



В таблице 1 представлена матрица ошибок классификации текстурных изображений с использованием нейронной сети. Она показывает вероятность попадания изображения из класса i в класс j . На главной диагонали этой матрицы располагаются вероятности правильной классификации.

Таблица 1 – Матрица ошибок классификации текстурных изображений

	bark	brick	bubbles	grass	leather	pigskin	rafia
bark	1	0	0	0	0	0	0
brick	0	1	0	0	0	0	0
bubbles	0	0	0.5714	0	0.1429	0	0
grass	0.2857	0	0	1	0	0	0
leather	0	0	0.2857	0	0.8571	0	0
pigskin	0	0	0.1429	0	0	1	0.1429
rafia	0	0	0	0	0	0	1

Литература

1. Naralick, R (1979). Statistical and structural approaches to texture. Proceedings of the IEEE 67, 786–804.
2. Шапиро Л., Стокман Дж. Компьютерное зрение / Пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.
3. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. / Пер. с англ. – М.: ООО «И.Д. Вильямс», 2006. – 1104 с.

Я.В. Соловьева, О.В. Борисова

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ МНОГОПОЛЬЗОВАТЕЛЬСКИМИ РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ

(Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет))

С ростом популярности систем управления базами данных (СУБД) в 70-80-х годах появилось множество различных моделей данных. Первые СУБД использовали иерархическую модель данных. Для чтения данных из иерархической базы данных требовалось перемещаться по записям, за один раз переходя на одну запись вверх, вниз или в сторону. Если структура данных оказывалась сложнее, чем обычная иерархия, простота структуры иерархической базы данных становилась её недостатком. В связи с этим была разработана новая сетевая модель данных. Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков. Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между



этимися записями. Как и иерархические базы данных, сетевые базы данных были очень жесткими. Наборы отношений и структуру записей приходилось задавать наперед. Изменение структуры базы данных обычно означало перестройку всей базы данных.

Недостатки иерархической и сетевой моделей привели к появлению новой, реляционной модели данных, созданной Коддом в 1970 году и вызвавшей всеобщий интерес. Реляционная модель была попыткой упростить структуру базы данных. Такие модели управления можно охарактеризовать простотой, удобным табличным представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных.

Реляционные базы данных позволяют хранить информацию в нескольких «плоских» (двухмерных) таблицах, связанных между собой посредством совместно используемых полей данных, называемых ключами. Реляционные базы данных предоставляют более простой доступ к оперативно составляемым отчетам (обычно через SQL) и обеспечивают повышенную надежность и целостность данных благодаря отсутствию избыточной информации [1].

В рамках данного проекта изучалась реляционная модель данных. Разрабатываемая система предназначена для организации доступа к удаленной базе данных различных пользователей в локальной сети за счет расширения функционала SQLite. Система позволит освободить человека от необходимости рутинной работы, а также сократить время на установку программного обеспечения (ПО).

При разработке автоматизированной системы необходимо предусмотреть:

- контроль корректности задания всех параметров;
- создать информационную базу для автоматизированного процесса управления реляционными базами данных;
- разграничение прав и обязанностей пользователей;
- выполнение ограничений целостности базы данных;
- возможность корректировки справочника пользователей;
- возможность выгрузки оперативной информации.

SQLite – это реляционная база данных запросы к которой можно осуществлять при помощи языка запросов SQL. База данных не поддерживает все особенности SQL и уступает в функциональности другим развитым СУБД, но вполне подходит для хранения и извлечения информации

Ядро базы данных является сердцевиной СУБД; оно отвечает за физическое структурирование и запись данных на диск, а также за физическое чтение данных с диска. Кроме того, оно принимает SQL-запросы от других компонентов СУБД (таких как генератор форм, генератор отчетов или модуль формирования интерактивных запросов), от пользовательских приложений и даже от других вычислительных систем [2]. SQL - язык создания приложений клиент/сервер, и программах для персональных компьютеров SQL используется



для организации связи через локальную сеть с сервером базы данных, в которой хранятся совместно используемые данные. SQL выполняет много различных функций.

