

Лекция. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ.

Учебные вопросы:

1. Понятие базы данных
2. Этапы проектирования БД
3. Реляционная модель данных
4. Основные объекты СУБД MS Access

1. Понятие базы данных

База данных (БД) - это средство накопления и организации больших массивов информации об объектах некоторой предметной области (ПО). БД должна отображать текущие данные о предметной области, накапливать, хранить информацию и предоставлять различным категориям пользователей быстрый доступ к данным. Для этого данные в базе должны быть структурированы в соответствии с некоторой моделью, отражающей основные объекты ПО, их свойства и связи между ними. БД является частью сложной системы, называемой банком данных или системой баз данных (СБД). Эта система (рис. 1) включает в себя собственно БД, программные, технические, языковые и организационно-методические средства, обеспечивающие централизованное накопление и коллективное многоцелевое использование данных.

Одна из компонент СБД - система управления БД (СУБД) представляет собой совокупность языковых и программных средств, с помощью которых БД создается и поддерживается в процессе эксплуатации.

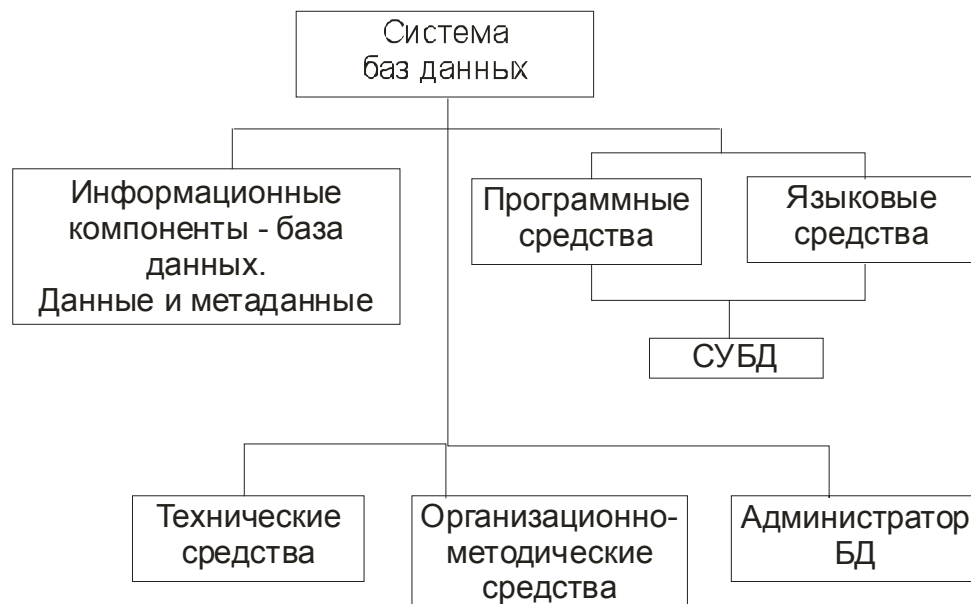


Рис. 1. Компоненты СБД

При проектировании и эксплуатации БД к ней предъявляются следующие требования:

1. Адекватность отображения ПО (полнота, целостность, непротиворечивость, актуальность данных).
2. Возможность взаимодействия пользователей разных категорий; обеспечение высокой эффективности доступа.
3. Дружественность интерфейса.
4. Обеспечение секретности и конфиденциальности.

5. Обеспечение взаимной независимости программ и данных.
6. Обеспечение надежности БД; защита данных от случайного и преднамеренного разрушения; возможность быстрого и полного восстановления данных в случае сбоев в системе.

Лицом, ответственным за создание, эксплуатацию и сопровождение БД является администратор базы данных (АБД). В его обязанности входит выполнение следующих функций:

1. Анализ предметной области, ее описание, формулировка ограничений целостности.
2. Проектирование структуры БД: состава и структуры файлов БД, связей между ними.
3. Задание ограничений целостности при описании структуры БД и процедур обработки данных.
4. Первоначальная загрузка и ведение БД.
5. Защита данных:
 - обеспечение порядка входа-в систему;
 - определение прав доступа пользователей к данным;
 - выбор и создание программно-технических средств защиты данных;
 - тестирование средств защиты данных;
 - сбор статистики об использовании данных;
 - исследование случаев нарушения защиты данных;
 - обеспечение восстановления БД, организация ведения системных журналов.
6. Анализ обращений пользователей к БД.
7. Работа с пользователями.
8. Работа над совершенствованием и динамическим развитием БД.

