Оглавление

[Введение 2](#_Toc264392482)

[Глава 1. Проектирование и обработка реляционных баз данных в среде СУБД MS Access 4](#_Toc264392483)

[1.1. Реляционная модель данных 4](#_Toc264392484)

[1.2. Проектирование БД 11](#_Toc264392485)

[1.2.1. Концептуальное проектирование 13](#_Toc264392486)

[1.2.2. Логическое проектирование 18](#_Toc264392487)

[1.2.3. Физическое проектирование 19](#_Toc264392488)

[1.3. Обработка реляционных БД в среде СУБД MS Access 21](#_Toc264392489)

[Глава 2. Методика изучения темы «Информационные системы и базы данных» в школьном курсе Информатики и ИКТ 33](#_Toc264392490)

[2.1.Методические рекомендации по изложению теоретического материала 36](#_Toc264392491)

[2.2. Организация практической работы 42](#_Toc264392492)

[2.3. Сравнительный анализ содержания в различных учебных пособиях 50](#_Toc264392493)

[Заключение 63](#_Toc264392494)

[Литература 64](#_Toc264392495)

# **Введение**

Основные идеи современной информационной технологии базируются на концепции баз данных. Согласно данной концепции основой информационной технологии являются данные, организованные в БД, адекватно отражающие реалии действительности в той или иной предметной области. Но сама база данных не может обслуживать запросы пользователя на поиск и обработку информации. Обслуживание пользователя осуществляет информационная система. Такая система позволяет облегчить труд человека, повысить качество и достоверность обрабатываемой информации. Хранящиеся в ИС данные должны быть легко доступны, чтобы предоставлять достоверную информацию в определенное время, конкретному лицу, в определенном месте и с ограниченными затратами.

Первые БД появились уже на заре 1 – го поколения ЭВМ и представляли собой отдельные файлы данных. По мере увеличения объемов и структурной сложности хранимой информации определилась необходимость создания удобных эффективных систем интеграции хранимых данных и управления ими. В конце 60 – х годов это привело к созданию первых коммерческих систему управления базами данных, поддерживающих организацию и ведения БД.

Целью моей работы является раскрытие основных этапов проектирования реляционных баз данных, обработки ее в среде СУБД MS Access и рассмотрении методики изучения баз данных в школьном курсе информатики и ИКТ.

Задачи:

* изучение теоретических основ проектирования реляционных баз данных;
* рассмотрение основ обработки реляционных баз данных в среде СУБД MS Access;
* анализ методики изучения темы «Информационные системы и базы данных» в школьном курсе информатики;
* разработка урока на тему « Создание и заполнение базы данных».

# **Глава 1. Проектирование и обработка реляционных баз данных в среде СУБД MS Access**

## **1.1. Реляционная модель данных**

База данных (БД) представляет собой совокупность сведений о реальных объектах, процессах, событиях или явлениях, относящихся к определенной сфере деятельности (предметной области) и отражающих состояние объекта или множества объектов, их свойства и взаимоотношения.

Классификация баз данных:

1. По содержанию хранимой информации:

* Фактографические БД содержат данные, представляемые в краткой форме, в строго фиксированных форматах; такие БД являются аналогами бумажных карточек, например библиотечного каталога;
* Документальные БД являются аналогом архивов документов, например, исторически документов.

1. По способу хранения:

* Централизованные (локальные) БД – вся информация хранится на одном ПК: это может быть автономный ПК или сервер сети, к которому имеют доступ пользователи;
* Распределенные (сетевые) БД используются в локальных и глобальных компьютерных сетях.

1. По структуре модели данных:

* Иерархические БД строится по принципу иерархии типов объектов, т.е. один тип считается главным (корень) от него отходит другой уровень объектов, который для корня является свойством, но каждый из них может быть отдельным объектом;
* Сетевые БД – это логическая модель данных в виде произвольного графа, в котором могут быть петли и циклы. Любой объект может быть главным и подчиненным и участвовать в любом количестве взаимодействий;
* Реляционные БД объекты и взаимодействия между ними представляются с помощью таблиц. Каждая таблица имеет ключ первичный, поле или комбинация полей, которые идентифицируют каждую строку таблицы.

Реляционная модель данных была предложена известным американским специалистом в области баз данных Э.Коддом в начале 70 – х годов. Она представляет собой совокупность таблиц с установленными между ними связями. Название «реляционная» происходит от английского слова relation – отношение. Этот термин указывает, что модель отражает отношения составляющих ее частей. Каждая строка таблицы содержит набор данных об одном объекте и называется *запись*. Каждый столбец содержит различные характеристики объекта и называется *поле*.

Реляционная модель данных обладает следующими свойствами:

* Каждый элемент таблицы (ячейка) представляет собой один элемент данных, т.е. в одной ячейке реляционной таблицы не может указываться более одного значения параметра.
* Все элементы одного столбца (поля) имеют одинаковый тип (числовой, символьный и др.), формат и смысл.
* Каждый столбец (поле) имеет уникальное имя.
* Одинаковые строки в таблице отсутствуют.
* Порядок расположения строк и столбцов в таблице безразличен.

Таблица такого рода называется *отношением*. База данных, построенная с помощью отношений, называется *реляционной базой данных*, все таблицы в которой связаны между собой. Одна связь всегда соединяет только две таблицы. Связи межу таблицами могут иметь один из трех типов: «один – к - одному», «один – ко – многим», «многие – ко – многим».

* Связь «один – к – одному» (условное обозначение 1:1) означает, что одной записи в таблице А может соответствовать не более одной связанной записи в таблице В, а одной записи в таблице В может соответствовать не более одной связанной записи в таблице А.
* Связь «один – ко – многим» (условное обозначение 1:) означает, что каждой записи в таблице А может соответствовать несколько записей в таблице В, а запись в таблице В не может иметь более одной соответствующей записи в таблице А. (например, таблица ученики и таблица класс соединены по коду класса. Родительская таблица класс, дочерняя таблица ученики).
* Связь «многие – ко – многим» (условное обозначение ) означает, что одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В могут соответствовать несколько записей в таблице А. В реляционных СУБД этот тип отношений можно реализовать только через третью таблицу.

Процесс уменьшение объемов БД, использующих разбиение БД на несколько таблиц, связанных друг с другом называется *оптимизацией структуры базы данных* или *нормализацией* БД. Нормализация БД позволяет:

* Обеспечить быстрый доступ к данным;
* Исключить ненужные повторения данных, которое может являться причиной ошибок при вводе, а также привести к нерациональному использованию дискового пространства;
* Обеспечить целостность данных, т.е. чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Теория нормализации отношений работает с 5 нормальными формами таблиц. Каждой нормальной форме соответствует некоторый определенный набор ограничений. Каждая последующая форма должна отвечать требованиям предыдущих плюс некоторые дополнительные требования.

*Первая нормальная форма* (1НФ)

Таблица, находящаяся в первой нормальной форме должна отвечать следующим требованиям:

- таблица не должна иметь повторяющихся записей;

- в таблице должны отсутствовать повторяющиеся группы полей.

|  |
| --- |
| Код сотрудника |
| Имя |
| Фамилия |
| Отчество |
| Дата рождения |
| Адрес |
| Телефон |
| Должность |
| Разряд |
| Зарплата |
| Рейтинг |
| Дата приема |
| Дата увольнения |

Рис.1. Таблица, структура которой соответствует 1НФ.

*Вторая нормальная форма* (2НФ)

Таблица, находящаяся во второй нормальной форме должна отвечать всем требованиям 1НФ, а также любое неключевое поле однозначно идентифицируется полным набором ключевых полей 2НФ применяется к таблицам, которые имеют составной ключ (см.рис.2).

|  |  |  |
| --- | --- | --- |
| Код физического лица |  | Код сотрудника |
| Имя |  | Код физического лица |
| Фамилия |  | Должность |
| Отчество |  | Разряд |
| Дата рождения |  | Зарплата |
| Адрес |  | Дата приема |
| Телефон |  | Дата увольнения |

Рис.2.Приведение таблицы ко второй нормальной форме.

*Третья нормальная форма* (3НФ)

Таблица, находящаяся в третьей форме должна отвечать всем требованиям 2НФ, а также ни одно из неключевых полей не идентифицируется при помощи другого неключевого поля. Другими словами в таблице нет полей, которые не зависят от ключа. На практике третья нормальная форма схем отношений в большинстве случаев достаточна, и приведение к ней процесс проектирования реляционных баз данных обычно заканчивается.

|  |
| --- |
| Код сотрудника |
| Код должности |
| Код физического лица |
| Дата приема |
| Дата увольнения |

|  |  |  |
| --- | --- | --- |
| Код сотрудника |  | Код должности |
| Код физического лица |  | должность |
| Должность |  | разряд |
| Разряд |  | зарплата |
| Зарплата |  |  |
| Дата приема |  |  |
| Дата увольнения |  |  |

Рис.3. Приведение базы данных к третьей нормальной форме.

Основные понятия реляционной модели данных являются:

* *Домен*. Наименьшая единица данных реляционной модели – это отдельное атомарное (неразложимое) для данной модели значение данных. Доменом называется множество атомарных значений одного и того же типа.
* *Тип данных* в реляционной модели данных полностью эквивалентно соответствующему понятию в алгоритмических языках. Все СУБД поддерживают следующие типы данных: целочисленные, вещественные, строковые, специальные типы данных для временных величин (дата и / или время).
* *Атрибут*. Столбцы отношения называют атрибутами, им присваиваются имена, по которым к ним затем производится обращение.
* *Ключ*. Простой ключ – ключ, содержащий только один атрибут. Составной ключ – это ключ, состоящий из несколько атрибутов.

Чтобы информация, хранящаяся в базе данных, была однозначной и непротиворечивой, в реляционной модели устанавливаются некоторые ограничительные условия. Ограничительные условия – это правила, определяющие возможные значения данных. Они обеспечивают логическую основу для поддержания корректных значений данных в базе. Такие ограничения целостности позволяют свести к минимуму ошибки, возникающие при обновлении и обработки данных.

Реляционная база данных представляет собой совокупность отношений, содержащих всю информацию, которая должна храниться в базе данных. Однако пользователи могут воспринимать такую базу данных как совокупность таблиц. Таким образом, реляционную базу данных можно рассматривать как хранилище данных, содержащих набор двухмерных таблиц. Набор средств управления подобным хранилищем называется реляционной системой управления базами данных. Она может содержать утилиты, приложения, службы, библиотеки и другие приложения.

