ВВЕДЕНИЕ

В настоящее время развитие электронных технологий происходит стремительным образом, поэтому существует необходимость быстро и эффективно оперировать большими, сложно взаимосвязанными объемами данных. Такую возможность могут предоставить базы данных. К преимуществам баз данных, а именно реляционных баз данных, в которых данные хранятся в виде таблицы, можно отнести компактность, легкую организацию работы с помощью единого языка SQL, надежность общего доступа [1].

Целью данной курсовой работы является создание информационной системы «Размещение животных», которая должна оперировать и взаимодействовать с большим объемом данных о животных, рабочих, видах животных, клеток. При этом система должна работать быстро. Реляционные базы данных предоставляют удобные инструменты для достижения этой цели, экономя время и ресурсы разработчика. Именно поэтому они и были выбраны для использования в данном проекте. Для реализации реляционной базы данных была выбрана СУБД Microsoft SQL Server, из-за простоты в использовании (упрощенное создание таблиц, отсутствие сложных операций управления базой данных и т. д.) и надежности платформ базы данных [3]. Кроме того, СУБД поддерживается средой разработки Microsoft Visual Studio 2015, которая была выбрана для создания прикладной программы, что позволило избежать использование внешних СУБД и их подключение.

1 АНАЛИЗ И КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Зоопарк составляет большую группу животных. Главное предназначение зоопарка ­ – содержание диких животных для демонстрации их городскому населению и просветительская деятельность по охране природы и животного мира, развивая в людях представление о том, что они являются частью мира живой природы и что ответственность за сохранение этого мира должна гармонично войти в их повседневную жизнь.

Зоопарк посещают люди любого возраста, поодиночке или группами. Практически каждый человек хоть раз посещал зоопарк. Также учёные используется зоопарк для исследования животных и сохранения их видов.

Зоопарк является престижным, развивающимся заведением. Люди любят заботиться о животных. Также зоопарк является дешёвым развлекательным местом. Поэтому количество зоопарков растёт.

В зоопарке находятся различные животные с различных мест планеты. Всем этим животным надо особый уход и имитировать их место обитание. Существуют животные приспособленные к сухому горячему климату, к влажному тёплому климату, к холодному климату, к переменчивому климату, к жизни в воде (тёплой или холодной). Обычно животные приспособленные к одинаковым условиям живут рядом, для удобства за ними ухаживать.

Есть различные сотрудники зоопарка: кассиры, смотрители за местом обитания животных, ремонтники, ветеринары, зоологи, бухгалтеры, директор. Директор – главный сотрудник зоопарка, который следит за всеми другими сотрудниками. Бухгалтеры следят за финансами зоопарка. Зоологии составляют рацион для животных. Ветеринары лечат животных, следят за их здоровьем. Ремонтники чинят дороги, клетки, вольеры, аквариумы и т.д. Смотрители кормят животных, убирают за ними. Кассиры продают билеты в зоопарк.

Зоопарки обычно имеют большую область и большое количество животных. Поэтому целью данной системы является помощь сотрудникам в слежении за животными и помощь заселения животных в клетки.

* 1. Описание информационных потребностей

Данная система предназначена для облегчения расселения животных и слежения за сотрудниками.

Основываясь на проведенном анализе предметной области, информационные потребности пользователей были представлены следующим образом: система будет рассчитана на сотрудников зоопарка.

Руководство хочет следить за работай подчинённых, проверять кто как сделал свою работу.

Информационные потребности пользователей (сотрудников зоопарка) данной информационной системы следующие:

а) вход в программу;

б) просмотр информации об имеющихся животных;

в) добавление животных;

г) удаление животных;

д) изменение информации о животных;

е) сортировка информации по номеру животного, кличке, весу, возрасту, виду, номеру клетки, нужной площади;

ё) просмотр статистики по количеству канцелярских товаров по названию, производителю, назначению и сфере применения;

ж) отчет информации о наличии канцелярских товаров, который содержит код товара, название, производитель, количество и цену;

з) просмотр информации о сотрудниках;

и) добавление сотрудника;

к) удаление сотрудника;

л) изменение информации о сотруднике;

м) сортировка информации по номеру сотрудника, имени, фамилии, возрасту, должности, номеру клетки;

н) просмотр информации о клетке;

о) добавление клетки;

п) удаление клетки;

р) изменение информации о клетке;

с) поиск информации по номеру клетки, площади, типу, количеству сотрудников, количеству животных;

т) фильтрация информации по типу клеток, количеству сотрудников, площади и количеству животных;

у) сортировка информации по типу клеток, количеству сотрудников, площади и количеству животных;

ф) просмотр информации о виде животного;

х) добавление вида животного;

ц) удаление вида животного;

ч) изменение информации о виде животного;

ш) просмотр обязанностей сотрудников по клеткам;

щ) добавление обязанностей сотрудников по клеткам;

ы) редактирование обязанностей сотрудников по клеткам;

ь) просмотр статистика по количеству животных по виду животных, клеток по типу клеток, рабочих по должностям;

э) составление отчётов принятых работников на работу в определённый месяц и животных живущих в зоопарке.

* 1. Описание объектов предметной области, их атрибутов и связей

На основе анализа предметной области и информационных потребностей можно выделить следующие объекты системы:

В соответствие с предметной областью, база данных будет сохранять следующую информацию:

а) данные о животном:

* + 1. код животного;
    2. кличка;
    3. вид животного;
    4. возраст;
    5. вес;
    6. номер места обитания;
    7. нужная свободная площадь для проживания;

б) данные о клетке:

1. номер клетки;
2. площадь;
3. тип;

в) данные о сотруднике:

1. код сотрудника;
2. ФИО;
3. дата рождения;
4. дата найма;
5. дата увольнения;
6. данные паспорта;
7. номер телефона;
8. должность;
9. номер клетки;
10. обязанность по клетке.

