

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию в своей рабочей тетради, законспектируйте основные тезисы.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: v.vika2014@mail.ru тел. 0721744922

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

ЛЕКЦИЯ

БАЗЫ ДАННЫХ И ИХ ВИДЫ. ОСНОВНЫЕ ПОНЯТИЯ

СОЗДАНИЕ И ВЕДЕНИЕ РАЗЛИЧНЫХ ЭЛЕКТРОННЫХ ДОКУМЕНТОВ

Явления реального мира зачастую могут быть описаны с помощью структурных взаимосвязей между совокупностями фактов. Для представления информации подобного рода в явлениях может быть использована структурная модель данных.

База данных (БД) — это поименованная совокупность структурированных данных о конкретных объектах реального мира в какой-либо предметной области.

База данных (БД) - это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

Структурирование данных предполагает существование (или установление) между ними каких-то отношений (связей). В зависимости от характера этих отношений можно выделить несколько *классификационных признаков структур данных*.

Классификация структур данных по различным признакам

1. По отношению порядка структуры данных делятся на упорядоченные и неупорядоченные.

В *упорядоченных структурах* элементы размещаются по порядку в соответствии со значением некоторого признака. Наиболее простым признаком является порядковый номер элемента; установление порядка в соответствии с номером называется *нумерацией*. При этом если весь набор имеет один общий

идентификатор (например, М), то отдельным данным присваиваются собственные идентификаторы — индексы (например, М5 или Мб). Чаще всего индекс задается целым числом, хотя это необязательно (в качестве индекса может выступать любой знак из конечного алфавита). Лексикографический порядок индексов определяет отношение следования между элементами структуры, т. е. элемент Мб следует за элементом М5, а элемент Ма располагается перед элементом Мб. Примером структур, в которых упорядочение производится по номеру элемента, являются *массивы*. Порядковый номер элемента можно считать внешним признаком, который может присваиваться элементу независимо от

его значения. Например, регистрационный номер документа определяется только временем его поступления в учреждение, а не его содержанием. Помимо нумерации, в структурах данных используется упорядочение по значению некоторого внутреннего признака (например, размещение фамилий в алфавитном порядке или группы предприятий — в порядке убывания их рентабельности); такое упорядочение называется *ранжированием*.

Примером *неупорядоченных структур* являются *множества* — в них не определен порядок элементов; единственное, что можно установить для каких-то конкретных данных, так это их принадлежность (или непринадлежность) выбранному множеству.

2. По характеру отношений между элементами структуры данных подразделяют на линейные и нелинейные.

В *линейных структурах* все элементы равноправны. К ним относят *массив, множество, стек, очередь*.

В *нелинейных структурах* между элементами существуют отношения подчиненности или они могут быть связаны логическими условиями. К ним относят *деревья, графы, фреймы*.

1. По однородности структуры данных делят на однородные и неоднородные. К *однородным* относят структуры, содержащие элементарные данные только одного типа. Примерами однородных структур являются *массивы, множества, стеки*. Неоднородные структуры объединяют данные разных типов. К *неоднородным* структурам относят *записи*.

2. По технологии обработки данных БД подразделяют на централизованные и распределенные.

Централизованная БД хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Такой способ использования баз данных часто применяют в локальных сетях ПК.

Распределенная БД состоит из нескольких (возможно, пересекающихся или даже дублирующих друг друга) частей, хранимых в различных ЭВМ вычислительной -сети.

Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

5. По способу доступа к данным различают БД с локальным и удаленным (сетевым) доступом. Системы централизованных БД с сетевым доступом предполагают различные архитектуры подобных систем, такие, как файл-сервер и клиент-сервер.

Файл-сервер. Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной БД. Файлы БД в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производят обработку. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать на рабочих » станциях локальные БД — для своих нужд. Концепция «файл-сервер» условно отображена на рис. 1.

