

В настоящее время для оперативного, гибкого и эффективного управления предприятиями, фирмами и организациями различных форм собственности, телекоммуникационными средствами гражданского и военного назначения, информационно-вычислительными, экологическими, радиолокационными системами широко внедряются системы автоматизированного управления, ядром которых являются базы данных (БД). При большом объеме информации и сложности, производимых с ней операций проблема эффективности средств организации хранения, доступа и обработки данных приобретет особое значение. Учитывая важность и значимость баз данных в современной жизни, весьма серьезные требования предъявляются к квалификации специалистов, создающих приложения на их основе[1]. Приложение баз данных, как следует уже из названия, предназначено для взаимодействия с некоторым источником данных — базой данных (БД). Взаимодействие подразумевает получение данных, их представление в определенном формате для просмотра пользователем, редактирование в соответствии с реализованными в программе бизнес-алгоритмами и возврат обработанных данных обратно в базу данных. В качестве источника данных могут выступать как собственно базы данных, так и обычные файлы — текстовые, электронные таблицы и т. д. Базы данных обслуживаются специальными программами — системами управления базами данных, которые делятся на: · локальные, преимущественно одно-пользовательские, предназначенные для настольных приложений, · серверные — сетевые (часто удаленные), многопользовательские, функционирующие на выделенных компьютерах — серверах. Механизм внутреннего представления данных является ядром приложения баз данных. Он обеспечивает хранение полученных данных в приложении и предоставляет их по запросу других частей приложения. Пользовательский интерфейс обеспечивает просмотр и редактирование данных, а также управление данными и приложением в целом. Бизнес-логика приложения представляет собой набор реализованных в программе алгоритмов обработки данных. Между приложением и собственно базой данных находится специальное программное обеспечение (ПО), связывающее программу и источник данных и управляющее процессом обмена данными. Это ПО может быть реализовано самыми разнообразными способами, в зависимости от объема базы данных, решаемых системой задач, числа пользователей, способами соединения приложения и базы данных. Специфика республики Йемен в данном вопросе заключается в следующем: каждая провинция в Йемен имеет свое направление в производстве и каждый гражданин должен проживать именно там, где работает. Соответственно вопрос аренды жилья становится более чем, актуален. Однако, имеющиеся в республике риелторские агентства, не имеют общих баз данных для эффективного обмена информацией. Решение данного вопроса было предложено магистром кафедры автоматизированных систем сбора и обработки информации

Казанского национального исследовательского технологического университета Обеди Абдулфаттахом. Совместно с преподавателями кафедры была разработана распределенная автоматизированная система управления агентства недвижимости для республики Йемен. При исследовании подобных систем, оказалось, что существуют всего несколько программных продуктов в виде веб-сайтов, которые предоставляют услуги по недвижимости, либо в виде настольных приложений, которые помогают в работе агентств недвижимости. Наиболее заметные недостатки таких продуктов в том, что:

- Они не выполняют всю работу предприятия, а лишь её часть, такие как сохранение и поиск объектов.
- Используют локальную базу данных.

Данная разработанная распределенная система позволяет выполнять следующие функции:

- Операции купли-продажи и аренды.
- Сохранение и поиск данных объектов, клиентов, сделок, и сотрудников предприятия.
- Оформление и вывод отчетов и готовых договоров операции купли-продажи и аренды.
- Расчет объемов продаж, аренды и прибыли за определенный период времени.
- Представление данных в виде графиков и диаграмм для анализа статуса операции купли-продажи и аренды.
- Использование всеми филиалами агентства одной распределенной базы данных.

При разработке системы такого класса возникает ряд задач, среди которых наиболее важными является следующие:

- Обеспечение высокой скорости передачи и обмена данными между серверами.
- Обеспечение высокой скорости ответа на запросы, при обращении к данным, и увеличение скорости обработки данных.
- Формирование удобных форм клиентского приложения для обработки данных.
- Анализ производительности и оптимизация работы системы.