В жизненном цикле БД одним из наиболее важных этапов является этап проектирования, от результатов которого зависит эффективность дальнейшего использования БД в решении задач предметной области. Главная задача, которая решается в процессе проектирования, - это организация данных: интегрирование, структурирование и определение взаимосвязей. Способ организации данных определяется логической моделью, которая отражает основные сущности ПО и их взаимосвязи. Различные формы представления связей между объектами породили существование различных логических моделей данных, например: иерархическую, сетевую, реляционную. Наибольшую популярность к середине 1980-х годов приобрела реляционная модель в силу ее простоты и математической обоснованности. Как следствие большинство современных СУБД поддерживают эту модель. Поэтому настоящее пособие посвящено проектированию и разработке реляционных баз данных (РБД).

2. Этапы проектирования БД

При проектировании БД организацию данных принято рассматривать на трех уровнях: информационно-логическом (инфологическом), датологическом (концептуальном) и физическом. Этим уровням соответствуют инфологическая, концептуальная и физическая модели предметной области. Весь процесс проектирования может быть разбит на три этапа (рис. 2).

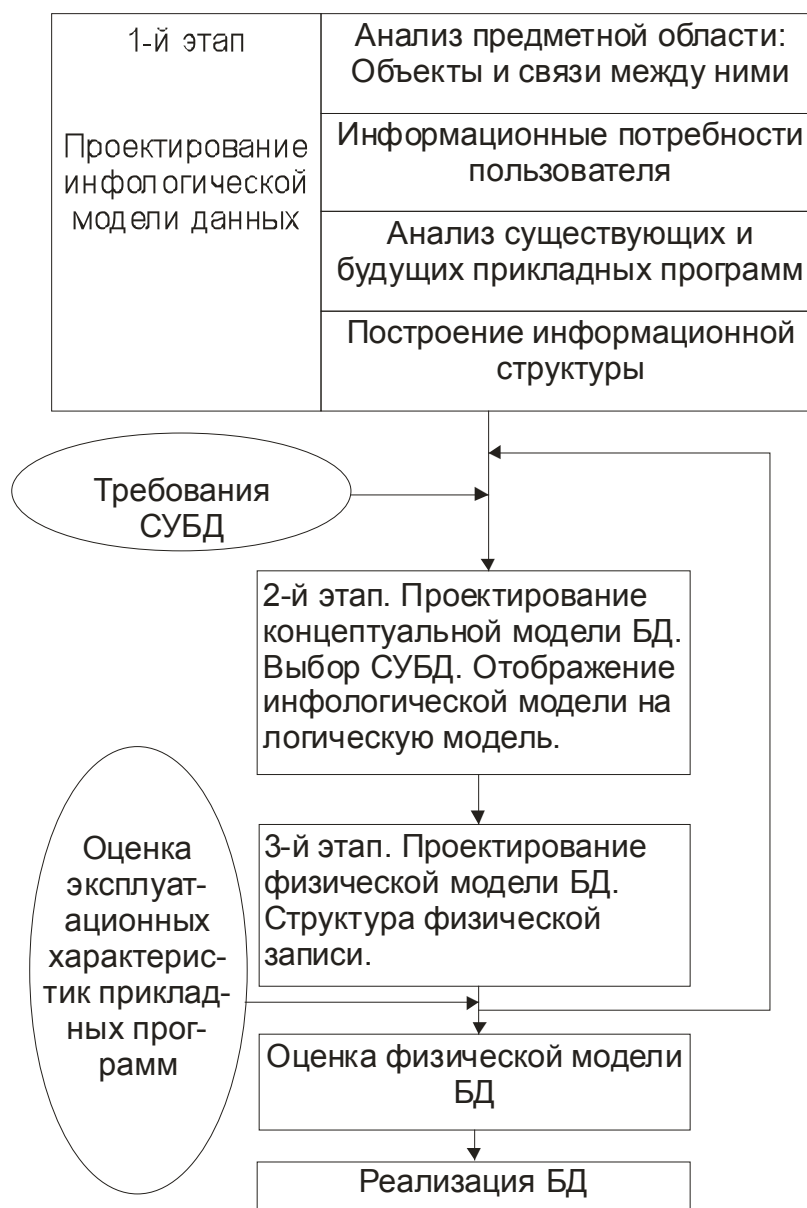


Рис. 2. Этапы проектирования БД

3. Реляционная модель данных

Модель данных - это правила, которые определяют структуру данных, допустимые реализации данных и допустимые операции над данными.

