

Министерство образования и науки Украины
Национальный технический университет Украины
«Киевский политехнический институт им. Игоря Сикорского »
Факультет информатики и вычислительной техники
Кафедра вычислительной техники

Лабораторная работа № 1

по дисциплине «Организация баз данных»
на тему «Система управления данными»

ВЫПОЛНИЛА:
Группа 2-го курса ФИВТ
группы ИВ-83

ПРОВЕРИЛ:
Доцент каф. ОТ
к.т.н., с.н.с.
Болдак А.О.

План документации:

1. Основная характеристика и история создания СУБД
2. Управления данными
3. Принципы организации данных, лежащие в современных СУБД

Основная характеристика и история создания СУБД

Все СУБД можно разделить:

- на профессиональные, или промышленные;
- персональные (настольные).

Профессиональные (промышленные) СУБД представляют собой программную основу для разработки автоматизированных систем управления крупными экономическими объектами. На их базе создаются комплексы управления и обработки информации крупных предприятий, банков или даже целых отраслей. Первостепенными условиями, которым должны удовлетворять профессиональные СУБД, являются:

- возможность организации совместной параллельной работы большого количества пользователей;
- устойчивость по отношению к сбоям различного рода, в том числе наличие многоуровневой системы резервирования хранимой информации;

Промышленные СУБД к настоящему моменту имеют уже достаточно богатую историю развития. В настоящее время характерными представителями профессиональных СУБД являются такие программные продукты, как Oracle, DB2, Sybase, Informix, Progress.

Основоположниками СУБД Oracle стала группа американских разработчиков, которые более двадцати лет тому назад создали фирму Relational Software Inc. и поставили перед собой задачу создать систему, на практике реализующую идеи, изложенные в работах Э. Ф. Кодда и К. Дж. Дейта. Результатом их деятельности стала реализация переносимой реляционной системы управления базами данных с базовым языком обработки 50Б. В 1979 г. заказчикам была представлена версия Oracle для мини-компьютеров PDP-11 фирмы Digital Equipment Corporation сразу для нескольких операционных систем: RSX-11, IAS, RSTS и UNIX.

Чуть позже Oracle был перенесен на компьютеры VAX под управлением VAX VMS. Значительная часть кода была написана на ассемблере, и поэтому процесс переноса системы на новую платформу требовал значительных

усилий. Основным отличием Oracle очередной, третьей версии было то, что она была полностью написана на языке С. Такое решение обеспечивало переносимость системы на многие новые платформы, в частности, на различные клоны UNIX.

Второй важной особенностью новой (1983 г.) версии была поддержка концепции транзакции. Примерно в это же время фирма получила новое имя — Oracle Corporation— и заняла лидирующее место на рынке производителей СУБД. Четвертая версия Oracle характеризовалась расширением перечня поддерживаемых платформ и операционных систем. Oracle был перенесен как на большие ЭВМ фирмы IBM (мэйнфреймы), так и на персональные компьютеры, работающие под управлением MS DOS.

В четвертой версии был сделан важный шаг в развитии технологий поддержки целостности баз данных. Для многопользовательских систем было предложено оригинальное решение Oracle поддержки «непротиворечивости чтения». В пятой версии была впервые реализована СУБД с архитектурой «клиент - сервер». Последующие версии СУБД Oracle были ориентированы на построение крупномасштабных систем обработки транзакций, изменение методов реализации систем ввода/вывода, буферизации, подсистем управления параллельным доступом, резервирования и восстановления. Также была реализована поддержка симметричных мультипроцессорных архитектур.

Проект и экспериментальный вариант СУБД Ingres были разработаны в университете Беркли под руководством одного из наиболее известных в мире ученых и специалистов в области баз данных Майкла Стоунбрейкера. С самого начала СУБД Ingres разрабатывалась как мобильная система, функционирующая в среде ОС UNIX.

Первая версия Ingres была рассчитана на 16-разрядные компьютеры. И работала главным образом на машинах серии PDP. Это была первая СУБД, распространяемая бесплатно для использования в университетах. Впоследствии группа Стоунбрейкера перенесла Ingres в среду ОС UNIX BSD, которая также была разработана в университете Беркли. Семейство СУБД Ingres из университета Беркли принято называть университетской Ingres. В начале 80-х была образована компания RTI (Relational Technology Inc.), которая разработала и стала продвигать коммерческую версию СУБД Ingres. В настоящее время коммерческая Ingres поддерживается, развивается и продается компанией Computer Associates. Сейчас это одна из наиболее развитых коммерческих реляционных СУБД. В то же время, по поводу университетской Ingres имеется много высококачественных публикаций.

