

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ИС

КУРСОВАЯ РАБОТА
по дисциплине «Управление данными»
Тема: Разработка базы данных для автоматизации деятельности
выставки собак

Студент гр. 6373

Васильев М.С.

Преподаватель

Татарникова Т.М.

Санкт-Петербург

2018

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (КУРСОВОЙ ПРОЕКТ)

Студент Васильев М.С.

Группа 6373

Тема работы: Разработка базы данных для автоматизации деятельности выставки собак

Содержание пояснительной записки: Задание на курсовую работу, «Содержание», «Введение», «Анализ предметной области», «Создание ER-модели», «Нормализация бд», «Описание БД в СУБД», «Разграничение прав», «Заключение», «Список использованных источников».

Предполагаемый объем пояснительной записки:

Не менее 15 страниц

Дата выдачи задания: 15.09.2018

Дата сдачи работы: 19.11.2018

Дата защиты работы:

Студент

Васильев М.С.

Преподаватель

Татарникова Т.М

АННОТАЦИЯ

В данной курсовой работе производится проектирование реляционной базы данных для сайта, обеспечивающего поддержку выставки собак. На практике рассматривается анализ предметной области базы данных и логическое проектирование. По результатам курсовой работы получена логическая схема реляционной БД в третьей нормальной форме и получен рабочий прототип сайта, для обеспечения поддержки собачьих выставок.

SUMMARY

This course project presents process of designing relational database, for supporting dog expositions. We examined on a specific example database subject analysis and logical design. As a result, we got a logical scheme of relational database reduced to a third normal form and working prototype for site to support dog expositions.

СОДЕРЖАНИЕ

	Введение	5
1.	Анализ предметной области	6
1.1.	Выбор подхода	6
1.2.	Выбор оперируемых объектов	6
1.3.	Выбор групп пользователей системы	6
2.	Создание ER-модели	7
3.	Нормализация базы данных	9
3.1.	Первая нормальная форма	9
3.2.	Вторая нормальная форма	9
3.3.	Третья нормальная форма	9
4.	Описание БД в СУБД	10
4.1.	Таблицы	10
4.2.	Запросы с листингами SQL	11
4.3.	Реализация интерфейса к БД	13
5.	Разграничение прав	15
5.1.	Доступы каждой группы	15
5.2.	Детали реализации	15
	Заключение	16
	Список использованных источников	17

ВВЕДЕНИЕ

В теории БД методология проектирования рассматривается как совокупность человеко-машинных инструментов и средств, применяемых для последовательной разработки проекта структуры баз данных. Совокупность процедур проектирования централизованной БД, которые можно объединить в четыре этапа.

На этапе формулирования и анализа требований устанавливаются цели организации, определяются требования к БД. Эти требования документируются в форме доступной конечному пользователю и проектировщику БД. Обычно при этом используется методика интервьюирования персонала различных уровней управления.

Этап концептуального проектирования заключается в описании и синтезе информационных требований пользователей в первоначальный проект БД. Результатом этого этапа является высокоуровневое представление информационных требований пользователей на основе различных подходов.

В процессе логического проектирования высокоуровневое представление данных преобразуется в структуру используемой СУБД. Полученная логическая структура БД может быть оценена количественно с помощью различных характеристик (число обращений к логическим записям, объем данных в каждом приложении, общий объем данных и т.д.). На основе этих оценок логическая структура может быть усовершенствована с целью достижения большей эффективности.

На этапе физического проектирования решаются вопросы, связанные с производительностью системы, определяются структуры хранения данных и методы доступа.

В данной работе мы рассмотрим первые три этапа на примере проектирования некоторой абстрактной реляционной базы данных.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Выбор подхода

В настоящее время при проектировании БД используют два подхода. Первый из них основан на стабильности данных, что обеспечивает наибольшую гибкость и адаптируемость к используемым приложениям. Применение такого подхода целесообразно в тех случаях, когда не предъявляются жесткие требования к эффективности функционирования (объем памяти и время поиска), существует большое количество разнообразных задач с изменяемыми и непредсказуемыми запросами.

