



БД И СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ

ЛЕКЦИЯ 9



БЭКЕНД И БАЗА ДАННЫХ

- Требования к серверу вашего проекта определяют тип серверной архитектуры, которая является наиболее подходящей для него. Другие важные факторы, которые следует учитывать для создания идеального приложения, включают затраты и обязательные характеристики вашей архитектуры.
- Существует три класса серверной архитектуры. Это:
 - Бессерверный бэкенд.
 - Серверно-ориентированный бэкенд, и
 - Децентрализованный бэкенд.

БАЗА ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ

Почему база данных имеет важное значение?

- Конечно, все знания кодирования в мире не помогут вашему бизнесу, если вы работаете с некачественной базой данных. Несмотря на то, что база данных является частью бэкенда, то есть клиенты не могут взаимодействовать с ней напрямую, эффективность вашей базы данных в значительной степени влияет на их пользовательский опыт. Даже если они не видят этого, они узнают, когда у вас некачественная база данных.
- Существует множество вещей, которые могут привести к созданию плохой базы данных, например:
 - Слишком быстрое завершение создания базы данных командой разработчиков
 - нехватка ресурсов
 - Пренебрежение обслуживанием базы данных после ее создания
 - неопытность в создании баз данных

Медленный отклик

- Одним из плохих результатов плохой базы данных является медленное время загрузки. Когда страницы загружаются медленно, клиенты, скорее всего, уйдут с вашего сайта, не успев даже взглянуть на вашу веб-страницу.
- Даже если ваши страницы загружаются, клиенты могут испытывать трудности с вводом платежной информации или завершением покупки. Для сайта электронной коммерции медленная загрузка, по сути, делает сайт бесполезным. Клиенты не могут совершить покупку, и это тоже может их расстроить.

БАЗА ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ

- Показатели плохой базы данных
- К плохому дизайну базы данных может привести несколько причин, но, пожалуй, чаще всего это результат неопытности дизайнера базы данных, нереальных сроков разработки бэк-энда и базы данных, отсутствия надлежащего обслуживания базы данных и/или нехватки ресурсов.
- Это обычно приводит к проблемам, которые впоследствии ухудшают удобство использования сайта.

Плохая база данных замедляет работу

- Например, сайт может значительно замедлиться из-за плохой базы данных - страницы могут загружаться медленно, операции электронной коммерции могут обрабатываться слишком долго, а запросы пользователей на обновление данных, как в нашем примере выше, могут вообще не работать.

БАЗА ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ

Неисправные базы данных приводят к перебоям в работе сайта

- Хуже того, проблемы с базой данных могут даже привести к перебоям в работе системы, если закончится место для хранения данных. Такая проблема возникает в результате большого количества томов данных или какого-то другого ресурса, который становится недоступным (например, раздел, набор данных или какой-то пакет).

Неправильно спроектированные базы данных провоцируют ненужную модернизацию оборудования

- Когда такое происходит, нередко организации постоянно выбрасывают деньги на решение проблемы, пытаются устранить ее путем модернизации оборудования, например, серверов с большей памятью и объемом памяти.
- Хотя более мощный сервер может временно решить проблему, если проблемы с базой данных не решаются, дорогостоящая модернизация оборудования становится регулярной.
- Если ваша компания часто обновляет оборудование, это может указывать на более серьезную проблему в базе данных вашего сайта.

БАЗА ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ

- Перебои в работе сайта
- Плохо построенные базы данных неизбежно рано или поздно исчерпают свой объем. Когда это произойдет, весь ваш сайт может перейти в автономный режим или отключить некоторые важные функции, что сразу же заметят ваши клиенты. Предприятия часто совершают ошибку, пытаясь решить эту проблему путем модернизации серверов и систем хранения данных, но это временное и дорогостоящее решение реальной проблемы.
- Зачастую дешевле устранить истинную проблему: как функционирует база данных.
- Лучшее решение - начать с хорошо построенной базы данных, но даже если база данных вашей компании уже существует, некоторые модификации и решения в конечном итоге сделают ваш сайт более эффективным и менее дорогостоящим в долгосрочной перспективе.

БАЗА ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ

- Наличие хорошей базы данных означает, что пользователи смогут легко искать страницы на вашем сайте. Клиенты, которые начинают поиск товара или услуги с поиска, с гораздо большей вероятностью совершат покупку, и вы не захотите их отговаривать. Однако создание отличного внутреннего поиска на сайте может дорого обойтись вашей компании.
- Разработка бэкенда может показаться пугающей на первый взгляд, но вам не нужно знать все, чтобы последовательно формировать планы и принимать основанные на данных решения о том, как должна работать ваша ИТ-команда. Базы данных влияют на каждый аспект вашего сайта, поэтому их так важно поддерживать.

СЦЕНАРИИ ИСПОЛНЕНИЯ

- Что такое сценарии на стороне сервера?
- В статьях и обсуждениях, посвященных разработке внутренних компонентов веб-сайтов, часто можно услышать упоминание термина "серверные скрипты". Скрипты на стороне сервера - это важная часть того, что происходит за кулисами веб-сайта.
- Скрипты на стороне сервера - это то, что веб-разработчики используют для создания логики обработки данных веб-приложения. Пользователи сайта не видят эти скрипты, однако они оказывают на них непосредственное влияние через пользовательский опыт сайта.
- Опять же, скрипты на стороне сервера отвечают за передачу данных между сервером и базой данных, которые предоставляются внешним или внутренним приложением.

