МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ**

**Кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Вариативная самостоятельная работа**

Анализ источников по теме «Технологии баз данных (Database engineering)»

Обучающейся 4 курса

Васильевой Марины Андреевны

Научный руководитель:

кандидат физико-математических наук,   
доцент кафедры ИТиЭО

Власов Дмитрий Викторович

Санкт-Петербург

2024

# Содержание

[Предисловие 3](#_Toc185722650)

[1. Развитие систем хранения данных 4](#_Toc185722651)

[2. Современные технологии баз данных 7](#_Toc185722652)

[2.2 Популярные СУБД 8](#_Toc185722653)

[2.3 Облачные технологии и базы данных 9](#_Toc185722654)

[3.1 Безопасность баз данных: проблемы и перспективы 11](#_Toc185722655)

[3.2 Проблема масштабируемости 12](#_Toc185722656)

[Заключение 14](#_Toc185722657)

[Список использованных источников 15](#_Toc185722658)

# Предисловие

Согласно Мамедли Р.Э., для хранения и управления данными в современных компьютерных системах применяются базы данных. Базы данных участвуют почти во всех областях нашей жизни: в учебных заведениях, в управлении бизнесом или производством, в банковском деле, в решении научных, инженерных и медицинских задач. Базы данных возникли из-за необходимости управлять большими объемами данных в организованном и эффективном виде. Эффективное управление данными требует применения современных технологий баз данных, которые обеспечивают их сбор, хранение, обработку и анализ.

Разработка и проектирование баз данных (Database Engineering) постоянно развиваются, охватывая как традиционные реляционные базы данных, так и современные нереляционные (NoSQL) и распределённые системы. Это связано с ростом объёмов данных, развитием технологий Big Data, машинного обучения и облачных вычислений.

Актуальность исследования обусловлена необходимостью анализа текущего состояния технологий баз данных, включая их возможности, ограничения и направления дальнейшего развития. Понимание этих аспектов важно для успешного применения баз данных в реальных проектах, особенно в условиях постоянного роста требований к масштабируемости, безопасности и производительности.

Настоящая работа основана на анализе научной литературы, статей и профессиональных материалов, что позволяет получить объективное представление о состоянии и тенденциях в данной области.

# Развитие систем хранения данных

История развития баз данных насчитывает более 50 лет [12].

**0-й этап: Файловые системы (1958 г.)**

* **Характеристика:**
  + Использование файлов как единицы хранения данных;
  + Отсутствие централизованного управления данными, что вело к дублированию и сложности синхронизации.
* **Проблемы:**
  + Избыточность и несогласованность данных;
  + Зависимость структур данных и прикладных программ;
  + Низкая производительность;
  + Трудности в доступе и обработке больших объёмов данных.
* **Пример:**
  + Магнитные ленты и перфокарты.

**1-й этап: Монолитная архитектура (1960—1980 гг.)**

* **Характеристика:**
  + Программные решения и данные тесно связаны;
  + Использование сетевых и иерархических моделей баз данных (например, IMS, CODASYL).
* **Проблемы:**
  + Ограниченная масштабируемость;
  + Сложность изменения структуры данных.
* **Пример:**
  + СУБД, основанные на иерархической структуре данных (IBM IMS).

**2-й этап: Архитектура "файл-сервер" (1975—1995 гг.)**

* **Характеристика:**
  + Централизованное хранение данных на сервере, доступ через локальную сеть;
  + Появление реляционных моделей данных благодаря работе Эдгара Кодда;
  + Стандартизация языка SQL.
* **Проблемы:**
  + Ограниченная производительность при увеличении числа пользователей;
  + Высокая нагрузка на сервер;
  + Значительный сетевой трафик и передача больших объемов данных.
* **Пример:**
  + Реляционные базы данных (Oracle, Sybase).

**3-й этап: Трёхуровневая архитектура (1985 г. — н.в.)**

* **Характеристика:**
  + Разделение на уровни: клиент, сервер приложений и база данных;
  + Увеличение безопасности и масштабируемости;
  + Появление объектно-ориентированных баз данных.
* **Технологические тренды:**
  + Расширение использования реляционных баз данных;
  + Интеграция с интернет-приложениями;
  + Стандарты SQL: SQL-89, SQL-92, SQL-99, SQL:2003, SQL:2006, SQL:2008.
* **Преимущества:**
  + Клиент формирует запросы на SQL, сервер выполняет основную обработку данных и передаёт клиенту только необходимые данные, что снижает сетевой трафик;
  + Высокий уровень безопасности благодаря ограничениям доступа, шифрованию и управлению ролями.
* Пример:
  + MySQL, PostgreSQL, Microsoft SQL Server.

