Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(МИВлГУ)

Факультет Информационных технологий и радиоэлектроники

Кафедра ПМИ

КУРСОВАЯ РАБОТА

по \_\_\_\_\_\_\_\_\_\_Базам данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование дисциплины)

Тема Информационная система на тему: “Зоопарк”\_\_\_\_\_

Руководитель

Колпаков А.А.

(фамилия, инициалы)

(подпись) (дата)

Студент ИБ-122

(группа)

Мошков Т. Д.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

Введение

Современные зоопарки являются не только местом для развлечения и отдыха, но и важными учреждениями, занимающимися сохранением биоразнообразия, образованием и научными исследованиями. В условиях растущего интереса к экологии и защите животных необходимо эффективно управлять данными о большом количестве видов, их характеристиках, состоянии здоровья и условиях содержания. Учитывая разнообразие животных, которые содержатся в зоопарках, включая млекопитающих, птиц, рептилий и амфибий, адекватная обработка и хранение информации становятся актуальными задачами.

Сложность учета данных о животных возрастает из-за необходимости хранить информацию о различных видах, их индивидуальных характеристиках, ветеринарных карточках, а также взаимодействии с посетителями и образовательными программами. Ручная обработка данных становится неэффективной и может привести к ошибкам, что может негативно сказаться на состоянии животных и репутации зоопарка. Кроме того, отсутствие оперативного доступа к информации о состоянии здоровья, питании и поведении животных может затруднить принятие решений в экстренных ситуациях.

Целью данной курсовой работы является разработка информационной системы для работы с базой данных зоопарка, которая будет эффективно организовывать хранение, обработку и анализ информации о животных. В результате реализации данной системы каждый сотрудник зоопарка сможет легко получать доступ к необходимым данным, что значительно повысит эффективность работы учреждения и обеспечит более качественный уход за животными.

1. Анализ технического задания

1.1. Описание предметной области

Предметная область данной курсовой работы — зоопарк, который представляет собой часть реального мира, данные о которой мы хотим отразить в базе данных. Основной задачей является проектирование и разработка приложения для автоматизации управления учётом животных, их здоровья и взаимодействия с посетителями. Предполагаемая база данных должна обеспечивать работу зоопарка по учету различных видов животных, автоматизированную выдачу отчетов о состоянии популяций, здоровье животных и посещаемости.

Процесс управления данными о животных осуществляется следующим образом:

Поиск информации о животных. Сотрудники зоопарка, включая ветеринаров и кураторов, взаимодействуют с базой данных для получения информации о каждом животном, включая его характеристики, состояние здоровья и особенности содержания. На этом этапе сотрудники могут также получить доступ к рекомендациям по уходу и питанию.

Регистрация новых животных. При поступлении нового животного в зоопарк главный ветеринар или специалист по содержанию регистрируют его в базе данных. Для регистрации необходимо предоставить информацию о виде, возрасте, состоянии здоровья, а также ветеринарные документы.

Возможное событие «Животное недоступно». В некоторых случаях животное может быть временно недоступно для посетителей, например, из-за болезни или проведения ветеринарных процедур. В таких ситуациях сотрудники должны иметь возможность обновлять информацию в базе данных, чтобы избежать недоразумений.

Взаимодействие с посетителями. Сотрудники зоопарка могут регистрировать посещения, предоставлять информацию о животных и организовывать мероприятия. При этом важно отслеживать отзывы и пожелания посетителей для улучшения услуг.

Выделим базовые сущности этой предметной области:

Животные.

Атрибуты животных: название, вид, класс, возраст, пол, страна обитания, масса, высота тела, продолжительность жизни, номер ветеринарной карточки.

Сотрудники зоопарка.

Атрибуты персонала: ФИО, должность, телефон. Эти данные позволят эффективно управлять процессами и взаимодействием между сотрудниками.

Посетители.

Атрибуты посетителей: ФИО, контактные данные.

Таким образом, создание базы данных для зоопарка позволит автоматизировать процессы учета, обеспечить более качественный уход за животными и улучшить взаимодействие с посетителями, что в итоге повысит эффективность работы учреждения.

1.2. Аналоги

В настоящее время реализовано огромное количество АС для автосалонов.

Примером такой системы может являться управление-зоопарком.рф онлайн сервис для управления зоопарком на базе “1C:Предприятие”.