г) данные о виде животного:

1. название;
2. дружественный вид;
3. подходящая клетка.

Схема взаимодействия объектов предметной области приведена на рисунке 1.1.

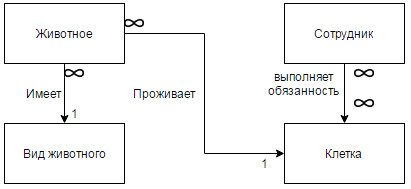


Рисунок 1.1 — Диаграмма взаимодействий объектов

* 1. Описание ограничений целостности

Для нормального функционирования базы данных вводятся следующие ограничения:

а) ограничения касательно объектов:

1) каждый сотрудник однозначно идентифицируется его кодом, поскольку может существовать совпадение имен сотрудников, их должностей, дат найма и рождения, номера телефона;

2) каждое животное однозначно идентифицируется по своему коду, поскольку животные могут иметь одинаковую кличку, номер клетки, вид, возраст, вес, нужную площадь для проживания;

3) каждая клетка идентифицируется номером, поскольку клетки могут иметь одинаковую площадь, тип;

4) каждый вид животного идентифицируется названием вида, поскольку виды животных могут иметь одинаковые дружественные виды животных и подходящие клетки.

б) ограничения касательно связей между объектами:

1) сотрудник может обслуживать несколько клеток, клетка обслуживается несколькими сотрудниками;

2) животное живёт в одной клетке. В одной клетке живёт много животных;

3) животное имеет один вид, вид имеет несколько животных;

## Описание существующего документооборота в предметной области

Документооборот в предметной области состоит из:

1. лист с людьми принятых на работу в определённый месяц;
2. документ с данными о животных зоопарка;

1.5 Описание алгоритмических зависимостей

В данной предметной области имеются следующие алгоритмические зависимости:

а) когда добавляется животное в клетку проверяется хватает ли места для этого животного, подходящая ли клетка и живут ли в этой клетке дружественные виды животных. Если так, то оно добавляется в клетку;

б) свободное место в клетке считается так: площадь клетки минус сумма нужных площадей для каждого животного в этой клетке;

в) при удалении клетки из базы данных, проверяется есть ли там животные;

г) при удалении вида из базы, делается проверка, имеется ли животное, который имеет этот вид;

д) автоматическое распределение животного в клетку при добавление его в базу;

2. ПОСТАНОВКА ЗАДАЧИ

Необходимо организовать процесс поселения животных по клеткам зоопарка с внесением в базу данных информации о видах, животных. Для этого необходимо обладать информацией о клетках, имеющихся в наличии, и сотрудниках, которые будут обслуживать клетки. Существует следующая информация о животных: код животного, кличка, вес, возраст, вид, нужная площадь для проживания, номер клетки; о сотрудниках: код сотрудника, ФИО, дата рождения, дата найма, дата увольнения, номер телефона, паспортные данные, должность, номер клетки, обязанность; о клетках: номер клетки, площадь, тип, количество сотрудников, количество животных; о видах животных: название вида, дружественный вид, подходящий тип клетки для проживания. Так как целью данного проекта является оптимизация процесса заселения животных по клеткам в зоопарке и облегчение просмотра информации о животных, клетках, сотрудниках, видах, разрабатываемая система должна иметь высокую надежность, эргономичный и интуитивно понятный интерфейс и выполнять базовые необходимые функции и функции, которые являются задачами автоматизации в данной системе.

В данной информационной системе необходимо реализовать:

а) просмотри информации об имеющихся клетках;

б) добавление, удаление и редактирование данных о клетках;

в) поиск информации по номеру клетки, типу клетки, площади, количеству работников;

г) фильтрация информации по типу клетки, площади, количеству работников;

д) сортировка информации по номеру клетки, типу, площади, количеству работников;

е) просмотр информации о сотрудниках;

ж) добавление, удаление и редактирование информации о сотрудниках;

з) просмотр информации о животных;

и) добавление, удаление и редактирование информации о животных;

к) просмотр информации о видах;

л) добавление, удаление и редактирование информации о видах;

м) просмотр статистика по количеству животных по виду животных, клеток по типу клеток, рабочих по должностям;

н) составление отчётов принятых работников на работу в определённый месяц и животных живущих в зоопарке.

3. Проектирование базы данных

3.1 UML моделирование.

На основании концептуальной модели предметной области, которая была получена в первом разделе для диаграммы UML Use Case, были определены следующий актер: работник. Работник управляет информацией о животных, при этом работник непременно участвует в прецеденте «Получение информации о животном». Аналогично работник управляет информацией о работниках, клетках, видах животных и участвует в прецедентах «Получение информации о работнике», «Получение информации о клетке», «Получение информации о виде».

Диаграмма UML Use Case приведена на рисунке 3.1.



Рисунок 3.1 — Диаграмма UML Use Case

3.2 Построение ER-диаграммы

На основании концептуальной модели предметной области, которая была получена в первом разделе можно получить такие сущности:

а) стержневая сущность «Животное» с атрибутами: код животного\*, кличка, вид животного, возраст, вес, номер клетки, требуемая площадь;

б) стержневая сущность «Клетка» с атрибутами: номер клетки\*, тип клетки, площадь;

в) стержневая сущность «Работник» с атрибутами: код работника\*, ФИО, дата рождения, дата найма, дата увольнения, обязаность, данные паспорта, номер телефона;

г) стержневая сущность «Вид» с атрибутом: название вида\*;

д) стержневая сущность «Дружелюбный вид» с атрибутами: код дружбы\*, название вида, дружелюбный вид;

е) стержневая сущность «Подходящая клетка» с атрибутами: код подходящей клетки\*, название вида, тип клетки;

ж) ассоциативная сущность «Место работы» с атрибутами: код места работы\*, номер клетки, код работника, обязанность.

