БАЗЫ ДАННЫХ

ДАННЫЕ

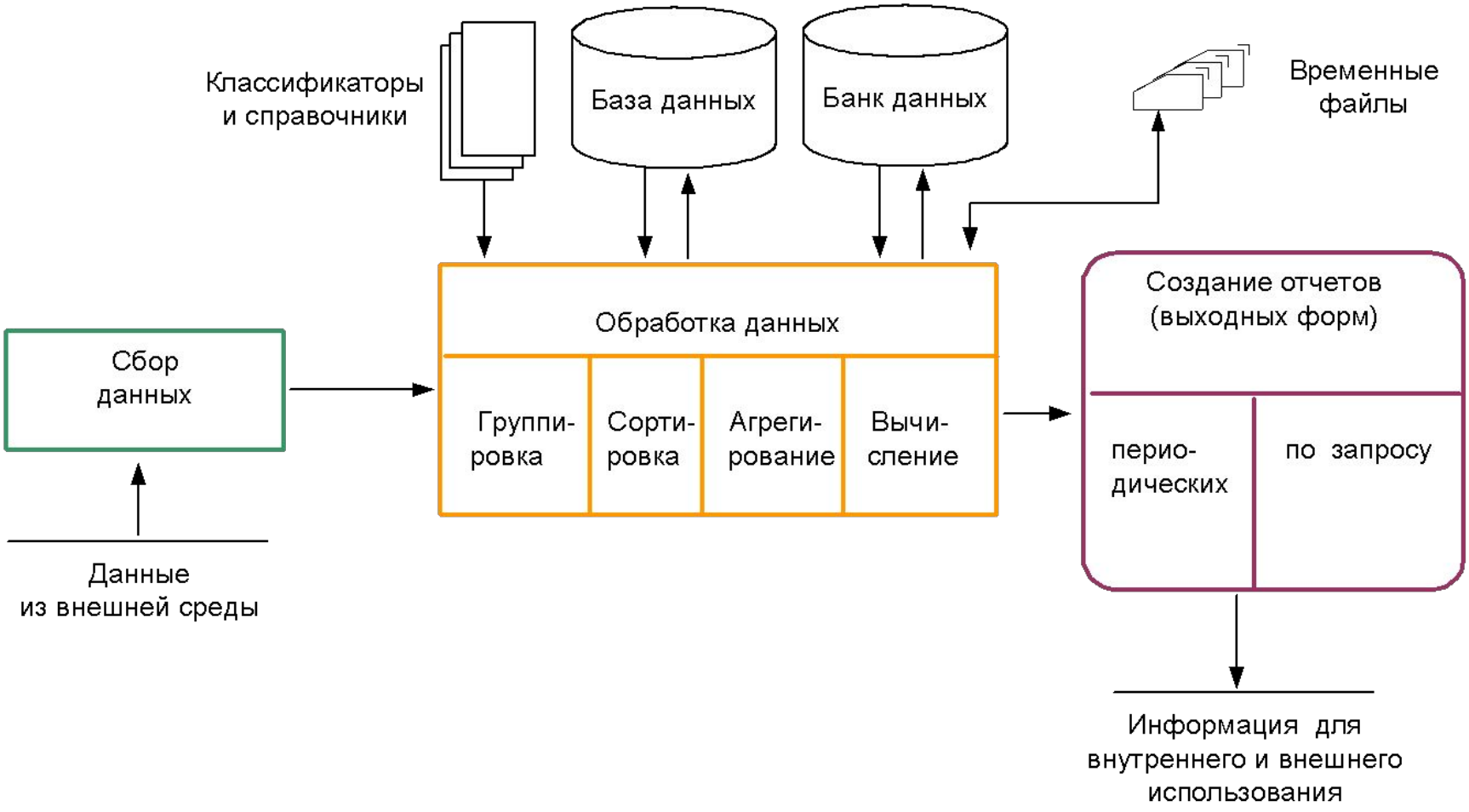
Любая программа имеет дело с некоторой внешней по отношению к ее коду информацией, задающей какие-либо параметры или режим ее работы. Такую информацию также называют данными программы.

Очевидно, что в зависимости от типа решаемых задач проблемы организации работы с данными будут качественно различными.

При решении хозяйственных, экономических и финансовых задач приходится иметь дело с обширными и взаимозависимыми массивами данных. Такие сложные наборы данных традиционно принято называть базами данных.

Концепция баз данных возникла в начале 50-х годов двадцатого столетия.

ОСНОВНЫЕ КОМПОНЕНТЫ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ ОБРАБОТКИ ДАННЫХ



ПРИ ОБРАБОТКЕ НА ЭВМ ДАННЫЕ ТРАНСФОРМИРУЮТСЯ, УСЛОВНО ПРОХОДЯ СЛЕДУЮЩИЕ ЭТАПЫ:

данные как результат измерений и наблюдении;

данные на материальных носителях информации (таблицы, протоколы, справочники);

модели (структуры) данных в виде диаграмм, графиков, функции;

данные в компьютере на языке описания данных;

базы данных на машинных носителях.

ПОНЯТИЕ БАЗЫ ДАННЫХ

Цель любой информационной системы — обработка данных об объектах реального мира.

База данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления и в конечном счете автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков.

Сделать это возможно, только если данные структурированы, т.е. специальным образом организованы.

СТРУКТУРИРОВАНИЕ - ЭТО ВВЕДЕНИЕ СОГЛАШЕНИЙ О СПОСОБАХ ПРЕДСТАВЛЕНИЯ ДАННЫХ

Объект

Студент

Личное дело № 16493, Сергеев Петр Михайлович, дата рождения 1 января 1992 г., Л/д № 16593, Петрова Анна Владимировна, дата рожд. 15 марта 1990 г., № личн. дела 16693, д.р. 14. 04.91. Анохин Андрей Борисович.

Ключ

Атрибуты

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.92
16393	Петрова	Анна	Владимировна	15.03.90
16693	Анохин	Андрей	Борисович	14.04.91

Запись

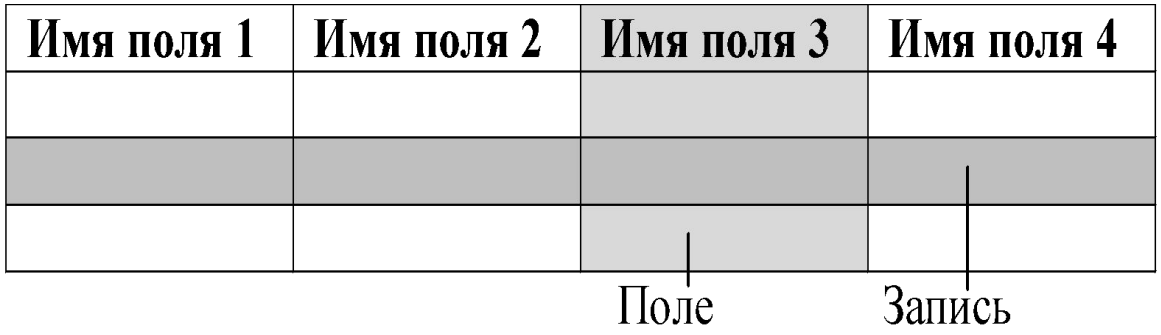
СТРУКТУРНЫЕ ЭЛЕМЕНТЫ БАЗЫ ДАННЫХ

поле — элементарная единица логической организации данных, которая соответствует неделимой единице информации — реквизиту. Для описания поля используются следующие *характеристики:*

- ***имя*, например. Фамилия, Имя, Отчество, Дата рождения;**
- ***тип*, например, символьный, числовой, календарный;**
- ***длина*, например, 15 байт, причем будет определяться максимально возможным количеством символов;**
- ***точность* для числовых данных, например два десятичных знака для отображения дробной части числа.**

запись — совокупность логически связанных полей. Экземпляр записи — отдельная реализация записи, содержащая конкретные значения ее полей ;

таблица — совокупность экземпляров записей одной структуры .



БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ БД (СУБД)

База данных (БД) - это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

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

КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

По *технологии обработки* данных базы данных подразделяются на:

централизованные

распределенные.

КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

По *способу доступа к* данным базы данных разделяются:

на базы данных с *локальным доступом* и

базы данных с *удаленным* (*сетевым) доступом.*

КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

Системы централизованных баз данных с сетевым доступом предполагают различные *архитектуры* подобных систем:

файл-сервер;

клиент-сервер.

МОДЕЛИ ДАННЫХ

С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

Модель данных определяется следующими компонентами:

- **структура данных;**
- **множеством допустимых операций.**

Другими словами модель данных это совокупность структур данных и операций их обработки.

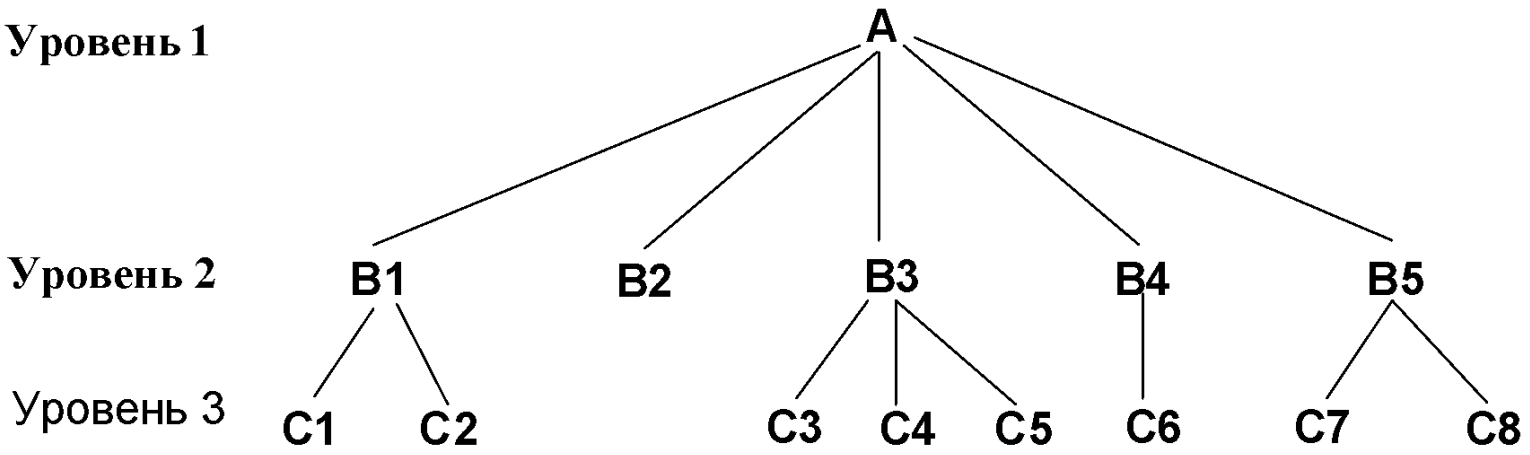
МОДЕЛИ ДАННЫХ

В современных СУБД используются следующие модели либо их комбинации:

- **иерархическая;**
- **сетевая;**
- **реляционная.**

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

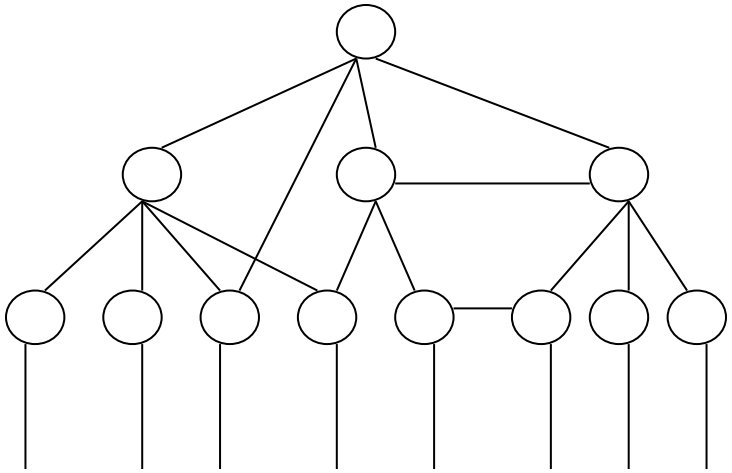
Иерархичеекая структура данных представляет совокупность элементов, связанных между собой по правилам подчиненности или иерархии при этом любой элемент может подчиняться только одному какому-нибудь другому элементу. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево).



СЕТЕВАЯ МОДЕЛЬ ДАННЫХ

Концепция сетевой модели данных связана с именем Ч. Бахмана.

Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков

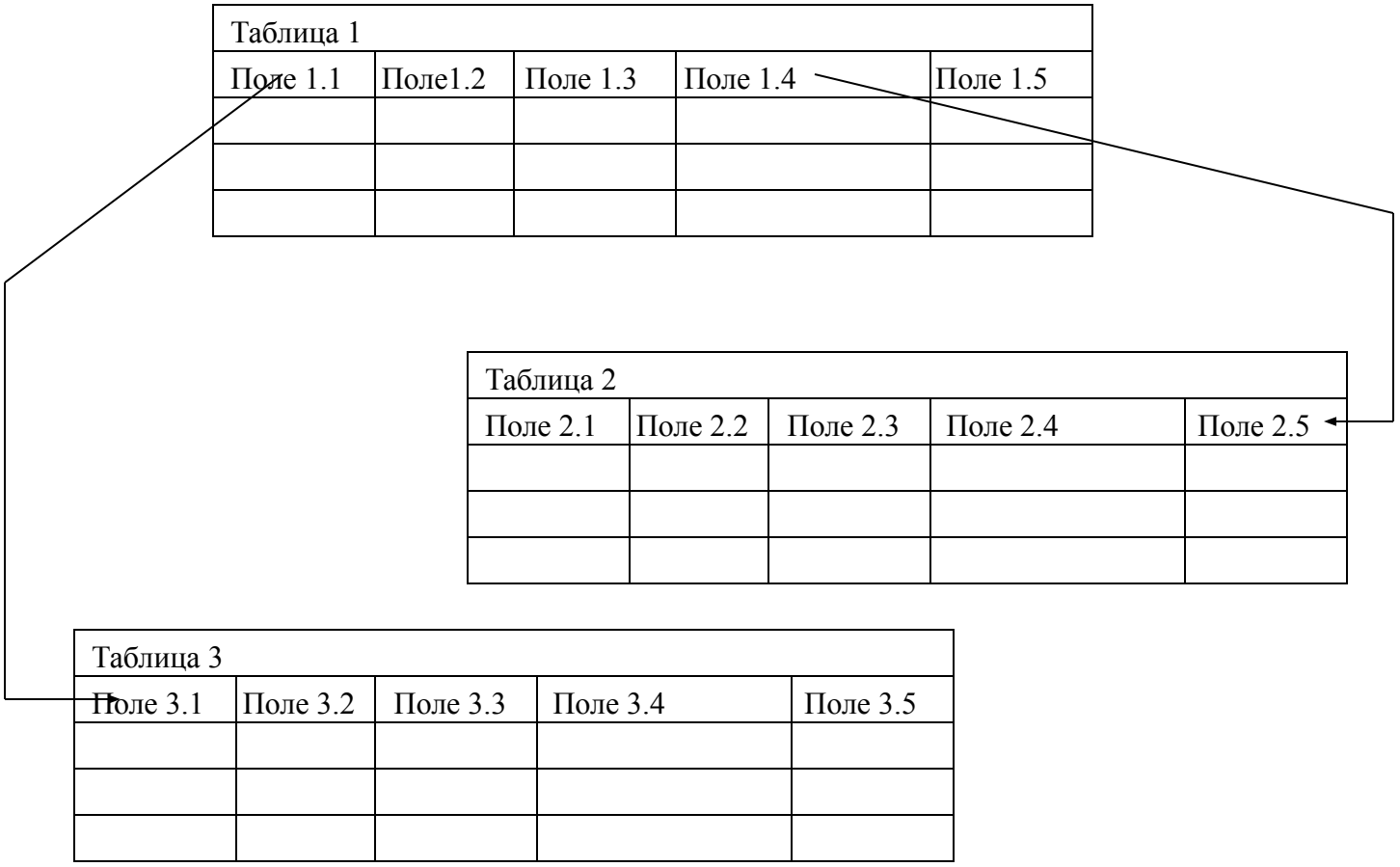


РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Понятие реляционный (англ. relation — отношение) связано с разработками известного американского специалиста в области систем баз данных Е. Кодда. Эти модели характеризуются простотой структуры данных, удобным для пользователя табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ



РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая *реляционная таблица* представляет собой двумерный массив.

Свойства реляционной БД:

- может состоять из нескольких таблиц;
- каждая запись состоит из набора полей (столбцов);
- информация в каждом поле (столбце) в таблице однородная, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и размер;
- в таблице может быть ключевое поле (информация в данном поле уникальна);
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.92
16393	Петрова	Анна	Владимировна	15.03.90
16693	Анохин	Андрей	Борисович	14.04.91

НОРМАЛИЗАЦИЯ ОТНОШЕНИЙ

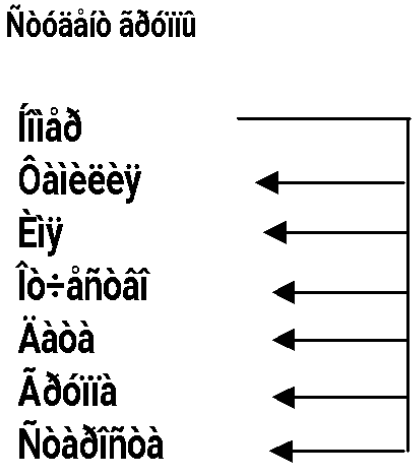
Одни и те же данные могут группироваться в таблицы различными способами, т.е. возможна организация различных наборов отношений взаимосвязанных информационных объектов. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование данных и упрощающей процедуры их обработки и обновления.

Е.Коддом выделены три нормальные формы отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей (самой совершенной) нормальной форме.

ПЕРВАЯ НОРМАЛЬНАЯ ФОРМА

Отношение называется нормализованным или приведенным к *первой нормальной форме*, если все его атрибуты простые (далее неделимы). Преобразование отношения к первой нормальной форме может привести к увеличению количества реквизитов (полей) отношения и изменению ключа.

Например, отношение Студент = *(Номер*, Фамилия, Имя, Отчество, Дата, Группа и др.) находится в первой нормальной форме.



ВТОРАЯ НОРМАЛЬНАЯ ФОРМА

Описательные реквизиты информационного объекта логически связаны с общим для них ключом, эта связь носит характер функциональной зависимости реквизитов.

Отношение будет находиться во второй нормальной форме, если оно находится в первой нормальной форме, и каждый неключевой атрибут функционально полно зависит от составного ключа.

Отношение Студент = *(Номер,* Фамилия, Имя, Отчество, Дата, Группа) находится в первой и во второй нормальной форме одновременно, так как описательные реквизиты однозначно определены и функционально зависят от ключа Номер.

Отношение Успеваемость = *(Номер,* Фамилия, Имя, Отчество, *Дисциплина,* Оценка) находится в первой нормальной форме и имеет составной ключ Номер+Дисциплина. Это отношение не находится во второй нормальной форме, так как атрибуты Фамилия, Имя, Отчество не находятся в полной функциональной зависимости с составным ключом отношения.

ТРЕТЬЯ НОРМАЛЬНАЯ ФОРМА

Отношение будет находиться в третьей нормальной форме, если оно находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если в состав описательных реквизитов информационного объекта Студент включить фамилию старосты группы (Староста), которая определяется только номером группы, то одна и та же фамилия старосты будет многократно повторяться в разных экземплярах данного информационного объекта. В этом случае наблюдаются затруднения в корректировке фамилии старосты в случае назначения нового старосты, а также неоправданный расход памяти для хранения дублированной информации.

ПРИМЕР «РАСЩЕПЛЕНИЯ» СТРУКТУРЫ ИНФОРМАЦИОННОГО ОБЪЕКТА

Для устранения транзитивной зависимости описательных реквизитов необходимо провести "расщепление" исходного информационного объекта. В результате расщепления часть реквизитов удаляется из исходного информационного объекта и включается в состав других (возможно, вновь созданных) информационных объектов.

ТИПЫ СВЯЗЕЙ

Все информационные объекты предметной области связаны между собой. Различаются *связи* нескольких типов, для которых введены следующие обозначения:

один к одному (1 : 1);

один ко многим (1 : M);

многие ко многим (M : M).

ЯЗЫК SQL

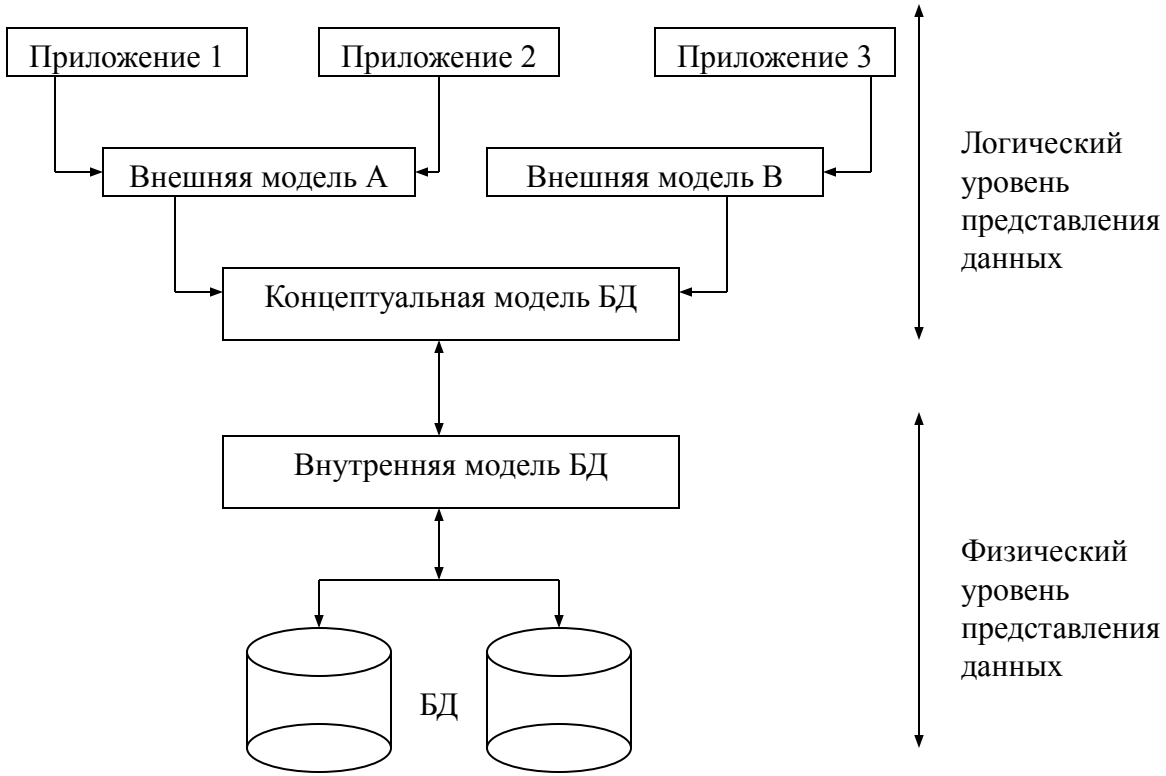
В разработанной Коддом реляционной модели были определены как требования к организации таблиц, содержащих данные, так и язык, позволяющий работать-с ними. Впоследствии этот язык получил название *SQL (Structured Query Language* — структурированный язык запросов).