**4-й этап: Облачные технологии (2012 г. — н.в.)**

* **Характеристика:**
  + Хранение данных в облачных средах, таких как AWS, Google Cloud, Azure;
  + Появление NoSQL баз данных для работы с большими объёмами данных (Big Data);
  + Виртуализация и распределённые системы хранения;
  + Database as a Service (DaaS) – клиент получает доступ к базе данных без необходимости её настройки и администрирования;
  + Обеспечение масштабируемости и отказоустойчивости через распределённые системы.
* **Особенности:**
  + Высокая доступность и масштабируемость;
  + Модели оплаты «по запросу».
* **Пример:**
  + MongoDB, DynamoDB, Google BigQuery.

# Современные технологии баз данных

2.1 Подходы к организации данных: реляционные и нереляционные модели

Согласно ГОСТу [7], база данных (database) – совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь.

Базы данных можно разделить на несколько типов, включая реляционные, нереляционные (NoSQL) и распределенные базы данных. Реляционные базы данных, такие как MySQL и PostgreSQL, используют таблицы для хранения данных и поддерживают язык SQL для управления данными. Нереляционные базы данных, такие как MongoDB и Cassandra, предлагают более гибкие структуры данных и лучше подходят для определенных типов приложений. Распределенные базы данных, такие как Google Bigtable и Amazon DynamoDB, обеспечивают высокую масштабируемость и отказоустойчивость.

Реляционные базы данных (RDBMS) являются наиболее распространенным типом баз данных[8]. Они используют таблицы для организации данных и поддерживают язык структурированных запросов (SQL) для управления этими данными. Основные концепции реляционных баз данных включают нормализацию, первичные и внешние ключи, индексы и транзакции. Нормализация помогает минимизировать избыточность данных и улучшить целостность данных, разбивая их на взаимосвязанные таблицы. Первичные ключи обеспечивают уникальность записей в таблице, а внешние ключи устанавливают связи между таблицами.

Нереляционные базы данных, также известные как NoSQL базы данных, предлагают более гибкие структуры данных по сравнению с реляционными базами данных[8]. Они могут быть документо-ориентированными, графовыми, столбцовыми или ключ-значение. Документо-ориентированные базы данных, такие как MongoDB, хранят данные в формате JSON или BSON, что позволяет легко работать с вложенными структурами данных. Графовые базы данных, такие как Neo4j, используют графы для представления данных и их связей, что делает их идеальными для социальных сетей и других приложений, где важны связи между данными. Столбцовые базы данных, такие как Cassandra, хранят данные в столбцах, что позволяет эффективно обрабатывать большие объемы данных. Базы данных ключ-значение, такие как Redis, хранят данные в виде пар ключ-значение, что делает их очень быстрыми для определенных типов операций.

Распределенные базы данных обеспечивают высокую масштабируемость и отказоустойчивость, распределяя данные по нескольким серверам или узлам. Это позволяет обрабатывать большие объемы данных и обеспечивать высокую доступность системы даже в случае отказа одного из узлов. Примеры распределенных баз данных включают Google Bigtable, Amazon DynamoDB и Apache Cassandra. Основные концепции распределенных баз данных включают репликацию, шардинг и согласованность данных. Согласованность данных обеспечивает целостность данных в распределенной системе, что может быть достигнуто с помощью различных моделей согласованности, таких как сильная согласованность, конечная согласованность и согласованность по времени.

