

Тема 2-2. Инструментальное программное обеспечение

*Основные положения. Системы управления базами данных (СУБД).
Основная характеристика и виды СУБД. Принципы выбора и поддержки.*

Инструментальная система – это программный продукт, обеспечивающий разработку прикладного программного обеспечения.

Программные продукты этого класса поддерживают все технологические этапы процесса проектирования, программирования (кодирования), отладки и тестирования создаваемых программ. К инструментальным системам относят: системы программирования; системы быстрой разработки приложений и системы управления базами данных.

Системы программирования (англ. «Programming system») включают различные языки программирования и ПО, способствующее написанию и отладке компьютерных программ (компилятор; интегрированную среду разработчика программ; отладчик; средства оптимизации кода программ; набор библиотек (возможно с исходными текстами программ); редактор связей; сервисные средства (утилиты) для работы с библиотеками, текстовыми и двоичными файлами; справочные системы; документатор исходного кода программы; систему поддержки и управления проектом программного комплекса).

Системы быстрой разработки приложений (англ. «rapid application development», RAD) представляют собой развитие обычных систем программирования. В RAD-системах во многом автоматизирован процесс программирования. Программист не пишет сам текст программы, а с помощью некоторых наглядных манипуляций указывает системе, какие задачи должны выполняться программой. После чего RAD-система сама генерирует текст программы.

Основу программного обеспечения баз данных (англ. «DataBase») представляют программные компоненты СУБД.

Система управления базами данных (СУБД, англ. «database management system», DBMS) – это комплекс программ, предназначенных для создания, ведения и совместного использования базы данных многими пользователями; это комплекс программных средств, предназначенный для создания структуры новой базы, наполнения её содержанием, редактирования содержимого и визуализации информации.

Далее будут рассматриваться системы управления базами данных.

Ядро СУБД обеспечивает создание БД, организацию ввода, обработки и хранения данных, т. е. именно то, что называется «управлением данными», а также настройку системы, средства тестирования, утилиты, обеспечивающие выполнение вспомогательных функций, таких как восстановление баз данных, сбор статистики о функционировании БД и др.

С точки зрения баз данных *СУБД* – это комплекс языков и программ, позволяющий создавать БД и управлять её работой; это инструментальная программа, предназначенная для создания на ЭВМ общей для множества приложений БД, поддержания её в актуальном состоянии и обеспечения эф-

фективного доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий.

В состав большинства СУБД включены программные компоненты, позволяющие автоматизировать проектирование систем обработки информации (генераторы отчётов, меню и др.). Подавляющее большинство СУБД работает в среде универсальных операционных систем (ОС) и взаимодействует с ОС при обработке обращений к БД.

Наличие в составе СУБД средств, ориентированных на разные категории пользователей позволяет работать с базой данных как профессионалам в области обработки данных, так и практически любым пользователям.

Укрупнено можно выделить персональные и корпоративные СУБД.

Персональные системы управления базами данных типа Clipper, Clarion и FoxPro не приспособлены для создания интегрированных (как правило, распределённых) систем, работающих с общей базой.

Отметим, что в абсолютном большинстве случаев речь идёт не о персональных, а о сетевых, в том числе распределённых (корпоративных) СУБД, предназначенных для создания интегрированных, многопользовательских систем и имеющих развитые словари данных.

Основные особенности *корпоративных СУБД* заключаются в том, что они изначально направлены на создание интегрированных, многопользовательских систем, имеют развитые словари данных, что значительно повышает роль системного анализа и моделирования при проектировании системы. Во-вторых, средства разработки данных в таких СУБД оптимизированы для коллективной разработки сложных систем в рамках единой продуманной стратегии. Всё это обуславливает неуклонно растущее количество успешных внедрений систем на базе профессиональных СУБД.

СУБД обеспечивает:

- описание и сжатие данных;
- манипулирование данными (запись, поиск, выдачу, изменение содержания);
- физическое размещение (изменение размеров блоков данных, записей, использование занимаемого пространства, сортировку);
- защиту от сбоев, поддержку целостности и восстановление;
- работу с транзакциями и файлами;
- безопасность данных.