## **1.2. Проектирование БД**

Первым шагом при создании базы данных является создание плана, который одновременно выступает в качестве руководства при внедрении базы данных и в качестве ее функциональной спецификации в ходе ее дальнейшего использования. Сложность и подробность проектирования базы данных определяется сложностью и размером приложения базы данных, а также количеством пользователей.

При проектировании базы данных, независимо от ее размера и сложности, необходимо придерживаться следующих основных шагов:

* сбор сведений;
* выделение объектов;
* моделирование объектов;
* определение типов данных для каждого объекта;
* определение связей между объектами.

Проектирование БД осуществляют:

* специалисты по информационным системам (программисты),
* специалисты предметной области,
* технический работник для ведения документации (помощник администратора БД),
* заказчик – основной носитель сведений о предметной области.

Существуют два основных подхода к проектированию систем баз данных: нисходящий и восходящий. При *восходящем* подходе работа начинается с самого нижнего уровня атрибутов (т.е. свойств сущностей и связей), которые на основе анализа существующих между ними связей группируются в отношения, представляющие типы сущностей и связи между ними. Например, процесс нормализации представляет собой вариант восходящего подхода при проектировании баз данных. Нормализация предусматривает идентификацию требуемых атрибутов с последующим созданием из них нормализованных таблиц, основанных на функциональных зависимостях между этими атрибутами.

Восходящий подход в наибольшей степени приемлем для проектирования простых баз данных с относительно небольшим количеством атрибутов. Однако использование этого подхода существенно усложняется при проектировании баз данных с большим количеством атрибутов, установить среди которых все существующие функциональные зависимости довольно затруднительно. Поскольку концептуальная и логическая модели данных для сложных баз данных могут содержать от сотен до тысяч атрибутов, очень важно выбрать подход, который помог бы упростить этап проектирования. Кроме того, на начальных стадиях формулирования требований к данным в крупной базе данных может быть трудно установить все атрибуты, которые должны быть включены в модели данных.

Более подходящей стратегией проектирования сложных баз данных является использование *нисходящего* подхода. Начинается этот подход с разработки моделей данных, которые содержат несколько высокоуровневых сущностей и связей, затем работа продолжается в виде серии нисходящих уточнений низкоуровневых сущностей, связей и относящихся к ним атрибутов. Нисходящий подход демонстрируется в концепции модели "сущность-связь". *Сущность-* это реальный или виртуальный объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

*Связь* – это соединение двух сущностей.

В этом случае работа начинается с выявления сущностей и связей между ними, интересующих данную организацию в наибольшей степени.

Кроме этих подходов для проектирования баз данных могут применяться другие подходы, например, подход "от общего к частному" или "смешанная стратегия проектирования". Подход "от общего к частному" напоминает восходящий подход, но отличается от него тем, что вначале выявляется набор основных сущностей с последующим расширением круга рассматриваемых сущностей, связей и атрибутов, которые взаимодействуют с первоначально определенными сущностями. В смешанной стратегии сначала используются восходящий и нисходящий подходы для создания разных частей модели, после чего все подготовленные фрагменты собираются в единое целое.

Процесс проектирования базы данных состоит из трех основных этапов: концептуальное, логическое и физическое проектирование.

### **1.2.1. Концептуальное проектирование**

Концептуальное проектирование оперирует информацией, независимой от любо фактической реализации (т.е. от любой конкретной системы технического и программного обеспечения). Цель концептуального проектирования именно в том и состоит, чтобы представить информацию в доступной пользователю форме, не зависящей от спецификации системы, но реализуемой несколькими системами.

Этап концептуального проектирования связан с описанием и синтезом разнообразных информационных требований пользователей к проекту информационной системы (базы данных) итогом его является составление информационной структуры, отображающей концептуальную модель предметной области.

Различают два подхода в концептуальном проектировании: это объектный подход и моделирование сущностей.

*Технология объектного представления* – это формулирование, определение и интеграция объектов высокого уровня, используемых для построения модели, объединенные в виде иерархии объектов. Основными вопросами, решаемыми при этом подходе, являются следующие:

Что понимают под объектами?

Какого контекстное содержание этих объектов?

Каковы описательные свойства каждого объекта?

*Моделирование сущностей*

Техника построения диаграмм сущностей, является в основном неформализованной, обладает конечным результатом спецификацию сущностей, атрибутов и связей. Для представления информации в диаграммы в модели «сущность – связь» конструктивными элементами модели служат сущности, атрибуты и связи. Пользователь описывает интересующие его объекты предметной области с помощью сущностей, затем определяет свойства сущностей, используя атрибуты, и, наконец, описывает соответствия между сущностями, используя связи.

Основа диаграммы:

* набор сущностей, который представляет или моделирует определенную совокупность сведений в предметной области,
* в свою очередь сущности могут быть описаны атрибутами, позволяющими детализировать свойства сущностей,
* один или несколько атрибутов могут служить идентификатором для обозначения отдельных экземпляров сущности,
* связь между сущностями отображают функциональные аспекты информации, представленной сущностями.

*Методология концептуального проектирования*

1. Анализ сущностей с дальнейшей детализацией.

Это нисходящий подход, который делит процесс проектирования на четыре стадии:

1. Моделирование представлений;
2. Объединение представлений;
3. Составление и анализ схемы;
4. Физическое проектирование.

Для концептуального проектирования представляют интерес только две первые стадии.

Под стадией *моделирование представлений* понимаются моделирование информации, требуемой для базы данных, отображающий различные аспекты деятельности организации.

Виды представлений являются:

* Общие представление (с точки зрения организации в целом);
* Прикладное представление;
* Информационное представление;
* Представление событий.

Моделирование представлений заключается в фактическом сборе информации на различных уровнях организации в соответствии с четырьмя видами представлений. Оно включает входную информацию от исполнителей, руководящего состава и конечных пользователей.

Основные правила моделирования:

* Проектное представления моделируются с помощью трех типов конструктивных элементов: сущностей, атрибутов и связей;
* Каждый компонент информации в проектном представлении изображается одним и только одним конструктивным элементом.

Моделирование проектных представлений состоит из последовательности шагов, завершающейся моделью локального представления.

Шаг 1. *Идентификация локальных представлений.*

Набор локальных представлений, соответствующим независимым областям данных, относящимся к функциональным областям. Например, в системе приема заказов в качестве локальных представлений могли бы выступать такие функциональные задачи, как прием заказов, расчеты с клиентами, история заказа.

Шаг 2. *Формулирование сущностей*.

Для каждого локального представления могут быть сформулированы сущности, требуемые для описания этого локального представления. Использование сущности в качестве конструктивного элемента. Часто некоторая порция информации может быть представлена как атрибут, сущность или связь. Например, тот факт, что двое служащих находятся в семейных отношениях, может быть выражен сущностью СЕМЬЯ, связью ЖЕНАТ – НА или атрибутом СУПРУГ (А).

Шаг 3. *Выбор идентифицирующего атрибута для каждой сущности*.

Идентификатор служит для однозначного распознавания отдельных элементов сущности и может состоять из одного или нескольких атрибутов, набор значений которых уникален.

Шаг 4. *Спецификация связей*.

Локальное представление дополняется информацией, определяющей типы связей: необязательная, возможная, обязательная, однозначная и условная. Одна из неформальных процедур для этого шага заключается в попарном объединении между собой всех сущностей, содержащих в данном представлении.

Шаг 5. *Добавление описательных атрибутов к сущностям*.

Атрибуты могут быть разделены на два класса: те, которые служат для идентификации экземпляров сущности, и те, которые описывают свойства сущности. Пример, сущность СЛУЖАЩИЙ, которая идентифицируется атрибутами НОМЕР и ИМЯ и описывается атрибутами ПЛАТЕЖНЫЙ – БАЛАНС и АДРЕС, при этом АДРЕС включает УЛИЦУ, ГОРОД, НОМЕР ДОМА.

Процесс *объединения представлений* заключается в интеграции различных представлений, полученных на предыдущей стадии, в единое для всей организации концептуальное представление информации и требований обработки данных. Интегрированное представление диаграммы информационной структуры составляют основу подхода к управлению базами данных.

Этот процесс включает анализ и принятие решений на нескольких уровнях:

*Несогласованность наименований*. Идентификация синонимов и омонимов среди элементов данных.

*Несогласованность идентификации*. Различная идентификация одних и тех же типов сущностей.

*Несогласованность агрегации*. Ограничение различных групп элементов на структурном уровне или операций над значениями элементов на уроке экземпляров. Пример, оценки за 1 четверть, 1 полугодие или за год.

*Дополняющие подмножества*. Распознавания взаимодополняющих друг друга подмножеств данных, таких, как «ученики дневной смены», «ученики вечерней смены» и «отчисленный ученики».

Основной результат процесса объединения представлений – глобальная информационная структура. Она является интеграцией обобщенного, прикладного, информационного представлений и представления событий.

1. Синтез атрибутов с дальнейшей интеграцией.

Эта методология называется восходящей, так как она начинается с синтеза атрибутов самого нижнего уровня, из которых затем формируются сущности и связи верхнего уровня. Результатом анализа требовании является полный список элементов данных, используемых в различных задачах организации. Эти элементы данных с помощью эвристических правил классифицируются по типам атрибутов, а так же в отношении членства в сущности.

*Композиция сущностей*. Атрибуты принадлежат сущностям двух типов: уникальным сущностям и неуникальным сущностям (зависимым).

*Формулирование связей*. На этой стадии, кроме двух типов сущностей, необходимо использовать так же другую информацию, такую, как политика организаций, а так же информацию, полученную в результате собеседований. Эта информация используется для определения связей между типами сущностей.

*Графическое представление*. На конечной стадии вышеописанные атрибуты, сущности и связи оформляются графически в терминах модели «сущность – связь». Используется следующая последовательность действий:

* Изобразить все уникальные и неуникальные сущности;
* Изобразить графически все связи между сущностями;
* Представить перекрестные связи между сущностями.

Корректность модели тщательно проверяется при анализе архитектуры системы. При этом требования пользователя к данным и процессам рассматривается с различных позиций.

### **1.2.2. Логическое проектирование**

Целью логического проектирования (проектирование реализаций) является составление логической модели предметной области.

