Министерство образования и науки РФ

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина»  
Институт радиоэлектроники и информационных технологий – РТФ  
Департамент информационных технологий и автоматики

Оценка работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЛЯЦИОННЫХ СУБД**

Реферат по дисциплине «Методы и средства проектирования информационных систем и технологий»

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Зеленская Е.В.  
Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Тимошенкова Ю.С.  
Группа РИМ–161203

Екатеринбург

2017

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc481420553)

[1 ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ 4](#_Toc481420554)

[1.1 Модели баз данных 4](#_Toc481420555)

[1.2 Реляционная модель 5](#_Toc481420556)

[1.3 Реляционные СУБД 7](#_Toc481420557)

[2 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕЛЯЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ 10](#_Toc481420558)

[2.1 Oracle 11](#_Toc481420559)

[2.2 MySQL 14](#_Toc481420560)

[2.3 PostgreSQL 17](#_Toc481420561)

[2.4 SQLite 20](#_Toc481420562)

[ЗАКЛЮЧЕНИЕ 23](#_Toc481420563)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc481420564)

[ПРИЛОЖЕНИЕ А Таблицы характеристик СУБД 25](#_Toc481420565)

# ВВЕДЕНИЕ

Развитие информационных технологий неразрывно связано с увеличением количества хранимых данных. Поэтому очевидна необходимость развития специализированных инструментов. Инструментарий современных реляционных систем управления базами данных постоянно совершенствуется и развивается, появляются новые сервисы.

На сегодняшний день реляционная модель данных является основной при построении баз данных. Так происходит, потому что организация хранения информации в виде таблиц интуитивно понятна пользователям. Эта особенность привела к появлению множества реляционных систем управления базами данных. Их характеристики существенно различаются. Подбор наиболее подходящей системы управления базами данных стал нетривиальной задачей.

Цель работы – проанализировать и сравнить существующие реляционные системы управления базами данных. Достижение поставленной цели обеспечивается решением следующих задач:

1. изучение теоретических основ реляционных баз данных;
2. обзор сведений о реляционных системах управления базами данных;
3. сравнение и анализ существующих программных пакетов для работы с реляционными базами данных.

Системы баз данных сегодня являются основой построения большинства информационных систем и используются при автоматизации практически всех сфер человеческой деятельности.

# 1 ОСНОВНЫЕ ПОНЯТИЯ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

## 1.1 Модели баз данных

Каждая система поддерживает различные модели и структуры баз данных. Эта модель и определяет, как создаваемая СУБД будет оперировать данными. Существует довольно немного моделей БД, которые предоставляют способы четкого структурирования данных, самая популярная из таких моделей - реляционная модель.[1]

База данных (БД) – это поименованная совокупность структурированных данных, относящихся к определенной предметной области. Под предметной областью будем понимать некоторую часть реального мира, информация о которой представлена в базе данных. [2]

Система управления базами данных (СУБД) – это комплекс программных средств, предназначенных для создания и сопровождения базы данных. Можно выделить комплекс основных понятий, с помощью которых описываются принципы построения баз данных. [3]

Информационный объект – это описание некоторого реального объекта, явления, процесса, события в виде совокупности логически связанных атрибутов. Например, информационный объект СТУДЕНТ может быть представлен в базе данных с помощью следующих атрибутов: Код студента, Фамилия, Имя, Отчество, Год рождения. Это можно записать так: СТУДЕНТ (Код студента, Фамилия, Имя, Отчество, Год рождения) [1-3]

Ядром любой базы данных является модель данных. Модель данных – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними. Рассмотрим три основных типа моделей данных: иерархическую, сетевую и реляционную. [1-3]

Иерархическая модель данных представляет собой совокупность элементов данных, расположенных в порядке их подчинения и образующих по структуре перевернутое дерево. К основным понятиям иерархической модели данных относятся: уровень, узел и связь. Узел – это совокупность атрибутов данных, описывающих информационный объект.[2]