ПРИМЕР

Вот пример того, как это работает:

- Пользователь посещает веб-сайт и входит в свою учетную запись.
- Он обновляет фотографию своего профиля и адрес электронной почты.
- Как только пользователь вносит изменения, внешнее приложение передает эти данные сценариям на стороне сервера.
- Затем серверные скрипты передают обновленную информацию в конечное приложение и обрабатывают изменения на сервере.
- Наконец, информация о пользователе обновляется в базе данных, выполняя запрос пользователя.
- Сценарии на стороне сервера используются для получения любого типа данных, запрашиваемых пользователем, будь то выпадающие меню, радиокнопки, текстовые поля и так далее.

СЦЕНАРИИ ИСПОЛНЕНИЯ

- Скрипты на стороне сервера:
 - Могут быть активированы кодом сайта и выполняются на сервере
 - Передача информации от сервера к базе данных
 - Выполняют запросы пользователей, передавая данные между сервером сайта и браузером (аналогично нашему примеру выше).
 - Отвечает за работу динамических функций веб-приложений, таких как проверка пользователя, получение данных, навигация по страницам сайта.
 - Является основным игроком в том, как проектируется, разрабатывается и поддерживается база данных
 - Используется для создания интерфейсов прикладного программирования, или API.
- Серверные скрипты можно рассматривать как библиотеку вашего сайта.
- Представьте, что база данных сайта - это огромная, обширная библиотека. Когда пользователь посещает ваш сайт и запрашивает что-то конкретное, серверные скрипты действуют как библиотекарь, находя именно ту часть данных, которую он запрашивает, и предоставляя ее пользователю в браузере.
- В идеале все это происходит быстро, беспрепятственно и безопасно.

СЕРВЕРНО-ОРИЕНТИРОВАННЫЙ БЭКЕНД

- Серверные бэкенды могут быть размещены в облаке. Некоторые из преимуществ этой бэкенд-архитектуры заключаются в упрощенном обмене данными и беспроблемной интеграции с различными вычислительными устройствами. Поскольку эта архитектура централизована, она предлагает единую точку доступа, которая может находиться в вашем доме, офисе или любом другом месте.
- Серверно-ориентированная архитектура состоит из следующих трех уровней, которые мы рассматриваем ниже.
- **API:** Этот термин означает интерфейс прикладного программирования. Это больше похоже на промежуточную утилиту, которая позволяет разработчику добавлять внутренние функции на веб-сайт или приложение. Например, его можно использовать для интеграции в социальные сети, отправки сообщений, интеграции с функциями смартфона и многого другого. Однако преимущества и возможности API во многом определяются возможностями технологии, использующей сервис API.

СЕРВЕРНО-ОРИЕНТИРОВАННЫЙ БЭКЕНД

- **База данных:** Как следует из названия, база данных представляет собой хранилище большого объема данных, доступных пользователям и компьютерным программам. Базы данных создаются и управляются с использованием различных типов методов моделирования. Все типы технологий баз данных предлагают утилиты для создания и управления базой данных. Разработчики также могут создавать внешние приложения для управления данными в базе данных. Такие приложения взаимодействуют с базой данных с помощью набора команд, называемых запросами, для хранения новых данных или извлечения существующих.
- **Приложение:** Прикладной уровень состоит из компьютерных программ, которые отправляют различные запросы в базу данных. Как только база данных отвечает на запросы, она отображается в соответствующих разделах интерфейса приложения.

БЕССЕРВЕРНЫЙ БЭКЕНД

- Бессерверная бэкенд-архитектура имеет серверы, управляемые третьей стороной, которая занимается развертыванием и управлением серверами. По сути, бэкенд-провайдер берет на себя обслуживание сервера и масштабирование, когда возникает такая необходимость. Таким образом, разработчику не нужно беспокоиться о неприятностях с сервером. Серверный поставщик предлагает панель мониторинга приложений для управления функциями приложений.
- Этот тип архитектуры помогает разработчикам быстро обрабатывать проекты приложений. Запуск облачных приложений также является беспрепятственным, поскольку сторонние компании заботятся об управлении и масштабировании серверов. Помимо освобождения разработчика от проблем с управлением серверами, эксперты считают бессерверную архитектуру более безопасной. Причина проста: опытные бэкенд-провайдеры могут более эффективно решать проблемы безопасности.
- Поскольку архитектура бессерверного сервера основана на облачных технологиях, она более уязвима для атак. Поэтому необходимо проявлять особое внимание к этому, чтобы сделать сервер безопасным.

