

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: _____

ПРЕПОДАВАТЕЛЬ:

Старший преподаватель / _____ / _____ / Е. В. Павлов
(должность, учёная степень, звание) (подпись) (дата защиты) (инициалы, фамилия)

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

«ОПРЕДЕЛЕНИЕ СТРУКТУР И ЭЛЕМЕНТОВ ДАННЫХ
ПРЕДМЕТНОЙ ОБЛАСТИ. СОСТАВЛЕНИЕ СЛОВАРЯ ДАННЫХ»

ПО КУРСУ: «ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СИСТЕМ»

РАБОТУ ВЫПОЛНИЛ (-А) СТУДЕНТ (-КА): 4936 / М. Р. Назаров
(номер группы) (инициалы, фамилия)

/ _____ / 11.10.2021
(подпись студента) (дата отчета)

ВВЕДЕНИЕ

Для демонстрации процессов, которые происходят в системе, используют DFD, в свою очередь для представления данных и отношений между ними применяют модели данных. Одна из наиболее широко используемых моделей данных — диаграмма «сущность-связь» (*Entity-Relationship Diagram*, далее ERD), которую также используют в качестве инструмента для анализа требований. Основное назначение ERD заключается в анализе компонентов данных системы и их связей для определения логической или физической (реализации) структуры базы данных. Все элементы данных, которые представлены на ERD, подробно описывает словарь данных (*Data Dictionary*), предназначенный для сбора, организации (систематизации) и документирования конкретных фактов о системе.

Поскольку определения данных часто повторно используются в других приложениях, соответственно применение единообразных определений данных снижает вероятность возникновения ошибок интеграции и интерфейса. Таким образом, словарь данных является одним из атрибутов повышения качества разработки, так как служит для определения критериев проверки данных, облегчает поиск необходимой информации, позволяет избежать ненужных повторов и ошибок, связанных с тем, что участники проекта по-разному понимают ключевые данные.

Цель работы заключается в изучение способов описания информации об используемых в системе сущностях данных и получение практических навыков составления словаря данных.

Задачи лабораторной работы.

Для достижения поставленной в лабораторной работе цели необходимо выполнить следующие задачи:

Составить даталогическую схему реляционной модели данных (на основе ER-диаграммы):

- 1) Допускается описание *фрагмента* даталогической модели:
 - не менее 6 ключевых сущностей (объектов предметной области);
 - суммарно не менее 24 описательных (неключевых) атрибутов.
- 2) Набор атрибутов для каждой сущности должен быть задан в соответствии

с требованиями как предметной области, так и задач, для выполнения которых предназначена система;

- 3) На ER-диаграмме должны быть в явном виде указаны первичные (PK) и внешние ключи (FK);
- 4) ER-диаграмма должна содержать одно или несколько отношений «многие ко многим» (без реструктуризации);
- 5) Отдельно должна быть представлена ER-диаграмма с реструктуризацией всех отношений «многие ко многим» на два отношения «один ко многим» (связующие таблицы также могут содержать дополнительные атрибуты, если в этом есть необходимость с точки зрения задач системы);
- 6) Представленная на ER-диаграмме информация, должна находиться в 3NF.

Определить элементы выделенных сущностей в словаре данных:

- 7) Словарь данных должен отражать минимальные критерии для проверки элементов данных (атрибуты сущностей) с точки зрения их отображения, хранения и выполняемых над ними операций; таким образом:
- 8) В словаре данных необходимо указать точную или предполагаемую *длину элементов в символах* (не в байтах) или диапазон числовых значений и соответствующий заданной длине тип данных. Если длина элемента неизвестна, то ограничение длины должно быть задано соответствующим типом данных. При этом рекомендуется использовать тип данных наименьшего размера, который гарантирует хранение всех возможных значений. Принятые в работе типы данных должны быть раскрыты в приложении к отчету;
- 9) Также требуется указать список разрешенных значений или значений по умолчанию, если это необходимо. В случае неочевидного использования атрибутов в системе, каждый такой атрибут должен быть сопровожден соответствующим пояснением.

Индивидуальный вариант задания:

41 | Сервис для владельцев домашних животных

1. Даталогическая модель базы данных

Задание на работу допускает построение неполной модели, таким образом, на рисунках 1 и 2 представлены *фрагменты* даталогической модели с отношением «многие-ко-многим» и его представлением в базе данных соответственно.

