

РЕФЕРАТ

Научно-исследовательская работа 24 с., 5 рис., 4 табл., 10 ист., 1 прил.

БАЗЫ ДАННЫХ, СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, ФАЙЛ-СЕРВЕРНЫЕ, КЛИЕНТ-СЕРВЕРНЫЕ, ВСТРАИВАЕМЫЕ, ИЕРАРХИЧЕСКИЕ СУБД, СЕТЕВЫЕ СУБД, РЕЛЯЦИОННАЯ СУБД

Объект исследования — система управления базами данных.

Целью работы является проведение анализа существующих систем управления базами данных по модели данных и по способу доступа к базе данных. Поставленная цель достигается путем рассмотрения и классификации существующих систем управления базами данных.

Результат данной работы показал, что каждая система управления базами данных имеет ряд преимуществ и недостатков и применим в зависимости от требований и ограничений к задаче.

СОДЕРЖАНИЕ

РЕФЕРАТ	2
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	5
ВВЕДЕНИЕ	6
1 Анализ предметной области	7
1.1 Базовые понятия и термины	7
1.2 Способы классификации СУБД	8
2 Существующие системы управления базами данных по модели данных	9
2.1 Обзор СУБД по модели данных	9
2.1.1 Иерархическая СУБД	9
2.1.2 Сетевая СУБД	10
2.1.3 Реляционные БД	12
2.2 Критерии сравнения систем управления базами данных	13
2.3 Сравнение систем управления базами данных	13
3 Существующие системы управления базами данных по способу доступа к базе данных	16
3.1 Обзор СУБД по способу доступа к базе данных	16
3.1.1 Файл-серверные	16
3.1.2 Клиент-серверные	17
3.1.3 Встраиваемые	18
3.2 Критерии сравнения архитектур	20
3.3 Сравнение архитектур СУБД	21
ЗАКЛЮЧЕНИЕ	22

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23
ПРИЛОЖЕНИЕ А	24

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В текущей расчетно-пояснительной записке применяются следующие сокращения и обозначения.

БД — База данных.

СУБД — Система управления базами данных.

ЭВМ — Электронно-вычислительная машина.

ВВЕДЕНИЕ

В современном информационном мире, где объемы данных постоянно растут, системы управления базами данных играют ключевую роль в обеспечении эффективного хранения, организации и управления информацией. Эти мощные программные инструменты стали неотъемлемой частью информационной инфраструктуры предприятий, образовательных учреждений, государственных учреждений и многих других организаций.

Целью работы является проведение анализа существующих систем управления базами данных по модели данных и по способу доступа к базе данных.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области системы управления базами данных;
- провести обзор существующих систем управления базами данных по модели данных и по способу доступа к базе данных;
- сформулировать критерии сравнения этих систем;
- классифицировать существующие системы управления базами данных по модели данных и по способу доступа к базе данных.

1 Анализ предметной области

Прежде чем приступить к теме исследования, следует описать предмет исследования и связанные с ним термины. В данной секции описаны понятия, которые будут использоваться в работе.

1.1 Базовые понятия и термины

Данные — это информация, зафиксированная в некоторой форме, пригодной для последующей обработки, передачи и хранения, например, находящаяся в памяти ЭВМ или подготовленная для ввода в ЭВМ [1].

База данных — структурированное поименованное хранилище информации [1].

СУБД — специализированное программное обеспечение, обеспечивающее доступ к базе данных как к совокупности её структурных единиц. СУБД необходима для создания и поддержки базы данных информационной системы в той же степени, как для разработки программы на алгоритмическом языке транслятор [1].

Программные составляющие СУБД включают в себя ядро и сервисные средства (утилиты) [1].

- 1) Ядро СУБД — это набор программных модулей, необходимый и достаточный для создания и поддержания БД, то есть универсальная часть, решающая стандартные задачи по информационному обслуживанию пользователей.
- 2) Сервисные программы предоставляют пользователям ряд дополнительных возможностей и услуг, зависящих от описываемой предметной области и потребностей конкретного пользователя.