Изменения, которые вносятся в структуру базы данных на этом этапе, определяются стремлением удовлетворить требованиям конкретной СУБД и наиболее общим ограничением, специфицированном в требованиях пользователей.

Основной задачей проектирование реализации являются разработка СУБД – ориентированной схемы, которая удовлетворяет всему диапазону требований пользователей, начиная с требований целостности и непротиворечивости проектируемой базы данных и заканчивая показателями эффективности функционирования при ее расширении и усложнении.

Содержание процесса проектирования реализации

1. Исходные данные:

*СУБД – независимая схема*. Исходная схема, которая в дальнейшем будет преобразовываться на фазе проектирования реализации.

*Количественная оценка эксплуатационных характеристик*. Спецификация требований целостности, восстанавливаемости, безопасности.

*Количественная оценка объема и частоты выполнения приложений*. Размер базы данных, который оценивается исходя из количества экземпляров данных и частоты выполнения приложений.

*Требования непротиворечивости*. Правила поддержания взаимной непротиворечивости элементов данных, правила устранения противоречивости данных.

*Характеристики СУБД*. Правила задания СУБД – ориентированных логических схем и подсхем.

*Вычислительные средства*. Ограничения на конфигурацию и объем аппаратного и математического обеспечения.

1. Результаты:

*СУБД – ориентированная схема*. Спецификация структуры базы данных, которая может быть реализована конкретной СУБД, не содержит большинства физических параметров, определяющих группирование записей или размеры блоков. Однако она может включать некоторые параметры путей доступа, такие, как упорядоченность, указатели и механизмы поиска.

*Спецификация для физического проектирования*. Полностью документированные схемы и подсхемы с указанием объема, частоты выполнения приложений.

*Руководство для группы сопровождения базы данных*. Краткие требования, ограничения и данные об имеющихся в наличии технических средствах и математическом обеспечении для администратора базы данных.

### **1.2.3. Физическое проектирование**

Цель: привязка базы данных к физической памяти.

Третьим и самым нижним уровнем представления базы данных является физический уровень. Физическая организация данных оказывает основное влияние на эксплуатационные характеристики проектируемой базы, так как именно на этом уровне осуществляется ее привязка к физической памяти.

Очевидными компонентами результирующей физической структуры базы данных являются: формат хранимой записи, спецификация размещения хранимой записи и методы доступа. Эти спецификации должны удовлетворять все эксплуатационным требованиям и ограничениям, налагаемым техническими и программными средствами системы.

Физический этап делится на 2 под этапа:

1. Принимаются основные решения

- о формате хранимых данных;

- избыточность, сжатие хранимых данных;

- кластеризация хранимых данных (объединение записей различного типа в физические группы, для эффективного размещения данных);

- проектирование методов доступа (структура памяти, индексы записи, механизм поиска).

2. Учет ограничений и проектирование программ

- вопросы целостности и безопасности данных;

- проектирование программ.

Так как основной целью физического проектирования базы данных является описание способа физической реализации логического проекта базы данных. Под этим подразумевается следующее принципы преобразования:

* Каждая сущность преобразуется в таблицу, и имя сущности становится именем таблицы;
* Каждый атрибут становится столбцом таблицы с тем же именем, уточняется тип данных, выбирается более точный формат;
* Идентифицирующие атрибуты сущности превращаются в первичный ключ таблицы;
* Отношения «многие к одному» и «один к одному» становятся внешними ключами;
* Для отношений «многие ко многим» создается таблица, столбцами которой являются уникальные идентификаторы связываемых сущностей.

## **1.3. Обработка реляционных БД в среде СУБД MS Access**

Для работы с данными используются специальные программы – системы управления базы данных (СУБД). Выбор СУБД зависит от модели, которая положена в основу базы данных (иерархическая, сетевая и реляционная). Система управления базами данных – комплекс программных и лингвистических средств общего или специального назначения, реализующий поддержку создания баз данных, централизованного управления и организации доступа к ним различных пользователей в условиях принятой технологии обработки данных.

Задачи СУБД:

* Хранение информации в структурированном виде;
* Обновление информации;
* Поиск нужной информации;
* Выдача информации пользователю в удобном для него виде;
* Устранение избыточности данных.

СУБД обеспечивает:

* Описание и сжатие данных;
* Манипулирование данными;
* Физическое размещение и сортировку записей;
* Работу с транзакциями и файлами;
* Безопасность данных.

Архитектура построения сетевой работы СУБД и БД:

* Централизованная (СУБД и БД размещаются в центральном ПК, а пользовательский ПК – только устройство ввода и отображение информации);
* Файл – сервер (БД – на сервере, копии СУБД на пользовательских ПК);
* Клиент - сервер (БД, СУБД – серверная – на сервере, а СУБД – клиентская части на пользовательском ПК);
* Трехуровневая (отдельно вынесен сервер приложений с деловой логикой и интерфейсом, БД – на сервере, тонкий клиент – ПК пользователя – пример, web- браузер).

Существуют СУБД, ориентированные на программиста и СУБД, ориентированные на конечного пользователя. СУБД MS Access относится к системам, ориентированным на пользователя. Она позволяет пользователю, не прибегая к программированию, выполнять основные действия с базами данных: создание, редактирование и манипулирование данными.

MS Access работает в операционной среде MS Windows, может использоваться как на автономном ПК, так и в локальной компьютерной сети. Основное назначение Access – работа с реляционными базами данных.

Стандартным языком реляционных СУБД является язык SQL (Structured Query Language – структурированный язык запросов). Он используется при создании запросов, а так же для обновления и управления реляционными базами данных, такими как базы данных Microsoft Access. Когда пользователь создает запрос в режиме конструктора запроса, Microsoft Access автоматически создает эквивалентную инструкцию SQL. Пользователь имеет возможность просматривать и изменять инструкцию SQL в режиме SQL.

Объектами СУБД Access являются *таблицы, запросы, формы, отчеты*, *макросы и модули*.

*Таблица* – это главный тип объектов. Все остальные являются производными от таблиц. Элементы данных, составляющих таблицу, - это записи и поля. В таблицах хранятся данные и отображаются все записи и поля, а так же в таблицах хранятся установленные связи. Прежде чем создать базу данных в Access, необходимо разработать структуру таблиц БД и установить связи между ними.

*Структура таблицы* – это описание порядка следования полей, их типов, размеров, имен и других признаков.

*Запись* – последовательность значений, описанных в структуре.

*Поле* – элемент записи, содержащий информацию об одном свойстве объекта и характеризующийся следующими параметрами:

* Имя поля (записывается без пробелов – как имя переменной);
* Тип значений поля (текстовый, числовой, логический, дата – время);
* Размер поля (зависит от типа возможных значений);
* Значение поля (соответствует конкретной записи);

При разработке структуры таблиц следует исключать многократное повторение одних и тех же записей. С этой целью необходимо изменить структуру таблицы, уменьшив ее объем за счет разбиения на несколько взаимосвязанных таблиц, процесс нормализации модели данных.

Структуру таблиц в многотабличной базе данных надо создавать таким образом, чтобы можно было установить между ними связь, т.е. таблицы должны содержать общие поле, одинаковые по типу и формату хранимых данных.

*Схема данных* – описание структуры связей в многотабличной базе данных.

В MS Access существуют различные способы создания таблиц. (см. табл.1).

Таблица 1. Способы создания таблиц в MS Access

|  |  |
| --- | --- |
| **Режим** | **Описание** |
| Режим таблицы | Необходимый макет таблицы формируется соответствующим вводом полей в заголовок таблицы, добавлением либо удалением соответствующих столбцов (для этого удобно использовать контекстное меню выделенного столбца). Не закрывая подготовленного макета таблицы, вводятся данные в строки таблицы. При сохранении таблицы автоматически, анализируются данные и соответствующему полю присваивается тип данных; кроме того, происходит запрос о ключевом поле и об имени таблицы. |
| Мастер таблиц | Новая таблица формируется на основе существующих таблиц (шаблонов), имеющих различные поля. Категории шаблонов (деловые, личные и т.п.), образцы таблиц, а также наличие понятных рекомендаций не вызывают сложностей при создании новой таблицы. |
| Конструктор | Формирование структуры таблицы происходит при заполнении соответствующего бланка, в котором указывается необходимые поля, тип данных, свойства поля и т.д. Режим конструктора предоставляется наиболее широкие возможности по созданию объектов таблицы, и его удобно использовать для внесения необходимых корректив в таблицы, созданные различными способами. |
| Импорт таблиц | Осуществляется импорт данных и объектов из внешнего файла в текущую БД. |
| Связь с таблицами | Создаются таблицы, связанные с таблицами внешнего файла. |

Работа по созданию таблиц и схемы данных может быть разбита на следующие этапы:

1. Создание и определение структуры таблицы:

* Создание новой таблицы (перейти к объекту Таблицы и выбрать один из способов создания таблиц);
* Определение полей, типов данных, описаний (при необходимости) и свойства полей, включая маски ввода и условия на значение;
* Задание первичного ключа (возможны составные);
* Создание индексов для необходимых полей;
* Сохранение таблицы в базе.

1. Связывание таблиц в схему данных с учетом требований целостности данных (это означает, что связанные поля должны иметь один тип и формат данных).
2. Определение полей подстановки для удобства работы с данными при заполнении таблиц (тип данных «Мастер подстановок»).

Пример таблиц и схемы данных из базы данных «Студенческий Трудовой Отряд». С помощью этой базы данных сотрудники организации могут контролировать заказы различных магазинов. Так же могут узнать информацию о работниках организации ( личные ,рабочие и медицинские) .

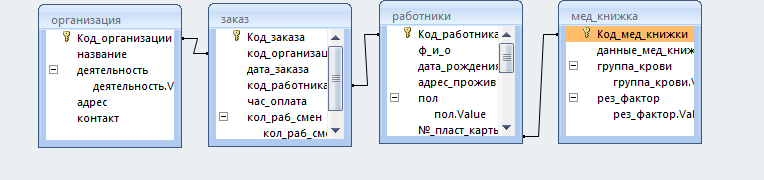


Рис.4. Схема данных «СТО».

Структура таблицы: мед\_книжка

**Столбцы**

Код\_мед\_книжки Текстовый 255

DisplayControl: Поле

данные\_мед\_книж Текстовый 255

DisplayControl: Поле

группа\_крови 4

рез\_фактор 4

*формы* – это настраиваемые диалоговые окна, сохраняемые в базе данных в виде объектов специального типа. Используются для просмотра, ввода и редактирования одной записи таблицы. Формы могут содержать элементы управления, с помощью которых осуществляется доступ к данным в таблицах: текстовые поля для ввода и правки данных, кнопки, флажки, переключатели, списки, надписи, а также рамки объектов для отображения графики и объектов OLE (рисунки, звуки и т.д.).