Другой подход базируется на стабильности процедур запросов к БД и является предпочтительным при жестких требованиях к эффективности функционирования, особенно это касается быстродействия.

Мною был выбран системный подход, так как жестких требований к эффективности функционирования не было представлено.

1.2. Выбор оперируемых объектов

В качестве оперируемых объектов были выбраны следующие неотъемлемые составляющие собачьих выставок: собаки, участники, эксперты, ринги и призы.

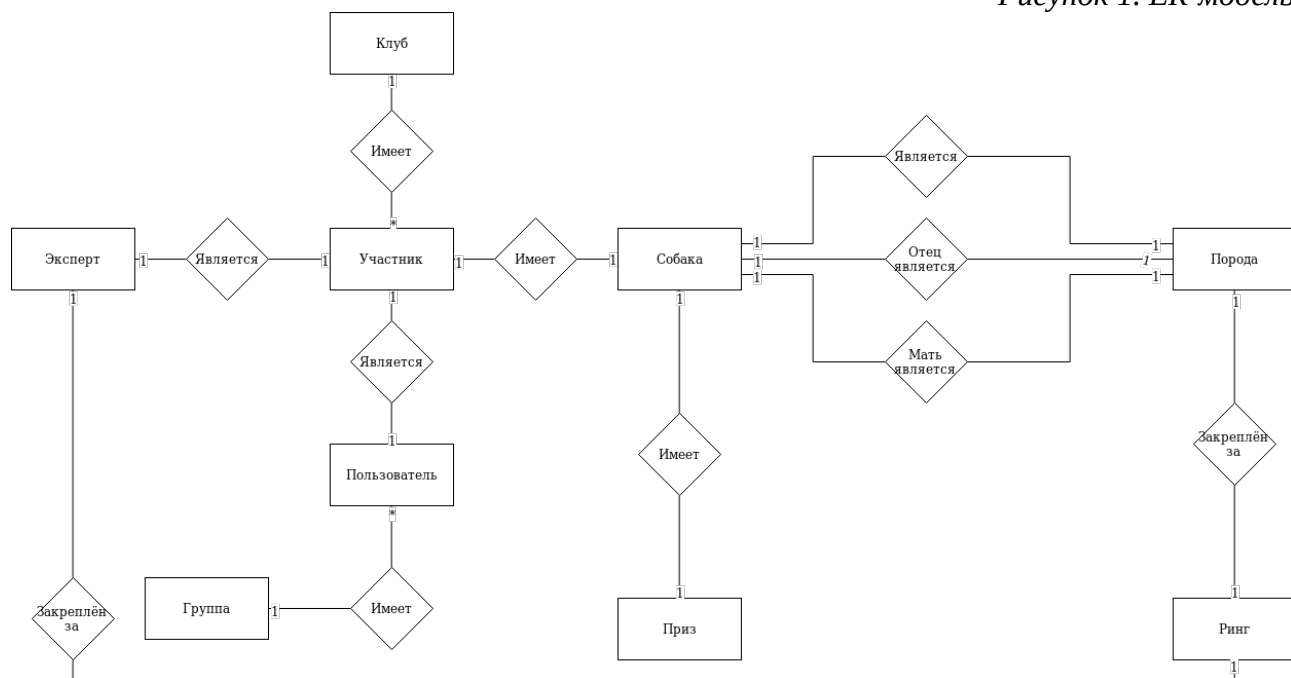
1.3. Выбор групп пользователей системы

В качестве групп пользователей были выбраны администраторы и обычные пользователи, так как конечной целью данной системы является заполнение её данными администратором, и получением данных из неё пользователем.

