подрастет в 25 раз: с $2 млрд в 2013 г. до $50 млрд к 2020 г. "Утопая в данных, компании пытаются извлечь нужные, - рассказывает генеральный директор Platfora **Бен Вертер** (Ben Werther). - Руководители компаний охотно признают, что до сих пор слишком часто принимают решения интуитивно, поскольку не могут адекватно оценить собранные данные по причине отсутствия доступа и возможности интерпретировать большие пласты новых неоднородных данных достаточно быстро".

Platfora специализируется на разработке аналитических решений для работы с большими данными, получила в 2014 году инвестиции в размере $38 млн.

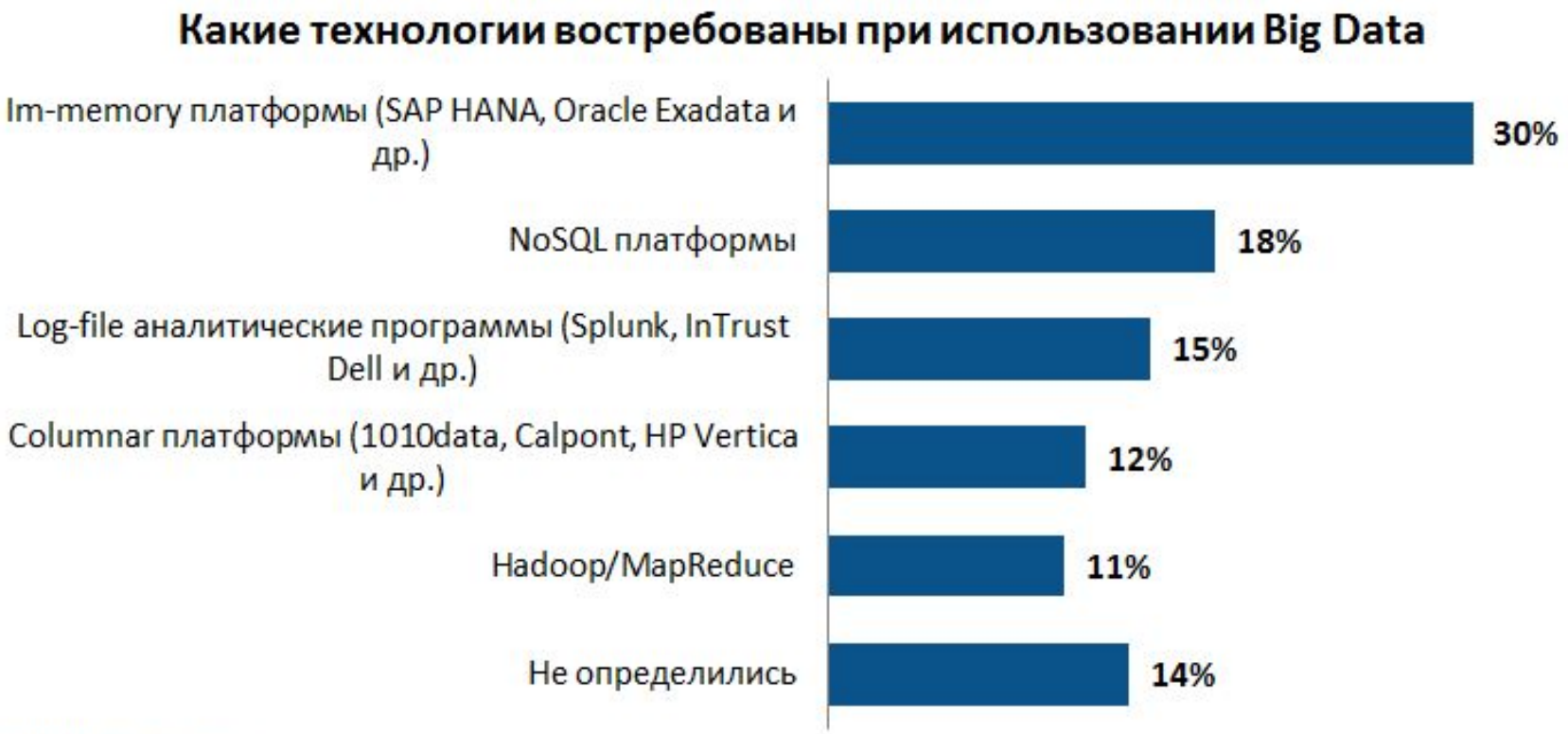
Big Data объем рынка



Источник: Wikibon

- приложения и аналитика составляет 36% выручки Big Data в 2014 году принесли приложения и аналитика Больших Данных, 17% — вычислительное оборудование и 15% — технологии хранения данных. Меньше всего выручки было сгенерировано NoSQL технологиями, инфраструктурным оборудованием и обеспечением сетью компаний