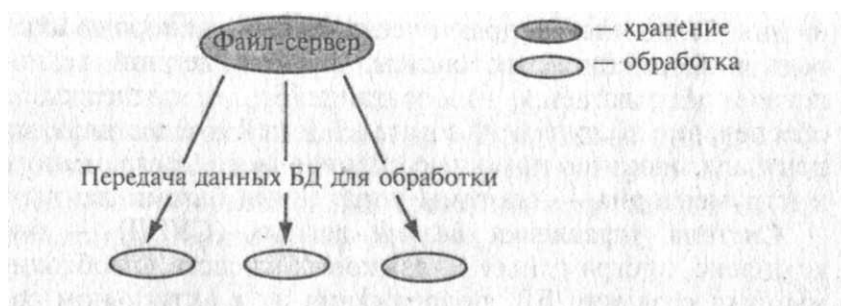


Рис. 1. Схема обработки информации в БД по принципу «файл-сервер»

Клиент-сервер. В соответствии с этой концепцией подразумевается, что, помимо хранения централизованной БД, центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиент

том (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные (но не файлы) транспортируются по сети от сервера к клиенту. Спецификой архитектуры «клиент-сервер» является использование *языка запросов SQL*. Концепция «клиент-сервер» условно изображена на рис. 2.



Рис. 2. Схема обработки информации в БД по принципу «клиент-сервер»

База данных, банк данных, система управления базой данных, администратор базы данных

БД (в том или ином виде) — неотъемлемая часть решения большинства практических задач, в первую очередь информационных систем. В современной технологии предполагается, что создание БД, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — системы управления базами данных.

Операции в БД:

- ◆ ввод основных и дополнительных данных;
- изменение и корректировка уже имеющейся информации;
- создание новых структур хранения данных;
- организация взаимодействия с другими БД и прикладными программами;

- длительное хранение данных;
- ◆ передача информации (межкомпьютерный обмен);
- ◆ защита от несанкционированного доступа;

Средства защиты данных можно разделить на две группы: 1. различные способы запрещения доступа к данным; 2. средства скрытия источников данных;

- ◆ резервное копирование данных;
 - сортировка информации (по алфавиту) обеспечивает универсальность организации операций над данными и удобный пользовательский интерфейс;
 - просмотр данных;
- ◆ автоматическое обновление данных, т.е. достаточно один раз внести изменение, и оно автоматически будет отражено везде, где это необходимо.

Система управления базами данных (СУБД) — это комплекс программных и языковых средств, необходимых для создания БД, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

СУБД обладает следующими рабочими характеристиками:

- **полнота.** Чем больше информации, тем более вероятность нахождения полезных данных;
- **структурность.** Чем лучше структурирована информация, тем она полезнее и ценнее;
- **удобство.** Достигается выполнением первых трех рабочих характеристик. Для работы с СУБД необходимо несколько типов языков:
 - язык для написания программы;
 - язык манипулирования данными, обеспечивающий интерфейс между программой и системой управления БД;
 - язык описания данных.

Централизованный характер управления данными в БД предполагает необходимость существования некоторого лица (группы лиц), на которое

возложены функции администрирования данными, хранимыми в базе. Системы управления базами данных выполняют две основные функции:

- хранение и ведение представления структурной информации (данных);
- преобразование по некоторому запросу хранимого представления в структурную информацию.

Процедура хранения информации в ПК состоит в том, чтобы сформировать и поддерживать структуру хранения данных в памяти компьютера.

Процедура актуализации данных позволяет изменить значения данных, записанных в базе, либо дополнить определенный раздел, группу данных. Устаревшие данные могут быть удалены с помощью соответствующей операции.

Процедура извлечения данных необходима для пересылки из БД необходимых сведений для преобразования, отображения или передачи по вычислительной сети.

Современные структуры хранения данных должны быть независимыми от программ, использующих эти данные, и реализовывать принципы полноты и минимальной избыточности. Такие структуры и получили название БД. Процедуры создания структуры хранения (БД), актуализации, извлечения и удаления данных осуществляют с помощью специальных программ, называемых **системами управления базами данных**.

Хранение и передача данных тесно связаны между собой, для выполнения этих процедур используют сетевые информационные технологии. Программы, предназначенные только для хранения и передачи данных, носят название **информационных хранилищ** и представляют собой компьютеризованные архивы.

БД и СУБД являются частью *Банков данных*, которые хранят сведения из самых различных областей человеческой деятельности: это библиотечное и банковское дело, образование и медицина, управление предприятием и государством, право, экология, транспорт, туризм и многое другое.