Достоинства:

* Автоматизация процессов: Система автоматизирует основные задачи управления зоопарком, такие как учет животных, управление их кормлением и медицинским обслуживанием.
* Удобный интерфейс: Интерфейс сделан интуитивно понятным, что упрощает обучение сотрудников и позволяет быстро начинать работу в системе.
* Поддержка учета и статистики: Поддержка аналитики позволяет следить за изменениями в популяции животных, фиксировать данные о болезнях, смертности и рождении.
* Уведомления и напоминания: Система может автоматически напоминать о важных задачах, таких как график вакцинации или другие медицинские процедуры.
* Мобильное приложение: Наличие мобильного приложения позволяет сотрудникам оперативно вносить данные и просматривать нужную информацию непосредственно на месте.

Недостатки:

* Зависимость от Интернета: Система требует постоянного подключения к Интернету, что может быть проблематично в случае временной недоступности сети.
* Затраты на внедрение: Стоимость внедрения и адаптации системы может быть значительной, особенно для небольших зоопарков.
* Ограниченная кастомизация: Возможности настройки системы под специфические нужды зоопарка могут быть ограничены.
* Потребность в обучении: для полного использования функционала требуется обучение персонала, что может занять некоторое время.
* Отсутствие некоторых функций: Некоторые функции, которые могли бы быть полезными, например, интеграция с внешними системами учета или финансирования, могут отсутствовать.

В качестве другого примера можно привести систему «Матрица. Обмен с ГИС Меркурий» – программное обеспечение, предназначенное для автоматизации учета, планирования и анализа работы предприятий, связанных с животноводческой продукцией: от производителей и переработчиков до торговых компаний, складов и ветеринарных служб.

Достоинства системы «Матрица. Обмен с ГИС Меркурий»:

* Полная автоматизация процессов обмена данными с ГИС «Меркурий», что снижает трудозатраты и повышает точность учёта продукции.
* Обеспечение единой базы данных для животноводческой продукции, контрагентов и операций, связанной с её оборотом.
* Инструменты для гибкого управления производственными и логистическими процессами в соответствии с требованиями системы ветеринарного контроля.
* Поддержка аналитики, отчетности и панелей мониторинга для отслеживания статуса продукции и выполнения требований законодательства.
* Возможность интеграции с другими учетными и ERP-системами для комплексного управления данными предприятия.

Недостатки:

* Высокие первоначальные затраты на внедрение и обучение персонала.
* Возможные сложности при интеграции с существующими системами управления и учёта на предприятии.
* Необходимость адаптации бизнес-процессов предприятия к требованиям системы и законодательным стандартам.

1.3. Анализ и выявление лучшей среды разработки.

Для разработки автоматизированной информационной системы (АИС) для зоопарка рассмотрим, какой язык программирования, среду разработки и базу данных лучше выбрать, учитывая, что для курсовой работы подойдут решения, которые оптимально сочетают функциональность и удобство.

Таблица 1 - C++ и C# для АИС «Зоопарк»

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **C++** | **C#** |
| **Удобство разработки** | Требует более сложного синтаксиса и управления памятью вручную, что может увеличить объем работы | Более простой, высокоуровневый синтаксис, автоматическое управление памятью, что делает разработку проще |
| **Поддержка ООП** | Полная поддержка ООП, но требует написания большего объема кода | Полная поддержка ООП, встроенные удобные средства для работы с объектами и коллекциями |
| **Скорость разработки** | Длительная, требует дополнительного времени на отладку и проверку работы с памятью | Высокая скорость разработки за счет сборки мусора и готовых библиотек для работы с базами данных |
| **Интеграция с базой данных** | Требуется использование сторонних библиотек для подключения к базам данных | Встроенные библиотеки для работы с базами данных, что упрощает интеграцию |
| **Применение** | Рекомендуется для системных и высокопроизводительных приложений | Идеально подходит для бизнес-логики и разработки информационных систем |

Для курсовой работы на тему АИС «Зоопарк» лучше выбрать C#, так как он предоставляет удобные инструменты для быстрой разработки информационных систем и обладает встроенными средствами для работы с базами данных. Благодаря поддержке объектно-ориентированного программирования разработка становится более структурированной и управляемой, что облегчает работу с большими объемами данных.

Кроме того, C# имеет обширную библиотеку и множество готовых компонентов, что ускоряет процесс разработки и позволяет сосредоточиться на реализации бизнес-логики, а не на рутинных задачах.