Инфологическая модель описывает предметную область на содержательном уровне. На первом этапе при ее разработке осуществляется анализ предметной области, решаемых задач, запросов пользователей и документов, отражающих события и процессы, протекающие в ПО. Результатом этого анализа являются списки объектов предметной области, перечни их свойств или атрибутов, определение связей между объектами и описание структуры ПО в виде диаграммы. Для каждого из атрибутов указываются ограничения на их возможные значения, определяемые свойствами ПО. Такие ограничения называются ограничениями целостности данных. Инфологическая модель объединяет в единое «обобщенное представление» требования отдельных пользователей и служит средством общения между ними, поэтому разрабатывается без учета особенностей представления данных в памяти ЭВМ.

Концептуальная модель описывает объекты и связи ПО на формальном уровне. Ее разработка ведется на втором этапе и основывается на инфологической модели, полученной на первом этапе. В процессе разработки осуществляется выбор

типа модели данных и определяются ее элементы. Каждая СУБД поддерживает только одну из моделей. Выбор модели данных и выбор СУБД тесно взаимосвязаны.

Внутренняя, или физическая, модель данных определяет способ размещения данных непосредственно на машинном носителе, учитывает распределение данных, методы доступа и способы индексирования. В современных прикладных программных средствах этот уровень организации обеспечивается автоматически без вмешательства пользователя. Пользователь, как правило, оперирует в прикладных программах и универсальных программных средствах представлениями СУБД на организацию данных. Таким образом основная задача проектирования заключается в создании инфологической модели ПО и концептуальной БД.

3.1. Определение РБД

РБД представляет собой совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Отношением называется любая взаимосвязь между объектами и/или их свойствами. Различают взаимосвязи между объектами, между свойствами одного объекта и между свойствами различных объектов. Отношение задается своим именем и списком атрибутов - элементов, связанных этим отношением:

<имя отношения>(<список атрибутов>)

Имя отношения выбирается таким образом, чтобы оно поясняло смысл связи между элементами отношения (семантику отношения).

Для описания некоторого свойства объекта или связи используется простейший неделимый элемент данных, называемый атрибутом.

Атрибут характеризуется именем, типом, значением и другими свойствами.

Имя атрибута - это условное обозначение атрибута в процессах обработки данных. Оно должно быть уникальным в пределах одного и того же отношения.

Значение атрибута - величина, характеризующая некоторое свойство объекта и связи.

Атрибуты соответствуют классам сущностей, объединяемых данным отношением. Список имен атрибутов отношения и их характеристик называют схемой отношения.

Характеристики атрибутов задают область допустимых значений (ОДЗ) для каждого аргумента отношения.

Атрибут или набор атрибутов, которые могут быть использованы для однозначной идентификации конкретного кортежа (конкретного экземпляра отношения), называется первичным ключом отношения или просто ключом.

Математически отношение определяется следующим образом: пусть даны N множеств данных $D_1, D_2, D_3, \dots, D_N$, тогда R есть отношение (связь) между этими множествами, если R - множество упорядоченных кортежей вида $\langle d_1, d_2, \dots, d_N \rangle$, где $d_1 \in D_1, d_2 \in D_2, \dots, d_N \in D_N$. Множества D_1, D_2, \dots, D_N являются множествами возможных значений атрибутов A_1, A_2, \dots, A_N отношения R и называются доменами. Таким образом, отношение является подмножеством декартова произведения одного или более доменов $D_1 \times D_2 \times D_3 \times \dots \times D_N$. Каждый кортеж в отношении уникален.

3.2. Целостность реляционных данных

Логические ограничения, которые накладываются на данные, называются ограничениями целостности. Они формулируются в соответствии со свойствами ПО в форме предикатов, которые для одних множеств данных могут иметь значение истина, для других - ложь. Ограничения используются в моделях данных для поддержания целостности данных при функционировании системы. То есть СУБД

должна контролировать соответствие данных заданным ограничениям при переводе БД из одного состояния в другое. Использование ограничений связано также с адекватностью отражения ПО с помощью данных, хранимых в БД.

Выделяют два основных вида ограничений: внутренние и явные.

Внутренние - это ограничения, свойственные собственно модели данных. Они накладываются на структуру отношений, на связи, на допустимые значения наборов данных, заложенные в выбранной модели данных.

Явные ограничения - это ограничения, задаваемые семантикой ПО. Они описывают области допустимых значений атрибутов, соотношение между атрибутами, динамику их изменения и т. д.