Определим связи сущностей. Известно, что в одной клетке может жить много животных, а одно животное живёт в 1 клетке. Это позволяет понять, что связь между сущностями «Клетка» и «Животное» 1:М (один ко многим). В таком случае, связь между сущностями осуществима по атрибуту «Номер клетки».

Животное имеет один вид, в то время как одним видом могут владеть много животных. Из этого следует, что связь между сущностями «Вид» и «Животное» 1:М (один ко многим) и связь между этими сущностями осуществима по атрибуту «Название вида».

Вид имеет несколько дружб, а одна дружба происходит с одним видом. Следовательно, связь между сущностями «Вид» и «Дружелюбный вид» 1:М (один ко многим). Связь осуществима по атрибуту «Название вида».

Вид имеет несколько подходящих типов клеток, а одна подходящая клетка имеет один вид животного. Следовательно, связь между сущностями «Вид» и «Подходящая клетка» 1:М (один ко многим). Связь осуществима по атрибуту «Название вида».

Работник может следить за несколькими клетками, а за клеткой могут следить несколько работников. Следовательно, связь между сущностями «Работник» и «Клетка» N:M (многие ко многим). Так как связь многие ко многим решено добавить ассоциативную сущность под названием «Место работы». Связь осуществима по атрибутам «Код работника» и «Номер клетки».

ER-диаграмма приведена на рисунке 3.2.

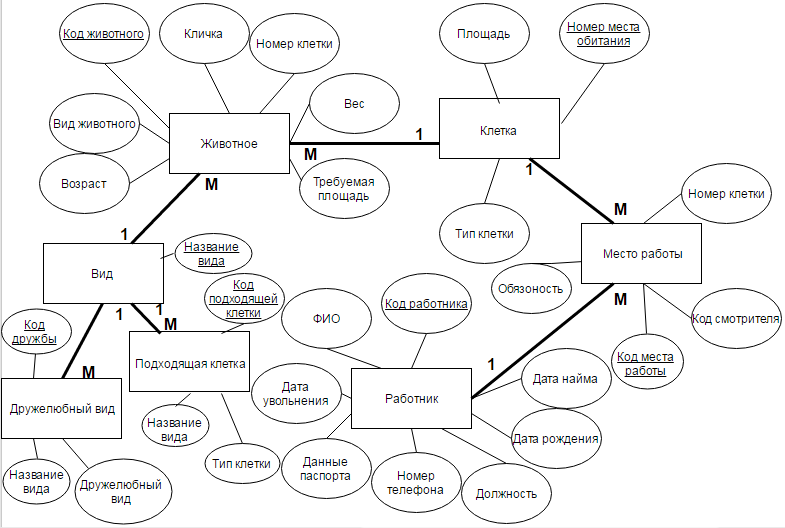


Рисунок 3.2 — ER-диаграмм

3.3 Построение схемы реляционной базы данных в третьей нормальной форме

Нормализацией базы данных называется процедура, производимая над базой данных с целью удаления в ней избыточности.

Нормализация базы данных происходит путем приведения ее таблиц к

нормальным формам.

1НФ представляет собой отношение. Для него характерны атомарность значений на пересечении столбцов и строк, кортежи и атрибуты должны быть неупорядоченные и кортежи не дублированы.

Для 2НФ характерны особенности 1НФ и полная функциональная зависимость не ключевых атрибутов от первичного ключа отношения.

3НФ – состояние отношения, которое находится во 2НФ и в котором между атрибутами нет транзитивных зависимостей.

Переход к следующей нормальной форме возможен только тогда, когда удовлетворяются все условия предыдущей формы [1].

Для предметной области построим универсальное отношение, включающее описание ранее выделенных объектов и их свойств (см. рис. 3.3)

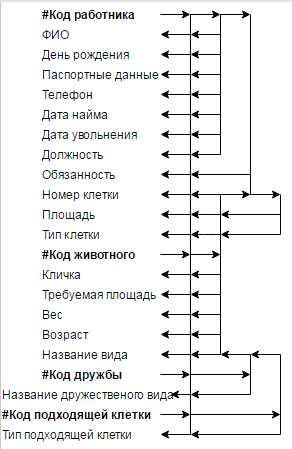


Рисунок 3.3 – Универсальное отношение проектируемой базы данных

Данное универсальное отношение удовлетворяет требования 1НФ, то есть является отношением реляционных баз данных, а именно: кортежи не упорядочены, атрибуты не упорядочены, информация кортежей не дублирована, значения атрибутов атомарные. Это универсальное отношение не является 2НФ, так как поле «День рождения» зависит от поля «Код работника» и от общего ключа «Код работника, Код животного, Код дружбы, Код подходящей клетки» (не полная функциональная зависимость). Поле «Название дружественного вида» зависит от поля «Код дружбы» и от общего ключа «Код дружбы, Код работника, Код животного, Код подходящей клетки» (не полная функциональная зависимость). Поле «Тип подходящей клетки» зависит от поля «Код подходящей клетки» и от общего ключа «Код дружбы, Код работника, Код животного, Код подходящей клетки» (не полная функциональная зависимость). Продолжим процедуру нормализации.

Разобьём отношение на 4 таблицы. Первая таблица Т1 (см. рис. 3.4) находится во 2НФ, поскольку все атрибуты зависят от «Код дружбы». Поскольку таблица Т1 находится во 2НФ и не имеет транзитивных зависимостей, то она находится в 3НФ. Дадим ей название «Дружественный вид».