Модель данных — это совокупность правил порождения структур данных в базе данных, операций над ними, а также ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения.

1.2 Способы классификации СУБД

Классификация СУБД является важным этапом в области баз данных, поскольку позволяет систематизировать и структурировать разнообразие существующих решений.

Эта задача включает определение различных критериев, которые позволяют группировать СУБД по их основным характеристикам.

- 1) По модели данных: означает разделение СУБД на группы в соответствии с тем, как они моделируют и организуют данные внутри базы данных.
- 2) По способу доступ к БД: предполагает разделение их в зависимости от того, каким образом приложения и пользователи могут взаимодействовать с данными в базе.

2 Существующие системы управления базами данных по модели данных

2.1 Обзор СУБД по модели данных

В данном разделе рассмотрены существующие СУБД по модели данных, которые используются в настоящее время. Все они различаются по своим особенностям и применяемым технологиям.

2.1.1 Иерархическая СУБД

В иерархической модели данных используется представление базы данных в виде древовидной структуры, состоящей из объектов различных уровней [2].

В моделях данных, где устанавливаются отношения предок-потомок, каждый объект может содержать несколько объектов более низкого уровня. Эти отношения образуют иерархическую структуру данных, где один объект является предком для другого объекта, который является его потомком. При этом, в иерархической модели данных каждый объект-потомок может иметь только одного предка. Это означает, что объект-потомок находится в строгом отношении с одним объектом-предком. Однако, объект предок может иметь несколько объектов-потомков, что позволяет создавать ветвистую иерархию.

Иерархические СУБД применяются в ситуациях, когда данные естественным образом организованы в иерархическую структуру.

К преимуществам иерархической СУБД относят:

- быстрый и эффективный поиск данных;
- легко добавлять/удалять информацию;
- эффективное хранение данных;

- производительность.

Недостатками таких моделей являются:

- ограниченная гибкость;
- сложно поддерживать и обновлять;
- ограниченная поддержка манипулирования данными;
- ограниченная совместимость;
- отсутствие стандартизации.

Файловая система представлена на рисунке 1.



Рисунок 1 – Файловая система

2.1.2 Сетевая СУБД

Сетевая модель данных является расширением иерархической модели, предоставляя более гибкие возможности для организации данных и установления связей между записями [3].

В сетевой модели каждый потомок может иметь несколько предков, в отличие от иерархической модели, где каждый потомок имеет только одного предка. Это позволяет создавать сложные и перекрестные связи между записями, что способствует более гибкому представлению и организации данных.

Сетевая база данных состоит из двух основных компонентов: экземпляров записей и экземпляров связей. Каждый экземпляр связи определяет отношение между экземпляром предка и одним или несколькими экземплярами потомка. Экземпляр связи содержит ссылку на экземпляр предка и упорядоченный набор ссылок на экземпляры потомка.

Сетевая архитектура представлена на рисунке 2.