В имеющихся стандартах и технологиях построения сети передачи данных для решения первой задачи разработаны несколько современных стандартов для построения сети с высокими качествами передачи, в которых скорость передачи достигается до 100 мб/с, такие как: Стандарт IEEE 802.3 (сеть Ethernet), Стандарт IEEE 802.5 (сеть Token-Ring) и Сеть FDDI (стандарт ISO 9314 ) и другие. Другие задачи решаются путем применения разработанной процедуры оптимизации реляционных баз данных, которая называется построением нормальных форм, или нормализацией базы данных. Суть этой процедуры в том, что таблицы могут содержать повторения данных, что затрудняет манипуляцию данными. Поэтому большие таблицы разбиваются на части с меньшим числом атрибутов [2]. Следует выделить два класса систем распределенной обработки и системы распределенных данных:

- Системы распределенной обработки в основном отражают структуру и свойства многопользовательских операционных систем с базой данных, размещенной на центральном компьютере.
- Системы распределенных данных обеспечивают обработку распределенных запросов, когда для обработки одного запроса используются информационные ресурсы, размещенные на различных ЭВМ сети. При этом, следует говорить как о распределенных файловых системах, так и о распределенных базах данных. Для

распределенных баз данных свойственны следующие характеристики:

- база данных – это логически свя-занные, разделяемые на некоторое количество фрагментов данные;
- фрагменты распределяются по разным узлам, которые связаны между собой сетевыми соединениями;
- может быть предусмотрена репликация фрагментов;
- доступ к данным на каждом узле происходит под управлением СУБД, которая на каждом узле должна поддерживать работу, как локальных приложений, так и глобальных. Распределенная БД состоит из набора узлов, связанных коммуникативной сетью, в которой каждый узел – это полноценная СУБД сама по себе. Узлы взаимодействуют между собой таким образом, что пользователь любого из них может получить доступ к любым данным в сети так, как-будто они находятся на его собственном узле. Каждый узел по себе является системой базы данных. Любой пользователь может выполнить операции над данными на своем локальном узле точно так же, как если бы этот узел вовсе не входил в распределенную систему. Распределенную систему баз данных можно рассматривать, как партнерство между отдельным и локальным СУБД, на отдельных локальных узлах[3].

Фундаментальный принцип создания распределенных баз данных: для пользователя: распределенная система должна выглядеть так же, как нераспределенная система. Следствием данного принципа являются определенные правила:

- Локальная независимость. Узлы в распределенной системе должны быть независимы или автономны. Локальная независимость означает, что все операции на узле контролируются этим узлом.
- Отсутствие опоры на центральный узел. Локальная независимость предполагает, что все узлы в распределенной системе должны рассматриваться как равные. Поэтому не должно быть никаких обращений к «центральному» или «главному» узлу с целью получения некоторого централизованного сервиса.
- Непрерывное функционирование. Распределенные системы должны предоставлять более высокую степень надежности и доступности.
- Независимость от расположения. Пользователи не должны знать, где именно данные хранятся физически.
- Независимость от фрагментации. Система поддерживает независимость от фрагментации, если данная переменная - отношение может быть разделена на части или фрагменты при организации ее физического хранения. В этом случае данные могут храниться в том месте, где они чаще всего используются, что позволяет достичь локализации большинства операций и уменьшения сетевого трафика.
- Независимость от репликации. Система поддерживает репликацию данных, если данная хранимая переменная-отношение – или в общем случае данный фрагмент данной хранимой переменной-отношения – может быть представлен несколькими отдельными копиями или репликами, которые хранятся на нескольких отдельных узлах.
- Обработка распределенных запросов. Суть в том, что для запроса может потребоваться обращение к нескольким узлам. В такой системе может быть много возможных способов пересылки данных, позволяющих

выполнить рассматриваемый запрос. · Управление распределенными транзакциями. Существуют два главных аспекта управления транзакциями: управление восстановлением и управление параллельностью обработки. Что касается управления восстановлением, то, чтобы обеспечить атомарность транзакции в распределенной среде, система должна гарантировать, что все множество относящихся к данной транзакции агентов, или зафиксировало свои результаты, или выполнило отказ. Агент – процесс, который выполняется для данной транзакции на отдельном узле. Что касается управления параллельностью, то оно в большинстве распределенных систем базируется на механизме блокирования. Так же, как и в нераспределенных системах. · Аппаратная независимость. Желательно иметь возможность запускать одну и ту же СУБД на различных аппаратных платформах и, более того, добиться, чтобы различные машины участвовали в работе распределенной системы как равноправные партнеры. · Независимость от операционной системы. Возможность функционирования СУБД под различными операционными системами. · Независимость от сети. Возможность поддерживать много принципиально различных узлов, отличающихся оборудованием и операционными системами, также ряд типов различных коммуникационных сетей. · Независимость от типа СУБД. Необходимо, чтобы экземпляры СУБД на различных узлах все вместе поддерживали один и тот же интерфейс, и совсем необязательно, чтобы это были копии одной и той же версии СУБД [4].

Предложенные технологии и методы создания распределенной системы позволили: построить сети передачи данных по новым технологиям и разработать распределенную автоматизированную систему обработки данных, которая отвечает всем современным требованиям.