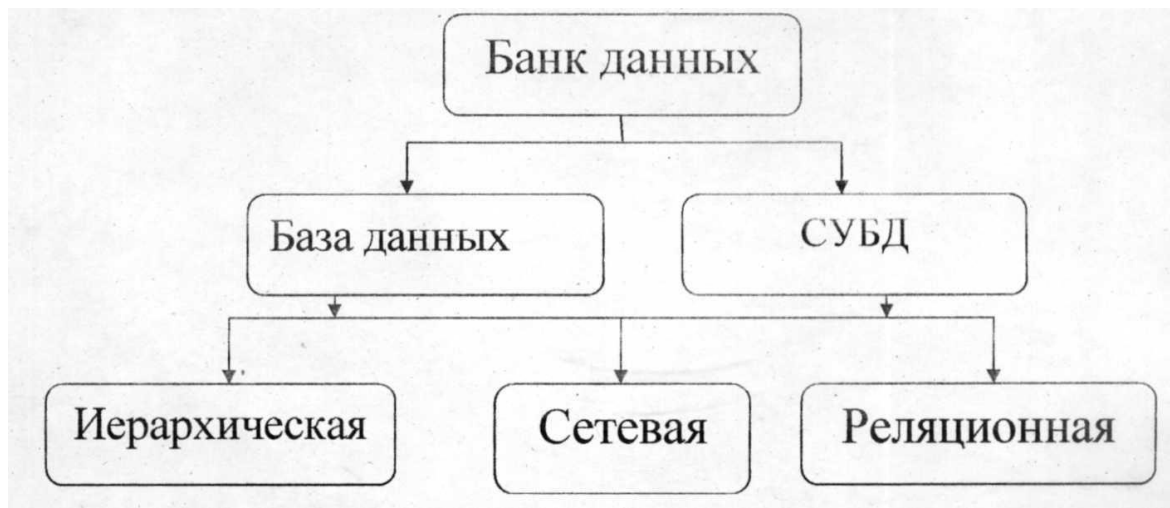


Рис.3. Структура банка данных

Уровни представления данных

Ядром любой БД является модель данных - множество структур данных, ограничений целостности и операций манипулирования данными. **Модель данных** — это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.



Рис. П. Пример иерархической структуры БД

СУБД основывается на использовании иерархической сетевой или реляционной моделей, на комбинации этих моделей или на некотором их подмножестве.

Иерархическая модель данных. Иерархическая структура — это совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево), вид которого представлен на рис. 10.

Уровень 1 . А

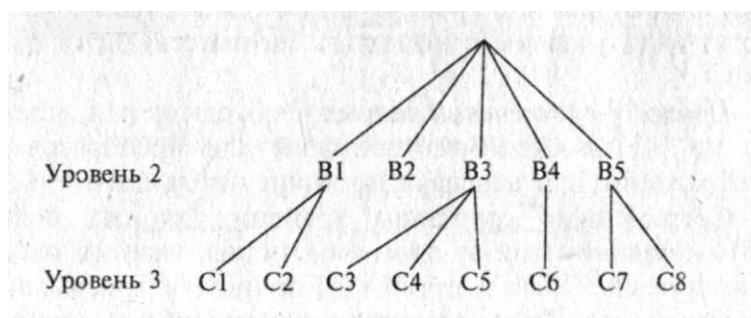


Рис. 10. Графическое изображение иерархической структуры БД

К основным понятиям иерархической структуры относятся: уровень, узел (элемент), связь.

Узел — это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел *и* более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т. д. уровнях. Количество деревьев в БД определяется числом корневых записей.

К каждой записи БД существует только один (иерархический) путь от корневой записи. Например, как видно из рис. 10, для записи С4 путь проходит через записи А и В3.

Из приведенного на рис. 11 примера очевидно, что иерархическая структура правомерна, так как каждый студент учится в определенной (только одной) группе, которая относится к определенному (только одному) институту.

Сетевая модель данных. В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

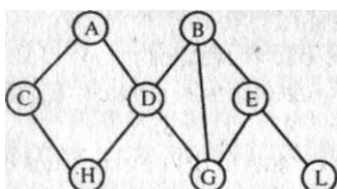


Рис. П. Пример иерархической структуры БД

П

Примером сложной сетевой структуры может служить структура БД, содержащая сведения о работах студентов учебных заведений. Возможно участие одного студента в нескольких разработках, а также участие нескольких студентов в разработке одной темы.

Графическое изображение примера, состоящего только из двух типов записей, показано на рис. 12 и рис. 13. Единственное отношение представляет собой сложную связь между записями в обоих направлениях.



Реляционная модель данных.