ДЕЦЕНТРАЛИЗОВАННЫЙ БЭКЕНД

- Этот тип архитектуры состоит из централизованной сети серверов, находящихся в различных физических местах. Поскольку нет центрального расположения, никто не берет на себя ответственность за управление бэкендом.
- Этот тип сети часто использует технологии открытого исходного кода, работающих на отдельных терминалах одноранговой сети. Две из самых популярных децентрализованных сетей- это Биткойн и Эфириум.
- Программисты могут использовать любой язык для создания внешнего и пользовательского интерфейса для такого рода бэкенда. Фронтенд-хранилище децентрализованной архитектуры может быть размещено в бэкенд-приложениях, таких как Swarm или IPFS.

БАЗЫ ДАННЫХ ДЛЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

- Начнем обзор с классических реляционных СУБД, плавно перейдем к NoSQL и завершим обзор решениями для кэширования. Я не поклонник MySQL и ничего хорошего или плохого я об этой базе данных не скажу.

POSTGRESQL

- PostgreSQL – это надежная реляционная СУБД корпоративного уровня с открытым исходным кодом и многолетней историей разработки. Все, что вы когда-либо хотели бы получить от реляционной базы данных, присутствует в PostgreSQL. Если вас беспокоит совместимость, обслуживание тысяч запросов из сотен таблиц, и максимальное использование SQL, правильно настроенный PostgreSQL отлично справится с этой задачей.
- Поскольку PostgreSQL полагается на стратегию масштабирования для операций записи или объемов данных, он должен максимально использовать доступные вычислительные ресурсы. PostgreSQL делает это с помощью различных стратегий индексирования и типов индексов, которые наилучшим образом соответствуют рабочей нагрузке запроса.

POSTGRESQL

- PostgreSQL поддерживает расширяемость множеством способов, такие как хранимые функции и процедуры, использование процедурных языков PL/PGSQL, Perl, Python и др.. Многие расширения предоставляют дополнительные функции, включая PostGIS – модуль для геопространственного анализа.
- Если у вас уже есть модель структурированных данных, то PostgreSQL будет лучшим вариантом.

КТО ПОЛЬЗУЕТСЯ POSTGRESQL

- Бэкенд-разработчики, которым приходится взаимодействовать с базами данных при работе с «серверной» частью сайта.
- Администраторы и разработчики баз данных — специалисты, основная задача которых заключается в обслуживании и поддержке работоспособности базы.
- DevOps-инженеры, обслуживающие рабочую инфраструктуру проекта.

ПРЕИМУЩЕСТВА POSTGRESQL

- Иногда СУБД называют бесплатным аналогом [Oracle Database](#). Обе системы адаптированы под большие проекты и высокую нагрузку. Но есть разница: они по-разному хранят данные, предоставляют разные инструменты и различаются возможностями. Важная особенность PostgreSQL в том, что эта система — feature-rich: так называют проекты с широким функционалом.
- **Объектно-реляционная модель.** Традиционно популярные СУБД — реляционные. Это значит, что данные, которые в них хранятся, представляются в виде записей, связанных друг с другом отношениями, — relations. Получаются связанные списки, которые могут иметь между собой те или иные отношения, — так и образуется таблица.
- Существует еще одна популярная модель — объектная. Данные представляются в виде объектов, их атрибутов, методов и классов. Объектная модель поддерживает возможности, о которых мы подробно рассказывали в [статье про ООП](#), например наследование.
- PostgreSQL — объектно-реляционная СУБД. Это значит, что она поддерживает и объектный, и реляционный подход.
- **Поддержка множества типов данных.** Еще одна особенность PostgreSQL — поддержка большого количества типов записи информации. Это не только стандартные целочисленные значения, числа с плавающей точкой, строки и булевы значения («да/нет»), но и денежный, геометрический, перечисляемый, бинарный и другие типы. PostgreSQL «из коробки» поддерживает битовые строки и сетевые адреса, массивы данных, в том числе многомерные, композитные типы и другие сложные структуры. В ней есть поддержка [XML](#), [JSON](#) и NoSQL-баз.
- При необходимости к СУБД можно подключить поддержку типов данных, которые нужны в конкретном проекте. В PostgreSQL есть несколько внутренних форматов, которые используются только в ней.

ПРЕИМУЩЕСТВА POSTGRESQL

- **Поддержка сложных запросов.** PostgreSQL работает со сложными, составными запросами. Система справляется с задачами разбора и выполнения трудоемких операций, которые подразумевают и чтение, и запись, и валидацию одновременно. Она медленнее аналогов, если речь заходит только о чтении, но в других аспектах превосходит конкурентов.
- **Написание функций на нескольких языках.** В PostgreSQL можно писать собственные функции — пользовательские блоки кода, которые выполняют те или иные действия. Эта возможность есть практически в любых СУБД, но PostgreSQL поддерживает больше языков, чем аналоги. Кроме стандартного SQL, в PostgreSQL можно писать на C и C++, Java, Python, PHP, Lua и Ruby. Он поддерживает V8 — один из движков JavaScript, поэтому JS тоже можно использовать совместно с PgSQL. Реализована поддержка Delphi, Lisp и прочих редких языков. При необходимости можно расширить систему под другие ЯП.
- Модификация SQL, которая используется в PostgreSQL, называется PL/pgSQL. Это процедурное расширение, которое поддерживает сложные вычисления и дополняет «классический» SQL новыми возможностями.
- **Одновременная модификация базы.** Важная особенность PostgreSQL — возможность одновременного доступа к базе с нескольких устройств. В СУБД реализована клиент-серверная архитектура, когда база данных хранится на сервере, а доступ к ней осуществляется с клиентских компьютеров. Так, например, реализуются разнообразные сайты. Одна из возможных сложностей — ситуация, когда несколько человек одновременно модифицируют базу и нужно избежать конфликтов.
- В PostgreSQL для этого используется технология MVCC — Multiversion Concurrency Control, многоверсионное управление параллельным доступом. Каждый пользователь получает снимок — «снимок» базы, в который вносятся изменения. Только после фиксации транзакции они поступают в исходную базу данных. Пока человек вносит изменения, они не видны другим пользователям. Конфликтов не возникает, как и необходимости блокировать чтение или запись.