Сетевая модель данных основана на тех же основных понятиях (уровень, узел, связь), что и иерархическая модель, но в сетевой модели каждый узел может быть связан с любым другим узлом.[3]

Реляционная модель и реляционные БД могут быть очень мощным инструментом, но только если программист знает, как с ними обращаться. На сегодняшний день на рынке появились еще NoSQL системы с обещанием избавиться от старых проблем БД и добавить новый функционал. Исключая жесткую структуру данных, при этом сохранив реляционный стиль, эти СУБД предлагают более свободный способ работы с ними и гораздо большие возможности для их настройки. Хотя не обходится и без возникновения новых проблем. [1-3]

## 1.2 Реляционная модель

Впервые принципы реляционной модели были сформулированы в 1969—1970 годах Э. Ф. Коддом. Реляционная модель предложила математический способ структурирования, хранения и использования данных. По сути он расширил плоскую и сетевую модели, объединив их в реляционную. Основное преимущество которой было объединение данных в группы, именно реляционная модель позволила хранить данные в структурированном табличном виде (ФИО, адрес).[1-4]

В реляционной СУБД все обрабатываемые данные представляются в виде плоских таблиц. Информация об объектах определенного вида представляется в табличном виде: в столбцах таблицы сосредоточены различные атрибуты объектов, а строки предназначены для сведения описаний всех атрибутов к отдельным экземплярам объектов. [1-3]

Модель, созданная на этапе инфологического моделирования, в наибольшей степени удовлетворяет принципам реляционности. Однако для приведения этой модели к реляционной необходимо выполнить процедуру, называемую нормализацией.[5]

Теория нормализации оперирует с пятью нормальными формами. Эти формы предназначены для уменьшения избыточности информации, поэтому каждая последующая нормальная форма должна удовлетворять требованиям предыдущей и некоторым дополнительным условиям. При практическом проектировании баз данных четвертая и пятая формы, как правило, не используются. Мы ограничились рассмотрением первых четырех нормальных форм.[6]

Отношение - абстракция описываемого объекта как совокупность его свойств. Экземпляр отношения - совокупность значений свойств конкретного объекта. Первичный ключ - идентифицирующая совокупность атрибутов, т.е. значение этих атрибутов уникально в данном отношении. Не существует двух экземпляров отношения, содержащих одинаковые значения в первичном ключе. Простой атрибут - атрибут, значения которого неделимы. Сложный атрибут - атрибут, значением которого является совокупность значений нескольких различных свойств объекта или несколько значений одного свойства.[1]

Благодаря десятилетиям разработки, СУБД достигли довольно высокого уровня в производительности и отказоустойчивости. Опытом разработчиков и сетевых администраторов было доказано, что все эти инструменты отлично справляются со своими функциями в приложениях любой сложности, не теряют данных даже при некорректных завершениях работы. [1-3]

Несмотря на большие ограничения в формировании и управлении данными, реляционные базы данных сохраняют широкие возможности по настройке и предлагают довольно большой функционал. [1-3]

## 1.3 Реляционные СУБД

Реляционная СУБД (или РСУБД) - система управления реляционными БД. Такие модели управления можно охарактеризовать простотой, удобным табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Реляционные СУБД берут своё название от модели БД с которой работают. На данный момент и, наверное, в ближайшем будущем эти СУБД будут наиболее популярным выбором для хранения данных. [1-3]

В реляционных базах данные хранятся в виде таблиц, состоящих из строк и столбцов. Каждая таблица имеет собственный, заранее определенный набор именованных полей. Столбцы таблиц реляционной базы могут содержать скалярные данные фиксированного типа, например, числа, строки или даты. Таблицы в реляционной базе данных могут быть связаны отношениями «один-к-одному» или «один-ко-многим». Количество строк записей в таблице неограниченно, и каждая запись соответствует отдельной сущности.[4]

Реляционные СУБД используют строго описанные структуры данных - схемы. Схема базы данных включает в себя описание содержания, структуры и ограничений целостности, т.е. она определяет таблицы, поля в каждой таблице, а также отношения между полями и таблицами.[5]

Реляционные базы данных занимают сейчас доминирующее положение (MS SQL Server, MS Access, InterBase, FoxPro, PostgreSQL, Paradox и другие).