Формы в Access используются для поддержки следующих задач:

* Ввод, редактирование и просмотр информации, находящейся в таблицах;
* Отображение результатов запросов;
* Организация экранного окна поиска необходимой информации по БД;
* Распечатка данных в организованном виде;
* Сохранение в виде отчета с последующей модификацией;
* Создание пользовательских экранных форм, облегчающих работу с базой данных в целом.

Таблица 2. Способы создания форм в MS Access

|  |  |
| --- | --- |
| **Режим** | **Описание** |
| Конструктор | Позволяет создать новую форму самостоятельно с использованием окна конструктора форм. |
| Мастер форм | Автоматическое создание формы одного из трех стандартных типов (в столбец, ленточную или табличную) на основе выбранных полей. |
| Автоформа: в столбец | Автоматическое создание формы с полями в один столбец. |
| Автоформа: ленточная | Автоматическое создание ленточной формы. |
| Автоформа: табличная | Автоматическое создание табличной формы. |
| Автоформа: свободная таблица | Автоматическое создание формы в режиме свободной таблице. |
| Автоформа: свободная диаграмма | Автоматическое создание формы в режиме свободной диаграммы. |
| Диаграмма | Создание формы с диаграммой. |
| Свободная таблица | Создание формы со свободной таблицей. |

Макет формы состоит из разделов:

* Заголовок формы – определяет верхнюю часть формы. В область заголовка можно поместить текст, графику и другие элементы управления.
* Верхний колонтитул – сведения, которые необходимо размещать на каждой странице при печати.
* Область данных – содержит основные данные таблиц.
* Нижний колонтитул – дата, номера страниц, сведения, которые должны быть внизу страницы.
* Область примечаний – кнопки и инструкции по использованию формы.

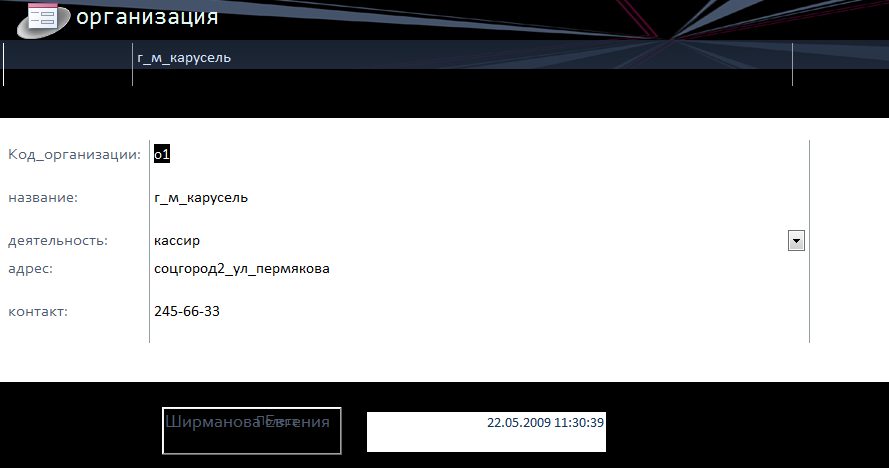


Рис.5.Форма для редактирования и заполнения таблицы организация.

Таблицы и Формы – это режимы, используемых для создания и просмотра базы данных.

*Запрос* – это динамический (виртуальный) набор данных, которые существуют только во время выполнения запроса. Запросы позволяют осуществить выборку данных по некоторому критерию или выполнить определенные действия с данными. Одновременно выборка может производиться из 16 таблиц. В запрос можно включать до 225 полей. С помощью запросов можно создавать новые таблицы, используя данные из одной или нескольких существующих таблиц. В Access предусмотрено создание запросов в двух режимах:

QBE (Query – by – Example) – язык запросов по образцу, использующий графический бланк для конструирования запросов (режим Конструктор).

SQL (Structured Query Language) – встроенный структурированный язык запросов.

Основные типы запросов:

* Запрос на выборку – выводит данные из одной или нескольких таблиц в соответствии с заданными критериями;
* Запрос с параметром – сопровождается выводом на экран одного или более диалоговых окон, предназначенных для ввода пользователем конкретных значений параметров запроса;
* Групповой запрос – предусматривает выполнение вычислений с использованием данных некоторой группы записей;
* Перекрестный запрос – выводит результаты статистических расчетов (сумма, количество) для данных, которые находятся в одной или нескольких таблиц (результаты таких запросов используются для анализа данных и создания диаграмм);
* Модифицирующие запросы – используются для создания новых таблиц из результатов запросов и для внесения изменений в данные существующих таблиц. Различают запросы на обновление, удаление, добавление данных.
* SQL – запрос – используются средства языка запросов SQL для создания запросов трех видов: запрос на объединение, запрос к серверу, управляющий запрос.

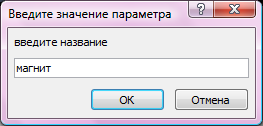
Таблица 3. Режимы создания запросов в MS Access

|  |  |
| --- | --- |
| **Режим** | **Описание** |
| Конструктор | Самостоятельно создание запроса в графическом бланке QBE. |
| Простой запрос | Создание запроса с помощью мастера выбором соответствующих полей одной или нескольких таблиц. |
| Перекрестный запрос | Создание запросов с помощью мастера, результаты которого представлены в виде двумерной таблицы. |
| Повторяющиеся записи | Создание запроса с помощью мастера на поиск повторяющихся записей в простой таблице или запросе. |
| Записи без подчинений | Создание запроса с помощью мастера на поиск записей, которым не соответствует ни одна запись в подчиненной таблице. |

Пример запроса с параметром «Поиск организации».

Ввод: символы названия магазина

Образец ввода:



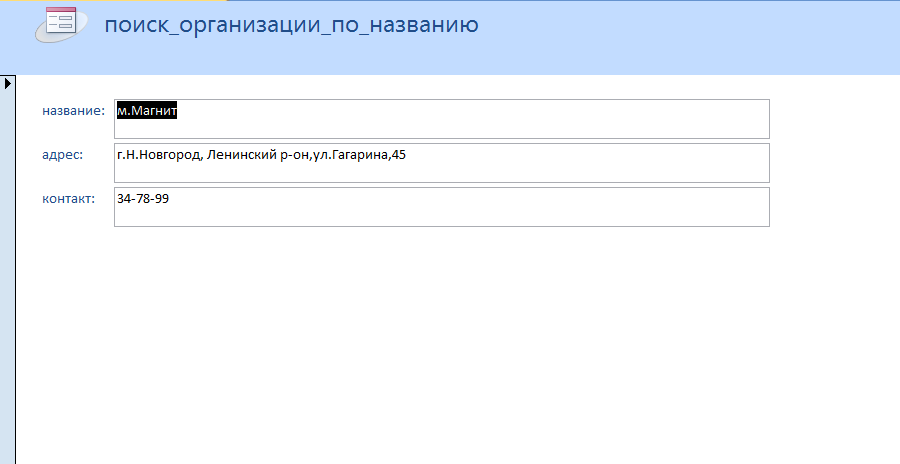
SQL-запрос:

SELECT организация.название, организация.адрес, организация.контакт

FROM организация

WHERE (((организация.название) Like "\*" & [введите название] & "\*"));

Форма вывода:



*Отчет* – объект, предназначенный для анализа данных и вывода их в файл, текстовый документ или на экран по созданию образа отчета.

Источником данных для отчета может быть таблица или запрос. Кроме данных, полученных из таблиц, в отчете могут отображаться вычислительные по исходным данным значения, например итоговые суммы.

Создание отчета предусматривает выполнение следующих основных этапов:

1. Определение макета отчета. На данном этапе необходимо определить цель отчета и составить общее представление о том, какого вида будет отчет. Следует учесть, что отчет, создаваемый средствами Access, может включать:

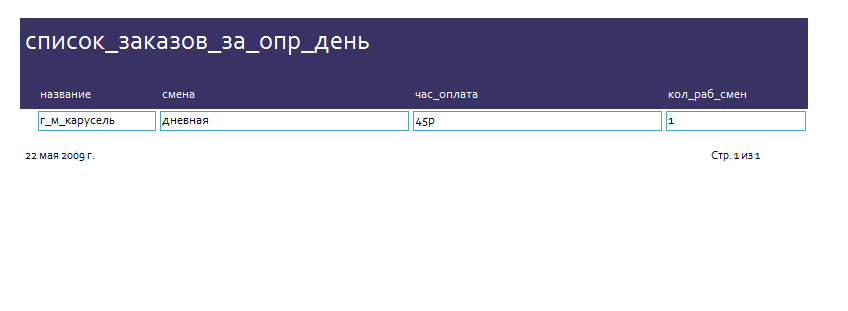
* Иерархию в представлении данных;
* Формирование заголовков и примечаний, как для групп данных, так и для всего отчета;
* Использование необходимых вычислений, как для групп данных, так и для всего отчета;
* Формирование колонтитулов для всего отчета.

1. Сбор данных. На этом этапе необходимо определить, какая таблица либо запрос будет служить источником данных для создаваемого отчета. Если необходимая информация присутствует в различных таблицах и не создано подходящего запроса, рекомендуется перед созданием отчета определиться с необходимыми полями и создать запрос, который будет служить источником данных для отчета.
2. Создание отчета. Одним из способов: конструктор, мастер отчетов, автоотчет (в столбец, ленточный, табличный) и другие.
3. Распечатка отчета.

Структура отчета:

* *Заголовок отчета*. Печатается один раз в начале отчета и может содержать эмблему, название и дату печати отчета.
* *Верхний колонтитул*. Печатается в верхней части каждой страницы отчета и может содержать заголовки столбцов.
* *Область данных*. Содержит основную часть данных, которые требуются представить в отчете.
* *Нижний колонтитул*. Печатается в нижней части каждой страницы отчета и может содержать номера страниц.
* *Примечание отчета*. Печатается один раз в конце отчета и может содержать результаты вычислений над всеми записями отчета.

Пример отчета к запросу «Список заказов за определенный день»



*Макрос* – макрокоманда или набор макрокоманд, используемых для автоматического выполнения некоторых операций.