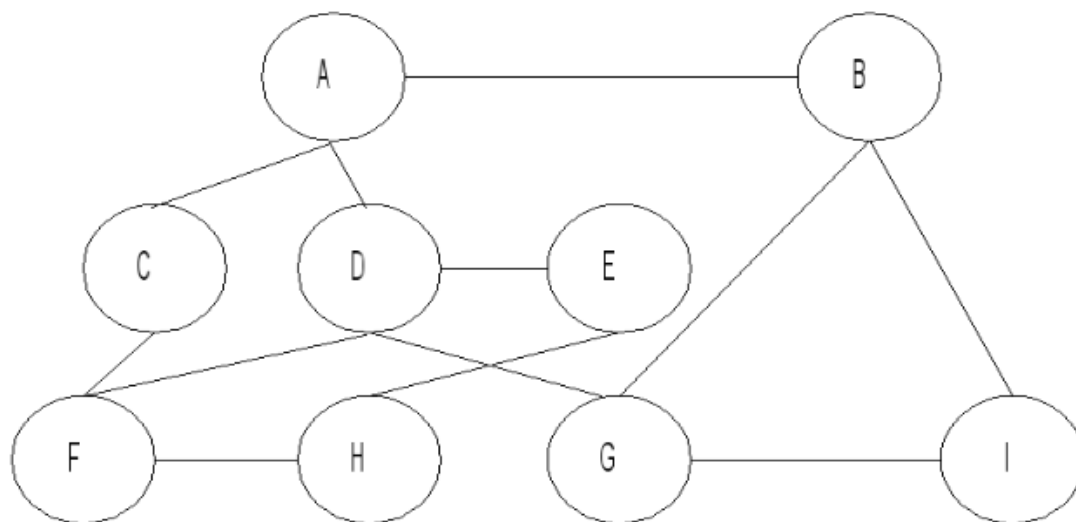


Рисунок 2 – Сетевая модель данных

К преимуществам сетевая СУБД относят [4]:

- обработка больших объемов информации (возможность построения на основе таких СУБД «хранилищ данных»);
- поддержка аналитической обработки данных;
- эффективная реализация обработки данных по показателям затрат памяти и оперативности.

Недостатками таких моделей являются [4]:

- высокая сложность и жесткость схемы БД, построенной на ее основе;
- сложность для понимания и выполнения обработки информации в БД обычным пользователем;
- ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями;

2.1.3 Реляционные БД

Реляционная база данных — это набор данных с predetermined отношениями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк. Таблицы хранят информацию об объектах, представленных в базе данных. Таблицы состоят из строк — записей и столбцов — полей. Каждый столбец таблицы хранит определенный тип данных, каждая ячейка содержит значение атрибута. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешнего ключа. Доступ к этим данным можно получить многими способами без реорганизации таблиц базы данных [5].

Реляционная модель является удобной и наиболее широко используемой формой представления данных. Широко распространенной реляционной СУБД является программа фирмы Microsoft Access [3]. Специальные программы класса СУБД (Oracle, MS SQL, Adabas, Informix, MySQL, SQLite и др.) разрабатываются таким образом, чтобы предоставлять пользователям широкие возможности их развития.

Преимуществами такой модели являются [6]:

- структурированность и организация данных;
- целостность данных;
- гибкий язык запросов;
- поддержка транзакций и согласованности;
- удобство администрирования;
- безопасность данных.

Недостатками реляционной модели являются [3]:

- невысокую эффективность;
- возможность обеспечения быстрого поиска с помощью индексов.

2.2 Критерии сравнения систем управления базами данных

В данном разделе описаны критерии, которые используются для выбора СУБД при создании информационных систем.

Таблица 1 – Критерии выбора СУБД [7]

Критерий	Описание
Моделирование данных	Используемая модель данных
Особенности архитектуры и функциональные возможности	Мобильность, Масштабируемость, Распределенность, Сетевые возможности
Контроль работы системы	Контроль использования памяти компьютера
Производительность	Оценивает скорость выполнения запросов и обработки данных
Надежность	Связана с устойчивостью системы к сбоям, ее способностью обеспечивать доступность данных
Требования к рабочей среде	определяют условия, под которыми СУБД должна функционировать

2.3 Сравнение систем управления базами данных

В данном разделе описаны все описанные СУБД, а также дано сравнение этих СУБД по всем критериям, указанным в таблице 1.

Таблица 2 – Сравнение СУБД

Критерий	СУБД		
	Иерархическая	Сетевая	Реляционная
Моделирование данных	Иерархическая структура, древовидная организация	Сетевая структура, разрешает несколько родителей	Табличная структура, данные представлены в таблицах
Особенности архитектуры и функциональные возможности	Простая, но ограниченная в функциональности	Более гибкая архитектура с поддержкой нескольких родительских связей	Сложные запросы через язык SQL, интерфейс
Контроль работы системы	Ограниченный контроль, сложно обеспечить целостность	Более гибкий контроль, но сложность управления	Эффективный контроль через язык SQL и ограничения
Производительность	Зависит от структуры и запросов, может быть эффективной для определенных задач	Улучшенная для некоторых сценариев	Эффективные операции с использованием индексов, оптимизированные для чтения данных
Надежность	Ограниченная надежность из-за ограничений структуры	Улучшенная надежность с гибкостью в построении связей	Высокая надежность через использование транзакций и ограничений целостности
Требования к рабочей среде	Зависит от конкретной реализации и объема данных	Зависит от конкретной реализации и структуры	Обширные требования к хранению и обработке данных

Из проведенного сравнения видно, что каждый тип СУБД имеет свои особенности и подходит для определенных сценариев использования.