ПРЕИМУЩЕСТВА POSTGRESQL

- **Соответствие ACID.** ACID — это набор принципов для обеспечения целостности данных. Аббревиатура расшифровывается как Atomicity, Consistency, Isolation, Durability — атомарность, согласованность, изолированность, прочность. Если база данных соответствует этим принципам, она ведет себя максимально предсказуемо и надежно. В ней низок риск конфликта или непредвиденного поведения системы.
- PostgreSQL соблюдает требования ACID благодаря технологии MVCC. Это делает систему надежной и безопасной в использовании, а данные — защищенными от возможных сбоев, ошибок и потерь.
- **Возможность расширения.** Разработчик может написать для СУБД собственные типы и их преобразования, операции и функции, ограничения и индексы, собственный процедурный язык для запросов. PostgreSQL можно модифицировать практически под любую нестандартную задачу.
- **Высокая мощность и широкая функциональность.** PostgreSQL — возможно, единственная бесплатная СУБД с открытым исходным кодом, которая рассчитана на работу с объемными и сложными проектами. Она мощная, производительная, способна эффективно работать с большими массивами данных. Есть примеры реального использования СУБД для баз данных в несколько петабайт с сотнями тысяч запросов в секунду. На главной странице [официального сайта PostgreSQL](#) называют «самой продвинутой бесплатной СУБД». Система действительно имеет высокую функциональность и не уступает платным продуктам.

ПРЕИМУЩЕСТВА POSTGRESQL

- **Открытость.** PostgreSQL — ПО с открытым исходным кодом, которое распространяется по свободной лицензии. Это означает, что любой разработчик может посмотреть, как написана система, или предложить для нее свои правки. СУБД разрабатывается сообществом энтузиастов и в определенной степени никому не принадлежит, а значит, ее можно свободно и без ограничений использовать в своих проектах.
- На базе PostgreSQL существуют коммерческие продукты с платным доступом — ими обычно пользуются крупные компании, которым нужна дополнительная функциональность. Это, например, связь с Oracle Database или продвинутый веб-интерфейс для администрирования БД.
- **Минимальное количество багов.** PostgreSQL — проект, который известен высоким качеством отладки. Каждая версия системы появляется в доступе только после полной проверки, поэтому СУБД очень стабильна. Частая проблема бесплатных проектов — новые версии с большим количеством багов, но в случае с PostgreSQL такой проблемы нет.
- Согласно независимым автоматизированным исследованиям, в исходном коде СУБД есть одна ошибка на 39 000 строк кода. Это в пять раз меньше, чем в [MySQL](#), и в пятьдесят раз меньше, чем в ядре операционной системы [Linux](#).
- **Кроссплатформенность.** Чаще всего PostgreSQL используют на серверах с операционными системами семейства Linux, но СУБД поддерживает и другие ОС. Ее можно установить в системы на базе Windows, BSD, macOS и Solaris. Кроме того, у PostgreSQL есть автономный веб-сервер PostgREST, с которым можно работать с помощью [REST API](#). СУБД можно развернуть и в облаке.

MONGODB

- MongoDB – это система управления базами данных, ориентированная на документы, и в настоящее время она является самым популярным на рынке решением NoSQL.
- В отличие от реляционных баз данных, MongoDB основана на наборе документов, каждый из которых состоит из атрибутов ключ/значение. Отдельный документ можно рассматривать как эквивалент строки в таблице, где каждый ключ аналогичен имени столбца, а значение каждого ключа аналогично соответствующему значению строки. Основное отличие состоит в том, что документ не ограничен определенной схемой или столбцами в таблице. Два документа могут иметь одинаковые элементы, вроде поля идентификатора, а также иметь совершенно разные элементы.
- Поскольку MongoDB допускает динамическое изменение схемы, их легко вносить без необходимости переделывать существующую структуру базы данных. Кроме того, иерархия документов легко сопоставляется с иерархиями объектов в коде приложения, упрощая операции создания, чтения, обновления и удаления.

MONGODB

Коллекция

- Коллекция — набор документов MongoDB. Она представляет собой эквивалент таблицы в РСУБД. Коллекция размещается внутри одной базы данных и не требует описания схемы. Документы в коллекции могут иметь разные поля, и, как правило, их объединяет схожее или общее назначение.