Существует несколько типов СУБД.

По типу используемой модели данных обычно выделяют: иерархические, сетевые и реляционные СУБД. Развитие технологий обработки данных привело к появлению постреляционных, объектно-ориентированных, гибридных систем, многомерных БД, в разной степени соответствующих упомянутым классическим моделям.

В *иерархической системе* данные в соответствии с ветвящимся деревом их признаков располагаются в двухмерных файлах и образуют деревья признаков. Соответственно этому происходит и поиск необходимых сведений. В *иерархической модели* все записи связаны в виде древовидной структуры.

Первые иерархические и сетевые СУБД созданы в начале 1960-х годов в качестве среды и системы управления миллионами записей (связанных друг с другом иерархическим образом). Среди реализуемых на практике СУБД этого типа преобладает система *IMS* (англ. «Information Management System» компании IBM).

В иерархической модели отношения данных организованы в виде совокупностей деревьев, состоящих из объектов различных уровней, где *дерево* – это структура данных, в которой тип сегмента потомка связан только с одним типом сегмента предка.

Верхний уровень в такой модели занимает один объект. Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Второй уровень включает объекты второго уровня и т. д.

Объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами.

Иерархическая модель представляет собой связный неориентированный граф древовидной структуры, объединяющий сегменты.

Графически: *предок* – это точка на конце стрелки, а *потомок* – точка на острие стрелки.

В базах данных определено, что точки – это типы записей, а стрелки представляют отношения «один-к-одному» или «один-ко-многим».

Отношение «один-к-одному» возникает, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице.

Родительская реляция (таблица) – это таблица, поля которой входят в другую таблицу.

Дочерняя реляция (таблица) – это таблица, поля которой используют информацию из полей другой таблицы, являющейся по отношению к данной родительской.

Отношение «многие-ко-многим» возникает, когда многим записям в родительской таблице соответствуют несколько записей в дочерней таблице.

Например, случай отношения «Имеет текущий счёт» обладающего мощностью «многие-ко-многим» означает, что у одного клиента может быть несколько текущих счетов, и что каждым из них могут пользоваться несколько клиентов (рис. 8).



Рис. 8. Отношение «многие ко многим»

Для преобразования отношения «многие-ко-многим» целесообразно создать таблицу пересечения, представляющую элементы двух других таблиц, находящихся в отношении «один-ко-многим».

К недостаткам иерархической модели данных относят:

- отсутствие явного разделения логических и физических характеристик модели;
- дополнительные манипуляции, необходимые для представления не-иерархических отношений данных;
- непредвиденные запросы, которые могут требовать реорганизации базы данных.

Сетевые системы (модули СУБД)

Сети – это естественный способ представления отношений между объектами. Они широко применяются в математике, исследованиях операций, химии, физике, социологии и других областях знаний.

Сетевую модель можно представить математической структурой, называемой *направленным графом*. Направленный граф имеет простую структуру. Он состоит из точек или *узлов*, соединенных стрелками или ребрами. В контексте моделей данных узлы можно представить как типы записей данных, а ребра – как отношения «один-к-одному» или «один-ко-многим».

Сетевая модель является развитием и расширением иерархической модели СУБД. Структура графа делает возможными простые представления иерархических отношений (таких, например, как генеалогические данные). Основное отличие такой модели заключается в возможности организации связи каждого элемента с любым другим элементом модели, а также наличие указателей в обоих направлениях, соединяющих родственную информацию.

Когда в отношении между данными порожденный элемент имеет более одного исходного элемента, такое отношение нельзя описать с помощью иерархической модели. Его описывают в виде сетевой модели. Таким образом, если в иерархических структурах запись-потомок должна иметь только одного предка, то в сетевой структуре данных потомок может иметь любое число предков.

К недостаткам сетевой модели данных следует отнести высокую сложность и жёсткость схемы БД, сложность для понимания и выполнения обработки информации в БД обычным пользователем. Кроме того, в сетевой модели данных ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

В сетевой модели существует две основные структуры данных: типы записей и наборы (Таблица 5).