- Моделирование данных:
 - Иерархическая и сетевая СУБД подходят для данных с явной структурой, где важны отношения и иерархии;
 - Реляционная СУБД эффективна для хранения данных в табличной форме, обеспечивая стандартизацию;
- Контроль работы системы:
 - Реляционная СУБД выделяется эффективным контролем через SQL и поддержкой транзакций;
- Особенности архитектуры и функциональные возможности:
 - Сетевая СУБД предоставляет более гибкую архитектуру, поддерживая несколько родительских связей;
 - Реляционная СУБД отличается стандартизированным интерфейсом и поддержкой сложных SQL-запросов;
- Производительность и надежность:
 - Иерархическая и сетевая СУБД подходят для специфических сценариев, их производительность зависит от структуры и запросов;
 - Реляционная СУБД выделяется высокой производительностью и надежностью за счет эффективных операций и поддержки ACID;
- Требования к рабочей среде:
 - Реляционная СУБД требует обширных ресурсов для хранения и обработки данных;
 - Иерархическая и сетевая СУБД управляются более удобно, но с ограничениями структуры.

Итак, выбор типа СУБД зависит от конкретных требований проекта, структуры данных и предполагаемых сценариев использования.

3 Существующие системы управления базами данных по способу доступа к базе данных

3.1 Обзор СУБД по способу доступа к базе данных

3.1.1 Файл-серверные

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере [2]. Возможна многопользовательская работа с одной и той же БД, когда каждый пользователь со своего компьютера запускает приложение, расположенное на сетевом сервере. Тогда на компьютере пользователя запускается копия приложения. В архитектуре файл-сервер вся тяжесть выполнения запросов к базе данных и управления целостностью базы данных ложится на приложение пользователя. По каждому запросу к БД из приложения данные из таблиц БД перегоняются на компьютер пользователя, независимо от того, сколько реально нужно данных для выполнения запроса. После этого выполняется запрос. Каждый пользователь имеет на своем компьютере локальную копию данных. На каждом клиентском компьютере (рабочей станции) располагается СУБД, который время от времени обновляет из реальной БД, расположенной на сетевом сервере. На данный момент файл-серверная технология считается устаревшей.

Такая архитектура имеет следующие преимущества [8]:

- низкая стоимость разработки, высокая скорость разработки;
- недорогой вариант для многопользовательской одновременной работы;
- невысокая стоимость обновления и изменения программного обеспечения.

Однако у архитектуры файл-сервер есть ряд недостатков [8]:

- рост числа клиентов резко увеличивает объем трафика и нагрузку на

сети передачи данных;

- высокие затраты на сопровождение сервисов бизнес-логики на каждой клиентской рабочей станции;
- низкая надежность системы.

На рисунке 3 приведена файл-серверная архитектура.