Рисунок 3.4 – Таблица Т1

Вторая таблица Т2 (см. рис. 3.5) находится во 2НФ, поскольку все атрибуты зависят от «Код подходящей клетки». Поскольку таблица Т2 находится во 2НФ и не имеет транзитивных зависимостей, то она находится в 3НФ. Дадим ей название «Подходящая клетка».



Рисунок 3.5 – Таблица Т2

Третья таблица Т3 (см. рис. 3.6) находится во 2НФ, поскольку все атрибуты зависят от «Код животного».

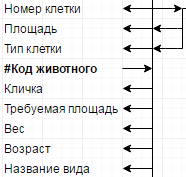


Рисунок 3.6 – Таблица Т3

Четвёртая таблица Т4 (см. рис. 3.7) находится не во 2НФ, так как поле «Обязанность» зависит от общего ключа «Код работника, Номер клетки» (не полная функциональная зависимость).

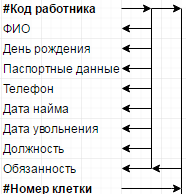


Рисунок 3.7 – Таблица Т4

Для получения 2НФ разобьем таблицу Т4 на две. Первая таблица Т4.1 (см. рис. 3.8) находится во 2НФ, поскольку все атрибуты зависят от «Код работника». Поскольку таблица Т4.1 находится во 2НФ и не имеет транзитивных зависимостей, то она находится в 3НФ. Дадим ей название «Работник».

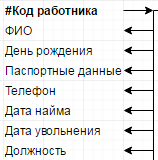


Рисунок 3.8 – Таблица Т4.1

Вторая таблица Т4.2 (см. рис. 3.9 находится во 2НФ, поскольку все атрибуты зависят от общего ключа «Код работника, Номер клетки». Поскольку таблица Т4.2 находится во 2НФ и не имеет транзитивных зависимостей, то она находится в 3НФ. Дадим ей название «Место работы».



Рисунок 3.9 – Таблица Т4.2

Для получения 3НФ, разобьем таблицу Т3 на две таблицы. Первая таблица Т3.1 (см. рис. 3.10) находится в 3НФ поскольку находится во 2НФ (все атрибуты зависят от «Номер клетки») и не имеет транзитивных зависимостей. Назовем таблицу «Клетка».



Рисунок 3.10 – Таблица Т3.1

Вторая таблица Т3.2 (см. рис. 3.11) находится в 3НФ поскольку находится во 2НФ (все атрибуты зависят от «Код животного») и не имеет транзитивных зависимостей. Назовем таблицу «Животное».

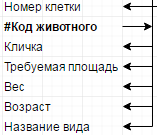


Рисунок 3.11 – Таблица Т3.2

Проанализировав полученные сущности, приходим к выводу, что все таблицы находятся в 3НФ, для того чтобы не было зависимости многие ко многим добавим таблицу «Вид». Нормализацию считаем завершенной.

В результате нормализации получаем схему базы данных в 3НФ (см. рис. 3.12).

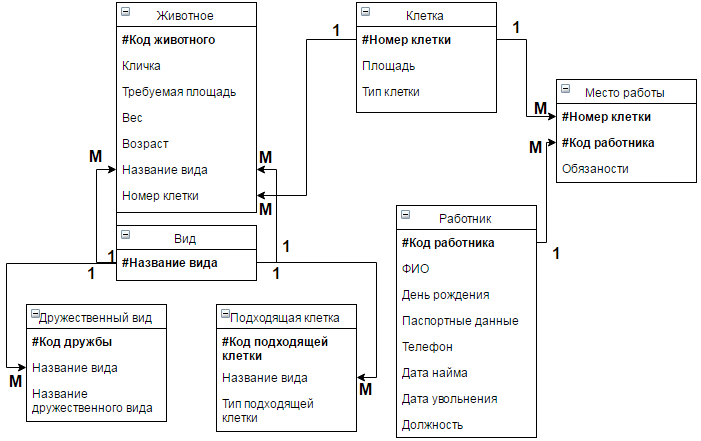


Рисунок 3.12 – Схема базы данных в 3НФ

4 ОПИСАНИЕ ПРОГРАММЫ

4.1 Общие сведения

Разрабатываемый программный продукт создавался в интегрированной среде разработки MS Visual Studio 2015 Community. В качестве СУБД была выбрана Microsoft SQL Server.

Преимущества данной СУБД: простота в использовании (упрощенное создание таблиц, отсутствие сложных операций управления базой данных и т. д.), а также то, что она является одной из самых надежных платформ базы данных.

СУБД поддерживается средой разработки Microsoft Visual Studio 2015 и поэтому нет необходимости в использовании внешних СУБД и их подключении.

Требования к операционной системе, памяти и процессору:

память: не менее 4 ГБ на жестком диске;

процессор: процессор х86, х64: 1,6 ГГц, Pentium III или выше;

4.2 Запуск программы

Для того, чтобы запустить программу, необходимо открыть папку с программой и запустить исполняемый файл «Course\_BD.exe». После этого последует открытие программы.

4.3 Назначение и логическая структура

В постановке задачи, описанной выше, указаны все пункты, которые являются обязательными для исполнения. Данная программа соответствует этим требованиям целиком и полностью.

Главное окно программы выступает навигатором, при помощи которого можно попасть в желаемый раздел (кнопки «Животные», «Работники», «Клетки», «Отчеты», «Статистики»). В окне каждой из категорий есть такие кнопки как «Добавить», «Удалить», «Изменить», в некоторых окнах есть кнопка «Обновить».

Также есть элементы управления TextBox, ComboBox. В окнах присутствует блок отображения данных, представленный элементом DataGridView.

При создании интерфейса брались во внимание законы эргономики, нацеленные на максимальное сохранение времени и минимизацию лишних действий.