Документ

- Документ — это набор пар ключ-значение. Он обладает динамической схемой — документы одной коллекции необязательно должны иметь одинаковый набор полей или структуру, а их общие поля могут содержать разные типы данных.

СРАВНЕНИЕ

РСУБД	MongoDB
База данных	База данных
Таблица	Коллекция
Кортеж/строка	Документ
Столбец	Поле
Соединение таблиц	Встроенные документы
Первичный ключ	Первичный ключ (Ключ по умолчанию <code>_id</code> , предоставляемый MongoDB)
Сервер и клиент базы данных	
mysqld/Oracle	mongod
mysql/sqlplus	mongo

ПРИМЕР ДОКУМЕНТА

```
{
  _id: ObjectId(7df78ad8902c)
  title: 'MongoDB Overview',
  description: 'MongoDB is no sql database',
  by: 'Nuances of programming',
  url: 'https://nuancesprog.ru/',
  tags: ['mongodb', 'database', 'NoSQL'],
  likes: 100,
  comments: [
    {
      user: 'user1',
      message: 'My first comment',
      dateCreated: new Date(2011,1,20,2,15),
      like: 0
    },
    {
      user: 'user2',
      message: 'My second comments',
      dateCreated: new Date(2011,1,25,7,45),
      like: 5
    }
  ]
}
```

ПРЕИМУЩЕСТВА MONGODB

Преимущества MongoDB перед РСУБД

- Отсутствие схемы. MongoDB — это документоориентированная база данных, в которой одна коллекция содержит различные документы. Количество полей, содержимое и размер документов могут отличаться.
- Понятная структура объекта.
- Отсутствие сложных соединений Join.
- Расширенные возможности запросов. MongoDB поддерживает динамические запросы документов, используя язык запросов на их основе, практически не уступающий в эффективности SQL.
- Настройка.
- Легкость масштабирования.
- Не требуется преобразование объектов приложения в объекты базы данных.
- Использует внутреннюю память для хранения (оконного) рабочего набора, обеспечивая ускоренный доступ к данным.

ПРЕИМУЩЕСТВА MONGODB

Аргументы в пользу MongoDB

- Документоориентированное хранение данных. Данные хранятся в виде документов в формате JSON.
- Индекс для любого атрибута.
- Репликация и высокая доступность.
- Автоматический шардинг.
- Полнофункциональные запросы.
- Быстрые обновления.
- Профессиональная поддержка, предоставляемая MongoDB.

Области использования MongoDB

- Большие данные.
- Управление и доставка контента.
- Мобильная и социальная инфраструктуры.
- Управление пользовательскими данными.
- Центр управления и обработки данных (Data Hub).

POSTGRESQL VS MONGODB

- Различные тесты показали, что PostgreSQL превосходит MongoDB в рабочих нагрузках хранилищ данных и анализа данных. Однако при сравнении операций JSON между PostgreSQL и MongoDB есть тесты, которые показывают преимущества для обеих баз данных.
- Можно посмотреть различные бенчмарки сравнивающие эти две базы данных, и каждый хвалит свое решение. Если же подойти в вопросу с технической стороны, то PostgreSQL использует бинарные деревья для индексирования, которые отлично работают на HDD. В тоже время большинство современных баз данных (RocksDB, MongoDB, Tarantool и т.д.) используют LSM-дерево (Log-structured merge-tree – журнально структурированное дерево со слиянием) вместо классического B-Tree.
- Бинарное дерево ориентировано на операции чтения и поиск, дерево LSM ориентировано на доступ к часто изменяемым данным и работу с SSD.

POSTGRES VS MONGODB

- Система управления базами данных Postgres в 4–15 раз быстрее MongoDB при тестировании производительности транзакций, проведенном OnGres – компанией, специализирующейся на предоставлении программного обеспечения и услуг для баз данных и спонсируемой EnterpriseDB.
- С одной стороны, Postgres умеет вставлять записи в таблицы без индексов практически с линейной скоростью записи на диск (сотни мегабайт в секунду). Однако если таблица содержит индексы, а вставленные ключи имеют случайные значения, тогда мы наблюдаем резкое снижение производительности.
- PostgreSQL представляет классический подход к хранению структурированных данных в виде таблиц. С другой стороны, MongoDB требует от разработчика переосмысления подхода к хранению данных. Конечно, эта СУБД не дает полной свободы действий, но предоставляет дополнительный инструмент, повышающий производительность базы данных – коллекции.

ORACLE

- Oracle - это наиболее широко используемая коммерческая система управления реляционными базами данных, встроенная в такие языки сборки, как C, C++ и Java. Последняя версия этой базы данных, 21c, содержит множество новых функций.
- Oracle - это система управления базами данных, которая стоит выше других. В целом, это наиболее широко используемая СУБД. Она занимает меньше места и быстрее обрабатывает данные, а также включает несколько новых полезных функций, таких как JSON из SQL.