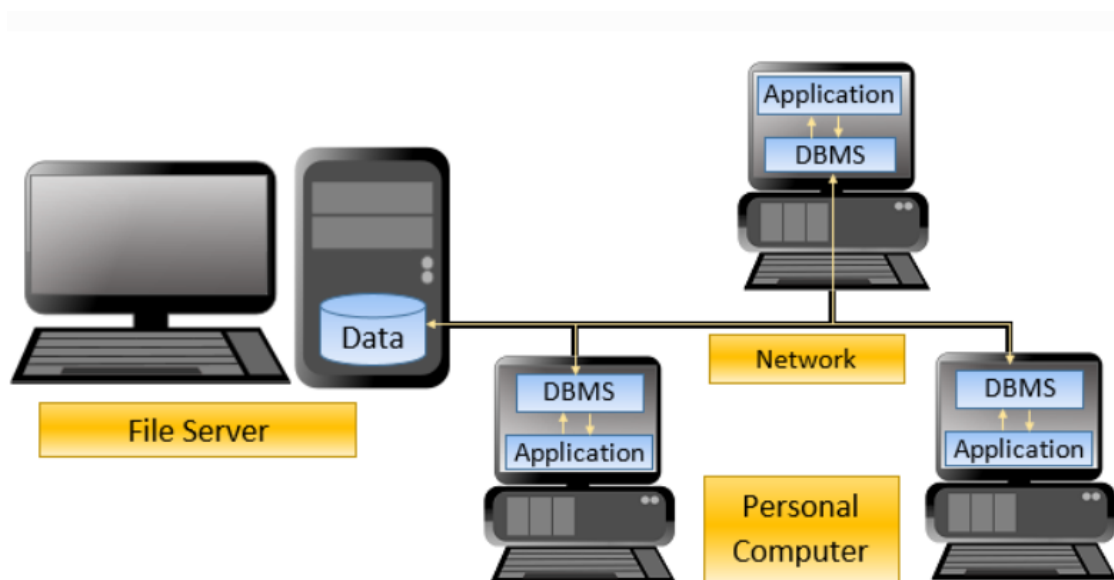


Рисунок 3 – Архитектура файл-сервер [9]

3.1.2 Клиент-серверные

Клиент-серверная СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Клиентом (front end) является запрашивающая машина, сервером (back end) — машина, которая отвечает на запрос. Характерной особенностью архитектуры клиент-сервер является перенос вычислительной нагрузки на сервер базы данных. Особенно при выполнении запроса в базе данных по заданному пользователем запросу, так как клиентскому приложению посылается результат выполнения запроса, по сети путешествуют только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Результат выполнения запроса, который на несколько порядков меньше по объему, чем весь база данных (как

в архитектуре файл-сервер), возвращается клиенту, после чего интерпретирует его необходимым образом и представляет пользователю.

На рисунке 4 приведена клиент-серверная архитектура.

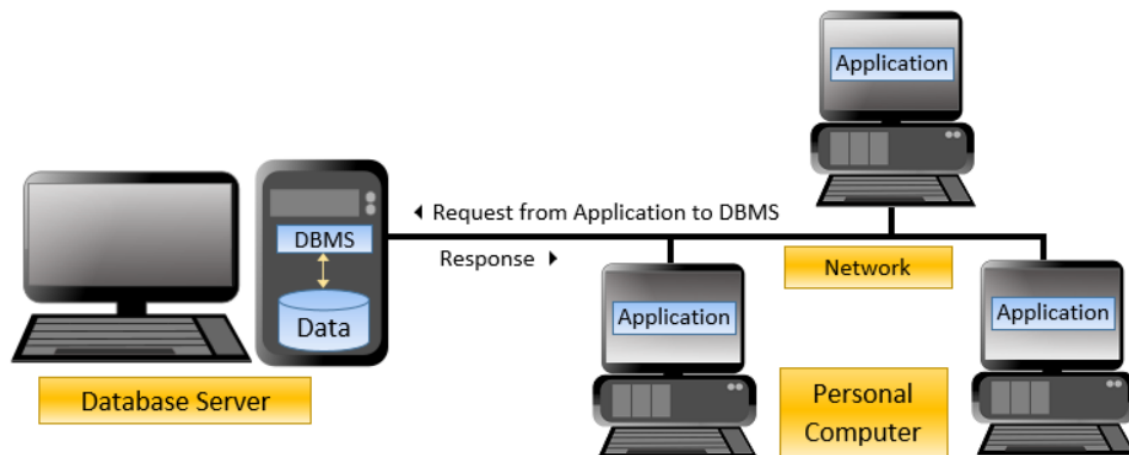


Рисунок 4 – Архитектура клиент-сервер [9]

База данных в этом случае помещается на сетевом сервере, как и в архитектуре файл-сервер, однако прямого доступа к базе данных из приложений не происходит. Функцию прямого обращения к базе данных осуществляет СУБД.