- организация данных. SQL дает пользователю возможность изменять структуру представления данных, а также устанавливать отношения между элементами базы данных;
- чтение данных. SQL дает пользователю или приложению возможность читать из базы данных содержащиеся в ней данные и пользоваться ими;
- обработка данных. SQL дает пользователю или приложению возможность изменять базу данных, т.е. добавлять в нее новые данные, а также удалять или обновлять уже имеющиеся в ней данные;
- управление доступом. С помощью SQL можно ограничить возможности пользователя по чтению и изменению данных и защитить их от несанкционированного доступа;
- совместное использование данных. SQL координирует совместное использование данных пользователями, работающими параллельно, чтобы они не мешали друг другу;
- целостность данных. SQL позволяет обеспечить целостность базы данных, защищая ее от разрушения из-за несогласованных изменений или отказа системы [3].

АС имеет архитектуру «клиент-сервер» ПО разделено на две части – клиентскую и серверную. БД размещается на компьютере – сервере сети (сервере или удаленном сервере) и называется также удаленной БД. Приложение, осуществляющее работу с этой БД, находится на компьютере пользователя. Приложение пользователя является клиентом, его также называют приложением-клиентом.

Клиент и сервер взаимодействуют следующим образом. Логика представления данных и бизнес-логика размещаются на клиенте, который, в случае, когда сервером является СУБД, общается с логикой хранения и накопления данных на сервере, используя язык SQL. Таким образом, в архитектуре «клиент-сервер» клиент посылает запрос и получает только те данные, которые ему действительно нужны. Вся обработка запроса выполняется на удаленном сервере[4].

На рисунке 1 показана главная форма работы с сервером СУБД, которая выполняет основные функции серверной части приложения:

- выполнять соединение с сервером и отключение от него;
- выполнять обработку полученных данных.

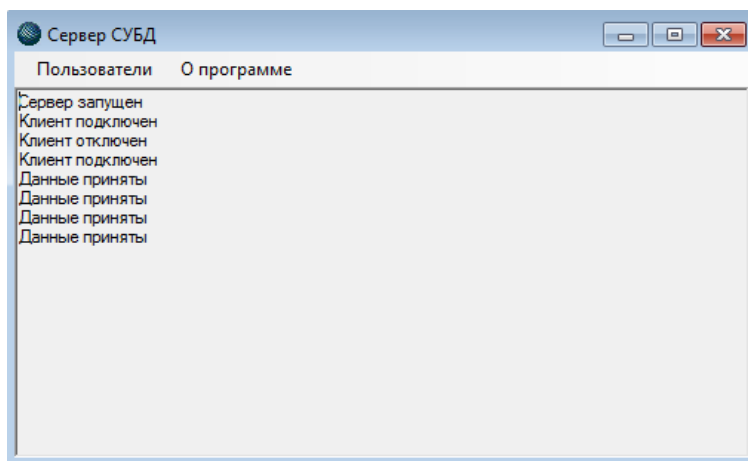


Рис. 1. Главная форма сервера СУБД

На рисунке 2 изображена экранная форма клиентской части. Основные функции, которые она выполняет:

- создание запроса.
- обработка запроса: открытие файла, считывание строки из файла, запись файла в базу, закрытие файла.
- Получение результата выполнения запроса.

Также при разработке АС предусмотрено разграничение прав и обязанностей пользователей: для того чтобы войти в систему нужно добавить нового пользователя и войти под его логином и паролем.

Таким образом, реализованная в рамках данной работы система управления реляционными базами данных.

Разработанная подсистема отвечает всем требованиям задания и реализует следующие основные функции:

- выполнение соединения с сервером и отключение от него;
- создание запросов;
- обработка запросов;
- получение результата выполнения запроса.