ORACLE

- **По модели данных — объектно-реляционная**
- Это значит, что система объединяет в себе две модели хранения информации: объектно-ориентированную и реляционную.
- Реляционная модель представляется как набор отношений между записями. Одни данные связаны с другими — так формируется база. Визуально ее можно представить как двумерную таблицу; математически — как модель, построенную на отношениях.
- Объектно-ориентированная модель воспринимает данные как объект. У объекта есть атрибуты, которые описывают его свойства, и методы — они нужны для взаимодействия с другими объектами. Каждый объект принадлежит к классу — это понятие можно представить как «схему» объекта. Объектно-ориентированный подход используется во многих языках программирования и упоминается во многих наших статьях. А в базах данных он нужен для работы с данными, у которых сложная структура.
- Oracle Database работает и с объектно-ориентированной, и с реляционной моделью.

ORACLE

- Система работает по принципу «клиент — сервер». Это означает, что ее основная часть размещается на сервере, там же, где и база данных. Человек работает с интерфейсом приложения-клиента. Клиентская часть управляет только пересылкой и получением информации от сервиса.
- Связь между клиентом и сервером обеспечивает специальный компонент, который в Oracle называется SQL *Net.
- Достоинства такого подхода — в высоком уровне безопасности и легком доступе для клиентов. Клиент-серверная организация разгружает сеть и снимает вычислительную нагрузку с клиентских компьютеров. А вот сервер для такой СУБД должен быть мощным.
- Полная документация по СУБД называется Oracle Concepts. Для прежних версий системы она есть на русском языке, для новых — на английском.

КАК РАБОТАЕТ ORACLE DATABASE

- Информация в системе хранится в отдельных базах — инстансах (instance) или экземплярах БД. Это не физические, а логические понятия, которые состоят из процессов и оперативной памяти. Все содержимое одного экземпляра имеет единую системную глобальную область (SGA) — часть оперативной памяти, с которой работает.
- Внутри экземпляров расположены логические пространства, которые называются табличными — tablespaces. Табличные пространства содержат компоненты данных — как файлы в папках. Файлы имеют расширение .dbf.
- СУБД состоит из одного или нескольких инстансов и программного обеспечения, которое ими управляет. Система поддерживает работу с независимыми базами (PDB) в рамках одного инстанса. Она может работать и с мультиарендной архитектурой (CDB), где множественными клиентами управляет один экземпляр приложения. В Oracle поддерживаются кластеризация и секционирование — физическое разделение элементов баз данных без потери доступа.

ORACLE

- Самая новая версия СУБД — 21c. Она включает больше 200 инноваций, в том числе поддержку неизменяемых блокчейн-таблиц, поддержку JavaScript, встроенную в СУБД, поддержку типа данных JSON в бинарном формате и другие.
- Для актуальных версий временами появляются обновления и пакеты патчей, которые называются PSU. Они расширяют возможности СУБД. Кроме того, Oracle предлагает использовать систему вместе с другими платными продуктами корпорации. Это Oracle Server, Oracle RPAS и другие решения в области информационных технологий. Для обучения можно пользоваться специальной бесплатной версией Express Edition (XE) или специальным облачным набором Always Free.

MYSQL

- MySQL - одна из самых узнаваемых технологий в современной экосистеме больших данных. Ее часто называют самой популярной базой данных, и в настоящее время она широко и эффективно используется вне зависимости от отрасли, поэтому очевидно, что каждый, кто занимается корпоративными данными или общими информационными технологиями, должен, по крайней мере, стремиться к базовому знакомству с MySQL.
- С MySQL даже новички в реляционных системах могут сразу же создавать быстрые, мощные и безопасные системы хранения данных. Программный синтаксис и интерфейсы MySQL также являются идеальными шлюзами в широкий мир других популярных языков запросов и структурированных хранилищ данных.
- MySQL - это система управления реляционными базами данных (RDBMS), разработанная компанией Oracle и основанная на языке структурированных запросов (SQL).
- MySQL является неотъемлемой частью многих наиболее популярных программных стеков для создания и поддержки всего - от веб-приложений, ориентированных на клиента, до мощных, ориентированных на данные, сервисов B2B. Благодаря открытому исходному коду, стабильности и богатому набору функций, а также постоянному развитию и поддержке со стороны Oracle, MySQL используется в таких важных для интернета организациях, как Facebook, Flickr, Twitter, Wikipedia и YouTube.

ОСОБЕННОСТИ MYSQL

- Поскольку MySQL имеет самое широкое применение во многих отраслях, бизнес-пользователи, от начинающих веб-мастеров до опытных менеджеров, должны стремиться понять ее основные характеристики. Принятие решения об использовании этой технологии и эффективное информирование о ней начинается с обзора базовой доступности, структуры, философии и удобства использования MySQL.

MySQL широко совместима

- Хотя MySQL часто ассоциируется с интернет-приложениями или веб-сервисами, она была разработана для широкой совместимости с другими технологиями и архитектурами. РСУБД работает на всех основных вычислительных платформах, включая операционные системы на базе Unix, такие как многочисленные дистрибутивы Linux или Mac OS, и Windows.
- Клиент-серверная архитектура MySQL означает, что она может поддерживать различные бэкенды, а также различные интерфейсы программирования. Данные могут быть напрямую перенесены из MySQL в его развитые версии (например, MariaDB), а также в большинство других РСУБД благодаря сходству архитектуры и языка.
- Существующие инструменты миграции Oracle и сторонних производителей позволяют MySQL перемещать данные в и из обширного набора общих систем хранения данных, независимо от того, предназначены ли они для локального или облачного использования. MySQL может быть развернут в виртуализированных средах, распределенных или централизованных, и даже существует в виде портативных автономных библиотек для учебных целей, тестирования или небольших приложений.
- Широкая совместимость MySQL со всеми этими другими системами и программным обеспечением делает ее особенно практичным выбором РСУБД в большинстве ситуаций.