Таблица 1 – Различия реляционных и нереляционных баз данных[6]

|  |  |  |
| --- | --- | --- |
| Категория | Реляционная БД | Нереляционная БД |
| Модель данных | Таблицы | Пара «ключ-значение», документ или график |
| Тип данных | Структурированные | Структурированные, частично структурированные или неструктурированные |
| Целостность данных | Высокий уровень при полном соответствии требованиям ACID | Конечная модель непротиворечивости |
| Производительность | Улучшается за счет добавления дополнительных ресурсов на сервер | Улучшается за счет добавления большего количества серверных узлов |
| Масштабирование | Горизонтальное масштабирование требует дополнительных стратегий управления данными | Горизонтальное масштабирование является простым |

# 2.2 Популярные СУБД

Согласно источнику CNews [2], PostgreSQL возглавила мировой рейтинг роста популярности СУБД и стала абсолютным лидером среди популярных СУБД в России. Но в абсолютном мировом рейтинге популярности PostgreSQL сохранила четвертое место, уступив Oracle, MySQL и Microsoft SQL Server. При этом две последние потеряли за 2023 г. 89 и 43 балла соответственно.

Рассмотрим преимущества некоторых из них [11].

**MySQL –** это реляционная СУБД с открытым исходным кодом, которая поддерживает табличные базы данных с простой структурой и сложными условиями запросов.

Преимущества: гибкость, высокая скорость обработки данных, простой интерфейс.

Применение: широко используется в сфере e-commerce, финтеха, IT для построения прогностических моделей.

**Microsoft SQL Server** – это разработка Microsoft, которая совместима с Windows и Linux.

Преимущества: простой интерфейс, надёжность хранения данных, интеграция с продуктами Microsoft, включая Excel.

Применение: используется для обработки данных из Microsoft Excel в задачах Data Mining.

**Oracle** – это объектно-реляционная СУБД, известная своей надёжностью и быстродействием.

Преимущества: простая установка и настройка, расширяемый функционал, высокая производительность.

Применение: решает задачи анализа продаж и построения моделей потребностей клиентов в зависимости от рыночных условий в e-commerce.

**PostgreSQL** – это объектно-реляционная СУБД с поддержкой работы на большинстве популярных платформ.

Преимущества: подходит для веб-сервисов, сайтов и облачных решений.

Применение: активно используется в облачных платформах для обработки больших объемов данных.

# 2.3 Облачные технологии и базы данных

Облачные технологии предоставляют новые подходы к управлению базами данных, перераспределяя функции и задачи между внутренними администраторами баз данных (АБД) и внешними провайдерами облачных услуг. Использование таких сервисов, как Amazon Web Services (AWS), Microsoft Azure и Google Cloud, позволяет организациям оптимизировать управление данными, улучшить их доступность и снизить затраты на инфраструктуру [10].

Преимущества облачных баз данных:

* Масштабируемость;
* Экономия ресурсов;
* Высокая доступность и отказоустойчивость;
* Простота администрирования;
* Интеграция с аналитическими инструментами.

Основные провайдеры облачных баз данных:

**Amazon Web Services (AWS)** предлагает широкий спектр сервисов для управления базами данных:

* Amazon RDS (Relational Database Service) – для реляционных баз данных, таких как MySQL, PostgreSQL, Oracle и Microsoft SQL Server.
* Amazon DynamoDB – высокопроизводительная NoSQL база данных для приложений с интенсивной нагрузкой.
* Amazon Redshift – решение для анализа больших данных с возможностью интеграции с BI-инструментами.

**Google Cloud Platform (GCP)** предоставляет решения, оптимизированные для интеграции с инструментами анализа данных и машинного обучения:

* Cloud SQL – полностью управляемый сервис для реляционных баз данных (MySQL, PostgreSQL, SQL Server).
* Firestore – документо-ориентированная база данных для быстрого масштабирования приложений.
* BigQuery – мощная аналитическая платформа для обработки больших объёмов данных.

**Microsoft Azure** предлагает гибкие решения для хранения данных с высокой интеграцией с продуктами Microsoft:

* Azure SQL Database – облачная версия Microsoft SQL Server.
* Azure Cosmos DB – масштабируемая мульти-модельная база данных, поддерживающая документо-ориентированную и графовую модели.
* Azure Synapse Analytics – инструмент для аналитики больших данных.

# 3.1 Безопасность баз данных: проблемы и перспективы

Согласно статье Полтавцевой и Хабарова, вопросы защиты данных являются критическими при обеспечении безопасности современных корпоративных систем. Атаки на хранилища и БД являются одними из самых опасных для предприятий и организаций. Согласно статистике компании infowateh [13], только за первое полугодие 2024 года утечек информации в России стало на 10% больше.