Каждая кнопка предоставляет определенный перечень услуг:

а) «Добавить» позволяет добавить информацию о животном/виде/клетке/работнике;

б) «Удалить» удаляет информацию о выбранном объекте, в случае с работником, увольняет его;

в) «Изменить» позволяет изменить информацию о выбранном объекте;

г) «Обновить» обновляет информацию в DataGridView;

д) «Отчет» формирует соответствующий отчет;

е) «Статистика» выводит соответствующую статистику.

TextBox и ComboBox – это элементы поиска, фильтрации и сортировки в которых пользователь может задать параметры, по которым хотел бы найти, отфильтровать нужную информацию.

4.4 Описание физической модели базы данных

В качестве СУБД была выбрана Microsoft SQL Server из-за простоты и надежности. Кроме того, эта СУБД поддерживается Microsoft Visual Studio, поэтому описание физической модели представлены в виде скриншотов из этой программы с описанием.

Структура таблицы «Animals» приведена на рисунке 4.1. На приведенном рисунке перечислены поля таблицы «Animals». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля, кроме поля Name, также не допускают значение NULL. Также в этой таблице поле «Number\_Cell» – это внешний ключ, который ссылается на поле «Id» в таблице «Cells».

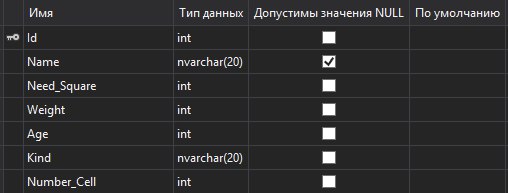


Рисунок 4.1 – Структура таблицы «Animals»

Структура таблицы «Cells» приведена на рисунке 4.2. На приведенном рисунке перечислены поля таблицы «Animals». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL.

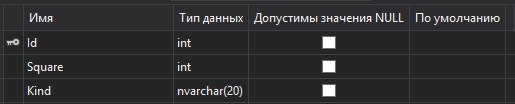


Рисунок 4.2 – Структура таблицы «Cells»

Структура таблицы «cells\_wards» приведена на рисунке 4.3. На приведенном рисунке перечислены поля таблицы «cells\_wards». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL. Также в этой таблице поле «Id\_cell» – это внешний ключ, который ссылается на поле «Id» в таблице «Cells», а также внешним ключом является поле «Id\_ward», который ссылается на поле «Id» в таблице «Wards».

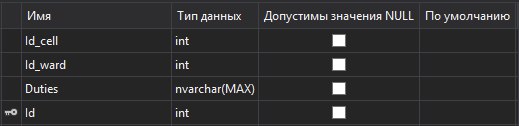


Рисунок 4.3 – Структура таблицы «cells\_ward»

Структура таблицы «Kinds\_Animals» приведена на рисунке 4.4. На приведенном рисунке перечислено поле таблицы «Kinds\_Animals», которое является первичным ключом, на которое наложено ограничение непустое и уникальное.

https://pp.vk.me/c638821/v638821754/15145/FB0T5UJb_ng.jpg

Рисунок 4.4 – Структура таблицы «Kinds\_Animals»

Структура таблицы «Friends» приведена на рисунке 4.5. На приведенном рисунке перечислены поля таблицы «Friends». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL. Также в этой таблице поле «Kind» – это внешний ключ, который ссылается на поле «Id» в таблице «Cells».

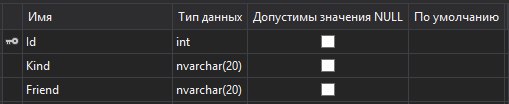


Рисунок 4.5 – Структура таблицы «Friends»

Структура таблицы «Good\_Cells» приведена на рисунке 4.6. На приведенном рисунке перечислены поля таблицы «Good\_Cells». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля также не допускают значение NULL. Также в этой таблице поле «Kind» – это внешний ключ, который ссылается на поле «Id» в таблице «Cells».

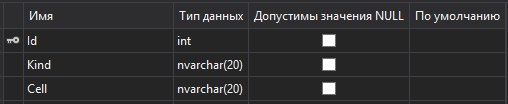


Рисунок 4.6 – Структура таблицы «Good\_Cells»

Структура таблицы «Wards» приведена на рисунке 4.6. На приведенном рисунке перечислены поля таблицы «Wards». Как первичный ключ обозначено поле «Id», на которое наложено ограничение непустое и уникальное. Остальные поля, кроме полей «Phone», «Work\_Finish», также не допускают значение NULL.

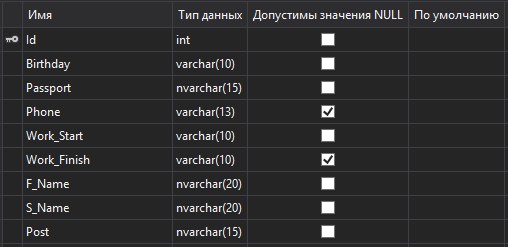


Рисунок 4.7 – Структура таблицы «Wards»

4.5 Описание программной реализации

При запуске программы открывается главное окно программы (см. рис. 4.8). Данное окно является навигационным, и пользователь имеет возможность выбрать интересующий его раздел («Животные», «Работники», «Клетки», «Отчёты», «Статистика»).

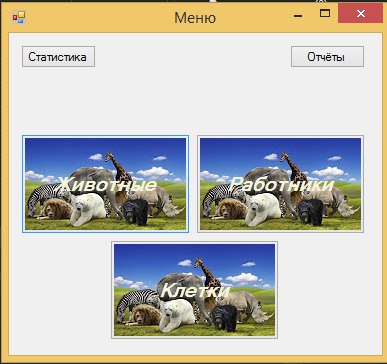


Рисунок 4.8 – Главное окно программы

При нажатии на кнопку «Клетки» пользователь попадает в соответствующее окно (см. рис. 4.9). В данном окне пользователь может просматривать все клетки, которые доступны в зоопарке.