2. СОЗДАНИЕ ER-МОДЕЛИ

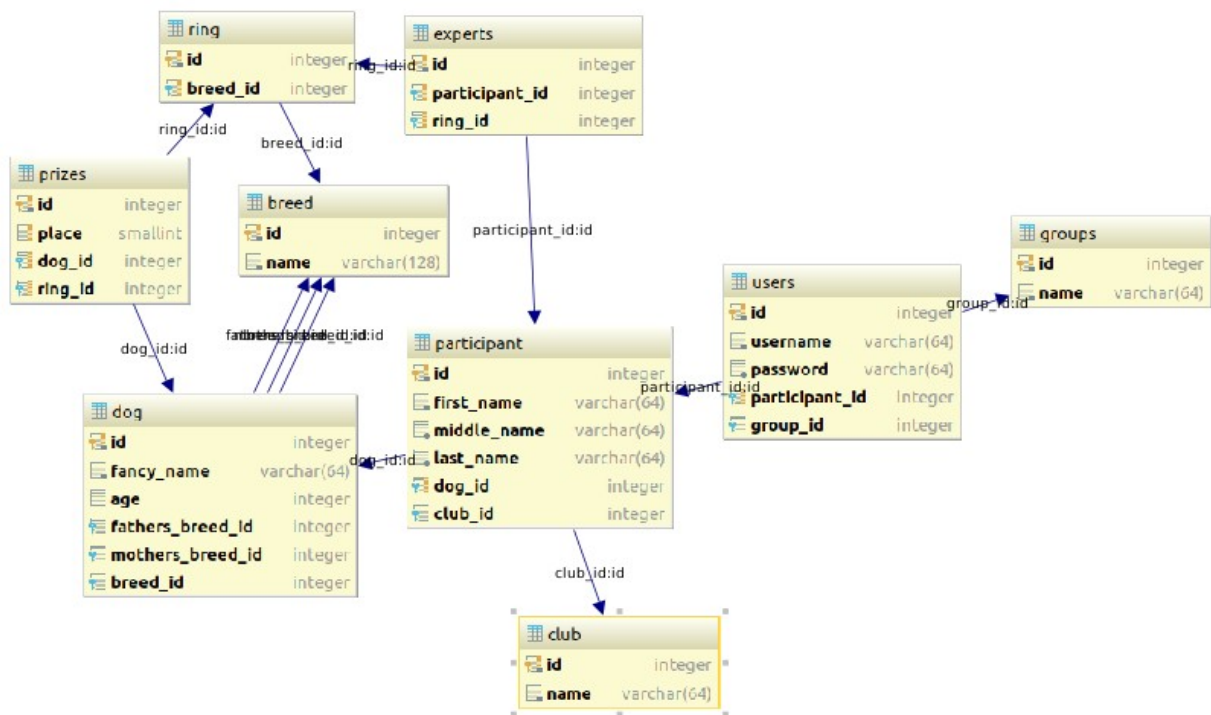
В соответствии с предметной областью была составлена следующая ER-диаграмма.

Рисунок 1. ER-модель



А также, диаграмма с уточненными атрибутами и названиями ключей.

Рисунок 2. Уточнённая ER-модель



3. НОРМАЛИЗАЦИЯ БАЗЫ ДАННЫХ

3.1. Первая нормальная форма

Отношение находится в первой нормальной форме, когда значения его атрибутов являются простыми, т. е. Не являются множеством или повторяющейся группой.

Все отношения в нашей БД изначально находятся в первой нормальной форме.

3.2. Вторая нормальная форма

Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме, если все её атрибуты, не входящие в состав первичного ключа, функционально от него зависят.

Все отношения нашей БД уже удовлетворяют данному требованию, следовательно она уже находится во второй нормальной форме.

3.3. Третья нормальная форма

Отношения находятся в третьей нормальной форме, если они находятся во второй нормальной форме и при этом любой атрибут, не входящий в состав первичного ключа, функционально зависит только от него.

Все отношения нашей БД уже удовлетворяют данному требованию, следовательно она уже находится в третьей нормальной форме.