Реляционная модель данных организует и представляет данные в виде таблиц или реляций (англ. «relation» – отношений).

Модель характеризуется простой структурой данных, удобным для пользователя табличным представлением, возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Тип записей	Совокупность логически связанных элементов данных
Набор	отношение «один-ко-многим» между двумя типами записей
Простая сеть	структура данных, в которой все бинарные отношения имеют мощность «один-ко-многим»
Сложная сеть	структура данных, в которой одно или несколько бинарных отношений имеют мощность «многие-ко-многим»
Тип записи	формальная запись, созданная для того, чтобы преобразовать
Связи	сложную сеть в эквивалентную ей простую сеть

В *реляционных системах* данные представляются в форме двумерных таблиц, определяющих взаимосвязь (отношение) записей. Таблицы (сущности) включают присущие им атрибуты и могут быть связаны между собой отношениями «один-ко-многим», «один-к-одному» или «многие-ко-многим».

Таблица содержит строки и столбцы, на пересечении которых образуются ячейки (поля) и обладает следующими свойствами:

- все ячейки в столбце таблицы однородные, то есть все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.);
- каждый столбец имеет уникальное имя;
- в строке формируются сведения о каком-либо объекте (запись);
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

Неделимым элементом записи является *поле* – это данное, описывающее какой-либо аспект (или атрибут) объекта. Поля имеют имена. Разные таблицы могут иметь поля с одинаковыми именами, но лучше этого избегать.

В простых СУБД (системы FoxPro, Dbase, Access) табличная база данных реализуется как один файл или в рамках совокупности файлов. В последнем случае каждая таблица помещается в отдельном файле, а строки (записи) любой таблицы являются записями файла.

Существует два подхода к проектированию реляционной базы данных.

Первый подход заключается в том, что на этапе концептуального проектирования создаётся не концептуальная модель данных, а непосредственно реляционная схема базы данных, состоящая из определений реляционных таблиц, подвергающихся нормализации.

Второй подход основан на механическом преобразовании созданной ранее функциональной модели в нормализованную реляционную модель. Этот подход чаще всего используется при проектировании больших, сложных схем баз данных, необходимых для корпоративных информационных систем.

Реляционные СУБД характеризуются простотой, гибкостью и точностью, возможностью одновременно работать с данными, размещёнными в нескольких таблицах. Поэтому, реляционные БД ориентированы на быстрый доступ к небольшим объёмам данных.

В реляционной модели одинаково представлены элементы данных и их связи. Широко распространённой реляционной СУБД является программа

фирмы Microsoft Access. Специальные программы класса СУБД (Oracle, MS SQL, Adabas, Informix, MySQL, SQLite и др.) разрабатываются таким образом, чтобы предоставлять пользователям широкие возможности их развития.

К недостаткам традиционных реляционных СУБД можно отнести невысокую эффективность, возможность обеспечения быстрого поиска и выборки отдельных строк в таблице лишь с помощью индексов (такое решение эффективно только при обработке небольших групп записей и высокой интенсивности модификации данных в базах данных).

К постреляционным моделям СУБД можно отнести *Объектно-реляционную СУБД* (ОРСУБД), представляющую собой надстройку над реляционной схемой, поддерживающую некоторые технологии, реализующие объектно-ориентированный подход.

Характерными свойствами ОРСУБД являются: 1) комплексные данные, 2) наследование типа, и 3) объектное поведение.

Комплексные данные можно реализовать через постоянно-хранимые объекты англ. «(persistent objects)». Создание комплексных данных в большинстве существующих ОРСУБД основано на предварительном установлении схемы через определяемый пользователем тип (англ. «user-defined type», UDT). При этом можно использовать встроенные конструкторы составных типов.

Иерархия структурных комплексных данных предлагает дополнительное свойство, *наследование типа*. Это означает, что структурный тип может иметь подтипы, использующие все его атрибуты и содержащие дополнительные атрибуты, специфицированные в подтипе.

