MS Access. Реляционные базы данных

Лекции по дисциплине «Основы программирования и информационных технологий», прочитанные на кафедре ФМЭГ ФТ-факультета НТУ «ХПИ»

Повестка дня

- ТЕМА 1. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ
- ТЕМА 2. ТАБЛИЦЫ
- ТЕМА 3. ЗАПРОСЫ
- ТЕМА 4. ФОРМЫ
- ТЕМА 5. ОТЧЕТЫ
- ТЕМА 6.СТРАНИЦЫ ДОСТУПА К ДАННЫМ

Введение. 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.

Базы данных

Для этого данные в базе должны быть структурированы в соответствии с некоторой моделью, отражающей основные объекты предметной области, их свойства и связи между ними.

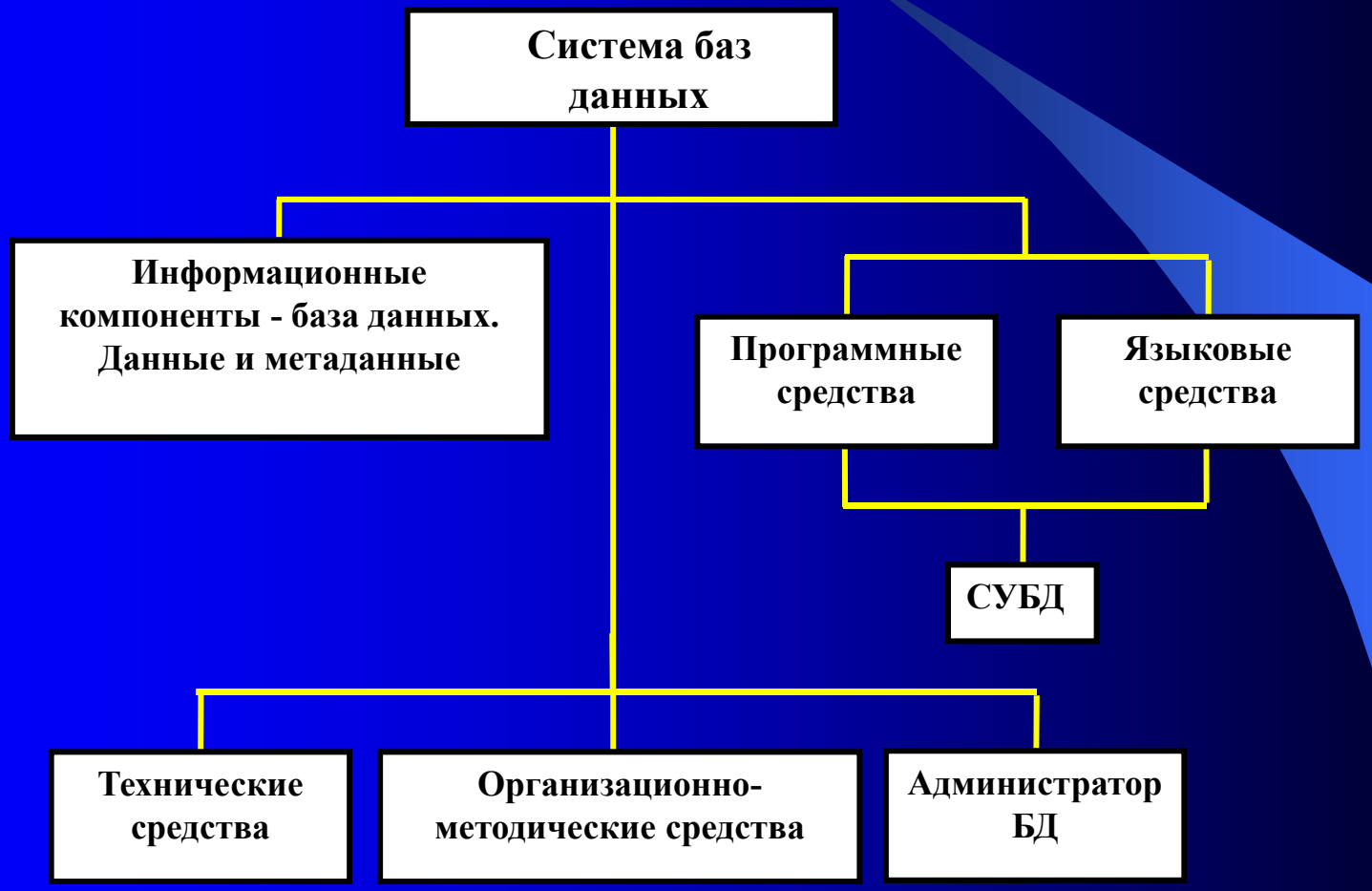
Тема 1. ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