4. ОПИСАНИЕ БД В СУБД

4.1. Таблицы

Id	Name

Таблица 1. Порода

Id	Name

Таблица 2. Клуб

Id	fancy_name	age	fathers_breed_id	mothers_breed_id	breed_id

Таблица 3. Собака

id	participant_id	ring_id

Таблица 4. Эксперт

id	name

Таблица 5. Группа

id	first_name	second_name	last_name	dog_id	club_id

Таблица 6. Участник

id	place	dog_id

Таблица 7. Приз

id	breed_id

Таблица 8. Ринг

id	username	password	participant_id	group_id

Таблица 9. Пользователь

4.2. Запросы с листингами SQL

Получение пароля пользователя по имени пользователя:

```
SELECT password  
FROM %(users_table)s  
WHERE username = %(username)s
```

Получение группы пользователя по имени пользователя:

```
SELECT %(groups_table)s.name  
  
FROM %(groups_table)s INNER JOIN %(users_table)s on %  
(groups_table)s.id = % (users_table)s.group_id  
  
WHERE %(users_table)s.username = %(username)s;
```

Получение экспертов по заданной породе:

```
SELECT %(participant_table)s.id  
  
FROM %(ring_table)s, %(experts_table)s, %(participant_table)s  
  
WHERE %(id)s = %(ring_table)s.breed_id  
  
AND %(experts_table)s.ring_id = %(ring_table)s.id  
  
AND %(participant_table)s.id = %(experts_table)s.participant_id
```

Получение списка пород по заданному клубу:

```
SELECT %(breed_table)s.name  
  
FROM %(participant_table)s, %(dog_table)s, %(breed_table)s  
WHERE %(participant_table)s.dog_id = %(dog_table)s.id  
AND %(dog_table)s.breed_id = %(breed_table)s.id  
AND %(participant_table)s.club_id = %(id)s
```

Получение списка призов по заданному клубу:

```
SELECT count(%(prize_table)s) FILTER (WHERE %(prize_table)s.place  
= 1) as first_place,
```

```

count(%(prize_table)s FILTER (WHERE %(prize_table)s.place = 2) as
second_place,
count(%(prize_table)s FILTER (WHERE %(prize_table)s.place = 3) as
third_place
FROM %(prize_table)s, %(participant_table)s, %(dog_table)s, %
(club_table)s
WHERE %(prize_table)s.dog_id = %(dog_table)s.id
AND %(participant_table)s.dog_id = %(dog_table)s.id
AND %(participant_table)s.club_id = %(club_table)s.id
AND %(club_table)s.id = %(id)s
GROUP BY %(club_table)s.name;

```

Получение отчета о выступлении по заданному клубу:

```

SELECT %(participant_table)s.last_name, %
(participant_table)s.first_name,
%(participant_table)s.middle_name, %(prizes_table)s.place, %
(dog_table)s.fancy_name,
%(breed_table)s.name
FROM %(prizes_table)s, %(participant_table)s, %(dog_table)s, %
(breed_table)s
WHERE %(prizes_table)s.dog_id = %(participant_table)s.dog_id
AND %(participant_table)s.club_id = %(id)s
AND %(participant_table)s.dog_id = %(dog_table)s.id
AND %(dog_table)s.breed_id = %(breed_table)s.id;

```

Получение собаки по заданному эксперту:

```

SELECT %(participant_table)s.dog_id
FROM %(experts_table)s, %(participant_table)s
WHERE %(id)s = %(experts_table)s.ring_id
AND %(experts_table)s.participant_id = %(participant_table)s.id

```

Получение ринга по заданному участнику:

```

SELECT %(ring_table)s.id
FROM %(participant_table)s, %(dog_table)s, %(ring_table)s
WHERE %(participant_table)s.dog_id = %(dog_table)s.id
AND %(dog_table)s.breed_id = %(ring_table)s.breed_id
AND %(participant_table)s.id = %(participant)s

```

Получение отчета о выступлении заданного участника:

```

SELECT %(participant_table)s.id, %(prizes_table)s.place

```

```

FROM %(prizes_table)s, %(participant_table)s
WHERE %(participant_table)s.dog_id = %(prizes_table)s.dog_id
AND %(participant_table)s.id = %(id)s;

```

Получение списка незанятых рингов:

```

SELECT %(ring_table)s.id

FROM %(ring_table)s LEFT JOIN %(experts_table)s ON %
(ring_table)s.id = %(experts_table)s.ring_id
WHERE %(experts_table)s.id IS NULL;

```

4.3. Реализация интерфейса к БД

Интерфейс к бд, был реализован в виде одностраничного веб-приложения с серверной частью, отвечающей за работу с бд. Ниже представлены примеры пользовательского интерфейса и детали реализации.