Объектное поведение закладывается через описание программных объектов. Такие объекты должны быть сохраняемыми и переносимыми для обработки в базе данных, поэтому они называются обычно как постоянные (или долговременные) объекты. Внутри базы данных все отношения с постоянным программным объектом есть отношения с его объектным идентификатором (OID).

К объектно-реляционным СУБД относятся Oracle Database, PostgreSQL и др.

Объектно-ориентированные системы (применяются с конца 1980-х годов) основаны на объектно-ориентированной архитектуре и позволяют работать со сложными типами данных, хранимых в виде объектов и отличаются высокой производительностью при обработке транзакций.

Под *объектом* понимается «нечто, имеющее четко определенные границы» (определение известного американского специалиста Г. Буча). Объекты, обладающие одинаковыми свойствами, составляют классы (например, курица, пингвин и чайка – объекты класса «птицы»). Обычно класс описывается как новый тип данных, а объекты (экземпляры класса) – определённые на его основе переменных¹.

¹ Зеленков, Ю. А. Введение в БД [Электронный ресурс].–Режим доступа: http://www.mstu.edu.ru/education/materials/zelenkov/ch_6_3.html.

Для объектно-ориентированных СУБД (ООСУБД) характерны использование предложений, касающихся управления объектами и правилами, управления распределёнными данными, языков программирования четвёртого поколения (4GL), технологии тиражирования данных и других достижений в области обработки данных. Таким образом, практически полностью снимаются ограничения на типы данных.

В объектно-ориентированных, в отличие от реляционных, базах данных хранятся не записи, а объекты. Такую модель можно рассматривать по-слойно, на разных уровнях абстракции, при этом имеется возможность определения новых типов данных и операций с ними.

ООСУБД обычно используют, когда требуется высокопроизводительная обработка данных, имеющих сложную структуру. Подходят они и для организации распределённых вычислений, а также для применения в деловой сфере.

К ООСУБД относят: Caché, Cerebrum, IBM Lotus Notes/Domino, Jasmine, ObjectStore.

Гибридные системы объединяют положительные качества реляционных и объектно-ориентированных систем.

Они соединяют средства обработки транзакций реляционных СУБД с поддержкой многочисленных типов данных объектно-ориентированных СУБД. Гибридные СУБД используют язык структурированных запросов (SQL).

Как отмечалось ранее, в данном пособии рассматриваются автоматизированные информационно-поисковые системы. Они базируются на использовании баз и банков данных.

База данных (БД, англ. «Database») – это поименованная совокупность взаимосвязанных данных, находящихся под руководством систем управления базами данных.

Банк данных (БНд, англ. «Data Bank») – это логическая, тематическая или иная совокупность баз данных; это так же система специально организованных данных, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного многоцелевого использования данных.

В БД размещают различные *виды информации*:

- библиографические данные (записи);
- фактографические данные (записи);
- полнотекстовые документы (записи);
- справочные данные (в том числе указатели);
- математические или численные (цифровые, табличные) данные;
- графические данные;
- мультимедийные данные.

Базы данных обычно имеют различную структуру и тип. Наиболее распространена двухмерная табличная структура организации данных фиксированной длины, используемая в реляционных БД. В этом случае они по горизон-

тали представляют последовательные упорядоченные записи (строки), содержащие данные, размещённые в отдельных полях (ячейках).

Все сведения, присутствующие в полях одной строки, образуют совокупность данных об объекте, например, о книге. В этом случае такая *запись* называется *библиографической (БЗ)* и соответствует *библиографическому описанию (БО)* конкретной книги. Чаще всего сведений в БЗ гораздо больше, чем в стандартном БО. Одна характеристика (реквизит) БО (например, инвентарный номер) по разным объектам содержится в полях, образующих колонку или столбец. Очевидно, что при поиске отдельного реквизита в БД производится выбор этого реквизита, а затем в соответствующем столбце осуществляется поиск.

Кроме того, используются и БД с произвольной (нефиксированной) длиной поля. В этом случае данные записываются последовательно с заранее оговоренной последовательностью полей записи и устанавливаемыми разделителями между ними. Обычно каждая такая запись содержит: маркер записи, состоящий из 24-х символов; справочник; поля данных переменной длины, отделённые друг от друга разделителем поля и разделитель записи. При этом для поиска отдельного реквизита нужно «просмотреть» всю БД.

