# ПЛАН-КОНСПЕКТ УРОКА

|  |  |  |
| --- | --- | --- |
| 1 | **ФИО (полностью)** | Моисеева Елена Николаевна |
| 2 | **Место работы** | МБОУ «Средняя школа №1» |
| 3 | **Должность** | учитель информатики |
| 4 | **Предмет** | Информатика и ИКТ |
| 5 | **Класс** | 11 класс |
| 6 | **Тема и номер урока в теме** | Технологии разработки и использования информационных систем. Проектирование многотабличной базы данных. Создание БД. 5 и 6 уроки. |
| 7 | **Базовый учебник** | Угринович Н.Д. Информатика и ИКТ: учебник для 11 класса - М.: БИНОМ, Лаборатория знаний, 2010.-295с.: илл. |

**8. Цель урока:**

***Обучающая***: формировать у учащихся знаний об основных объектах базы данных MS Access, освоить новые возможности СУБД MSAccess, приблизить овладение СУБД MSAccess к профессиональному уровню

***Развивающая***: развитие умения анализировать, сопоставлять, сравнивать, выделять главное, приобрести опыт использования теоретических знаний и средств СУБД MSAccess в реализации конкретной задачи

***Воспитывающая***: понимать значимость хранения и структурирования информации

**9. Тип урока**: урок изучения нового материала, комбинированный урок.

**10. Формы работы учащихся:** фронтальная, индивидуальная.

**11. Необходимое техническое оборудование:** компьютерный класс с автоматизированным рабочим местом учителя с мультимедийным проектором и автоматизированными рабочими местами ученика с выходом в интернет.

**12. Структура и ход урока:**

***Таблица 1.***

**СТРУКТУРА И ХОД УРОКА**

| **№** | **Этап урока** | **Название используемых ЭОР**  *(с указанием порядкового номера)* | **Деятельность учителя**  *(с указанием действий с ЭОР, например, демонстрация)* | **Деятельность ученика** | **Время**  *(мин.)* | **УУД** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **5** | **6** | **7** | **8** |
| 1 | Организаци-онный момент. |  | Проверяет готовность учащихся к уроку | Готовятся к уроку, сообщают об отсутствующих | 1 | Вступать в диалог |
| 2 | Мотивация учащихся | Презентация  «Проектирование базы данных» | Учитель объявляет тему урока | Учащиеся воспринимают информацию | 1 | Формирова-ние учебной мотивации |
| 3 | Фронталь-ный опрос | Презентация  «Проектирование базы данных» или тест | Формулирует тест учащимся на компьютере или на карточках, помогает выполнять задание | Выполняют тест на компьютере или на карточках. | 8 | Формирование умения анализа и синтеза |
| 4 | Изучение нового материала | Презентация  «Проектирование базы данных» | Объясняет новый материал, вовлекая учащихся в обсуждение, демонстрирует слайды в нужной последовательности Отвечает на вопросы учеников. | Воспринимают звуковую и графическую информацию, включаясь в обсуждение, задают вопросы учителю. | 17 | Строить логическую цепь размышле-ний |
| 5 | Практичес-кая работа | Работа с MS Office Access | Объясняет новый материал, вовлекая учащихся в проектную деятельность, демонстрирует этапы работы в нужной последовательности Отвечает на вопросы учеников. | Воспринимают звуковую и графическую информацию, включаясь в проектную деятельность, задают вопросы учителю. | 50 | Строить логическую цепь размышле-ний в проектной деятельности |
| 6 | Рефлексия |  | Задает вопросы учащимся | Отвечают на вопросы учителя | 2 | Построение логической цепи размышле-ний |
| 7 | Домашнее задание |  | Даёт пояснения по выполнению домашнего задания | Слушают учителя, по необходимости делают записи | 1 | Осмысленно читать, извлекая нужную информацию, отбрасывая второстепен-ную информацию. |

**Ход урока:**

**Основные понятия**:

* Информационная система
* Жизненный цикл ИС
* Этапы проектирования БД
* Проектирование БД

1. Организационный момент.
2. Сообщение темы и целей урока.
3. Фронтальный опрос.

*Вопрос 1. Дайте определение базы данных*

**Информационные структуры, содержащие взаимосвязанные данные о реальных объектах и хранящиеся во внешней памяти, называются базами данных.**

В широком смысле слова можно сказать, что **база данных** – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области или разделе предметной области.