DogExpo	Home	Dogs	Participants	Clubs	Rings	Breeds	Experts	Prizes	Logout
Dog list									Add Dog
id	fancy_name	age	breed_id	fathers_breed_id	mothers_breed_id				
1	Шарик	1	Пудель	Пудель	Пудель				
2	Бобик	2	Шпиц	Шпиц	Шпиц				
3	Тузик	3	Хаски	Хаски	Хаски				
4	Мухтар	31	Овчарка	Овчарка	Овчарка				
5	Рекс	12	Овчарка	Овчарка	Овчарка				

Рисунок 3. Список собак

DogExpo	Home	Dogs	Participants	Clubs	Rings	Breeds	Experts	Prizes	Logout
Fancy name: <input type="text"/>									
Age: <input type="text"/>									
Breed: <input type="text"/>									
Mothers breed: <input type="text"/>									
Fathers breed: <input type="text"/>									
<input type="submit" value="Submit"/>									

Рисунок 4. Создание собаки

В ходе работы были использованы следующие технологии для создания интерфейса:

Серверная часть - Python 3.7 + FlaskRESTful

Клиентская часть - Javascript ES6 + Vue.js.

Для локального тестирования и запуска потребуются установленные Python (версии не ниже 3.6), pip, virtualenv, NodeJS. А также, установленный и запущенный redis. Все представленные ниже инструкции верны для дистрибутива Ubuntu 18.

Для сборки серверной части требуется выполнить следующие шаги в терминале:

1. `python3 -m virtualenv venv`
2. `source ./venv/bin/activate/`
3. `pip install -r requirements.txt`
4. `flask run app.py --host=0.0.0.0 --port=5000`

Для сборки клиентской части требуется выполнить следующие шаги в терминале:

1. `cd ./web/`
2. `npm install`
3. `cd ./src/`
4. `vue serve`

После этого, клиент будет доступен из браузера по адресу <http://localhost:8080/>.

5. РАЗГРАНИЧЕНИЕ ПРАВ

5.1. Доступы каждой группы

В данной работе, пользователи разделены на две группы:Администратор и Пользователь.

Для группы Администратор, предусмотрены операции чтения и записи.

Для группы Пользователь, предусмотрена только операция чтения.

5.2. Детали реализации

Ограничения на запись и чтения произведены на уровне HTTP-запросов, что на данный момент является лидирующим методом разграничения прав пользователей. Также, пользователи каждой группы видят только элементы интерфейса, которые для них доступны.

Для группы администраторов доступны как и POST так и GET запросы, в отличии от группы пользователей, для которых доступны только GET запросы.

Само разграничение происходит на этапе обработки запроса на сервере, путём декорирования функции-обработчика. Перед обработкой, происходит проверка группы текущего пользователя, которая хранится в сессии в кэше (key-value хранилище, на основе структуры данных хэш-таблица), доступ к которой быстрее, чем постоянное обновление группы пользователя из базы данных.

Группа пользователя добавляется в кэш, на этапе входа в систему и при выходе из системы удаляется из него.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была создана реляционная база данных. В качестве области применения были выбраны выставки собак. В ходе выполнения курсовой работы была построена концептуальная модель, а затем и первичный вид реляционной базы данных. Методом последовательного приведения к видам нормальной формы база данных была приведена к третьей нормальной форме, также была построена ER-диаграмма раскрывающая связи и объекты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Цехановский В.В. Методические указания к курсовой работе по дисциплине "Управление данными". СПб: СПбГЭТУ "ЛЭТИ", 2004. 29 с.
2. Дейт К. Дж. Введение в системы баз данных. М.: «Вильямс», 2006. 1328 с.