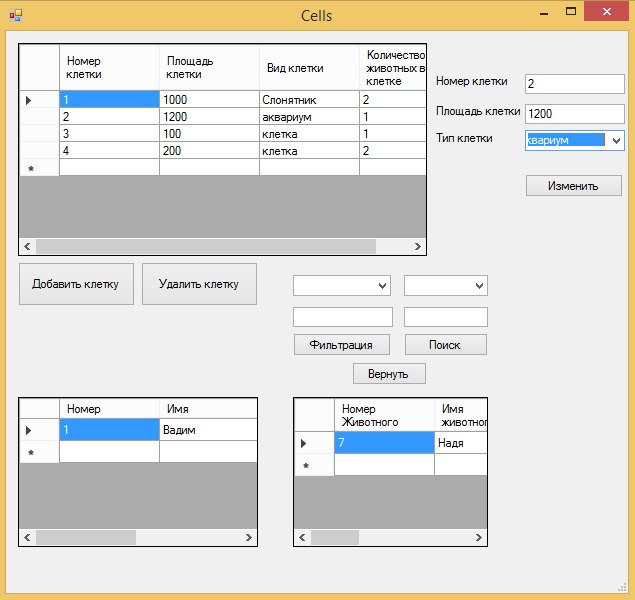


Рисунок 4.9 – Окно клеток

Также в данном окне пользователь имеет возможность осуществлять поиск по клеткам, фильтровать их и производить сортировку по возрастанию и убыванию (при поиске подходящие подсвечивается, при фильтрации остается только необходимое). Фильтрация поддерживает частичное совпадение. При нажатии кнопки «Добавить клетку», откроется окно добавления клетки (см. рис. 4.10). На нем есть 2 TextBox, 1 ComboBox и две кнопки: «ОК» и «Назад».

При нажатии кнопки «ОК» запись будет добавлена в базу данных, а при нажатии на кнопку «Назад» вас вернёт на окно клеток.

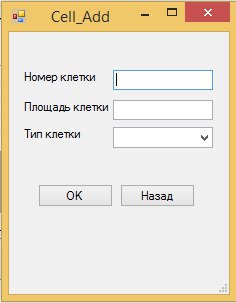


Рисунок 4.10 – Окно добавление клетки

При нажатии на клетку в таблице, получаете перечень работников, работающих в этой клетки, и перечень животных, живущих в этой клетке. Также вы можете изменить данные клетки и, при нажатии на кнопку «Изменить», изменения заносятся в базу данных.

При нажатии кнопки «Удалить клетку», запись о выбранной клетке будет удалена.

При выборе категории «Работники» на главной форме, пользователь попадает в соответствующее окно (см. рис. 4.11). В данном окне пользователь может просматривать всех работников, которые работают в зоопарке.

Кнопки «Добавить работника», «Изменить», «Удалить работника» выполняют аналогичные функции соответствующим кнопкам в окнах клеток.

Клетки, в которых работает выбранный работник, отображается в DataGridView. Также можно изменить обязанность по клетке с помощью кнопки «Изменить обязанность». Добавление и удаление клетки, в которой будет работать работник, осуществляется с помощью кнопок «Добавить клетку», «Удалить клетку».

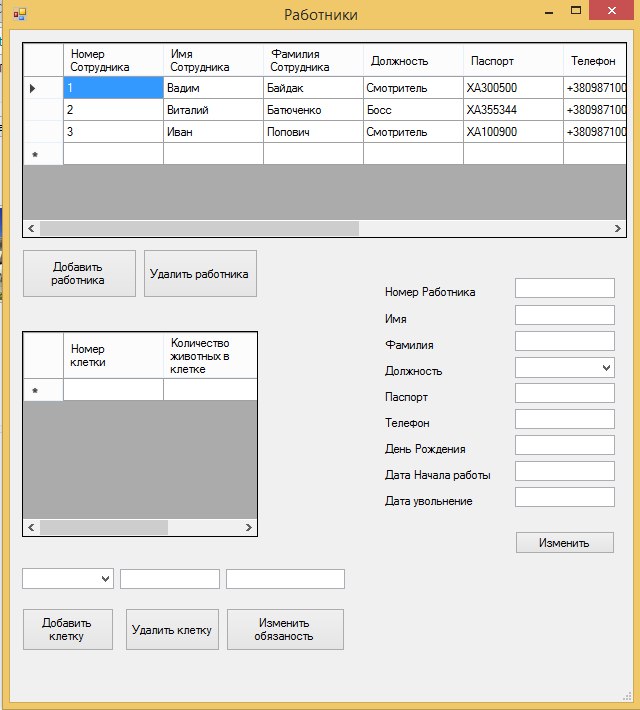


Рисунок 4.11 – Окно работников

При выборе категории «Животные» на главной форме, пользователь попадает в соответствующее окно (см. рис 4.12). В данном окне пользователь может просматривать всех животных, которые находятся в зоопарке. Кнопка «Добавить животного» выполняет аналогичные функции соответствующим кнопкам. Кнопка «Обновить» обновляет информацию в DataGridView. Кнопка «Виды животных» открывает окно видов животных (см. рис. 4.13).

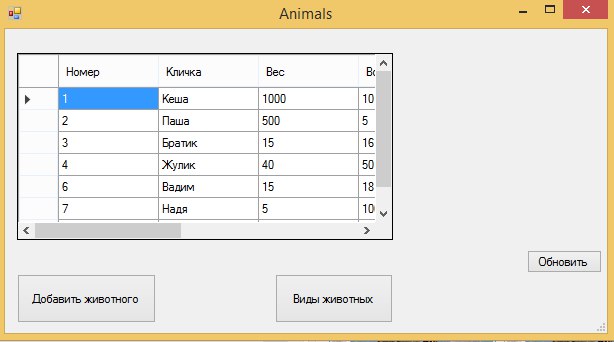


Рисунок 4.12 – Окно животных

Кнопки добавления и удаления работают аналогично.