*Вопрос 2. Назовите отличительные признаки БД:*

**База данных хранится и обрабатывается в вычислительной системе. Таким образом, любые внекомпьютерные хранилища информации (архивы, библиотеки, картотеки и т. п.) базами данных не являются.**

Данные в базе данных логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.

*Вопрос 3. На какие группы делятся БД? Дайте им определение.*

**Фактографические БД** содержат короткие сведения об объектах, поданные в точно определенном формате, например, Автор, название, год издания …

В **документальных БД** содержится информация разного типа: текстовая, звуковая, графическая, мультимедийная. Например, БД современной музыки может содержать тексты и ноты песен, фотографии авторов, звуковые записи, видеоклипы.

*Вопрос 4. Что такое информационно-поисковая система?*

Совокупность базы данных и обслуживающих программ называется **информационно-поисковой системой (ИПС).** ИПС обычно имеет конкретное предметное назначение, например: ИПС книжного фонда публичной библиотеки; ИПС кадровой информации крупного предприятия и т.п.

*Вопрос 5. СУБД - …*

**СУБД** – Это программное обеспечение (ПО), которое позволяет создавать БД, обновлять и дополнять информацию, обеспечивать гибкий доступ к информации

*Вопрос 6. Назовите требования к СУБД*

* + - **Возможность манипулирования данными.**
    - **Возможность поиска и формирование запросов.**
    - **Обеспечение целостности (согласованности) данных.**
    - **Обеспечение защиты и секретность.**

*Вопрос 7. Дайте классификацию БД по модели данных*

* + - **Иерархические**
    - **Сетевые**
    - **Реляционные**

Дайте характеристику модели, приведите примеры.

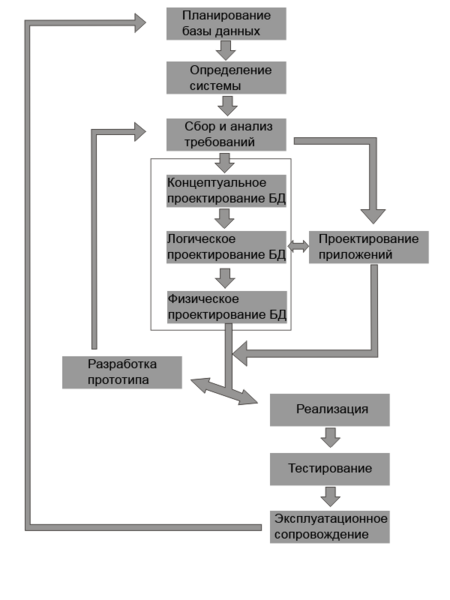
1. Объяснение нового материала

**Обзор жизненного цикла информационных систем**

Начиная с 1970-х годов системы баз данных стали постепенно заменять файловые системы, использовавшиеся как часть инфраструктуры информационных систем (Information System - IS) организаций. Параллельно с этим росло признание того факта, что данные являются важным корпоративным ресурсом, к которому нужно относиться так же бережно, как и к другим ресурсам организации. Это привело к тому, что во многих организациях появились целые отделы или функциональные подразделения, занимавшиеся администрированием данных (АД) и администрированием баз данных (АБД). Они отвечали за обработку и управление корпоративными данными и корпоративными базами данных.

База данных является фундаментальным компонентом информационной системы, а ее разработку и использование следует рассматривать с точки зрения самых широких требований организации. Следовательно, жизненный цикл информационной системы организации неотъемлемым образом связан с жизненным циклом системы базы данных, поддерживающей ее функционирование. Жизненный цикл информационной системы обычно состоит из нескольких этапов: планирование, сбор и анализ требований, проектирование, создание прототипа, реализация, тестирование, преобразование данных и сопровождение.

Все этапы жизненного цикла информационной системы здесь рассматриваются с точки зрения разработки приложения баз данных. Однако следует отметить, что разработку любого приложения базы данных всегда полезно рассматривать с более широкой точки зрения - как разработку определенного компонента всей информационной системы организации в целом.