SQL был впервые реализован фирмой IBM в начале 70-х годов двадцатого века под названием Structures English Query Language (SEQUEL). В дальнейшем SQL стал стандартом *de facto* языка работы с реляционными базами данных. Этот его статус был впервые зафиксирован в 1986 году Американским национальным институтом стандартов (ANSI). Другими достаточно известными стандартами SQL стали стандарты ANSI SQL-92, ISOSQL-92. В составе SQL могут быть выделены следующие группы инструкций:

- язык описания данных — DDL (Data Definition Language);
- язык манипулирования данными — DML (Data Manipulation Language);
- язык управления транзакциями.

АРХИТЕКТУРА СУБД

Базы данных и программные средства их создания и ведения (СУБД) имеют многоуровневую архитектуру



АРХИТЕКТУРА СУБД

Различают концептуальный, внутренний и внешний уровни представления

данных баз данных, которым соответствуют модели аналогичного назначения.

Концептуальная модель **состоит из множества экземпляров различных типов данных, структурированных в соответствии с требованиями СУБД к логической структуре базы данных.**

Внешняя модель **является подмножеством концептуальной модели. Возможно пересечение внешних моделей по данным. Частная логическая структура данных для отдельного приложения (задачи) или пользователя соответствует внешней модели или подсхеме БД. С помощью внешних моделей поддерживается санкционированный доступ к данным БД приложений (ограничен состав и структура данных концептуальной одели БД, доступных в приложении, а также заданы допустимые режимы обработки этих данных: ввод, редактирование, удаление, поиск).**

Внутренний уровень отображает требуемую организацию данных в среде хранения и соответствует физическому аспекту представления данных. ***Внутренняя модель*** **состоит из отдельных экземпляров записей, физически хранимых во внешних носителях.**

СУБД MS Access и ее основные возможности

ОБЩАЯ ХАРАКТЕРИСТИКА СУБД MS ACCESS

Microsoft Access в настоящее время является одной из самых популярных среди настольных (персональных) программных систем управления базами данных.

MS Access имеет:

- **высокую степень универсальности и продуманности интерфейса, который рассчитан на работу с пользователями самой различной квалификации.**
- **глубоко развитые возможности интеграции с другими программными продуктами, входящими в состав Microsoft Office, а также с любыми программными продуктами, поддерживающими технологию OLE;**
- **богатый набор визуальных средств разработки.**

ОСНОВНЫЕ ЭТАПЫ РАЗРАБОТКИ БАЗЫ ДАННЫХ В СРЕДЕ MS ACCESS

- **разработка и описание структур таблиц данных;**
- **разработка схемы данных и задание системы взаимосвязей между таблицами;**
- **разработка системы запросов к таблицам базы данных и (при необходимости) . их интеграция в схему данных;**
- **разработка экранных форм ввода/вывода данных;**
- **разработка системы отчетов по данным;**
- **разработка программных расширений для базы данных, решающих специфические задачи по обработке содержащейся в ней информации, с помощью инструментария макросов и модулей;**
- **разработка системы защиты данных, прав и ограничений по доступу.**

ТИПЫ ДАННЫХ

Текстовый

- Текст или комбинация текста и чисел, например, адреса, а также числа, не требующие вычислений, например, номера телефонов, инвентарные номера или почтовые индексы.
- Сохраняет до 255 знаков. Свойство **Размер поля (FieldSize)** определяет максимальное количество знаков, которые можно ввести в поле.

Поле MEMO

- Длинный текст или числа, например, примечания или описания.
- Сохраняет до 65 536 знаков.

Числовой

- Данные, используемые для математических вычислений, за исключением финансовых расчетов (для них следует использовать тип «Денежный»).
- Сохраняет 1, 2, 4 или 8 байтов; 16 байтов для кодов репликации (GUID). Конкретный тип числового поля определяется значением свойства **Размер поля (FieldSize)**.

Дата/время

- Значения дат и времени.
- Сохраняет 8 байтов.

Денежный

- Используется для денежных значений и для предотвращения округления во время вычислений.
- Сохраняет 8 байтов.

ТИПЫ ДАННЫХ

Счетчик

- Автоматическая вставка уникальных последовательных (увеличивающихся на 1) или случайных чисел при добавлении записи.
- Сохраняет 4 байта;

Логический

- Данные, принимающие только одно из двух возможных значений, таких как «Да/Нет», «Истина/Ложь», «Вкл/Выкл». Значения Null не допускаются.
- Сохраняет 1 бит.

Поле объекта OLE

- Объекты OLE (такие как документы Microsoft Word, электронные таблицы Microsoft Excel, рисунки, звукозапись или другие данные в двоичном формате), созданные в других программах, использующих протокол OLE.
- Сохраняет до 1 Гигабайта (ограничивается объемом диска).

Гиперссылка

- ГиперссылкиГиперссылки. Гиперссылка может иметь вид пути UNCГиперссылки. Гиперссылка может иметь вид пути UNC либо адреса URL.
- Сохраняет до 64 000 знаков.