Понятие «реляционный» (от англ. *relation* — отношение) связано с разработками известного американского специалиста в области систем баз данных **Рис. 12.** Графическое изображение сетевой структуры данных, удобным для

пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Студент (номер зачетной книжки, фамилия, группа)

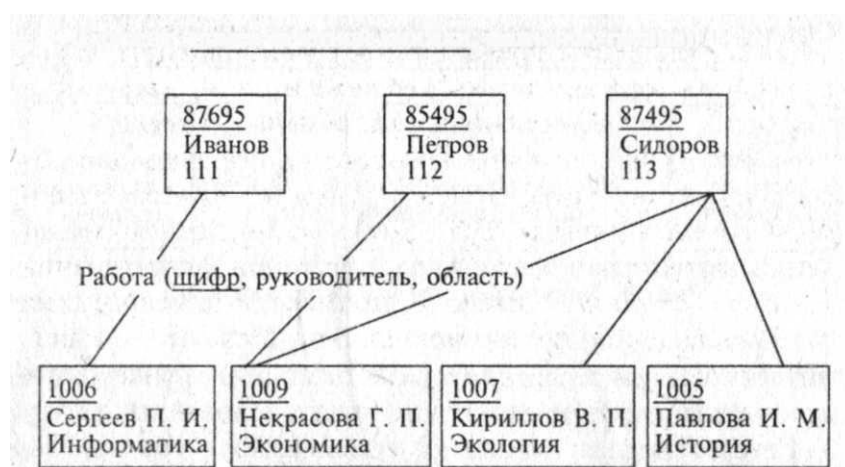


Рис. 13. Пример сетевой структуры БД

Реляционная модель ориентирована на организацию данных в виде двухмерных таблиц. Каждая **реляционная таблица** представляет собой двухмерный массив и обладает следующими свойствами:

— каждый элемент таблицы — один элемент данных;

— все столбцы в таблице однородны, т. е. все элементы в столбце имеют одинаковые тип (числовой, символьный и т. д.) и длину;

- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;

— порядок следования строк и столбцов может быть произвольным.

В качестве примера можно представить информацию о студентах вуза в виде *реляционной таблицы*:

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16 493	Сергеев	Петр	Михайлович	01.01.76	111
16 593	Петрова	Анна	Владимировна	15.03.75	112
16 693	Анохин	Андрей	Борисович	14.04.76	111

Здесь отношения представлены в виде таблицы, строки которой соответствуют *кортежам*, или *записям*, а столбцы — атрибутам отношений, доменам, *полям*.

Поле, каждое значение которого однозначно определяет соответствующую запись, называется *простым ключом* (ключевым полем). Если записи однозначно определяются значениями нескольких полей, то такая таблица БД имеет *составной ключ*. В примере, показанном на рис. 14, ключевым полем таблицы является «Номер личного дела».



Рис. 14. Пример реляционной модели

Чтобы связать две реляционные таблицы, необходимо ключ первой таблицы ввести в состав ключа второй таблицы (возможно совпадение ключей), в противном случае

нужно ввести в структуру первой таблицы *внешний ключ* — ключ второй таблицы. На рис. 14 показана реляционная модель, построенная на основе отношений «Студент, Сессия, Стипендия», где:

- Студент — Номер, Фамилия, Имя, Отчество, Пол, Дата рождения, Группа;
- Сессия — Номер, Оценка 1, Оценка 2, Оценка 3, Оценка 4, Результат;
- Стипендия — Результат, Процент.

Таблицы СТУДЕНТ и СЕССИЯ имеют совпадающие ключи (*Номер*), что дает возможность легко организовать связь между ними. Таблица СЕССИЯ имеет

первичный ключ *Номер* и содержит внешний ключ *Результат*, который обеспечивает ее связь с таблицей СТИПЕНДИЯ.

База данных — средство организации хранения и управления большим количеством упорядоченной разнородной информации.

Базу данных можно представить в виде таблицы с конечным числом столбцов и неопределённым числом строк. Примером базы данных может служить классный журнал или итоговая ведомость, куда заносятся оценки за четверть. Количество столбцов ограничено — их столько, сколько предметов изучается, а количество учащихся может изменяться. Пустая ведомость — это база данных, из которой удалены все записи, но остаются названия полей базы, то есть её структура. При создании базы данных необходимо сначала разработать её структуру, а затем заполнить её информацией — эти функции и выполняются с помощью СУБД. Вторым этапом является ввод и редактирование записей в таблицу. БД считается созданной, даже если она пустая.