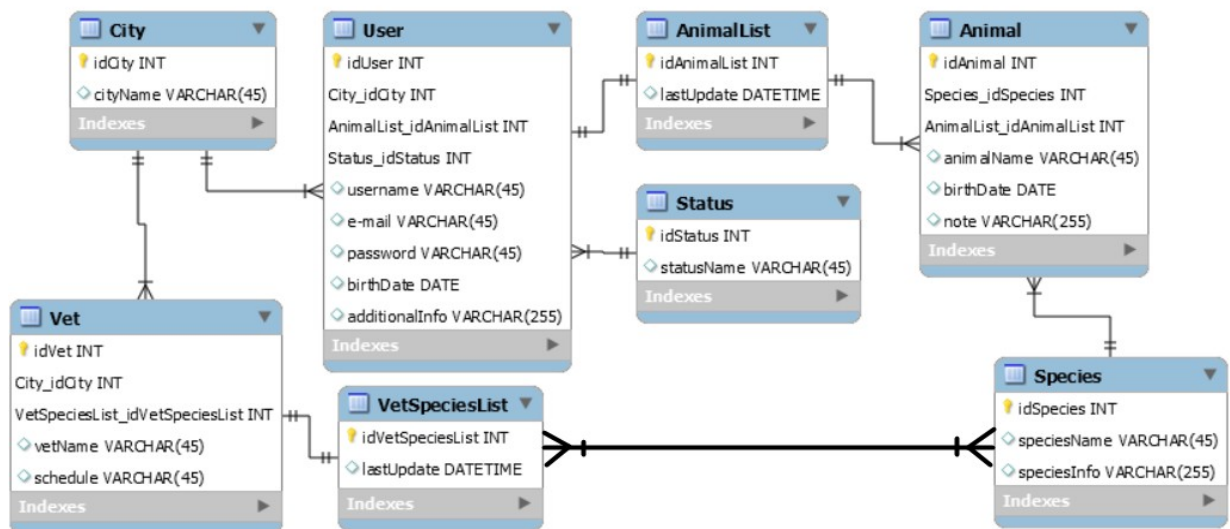


Рисунок 1 — Фрагмент даталогической модели для системы
«Сервис для владельцев домашних животных»

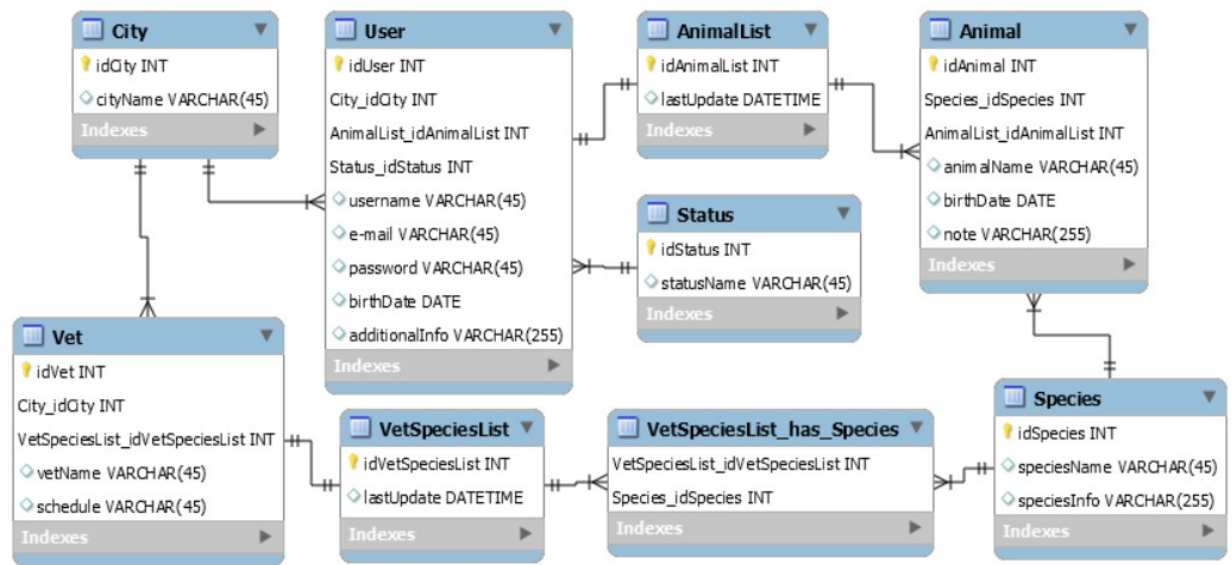


Рисунок 2 — Фрагмент даталогической модели
с реструктуризацией отношений «многие-ко-многим»

2. Словарь данных

Принятые в словаре данных обозначения представлены в *ПРИЛОЖЕНИИ А*.