По мнению несведущих оппонентов потребуется слишком много времени для поиска даже по одному реквизиту. Однако к подобным БД автоматически, в процессе работы с ними, создаются вспомогательные *индексные файлы*. Они содержат нумерацию записей в соответствии с содержанием полей в БД. Например, индексный файл для записей с числовым полем содержит последовательность номеров записей этой базы, которая ранжирована по возрастанию или убыванию чисел в соответствующем числовом поле. Индексные файлы могут быть созданы для любых полей БД.

Структура организации данных в БД определяет формат записи.

По топологии хранения данных различают локальные и распределённые БД.

Распределённые базы данных, РБД (англ. «Distributed DataBase», DDB) представляют определённым образом связанные между собой БД, рассредоточенные на какой-либо территории (локально или регионально), обеспечивающие свободный обмен информацией и поиск данных в них.

Такие БД могут располагаться на различных узлах компьютерной сети. Обычно в них применяются протоколы синхронизации транзакций, обеспечивается сокращение расходов на пересылку данных между узлами вычислительной сети в ходе выполнения распределённого запроса посредством репликации данных и др.

Выделяют однородные и неоднородные РБД.

Часто данные размещаются в БД и СУБД по месту своего возникновения или наиболее эффективного использования в ЭВМ, удалённых друг от друга на большие расстояния, хотя каждая из этих ЭВМ управляет своими локальными СУБД. Возникает необходимость решения задач с распределёнными БД путём организации между ЭВМ сети передачи данных по каналам связи, а также обеспечения технической и программной поддержки обмена данными между ними.

Для работы с *распределёнными данными* создаются системы управления распределёнными базами данных (СУРБД), оснащённые каталогами, хранящими структуру сети, информацию о локальных СУРБД и БД, а также программным обеспечением, управляющим взаимодействием прикладной программы и конкретной локальной БД сети.

Управление однотипными локальными СУРБД осуществляется просто. В противном случае в сеть РБД включают различные программные и технические устройства, обеспечивающие единый интерфейс, согласование и возможность выполнения информационных процессов (промежуточную интерфейсную СУРБД, протокол Z39.50 и др.).

Ныне построение современных распределённых информационных систем связано с реляционными и объектно-ориентированными СУБД.

По типологии доступа и характеру использования хранимой информации БД могут быть разделены на специализированные (например, для статистической обработки информации) и интегрированные.

Автоматизированные интегрированные СУБД обеспечивают доступ к удалённым информационным и техническим ресурсам, а также возможность работы различных категорий пользователей с разнородной по формам представления информацией. К ним относят локальные, корпоративные и глобальные сети.

По функциональному назначению (характеру решаемых с помощью БД задач и использования данных) выделяют операционные и справочно-информационные БД. К последним можно отнести ретроспективные БД (электронные каталоги библиотек, БД статистической информации и т.д.), используемые для информационной поддержки основной деятельности, и не предполагающие внесение изменений в существующие записи, например, по результатам этой деятельности.

Операционные БД предназначены для управления различными технологическими процессами. В этом случае данные не только извлекаются из БД, но и изменяются и добавляются, в том числе в результате этого использования.

По степени доступности выделяют общедоступные и БД с ограниченным доступом пользователей. В последнем случае говорят об управляемом доступе, индивидуально определяющем не только набор доступных данных, но и характер операций которые доступны пользователю.

Представленные классы не исчерпывают, а отражают исторически сложившееся состояние дел в сферах разработки и применения ИС (БД).

Как отмечалось ранее, БД и БнД создаются и эксплуатируются с помощью СУБД.

Системы управления базами данных можно классифицировать:

1) *по используемому языку общения*: замкнутые, имеющие собственные самостоятельные языки общения пользователей с БД; открытые, в которых для общения с БД используется язык программирования, «расширенный» операторами языка манипулирования данными (ЯМД).