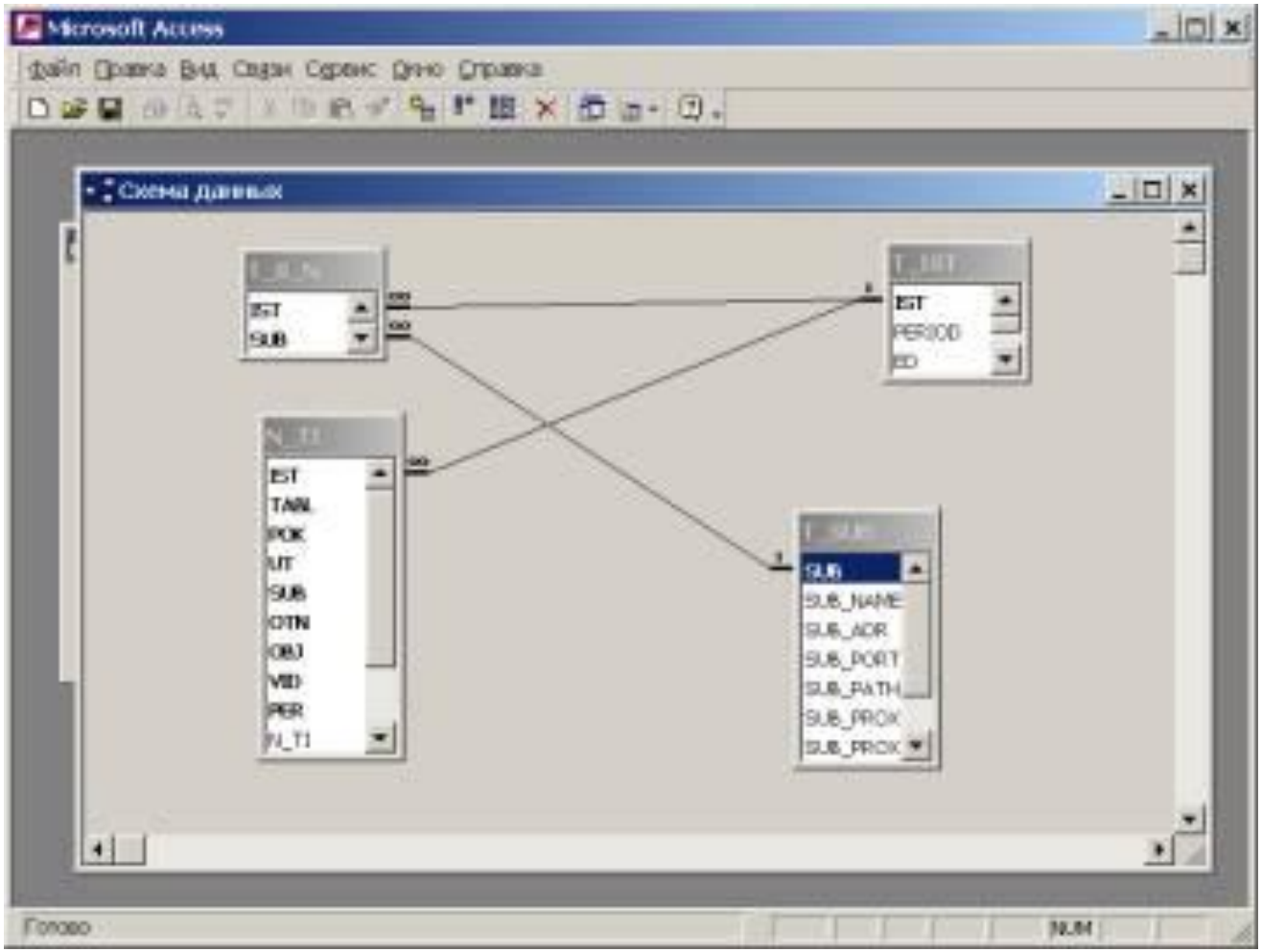
Мастер подстановок

- Создает поле, позволяющее выбрать значение из другой таблицы или из списка значений, используя поле со списком. При выборе данного параметра в списке типов данных запускается мастер для автоматического определения этого поля.
- Для сохранения требуется тот же размер, что и у первичного ключаДля сохранения требуется тот же размер, что и у первичного ключа, соответствующего полю подстановок, — обычно 4 байта.

СОЗДАНИЕ СХЕМЫ ДАННЫХ

Механизм описания логических связей между таблицами в Access реализован в виде объекта, называемого Схемой данных. Перейти к ее созданию можно из панели инструментов База данных через меню Работа с базами данных > Схема данных, который будет иметь схема данных для построенных на предыдущих шагах таблиц

СОЗДАНИЕ СХЕМЫ ДАННЫХ



СОЗДАНИЕ СХЕМЫ ДАННЫХ

Интерфейс задания связей между полями в схеме основан на «перетаскивании» (перемещении при нажатой левой кнопки мыши) выбранного поля и «наложении» его на то поле, с которым должна быть установлена связь. Для связывания сразу нескольких полей их следует перемещать при нажатой клавише Ctrl.

Важнейшей задачей, которую позволяет решать схема, является обеспечение *логической целостности данных в* базе.

РАЗРАБОТКА ЗАПРОСОВ К БАЗЕ ДАННЫХ

К простейшим задачам обработки могут быть отнесены:

- поиск записи по условию (см. функцию меню Правка > Найти);
- сортировка записей в требуемом порядке (см. функцию меню Записи > Сортировка);
- получение выборки записей таблицы, удовлетворяющей заданному условию, или, как еще говорят, задание фильтра для таблицы (Записи > Фильтр).

В результате выполнения запросов отобранные данные представляются в виде таблиц, к которым также могут быть созданы новые запросы

РАЗРАБОТКА ЗАПРОСОВ К БАЗЕ ДАННЫХ

Понятие *запроса* в Access употребляется в расширительном плане. Его следует трактовать как некоторую команду на выбор, просмотр, изменение, создание или удаление данных.

Наиболее распространенным типом запросов является *запрос на выборку*. Данный тип, собственно говоря, и устанавливается по умолчанию для вновь создаваемого запроса.

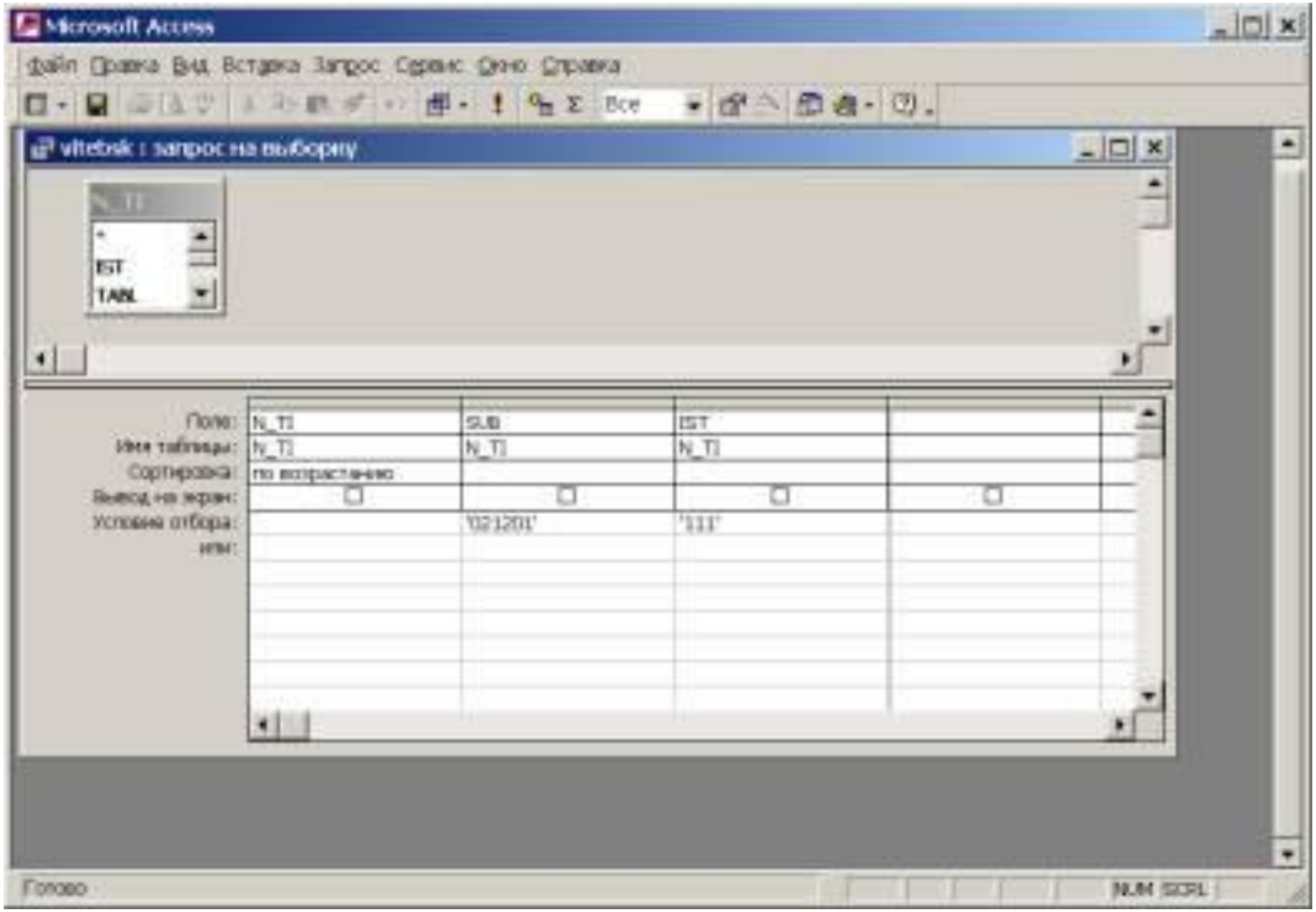
При работе с системой данных очень часто возникает задача соединения данных из различных связанных таблиц в одну. (*Запрос на объединение*)