***Жизненный цикл приложения баз данных***

Как уже упоминалось выше, система базы данных является фундаментальным компонентом более широкого понятия - информационной системы организации. Следовательно, жизненный цикл приложений баз данных неразрывно связан с жизненным циклом информационной системы. Этапы жизненного цикла приложения базы данных показаны на рисунке. Следует признать, что эти этапы не являются строго последовательными, а предусматривают в некоторых случаях возврат к предыдущим этапам с помощью обратных связей (feedback loops). Например, при проектировании базы данных могут возникнуть проблемы, для разрешения которых потребуется вернуться к этапу сбора и анализа требований. Обратные связи могут возникать почти между всеми этапами, но на рисунке показаны только наиболее важные из них. Основные сведения о наиболее важных мероприятиях, связанных с реализацией каждого этапа жизненного цикла приложения базы данных, приведены в таблице.

Для малых приложений с небольшим количеством пользователей жизненный цикл может оказаться не очень сложным. Однако он может стать чрезвычайно сложным при проектировании среднего или крупного приложения базы данных, с десятками и даже тысячами пользователей, сотнями запросов и прикладных программ. Ниже в этой главе основное внимание уделяется тем действиям, которые связаны с разработкой средних и крупных приложений баз данных. В следующих разделах более подробно рассматриваются основные мероприятия, связанные с осуществлением каждого этапа жизненного цикла приложения базы данных.

|  |  |
| --- | --- |
| Основные действия, выполняемые на каждом этапе жизненного цикла приложения базы данных: **Этап** | **Описание** |
| Планирование разработки базы данных | Планирование наиболее эффективного способа реализации этапов жизненного цикла системы |
| Определение требований к системе | Определение диапазона действий и границ приложения базы данных, состава его пользователей и областей применения |
| Сбор и анализ требований пользователей | Сбор и анализ требований пользователей из всех возможных областей применения |
| Проектирование базы данных | Полный цикл разработки включает концептуальное, логическое и физическое проектирование базы данных |
| Выбор целевой СУБД (необязательный этап) | Выбор наиболее подходящей СУБД для приложения базы данных |
| Разработка приложений | Определение пользовательского интерфейса и прикладных программ, которые используют и обрабатывают данные в базе данных |
| Создание прототипов (необязательный этап) | Создание рабочей модели приложения базы данных, которая позволяет разработчикам или пользователям представить и оценить окончательный вид и способы функционирования системы |
| Реализация | Создание внешнего, концептуального и внутреннего определений базы данных и прикладных программ |
| Преобразование и загрузка данных | Преобразование и загрузка данных (и прикладных программ) из старой системы в новую |
| Тестирование | Приложение базы данных тестируется с целью обнаружения ошибок, а также его проверки на соответствие всем требованиям, выдвинутым пользователями |
| Эксплуатация и сопровождение | На этом этапе приложение базы данных считается полностью разработанным и реализованным. Впредь вся система будет находиться под постоянным наблюдением и соответствующим образом поддерживаться. В случае необходимости в функционирующее приложение могут вноситься изменения, отвечающие новым требованиям. Реализация этих изменений проводится посредством повторного выполнения некоторых из перечисленных выше этапов жизненного цикла |

***Подходы к проектированию базы данных***

Существуют два основных подхода к проектированию систем баз данных: нисходящий и восходящий.

Восходящий подход в наибольшей степени приемлем для проектирования простых баз данных с относительно небольшим количеством атрибутов.

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.

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.

***Критерии оценки модели данных***

Оптимальная модель данных должна удовлетворять критериям, перечисленным в таблице. Однако иногда эти критерии несовместимы, поэтому приходится идти на некоторый компромисс. Например, в погоне за наибольшей выразительностью модели данных можно утратить ее простоту.

|  |  |
| --- | --- |
| **Критерий** | **Описание** |
| Структурная достоверность | Соответствие способу определения и организации информации на данном предприятии |
| Простота | Удобство изучения модели как профессионалами в области разработки информационных систем, так и обычными пользователями |
| Выразительность | Способность представлять различия между данными, связи между данными и ограничения |
| Отсутствие избыточности | Исключение излишней информации, т.е. любая часть данных должна быть представлена только один раз |
| Способность к совместному использованию | Отсутствие принадлежности к какому-то особому приложению или технологии и, следовательно, возможность использования модели во многих приложениях и технологиях |
| Расширяемость | Способность развиваться и включать новые требования с минимальным воздействием на работу уже существующих приложений |
| Целостность | Согласованность со способом использования и управления информацией внутри предприятия |
| Схематическое представление | Возможность представления модели с помощью наглядных схематических обозначений |

**Этапы проектирования базы данных**

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

***Концептуальное проектирование базы данных***

Концептуальное проектирование базы данных - процесс создания модели, на основе используемой информации, не зависящей от любых физических аспектов ее представления.