Более того, университетскую Ingres можно опробовать на практике и даже посмотреть ее исходные тексты.

Перечисленные выше (для СУБД Oracle) тенденции носят универсальный характер и определяют пути развития других программных продуктов, что вполне, объясняется жесткой конкурентной ситуацией, сложившейся на данном рынке.

Персональные системы управления данными — это программное обеспечение, ориентированное на решение задач локального пользователя или компактной группы пользователей и предназначенное для использования на микроЭВМ (персональном компьютере). Это объясняет и их второе название — настольные. Определяющими характеристиками настольных систем являются:

1. Относительная простота эксплуатации, позволяющая создавать на их основе работоспособные приложения как «продвинутым» пользователям, так и тем, чья квалификация невысока;
2. Относительно ограниченные требования к аппаратным ресурсам.

Несмотря на неизбежные различия, обуславливавшиеся замыслами разработчиков, все перечисленные системы в ходе своей эволюции приобрели ряд общих конструктивных черт, среди которых, прежде всего, могут быть названы:

- наличие визуального интерфейса, автоматизирующего процесс создания средств, манипуляции данными, - экранных форм, шаблонов отчетов, запросов и т. п.;
- наличие инструментов создания объектов базы данных в режиме диалога: Experts в Paradox, Wizards в Access, Assistants в Approach;
- наличие развитого инструментария создания программных расширений в рамках единой среды СУБД: язык разработки приложений PAL в Paradox, VBA (Visual Basic for Applications в Access встроенная поддержка универсальных языков управления данными.

В последние годы наметилась устойчивая тенденция к стиранию четких граней между настольными и профессиональными системами. Последнее, в первую очередь, объясняется тем, что разработчики в стремлении максимально расширить потенциальный рынок для своих продуктов постоянно расширяют набор их функциональных характеристик.

УПРАВЛЕНИЕ ДАННЫМИ

- 1) НЕОБХОДИМОСТЬ УПРАВЛЕНИЯ ДАННЫМИ
- 2) ДИЗАЙН БАЗЫ ДАННЫХ
- 3) УПРАВЛЕНИЕ ДАННЫМИ И ЭКСПЛУАТАЦИЯ
- 4) ДОСТУП К ДАННЫМ И РАСПРОСТРАНЕНИЕ

- Данные должны храниться надежно, но должны быть легко доступны для анализа.
- База данных должна хранить исходные необработанные данные.
- Система управления данными должна быть максимально интегрирована с системой сбора данных.
- Интерфейс «человек-компьютер» должен помочь пользователю получить максимальную отдачу от системы.
- Ввод данных должен включать функции импорта и средства проверки, обработка должна использовать встроенные функции для общих процедур, а отчетность должна быть гибкой и включать средство экспорта.
- Доступ должен контролироваться, чтобы обеспечить целостность и конфиденциальность базы данных.

1) Необходимость управления данными

Принятие решений при разработке, планировании и управлении, в значительной степени зависит от обработанной информации. Данные должны быть интерпретированы, прежде чем они могут быть использованы. Объем необработанных первичных данных часто очень велик, и поэтому его можно эффективно использовать только в том случае, если он хранится в системе управления базами данных (СУБД).

Функции СУБД:

- Обеспечить соответствие данных стандартным классификациям;
- Обеспечить достоверность данных;
- Обеспечить целостность данных и внутреннюю согласованность;
- Обеспечить и поддерживать первичные данные;
- Обеспечить легкий доступ к первичным данным;
- Эффективно обрабатывать данные по мере необходимости;
- Разрешить интеграцию различных наборов данных, тем самым увеличивая их общую полезность.

Основополагающим принципом является хранение всех данных в том виде, в котором они были собраны, в их первичной форме. Это обеспечивает гибкость в способах обработки данных (например, отфильтрованных, агрегированных, преобразованных) и гарантирует, что все вычисления воспроизводятся из исходных данных, включающих все ревизии. Учитывая значительные инвестиции в сбор данных и низкие затраты на хранение и обработку, нет особых причин не хранить полные данные в их первичной форме.