*Модуль* – программа, написанная на языке Vbasic (совокупность описаний, инструкций и процедур, сохраненная под общим именем).

С помощью Access создаются и эксплуатируются личные базы данных, а также базы данных организаций с относительно небольшим объемом данных. СУБД Access объединяет сведения из разных источников в одной реляционной базе данных. Создаваемые формы, запросы и отчеты позволяют быстро и эффективно обновлять данные, получать ответы на вопросы, осуществлять поиск нужных данных, анализировать данные, печатать отчеты. В базе данных сведения из каждого источника сохраняются в отдельной таблице. При работе с данными из нескольких таблиц устанавливаются связи между таблицами. Для поиска и отбора данных, удовлетворяющих определенным условиям, создается запрос. Запросы позволяют также обновить или удалить одновременно несколько записей, выполнить встроенные или специальные вычисления. Для просмотра, ввода или изменения данных прямо в таблице применяются формы. Форма позволяет отобрать данные из одной или нескольких таблиц и вывести их на экран, используя стандартный или созданный пользователем макет. Для анализа данных или распечатки их определенным образом используется отчет.

# **Глава 2. Методика изучения темы «Информационные системы и базы данных» в школьном курсе Информатики и ИКТ**

Тема «Информационные системы и Базы данных» традиционно входит во все программы базового курса информатики. В базовых учебных пособиях изучается после темы информационное моделирование.

До темы БД изучались информационные и натуральные модели, виды информационных моделей: вербальные, графические, математические. Табличная организация информации.

*Основные цели изучения темы информационные системы и базы данных*. Дать ученикам представления о назначении информационных систем и баз данных. Познакомить детей с основами реляционных баз данных. Обучить их основным приемам работы с одной из реляционных СУБД. Обучить учеников организации поиска, сортировки, редактирования данных. В углубленном варианте: дать детям представления о проблемах проектирования реляционных баз данных.

Изучаемые вопросы:

* Назначение информационных систем и баз данных;
* Классификация БД;
* Структура реляционной базы данных;
* Элементы РБД: главный ключ, имя, значение и тип поля;
* Назначение СУБД; режимы работы СУБД;
* Поиск информации в базе данных;
* Логические выражения в условиях поиска и удаления записей;
* Сортировка; ключи сортировки;
* Элементы проектирования РБД; нормализация данных;

Примерная программа по информатике и информационным технологиям

Тема «Информационные системы и базы данных»

|  |  |  |
| --- | --- | --- |
| ОСНОВНОЕ ОБЩЕЕ ОБРАЗОВАНИЯ | СРЕДНЕЕ (ПОЛНОЕ) ОБЩЕЕ ОБРАЗОВАНИЕ  Базовый уровень | СРЕДНЕЕ (ПОЛНОЕ) ОБЩЕЕ ОБРАЗОВАНИЕ  Профильный уровень |
| Табличные базы данных: основные понятия, типы данных, системы управления базами данных и принципы работы с ними.  Ввод и редактирование записей.  Условия поиска информации; логические значения, операции, выражения.  Поиск, удаление и сортировка данных. | Понятие и типы информационных систем. Базы данных (табличные, иерархические, сетевые). Системы управления базами данных (СУБД). Формы представления данных Реляционные базы данных. Знакомство с системой управления базами данных Access. Создание структуры табличной базы данных. Осуществление ввода и редактирования данных. Упорядочение данных в среде системы управления базами данных. Формирование запросов на поиск данных в среде системы управления базами данных. | Представление о системах управления базами данных, поисковых системах. Организация баз данных. Использование инструментов поисковых систем (формирование запросов) для работы с образовательными порталами. |

Практические работы

* Поиск записей в готовой базе данных;
* Сортировка записей в готовой базе данных;
* Создание однотабличной базы данных и работа с ней (создание запросов, форм и отчетов);

Центральные понятия этого раздела: *база данных, система* *управления базами данных и СУБД MS Access*.

Основные понятия:

* *Главный ключ* – это поле или совокупность полей, которое однозначно определяет запись в таблице или главный ключ – это идентификатор записи. В базе данных слово «ключ» имеет несколько употреблений: *ключ поиска* – поле, по значению которого ищется запись в БД, *ключ* *сортировки* – поле, по значению которого происходит упорядочение записей.
* *Структура таблицы* – это описание порядка следования полей, их типов, размеров.
* *Запись* – последовательность значений, описанных в структуре (строка в таблице).
* *Поле* – элемент записи, содержащий информацию об одном свойстве объекта (столбец таблицы) и характеризуется следующими параметрами:

- *имя поля* (записывается без пробелов – как имя переменной);

- *тип значения поля* (текстовый, числовой, логический, дата-время);

- *размер поля* (зависит от типа возможных значений);

- *значение поля* (соответствует конкретной записи);

* *Величина* – это отдельный информационный объект, имеющий собственное имя и занимающий место в памяти компьютера.

## **2.1.Методические рекомендации по изложению теоретического материала**

**1**.Изучение БД следует начать с обоснования актуальности данного приложения компьютерной техники. Задачу можно сформулировать следующим образом: имеется большой объем данных о какой – то реальной системе объектов, например, о книгах в библиотеке. Необходимо организовать хранение этой информации таким образом, что бы ее было удобно просматривать, пополнять, изменять, искать нужные сведения, осуществлять сортировку в любом порядке. В наше время решению описанных задач помогают компьютеры. Компьютерные информационные системы позволяют хранить большие объемы данных, осуществлять в них быстрый поиск, вносить изменения, выполнять всевозможные манипуляции с данными. Пример информационных систем, система продажи железнодорожных и авиационных билетов.

Основой всякой информационной системы являются база данных – организованная совокупность данных на магнитных дисках. Первый вывод относительно организации больших БД - требуется большой объем дисковой памяти.

**2**.Следующий вопрос – классификация баз данных. *По характеру хранимой* *информации* БД делиться на фактографическую (аналогия библиотечные карточки) и документальные. *По способу хранения данных* делит БД на централизованные (на одном компьютере) и распределенные. *По структуре* *организации данных*: реляционные, иерархические и сетевые.

**3**.В базовом курсе информатики рассматриваются лишь фактографические РБД. Теоретически доказано, что любая система данных может быть отражена с помощью таблиц. Простейшая РБД содержит одну таблицу, более сложная – из множества связанных таблиц. Структура таблицы: строки таблицы называются записями, столбцы – полями. Объяснения данного материала следует проводить на конкретном примере, база данных «Студенческий Трудовой Отряд». Назначение: с помощью базы данных сотрудники организации могут контролировать заказы различных магазинов. Так же могут узнать информацию о работниках организации (личные ,рабочие и медицинские) . Следует подчеркнуть, что в базах данных каждая таблица должна иметь свое имя.

Основные представления, которые должны быть закреплены учениками:

* Всякая таблица содержит в себе информацию о некоторой реальной системе и, следовательно, является ее информационной моделью;
* Всякая запись в таблице – информация о конкретном объекте данной системы;
* Значение поля в каждой записи – это определенная характеристика (свойство, атрибут) объекта.

**4**.Введение основных понятий, связанных с записями и полями: главный ключ записи, имя поля, значения поля, тип поля.

В этой теме ученики впервые встречаются с понятием *величины*. В дальнейшим это понятие будет широко использоваться в электронных таблицах, в алгоритмах и программах. Поля является величинами. Каждое поле в таблице имеет имя, для каждого поля определен тип. Понятие типа величины связано с тремя ее свойствами:

* Множеством значений, которые может принимать величина;
* Множеством операций, которые можно выполнять с этой величиной;
* Формой внутреннего представления в памяти ЭВМ.

В большинстве случаев в БД используются 4 основных типа: символьный, числовой, «дата» и логический. Поле символьного типа может хранить значения любой последовательности символов; числовые поля могут содержать целые или дробные десятичные числа; дата – день, месяц, год; логические поля – значения логических величин (да – нет, истина – ложь).

Чтобы не путать символьный и числовой тип, нужно придерживаться следующего правила: если поле обозначает порядковый номер или цифровой код, то ему нужно назначать символьный тип. Если же поле обозначает количество чего – то, то это поле должно иметь числовой тип. Пример, символьные: номер дома, номер телефона и т.д., числовой: возраст человека, расстояние, вес объекта.

В данном разделе впервые в курсе информатики ученики встречаются с логическим типом данных, с логическими величинами. Первое понятие о логической величине можно дать такое: это ответ на альтернативный вопрос. Например, на улице идет дождь? Ответ на такой вопрос может быть только «да» или «нет». Синонимами являются «истина», «ложь». Если поле таблицы будет принимать только такие значения, то ему назначается логический тип.

**5.**После знакомства с основными понятиями, относящимися к организации информации в РБД, следует перейти к изучению программного обеспечения, предназначенного для работы с БД. Такое программное обеспечение называется СУБД – система управления базами данных.

Для персональных компьютеров существует целый рад СУБД реляционного типа. В состав пакета Microsoft Office входят реляционная СУБД Access, все чаще используемая в школе. Существуют два способа работы пользователя с базой данных: работа с помощью прикладных программ, заранее составленных программистом в среде СУБД, и работа путем непосредственного взаимодействия с СУБД. В начальный период появления и распространения БД использовался только первый способ. Позже стали появляться СУБД, ориентированные на работу с пользователем. Для них были созданы удобные диалоговые оболочки, позволяющие пользователю легко реализовывать свои потребности в работе с БД: пополнять и изменять базу, осуществлять поиск данных по любым условиям. Наиболее совершенной СУБД такого типа является MS Access.

У всякой СУБД существует свой язык описания данных и язык манипулирования данными. Если пользователю приходится работать в режиме посимвольного ввода команд, то он должен знать все подробности синтаксиса командного языка. Диалоговая оболочка – более высокоуровневое средство интерфейса. Любая команда – это информация, управляющая выполнением определенного вида работы. Она должна содержать все необходимые данные для этого. Обычно это имя команды и набор параметров. Так как описание среды какой – либо конкретной СУБД не приводиться в учебниках, адаптацию к конкретной системе должен выполнять учитель.

Изучение конкретной СУБД следует проводить по стандартной методической схеме: «среда – режимы работы – система команд – данные».

Для СУБД Access различаются следующие основные режимы работы:

* Режим работы с таблицей: «Таблица»;
* Режим работы с запросами: «Запрос»;
* Режим работы с отчетами: «Отчет»;
* Режим работы с формами: «Форма».