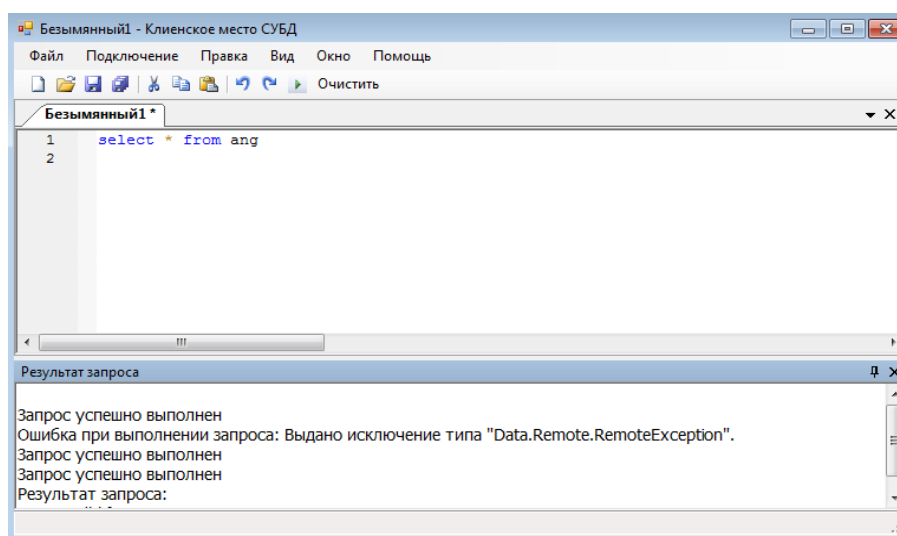


Рис. 2. Экранная форма клиентской части



Литература

- 1 Диго, С.М. Проектирование баз данных [Текст]: учебник/С.М. Диго. – М.: Финансы и статистика, 2002. – 216 с.
- 2 Кригель, А. Борис Трухнов. SQL. Библия пользователя. Язык запросов SQL = SQL Bible [Текст]/А. Кригель, Б. Трухнов. - М.: Диалектика, 2009.- 752 с.
- 3 Чигарина, Е. И. Лекции по курсу «Базы данных и базы знаний» [Текст]/Е. И. Чигарина. – Самара: СГАУ, 2009.
- 4 Когаловский, М.Р. Энциклопедия технологий баз данных[Текст]/М.Р. Когаловский - М.: Финансы и статистика, 2002. – 800 с.

Н.Н. Хрисанов

ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ МНОГОЭЛЕМЕНТНЫХ ДВУХПОЛЮСНЫХ ЦЕПЕЙ ПРИ СИНУСОИДАЛЬНОМ ВОЗДЕЙСТВИИ

(Самарский государственный технический университет)

Рассматривается метод определения параметров многоэлементных двухполюсных цепей, основанный на интегрировании переходного процесса при воздействии, имеющем вид одной или суммы нескольких экспонент[1].

Пусть структурная схема измерительного канала имеет вид, представленный на рис.1, где ИСВ – источник синусоидального воздействия; ДЭЦ – исследуемая двухполюсная электрическая цепь (в данном случае последовательная RC – цепь); ОУ – операционный усилитель; УПО – устройство предва-

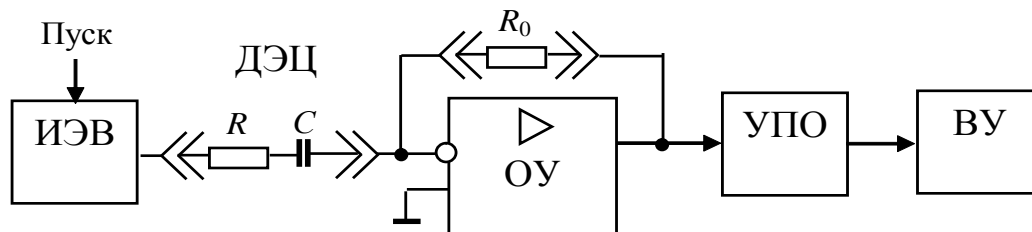


Рис.1. Схема для определения параметров последовательной RC цепи

рительной обработки, включающее в себя аналоговый интегратор; ВУ – вычислительное устройство; R_0 - опорный элемент.

При подаче на вход исследуемой цепи напряжения вида:

$$U_{ВХ}(t) = U_0 \sin \omega t = U_0/2\omega (e^{i\omega t} - e^{-i\omega t}).$$

Напряжение на выходе будет иметь:

$$U_{ВЫХ}(p) = -A \frac{p}{(p^2 + \omega^2)(p - p_1)},$$

где

$$A = U_0 \frac{R_0}{R}, \quad p_1 = -\frac{1}{RC}.$$