Требования к реляционным СУБД (по Кодду):

1. Явное представление данных (The Information Rule). Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть атомарны. Порядок строк в реляционной таблице не должен влиять на смысл данных.

2. Гарантированный доступ к данным (Guaranteed Access Rule). К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.

3. Полная обработка неизвестных значений (Systematic Treatment of Null Values). Неизвестные значения (NULL), отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций.

4. Доступ к словарю данных в терминах реляционной модели (Dynamic On-Line Catalog Based on the Relational Model). Словарь данных должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств.

5. Полнота подмножества языка (Comprehensive Data Sublanguage Rule). Система управления реляционными базами данных должна поддерживать единственный язык запросов, который позволяет выполнять все операции работы к данным: операции определения данных, операции манипулирования данными, управление доступом к данным, управление транзакциями.

6. Поддержка обновляемых представлений (View Updating Rule). Обновляемое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, модификации и удаления данных.

7. Наличие высокоуровневых операций управления данными (High-Level Insert, Update, and Delete). Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но по отношению к любому множеству строк.

8. Физическая независимость данных (Physical Data Independence). Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

9. Логическая независимость данных (Logical Data Independence). Представление данных в приложении не должно зависеть от структуры реляционных таблиц.

10. Независимость контроля целостности (Integrity Independence). Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. СУБД должна выполнять проверку заданных ограничений целостности и автоматически поддерживать целостность данных.

11. Независимость от распределенности (Distribution Independence). База данных может быть распределенной, может находиться на нескольких компьютерах, и это не должно оказывать влияние на приложения.

12. Согласование языковых уровней (Non-Subversion Rule). Не должно быть иного средства доступа к данным, отличного от стандартного языка работы с данными. Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и целостности, которые поддерживаются языком более высокого уровня. [1-3]

# 2 АНАЛИЗ СУЩЕСТВУЮЩИХ РЕЛЯЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Издание DB-Engines опубликовало новую редакцию своего рейтинга популярности СУБД. По методике расчёта рейтинг (рис. 1) СУБД напоминает рейтинг языков программирования TIOBE и учитывает популярность запросов в поисковых системах, число результатов в поисковой выдаче, объём обсуждений на популярных дискуссионных площадках и социальных сетях, число вакансий в агентствах по найму персонала и упоминаний в профилях пользователей. Позицию лидеров в рейтинге сохранили СУБД Oracle, MySQL и Microsoft SQL Server. Сравнительные таблицы большинства существующих реляционных СУБД представлены в приложении А.

Рисунок 1 – Рейтинг реляционных СУБД, используемых в мире  
на апрель 2017 года

## 2.1 Oracle

Oracle – Объектно-реляционная система управления базами данных. Одна из ранних версий стала первой коммерческой СУБД, поддерживающей язык запросов SQL. В настоящее время доступно шесть версий СУБД:

Enterprise Edition Наиболее полная версия, не имеет значительных ограничений. Данная корпоративная редакция продукта предназначается для крупных предприятий. Пользователям предоставляются опции, с помощью которых есть возможность архитектурно и функционально улучшить возможности сервера.

Standard Edition Ограничение по количеству процессорных разъемов (не более 4-х). Редакция предназначена для использования в предприятиях среднего размера или в подразделении крупной компании.

Standard Edition One Ограничение по количеству процессорных разъемов (не более 2-х). Кроме того, не поддерживается кластеризация. - Personal Edition Версия СУБД рассчитана на использование одним пользователем.