Big Data технологии

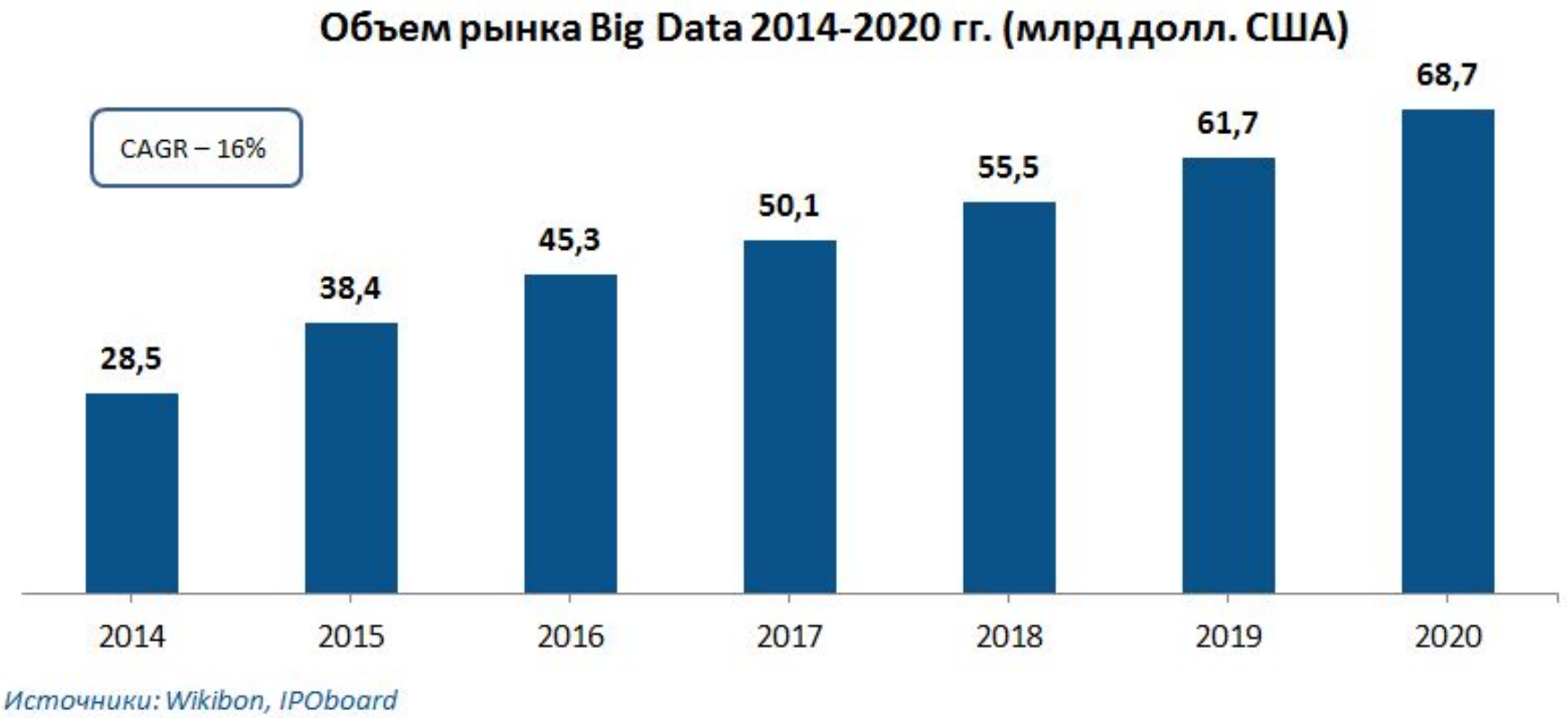


Источник: T-Systems

Big Data проблемы

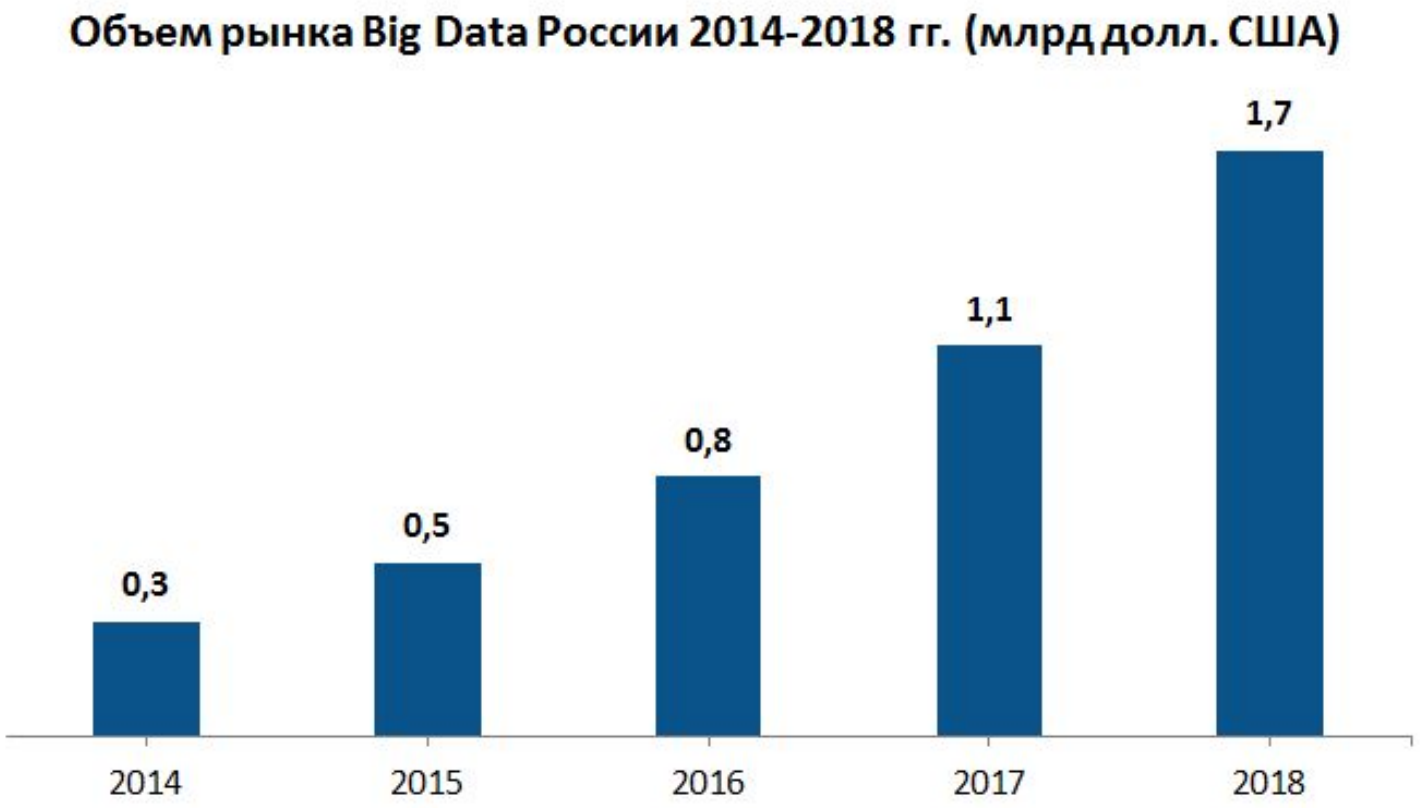
Источник: Accenture

Big Data прогноз



Источники: Wikibon, IPOboard

Dig Data прогноз Россия



Источники: IDC, IPOboard

Рынок Big Data России оценивается в 340 млн долл. США, из них 100 млн долл. США – решения SAP, остальные 240 млн долл. США – решения Oracle, IBM, SAS, Microsoft и др.

Объем хранимых данных в РФ ~ 2% от общемирового объема.

Dig Data примеры

- **Министерство труда Германии** использует Большие Данные при анализе поступающих заявок на выдачу пособий по безработице. Анализ информации выявил, что 20% пособий выплачивалось незаслуженно. Министерство сократило расходы на 10 млрд евро.
- **Детская больница Торонто** внедрила проект Project Artemis. Система ежесекундно отслеживает 1260 показателей состояния каждого ребенка, что позволяет прогнозировать нестабильное состояние ребенка и начать профилактику заболеваний.
- **Суперкомпьютер Watson компании IBM** анализирует в реальном времени поток данных по денежным транзакциям. По данным IBM, Watson на 15% увеличил количество выявленных мошеннических операций, на 50% сократил ложные срабатывания системы.
- Система **VISA** позволяет в автоматическом режиме вычислить операции мошеннического характера, что помогает предотвратить мошеннические платежи на сумму 2 млрд долл. США ежегодно.