При переходе на статистику на главной форме мы получаем три круговые диаграммы: соотношение количества типов клеток (см. рис. 4.14), количества видов животных (см. рис. 4.15), количества должностей (см. рис. 4.16).

При нажатии кнопки «Отчёты» на главном окне, переходите на окно отчётов (см. рис. 4.17). Сами отчёты приведены на рис. 4.18 и 4.19.

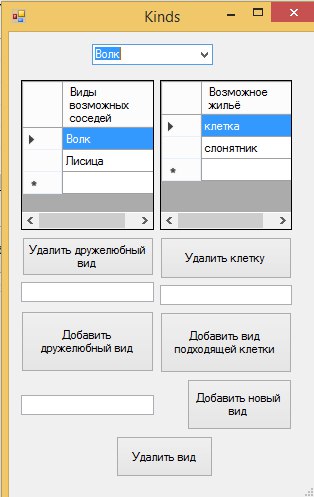


Рисунок 4.13 – Окно видов животных

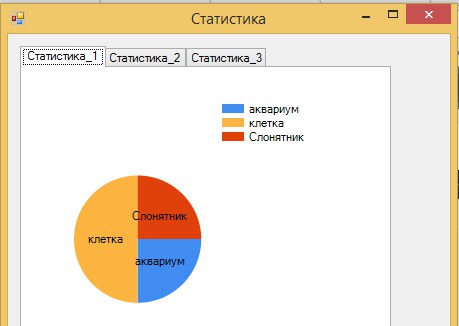


Рисунок 4.14 – Окно статистики по типам клеток

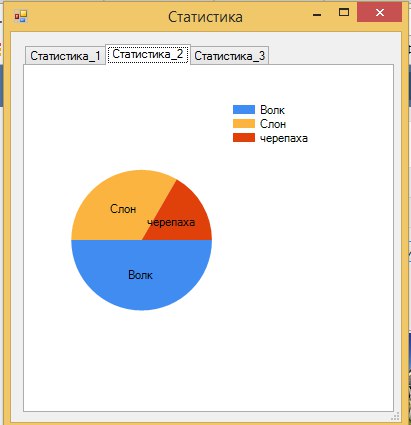


Рисунок 4.15 – Окно статистики по видам животных

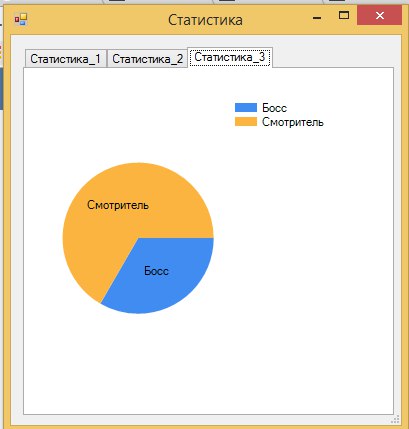


Рисунок 4.16 – Окно статистики по должностям

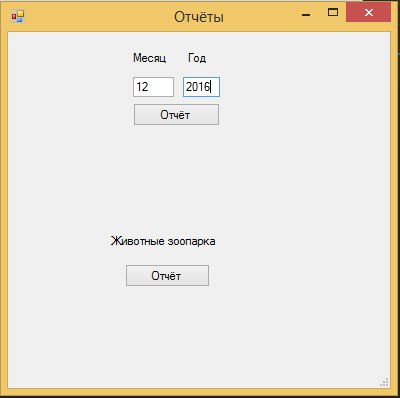


Рисунок 4.17 – Окно отчётов

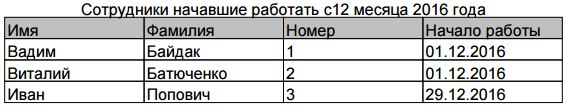


Рисунок 4.18 – Отчёт №1

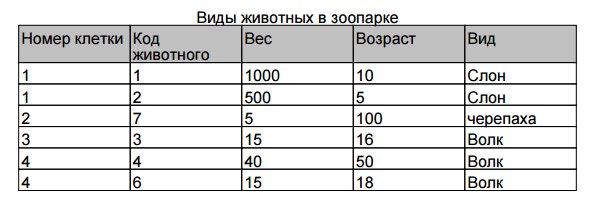


Рисунок 4.18 – Отчёт №2

4.6 Задача автоматизации

В качестве задачи автоматизации была выбрана функция автоматического распределения животных по клеткам. При заполнении данных в окне добавление животного и выборе в поле номер клетки, происходит добавление информации о новом животном в базу данных, при этом номер клетки выберется автоматически. Приоритеты по клеткам: с начало берутся клетки, где живут только виды нового животного, потом виды с которыми может жить новое животное, а потом пустые клетки, также при нескольких вариантах, берётся клетка в которой меньше осталось свободной площади. Код при добавлении с автоматическим заселением:

SqlConnection sqlconn = new SqlConnection(ConnectionString);

sqlconn.Open();

int i1 = -1;

int s1=0;

int i2 = -1;

int s2=0;

int i3 = -1;

for(int i = 1; i < comboBox\_cell.Items.Count; ++i)

{

string zapros = "Select F=(Select Square From Cells Where Id="+

comboBox\_cell.Items[i]+"), S=Coalesce(Sum(Need\_Square),0), G=(Select Count(Id)"+" From Animals Where Number\_Cell="