БД является частью сложной системы, называемой банком данных или системой баз данных (СБД). Эта система включает в себя собственно БД, программные, технические, языковые и организационно-методические средства, обеспечивающие централизованное накопление и коллективное многоцелевое использование данных.

ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

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

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;

Требования к БД

3. Дружественность интерфейса.
4. Обеспечение секретности и конфиденциальности.
5. Обеспечение взаимной независимости программ и данных.

Требования к БД

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

Администратор базы данных

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.

Администратор базы данных

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.

Администратор базы данных

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;

Администратор базы данных

5. Защита данных:

- сбор статистики об использовании данных;
- исследование случаев нарушения защиты данных;
- обеспечение восстановления БД, организация ведения системных журналов.

Администратор базы данных

6. Анализ обращений пользователей к БД.
7. Работа с пользователями.
8. Работа над совершенствованием и динамическим развитием БД.

1.1. Разработка баз данных

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:

организация данных:

- интегрирование,
- структурирование,
- определение взаимосвязей.

Способ организации данных определяется логической моделью, которая отражает основные сущности ПО и их взаимосвязи.

Разработка баз данных

Различные формы представления связей между объектами породили существование различных логических моделей данных, например:

- иерархическую,
- сетевую,
- реляционную.

Разработка баз данных

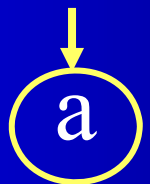
Наибольшую популярность к середине 1980-х годов приобрела реляционная модель в силу ее простоты и математической обоснованности. Как следствие большинство современных СУБД поддерживают эту модель.