Lite Данная редакция предназначена для использования в мобильных и встраиваемых устройствах. Также возможно использование на предприятиях малого размера.

Express Edition (XE) Бесплатная версия СУБД. Используется только один процессор, ограничения по количеству оперативной памяти (1 Гб) и объему базы данных (11 Гб) [8].

Все шесть редакций СУБД используют один исходный код и имеют аналогичный функционал, исключая некоторые специализированные инструменты конкретных версий. Основная задача стандартной, персональной и мобильной версий – уменьшение общей стоимости владения, простота установки и использования ПО.

Oracle Database является кроссплатформенным ПО. Это достигается за счет того, что около 80 % программного кода написано на платформо-независимом языке Си. А ядро сервера, составляющее остальные 20 % кода, переписывается под конкретную платформу, так как реализовано на машинно-зависимых языках [9].

### 2.1.1 Преимущества Oracle

Главным конкурентным преимуществом СУБД Oracle была высокая скорость обработки огромных массивов информации, которую тогда отметили все эксперты

Механизм пакетной обработки запросов с подтверждением. Транзакционный механизм позволяет не опасаться сбоев программной среды и оборудования. В случае успешного завершения обработки пакета SQL-команд сервер баз данных генерирует так называемый commit — подтверждение успешного выполнения, которое отсылается клиенту. В случае же аппаратного или программного сбоя происходит rollback — возвращение к предыдущей контрольной точке, благодаря чему целостность данных в любой ситуации не будет нарушена.

Oracle Database является признанным лидером в этой отрасли производства ПО, свидетельствует тот факт, что по итогам 2007 года Oracle Database принадлежит 47% мирового рынка систем управления базами данных.

Высочайшая надежность, поддержка многих компьютерных платформ, поддержка всевозможных вариантов архитектур (многопроцессорных кластеров, систем с массовым параллелизмом и т.д.), идентичность кода различных версий сервера БД. для всех платформ, гарантирующая предсказуемость работы Oracle на разных платформах, возможности разбиения крупных БД. на разделы, поддержка XML в хранимых процедурах, поддержка OLAP (Online Analytical Processing) и т.д.

### 2.1.2 Недостатки Oracle

Oracle рекомендуется использовать для очень больших централизованных или распределенных баз данных (хранилищ данных) коллектив­ного использования для крупных предприятий.

Общие рекомендации. Для организации хранилища данных больших объемов рекомендуется использовать SQL Server или Oracle, а в клиентском приложении доступ к данным этого хранилища осуществить средствами СУБД Access (через проект), FoxPro (через удаленные представления) или средствами языков программирования Visual Basic, Visual C++, Delphi, C++ Builder (через ADO), или инструментальными средствами обработки данных Oracle.

Сложность администрирования и необходимость покупать лицензию это главные недостатки данной СУБД.

### 2.1.3 Когда следует использовать Oracle

Когда необходимо систематизировать сложные данные.

Такая СУБД используется на больших предприятиях, во-первых закупка лицензии для частного использования – невыгодно, а во вторых мобильные разработки зачастую не используют такой объем данных.

Работа с важными данными и необходимость их полноценной защиты, целостности и возможности полноценной обработки.

Сложные пользовательские процедуры - если вам необходимо использовать пользовательские процедуры, то Oracle имеет встроенную поддержку для них.

Интеграция - если в будущем вы планируете переход на бесплатные СУБД, например, на PostgreSQL будет довольно просто по сравнению с другими платными СУБД.

### 2.1.4 Когда лучше отказаться от Oracle

Когда нет средств на поддержку БД и оплату получения лицензии для работы с ней.

Когда работа происходит с данными в виде документов или другие сложные данные.[9]

## 2.2 MySQL

MySQL - это самая распространенная полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. Обучиться использованию этой СУБД довольно просто, так как на просторах интернета вы легко найдете большее количество информации.

Благодаря популярности этой СУБД, существует огромное количество различных плагинов и расширений, облегчающих работу с системой.