+ comboBox\_cell.Items[i] + ")-(Select Count(Id) From Animals Where Number\_Cell="

+ comboBox\_cell.Items[i]+" And Kind=N'"+

comboBox\_kind.Text+"') FROM Animals Where Number\_Cell="+comboBox\_cell.Items[i]+";";

SqlDataAdapter oda = new SqlDataAdapter(zapros, sqlconn);

DataTable dt = new DataTable();

oda.Fill(dt);

int f = Convert.ToInt32(dt.Rows[0]["F"]);

int s = Convert.ToInt32(dt.Rows[0]["S"]);

int g = Convert.ToInt32(dt.Rows[0]["G"]);

int cur= Convert.ToInt32(comboBox\_cell.Items[i]);

if (s == 0)

{

if (i3 == -1)

{

i3 = cur;

}

}

else if (g == 0)

{

if (i1 == -1)

{

i1 = cur;

s1 = f - s;

}

else if (s1 > f - s)

{

s1 = f - s;

i1 = cur;

}

}

else

{

if (i2 == -1)

{

i2 = cur;

s2 = f - s;

}

else

{

if (s2 > f - s)

{

s2 = f - s;

}

}

}

}

int cur1 = -1;

if (i1 != -1)

{

cur1= i1;

}

else if (i2 != -1)

{

cur1 = i2;

}

else if (i3 != -1)

{

cur1 = i3;

}

string sql = "INSERT INTO Animals VALUES(" + textBox\_id.Text + " ,N'" + textBox\_name.Text + "' ," + textBox\_square.Text + " ,'" + textBox\_weight.Text + "' ," + textBox\_age.Text + " ,N'" + comboBox\_kind.Text + "' ," + cur1 + ");";

if (sqlconn.State != ConnectionState.Open)

sqlconn.Open();

SqlCommand command = new SqlCommand(sql, sqlconn);

int x = command.ExecuteNonQuery();

sqlconn.Close();

Close();

4.7 SQL запросы в программе

Запрос для заполнения DataGridView информацией о клетках:

SELECT Cells.Id AS 'Номер клетки', Cells.Square AS 'Площадь клетки', Cells.Kind AS 'Вид клетки', 'Количество животных в клетке'=(SELECT Count(Animals.Id) FROM Animals WHERE Animals.Number\_Cell=Cells.Id) FROM Cells GROUP BY Cells.Id, Cells.Square, Cells.Kind;

Запрос, который добавляет новую клетку в таблицу Cells

INSERT INTO Cells(Id, Square, Kind) VALUES(" + textBox\_name.Text + " ," + textBox\_S.Text + " ,N'" + textBox\_Kind.Text + "');

Запрос, который обновляет информацию о клетки в таблице Cells:

Update Cells SET Id=" + textBox\_name.Text + " ,Square=" + textBox\_S.Text + " ,Kind =N'" + textBox\_Kind.Text + "' WHERE Id=" + cur\_index+";

Запрос, который удаляет информацию о клетки из таблицы Cells:

DELETE From Cells Where Id=" + cur\_index + ";

Запрос для поиска с частичным совпадением:

SELECT Cells.Id AS 'Номер клетки', Cells.Square AS 'Площадь клетки', Cells.Kind AS 'Вид клетки', 'Количество животных в клетке'=(SELECT Count(Animals.Id) FROM Animals, Cells WHERE Animals.Number\_Cell=Cells.Id) FROM Cells Where Cells.Kind Like N'%" + textBox\_search.Text + "%' GROUP BY Cells.Id, Cells.Square, Cells.Kind;

Запрос для формирования статистики по количеству клеток по типу клеток:

Select Kind , Count(Id) From Cells Group By Kind

Запрос формирования отчёта о работниках, которых наняли в определённый месяц:

SELECT F\_Name As 'Имя', S\_Name As 'Фамилия', Id As 'Номер', Work\_Start As 'Начало работы' From Wards Where Work\_Start Like '\_\_."+textBox\_month.Text+"."+textBox\_year.Text+"'

ВЫВОДЫ

В данной курсовой работе была разработана информационная система «Размещение животных» для оптимизации и автоматизации процесса заселения животных по клеткам. Отличительной особенностью этой программы является использование базы данных, реализованной с помощью Microsoft SQL Server.

Интерфейс программы прост, понятен и удобен в использовании. Этой программой без труда сможет пользоваться даже неопытный пользователь.

Данная информационная система может быть использована, пусть и в качестве прототипа, в реальных магазинах канцелярских товаров, поскольку главной задачей было поставлено сделать программу максимально соответствующей данной предметной области. Данная программа позволяет упростить процесс расселения животных по клеткам.

Работа над информационной системой требовала детального разбора будущей базы данных, в ходе которого были изучены различные этапы оптимизации базы данных и проекта в целом. Среди них - ER-диаграмма, нормализация, концептуальный анализ и т. д.

Все это в совокупности позволило получить информационную систему в соответствие с поставленными задачами. В ходе работы над данным проектом были приобретены и отработаны навыки работы с различными СУБД, SQL- запросами и закреплены знания в направлении .NET.

СПИСОК ЛИТЕРАТУРЫ

1. Дейт К. Дж. Введение в системы баз данных. 6-е изд./ К. Дж. Дейт –М.: Вильямс, 2010. – 848с.

2. Зоопарк. Материалы из Википедии — свободной энциклопедии [Электронный ресурс] – Режим доступу: www/URL: https://ru.wikipedia.org/wiki/Зоопарк

3. Уинкуп С. Microsoft SQL Server 6.5/ С. Уинкуп; СПб: БХВ - Москва, 2011. - 896 c.

4. Зоопарк Харькова. Режим доступу: www/URL: https:// http://zoo.kharkov.ua