1.2. Этапы проектирования БД

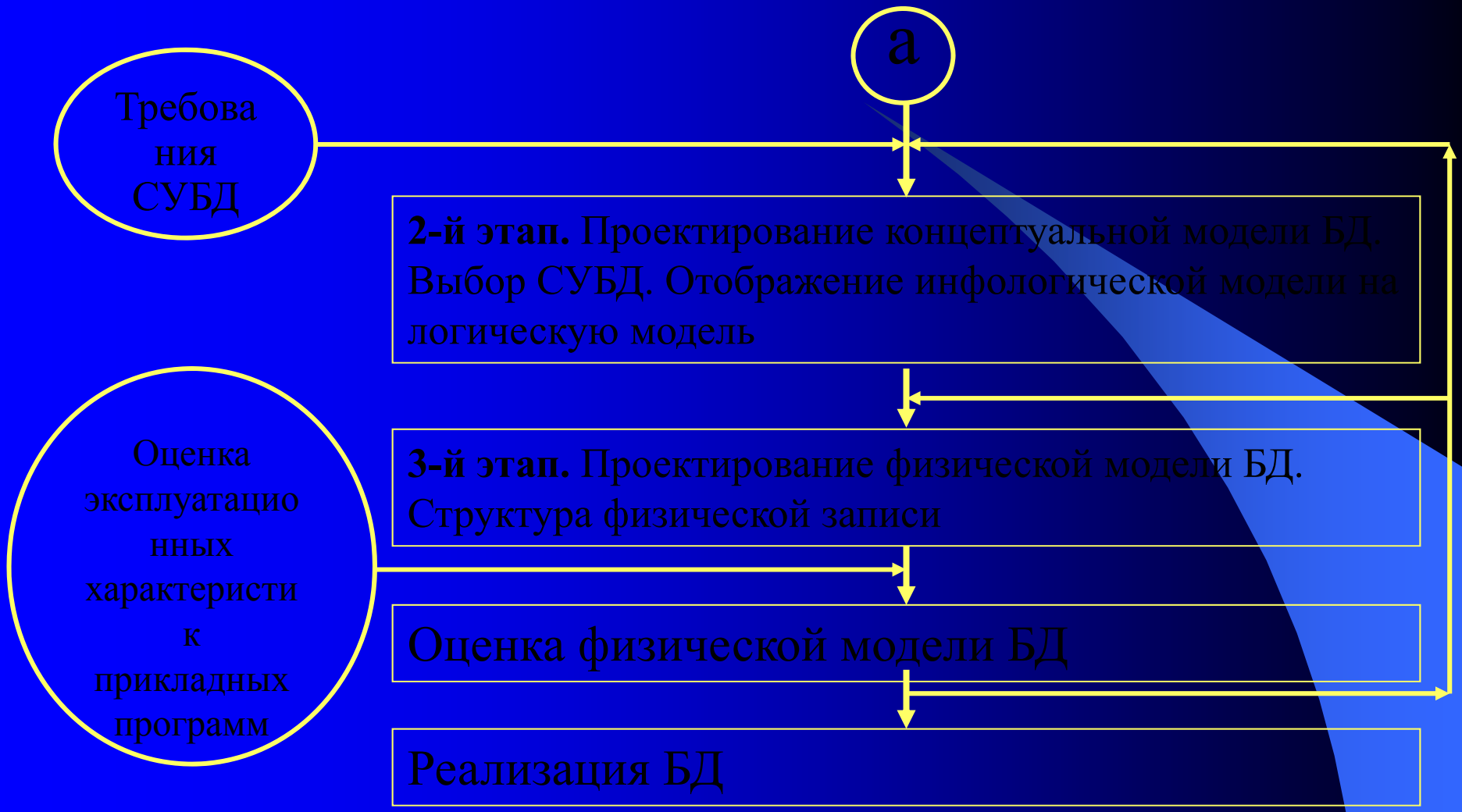
При проектировании БД организацию данных принято рассматривать на трех уровнях: информационно-логическом (мифологическом), даталогическом (концептуальном) и физическом. Этим уровням соответствуют инфологическая, концептуальная и физическая модели предметной области. Весь процесс проектирования может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

1.2. Этапы проектирования БД

1-й этап Проектирование мифологической модели данных	Анализ предметной области: объекты и связи между ними
	Информационные потребности пользователя (анализ запросов)
	Анализ существующих и будущих прикладных программ
	Построение информационной структуры



1.2. Этапы проектирования БД



1.3. 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.

1.3. Реляционная модель данных

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.

1.3. Реляционная модель данных

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.

1.3. Реляционная модель данных

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.

1.3.1. Определение РБД

Реляционная база данных (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.

1.3.1. Определение РБД

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**.

1.3.1. Определение РБД

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.

1.3.1. Определение РБД

Атрибуты соответствуют классам сущностей, объединяемых данным отношением. Список имен атрибутов, отношения и их характеристики 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**.

1.3.1. Определение РБД

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

1.3.1. Определение РБД

Пример 1. БД о поставке деталей может быть описана следующими отношениями:

- Деталь (<номер детали>, <название детали>, <цвет>, <вес>).
- Поставщик (<код поставщика>, <фамилия>, <город>).
- Поставка деталей (<код поставщика>, <номер детали>, <количество>).

Первичные ключи подчеркнуты.

1.3.1. Определение РБД

Ограничения на значения атрибутов могут быть заданы типом данного для соответствующего атрибута.