Несмотря на то, что в ней не реализован весь SQL функционал, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличии от SQLite работают со службами MySQL.

### 2.2.1 Преимущества MySQL

Простота в работе - установить MySQL довольно просто. Дополнительные приложения, например, GUI, позволяет довольно легко работать с БД.

Богатый функционал - MySQL поддерживает большинство функционала SQL.

Безопасность - большое количество функций, обеспечивающих безопасность, которые поддерживается по умолчанию.

Масштабируемость - MySQL легко работает с большими объемами данных и легко масштабируется.

Скорость - упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

### 2.2.2 Недостатки MySQL

Известные ограничения - по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях.

Проблемы с надежностью - из-за некоторых способов обработки данных MySQL (связи, транзакции, аудиты) иногда уступает другим СУБД по надежности.

Медленная разработка - Хотя MySQL технически открытое ПО, существуют жалобы на процесс разработки. Стоит заметить, что существуют другие довольно успешные СУБД, созданные на базе MySQL, например, MariaDB.

### 2.2.3 Когда следует использовать MySQL

Распределённые операции - если функционала SQLite не хватает, то стоит рассмотреть MySQL. Так как эта СУБД сочетает в себе продвинутый функционал и свободный доступ к исходному коду.

Высокий уровень безопасности - система безопасности MySQL включает в себе простые и в то же время достойные способы защиты доступа к данным.

Веб сайты и веб приложения - большинство сайтов и онлайн приложений спокойно работают с MySQL несмотря на некоторые ограничения. Будучи легкой в настройке и масштабируемой системой - MySQL проверена временем.

Индивидуальные решения - если вы работаете с каким-либо специфическим проектом, MySQL легко сможет вам помочь благодаря широким возможностям в настройке и функционалом.

### 2.2.4 Когда лучше отказаться от MySQL

Соответствие стандартам -Так как MySQL не ставит для себя целью - полностью соответствовать стандартам SQL, то эта СУБД не полностью поддерживает SQL. Если в будущем вы планируете перейти на подобную систему, то MySQL - не лучший выбор.

Многопоточность - хотя некоторые движки БД довольно легко выполняют параллельное чтение, параллельные операции чтения-записи могут создать проблемы.

Недостаток функционала - некоторые движки MySQL, например, не поддерживают полнотекстовый поиск.[10]

## 2.3 PostgreSQL

PostgreSQL является самым профессиональным из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. PostgreSQL или Postgres стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий.

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID).) Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций.

Хотя PostgreSQL и не может похвастаться большой популярностью в отличии от MySQL, существует довольно большое число приложений, облегчающих работу с PostgreSQL, несмотря на всю мощность функционала. Сейчас довольно легко установить эту СУБД используя стандартные менеджеры пакетов операционных систем.

### 2.3.1 Преимущества PostgreSQL

Открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой.

Большое сообщество - существует довольно большое сообщество, в котором вы запросто найдёте ответы на свои вопросы.

Большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими.

Расширения - существует возможность расширения функционала за счет сохранения своих процедур.

Объектность - PostrgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

### 2.3.2 Недостатки PostgreSQL

Производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL.

Популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество.

Хостинг - в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

### 2.3.3 Когда следует использовать PostgreSQL

Целостность данных - когда надежность и целостность данных - ваши требования, PostgreSQL будет, пожалуй, лучшим выбором.

Сложные пользовательские процедуры - если вам необходимо использовать пользовательские процедуры, то PostgreSQL имеет встроенную поддержку для них.

Интеграция - если в будущем вы планируете переход на платные СУБД, например, Oracle, то сделать это с PostgreSQL будет довольно просто по сравнению с другими бесплатными СУБД.

Сложная структура данных - по сравнению с другими открытыми СУБД PostgreSQL предоставляет больше возможностей для создания сложных структур данных без необходимости жертвовать какими-либо аспектами.