Столбцы в базе данных называют **полями**, а **строки** — **записями**. Каждое поле имеет своё **имя** и содержит отдельный элемент информации. Для каждого поля необходимо указывать его **имя**, **тип данных**, **размер**. От типа и размера поля зависит скорость доступа к БД и объём файла. **Тип данных** поля определяется значениями, которые предполагается вводить в поле.

Основные объекты окна БД имеют следующее назначение:

- **таблица** — основное средство для хранения информации в БД;
- **запрос** — это инструмент для извлечения необходимой информации из исходных таблиц и представления ее в удобной форме;
- **форма** — это основное средство для ввода данных, управления СУБД и вывода результатов на экран монитора;
- **отчёт** — это специальное средство для формирования выходных документов и вывода их на принтер;
- **макросы** в Access представляют собой совокупность внутренних команд, предназначенных для автоматизации работы с БД;
- **модули** являются программами, создаваемыми средствами языка VBA, и похожи на макросы в Word и Excel.

Работа с СУБД Access.

СУБД Access создаёт и обрабатывает реляционные базы данных, то есть она позволяет хранить данные не в одной, а в нескольких таблицах и устанавливать связь между ними. Такие таблицы называются связанными, т.е. объединёнными в единую базу. Для задания связи таблицы должны иметь поля с одинаковым типом данных. Связь между таблицами устанавливает отношение между совпадающими значениями в этих полях. Такая организация позволяет уменьшить избыточность хранимых данных, упрощает их ввод, удаление, поиск.

Для установления связи между таблицами необходимо, чтобы:

- связываемые поля имели одинаковый тип данных. Исключение составляет поле-счётчик, так как поле-счётчик может быть связано с числовым полем, имеющим размер «длинное целое».
- таблицы хранились внутри одной БД.
- главная таблица связывалась с подчинённой по ключу.

Для обеспечения целостности данных для связанных таблиц нужно помнить, что:

- в подчиненную таблицу не может быть добавлена запись с несуществующим в главной таблице ключом связи;
- в главной таблице нельзя удалить запись, если не удалены связанные с ней записи в подчиненной таблице;
- изменение значений ключа связи главной таблицы должно приводить к изменению соответствующих значений в записях подчиненной таблицы.

Та таблица, от которой идет связь, называется главной, а таблица, к которой эта связь ведёт, — подчинённой.

В каждой таблице должно быть уникальное поле, с помощью которого можно связать таблицы между собой. Такое поле называется полем первичного ключа, или первичным ключом.

Для того чтобы понять принципы разработки реляционных баз данных, требуется дать определения различных типов реляционных ключей и таблиц:

- **Базовая таблица.** В реляционной базе данных базовой таблицей называется таблица, которая включает один или несколько столбцов свойств объекта и содержит **первичный ключ**, который однозначно определяет этот объект.

Более того, базовая таблица должна содержать первичный ключ. Базовые таблицы часто называют **первичными**, поскольку они имеют **первичный ключ**.

- **Промежуточная таблица.** Таблица, не являющаяся базовой (т. к. она не объединяет свойства объекта или не содержит поле первичного ключа), которая используется для обеспечения связей между другими таблицами, называется **таблицей отношений**. Ключевые поля в таблицах отношений должны быть **внешними ключами**, связанными с первичными ключами базовой таблицы. То есть, таблица отношений состоит только из внешних ключей и не содержит независимых элементов данных.

- **Первичный ключ.** Первичный ключ состоит из набора значений, которые однозначно определяют запись базовой таблицы. Любому значению первичного ключа должна соответствовать одна и только одна строка таблицы. Первичный ключ включает одно поле только в том случае, если это поле не содержит повторяющихся значений.

- **Составные ключи.** Если для выполнения условий, накладываемых на значения первичного ключа, заданный ключ включает несколько полей таблицы, то тогда он называется **составным**.

- **Внешние ключи.** Внешний ключ — это столбец, значения которого соответствуют значениям первичного ключа другой связанной таблицы.

В программе предусмотрены пять возможностей создания таблицы:

- **Импорт таблиц** из другой базы. В зависимости от обстоятельств из импортируемой таблицы может поступить структура полей, их названия и свойства, а также и содержимое базы. Необходимые правки вносят вручную.