Таблица 2 - Visual Studio и SharpDevelop для разработки АИС

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Visual Studio** | **SharpDevelop** |
| **Функциональность** | Полный набор инструментов для разработки на C#, включая продвинутую отладку и рефакторинг кода | Базовый набор функций для разработки на C# |
| **Производительность** | Высокие требования к ресурсам, но идеален для крупных и средних проектов | Легковесный, занимает меньше ресурсов, но с ограниченными возможностями |
| **Интеграция с базами данных** | Поддержка интеграции с различными СУБД (MS SQL, SQLite и др.) через встроенные плагины и инструменты | Поддержка ограничена, но можно подключать базу данных вручную |
| **Расширяемость** | Широкий выбор расширений и плагинов для работы с разными фреймворками и библиотеками | Ограниченная поддержка плагинов, что уменьшает возможности настройки |
| **Подходит для учебных проектов** | Имеет бесплатную версию Community с полным функционалом, подходит для учебных проектов | Бесплатный, но с меньшим функционалом, требует больше ручных настроек |

Visual Studio Community – более удобный вариант для курсовой работы, так как предоставляет готовые инструменты для разработки информационных систем и включает встроенные средства для работы с базами данных.

Эта среда разработки предлагает широкий спектр функций, таких как мощный отладчик, поддержка различных языков программирования и интеграция с системами контроля версий. Кроме того, наличие активного сообщества и обширной документации позволяет легко находить решения для возникающих вопросов и проблем в процессе работы.

Таблица 3 – MS Access и SQLite для хранения данных АИС

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **MS Access** | **SQLite** |
| **Назначение** | Подходит для небольших локальных приложений и баз данных с ограниченным количеством пользователей | Встроенная база данных для мобильных и небольших настольных приложений |
| **Простота использования** | Дружелюбный интерфейс, позволяет легко настраивать таблицы и запросы через GUI | Легкая в использовании, интеграция с приложением напрямую через код |
| **Установка** | Требует Microsoft Office и работает только на Windows | Встроенная в приложение, не требует дополнительной установки |
| **Производительность** | Подходит для малых объемов данных, но не оптимизирована для высоких нагрузок | Подходит для малых и средних объемов данных, может работать в многопоточной среде |
| **Поддержка SQL** | Ограниченная поддержка SQL, более ориентирована на конечных пользователей | Полная поддержка SQL, что позволяет работать с запросами на уровне кода |
|  |  |  |

SQLite будет лучшим вариантом для курсовой работы, так как легко интегрируется с C# и требует меньше настроек, обеспечивая при этом высокую производительность для небольших и средних объемов данных.

2. Разработка моделей данных.

Этот этап является ключевым при разработке автоматизированной информационной системы (АИС). В ходе работы определяются сущности, их атрибуты, а также устанавливаются связи между сущностями. На основе созданной диаграммы "Сущность – связь" или логической модели разрабатываются функциональные модели системы и диаграмма потоков данных. Для создания базы данных необходимо преобразовать логическую модель в физическую.

2.1 Логическая модель

Создание логической модели для автоматизированной информационной системы (АИС) зоопарка начинается с анализа предметной области и выделения основных сущностей. По условиям задачи, в зоопарке имеются животные, вольеры, посетители и работники.

Сущность "Животные" представляет собой конкретных животных, содержащихся в зоопарке, таких как тигры, слоны и другие виды. Важно учитывать связь животных с их местом содержания, поэтому выделена сущность "Вольеры", где содержится информация о вольерах для каждого животного. Для управления работой зоопарка выделена сущность "Работники", которая связана с животными и вольерами через атрибут "ОтветственныйСотрудникID", что позволяет определить, кто отвечает за уход за определённым животным и вольером.

Также существует связь между "Работниками" и "Билетами", так как сотрудники зоопарка занимаются продажей билетов. Сущность "Посетители" связана с "Билетами", что позволяет отслеживать, какие билеты были приобретены каждым посетителем, а также учитывать дату и номер билета. Таким образом, логическая модель отображает взаимодействие между ключевыми сущностями: Посетители, Билеты, Работники, Животные и Вольеры.

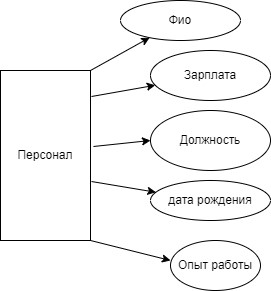


Рисунок 1 – атрибуты работников

1. Животные

Атрибуты животных страна обитания, вид, название, номер ветеринарной карты, пол и возраст представлены на рисунке 2

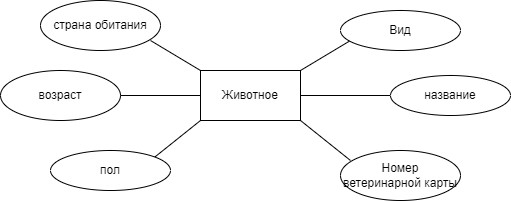


Рисунок 2 – атрибуты животных

1. Билет

Атрибуты билета: ID Билета, Дата приобретения, Кем выдан, Вид услуг, взрослый или детский билет, представлены на рисунке 3