СОЗДАНИЕ ЗАПРОСОВ К БАЗЕ ДАННЫХ

В процессе формирования запроса можно выделить ряд принципиальных этапов:

- описание структуры запроса (то есть указание того, какая информация должна выводиться в колонках таблицы запроса);
- задание порядка, в котором данные должны выводиться при выполнении запроса;
- задание условий вывода записей в запросе

КОНСТРУКТОР ЗАПРОСОВ



ИСПОЛЬЗОВАНИЕ ПОДСТАНОВОЧНЫХ ЗНАКОВ

Подстановочные знаки используются в качестве прототипов для других знаков при указании образца поиска в следующих случаях.

Известна только часть значения.

Требуется найти значения, начинающиеся с конкретной буквы или соответствующие определенному шаблону.

Эти же знаки можно использовать в окнах Поиск и Замена в базе данных Access.

ИСПОЛЬЗОВАНИЕ ПОДСТАНОВОЧНЫХ ЗНАКОВ

***** Соответствует любому количеству знаков. Может использоваться в качестве первого или последнего знака текстовой строки. wh* — поиск слов what, white и why.

? Соответствует любому текстовому знаку.

д?м — поиск слов дом, дым.

[ ] Соответствует любому одному знаку из заключенных в скобки.

д[ор]м — поиск слова дом, но не дым.

! Соответствует любому одному знаку, кроме заключенных в скобки. д[!о]м — поиск слова дым, но не дом.

-Соответствует любому знаку из диапазона. Необходимо указывать этот диапазон по возрастанию (от а до я (A до Z), но не от я до а (Z до A)).

-Д[о-ы]м — поиск слов дом, дым.

Соответствует любой цифре.

1#3 — поиск значений 103, 113, 123.

ОПЕРАТОРЫ СРАВНЕНИЯ С ОБРАЗЦОМ

Операторы Access упрощают создание выражений для выборки записей в запросах и относятся к операторам сравнения с образцом. Эти операторы возвращают True или False, в зависимости от соответствия значения в поле выбранной спецификации оператора. Наличие этих операторов в условиях на значение позволяет либо включать запись в запрос, если логическое значение, возвращаемое выражением, равно True, либо отвергать, если это значение — False.

ОПЕРАТОРЫ СРАВНЕНИЯ С ОБРАЗЦОМ

Оператор Between определяет, находится ли числовое значение в определенном диапазоне значений

Between (-100) And (100)

Оператор Is при использовании вместе с Null определяет, является ли значение Null или Not Null

Is Null Is Not Null

Оператор In ределяет, является ли строковое значение элементом списка значений

In ("Москва", "Киев", "Санкт-Петербург")

ОПЕРАТОРЫ СРАВНЕНИЯ С ОБРАЗЦОМ

Оператор Like можно использовать для поиска значений в полях, соответствующих указанному шаблону. В *шаблоне* можно либо указывать значение целиком (например, Like “Иванов”), либо использовать подстановочные знаки, чтобы найти значения в некотором интервале (например, Like “Ив*”).

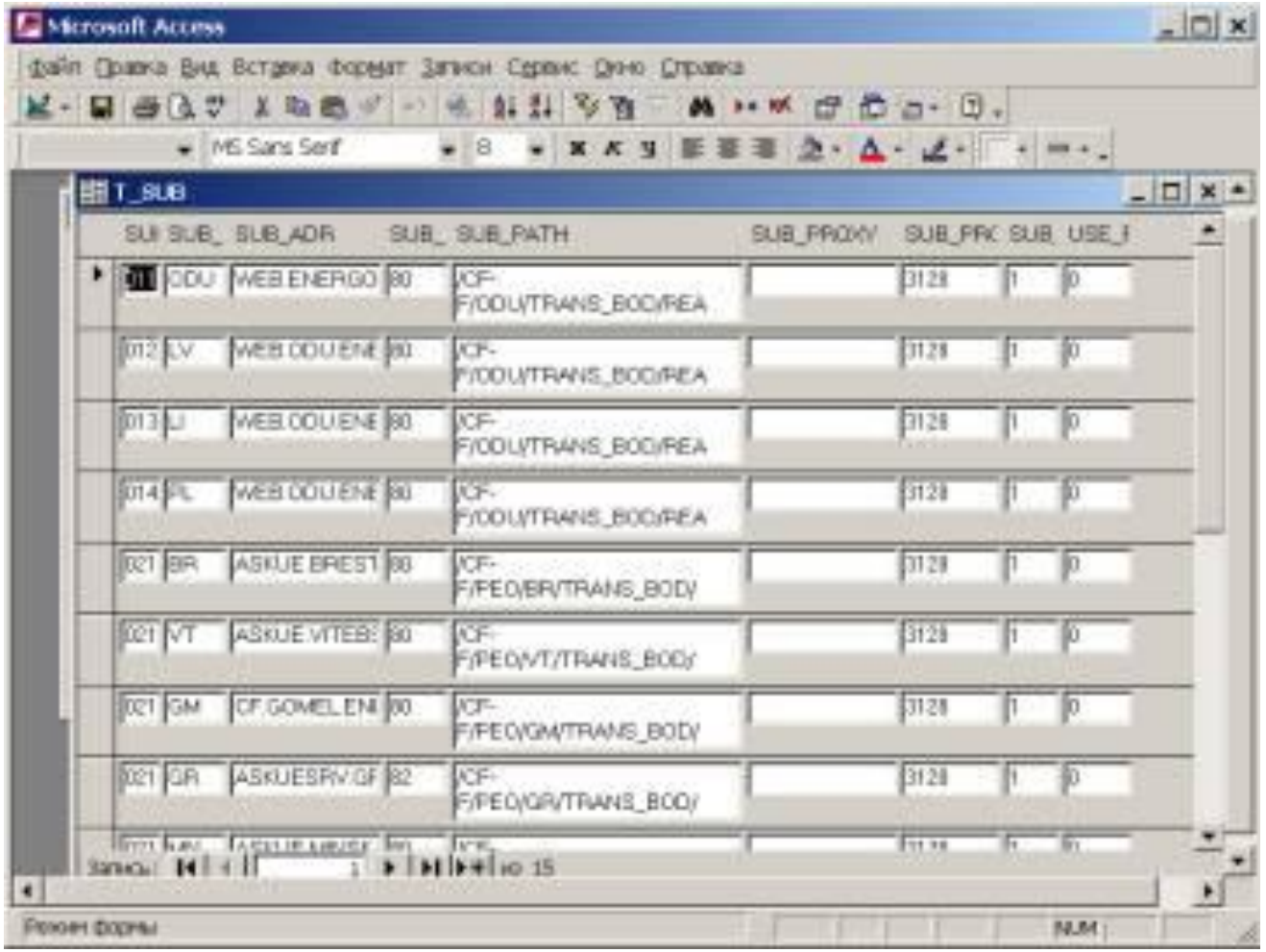
Оператор Like можно использовать в выражении для сравнения значения поля с текстовым выражением. Например, если ввести в запрос строку Like “С*”, запрос вернет все значения полей, начинающиеся с буквы «С».