- Режим **Связь с таблицами** применяется в тех случаях, когда речь идет о чужой таблице, которая находится на удаленном сервере и которую нельзя импортировать целиком. Это напоминает подключение к таблице для совместного использования её данных.

- **Мастер таблиц.** Это программа, ускоряющая создание структуры таблицы. Мастер задает ряд вопросов и, руководствуясь полученными ответами, создает структуру таблицы автоматически.

- **Режим таблицы** открывает заготовку, в которой все поля имеют формальные имена: Поле1, Поле2... и т. д. — и один стандартный текстовый тип. Таковую таблицу можно сразу наполнять информацией.

- **Конструктор** предоставляет возможность одновременно задавать поля будущей таблицы и устанавливать свойства этих полей.

Мощным средством обработки данных, хранимых в таблицах Access, являются запросы. С их помощью можно просматривать, анализировать и изменять данные из нескольких таблиц, а также можно использовать запросы в качестве источника данных для форм и отчётов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять вычисления над группами записей.

В Access можно создавать следующие типы запросов:

Запрос на выборку. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или из нескольких таблиц и выводятся в определённом порядке. Простые запросы на выборку практически не отличаются от фильтров, которые можно сохранять как запросы. Этот запрос можно использовать, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчёта и других действий.

Запрос с параметрами. Это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, на основе которых будет выполняться запрос.

Перекрытый запрос. Перекрытые запросы предназначены для группирования данных и представления их в компактном виде. Позволяют представить большой объем данных в виде, удобном для восприятия,

анализа, сравнения. Могут использоваться в качестве базового при создании отчёта. Отчёты позволяют выбирать из базы данных нужную информацию, оформить её в виде документа и распечатать. Источником данных может быть таблица, запрос или несколько взаимосвязанных таблиц. Отчеты и формы похожи, разница в том, что, в отличие от форм, отчёты не предназначены для ввода и корректировки данных. Отчёты состоят из разделов, подобных разделам форм. В процессе конструирования отчета формируется состав и содержимое разделов отчёта, размещение в нём значений, выводимых из полей связанных таблиц баз

данных, формируются заголовки, размещаются вычисляемые поля. Отчёт позволяет сгруппировать данные по нескольким уровням, для каждого из которых производится вычисление итогов, определяются заголовки и примечания.

Система управления базами данных Access. Объекты базы данных и их основные элементы»

Система управления базами данных (СУБД) MS Access является системой управления реляционными БД, работающей в среде Windows. 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. 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.

Основными компонентами (объектами) базы данных являются:

- таблицы,
- запросы,
- формы,
- отчеты
- макросы и модули.

Таблица – фундаментальная структура системы управления реляционными базами данных. В Microsoft Access таблица – это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). Каждое поле содержит отдельную часть записи, например фамилию, должность или инвентарный номер. Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу, например о сотрудниках или заказах.

Запрос – вопрос о данных, хранящихся в таблицах, или инструкция на отбор записей, подлежащих изменению. Ниже перечне-

– лены типы запросов, которые могут быть созданы с помощью Microsoft Access:

– запрос-выборка задает вопрос о данных, хранящихся в таблицах, и представляет полученный динамический набор в режиме формы или таблицы без изменения данных. Изменения, внесенные в динамический набор, отражаются в базовых таблицах;

– запрос-изменение изменяет или перемещает данные. К этому типу относятся запрос на добавление записей, запрос на удаление записей, запрос на создание таблицы или запрос на обновление;

– перекрестные запросы предназначены для группирования данных и представления их в компактном виде;

– запрос с параметрами позволяет определить одно или несколько условий отбора во время выполнения запроса;

– запросы SQL – это запросы, которые могут быть созданы только с помощью инструкций SQL в режиме SQL: запрос-объединение, запрос к серверу и управляющий запрос. Язык SQL (Structured Query Language) – это язык запросов, который часто используется при анализе, обновлении и обработке реляционных баз данных (например, Microsoft Access).

Форма – это объект Microsoft Access, в котором можно разместить элементы управления, предназначенные для ввода, изображения и изменения данных в полях таблиц.

Отчет – это объект Microsoft Access, который позволяет представлять определенную пользователем информацию в определенном виде, просматривать и распечатывать ее.