Архитектура клиент-серверная имеет следующие преимущества [8]:

- уменьшается сетевой трафик;
- уменьшается сложность клиентских приложений.

Однако у такой архитектуры есть несколько недостатков [8]:

- могут быть трудности с обновлением программного обеспечения на клиентах;
- высокая стоимость оборудования.

3.1.3 Встраиваемые

Встраиваемая СУБД — СУБД, которая может поставляться как составная часть некоторого программного продукта, не требуя процедуры самосто-

ятельной установки. Характер этой архитектуры допускает работу сотен или даже тысяч одновременных пользователей. Поскольку по сети передается минимум данных, он подходит для сред, где быстрые и недорогие сетевые соединения остаются недоступными. Встраиваемая СУБД предназначена для локального хранения данных своего приложения и не рассчитана на коллективное использование в сети. Физически встраиваемая СУБД чаще всего реализована в виде подключаемой библиотеки. Доступ к данным со стороны приложения может происходить через SQL либо через специальные программные интерфейсы. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном программном обеспечении, которое имеет дело с большими объемами данных.

На рисунке 5 приведена встраиваемая архитектура.

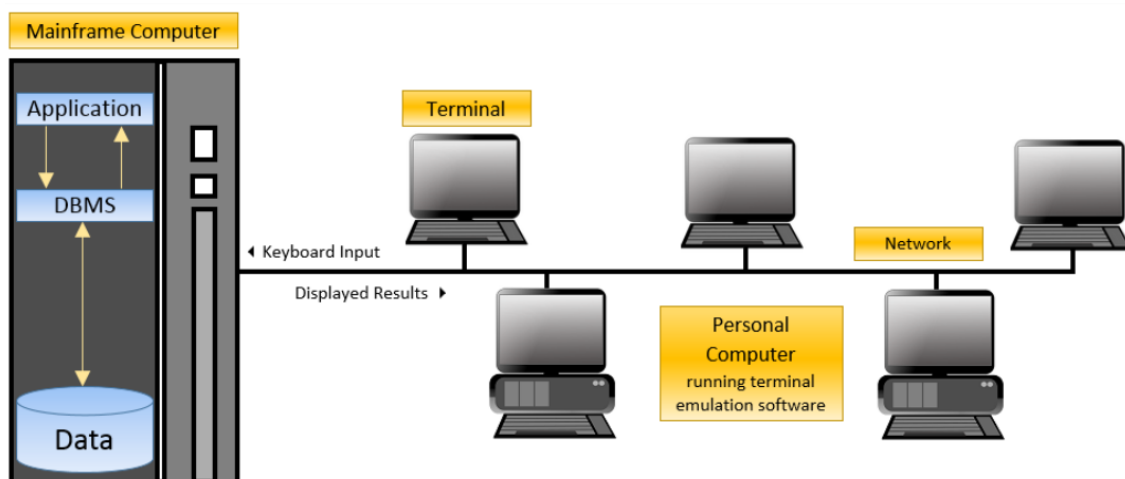


Рисунок 5 – Архитектура встраиваемая [9]

Встраиваемая архитектура имеет следующие преимущества [8]:

- легкость разработки и развертывания приложений;
- легкость обслуживания локальной БД;
- высокое быстродействие на операциях считывания и модификации одиночных записей.

Однако у такой архитектуры есть несколько недостатков [8]:

- высокий риск потери или повреждения данных;
- невозможность распределения вычислительной нагрузки.

3.2 Критерии сравнения архитектур

В данном разделе описаны критерии, которые используются для сравнения рассматриваемых архитектур в таблице 3.