Злоумышленников интересуют такие виды информации, как внутренняя операционная информация, персональные данные сотрудников, финансовая информация, информация о заказчиках/клиентах, интеллектуальная собственность, исследования рынка/анализ деятельности конкурентов, платежная информация. Эти сведения в итоге хранятся в корпоративных хранилищах и БД различного объема.

Вопросы комплексной безопасности БД привлекают внимание исследователей, им ежегодно посвящается ряд работ как в России, так и за рубежом. Можно отметить такие исследования, как классическая работа [4]. В ней рассматриваются подходы к обеспечению конфиденциальности, целостности и доступности СУБД, предотвращение, определение и игнорирование атак. Предлагаются подходы к обеспечению мандатного и ролевого дискреционного доступа к реляционному серверу. Данную тему развивает работа [3], затрагивающая те же вопросы обеспечения разделения доступа, привилегий, аудита и шифрования данных, а также вопросы применения для обеспечения защищенного доступа встроенных механизмов, таких как триггеры, представления и хранимые процедуры. Резюмирующая их работа [1] обобщает развитие подходов к безопасности в историческом аспекте.

Современные проблемы обеспечения безопасности БД связаны с несколькими ключевыми аспектами:

* Недостаточная защита большинства СУБД. Например, СУБД MySQL не может похвастаться тем же уровнем безопасности, что и Oracle Database Server, несмотря на широкое использование в важных секторах, таких как электронная коммерция и государственные структуры.
* Невнимание к безопасности. Программисты, администраторы и прикладные разработчики не уделяют должного внимания вопросам безопасности, что способствует эксплуатации уязвимостей, таких как SQL-инъекции.
* Разнообразие моделей и языков. Разные СУБД используют различные языковые диалекты для работы с данными, что затрудняет создание унифицированных решений по безопасности.
* Сложности с нереляционными БД. NoSQL-системы, только недавно появившиеся на рынке, имеют значительные проблемы с безопасностью, включая отсутствие стандартных механизмов аутентификации и шифрования.

Для улучшения безопасности необходимо перейти от реактивного подхода к комплексной защите данных. Это включает:

* Разработка комплексных методик безопасности, которые можно применить при создании как реляционных, так и нереляционных СУБД.
* Оценка и классификация угроз и уязвимостей, чтобы систематизировать и эффективно защищать СУБД, прогнозируя возможные угрозы при внедрении новых механизмов.
* Стандартизация механизмов безопасности, которые применимы к различным СУБД с минимальными изменениями, что позволит внедрить универсальные решения.

Перспективы развития включают:

1. Стандартизация и унификация методов безопасности, что приведет к созданию мультиплатформенных средств защиты.
2. Разработка теоретической базы безопасности СУБД, которая позволит учитывать особенности различных моделей данных (в том числе для NoSQL) и обеспечивать их защиту от угроз, специфичных для каждой модели.
3. Исследования в области предсказания угроз, что обеспечит более проактивный подход в защите БД.

# 3.2 Проблема масштабируемости

Масштабируемость и производительность являются ключевыми проблемами для современных баз данных, особенно в условиях увеличивающихся объемов данных и возрастающего числа пользователей. Чем больше данных, тем сложнее обеспечить быструю обработку запросов, минимальные задержки и стабильную работу системы при росте нагрузки.

Горизонтальное масштабирование (например, добавление новых серверов) может привести к проблемам с синхронизацией данных между узлами.

Производительность часто ограничена возможностями процессора, памяти и дисков. С ростом данных могут возникать проблемы с эффективным поиском, индексацией и хранением.

Реляционные базы данных масштабируются с помощью репликации или шардирования. Эти способы относятся к горизонтальной масштабируемости баз данных. [5]

Репликация – это процесс копирования и размещения идентичной информации на разных серверах. Этот метод предполагает два типа серверов: master и slave. Master – основной сервер, в который записывается новая информация или изменяется имеющаяся, slave служит для копирования информации с мастера и её чтения.

При репликации создается большое количество копий данных и в случае поломки основного сервера, всегда имеется возможность заменить его одной из копий. Классический вариант репликации – это один мастер – сервер и несколько слэйвов. Однако, если информационная система имеет сложную архитектуру, возможен вариант, где мастеров создается несколько. Основным недостатком при таком способе является рассинхронизация и задержки передачи данных между серверами.