СОЗДАНИЕ ЭКРАННЫХ ФОРМ ДЛЯ РАБОТЫ С ДАННЫМИ

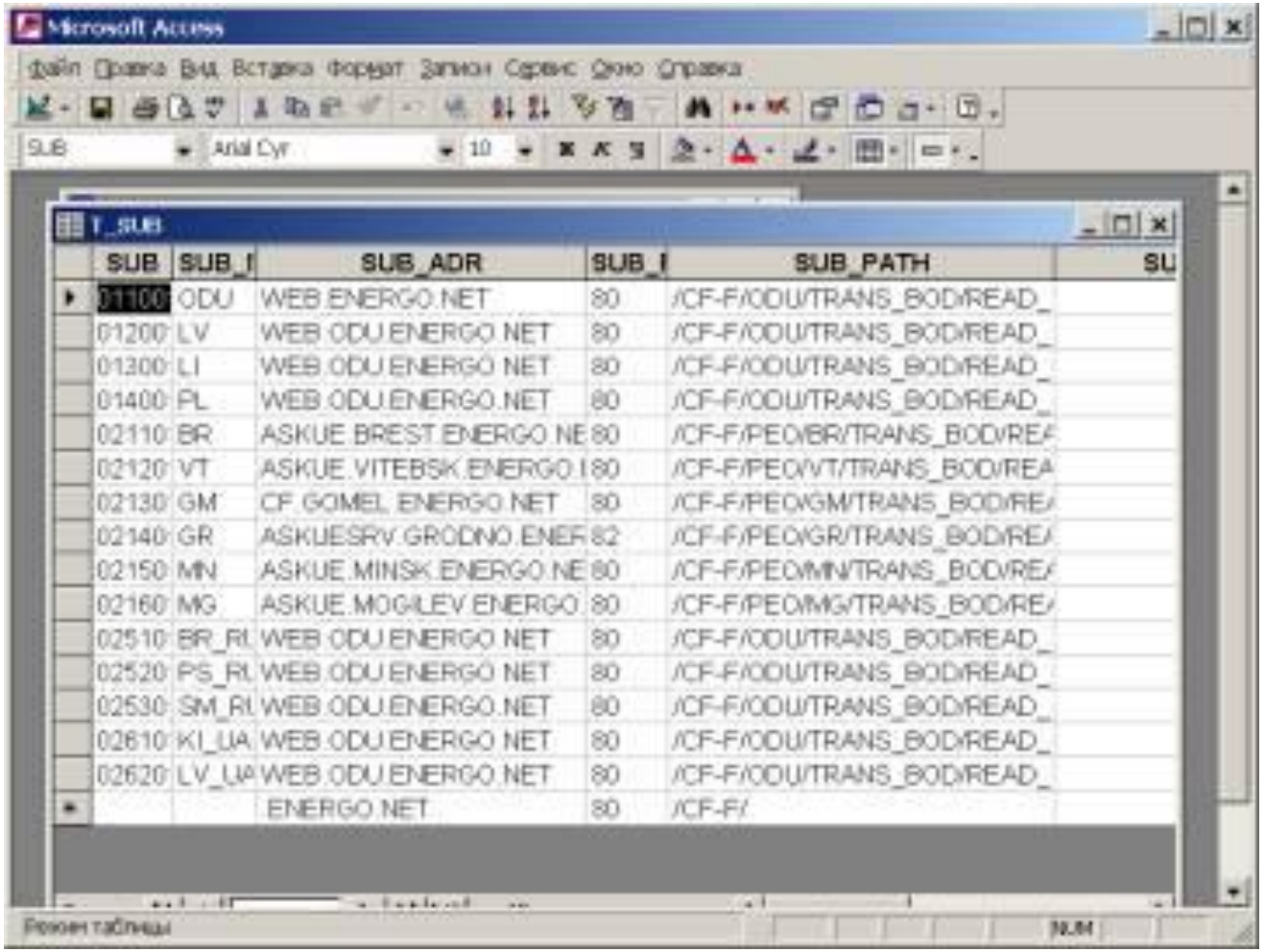
Формы предназначены для ввода и редактирования данных содержащихся в таблицах базы данных. Использование форм позволяет упростить и облегчить ввод данных потому что:

- **во-первых, структура таблицы должна строиться на основе логики задач хранения информации, которая, вообще говоря, может существенно отличаться от логики ее накопления и ввода;**
- **во-вторых, важным показателем качества автоматизированной системы является организация ее системы ввода/вывода в виде, максимально приближенном к традиционным формам представления информации на немашинных носителях.**
- **в-третьих, в сложной и развитой автоматизированной информационной системе должно обеспечиваться разделение доступа к различным группам, полей и записей для различных категорий пользователей в зависимости от выполняемых ими функций.**