(Режимы работы с макросами и модулями в базовом курсе не рассматриваются). В свою очередь, в каждом из отмеченных режимов есть подрежимы: «Просмотр», «Конструктор» и «Создать».

Например, «Таблица – Просмотр» - пользователь может просмотреть содержание таблицы, а так же отредактировать некоторые ее поля; «Таблица – Конструктор» - можно просмотреть описание структуры таблицы и внести в нее изменения. «Таблица – Создать» - описывается и создается структура новой таблицы.

**6**.Учебный материал ориентирован на два уровня изучения темы «Базы данных». Задача первого уровня: дать детям общие представление о базах данных, научить работать с готовой БД: осуществлять поиск информации; научить их создавать структуру и добавление записей; научить их создавать структуру и заполнять однотабличную базу данных. Дополнительная задача второго уровня: познакомить детей с основами проектирования БД.

Работа с СУБД начинается с запуска соответствующей программы, поэтому необходимо показать, где хранится программа и как ее запустить на исполнение. Первое понятие, которое должны усвоить ученики: база данных храниться в файле; чтобы начать с ней работать, необходимо открыть файл с БД. Затем учитель должен показать, как можно просмотреть на экране записи таблицы. Для этого должна быть заранее подготовлена демонстрационная БД.

Основная задача любой информационной системы – поиск информации в базе данных. Поиск происходит по запросу пользователя. В результате выполнения запроса получается таблица, состоящая из полей, указанных в команде. В эту таблицу включается информация из тех записей, которые удовлетворяют условию поиска представляющие собой логическое выражение. Логическое выражение – это некоторое высказывание по поводу значений полей базы данных; это высказывание по отношению к разным записям может быть истинным или ложным.

В СУБД Access для создания запросов используется конструктор запросов. В школьном курсе создаются запросы на поиск данных, на сортировку записей, где основными понятиями, которые должны усвоить ученики, являются «ключ сортировки» и «порядок сортировки». Ключ сортировки – это поле, по значению которого происходит упорядочение записей в таблице. Порядок сортировки имеет два варианта: по возрастанию значений ключа и по убыванию значений.

**7.**При углубленном изучении раздела БД ученики знакомятся с элементами проектирования БД. Изучается три этапа проектирования: концептуальное, логическое и физическое проектирование.

## **2.2. Организация практической работы**

Практическая работа делиться три этапа:

***1.Задачи*:** теоретические задания для закрепления основных понятий «главный ключ», «имя поля», «тип поля».

*Задания первого типа*: дано имя таблицы и перечень полей, требуется указать главный ключ и определить типы всех полей.

* ПОГОДА (ДЕНЬ, ОСАДКИ, ТЕМПЕРАТУРА, ДАВЛЕНИЕ, ВЛАЖНОСТЬ)

Перед скобками записано имя таблицы, в скобках через запятую перечислены имена полей. Ключевое поле подчеркивается.

Имя таблицы ПОГОДА, ключевое поле ДЕНЬ имеет тип «дата», ОСАДКИ – символьный тип, ТЕМПЕРАТУРА, ДАВЛЕНИЕ, ВЛАЖНОСТЬ – числовой тип.

* УСПЕВАЕМОСТЬ (УЧЕНИК, РУССКИЙ, АЛГЕБРА, ХИМИЯ, ФИЗИКА)

Поле УЧЕНИК – символьного типа, все остальные – числового. Оценкам имеет смысл определить числовой тип, так как в запросах могут использоваться вычисления, например, поиск по среднему баллу.

*Задания второго типа*: определена предметная область базы данных; требуется озаглавить таблицу, определить имена полей и их типы, назначить главный ключ. Эта задача имеет отношение к области проектирования БД и уже по этой причине сложная. Поэтому требовать полноты выполнения задания не следует. Достаточно, чтобы они указали несколько полей, имеющих отношения к данной теме, и правильно определили типы полей.

|  |  |
| --- | --- |
| **Страны мира** | **кинофильмы** |
| СТРАНА - символьный | НАЗВАНИЕ - символьный |
| СТОЛИЦА - символьный | СТРАНА -символьный |
| ПЛОЩАДЬ - числовой | ЖАНР - символьный |
| НАСЕЛЕНИЕ - числовой | РЕЖИССЕР - символьный |
| ПОСЕЩАЛ - логический | ДУБЛИРОВАН - логический |

Задачи могут быть предметом коллективного разбора на уроке, использоваться в контрольных и домашних заданиях.

***2.Упражнения****:* практические задания для работы в среде СУБД с целью отработки отдельных навыков. Упражнения выполняются на компьютере.

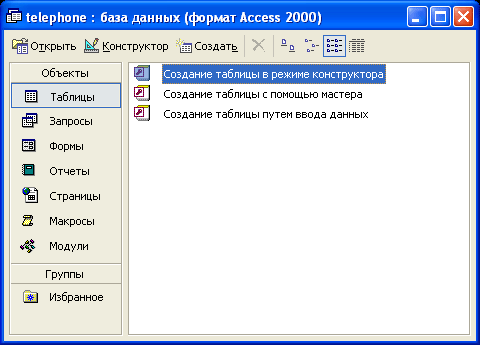
Примеры практических заданий

**1.Работа Создание базовых таблиц**

**Цель работы:** научиться создавать базовые таблицы в MS Access, добавлять новые поля в таблицы.

**Создание баз данных.** Запустите Microsoft Access

Перед вами откроется окно "База данных". Это окно является исходным элементом управления Microsoft Access.



На левой панели окна "База данных" сосредоточены элементы управления для вызова всех типов объектов Access (таблицы, запросы, формы и т.д.). На правой - элементы управления для создания новых объектов. Сюда же добавляются и создаваемые объекты. Окно имеет панель инструментов для осуществления основных операций с объектами (открытие, редактирование, удаление и т.п.).

**Запустите конструктор создания таблиц**, сделав двойной щелчок по соответствующему элементу управления или нажав кнопку "Конструктор" на панели инструментов. Перед вами откроется окно проектирования структуры таблицы.

**Задайте структуру основной таблицы проектируемой базы данных**. Будем считать, что все данные в нашей базе - текстовые. Задайте имена полей, их описания и длины, согласно приведенной ниже таблице:

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Описание** | **Длина** |
| Номер | Номер телефона | 12 |
| Имя | Имя абонента | 25 |
| Адрес | Адрес абонента | 40 |
| Код\_категории | Категория абонента | 2 |

**Укажите наличие ключевого поля** (в нашем примере поле "Номер"). Это можно сделать через контекстное меню (нажав правой кнопкой мыши на строке соответствующего поля и выбрав пункт "Ключевое поле" выпавшего меню), дав команду меню "Правка > Ключевое поле" или нажав на кнопку панели инструментов Microsoft Access.

**Сохраните в текущей базе данных созданную таблицу.** Сделать это можно с помощью команды меню "Файл > Сохранить" или с помощью кнопки панели инструментов Microsoft Access. Укажите имя сохраняемой таблицы: "ТЕЛЕФОНЫ". Закройте окно проектирования структуры таблицы и обратите внимание на то, что в окне "База данных" появился новый элемент - только что созданная таблица "ТЕЛЕФОНЫ".

**2.Создание запросов и работа с ними**

**Цель работы:** научиться создавать запросы к БД в MS Access, изменять их структуру, редактировать и добавлять новые критерии отбора.

**Откройте** окно "База данных" и **переключитесь** на вкладку объектов "Запросы". **Выберите** создание запроса в режиме конструктора. Вам будет предложено добавить таблицы. Сделайте это и закройте окно добавления таблиц. Перед вами откроется бланк запроса по образцу.

**Перетащите поля** «Имя», «Номер» и «Код\_категории» в формируемую таблицу. Укажите также параметр «Сортировка» (по возрастанию) для поля «Код\_категории». Это обеспечит группировку отображаемых записей по категориям. **Сохраните** составленный запрос (назовите, например, "Телефоны друзей") и **закройте** бланк запросов по образцу. Чтобы посмотреть результаты работы запроса, **откройте** его, сделав двойной щелчок по соответствующей записи в окне «База данных».

**3.Работа с Мастером форм в MS Access**

**Цель работы:** научиться создавать формы для ввода и просмотра данных в MS Access, изменять структуру, редактировать формы, добавлять новые поля.

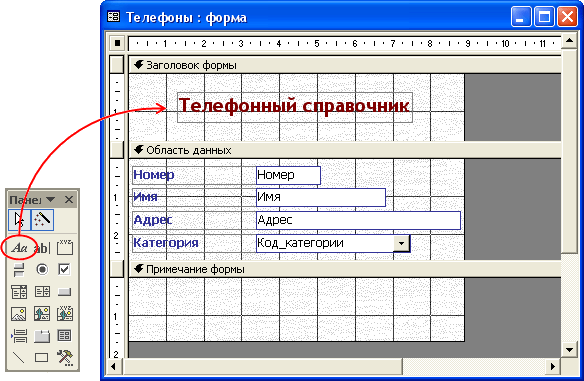
Создадим форму, с помощью которой будет удобно вводить новые записи в телефонный справочник. **Откройте** окно "База данных" и **переключитесь** на вкладку объектов "Формы". **Выберите** создание формы с помощью мастера. Перед вами откроется диалоговое окно, в котором будет необходимо ответить на ряд вопросов. **Укажите** следующие параметры создаваемой формы (на каждом шаге нажимайте кнопку "Далее"):

Форма строится на основе таблицы "ТЕЛЕФОНЫ". В форму необходимо включить все поля таблицы.

* Внешний вид формы - "в один столбец".
* Требуемый стиль - по вашему усмотрению.
* Имя формы - "Телефоны"

После выполнения работы мастера, перед вами сразу откроется созданная форма, с которой уже можно работать. Внесем, однако, в макет формы некоторые изменения. Добавим заголовок формы и примечания.

**Закройте** созданную форму и **откройте** ее в режиме конструктора. Перед вами откроется *макет формы*, а также *панель элементов*, содержащая заготовки и инструменты для создания элементов управления формы.



**Раздвиньте** с помощью мыши область заголовка формы, **добавьте** в заголовок элемент "Надпись", **введите** туда текст "Телефонный справочник", **укажите** желаемые параметры текста (шрифт, размер, цвет и т.п.). Аналогичным образом оформите и примечание формы. Введите туда свое имя (как автора базы данных), год создания базы данных или аналогичную информацию.

