

Тема: «Этапы развития СУБД»

Выполнила:
студентка 2-го курса ИВТ Кочеткова Мария

Первый этап развития СУБД

Связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation – DEC), разных моделях HP (фирмы Hewlett Packard).

Основные даты

- В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM.
- В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных — Conference of Data System Languages (CODASYL), который определил ряд фундаментальных понятий в теории систем баз данных, которые и до сих пор являются основополагающими для сетевой модели данных.

Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме.

Программы доступа к БД писались на различных языках, включая низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным, и запускались как обычные числовые программы.

Мощные операционные системы обеспечивали возможность условно параллельного выполнения всего множества задач. Эти системы можно было отнести к системам распределенного доступа, потому что база данных была централизованной, хранилась на устройствах внешней памяти одной центральной ЭВМ, а доступ к ней поддерживался от многих пользователей-задач.

ОСОБЕННОСТИ ПЕРИОДА

- Все СУБД базируются на мощных мультипрограммных операционных системах (MVS, SVM, RTE, OSRV, RSX, UNIX)
- Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС).
- Значительная роль отводится администрированию данных.
- Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных
- Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции.
- Появляются первые языки высокого уровня для работы с реляционной моделью данных. Однако отсутствуют стандарты для этих первых языков.

Второй этап развития СУБД

Связан с развитием персональных компьютеров.

Основные даты

- 1981 - была создана реляционная модель данных под руководством Э.Ф.Кодда, который применил к ней операции реляционной алгебры
- 1989 – создана Microsoft SQL Server. Система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft.

Представители этого семейства — очень широко использовавшиеся до недавнего времени СУБД Dbase (DbaseIII+, DbaseIV), FoxPro, Clipper, Paradox.

Персональные компьютеры начали всё больше появляться на рынке, они стали доступнее. Обычные пользователи начали интересоваться программами, но не стремились осваивать языки программирования, в ходе чего начали появляться и распространяться множественные приложения, предназначенные для неподготовленных пользователей, такие программы имели развитые и удобные пользовательские интерфейсы. Большинство СУБД предлагали развитый и удобный инструментарий для разработки готовых приложений без программирования (на основе готовых шаблонов экранных форм, отчетов, графических конструкторов запросов).

Из-за кажущейся простоты разработки СУБД разработчиками-дилетантами было создано множество систем-однодневок. Доступность ПК позволила увеличить количество областей знаний, которые применяли вычислительную технику в своей деятельности, так появились «настольные» (desktop) СУБД.

За счёт конкуренции среди поставщиков ПО системы сильно совершенствовались (добавление новых возможностей, улучшение интерфейса и быстродействия систем), однако снижая их стоимость.

Не зависимо от периода появлялись идейные разработчики, которые при создании СУБД использовали стандартные языки программирования. Однако, как показало дальнейшее развитие, перенос данных из нестандартных форматов в новые СУБД оказался гораздо труднее. В некоторых случаях трудозатраты были настолько велики, что легче было разработать новую СУБД.

ОСОБЕННОСТИ ПЕРИОДА

- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом. В редких случаях предполагалась последовательная работа нескольких пользователей
- Во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, то есть только внешний табличный вид структур данных.
- При наличии высокоуровневых языков манипулирования данными типа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.
- В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных.
- Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД и в связи с этим — к отсутствию инструментальных средств администрирования БД.
- Сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД.
- Было проведено много исследований параллельных и распределенных баз данных, а также начата работа над объектно-ориентированными базами данных
- Реляционная модель стала доминирующей среди моделей данных.

Третий этап развития СУБД

Связан с ростом локальных сетей и решением их проблем

В этот период началась разработка стандартов языков описания и манипулирования данными SQL89, SQL92, SQL99 и технологий по обмену данными между различными СУБД.

Разработка концепцией объектно-ориентированных БД — ООБД. Представителями СУБД, относящимся к этапу, можно считать MS Access 97 и все современные серверы баз данных Oracle7.3, Oracle 8.4 MS SQL6.5, MS SQL7.0, System 10, System 11, Informix, DB2, SQL Base и другие современные серверы баз данных, которых в настоящий момент насчитывается несколько десятков.