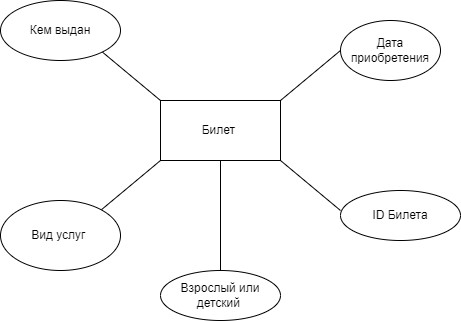


Рисунок 3 - атрибуты билета

1. Посетитель

У посетителя для построения базы данных мне будет важно знать только возраст и Ф.И.О., чтобы определить вид его билета и вести учет посетителей.

Необходимые атрибуты представлены на рисунке 4.

C:\Users\PC\AppData\Local\Packages\Microsoft.Windows.Photos_8wekyb3d8bbwe\TempState\ShareServiceTempFolder\Диаграмма без названия.drawio (4).jpeg

Рисунок 4 - атрибуты посетителя

В результате анализа предметной области была составлена коцептуальная модель данных, представленная на рисунке 5.

Изображение выглядит как диаграмма, рисунок, зарисовка, линия

Автоматически созданное описание

Рисунок 5 – концептуальная модель данных

На данной диаграмме отображены связи между основными сущностями системы зоопарка:

Посетитель – Покупает – Билет:

Посетитель покупает билет, который имеет атрибуты Номер и Дата.

Билет – Разрешает доступ – Животное:

Билет предоставляет доступ к животным в зоопарке.

Работник зоопарка – Продает – Билет:

Работник зоопарка отвечает за продажу билетов.

Работник зоопарка – Ухаживает – Животное:

Работник зоопарка ухаживает за животными. У работника есть атрибуты: Фамилия, Имя, Должность, Зарплата, Дата приема на работу.

Животное – Содержится в – Вольер:

Животное размещено в вольере, который имеет атрибуты Вид вольера и Корм.

Животное – имеет атрибуты: Вид, Название, Возраст, Пол.

Эти связи между сущностями отражают базовую логику функционирования зоопарка: посетители покупают билеты, которые позволяют им видеть животных, работники зоопарка ухаживают за животными и продают билеты, а животные содержатся в вольерах с определенными условиями.

2.2 IDEF0 диаграммы.

2.3 DFD диаграммы.

2.4 Приведение логической модели к физической.

Приведение логической модели к физической — это важный этап проектирования базы данных, на котором абстрактная логическая структура преобразуется в конкретные технические решения для реализации в системе управления базами данных (СУБД). Логическая модель описывает основные сущности, атрибуты и связи между ними, но не учитывает такие аспекты, как типы данных, индексы, ограничения, способы хранения данных и их оптимизацию для производительности.

Физическая модель, в свою очередь, детализирует каждый элемент с учётом возможностей выбранной СУБД. На этом этапе каждому атрибуту сущности в логической модели назначаются конкретные типы данных (например, INT, VARCHAR, DATE). Определяются первичные и внешние ключи, создаются индексы для оптимизации запросов, а также устанавливаются правила целостности данных, такие как ограничения уникальности и проверки на null-значения. Кроме того, физическая модель учитывает параметры производительности, распределение данных по таблицам и их размещение на физическом носителе, чтобы обеспечить оптимальное хранение и доступ к данным.

Таким образом, переход от логической модели к физической включает в себя не только техническую конкретизацию структуры данных, но и оптимизацию системы для её эффективного функционирования в реальной среде.

1.3. Логическая модель базы данных

В ходе разработки информационной системы была выделена логическая модель базы данных зоопарка Изображение выглядит как текст, снимок экрана, диаграмма, число

Автоматически созданное описание

Рисунок 7 – Логическая модель базы данных

Логическая модель состоит из пяти основных таблиц, каждая из которых отвечает за определенный аспект работы зоопарка. Связи между таблицами реализованы следующим образом: Посетители связаны с Билетами через ПосетительID, что позволяет отслеживать, какие билеты были куплены каждым посетителем.

Сотрудники связаны с Животными и Вольерами через ОтветственныйСотрудникID, что позволяет определить, кто отвечает за уход за конкретными животными и вольерами.

Животные связаны с Вольерами через ВольерID, указывая, в каком вольере находится каждое животное.

КАССУ убрать , все модели во второй раздел, концептуальную модель обьедени во одну , чтобы не было двух посетителей и тд.