Таблица 1 — Фрагмента словаря данных для рассматриваемой системы

Структура или элемент данных	Тип данных	Длина	Значение
City			
idCity	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
cityName	VARCHAR	45	Может содержать только символы латинского алфавита, подчеркивание и цифры

Структура или элемент данных	Тип данных	Длина	Значение
User			
idUser	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
Status_idStatus	INT	45	(FK) Внешний ключ — содержит идентификатор статуса
City_idCity	INT	10	(FK) Внешний ключ — содержит идентификатор города
AnimalList_idAnimalList	INT	10	(FK) Внешний ключ — содержит идентификатор списка животных
username	VARCHAR	45	Может содержать только символы латинского алфавита, подчеркивание и цифры
e-mail	VARCHAR	45	Должен соответствовать стандарту RFC 5322
password	VARCHAR	45	Может содержать символы латинского алфавита, числа и символы из следующего после двоеточия списка: ! @ # \$ % ^ & ? * _ Остальные символы, включая пробел, запрещены
birthDate	DATE	—	Дата рождения
additionalInfo	VARCHAR	255	Дополнительная информация о пользователе

Структура или элемент данных	Тип данных	Длина	Значение
Status			
idStatus	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
statusName	VARCHAR	45	Может содержать только символы латинского алфавита, подчеркивание и цифры

Структура или элемент данных	Тип данных	Длина	Значение
AnimalList			
idAnimalList	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
lastUpdate	DATETIME	—	Дата последнего обновления списка

Структура или элемент данных	Тип данных	Длина	Значение
Animal			
idAnimal	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
Specied_idSpecies	INT	10	(FK) Внешний ключ — содержит идентификатор вида животного
AnimalList_idAnimalList	INT	10	(FK) Внешний ключ — содержит идентификатор списка животных
animalName	VARCHAR	45	Имя питомца
bitrhDate	DATE	—	Дата рождения
note	VARCHAR	255	Примечание о животном

Структура или элемент данных	Тип данных	Длина	Значение
Species			
idSpecies	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
speciesName	VARCHAR	45	Латинское имя вида. Допускаются только латинские символы и точки
speciesInfo	VARCHAR	255	Информация о виде

Структура или элемент данных	Тип данных	Длина	Значение
VetSpeciesList			
idVetSpecies	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
lastUpdate	DATETIME	—	Дата последнего обновления списка

Структура или элемент данных	Тип данных	Длина	Значение
VetSpeciesList_has_Species			
VetSpeciesList_idVetSpeciesList	INT	10	(PK, FK1) Составной внешний ключ — ссылка на часть составного первичного ключа «idVetSpeciesList» таблицы «VetSpeciesList»
Species_idSpecies	INT	10	(PK, FK2) Составной внешний ключ — ссылка на часть составного первичного ключа «idSpecies» таблицы «Species»

Структура или элемент данных	Тип данных	Длина	Значение
Vet			
idVet	INT	10	(PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1
City_idCity	INT	10	(FK) Внешний ключ — содержит идентификатор города
VetSpeciesList_idVetSpeciesList	INT	10	(FK) Внешний ключ — содержит идентификатор списка специализации клиники по видам животных
vetName	VARCHAR	45	Название вет-клиники

schedule	VARCHAR	45	Расписание клиники
----------	---------	----	--------------------

ЗАКЛЮЧЕНИЕ

В результате выполнения настоящей лабораторной работы рассмотрены два основных метода моделирования данных: ER-диаграмма и словарь данных.

В соответствии с формулировкой задания составлен фрагмент даталогической модели для системы «Сервис для владельцев домашних животных». Модель построена на основе ER-диаграммы (нотация Crow's Foot) и включает в себя следующие сущности:

1. City
2. User
3. AnimalList
4. Animal
5. Species
6. Status
7. VetSpeciesList
8. Vet

Представленная на ER-диаграмме информация, находится в 3NF.

Составлен словарь данных, который включает в себя минимальные критерии для проверки элементов данных, такие как тип данных, длина и возможные или заданные значения.

Представленные в работе методы моделирования данных не являются взаимоисключающими, так как выполняют разные задачи. Однако словарь данных может быть дополнен соответствующими ER-диаграммами (как и наоборот).

Таким образом, можно заключить, что выполненная работа соответствует поставленной задаче и отвечает всем сформулированным в задании требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вигерс, Карл. Разработка требований к программному обеспечению = Software Requirements: пер. с англ.; 3-е издание, дополненное / Карл Виггерс, Джой Битти — СПб.: Издательство «БНВ», 2020. — 736 с.: ил.
2. What is Entity Relationship Diagram (ERD)? [Электронный ресурс]. — Visual Paradigm, 2021. — URL: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/> (дата обращения: 20.09.2021)
3. SQL Data Types for MySQL, SQL Server, and MS Access [Электронный ресурс]. — W3Schools, 1999-2021. — URL: https://www.w3schools.com/sql/sql_datatypes.asp (дата обращения: 20.09.2021)
4. MySQL 8.0 Reference Manual: Chapter 11 Data Types [Электронный ресурс]. — Oracle Corporation, 2021. — URL: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html> (дата обращения: 20.09.2021)