Макрос – одна или несколько макрокоманд, которые можно использовать для автоматизации конкретной задачи. Макрокоманда – основной строительный блок макроса; самостоятельная инструкция, которая может быть объединена с другими макрокомандами, чтобы автоматизировать выполнение задачи.

Модуль – набор описаний, инструкций и процедур, сохраненных под одним именем. В Microsoft Access имеется три типа модулей: модуль формы, отчета и

общий модуль. Мо-дули форм и отчетов содержат локальную программу для форм или отчетов. Если проце-дуры общего модуля явным образом не объявлены личными для модуля, в котором они появляются, они распозна-ются и могут вызываться процедурами из других модулей этой базы данных.

База данных может содержать несколько модулей, в том числе общие модули, модули форм и модули отчетов.

2. Типы данных СУБД Access

Для каждого поля таблиц базы данных необходимо указать тип данных. Тип данных оп-ределяет вид и диапазон допустимых значе-ний, которые могут быть введены в поле, а также объем памяти, вы-деляющийся для этого поля. Перечень типов данных полей и опи-са-ние значений, сохраняемых в таких полях, приведены в следующей таблице.

Тип данных

Содержимое

Текстовый

Текст и числа, например, имена и адреса, номера телефонов и почто-вые индексы. Текстовое поле может содержать до 255 сим-волов

Поле Мемо

Длинный текст и числа, например комментарии и пояснения. Мемо-поле может содержать до 64 000 символов

Числовой

Общий тип для числовых данных, допускающих проведение ма-тематических расчетов, за исключением расчетов для денежных зна-чений. Свойство Размер поля (FieldSize) позволяет указать различные типы числовых данных

Дата/время

Значения даты и времени. Пользователь имеет возможность вы-брать один из многочисленных стандартных форматов или со-здать специ-альный формат

Денежный

Денежные значения. Не рекомендуется использовать для прове-дения денежных расчетов значения, принадлежащие к числово-му типу дан-ных, так как

последние могут округляться при расчетах. Значения типа «Денежный» всегда выводятся с указанным числом десятичных знаков после запятой

Счетчик

Автоматически вставляющиеся последовательные номера. Нумерация начинается с 1. Поле счетчика удобно для создания ключа. Это поле является совместимым с полем числового типа, для которого в свойстве Размер поля (FieldSize) указано значение «Длинное целое»

Логический

Значения «Да»/«Нет», «Истина»/«Ложь», «Вкл»/«Выкл», т. е. одно из двух возможных значений

Поле объекта OLE

Объекты, созданные в других программах, поддерживающих протокол OLE. Объекты связываются или внедряются в базу данных Microsoft Access через элемент управления в форме или отчете

3. Создание новой базы данных

Создание новой базы данных Access осуществляется в соответствии с ее структурой, полученной в результате немашинного проектирования, которое заключается в создании информационно-логической модели предметной области.

Структура реляционной базы данных определяется составом таблиц и их взаимосвязями. Создание реляционной базы данных с помощью СУБД Access на компьютере начинается с формирования структуры таблиц. При этом формируется состав полей и задается их описание.

После формирования структуры таблиц создается схема данных, в которой устанавливаются связи между таблицами. Access запоминает и использует эти связи при заполнении таблиц и обработке данных. Завершается создание базы данных процедурой заполнения таблиц конкретной информацией.

После запуска MS Access одновременно с его окном открывается первое диалоговое окно, позволяющее начать создание базы данных или открыть существующую базу данных.

На закладках окна базы данных представлены основные типы объектов базы данных: таблицы, запросы, формы, отчеты, макросы, модули.

В составе окна базы данных находятся три управляющие кнопки:

– первая кнопка выполняет три функции:

Открыть, если выбрана таблица, форма или запрос; Просмотреть, если выбран отчет; Запустить, если выбран макрос;

– кнопка Конструктор позволяет перейти в режим доработки любого ранее созданного объекта;

– кнопка Создать позволяет приступить к созданию нового объекта любого выбранного типа.

При создании объекта предоставляется возможность выбора режима его разработки. Это может быть мастер или конструктор, или некоторый другой режим, зависящий от выбранного типа объекта.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. *Дайте определение термина «база данных».*
2. *Дайте определения понятий «данные» и «знания».*
3. *Дайте определения понятий «банк данных», «система управления базой данных», «администратор базы данных».*
4. *Назовите способы организации связи между данными.*
5. *Определите область задач, решаемых с помощью баз данных*