2) *по числу поддерживаемых СУБД уровней моделей данных*: одно-,

двух-, трёхуровневые системы.

3) *по выполняемым функциям*: операционные, предполагающие иные виды обработки по получению информации, не хранящейся в явном виде в БД; информационные, позволяющие организовать хранение данных, поиск и выдачу нужных данных из БД и поддерживать их целесообразность и актуальность;

4) *по сфере применения*: универсальные, настраиваемые на любую предметную область путём создания соответствующей БД и прикладных программ; проблемно-ориентированные на определённые процедуры обработки данных, присущих конкретной области применения;

5) *по допустимым режимам работы*: пакетные; с использованием телеобработки.

В СУБД используют два способа взаимодействия с конечными пользователями:

1. *Распределение времени*. В этом случае каждый участник сети как бы пользуется собственной ЭВМ. Основная задача разработчиков и администраторов сети – защита данных от несанкционированного доступа и обеспечение взаимной изоляции участников.

2. *Обеспечение групповых решений* подразумевает организацию взаимодействия пользователей в процессе принятия решений. Данный метод сочетает коммуникационную, вычислительную технологию и технологию принятия решений для реализации группой лиц сложных неструктурированных задач.

Масштаб СУБД определяется уровнем организации, спектром информационного обслуживания, объёмом информационных потоков и массивов. В этом случае СУБД делятся на системы для:

- организации или её подразделения;
- локальных (региональных или отраслевых) структур;
- глобальных (межотраслевых, межрегиональных) служб.

Последние служат для предоставления удалённым пользователям доступа к информации по телекоммуникационным сетям.

Выбор и поддержка СУБД

Одинаковых СУБД не бывает. Следовательно, то, что хорошо работает в одной СУБД, несмотря на заверения производителей, может плохо работать или вообще не работать в другой.

При выборе любой программы первоначально рассматриваются её возможности, среда применения, пользователи, их квалификация и т. п.

СУБД должна удовлетворять текущим и будущим потребностям организации с учётом финансовых затрат на приобретение необходимого оборудования и самой системы, разработку и доработку необходимого программного обеспечения на её основе, а также обучение персонала. К другим критериям выбора СУБД следует отнести состав и масштаб решаемых с её помощью задач, требования к объёмам обрабатываемой информации и производительности СУБД, возможности поставщика осуществлять поддержку сво-

его продукта, приемлемые стоимость лицензии и эксплуатации СУБД и т. д. При этом заказчик может добавить свои специфические требования.

При выборе СУБД следует обратить внимание на её открытость (в первую очередь, переносимость и стандартную стыкуемость); стабильность, подтверждаемую общим временем на рынке и количеством инсталляций; перспективностью применения и др.

В каждом конкретном случае рекомендуется начинать с тестирования различных СУБД на возможность решения поставленных задач.

Если в результате выявляется много проблем, не разрешаемых в рамках выбранной СУБД, то целесообразно рассмотреть вопрос о выборе иной СУБД. Лучше решить это ещё на стадии проектирования, поскольку в дальнейшем замена СУБД, как правило, будет связана с целым набором проблем.

Важным параметром любой СУБД являются способы хранения и доступа к данным. Универсальных решений не существует, так каждый производитель СУБД создаёт собственные способы хранения и доступа к данным.

Другим важным аспектом считается возможности импорта и экспорта данных во внешние системы. Они могут обеспечиваться утилитами, входящими в состав конкретной СУБД, и (или) специальными средствами обмена данными проектируемой АБИС. Причём следует отметить, что если обмен данными осуществляется между однородными базами данных (одного производителя и одной версии), то проблемы не возникают и без ограничений используются средства импорта и экспорта данных во внутреннем формате данной СУБД. Если обмен данными реализуется между СУБД разных производителей, то в простейшем случае для импорта и экспорта самих данных могут использоваться текстовые файлы. Лучше же, когда имеется возможность конвертации данных между соответствующими СУБД, особенно, когда обмен данными осуществляется в режиме реального времени.