Концептуальное проектирование базы данных абсолютно не зависит от таких подробностей ее реализации, как тип выбранной целевой СУБД, набор создаваемых прикладных программ, используемые языки программирования, тип выбранной вычислительной платформы, а также от любых других особенностей физической реализации.

При разработке концептуальная модель данных постоянно подвергается тестированию и проверке на соответствие требованиям пользователей. Созданная концептуальная модель данных предприятия является источником информации для этапа логического проектирования базы данных.

***Логическое проектирование базы данных***

Второй этап проектирования базы данных называется логическим проектированием базы данных. Его цель состоит в создании логической модели данных для исследуемой части предприятия. Концептуальная модель данных, созданная на предыдущем этапе, уточняется и преобразуется в логическую модель данных. Логическая модель данных учитывает особенности выбранной модели организации данных в целевой СУБД (например, реляционная модель).

Если концептуальная модель данных не зависит от любых физических аспектов реализации, то логическая модель данных создается на основе выбранной модели организации данных целевой СУБД. Иначе говоря, на этом этапе уже должно быть известно, какая СУБД будет использоваться в качестве целевой - реляционная, сетевая, иерархическая или объектно-ориентированная. Однако на этом этапе игнорируются все остальные характеристики выбранной СУБД, например, любые особенности физической организации ее структур хранения данных и построения индексов.

В процессе разработки логическая модель данных постоянно тестируется и проверяется на соответствие требованиям пользователей. Для проверки правильности логической модели данных используется метод нормализации. Нормализация гарантирует, что отношения, выведенные из существующей модели данных, не будут обладать избыточностью данных, способной вызвать нарушения в процессе обновления данных после их физической реализации. Помимо всего прочего, логическая модель данных должна обеспечивать поддержку всех необходимых пользователям транзакций.

Созданная логическая модель данных является источником информации для этапа физического проектирования и обеспечивает разработчика физической базы данных средствами поиска компромиссов, необходимых для достижения поставленных целей, что очень важно для эффективного проектирования. Логическая модель данных играет также важную роль на этапе эксплуатации и сопровождения уже готовой системы. При правильно организованном сопровождении поддерживаемая в актуальном состоянии модель данных позволяет точно и наглядно представить любые вносимые в базу данных изменения, а также оценить их влияние на прикладные программы и использование данных, уже имеющихся в базе.

***Физическое проектирование базы данных***

Физическое проектирование базы данных. Процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты.

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области). Хотя эта структура не зависит от конкретной целевой СУБД, она создается с учетом выбранной модели хранения данных, например реляционной, сетевой или иерархической. Однако, приступая к физическому проектированию базы данных, прежде всего, необходимо выбрать конкретную целевую СУБД. Поэтому физическое проектирование неразрывно связано с конкретной СУБД. Между логическим и физическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, способны повлиять на структуру логической модели данных.

Проектирование базы данных - это итерационный процесс, который имеет свое начало, но не имеет конца и состоит из бесконечного ряда уточнений. Его следует рассматривать, прежде всего, как процесс познания. Как только проектировщик приходит к пониманию работы предприятия и смысла обрабатываемых данных, а также выражает это понимание средствами выбранной модели данных, приобретенные знания могут показать, что требуется уточнение и в других частях проекта. Особо важную роль в общем процессе успешного создания системы играет концептуальное и логическое проектирование базы данных. Если на этих этапах не удастся получить полное представление о деятельности предприятия, то задача определения всех необходимых пользовательских представлений или обеспечения защиты базы данных становится чрезмерно сложной или даже неосуществимой. К тому же может оказаться затруднительным определение способов физической реализации или достижения приемлемой производительности системы. С другой стороны, способность адаптироваться к изменениям является одним из признаков удачного проекта базы данных. Поэтому вполне имеет смысл затратить время и энергию, необходимые для подготовки наилучшего возможного проекта.

На рисунке показана взаимосвязь между этой архитектурой и проектированием баз данных - концептуальным, логическим и физическим.