ПРИЛОЖЕНИЕ А

Принятые в работе типы данных

1	TINYINT	Целочисленный тип размером 1 байт Со знаком от -128 до 127, без знака от 0 до 255
2	SMALLINT	Целочисленный тип размером 2 байта Со знаком от -32 768 до 32 767, без знака от 0 до 65 535
3	MEDIUMINT	Целочисленный тип размером 3 байта Со знаком от -8 388 608 до 8 388 607, без знака от 0 до 16 777 215
4	INT	Целочисленный тип размером 4 байта Со знаком от -2 147 483 648 до 2 147 483 647, без знака от 0 до 4 294 967 295
5	BIGINT	Целочисленный тип размером 8 байт Со знаком от -2^{63} до $2^{63}-1$, без знака от 0 до $2^{64}-1$
6	DECIMAL	Тип с фиксированной точкой DECIMAL (size, d), где size — общее количество цифр (максимум 65), d — количество цифр после точки (максимальное значение для d — 30). Значения по умолчанию — 10 (для size) и 0 (для d).
7	FLOAT	Тип с плавающей точкой размером 4 байта В текущих версиях данный тип выражается как FLOAT (n), где n определяет, будет ли значение сохранено как FLOAT или преобразовано в DOUBLE. При n от 0 до 23 значение хранится в виде 4-байтового столбца с одинарной точностью, при n от 24 до 53 в виде 8-байтового столбца с двойной точностью (тип DOUBLE). По умолчанию значение n равно 53 (двойная точность). Диапазон значений для одинарной точности: от -3.40E+38 до -1.18E-38, 0 и от 1.18E-38 до 3.40E+38 Диапазон значений для двойной точности: от -1.79E+308 до -2.23E-308, 0 и от 2.23E-308 до 1.79E+308
8	DOUBLE	Тип с плавающей точкой размером 8 байт (двойная точностью)
9	BIT	Целочисленный тип данных, который может принимать значения 0, 1 или NULL (используется для хранения битовых значений) BIT (n), где n — количество битов (от 1 до 64)
10	DATE	Хранение даты в формате YYYY-MM-DD Поддерживает диапазон от 1000-01-01 до 9999-12-31
11	DATETIME	Хранение даты и времени в формате YYYY-MM-DD hh:mm:ss Поддерживает диапазон от 1000-01-01 00:00:00 до 9999-12-31 23:59:59
12	TIME	Хранение значения времени в формате hh:mm:ss Поддерживает диапазон от -838:59:59 до 838:59:59 Используется не только для представления времени дня (которое должно быть меньше 24 часов), но и для прошедшего времени или временного интервала между двумя событиями
13	YEAR	Хранение значения года в формате YYYY Тип YEAR занимает 1 байт, поэтому поддерживает диапазон от 1901 до 2155 и 0000 (MySQL 8.0 не поддерживает задание года в двузначном формате)

14	CHAR	Строка фиксированной длины (может содержать буквы, цифры и специальные символы). CHAR (size), где size — длина строки в символах (от 0 до 255, по умолчанию 1)
15	VARCHAR	Строка переменной длины (может содержать буквы, цифры и специальные символы). VARCHAR (size), где size — максимальная длина строки в символах (от 0 до 65535)
16	TINYTEXT	Хранение строки максимальной длины в 255 символов
17	TEXT	Хранение строки максимальной длины в 65 535 символов
18	MEDIUMTEXT	Хранение строки максимальной длины в 16 777 215 символов
19	LONGTEXT	Хранение строки максимальной длины в 4 294 967 295 символов

20	BINARY	Аналог CHAR, но данные хранятся в виде бинарной строки (бинарная строка состоит только из символов 0 и 1) BINARY (size), где size — длина строки в байтах (от 0 до 255, по умолчанию 1)
21	VARBINARY	Аналог VARCHAR, но данные хранятся в виде бинарной строки VARBINARY (size), где size — максимальная длина строки в байтах (от 0 до 65535)
22	TINYBLOB	Хранение BLOB размером до 255 байт включительно
23	BLOB	Хранение BLOB размером до 65 535 байт включительно
24	MEDIUMBLOB	Хранение BLOB размером до 16 777 215 байт включительно
25	LOB	Хранение BLOB размером до 4 294 967 295 байт включительно

26	ENUM	Специальный строковый тип, который принимает только одно значение из фиксированного списка значений. В списке ENUM, который определяется во время создания таблицы в базе данных, можно задать до 65 535 значений. Все недопустимые значения (которых нет в списке) при добавлении заменяются на пустые строки.
----	------	--