### 2.3.4 Когда лучше отказаться от PostgreSQL

Скорость - если быстрое чтение для вас единственный фактор, то стоит присмотреться к другим СУБД.

Простая настройка - если вам не нужна целостность данных, соответствие ACID или сложные структуры данных, то настройка PostgreSQL может быть сложной.

Репликация - если вы не готовы потратить время и энергию на то, что мог бы с легкостью сделать MySQL, то, наверное, проще было бы на нем и остаться.[7]

## 2.4 SQLite

Легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в этих файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

### 2.4.1 Преимущества SQLite

Файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины.

Используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены (RIGHT OUTER JOIN или FOR EACH STATEMENT), но основные все-таки поддерживаются.

Отличная при разработке и тестировании - в процессе разработки приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки написанной на языке C.

### 2.4.2 Недостатки SQLite

Отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях.

Отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

### 2.4.3 Когда следует использовать SQLite

Встроенные приложения - если вам важна возможность легкого переноса приложения и не важна масштабируемость. Например, однопользовательские приложения, мобильные приложения или игры.

Прямой доступ к диску - при необходимости напрямую обращаться к диску вы можете выиграть при переходе на эту СУБД в функционале и простоте использования SQL языка.

Тестирование - использование дополнительных процессов при тестировании функционала, очень замедляет приложение.

### 2.4.4 Когда лучше отказаться от SQLite

Многопользовательские приложения - если вам необходимо обеспечить доступ к данным для нескольких пользователей, да и к тому же различать их по правам доступа, то, наверное, полноценная СУБД (например, MySQL) будет более логичным выбором.

Запись больших объемов данных - одно из ограничений SQLite это операции записи. Разрешен только один процесс записи в промежуток времени, что сильно ограничивает производительность.[6]

# ЗАКЛЮЧЕНИЕ

Входе выполнения реферата были рассмотрены принципы устройства реляционных баз данных и основное программное обеспечение для управления ими. При выполнении работы был проведен анализ существующих программных решения для управления реляционными базами данных.

В работе подробно описаны самые популярные программные решения для управления реляционными БД. Кратко описаны основные характеристики основных реляционных СУБД, итоговые сравнительные таблицы представлены в приложении к работе.

При выборе СУБД для своего проекта или предприятия стоит учитывать характеристики проекта, возможность покупки лицензии и поддержки той или иной СУБД.

В ходе выполнения анализа можно отметить, что на сегодняшний день набирает популярность СУБД PostgreSQL. У PostgreSQL множество возможностей. Созданный с использованием объектно-реляционной модели, он поддерживает сложные структуры и широкий спектр встроенных и определяемых пользователем типов данных. Он обеспечивает расширенную ёмкость данных и заслужил доверие бережным отношением к целостности данных. Возможно, вам не понадобятся все те продвинутые функции хранения данных, которые мы исследовали в этой статье, но, поскольку потребности могут быстро возрасти, есть несомненное преимущество в том, чтобы иметь всё это под рукой.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Автоматизированные информационные технологии в экономике /под ред. проф. Г.А. Титоренко. – М.: ЮНИТИ, 2005. – 399с.
2. Информатика для юристов и экономистов /под ред. С.В. Симоновича. – СПб.: Питер, 2005. – 688с.
3. Информатика. Базовый курс. /Симонович С.В. и др. — Спб.: Питер, 2006. — 640 с.
4. Дейт К. Дж. Введение в системы баз данных - 8-е изд. — М.: «Вильямс», 2006.
5. Дрога А. А., Жукова П. Н., Копонев Д. Н., Лукьянов Д. Б., Прокопенко А. Н. Информатика и математика. – Белгород.: Белгородский юридический институт МВД РФ, 2008.
6. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика 3-е изд. — М.: «Вильямс», 2003.
7. Кузнецов С. Д. Основы баз данных. — 1-е изд. — М.: «Интернет-университет информационных технологий - ИНТУИТ.ру», 2005.
8. Скотт В. Эмблер, Прамодкумар Дж. Садаладж. Рефакторинг баз данных: эволюционное проектирование — М.: «Вильямс», 2007.
9. Леонтьев В.П. Новейшая энциклопедия персонального компьютера 2005. – М.:ОЛМА-ПРЕСС Образование, 2005. – 800с.
10. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных/ под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА, 2000. – 416с.
11. Экономическая информатика и вычислительная техника./ Под ред. В.П. Косарева. М.: Финансы и статистика, 2005. –592с.