ОСОБЕННОСТИ MYSQL

Базы данных MySQL являются реляционными

- Основной фактор, отличающий реляционные базы данных от других цифровых хранилищ, заключается в том, как данные организованы на высоком уровне. Базы данных типа MySQL содержат записи в нескольких, отдельных и высоко кодифицированных таблицах, в отличие от одного всеобъемлющего хранилища или коллекций полу- или неструктурированных документов.
- Это позволяет РСУБД лучше оптимизировать такие действия, как поиск данных, обновление информации или более сложные действия, такие как агрегирование. Логическая модель определяется над всем содержимым базы данных, описывая, например, значения, допустимые в отдельных столбцах, характеристики таблиц и представлений или то, как связаны индексы из двух таблиц.
- Реляционные модели остаются популярными по нескольким причинам. Они предоставляют пользователям интуитивно понятные, декларативные языки программирования - по сути, они сообщают базе данных о желаемом результате на языке, сходном или, по крайней мере, понятном с английским, вместо того, чтобы тщательно кодировать каждый шаг процедуры, ведущей к этому результату. Это переносит большую часть работы на РСУБД и SQL-движки, обеспечивая лучшее соблюдение логических правил и экономя ценные ресурсы и рабочую силу.

ОСОБЕННОСТИ MYSQL

У MySQL открытый исходный код

- Любой человек или предприятие может свободно использовать, изменять, публиковать и расширять кодовую базу MySQL от Oracle с открытым исходным кодом. Программное обеспечение выпускается под лицензией GNU General Public License (GPL).
- Если код MySQL необходимо интегрировать или включить в коммерческое приложение (или если программное обеспечение с открытым исходным кодом не является приоритетным), предприятия могут приобрести у Oracle коммерчески лицензированную версию.
- Опять же, эти варианты предоставляют организациям дополнительную гибкость при принятии решения о работе с MySQL. Публичный и основанный на сообществе характер релизов с открытым исходным кодом обогащает документацию MySQL и культуру онлайн-поддержки, а также гарантирует, что поддерживаемые или вновь разрабатываемые возможности никогда не отклоняются слишком далеко от текущих потребностей пользователей.

MySQL прост в использовании

- Хотя реляционная природа MySQL и вытекающие из нее жесткие структуры хранения данных могут показаться ограничивающими, табличная парадигма, пожалуй, наиболее интуитивно понятна и в конечном итоге обеспечивает большее удобство использования.
- На самом деле, MySQL идет на многие уступки в поддержке максимально возможного разнообразия структур данных, от стандартных, но богатых логических, числовых, буквенно-цифровых, дат и временных типов до более продвинутых JSON или геопространственных данных. Помимо простых типов данных и обширного набора встроенных функций, экосистема MySQL также включает в себя множество инструментов, облегчающих все - от управления сервером до составления отчетов и анализа данных.
- Независимо от общей архитектуры РСУБД, пользователи неизменно находят в MySQL функции, позволяющие моделировать и кодировать данные по своему усмотрению. MySQL остается одной из самых простых в освоении и использовании технологий баз данных.

MYSQL

SQL и MySQL: В чем разница?

- Реляционная модель была впервые описана в 1970 году в работе Эдгара Ф. Кодда. Один из первых коммерческих языков программирования, связанных с этой моделью, SQL, был разработан вскоре после этого в компании IBM. В течение некоторого времени SQL был наиболее широко используемым языком баз данных, принятым в качестве стандарта ANSI в 1986 году, а годом позже - в ISO.

SQL состоит из четырех подязыков, каждый из которых имеет свою область применения.

- DQL: Язык запросов к данным (DQL) является наиболее знакомым и используется для выполнения запросов к базам данных и извлечения информации из хранимых данных. Например, выбор и возврат максимального значения в столбце.
- DDL: Язык определения данных (DDL) используется для кодирования определенных структур и схем базы данных. Примером может служить создание таблицы или определение типов данных.
- DCL: язык управления данными (DCL) определяет доступ, полномочия и разрешения для пользователей и процессов, обращающихся к базе данных, включая предоставление прав администратора или ограничение пользователей только правами чтения.
- DML: И, наконец, язык манипулирования данными (DML) используется для внесения изменений в существующие компоненты базы данных, таких как вставка записей, обновление значений в ячейках или удаление данных.

Шведская компания MySQL AB впервые выпустила MySQL в 1995 году. Как и большинство программного обеспечения для баз данных, которое последовало за начальным подъемом реляционных систем, MySQL является просто расширением оригинального стандарта SQL, добавляя больше возможностей, поддержки, процедурного программирования, механизмов потока управления и многое другое.