В РМД существует два вида внутренних ограничений целостности:

1. Целостность по существованию - потенциальный ключ отношения не может иметь пустого (NULL) значения. Иными словами, так как потенциальный ключ отношения позволяет из всего множества экземпляров сущности выделить только один, то сущность, не имеющая идентификатора, не существует.
2. Целостность по связи - определяется понятием внешнего ключа (ВК) отношения: подмножество атрибутов отношения R_2 называется внешним ключом для отношения R_1 , если каждому значению ВК отношения R_2 найдется такое же значение первичного ключа в отношении R_1 . Внешний ключ является тем клеем, который обеспечивает связывание отдельных отношений РБД в единое целое. Целостность данных по связи означает систему правил, используемых в СУБД для поддержания связей между записями в связанных таблицах, а также обеспечивает защиту от случайного удаления или изменения связанных данных, от некорректного изменения ключевых полей.

Способы реализации внутренних ограничений целостности зависят от конкретной СУБД. Для СУБД Access они будут рассмотрены после общего знакомства с этой системой.

3.3. Реляционные операции. Операции над отношениями

Основными операциями над отношениями РМД являются 8 операций, входящих в реляционную алгебру Кодда (автора РМД). Реляционная алгебра Кодда включает традиционные операции над множествами: объединение, пересечение, разность (вычитание), декартово произведение - и специальные операции: выбор, проекция, соединение и деление. Совокупность этих операций образует замкнутую алгебру отношений. Замкнутость определяется тем, что аргументами операции реляционной алгебры являются отношения и результатом обработки всегда является новое отношение, которое также может быть аргументом в другой операции (по аналогии с обычной алгеброй чисел).

Рассмотрим основные операции реляционной алгебры.

Объединение - операция выполняется над двумя совместными отношениями R_1 , R_2 (с идентичной структурой - d_1, d_2, \dots, d_n). В результате операции объединения строится новое отношение $R=R_1 \cup R_2$. Отношение R имеет тот же состав атрибутов и совокупность кортежей исходных отношений. Причем в эту совокупность не включаются дубликаты по определению отношения.

Пересечение - операция выполняется над двумя совместными отношениями R_1 , R_2 . Результирующее отношение $R=R_1 \cap R_2$ содержит кортежи, которые есть в каждом из двух исходных. Результат пересечения имеет тот же состав атрибутов, что и в исходных отношениях.

Вычитание - операция выполняется над двумя совместными отношениями R_1 ,

R_2 . В результате операции вычитания строится новое отношение $R=R_1-R_2$ с идентичным набором атрибутов, содержащее только те кортежи первого отношения, R_1 , которые не входят в другое отношение, R_2 .

Декартово произведение выполняется над двумя отношениями R_1 , R_2 , имеющими в общем случае разный состав атрибутов: (d_1, d_2, \dots, d_n) и (p_1, p_2, \dots, p_m) соответственно. В результате операции образуется новое отношение $RD = R_1 \times R_2$, которое включает все атрибуты исходных отношений. Результирующее отношение состоит из всевозможных сочетаний кортежей исходных отношений R_1 и R_2 . Число кортежей декартова (мощность) отношения-произведения равно произведению мощностей исходных отношений.

Выбор - операция выполняется над одним отношением R . Для отношения R по заданному условию (предикату) осуществляется выборка подмножества кортежей. Результирующее отношение имеет ту же структуру, что и исходное отношение, но число его кортежей будет меньше (или равно) числа кортежей исходного отношения

Проекция - операция выполняется над одним отношением R . Операция проекции формирует новое отношение (RPR) с заданным подмножеством атрибутов исходного отношения R . Оно может содержать меньше кортежей, так как после отбрасывания в исходном отношении R части атрибутов (возможного исключения первичного ключа) могут образоваться кортежи-дубли, которые из результирующего отношения исключаются по определению.

Соединение выполняется для заданного условия соединения над двумя логически связанными отношениями. Исходные отношения R_1 и R_2 имеют разные структуры, в которых есть одинаковые атрибуты - внешние ключи (ключи связи). Операция соединения формирует новое отношение, структура которого является совокупностью всех атрибутов исходных отношений. Результирующие кортежи формируются соединением каждого кортежа из R_1 с теми кортежами R_2 , для которых выполняется условие соединения. В зависимости от этого условия соединение называется: естественным - равенство значений общих атрибутов отношений R_1 и R_2 ; эквисоединением - равенство значений атрибутов, входящих в условие соединения; тета-соединением - другой знак сравнения.

Операция соединения имеет большое значение для РБД, так как в процессе нормализации отношений исходное нормализованное отношение разбивается на несколько более мелких отношений, которые при выполнении запросов пользователя требуется, как правило, вновь соединять для восстановления исходного отношения.