Таблица 3 – Критерии сравнения архитектур СУБД

Критерий	Описание
Безопасность	Включая различные политики управления пользователями и контроля доступа к данным
Скорость разработки	Является ли архитектура быстрым в разработке
Производительность	Выполняются запросы с высокой скоростью и эффективностью
Гибкость	Определяет степень, до которой система может быть легко адаптирована
Простота эксплуатации	Оценка пользовательского интерфейса, документации и ресурсов поддержки
Стоимость	Внедрения и использования, включая лицензионные сборы, расходы на обслуживание и любые дополнительные затраты

3.3 Сравнение архитектур СУБД

Таблица 4 – Сравнение архитектур СУБД

Критерий	Файл-сервер	Клиент-сервер	Встр-мая
<i>Производительность</i>	✓/✗	✓	✓
<i>Безопасность</i>	✗	✓	—
<i>Скорость разработки</i>	✓	✓/✗	✓
<i>Гибкость</i>	✓/✗	✓	—
<i>Простота эксплуатации</i>	✓	✓/✗	✓
<i>Стоимость</i>	✓	✓/✗	✓

Полученные результаты сводятся к следующему:

- Файл-сервер — самая низкая стоимость и эффективная в использовании и высокая скорость разработки. Однако она имеет ограниченную гибкость, особенно в масштабировании, так же низкую безопасность и среднюю производительность.
- Клиент-сервер — самая гибкая архитектура, вторая по производительности и самая безопасность с возможностью централизованного управления доступом. Однако она имеет сложность в разработке а также среднюю простоту эксплуатации, так как требует более высокого уровня управления. Также, такая архитектура требует большие затраты на обслуживание и конфигурацию сервера.
- Встраиваемая — самая высокая по производительности, так как интегрирована напрямую в приложение. Также, она имеет высокую скорость разработки и легкость в эксплуатации и низкую стоимость. Ее безопасность зависит от функций, предоставленных встроенной СУБД, её гибкость зависит от функциональности встроенной СУБД.

ЗАКЛЮЧЕНИЕ

Из результатов сравнения было выявлено, что каждая из архитектур СУБД различена относительно других архитектур по своим преимуществам и недостаткам. Таким образом, выбор архитектуры СУБД зависит в первую очередь от типа проекта, которые необходимо проектировать, и нет единой архитектуры, которая была бы наилучшей для решения всех задач. Разработанная в ходе работы методика позволяет сравнивать архитектуры СУБД по модели данных, по способу доступа к базе данных и определять, какая из архитектур подходит для решения конкретной задачи.

Цель данной работы достигнута. В ходе данной работы были решены все задачи:

- проведен анализ предметной области системы управления базами данных;
- проведен обзор существующих систем управления базами данных по модели данных и по способу доступа к базе данных;
- сформулированы критерии сравнения этих систем;
- классифицированы существующие системы управления базами данных по модели данных и по способу доступа к базе данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 И.П. Карпова. Базы данных: учебное пособие. — Московский государственный институт электроники и математики (Технический университет)., 2009. С. 6 – 8.
- 2 Андреев А.В. Классификация СУБД. [Электронный ресурс]. Режим доступа: <https://scienceforum.ru/2016/article/2016019197>.
- 3 Инструментальное программное обеспечение. [Электронный ресурс]. Режим доступа: <https://laleshin.narod.ru/pto/T-2-2.pdf>.
- 4 Сетевая СУБД. [Электронный ресурс]. Режим доступа: <https://www.tadviser.ru/index.php/>.
- 5 Что такое реляционная база данных? [Электронный ресурс]. Режим доступа: <https://aws.amazon.com/ru/relational-database/>.
- 6 Реляционная база данных. [Электронный ресурс]. Режим доступа: <https://appmaster.io/ru/blog/chto-takoe-reliatsionnaia-baza-dannykh>.
- 7 Критерии выбора СУБД при создании информационных систем. [Электронный ресурс]. Режим доступа: <http://citforum.ru/database/articles/criteria/>.
- 8 Классификация и выбор СУБД. [Электронный ресурс]. Режим доступа: https://andpop.ru/files/lection/DBMS/07_DBMS_classification.pdf.
- 9 Архитектура базы данных. [Электронный ресурс]. Режим доступа: <https://app.myeducator.com/reader/web/617b/chapter01/gj149/#:~:text=File%20server%20architecture%20is%20a,the%20data%20by%20the%20network>.

ПРИЛОЖЕНИЕ А

Презентация к научно-исследовательской работе

Презентация содержит 14 слайдов.