2) ДИЗАЙН БАЗЫ ДАННЫХ

2.1) Методология

2.2) Человеко-машинный интерфейс

2.3) Компьютеризированная документация

2.4) Ввод данных

2.5) Обработка данных

2.6) Представление данных

2.1) Методология

Следует рассмотреть децентрализованную структуру базы данных, чтобы упростить управление базой данных и проверку данных. В распределенной системе данные вводятся и проверяются локально, но связаны с другими базами данных для анализа.

При рассмотрении подхода к созданию новой системы сбора данных предлагаются различные варианты. Они включают:

- Взять коммерчески доступное программное обеспечение и адаптировать его к новым требованиям;
- Соединение системы с различными программными компонентами;
- Создание собственной системы с нуля.

Преимущества и недостатки различаются для каждого подхода и должны тщательно взвешиваться перед выделением ресурсов.

Важным преимуществом пользовательской разработки является то, что ее можно настроить так, чтобы она точно соответствовала методологии выборки данных, поэтому система будет более эффективной и легко принятой. Другим возможным преимуществом является то, что проект базы данных также может быть использован в качестве инструмента для содействия разработке программы сбора данных. Если две фазы разработки

происходят одновременно, использование общей терминологии (то есть идентификации видов, методов отбора проб) и инструментов (то есть диаграмм потоков данных, анализа задач) может быть взаимовыгодным для обеих систем.

2.2) Человеко-машинный интерфейс

Важным для общей приемлемости СУБД является интерфейс «человек-компьютер» (НСІ). Ниже приведены некоторые основные принципы, которые могут быть использованы для разработки эффективных интерфейсов НСІ:

- Автоматизированные процедуры для руководства пользователей о том, как действовать при использовании системы;
- Использование графических структур, таких как командные кнопки в НСІ, предпочтительно с обычно применяемыми значками, для облегчения доступа к часто используемым функциям;

2.3) Компьютеризированная документация

Онлайновая помощь, документация, учебные пособия и обучение являются факторами, способствующими устойчивости базы данных. Особое внимание следует уделить разработке этих компонентов в системе. Предпочтительно, разработка этих компонентов должна идти параллельно с разработкой ПО / пользовательских интерфейсов. Однако это не устраняет необходимость в печатных копиях документации.

2.4) Ввод данных

Должна быть доступна функция «Импорт» для включения данных, обычно хранящихся в альтернативных форматах (например, текстовый процессор или электронная таблица). Эта функция должна обеспечивать целостность и качество данных.

Когда это применимо, следует разработать специальные структуры или программные ссылки для облегчения поиска данных из других компьютерных источников, таких как электронные журналы.

Проверка данных может осуществляться на различных уровнях, включая сбор, компиляцию, ввод данных в СУБД, обработку и анализ данных. Пользовательские интерфейсы ввода данных должны быть структурированы, чтобы обеспечить соблюдение наборов правил, применяемых для проверки входных данных.

2.5) Обработка данных

Особенностью технологии СУБД, которую следует использовать при разработке или изменении системы сбора данных, является возможность встраивания управления и обработки в базу данных с использованием хранимых процедур и запросов. Этот подход имеет следующие преимущества:

- Уменьшение количества необходимой внешней обработки;
- Обеспечение более немедленной проверки данных;
- Повышение гибкости для будущих модификаций системы.

Важным фактором при обработке данных является необходимость ведения контрольного журнала всех действий, выполняемых с данными, чтобы обеспечить последующий анализ качества информации.

Следует использовать параметры, чтобы сделать систему более гибкой. Параметры - это легко изменяемые значения, которые изменяют структуру и функции системы. Зачастую требования меняются в течение срока службы системы, и возможность расширения и модификации без существенных изменений конфигурации может сохранить жизнеспособность системы сбора данных.

2.6) Представление данных

Гибкость при создании отчетов из данных очень важна. Часто потенциальное использование данных не полностью распознается до того, как система заработает.

Для обеспечения гибкости отчета должна быть предусмотрена универсальная функция «экспорта». Функции, которые должна иметь эта функция:

- Определение тегов имен для всех экспортируемых атрибутов данных;
- Сводка типов данных и форматов;
- Записи переменной длины с выбранными пользователем разделителями полей (т.е. файлы ASCII с запятыми или табуляцией).