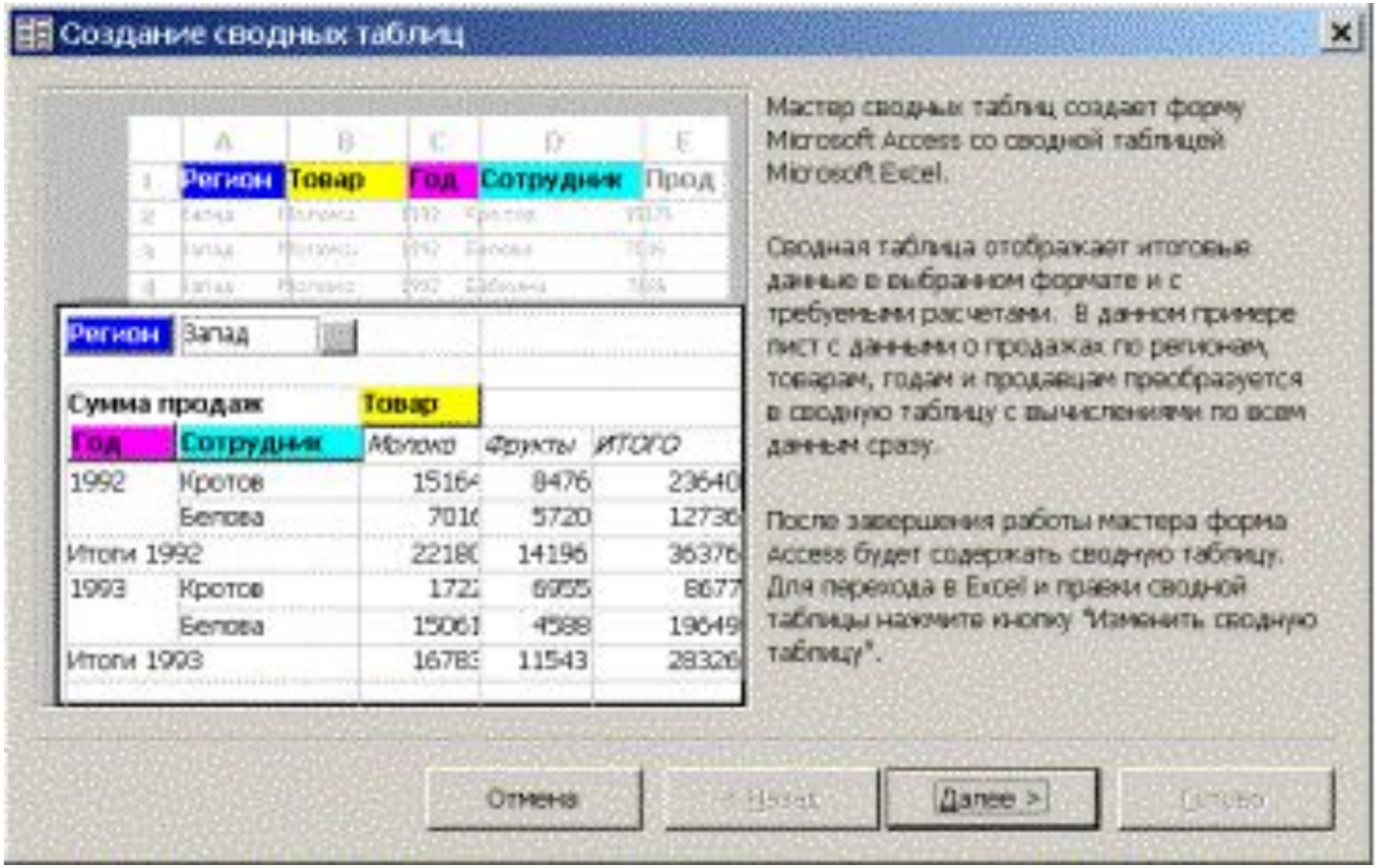
АВТОФОРМА: ЛЕНТОЧНАЯ



АВТОФОРМА: ТАБЛИЧНАЯ



СВОДНАЯ ТАБЛИЦА



ПОДГОТОВКА ОТЧЕТОВ

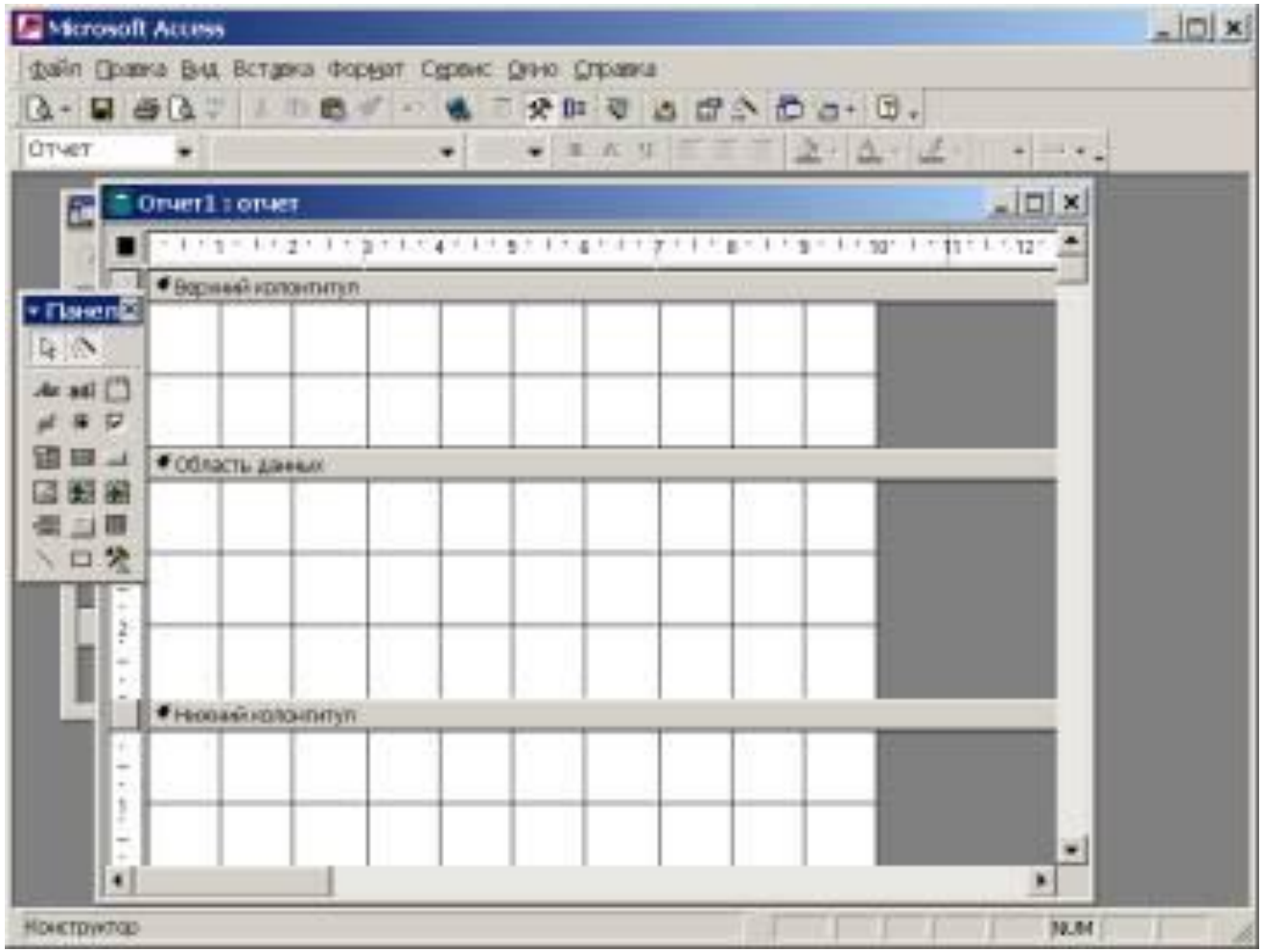
Неотъемлемой функцией любых программных систем, так или иначе связанных с обработкой данных, является представление отчетов по хранимой информации.

Под отчетом традиционно понимается специальным образом структурированное представление хранимых данных, выводимое (как правило) на жесткий бумажный носитель.

отличия отчетов от экранных форм:

- во-первых, отчеты являются исключительно средством вывода информации;
- во-вторых, организация данных в отчетах предполагает возможность их сложного, многоуровневого структурирования;
- в-третьих, структура информации, выводимой в отчете, должна быть согласована со структурой жесткого носителя. Также на внешний вид отчета значительное влияние оказывают параметры конкретного печатающего устройства, которое будет использовано для его вывода.

КОНСТРУКТОР ОТЧЕТОВ



СРЕДСТВА МАКРОПРОГРАММИРОВАНИЯ В MS ACCESS

Access, как и любая другая развитая программная система, обладает средствами разработки программных приложений, ориентированных на конечных пользователей. Эти средства базируются на инструментах двух типов:

- макросах и
- модулях.

МАКРОС

понятие *макроса* подразумевает наличие набора некоторых стандартных команд или макрокоманд таких, как:

- открытие формы,
- выполнение запроса,
- вывод отчета, и др.

из которых и конструируется сам макрос.

Макрос может содержать:

- Последовательность макрокоманд,
- *группу макросов.* Группой называют набор макросов, сохраняемый под общим именем,
- условные выражения, определяющие должна ли в запущенном макросе выполняться определенная макрокоманда

ВЫЗОВ МАКРОСА

вызов макроса может выполняться:

- по команде пользователя:
  - непосредственно из раздела Макросы главного окна базы данных,
  - либо с помощью меню или панели инструментов, с которыми он также может быть ассоциирован;
- по некоторому системному событию (**открытие** или **закрытие** формы, изменение управляющего элемента и т.п.).

МОДУЛИ

Модули, в отличие от макросов являются более мощным средством создания программных расширений в среде Access, максимально приближающимся по своим функциональным возможностям к программам создаваемым с помощью таких профессиональных инструментов, как Delphi, Visual Basic или Power Builder.

Для программирования в Access используется процедурный язык Visual Basic для приложений (VBA — Visual Basic for Applications) с добавлением объектных расширений и элементов SQL.

VBA – это объектно-орентированный язык программирования.

ТИПЫ МОДУЛЕЙ

Стандартные модули **содержат процедуры и функции, которые могут быть вызваны из любого окна базы данных. Такие модули содержат программный код универсального характера, предназначенный для применения в различных местах текущего приложения или даже в различных приложениях.**

Модули класса **используются для создания новых классов объектов. При создании конкретного объекта, являющегося экземпляром такого класса, любые процедуры, определенные в модуле, становятся свойствами и методами этого объекта.**