MYSQL

- Изначально MySQL был задуман для управления массивными базами данных, быстрее, чем существующее программное обеспечение для баз данных. Используемый в сложных операционных, транзакционных и производственных средах на протяжении десятилетий, MySQL развивался вместе с перемещением вычислений и хранения данных в облако.
- Хотя MySQL обычно устанавливается на отдельных машинах, сейчас она включает в себя глубокую поддержку распределенных приложений и включена в большинство облачных платформ данных.
- По сравнению со многими решениями для хранения и обработки данных, представленными сегодня на рынке, MySQL является более старой технологией, но она не показывает признаков снижения популярности или полезности. На самом деле, в последнее время MySQL переживает возрождение по сравнению даже с более специализированными современными системами хранения данных благодаря своей скорости, надежности, простоте использования и широкой совместимости.

MS SQL

- Microsoft SQL Server - одна из основных реляционных систем управления базами данных на рынке, которая обслуживает широкий спектр программных приложений для бизнес-аналитики и анализа в корпоративной среде.
- Основанная на языке Transact-SQL, она включает в себя набор стандартных расширений программирования языка, а ее приложение доступно для использования как в помещениях, так и в облаке.

Для чего именно предназначен Microsoft SQL Server?

- Microsoft SQL Server идеально подходит для хранения всей необходимой информации в реляционных базах данных, а также для управления такими данными без осложнений, благодаря визуальному интерфейсу и имеющимся в нем опциям и инструментам. Это жизненно важно, особенно для веб-сайтов, на которых есть возможность регистрации пользователей для входа в систему.
- Для компаний использование этого инструмента крайне важно благодаря предлагаемым им возможностям и утилитам. Если у вас есть список клиентов, каталог продукции или даже большой выбор мультимедийного контента, Microsoft SQL Server поможет управлять абсолютно всем. Он необходим для правильного функционирования веб-сайта или любого приложения.
- Его основной компонент состоит из реляционного движка, отвечающего за обработку команд, запросов, а также за хранение файлов, бб.дд, таблиц и буферов данных. Его вторичные уровни предназначены для управления памятью, программирования и администрирования взаимодействия запросов и ответов с серверами, на которых размещены базы данных.

ФУНКЦИИ И ОСОБЕННОСТИ MICROSOFT SQL SERVER:

- Одними из основных функций, отличающих Microsoft SQL Server, являются его разнообразные инструменты, направленные на управление и анализ данных, а также бизнес-аналитика, с помощью которой можно получить представление о вашем бизнесе и клиентах при поддержке машинного обучения.
- Microsoft SQL Server позволяет легко интегрировать данные в приложения и воспользоваться широким набором когнитивных сервисов для использования искусственного интеллекта в любых масштабах данных, как в локальных, так и в облачных средах благодаря интеграции с Azure AI.
- В целом, SQL Servers предлагает пользователю высокую доступность для обеспечения более быстрых процессов переключения. Его возможности in-memory позволяют повысить гибкость и простоту использования благодаря бесшовной интеграции с семейством серверов Microsoft Server.
- Основанный на открытом исходном коде, он очень легко доступен, и подавляющее большинство программистов, работающих в сфере веб-разработки, использовали Microsoft SQL Server в некоторых своих проектах, а также имеет большое сообщество, предлагающее поддержку другим пользователям.

ОСОБЕННОСТИ MS SQL

- Анализ всех ваших данных с помощью кластеров Big Data: возможность запросить весь массив данных от SQL Server до Oracle без репликации.
- Выбор языка и платформы: от Windows или Linux до развертывания Kubernetes.
- Интеллектуальные возможности баз данных: в памяти, поддержка постоянной памяти, оптимизированный для работы в памяти tempdb.
- Шифрование данных и соответствие нормативным требованиям: Система защиты, мониторинга и классификации данных позволила этой платформе в течение 9 лет входить в топ самых безопасных платформ по данным Национального института стандартов и технологий.
- Мобильный BI и масштабируемость: Позволяет легко интегрировать системы управления базами данных с любыми устройствами и службами Azure для повышения производительности и аналитических возможностей работы с данными.

MS SQL

- Самой последней версией является Microsoft SQL Server 2019, доступный в 4 различных версиях, предназначенных для различных потребностей и нужд:
- **Enterprise:** Являясь наиболее полной версией, она предназначена для тех организаций, которым требуется работать с высоким уровнем обслуживания критически важных рабочих нагрузок.
- **Standard:** Разработано для того, чтобы предложить малым предприятиям инструмент, обеспечивающий базовое управление данными для выполнения их приложений, поддерживающий общие средства разработки для локальных или облачных сред.
- **Express:** Эта версия предназначена для независимых разработчиков программного обеспечения или новичков, которые хотят иметь бесплатную базу данных в качестве учебного метода для составления небольших серверных и настольных приложений для своего использования.
- **Developer:** Интегрируя все функциональные возможности Enterprise, она предназначена для всех тех разработчиков, которые хотят иметь систему тестирования и разработки для компиляции любого типа приложений на SQL Server, без доступа к производственной среде.