3) УПРАВЛЕНИЕ ДАННЫМИ И ЭКСПЛУАТАЦИЯ

3.1) Архивы

3.2) Переоценка проекта

3.1) Архивы

База данных должна регулярно создавать резервные копии. Система всегда должна быть готова к серьезным аппаратным или программным сбоям и потере данных. Процедуры должны быть сделаны как можно более простыми, чтобы обеспечить регулярное резервное копирование.

Поскольку база данных развивается со временем и происходят изменения в информационных технологиях, архивирование данных необходимо для извлечения исторических данных, сохраненных в прежней структуре или дизайне.

3.2) Переоценка проекта

В результате установленных механизмов обратной связи и для обеспечения того, чтобы система управления данными соответствовала своим целям, следует проводить периодические оценки.

Рекомендуется постоянная программа оценки дизайна, чтобы гарантировать, что система использует преимущества последних разработок в области информационных технологий. Особое внимание следует уделить установлению процедур обновления архивных данных, чтобы данные в старом формате по-прежнему были доступны.

4) ДОСТУП К ДАННЫМ И РАСПРОСТРАНЕНИЕ

4.1) Владение данными и контроль

4.2) Сети связи

4.3) Компьютеризированная публикация

4.1) Владение данными и контроль

Государство или учреждение, из которого были получены данные, является основным владельцем данных. Признавая, что данные являются ресурсом и, следовательно, имеют ценности, экономические или иные, правительство должно осуществлять свое право на поддержание, безопасность и контроль доступа.

Контроль - это ограничение, налагаемое на способность отдельного лица, группы лиц, организаций или другого государства иметь частичный или полный доступ к данным, содержащимся в базе данных. Частичный доступ к данным - это невозможность выполнения любого из следующих действий: просматривать все данные, введенные и сохраненные системой, добавлять данные, редактировать данные, копировать данные или распространять / делиться данными любым способом.

Элементы управления должны использоваться для ограничения доступа в соответствии с любыми требованиями конфиденциальности и защиты данных от несанкционированных изменений. Наибольшее значение имеет защита первичных данных от случайного повреждения. Основная копия данных всегда должна быть защищена от записи. Однако, хотя контроль и безопасность важны, они не должны препятствовать законному доступу. В частности, функции безопасности и контроля СУБД никогда не должны мешать признанным государством научным учреждениям получать доступ к данным для исследования управления ресурсами.

4.2) Сети связи

Разработки в области коммуникационных технологий открывают новые возможности для распространения данных. Когда это возможно и уместно, при разработке СУБД следует учитывать структуры, которые будут способствовать распространению или позволят осуществлять прямой доступ к данным из удаленных мест.

4.3) Компьютеризированная публикация

Разработка ПО для учебных пособий, демонстраций и связанных с ними документов имеет важное значение для долгосрочной жизнеспособности базы данных. Эти документы могут храниться локально или, предпочтительно, на национальном уровне в форме, обеспечивающей доступ к сети.

Использование цифровых медиа также следует учитывать для распространения статистики. Например, Интернет предлагает недорогой способ обмена информацией, обеспечивающий безопасный доступ к данным и аналитическим результатам.

Принципы организации данных, лежащие в основе современных СУБД.

По способу установления связей между данными различают:

- *реляционную*
- *иерархическую*
- *сетевую модели*

Реляционная модель является простейшей и наиболее привычной формой представления данных в виде таблицы. В теории множеств таблице соответствует термин отношение (relation), который и дал название модели. Для нее имеется развитый математический аппарат - реляционное исчисление и реляционная алгебра, где для баз данных (отношений) определены такие хорошо известные теоретико-множественные операции, как объединение, вычитание, пересечение, соединение и др.

Достоинством реляционной модели является сравнительная простота инструментальных средств ее поддержки, недостатком - жесткость структуры данных (невозможность, задания строк таблицы произвольной длины) и зависимость скорости ее работы от размера базы данных. Для многих операций, определенных в такой модели, может оказаться необходимым просмотр всей базы.

Иерархическая и сетевая модели предполагают наличие связей между данными, имеющими какой-либо общий признак. В иерархической модели такие связи могут быть отражены в виде дерева-графа, где возможны только односторонние связи от старших вершин к младшим. Это облегчает доступ к необходимой информации, но только если все возможные запросы отражены в структуре дерева. Никакие иные запросы удовлетворены быть не могут.