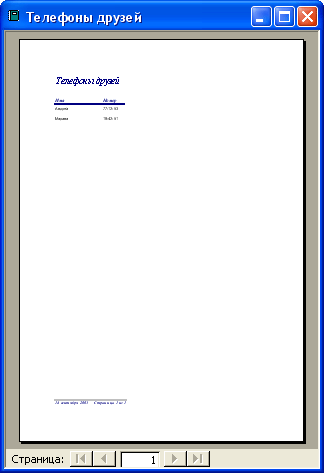
**Сохраните** и **закройте** макет формы. Запустите форму.

**4.Создание и работа с отчетами. Экспортирование**

**Цель работы:** научиться создавать отчеты, изменять структуру, редактировать отчеты. Ознакомиться с возможностями экспортирования таблиц.

Создадим список телефонов друзей в виде отчета. В окне "База данных" и **переключитесь** на вкладку объектов "Отчеты". **Выберите** создание отчета с помощью мастера. **Укажите** следующие параметры создаваемого отчета:

* Отчет строится на базе запроса "Телефоны друзей". Необходимо выбрать все доступные поля.
* Уровни группировки - не добавлять.
* Порядок сортировки - по имени.
* Макет для отчета - "табличный".
* Стиль отчета - по вашему усмотрению.
* Имя отчета - "Телефоны друзей".



Если есть необходимость внести какие-либо изменения в созданный отчет (например, скорректировать заголовок), то откройте отчет в режиме конструктора и сделайте это. Обратите также внимание на подпись документа, расположенную внизу страницы.

***3.Индивидуальные работы***: зачетные задания, требующие от учеников комплексного владения теоретическими знаниями и практическими навыками. Варианты индивидуальных работ учитель распределяет среди учеников по своему усмотрению. Возможный вариант, когда все делают одно и то же задание. В таком случае учителю легче осуществлять контроль, оказывать помощь ученикам.

Пример работы: **Поиск информации в геоинформационной системе**.

**Цель работы**: освоение приемов поиска информации в геоинформационной системе на примере ГИС «Карта Москвы».

Задание 1. Осуществить поиск приведенных ниже объектов по указанному адресу, найти ближайшую станцию метро, отметить на карте место, соответствующее заданному адресу.

Музеи, находящиеся в радиусе 1 км от указанного адреса:

|  |  |
| --- | --- |
| **Вариант** | **Адрес** |
| 1 | Петровка,17 |
| 2 | Арбат,12 |
| 3 | Кузнецкий мост,17 |

Банки, находящиеся в радиусе 1 км от указанного адреса:

|  |  |
| --- | --- |
| **Вариант** | **Адрес** |
| 4 | Покровский бульвар,13 |
| 5 | Неглинная,16 |
| 6 | Цветной бульвар,11 |

Задание 2. Найти адреса указанных вузов.

|  |  |
| --- | --- |
| **Вариант** | **Высшее учебное заведение** |
| 1 | Академия реставрации |
| 2 | Высшая школа экономики |
| 3 | Высший институт управления |
| 4 | Институт мировой экономики |
| 5 | Московская академия экономики и права |

**Программное обеспечение для изучения данной темы**:

* ОС : MS Windows;
* Прикладные программы: СУБД MS Access;
* электронные учебные комплексы: готовая демонстрационная база данных «СТО»;
* Отдельные программы для контроля и обучения : презентации созданные учителем.

## **2.3. Сравнительный анализ содержания в различных учебных пособиях**

Учебник «Информатика и ИКТ» для 9 класса, автор Угринович Н.Д.

Данная тема изучается на теоретическом уровне с готовой БД.

|  |  |  |  |
| --- | --- | --- | --- |
| Разделы стандарта | Главы, параграфы и пункты учебника | Практические работы | Кол-во часов |
| **Базы данных.** Поиск данных в готовой базе. Создание записей в базе данных | **Глава 3. Кодирование и обработка числовой информации**  3.4. Базы данных в электронных таблицах  3.4.1. Представление базы данных в виде таблицы и формы  3.4.2. Сортировка и поиск данных в электронных таблицах | Практическая работа3.5. Сортировка и поиск данных в электронных таблицах | 6 |

Учебник «Информатика и ИКТ. Базовый уровень» для 11 класса

Автор Угринович Н. Д.

Данная тема изучается на практическом и теоретическом уровне. Создание БД.

|  |  |  |  |
| --- | --- | --- | --- |
| Разделы стандарта | Главы, параграфы и пункты учебника | Практические работы | Кол-во часов |
| **Средства и технологии создания и преобразования информационных объектов**  Базы данных. Системы управления базами данных. Создание, ведение и использование баз данных при решении учебных и практических задач. | **Глава 3. Базы данных. Системы управления базами данных (СУБД)**  3.1. Табличные базы данных  3.2. Система управления базами данных  3.2.1. Основные объекты СУБД: таблицы, формы, запросы, отчеты  3.2.2. Использование формы для просмотра и редактирования записей в табличной базе данных  3.2.3. Поиск записей в табличной базе данных с помощью фильтров и запросов  3.2.4. Сортировка записей в табличной базе данных  3.2.5. Печать данных с помощью отчетов  3.3. Иерархические базы данных  3.4. Сетевые базы данных. | 3.1. Создание табличной базы данных  3.2. Создание формы в табличной базе данных  3.3. Поиск записей в табличной базе данных с помощью фильтров и запросов  3.4. Сортировка записей в табличной базе данных  3.5. Создание отчета в табличной базе данных  3.6.Создание генеалогического древа семьи | 8 |

Учебник «Информатика и ИКТ» для 9 класса

Автор И.Г. Семакин.

Данная тема изучается на практическом и теоретическом уровне. Работа с готовой БД.

|  |  |  |
| --- | --- | --- |
| Главы, параграфы и пункты учебника | Практические работы | Кол-во часов |
| Глава 3. Хранение и обработка информации в БД.  10. основные понятия.  11. что такое система управление БД.  12. создание и заполнение баз данных.  13. условие выбора и простые логические выражения  14. условие выбора и сложные логические выражения.  15. сортировка, удаление и добавление записей. | №7. Знакомство с СУБД. Создание и редактирование базы данных.  №8. Формирование запросов на выборку, замену и удаление записей с использованием логических операций.  №9. Формирование отчетов и форм. | 12 (6+6) |

Учебник «Информатика и ИКТ» базовый уровень для 11 класса

Автор И.Г. Семакин.

Данная тема изучается на практическом и теоретическом уровне. Создание и заполнение БД.

|  |  |  |
| --- | --- | --- |
| Главы, параграфы и пункты учебника | Практические работы | Кол-во часов |
| Глава 1. Информационные системы и базы данных.  1.1.Понятие информационной системы, классификация ИС.  1.2.локальные компьютерные сети.  1.3.основные понятия баз данных.  1.4.СУБД MS Access.  1.5.анализ предметной области.  1.6.анализ данных.  1.7.построение модели данных.  1.8. создание базы данных  1.9.запросы к базе данных.  1.10.конструктор запросов.  1.11.логические выражения и условия отбора.  1.12. ввод данных через форму.  1.13.запросы к полной базе данных. Удаление записей. Вычисляемые поля.  1.14реализация выборки, удаление вычисляемых полей в конструкторе запросов.  1.15.этап создания отчета в базе данных.  1.16.создание отчетов в MS Access.  1.17.геоинформационные системы. | №1. Обмен информацией в локальной сети.  №2. Знакомство с СУБД MS Access.  №3. Создание структуры и заполнения базы данных.  №4. Проектирование и создание базы данных.  №5. Реализация простых запросов на выборку.  №6. Ввод данных через форму.  №7. Реализация сложных запросов.  №8. Творческое задание на реализацию сложных запросов.  №9.Формирование отчетов в MS Access.  №10. Поиск информации в геоинформационной системе. | 18 |

Учебник «Информатика и ИКТ. Профильный уровень» для 11 класса

Автор Угринович Н. Д.

|  |  |
| --- | --- |
| Теория | Компьютерный практикум и решение задач |
| Тема 6. Технологии хранения, поиска и сортировки информации (СУБД) – 16 часов | |
| Представление осистемах управления базами данных, поисковых системах в компьютерных сетях, библиотечных информационных системах. Компьютерные архивы информации: электронные каталоги, базы данных. Организация баз данных. Примеры баз данных: юридические, библиотечные, здравоохранения, налоговые, социальные, кадровые. Использование инструментов системы управления базами данных для формирования примера базы данных учащихся в школе.  Использование инструментов поисковых систем (формирование запросов) для работы с образовательными порталами и электронными каталогами библиотек, музеев, книгоиздания, СМИ в рамках учебных заданий из различных предметных областей. Правила цитирования источников информации. | Система управления базами данных.  Создание структуры табличной базы данных.  Ввод и редактирование данных.  Поиск и сортировка данных.  Создание реляционных баз данных. |

Учебнику «Информационные системы и модели» элективный курс.

Автор И.Г. Семакин.

Раздел «Моделирование и разработка информационных систем» (35 ч)

*Информационные системы и системология* (9 ч). Понятие ИС; этапы разработки информационных систем. Основные понятия системологии: система, структура. Модели систем: модель черного ящика, модель состава, структурная модель. Графы, сети, деревья. Информационно – логическая модель предметной области.

*Реляционная модель данных и реляционная база данных* (14). Проектирование многотабличной базы данных. Понятие о нормализации данных. Типы связей между таблицами. Создание базы данных в среде реляционной СУБД (MS Access). Реализация приложений: запросы, отчеты.

*Базы данных на электронных таблицах* (6 ч). Создание базы данных в среде табличного процессора. Использование формы для ввода и просмотра списка, для выборки данных по критериям. Сортировка данных по одному или нескольким полям. Фильтрация данных. Сводные таблицы.

*Программирование приложений* (5 ч). Макросы: назначение, способы создания и использования. Структура программы на VBA. Разработка пользовательского интерфейса: диалоговые окна. Введение в программирование на VBA.

**Разработка урока для 9 класса по учебнику И.Г. Семакин.**

**Тема урока**: Создание и заполнение баз данных.

**Тип урока**: Урок формирования знаний.

**Цель познавательная**: Формирование знаний по созданию и заполнению баз данных в СУБД Access.

**Цель воспитательная**: формирование мировоззрения учащихся, организованности, собранности; воспитание активной жизненной позиции, сотрудничества с учащимися, внимания, самостоятельности, работоспособности.