Деление - операция выполняется над двумя отношениями R_1 и R_2 , имеющими в общем случае разные структуры и часть одинаковых атрибутов. В результате операции образуется новое отношение, структура которого получается исключением из множества атрибутов отношения R_1 множества атрибутов отношения R_2 . Результирующие строки образуются из тех строк отношения R_1 значения несовпадающих атрибутов которых одинаковые, а значения общих атрибутов образуют отношение, совпадающее с отношением R_2 .

3.4. Преимущества и недостатки реляционных моделей

Реляционные модели имеют ряд достоинств. К ним относятся: простота представления данных благодаря табличной форме, минимальная избыточность данных при нормализации отношений. В реляционных моделях обеспечивается: независимость приложений пользователя от данных, допускающая включение или

удаление отношений, изменение атрибутивного состава отношений. В отличие от иерархических и сетевых, РБД не требуют описания схемы данных и его генерации. К недостаткам реляционной модели можно отнести то, что нормализация данных реляционной модели приводит к значительной фрагментации данных, в то время как в большинстве задач необходимо объединение фрагментированных данных.

4. Основные объекты СУБД MS Access

СУБД MS Access (далее просто СУБД) является 32-разрядной системой управления РБД нового поколения, работающей в средах Windows 98/ME и Windows NT 4.0/2000/XP. В ней предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации. СУБД позволяет создавать приложения, работающие в среде Windows и полностью соответствующие потребностям заказчика.

В состав СУБД входят средства управления таблицами, запросами, формами, отчетами, макросами и модулями как самостоятельными объектами, хранящимися в одном файле БД (расширение .mdb). Благодаря этому создание связанных объектов и проверка целостности данных значительно облегчаются.

Состав файла БД. Архитектура СУБД

СУБД называет объектами все, что может иметь имя. В БД Access основными объектами являются таблицы, запросы, формы, отчеты, макросы и модули.

Таблицы

Таблица - это объект, определяемый для хранения данных. Каждая таблица включает информацию об объекте реального мира, например о клиентах фирмы. Таблица состоит из заголовка и тела. Заголовок включает имена атрибутов объекта (столбцов) и их свойства, например фамилию, телефон и адрес клиента. Тело содержит кортежи (строки), каждая строка представляет множество значений столбцов, в которых хранятся данные о конкретном экземпляре объекта. Например, информация о клиенте:

Семенов; 75-48-09; г. Рязань, ул. Полевая, 20.

Для каждой таблицы можно определить первичный ключ, обеспечивающий уникальность каждой строки, один или несколько индексов, обеспечивающих упорядоченность записей, и другие свойства.

Запросы

Запрос - это объект, который позволяет пользователю получить нужные данные из одной или нескольких базовых таблиц, \ и других запросов. В запросе можно указать условия, которым 1 должны удовлетворять данные. Благодаря этому запрос позволяет из большого массива информации, хранимой в БД, извлекать только нужные данные. Для создания запроса используют запрос по образцу (QBE) или инструкции SQL. Можно создавать запросы на выборку, обновление, удаление или на добавление данных. С помощью запросов можно создавать новые таблицы, используя данные одной или нескольких таблиц, которые уже существуют.

Формы

Форма - это объект, предназначенный в основном для ввода данных, отображения их на экране или управления работой приложения. Формы используются для того, чтобы реализовать требования заказчика к представлению данных из таблиц и запросов. Формы можно распечатать. С помощью формы можно

в ответ на некоторое событие запустить макрос или процедуру, выполняющие определенную обработку данных.

Отчеты

Отчет - это объект, предназначенный для создания документа, который впоследствии может быть распечатан или включен в документ другого приложения. Прежде чем выводить отчет на принтер, его можно просмотреть на экране.

Макросы

Макрос - это объект, представляющий собой структурированное описание одного или нескольких действий, которые будут выполняться в ответ на определенное событие. Например, можно определить макрос, который в ответ на выбор некоторого элемента в основной форме открывает другую форму. С помощью другого макроса можно осуществить проверку значения поля при изменении его содержимого. Можно из одного макроса запустить другой макрос или функцию модуля.

Модули

Модуль - это объект, содержащий программы на Microsoft Access Visual Basic, которые могут разрабатываться пользователем для реализации нестандартных процедур при создании приложения.

Все объекты в СУБД Access могут быть созданы пользователем с помощью конструктора или с помощью различных мастеров. Мастера помогают пользователю в режиме диалога создавать объекты, дают подсказки, предлагают свои решения, что облегчает работу начинающим и непрофессиональным пользователям.

Такие объекты, как формы и отчеты, состоят из графических элементов, называемых элементами управления. Основные элементы управления служат для связи объектов с записями таблиц, являющихся источниками данных.

Каждый объект и элемент управления имеет свои свойства, определяя которые можно настраивать объекты и элементы управления.