Указанный недостаток снят в сетевой модели, где, по крайней мере, теоретически, возможны связи "всех со всеми".

Поскольку на практике это, естественно, невозможно, приходится прибегать к некоторым ограничениям. Использование иерархической и сетевой моделей ускоряет доступ к информации в базе данных. Но поскольку каждый элемент данных должен содержать ссылки на некоторые другие элементы, требуются значительные ресурсы как дисковой, так и основной памяти ЭВМ. Недостаток основной памяти, конечно, снижает скорость обработки данных. Кроме того, для таких моделей характерна сложность реализации СУБД.

Наличие в СУБД языка программирования позволяет создавать сложные системы обработки данных, ориентированные под конкретные задачи и даже под конкретного пользователя.

Есть также СУБД, которые имеют только язык и не имеют оболочки пользователя. Они предназначены исключительно для программистов, и это системы компилирующего типа. Такие пакеты лишь с оговорками могут быть названы СУБД. Обычно их называют просто компиляторами.

Для легкого изменения или расширения баз данных, без изменения существующих приложений применяют пользовательский программный интерфейс - язык структурированных запросов или SQL.

SQL - декларативный язык программирования, который применяется для модификации или управления данными в соответствующей системе баз данных.

Преимущества и недостатки SQL:

Преимущества:

- 1) *Независимость от конкретной СУБД(система управления базами данных).*
- 2) *Наличие стандартов.*
- 3) *Спецификация решения задачи или Декларативность.*

Недостатки:

- 1) *Повторяющийся строки.*
- 2) *Неопределенные значения.*
- 3) *Высокая избыточность.*
- 4) *Использование указателей.*

Но, например, для больших наборов распределенных данных, будет удобно использовать NoSQL. NoSQL - это подход к проектированию баз данных, который способен вместить в себя большой спектр моделей данных. Он крайне эффективен, когда предприятию необходимо анализировать большой объем неструктурированных данных или данных, которые содержатся на виртуальных серверах.

Например, такие крупные компании, как Amazon или Google, использовали этот подход для описания и применения в узких оперативных целях, где была необходима высококачественная согласованность данных.

Преимущества и недостатки NoSQL:

Преимущества:

- 1) *Без ограничений на типы хранимых данных.*
- 2) *Лучше поддаются масштабированию.*
- 3) *Быстрая разработка.*
- 4) *Простые манипуляции с базами данных.*

Недостатки:

- 1) *Сильная привязанность к СУБД.*

- 2) *Переход с одной нереляционной базы данных на другую.*
- 3) *Собственные инструменты для БД.*

Когда речь заходит о виртуальной среде либо о публичном, частном или гибридном облаке, то применяют такое понятие, как Облачная база данных.

Облачная база данных представляет собой некий набор информации, который может быть структурным и неструктурным, и который в свою очередь находится на специальной платформе, то есть облако.

Важным моментом в описании этой базы является момент, где конкретно находится эта база данных.

Если для локальных баз данных можно достучаться через внутреннюю локальную сеть, то есть LAN, то для Облачных баз, данные находятся на специальных серверах или хранилищах, доступ к которым предоставляется через поставщиков, обслуживающих эти данные, и доступ к ним возможно получить исключительно только через интернет.

Облачные базы данных можно разделить на две категории: реляционные и нереляционные.

Основное отличие между ними в том, что нереляционные базы данных не используют табличную модель. Вместо этого они хранят контент, независимо от его структуры, как один документ. Эта технология хорошо подходит для неструктурированных данных, таких, как контент в социальных сетях, фотографии а также ролики.

Теперь поговорим более подробно о типах облачных баз данных. Можно наблюдать как традиционную схему, или как базу данных в виде услуги (DBaaS). В традиционной среде облако работает с визуальной машинной, то есть задачи и качество управления базами данных ложатся на IT-сотрудников организации.

Преимущества и недостатки Облачных баз данных:

Преимущества:

- 1) *Отсутствие физической инфраструктуры.*
- 2) *Экономия затрат (снижение эксплуатационных расходов).*
- 3) *Мгновенная масштабируемость.*
- 4) *Гарантии производительности.*

Недостатки:

- 1) *Ограниченный доступ к базовым серверам.*
- 2) *Очень мало информации о том, как ваши данные защищены от угроз кибербезопасности.*