При значительном росте количества локальных сетей остро встает задача согласованности данных, хранящихся и обрабатывающихся в разных местах, но логически друг с другом связанных, возникают задачи, связанные с параллельной обработкой транзакций — последовательностей операций над БД, переводящих ее из одного непротиворечивого состояния в другое непротиворечивое состояние. Успешное решение этих задач приводит к появлению распределенных баз данных, сохраняющих все преимущества настольных СУБД и в то же время позволяющих организовать параллельную обработку информации и поддержку целостности БД, но реализация такой архитектурой связана с рядом проблем, включающим в себя необходимость сложного ПО, крупную общую стоимость, проблемы безопасности, отсутствие стандартов и т.д.

Множится количество объединений ПК в локальные сети, все больше информации передается между компьютерами, что приводит к возникновению клиент-серверной модели, а также модели с совместным использованием файлов. Сеть позволяла совместно использовать дорогие принтеры и дисковые накопители большой емкости. В перспективе же пользователи хотели совместного использования их баз данных, что стимулировало развитие многопользовательских приложений баз данных для локальных сетей.

Поскольку многопользовательская обработка данных в локальной сети отличается от многопользовательской обработки данных на мейнфрейме наличием нескольких вычислителей, возникали дополнительные сложности по координации действий вычислителей. Так появилась клиент-серверная архитектура обработки данных. Клиент-серверные СУБД также имеют ряд недостатков, такие как высокие требования к пропускной способности коммуникационных каналов с сервером; слабая защита данных от взлома, в особенности от недобросовестных пользователей системы; высокая сложность администрирования и настройки рабочих мест пользователей системы и т.д.

ОСОБЕННОСТИ ПЕРИОДА

- Практически все современные СУБД обеспечивают поддержку полной реляционной модели, а именно:
 - О структурной целостности — допустимыми являются только данные, представленные в виде отношений реляционной модели;
 - О языковой целостности, то есть языков манипулирования данными высокого уровня (в основном SQL);
 - О ссылочной целостности, контроля за соблюдением ссылочной целостности в течение всего времени функционирования системы, и гарантий невозможности со стороны СУБД нарушить эти ограничения
- Большинство современных СУБД рассчитаны на многоплатформенную архитектуру, то есть они могут работать на компьютерах с разной архитектурой и под разными операционными системами.
- Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали развития средств администрирования БД с реализацией общей концепции средств защиты данных.
- Создание теоретических трудов по оптимизации реализаций распределенных БД и работе с распределенными транзакциями и запросами с внедрением полученных результатов в коммерческие СУБД.
- Для того чтобы не потерять клиентов, которые ранее работали на настольных СУБД, практически все современные СУБД имеют средства подключения клиентских приложений, разработанных с использованием настольных СУБД, и средства экспорта данных из форматов настольных СУБД второго этапа развития.

Четвертый этап развития СУБД

Связан с появлением новой технологии доступа к данным — интранет.

Основное отличие этого подхода от технологии клиент-сервер состоит в том, что отпадает необходимость использования специализированного клиентского программного обеспечения.

У каждого из упоминаемых подходов к работе с данными есть свои достоинства и свои недостатки, которые и определяют область применения того или иного метода, и в настоящее время все подходы широко используются.

Основное отличие этого подхода от технологии клиент-сервер состоит в том, что отпадает необходимость использования специализированного клиентского программного обеспечения. Для работы с удаленной базой данных используется стандартный браузер. При этом встроенный в загружаемые пользователем HTML-страницы код, написанный обычно на языке Java, Java-script, Perl и других, отслеживает все действия пользователя и транслирует их в низкоуровневые SQL-запросы к базе данных, выполняя, таким образом, ту работу, которой в технологии клиент-сервер занимается клиентская программа.

Удобство данного подхода привело к тому, что он стал использоваться не только для удаленного доступа к базам данных, но и для пользователей локальной сети предприятия. Простые задачи обработки данных, не связанные со сложными алгоритмами, требующими согласованного изменения данных во многих взаимосвязанных объектах, достаточно просто и эффективно могут быть построены по данной архитектуре. В этом случае для подключения нового пользователя к возможности использовать данную задачу не требуется установка дополнительного клиентского программного обеспечения. Однако алгоритмически сложные задачи рекомендуется реализовывать в архитектуре «клиент-сервер» с разработкой специального клиентского программного обеспечения.