**Внешняя схема**

**Концептуальная схема**

**Внутренняя схема**

**Физическое хранение данных**

Логическое/концептуальное проектирование базы данных

Физическое проектирование базы данных

Рассмотрим на конкретном примере методику проектирования многотабличной базы данных. Разработка начинается с системного анализа предметной области и построения ее инфологической модели. Инфологическая модель – это структурная модель реальной системы, отражающая ее основные составляющие и связи между ними.

**Спроектируем базу данных «Лошади».**

*Вопрос:* Какая информация о лошадях нам понадобится и какое поле будет ключевым?

|  |
| --- |
| Кличка лошади |
| Дата рождения |
| Порода |
| Масть |
| Пол |
| Коннозаводчик |
| Страна |
| Место выведения |
| Фото лошади |

Основные последствия неправильного проектирования – избыточность информации, ее противоречивость, потеря целостности, т.е. взаимосвязи между данными.

Теория реляционной базы данных была разработана в 70-х годах Е. Коддом. Он предложил технологию проектирования баз данных, в результате применения которой в полученной БД не возникает отмеченных недостатков. Сущность этой технологии сводится к приведению таблиц к третьей нормальной форме. Процесс называется нормализацией данных.

Сначала строится одна таблица, в которую заносится вся информация, необходимая для создания конкретной БД.

Недостатки такой организации данных - избыточность информации повторение одного и того же Коннозаводчика, придется каждый раз писать породу и масть. В такой БД велика вероятность иметь недостоверные, противоречивые данные. Полученная таблица соответствует первой нормальной форме. Требование первой нормальной формы заключается в том, чтобы каждое поле в отношении было неделимым.

Для устранения отмеченных недостатков требуется дальнейшая нормализация.

Необходимо установить основной ключ записей – Код\_лошади (или два поля Кличка\_лошади и Дата\_рождения). Согласно определению второй нормальной формы, все не ключевые поля должны функционально зависеть от полного ключа. Согласно третьей нормальной форме все поля, не зависящие от полного ключа, необходимо выделить в отдельное отношение. Такие поля находятся в транзитивной зависимости.

*Вопрос*: В нашем примере необходимо определить, какие поля зависят от ключевого поля *Код\_лошади (от поля Кличка\_лошади),* какие не зависят от него, а зависят от другого поля?

В результате имеем модель данных, состоящую из четырех взаимосвязанных таблиц:

|  |
| --- |
| **ПОРОДЫ** |
| Код\_породы |
| Порода |
| Цена |
| Место выведения |
| Описание\_породы |

|  |
| --- |
| **КОННОЗАВОДЧИКИ** |
| Код\_коннозаводчика |
| Коннозаводчик |
| Страна |
| Год\_начала |
| Поголовье |

|  |
| --- |
| **ЛОШАДИ** |
| Код\_лошади |
| Кличка |
| Дата\_рождения |
| Код\_породы |
| Код\_масти |
| Пол |
| Код\_коннозаводчика |
| Страна\_обитания |
| Фото лошади |

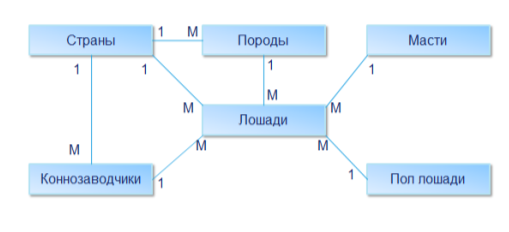
|  |
| --- |
| **МАСТИ** |
| Код\_масти |
| Масть |
| Описание\_масти |

Для удобства ввода информации в основные таблица можно создать дополнительные таблицы-справочники СТРАНА и ПОЛ\_ЛОШАДИ

|  |  |  |
| --- | --- | --- |
| **СТРАНЫ** |  | **ПОЛ\_ЛОШАДИ** |
| Страна |  | Пол |

Эти таблицы можно рассматривать как модель данных в реляционной СУБД. Чтобы эти таблицы представляли собой систему, между ними должны быть установлены связи. Связи позволяют определить соответствия между любыми данными в этих таблицах. Например: между кличкой лошади и коннозаводчиком «Кто является коннозаводчиком Бланки». Благодаря этим связям становится возможным получение ответов на запросы, требующие поиска информации в нескольких таблицах одновременно.

Для указания связей между таблицами построим схему базы данных.