Имя атрибута	Тип
Код поставщика	Символьный (3)
Фамилия	Символьный (6)
Город	Символьный (10)
Номер детали	Целый
Название детали	Символьный (6)
Цвет	Символьный (6)
Вес	Целый

1.3.1. Определение РБД

Другая форма представления отношения - табличная. Каждому отношению соответствует таблица с таким же именем. Атрибуту в таблице соответствует столбец с именем атрибута, а каждому кортежу отношения - строка таблицы. Строка таблицы часто называется **записью**, а значение атрибута - **полем записи**.

1.3.1. Определение РБД

Описание отношений Базы Данных «Поставка деталей» с помощью табл. 2-4.

Таблица 2. Деталь

Номер детали	Название детали	Цвет	Вес
101	Болт	Черный	3
102	Муфта	Синий	9

Таблица 3. Поставщик

Номер детали	Название детали	Город
П1	Иванов	Ярцево
П2	Алексин	Курск

1.3.1. Определение РБД

Таблица 4. Поставка деталей

Код поставщика	Номер детали	Количество
П1	102	40
П2	101	60

Здесь в отношении *Деталь* атрибутами являются *номер детали, название детали, цвет, вес*. Значениями этих атрибутов впервой строке таблицы будут соответственно <**101, Болт, Черный, 3**>. Каждая строка таблицы описывает конкретный экземпляр сущности *Деталь*.

1.3.1. Определение РБД

Доменами атрибутов будут: для номера детали D_1 - множество целых чисел, для названия детали D_2 - множество символьных строк, названий предметов, для цвета D_3, - множество символьных строк, названий цветов, для веса D_4 - множество целых чисел. В отношении два кортежа каждый состоит из четырех упорядоченных элементов. Для отношения Поставка деталей первичный ключ является составным <код поставщика, номер детали>.

1.3.1. Определение РБД

Таким образом, в ходе разработки РБД должен быть определен состав взаимосвязанных реляционных таблиц и определен состав атрибутов каждого отношения с указанием ограничений на их допустимые значения. Состав атрибутов должен отвечать требованиям нормализации. Нормализация отношений производится на этапе концептуального проектирования БД.

1.3.1. Определение РБД

Существует несколько **нормальных форм (НФ)** реляционной модели данных (РМД), которые позволяют исключить избыточное дублирование данных, обеспечить целостность и непротиворечивость данных.

При *первой нормальной форме* (1НФ) все атрибуты отношения должны быть простыми (атомарными, неделимыми) с точки зрения СУБД.

1.3.1. Определение РБД

НФ более высокого порядка основаны на понятии **функциональной зависимости**, которая может быть определена следующим образом.

Пусть A и B - подмножества атрибутов отношения. Говорят, что B функционально зависит от A (A—>B), если для каждого значения A существует ровно одно связанное значение B. **Функциональную зависимость** можно выявить, исходя из базовых свойств самих атрибутов путем анализа.

1.3.1. Определение РБД

При **второй нормальной форме (2НФ)** должна обеспечиваться **1НФ** и каждый неключевой атрибут функционально полно зависит от ключа.

Полная **функциональная зависимость** от ключа означает, что если ключ составной, то любой неключевой атрибут зависит от всего ключа и не зависит ни от какой его части.

1.3.1. Определение РБД

При **третьей нормальной форме (3НФ)** отношение должно находиться **2НФ**, а также каждый неключевой атрибут нетранзитивно зависит от ключа, проще говоря: отсутствуют **функциональные зависимости** между неключевыми атрибутами.

2. Основные объекты СУБД MS Access

Система управления БД **MS Access** (далее просто **СУБД**) является 32-разрядной системой управления реляционными базами данных 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.

2. 1. Состав файла БД. Архитектура СУБД.

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.

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

2.1.1. 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.

2.1.2. 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 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.

2.1.3. 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.

2.1.4. 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.

2.1.5. 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.

2.1.6. Модули

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

2.1.8. Объекты MS Access

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.

Источники дополнительных сведений

Проектирование баз данных. СУБД Microsoft Access: Учебное пособие для вузов / Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров., А. Н. Пылькин, Н. И. Цуканова. - М.: Горячая линия-Телеком, 2004, 240с.