Немаловажно решить задачу одновременного редактирования одного документа несколькими пользователями. С этой целью обычно изменения, сделанные пользователем в экранной форме, запоминаются в специальном буфере данных. При этом то, что он первоначально получил для редактирования, также запоминается в буфере начальных данных. Если пользователь нажимает кнопку «ОК», то выполняется попытка зафиксировать изменения пользователя и выполняется проверка отсутствия изменений, зафиксированных другим пользователем. Если во время редактирования данные были кем-то изменены, то пользователь получает предупреждение об этом и ему может быть предложено просмотреть изменения, после чего принять решение – сохранить свои изменения поверх или отказаться от них.

На рынке СУБД существуют различные решения. К ним в первую очередь относятся Oracle, Microsoft SQL Server. Однако они обладают высокой стоимостью решений, высокими требованиями к аппаратному обеспечению и потребностью в окружении специфическим программным обеспечением.

В небольших системах можно использовать более простые, в том числе бесплатные или условно-бесплатные, СУБД, например, MySQL, SQLite и т. п. Кратко рассмотрим их.

MySQL компании Sun Microsystems представляет собой быстрый многопоточный, многопользовательский и надежный SQL-сервер баз данных². Сервер MySQL предназначен как для критических по задачам производственных систем с большой нагрузкой, так и для встраивания в программное обеспечение массового распространения для малых и средних приложений. Краткая характеристика СУБД MySQL:

- распространённость и популярность продукта не вызывает сомнений – одна из самых известных и используемых платформ в мире. Большая часть мелких и средних проектов, связанных с базами данных, выполняется с её применением;
- простота использования и администрирования. В MySQL активно используется развитый графический интерфейс пользователя. Сервер способен функционировать долгое время в полностью автономном режиме;
- надёжность и устойчивость сервера. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании, является идеальным решением для малых и средних приложений. Поддерживается многопоточность и несколько одновременных запросов;
- надёжность данных при их хранении и обработке. Резервирование данных осуществляется в автоматическом или полуавтоматическом режиме;
- совместимость MySQL с Windows XP/2003 Server и т. д. Совместимость с распространёнными серверами приложений и интерфейс с языками C# и Delphi;
- гибкость настроек баз данных. Большое число поддерживаемых типов данных (чисел длиной от 1 до 4 байт, строк переменной длины и меток времени). Встроена утилита проверки и ремонта таблицы (isamchk). Легкость управления таблицей, включая добавление и удаление ключей и полей. Все операции работы со строками не обращают внимания на регистр символов в обрабатываемых строках. Гибкая система привилегий и паролей. До 16 ключей в таблице. Каждый ключ может иметь до 15 полей.

MySQL – это ПО с открытым кодом объёмом БД до 8 гигабайт. Это означает, что применять и модифицировать его может любой желающий. Его можно получать по Интернету и использовать бесплатно. При этом пользователь может изучить исходный код и изменить его в соответствии со своими потребностями. Использование программного обеспечения MySQL регламентируется лицензией GNU (англ. «General Public License»), а можно приобрести и коммерческую версию СУБД.

SQLite 3.0 – встраиваемый движок баз данных. SQLite (www.sqlite.org) не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а

² SQL – язык структурированных запросов.

предоставляет библиотеку, с которой программа компонуется и движок становится составной частью программы. Этот довольно простой набор API-функций является простым инструментом разработки приложений, использующих в своей работе базы данных.

Такой подход уменьшает накладные расходы, время отклика (очень быстрая система) и упрощает программу. СУБД хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на компьютере, где исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции файл, хранящий базу данных, блокируется. БД хранится в отдельных файлах, которые можно легко перенести на другой сервер. БД может иметь размер до двух терабайт.

Несколько процессов или потоков могут одновременно читать данные из одной базы. Запись в базу можно осуществить только в случае, если никаких других запросов в данный момент не обслуживается. В противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

Наиболее целесообразно использовать её как прикладную программу с небольшими базами данных, и приложение, не требующее администрирования БД, а также как движок БД небольших и средних по объёму и посещаемости интернет сайтов или для обучения языку SQL.