Шардирование или шардинг – второй способ масштабирования базы данных. Он подразумевает распределение информации между разными физическими серверами. Разные таблицы базы данных могут хранится на различных серверах.

Для масштабной информационной системы этот способ является оптимальным, особенно при использовании его вместе с репликацией. На практике организовать это довольно трудно, так как требуется наладить оптимальное межсерверное взаимодействие.

Выбор способа масштабирования зависит от конкретных требований с информационной системе и доступных ресурсов. Стоит так же учитывать, что масштабирование может повлечь за собой изменения в архитектуре приложения и базы данных и усложнить управление и обслуживание системы.

# Заключение

Тема «Технологии баз данных (Database engineering)» является ключевой для понимания современного состояния и развития технологий хранения и обработки данных, которые играют важную роль в различных областях человеческой деятельности. В ходе анализа источников и литературных данных было выявлено, что базы данных являются неотъемлемой частью всех современных информационных систем. Системы управления базами данных (СУБД) продолжают эволюционировать, предлагая новые подходы к организации данных, включая реляционные и нереляционные модели.

Особое внимание следует уделить вопросам безопасности баз данных, так как они становятся объектами атак и угроз.

Перспективы развития баз данных включают создание унифицированных решений, которые смогут эффективно обеспечивать защиту данных и поддерживать рост и развитие информационных систем в условиях новых технологических вызовов.

# Список использованных источников

1. Lesov P. Database security: a historical perspectiv. 2010. URL: <http://arxiv.org/ftp/arxiv/papers/1004/1004.4022.pdf> (дата обращения: 26.02.2016).
2. PostgreSQL возглавила мировой рейтинг роста популярности СУБД и стала абсолютным лидером среди популярных СУБД в России // CNews URL: <https://www.cnews.ru/news/line/2024-01-09_postgresql_vozglavila_mirovoj> (дата обращения: 21.12.2024).
3. Qiu M., Davis S. Database security mechanisms and implementation. IACIS, Issues in Inform. Syst. 2002, vol. 03, pp. 529-534.
4. Sandhu Ravi S., Jajodia Sushil. Data and database security and controls. Handbook of Information Security Management, Auerbach Publishers, 1993, pp. 181-199.
5. Бухараев Н.Р. Введение в реляционные базы данных и программирование на языке SQL /Н.Р.Бухараев. – Казань: Казан. ун-т, 2018. – 134 с.
6. В чем разница между реляционными и нереляционными базами данных? // Amazon Web Services URL: <https://aws.amazon.com/ru/compare/the-difference-between-relational-and-non-relational-databases/> (дата обращения: 21.12.2024).
7. ГОСТ 34.321-96. Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными: утвержден и введен в действие Постановлением Государственного комитета Российской Федерации по стандартизации и метрологии от 22.02.2001 N 88-ст. Дата введения: 2001-07-01. – URL: <https://docs.cntd.ru/document/1200017662> (дата обращения: 22.12.2024). – Текст: электронный.
8. Кузнецов С.Д. Основы современных баз данных. - 2-е изд. - М.: Интернет-Университет Информационных Технологий; Бином. Лаборатория Знаний, 2007. - 484 с.
9. М.А. Полтавцева, А.Р. Хабаров Безопасность баз данных: проблемы и перспективы // Программные продукты и системы. - 2016. - № ,том 29
10. Мамедли Р.Э. Системы управления базами данных: Учебное пособие. - Нижневартовск: Изд-во Нижневартовского государственного университета, 2021. - 214 с.
11. Обзор популярных СУБД // Центр развития компетенций в бизнес-информатике, логистике и управлении проектами Института открытых программ дополнительного образования Высшей школы бизнеса URL: <https://hsbi.hse.ru/articles/obzor-populyarnykh-subd/> (дата обращения: 21.12.2024).
12. Тема 1.1 Основы SQL: История возникновения // Logrocon URL: https://logrocon.ru/news/sql\_history (дата обращения: 21.12.2024).
13. Утечки информации в мире и России за первое полугодие 2024 года // InfoWatch URL: <https://www.infowatch.ru/sites/default/files/analytics/files/utechki-informatsii-v-mire-i-rossii-za-pervoye-polugodiye-dve-tysyachi-dvadtsat-chetvertogo-goda.pdf> (дата обращения: 21.12.2024).