# ПРИЛОЖЕНИЕ А Таблицы характеристик СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица А.1 – Сравнительные характеристики применимости СУБД | | | |
| СУБД | Параметры | | |
| Масштаб применения | Модель данных | Требуемый уровень квалификации персонала |
| Oracle Database | От персонального использования до предприятия крупных размеров | Клиент-серверная | Высокий и средний |
| Microsoft Access | Предприятия малых и средних размеров | Файл-серверная | Низкий |
| MS SQL Server | Предприятия малых, средних и крупных размеров | Клиент-серверная | Средний и низкий |
| Sybase Adaptive  Server Enterprise | Предприятия средних и крупных размеров | Клиент-серверная | Средний и низкий |
| ЛИНТЕР | Предприятия средних и крупных размеров | Клиент-серверная | Средний |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы А.1 | | | |
| СУБД | Параметры | | |
| Масштаб применения | Модель данных | Требуемый уровень квалификации персонала |
| MySQL | Предприятия малых и средних размеров | Клиент-серверная | Средний и низкий |
| IBM Database 2 | Предприятия средних и крупных размеров | Клиент-серверная | Средний |
| Firebird | Крупное предприятие | Клиент-Серверная | Высокий и средний |
| PostgreSQL | Предприятия малых и средних размеров | Клиент-Серверная | Средний |
| Interbase | Предприятия малых, средних и крупных размеров | Клиент-Серверная | Средний |
| SQLite | Предприятия малых и средних размеров | Встраиваемая | Средний |

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица А.2 – Сравнительные характеристики СУБД | | | |
| СУБД | Параметры | | |
| Разработчик | Доступность | Диалект языка SQL |
| Oracle Database | Oracle | Платно, но есть бесплатная облегченная версия | PL/SQL |
| Microsoft Access | Microsoft | Платно, но есть бесплатная облегченная версия | Jet SQL |
| MS SQL Server | Sybase AshtonTate Microsoft | Платно, но есть бесплатная облегченная версия | Transact-SQL |
| Sybase Adaptive  Server Enterprise | SAP SE Microsoft SQL Server Sybase | Проприетарная лицензия, т.е. несвободное ПО или полусвободное | Transact-SQL |
| ЛИНТЕР | РЕЛЭКС | Только платно | Стандарт SQL:2003 |
| MySQL | MySQL AB Sun Microsystems Oracle | Бесплатно, есть коммерческая версия | Частично соответствует стандарту SQL:2003 |

|  |  |  |  |
| --- | --- | --- | --- |
| Продолжение таблицы А.2 | | | |
| СУБД | Параметры | | |
| Разработчик | Доступность | Диалект языка SQL |
| IBM Database 2 | IBM | Платно, но есть бесплатная облегченная версия | Язык SQL DB2  PL/SQL |
| Firebird | Сообщество | Бесплатное распространение | Процедурный язык SQL (PSQL) |
| PostgreSQL | Сообщество | Бесплатно, есть коммерческая версия | Стандарт SQL:2011 |
| Interbase | Embarcadero  Technologies | Платно, есть пробная версия с временным ограничением | Стандарт ANSI-92 и частично стандарт ANSI-III |
| SQLite | Hwaci  сообщество | Бесплатное распространение | Частично соответствует стандарту SQL-92 |