В схеме указывается наличие связей между таблицами и типы связей.

При связи «один-к-одному» с одной записью в таблице связана одна запись в другой таблице. При наличии связи «один-ко-многим» одна запись в некоторой таблице связана с множеством записей в другой таблице.

На этапе ***физического проектирования*** выбирается ***СУБД***, удовлетворяющая требованиям проекта. Логическая схема преобразуется в объекты БД.

1. Практическая работа. Проектирование БД

***Задание 1.*** Создание новой базы данных.

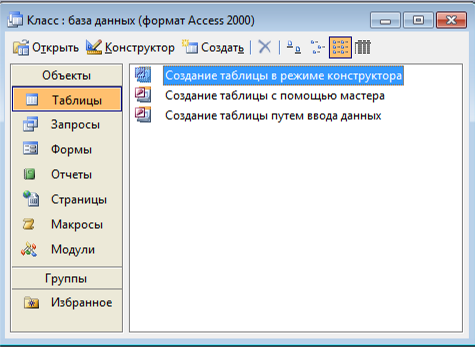
1. Загрузите ***Access***, в появившемся окне выберите пункт **Новая база данных**; в окне **"Файл новой базы данных"** задайте имя вашей базы **(пункт Имя Файла)**  и выберите папку **(пункт Папка),** где ваша база данных будет находиться. По умолчанию **Access** предлагает вам имя базы db1, а тип файла - Базы данные ***Access***. Имя задайте ***Лошади***, а тип файла оставьте прежним, так как другие типы файлов нужны в специальных случаях; щелкните по кнопке *<***Создать***>.*

***Задание2.*** Создать таблицы БД.

**Создание таблицы** – это определение имени и типа данных для каждого из полей с учетом требований и возможностей конкретной СУБД

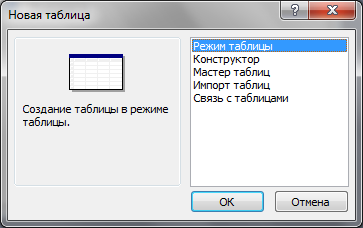
|  |  |
| --- | --- |
| **Объект** | **Описание** |
| **Таблицы** | Объекты, в которых хранятся данные. Выглядят во многом подобно ЭТ. |
| **Запросы** | Извлекает данные из таблиц на основе критериев, заданных пользователем. |
| **Формы** | Шаблоны отображения данных, облегчающие чтение и понимание данных в таблицах, так же используются для более комфортного ввода данных. |
| **Отчеты** | Шаблоны распечатывания данных и проведения дополнительных вычислений. |
| **Страницы** | Шаблоны для представления форм и отчетов в виде HTML-файлов для размещения их в Интернете. |
| **Макросы** | Специальные команды для автоматизации работы с БД |
| **Модули** | Программы на языке VBA для выполнения более сложных операций, которые не могут выполнить макросы. |

|  |  |  |
| --- | --- | --- |
| **Тип данных** | **Описание** | **Размер** |
| **Текстовый** | Алфавитно-цифровые символы | 0 – 255 символов |
| **Поле МЕМО** | Алфавитно-цифровые символы | 0 – 64000 символов |
| **Числовой** | Числовые значения | 1, 2, 4 или 8 байт |
| **Дата/время** | Дата и время | 8 байт |
| **Денежный** | Денежные значения | 8 байт |
| **Счетчик** | Автоматически увеличивающиеся номера | 4 байта |
| **Логический** | Логические значения Да/Нет | 1 бит ( 0 или -1 ) |
| **Поле объекта OLE** | Рисунки, диаграммы, звук и видео | До 1 Гбайт |
| **Гиперссылка** | Связь с ресурсом Internet | 0 – 6144 |
| **Мастер подстановок** | Данные, подставляемые из другой таблицы | Обычно 4 байта |

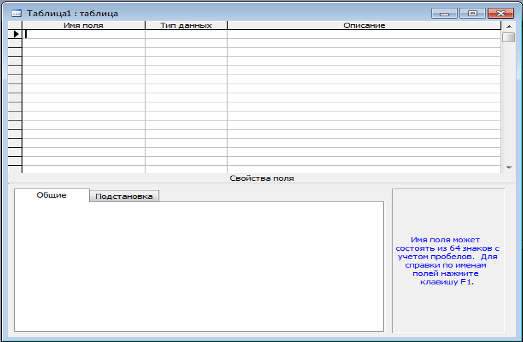


Для создания таблицы базы данных:

* В окне базы данных выберите вкладку **Таблицы**,  а затем щелкните по кнопке *<***Создать***>;*



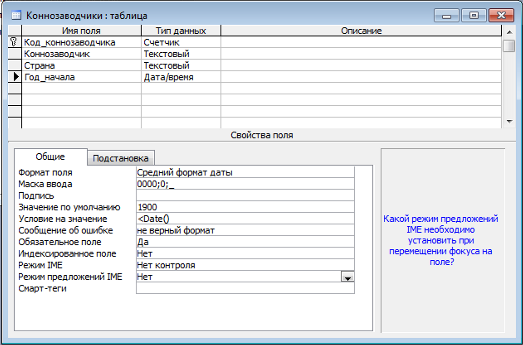
* В окне ***"*Новая таблица*"*** выберите пункт **Конструктор** и щелкните по кнопке <**ОК**>. В результате проделанных операций открывается окно таблицы в режим конструктора, в котором следует определить поля таблицы.



Для определения полей таблицы:

* Введите в строку столбца "**Имя поля"** имя первого поля;
* В строке столбца "**Тип данных"** щелкните по кнопке списка и выберите тип данных.

*Примечание.* Заполнение строки столбца "**Описание**" необязательно и обычно используется для внесения дополнительных сведений о поле.



Для каждого поля (если это требуется) необходимо определить свойства: формат поля, значение по умолчанию, условие на значение, сообщение об ошибке, обязательное поле, пустые строки.

Для определения всех остальных полей таблицы выполните аналогичные действия.

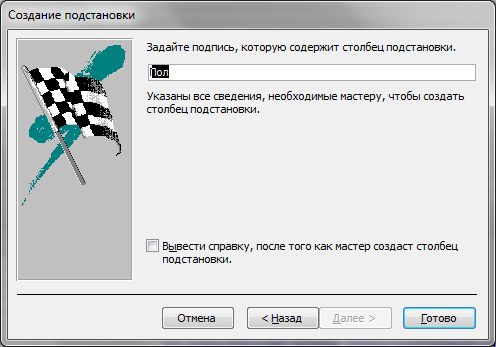
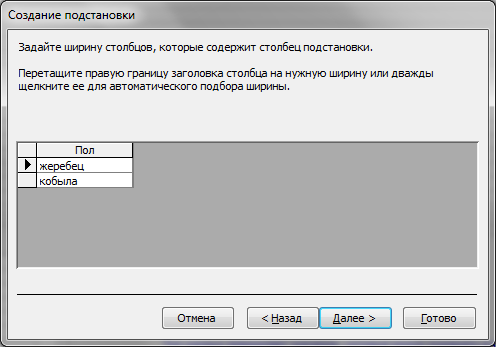
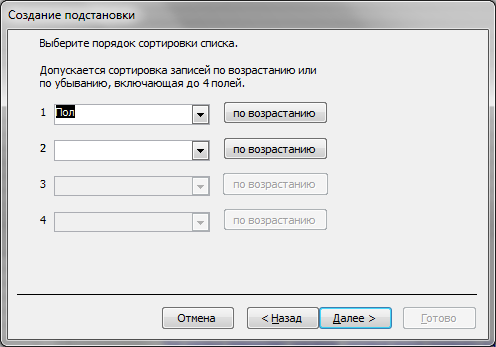
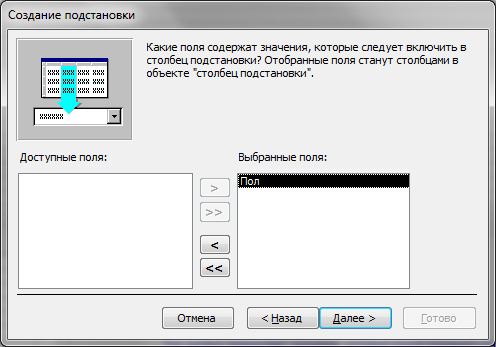
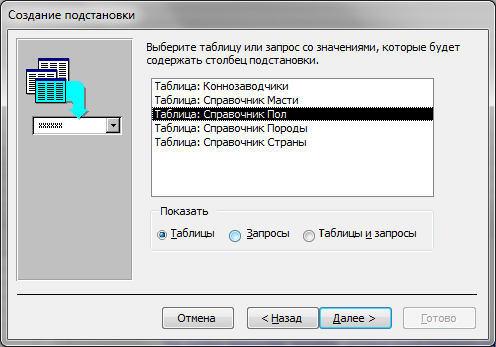
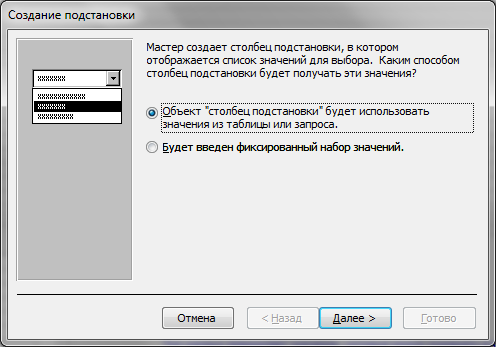
Для сохранения таблицы:

* выберите пункт меню **Файл**, **Сохранить**;
* в диалоговом окне "**Сохранение**" введите имя таблицы;
* щелкните по кнопке <**ОК**>.

*Примечание*. В результате щелчка по кнопке <ОК> **Access** предложит вам задать ключевое поле (поле первичного ключа), т.е. поле, однозначно идентифицирующее каждую запись, щелкните по кнопке <**Да**>.

Используя конструктор, создайте все таблицы БД.

В таблице **Лошади** для полей **Код\_породы, Код\_масти, Код\_коннозаводчика** задайте тип поля числовой, но лучше выполнить подстановку данных из таблиц, с которыми необходимо установить связь – это **Породы, Масти, Коннозаводчики**.



Не забудьте задать ключевые поля.

|  |
| --- |
| **ЛОШАДИ** |
| Код\_лошади |
| Кличка |
| Дата\_рождения |
| Код\_породы |
| Код\_масти |
| Код\_коннозаводчика |
| Пол |
| Страна\_обитания |
| Фото лошади |

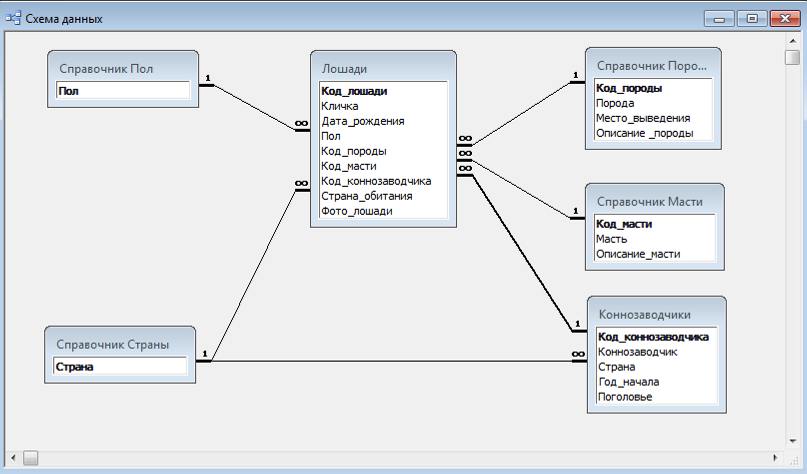
|  |
| --- |
| **ПОРОДЫ** |
| Код\_породы |
| Порода |
| Место выведения |
| Описание\_породы |
|  |
| **МАСТИ** |
| Код\_масти |
| Масть |
| Описание\_масти |

|  |
| --- |
| **КОННОЗАВОДЧИКИ** |
| Код\_коннозаводчика |
| Коннозаводчик |
| Страна |
| Год\_начала |
| Поголовье |

Для поля **Страна** из таблицы **Коннозаводчики** и поля **Страна\_обитания** из таблицы **Лошади** можно создать дополнительную таблицу-справочник **Страны**. Для этих полей выполните подстановку данных из созданного справочника.

То же самое сделайте и для поля **Пол** из таблицы **Лошади**.

После создания таблиц, прежде чем заполнять их, необходимо создать схему данных.



1. Итог урока, рефлексия. На уроке познакомились с понятиями: информационная система, жизненный цикл ИС, этапы проектирования БД. На практике спроектировали таблицы БД и создали схему БД.
2. Домашнее задание: повторение материала.