**Цель развивающая**: формирование интеллектуальных умений и общеучебных навыков, формирование умения осуществлять самоконтроль.

**Формы организации познавательной деятельности**: индивидуальная и фронтальная.

**Методы обучения**: частично-поисковый и репродуктивный

**Диагностируемые цели урока**

После урока ученик:

**Знает**

* Типы и форматы полей;
* Создание новой БД;
* Заполнение БД информацией.

**Умеет**

* Создавать структуры таблицы базы данных;
* Редактировать структуры таблицы базы данных (добавление и удаление полей, изменение имени, типа и ширины поля);
* Определить первичный ключ таблицы;
* Добавлять, удалять и редактировать записи базы данных.

**Углубляет** знания по созданию и обработки БД

**Развивает** интеллектуальные умения.

**Макроструктура урока**:

* Организационный этап.
* Постановка цели урока.
* Актуализация знаний.
* Введение новых знаний.
* Этап формирования умений и навыков.
* Подведение итогов урока.
* Выдача домашнего задания и инструктаж по его выполнению.

**К уроку**: изображение на интерактивной доске рабочий области MS Access.

На каждом рабочем компьютере размещена таблица « Абитуриенты».

1. **Организационный момент**

**Цель**: мотивация.

Приветствие. Уточнение отсутствующих и выявление причины. Предложить открыть тетради и записать число и тему урока.

Постановка цели урока

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| Предложить записать дату и тему урока:  Создание и заполнение баз данных | Делают записи, слушают | Записи в тетради. Активизация внимания учащихся |

1. **Актуализация знаний**

**Форма организации познавательной деятельности**: фронтальная.

**Метод обучение**: репродуктивный.

**Проектирование оценок:**1 оценка за правильный ответ

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| Так как целью сегодняшнего урока является формирований знаний по созданию и редактированию БД, то нам нужно повторить  Структуру таблицы: запись и поле,  основные типы полей и способ задания первичного ключа БД. | Слушают, отвечают на вопросы, дополняют друг друга. | На этом этапе у учеников происходит осознания основных понятий БД. |

1. **Введение новых знаний.**

**Форма организации познавательной деятельности**: фронтальная.

**Метод обучение**: репродуктивный.

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| 1.Обозначение проблемы  Для того чтобы создать таблицу БД нам нужно определить типы полей и форматы полей.  (учитель объясняет на таблице погода , которая изображена на интерактивной доске).  Типы полей: символьный (в таблице этому типу соответствует ОСАДКИ), числовой(ТЕМПЕРАТУРА,  ДАВЛЕНИЕ, ВЛАЖНОСТЬ), логический и «дата»(ДЕНЬ).  Формат числового поля обычно состоит из двух частей: длины и точности. длина – это полное количество символьных позиций, выделяемых под запись числа, точность – это количество позиций, выделяемых под дробную часть. Формат логической величины стандартный – 1 символ.  2.Создание базы данных начинается с описания структуры таблицы. По команде  .создать «имя файла»  Следующий этап задать первичный ключ в нашей таблице это ДЕНЬ.  3.Ввод данных производится по команде  .добавить запись  Ввод может происходить через форму, учитывающую структуру записей таблицы, которая была описана на этапе создания. Например, ввод первой записи через форму в таблицу «Погода» будет происходить в таком виде (на доске показана соответствующее действие и результат).  Так же мы можем изменить значения полей, изменить форматы полей, удалить одни поля и добавить другие. | Слушают, записывают в рабочею тетрадь главное, смотрят на интерактивной доске ход работы. Задают вопросы. | Сформированность последовательности действий, приводящих к решению проблемы. |

Таблица изображенная на интерактивной доске. Структура таблицы «Погода».

|  |  |  |  |
| --- | --- | --- | --- |
| **поле** | **тип** | **длина** | **точность** |
| ДЕНЬ | Дата | 8 |  |
| ОСАДКИ | Символьный | 11 |  |
| ТЕМПЕРАТУРА | Числовой | 5 | 1 |
| ДАВЛЕНИЕ | Числовой | 3 | 0 |
| ВЛАЖНОСТЬ | числовой | 3 | 0 |

Результат ввода первой записи через форму в таблицу (таблица изображена на интерактивной доске)

|  |  |
| --- | --- |
| ДЕНЬ | 15/03/10 |
| ОСАДКИ | Снег |
| ТЕМПЕРАТУРА | -3,5 |
| ДАВЛЕНИЕ | 746 |
| ВЛАЖНОСТЬ | 67 |

1. **Этап формирования умений и навыков**.

**Форма организации познавательной деятельности**: индивидуальная

**Метод обучение**: частично-поисковый.

Учащиеся присаживаются за свои рабочие места.

**Проектирование оценок**: оценить работу всех учащихся по следующему критерию: правильность выполнения задания.

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| Постановка задач:  1.отрыть БД.  2.открыть таблицу «Абитуриенты».  3. выполнить задания:   1. **Добавить** в таблицу две новых записи о следующих абитуриентах:   а) Авдеева Ирина Юрьевна, дата рождения - 17.03.1982, пол - ж, школа - 6, подготовительные курсы – нет, математика – 4, информатика – 5, русский язык – 4.  б) Пирогов Анатолий Андреевич, дата рождения - 6.06.1981, пол - м, школа - 63, подготовительные курсы – да, математика – 5, информатика – 4, русский язык – 5.  4. **Заменить** у абитуриентки **Захаровой** оценку по русскому языку на «четвёртку».  5. **Заменить** у абитуриента **Морозова** номер школы на **31**.  6. **Удалить** записи, содержащие сведения об абитуриентах, не сдававших экзамен по **русскому языку**. | Выполняют задания. Получают результат. | Правильность применения нового знания к своей индивидуальной работе. |

1. **Подведение итогов урока**

**Форма организации познавательной деятельности**: фронтальная.

**Метод обучение**: репродуктивный.

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| 1.Фронтальный опрос учащихся по новому материалу. Предлагается следующая система:  - какие основные типы поле используются в БД?  - что определяется форматом для разных типов полей?  - как происходит заполнение таблицы?  2.простовление оценок за работу на уроке. | Отвечают на вопросы | Систематизация знаний по созданию и заполнению таблицы БД. |

1. **Выдача домашнего задания**

|  |  |  |
| --- | --- | --- |
| Педагогическая деятельность учителя | Учебная деятельность ученика | Предполагаемый результат |
| Выдать задания на дом:  1.прочитать параграф 12  2.ответить на вопросы после параграфа 12 | Записывают задания на дом | Закрепление знаний, формирование умений и навыков. |

**Требования к знаниям и умениям учащихся по теме «Информационные системы и Базы данных»**

*Учащиеся должны знать/понимать*:

* Что такое информационная система;
* Что такое база данных, СУБД;
* Что такое реляционная база данных, ее элементы (записи, поля, ключи); типы и форматы полей;
* Структура команд поиска и сортировки информации в базах данных;
* Что такое логическая величина, логическое выражение;
* Что такое логические операции, как они выполняются;

При углубленном изучении:

* Основные этапы проектирования баз данных.

*Учащиеся должны уметь:*

* Открывать готовую БД в одной из СУБД реляционного типа;
* Организовывать поиск информации;
* Редактировать содержимое полей БД;
* Сортировать записи в БД по ключу;
* Добавлять и удалять записи в БД;
* Создавать и заполнять однотабличную БД в среде СУБД;

При углубленном изучении:

* Осуществлять концептуальное, логическое и физическое проектирование.

*использовать приобретенные знания и умения в практической деятельности и повседневной жизни для*:

* Поиска и отбора информации;
* Создание собственных баз данных.

## **Заключение**

В моей работе были рассмотрены основные этапы проектирования, создания и обработки реляционных баз данных. После проведения анализа методики изучения темы «Информационные системы и базы данных» в школьном курсе информатики можно сделать выводы:

* на базовом уровне тема «Информационные системы и базы данных» раскрывается на теоретическом уровне, не затрагивая вопросы об этапах проектирования баз данных;
* на профильном уровне (элективные курсы) раскрываются основные этапы проектирования и обработки реляционных баз данных;
* при изучении информационных систем и БД школьники получают представление о сущности информационных процессов, рассматривают примеры передачи, хранения и обработки информации в деятельности человека, живой природе и технике, классификацию информации, выделяют общее и особенное, устанавливают связи, сравнивают, проводят аналогии и т.д. Это помогает им осмысленно видеть окружающий мир, более успешно в нем ориентироваться, формировать основы научного мировоззрения;
* учащиеся после изучения данной темы на базовом уровне не имеют достаточных знаний для создания собственных баз данных.

## **Литература**

1. Лапчик М.П. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов / М.П.Лапчик, И.Г.Семакин, Е.К.Хеннер; под общей ред. М.П.Лапчик. – М.: Издательский центр «Академия», 2003.
2. Избачков Ю.С., Петров В.Н. Информационные системы: Учебник для вузов. 2 – е изд. – СПб.: Питер, 2005.
3. Основы обработки реляционных баз данных средствами СУБД MS Access: Практикум по информатике / Авт. – сост. И.В.Панова – Н.Новгород: НГПУ, 2009.
4. Семакин И.Г. Информатика. 11 – й класс / И.Г.Семакин, Е.К.Хеннер. - 2 –е изд. – М.: БИНОМ. Лаборатория знаний, 2005.
5. Семакин И.Г. Информатика и информационно – коммуникационные технологии. Базовый курс: Учебник для 9 класса / И.Г.Семакин, Л.А.Залогова, С.В.Русаков, Л.В.Шестакова. – М.: БИНОМ. Лаборатория знаний, 2005.
6. Семакин И.Г. Преподавание базового курса информатики в средней школе: Методическое пособие / И.Г.Семакин, Т.Ю.Шеина. – 2 – е изд., испр. и доп. – М.: Бином. Лаборатория знаний, 2004.
7. Программы для общеобразовательных учреждений: Информатика. 2 – 11 классы. – 2 – е изд., испр. и доп. – М.: БИНОМ. Лаборатория знаний, 2005.
8. Федеральный компонент государственного образовательного стандарта общего образования по информатике и информационным технологиям// (приказ Министерства образования РФ № 1089 от 05.03.04), 2004, стр. 2 – 35.