**Лабораторная работа №9**

**Разработка простейшего SQL-сервера с моделированием сетевого протокола доступа**

**Задание**

1. Изучить принципы организации, схемы и структуры базы данных, структуры таблиц и типов данных
2. Изучить основные типы запросов в базах данных
3. Изучить объектные компоненты и их свойства для доступа и работы с данными: DataBase (база данных), Connection (соединение с БД), Query (запрос), Command (SQL-командf), Table (таблица), DataSource (источник данных), Field (поле), DataSet (набор данных), RecordSet (набор записей), Record (запись)
4. Изучить интегрированные среды разработки ПО и программные средства управления базами данных
5. Изучить архитектуру и принципы работы распределенных баз данных
6. В качестве рабочей базы данных взять базу данных из Лабораторной работы № 4
7. Разработать программу-сервер (*Сервер*) базы данных и программу-клиент (*Клиент*) базы данных. Взаимодействие между *Клиентом* и *Сервером* осуществить на одном локальном компьютере путем моделирования сетевого протокола доступа;
8. *Сервер* получает SQL-запросы от клиента в виде сформированных текстовых строк и выполняет их
9. *Клиент* формирует текстовые строки SQL-запросов, направляет их *Серверу*, ждет ответа в виде набора записей или подтверждения результата операции и отображает результат выборки в своем компоненте-сетке (*StringGrid*);
10. Разработка программы *Сервер:*
    1. Создать главную форму программы *Сервер*. На форме расположить следующие элементы управления – главное меню программы (*MainMenu*); ниже – две кнопки (*Button1* и *Button2*); ниже – текстовое поле (*Edit*); ниже – многострочное текстовое поле (*Memo*);
    2. Реализовать следующие функции программы:
       1. Реализовать функции главного меню: «Подключиться» (*Сервер* динамически подключается к базе данных), «Сохранить» (в текстовом файле сохраняется информация о подключенном *Клиенте*, дате и времени сеанса), «Выйти» (закрывается программа, предварительно происходит отключение от базы данных);
       2. При нажатии на кнопку *Button1* программа *Сервер* должна перейти в режим ожидания подключения *Клиента.* При этом в заголовке формы должна появиться надпись: «Активен»;
       3. При нажатии на кнопку *Button2* программа *Сервер* должна перейти в режим разрыва связи с *Клиентом.* При этом в заголовке формы должна появиться надпись: «Выключен»;
       4. При подключении *Клиента* в текстовом поле (*Edit*) отобразить имя компьютера клиента;
       5. При получении строки SQL-запроса от *Клиента* отобразить текст запроса в поле (*Memo*);
11. Разработка программы *Клиент*:
    1. Создать главную форму программы *Клиент*. На форме расположить следующие элементы управления: главное меню программы (*MainMenu*); ниже – многострочное текстовое поле (*Memo*); ниже - компонент сетки, не связанной с данными (*StringGrid*); ниже – две кнопки (*Button1* и *Button2*) для перехода к первой или к последней строке компонента сетки (*StringGrid*); справа от компонента (*StringGrid*) - кнопку (*Button3 – «Отправить»*);
    2. Реализовать следующие функции программы:
       1. Реализовать функции главного меню: «Подключиться» (*Клиент* подключается к программе *Сервер*), «Отключиться» (*Клиент* отключается от программы *Сервер*); «Сохранить» (в текстовом файле сохраняется информация о последнем выполненном запросе, дате и времени сеанса), «Выйти» (закрывается программа. Предварительно проверяется подключение к серверу. Если подключение не разорвано, то предлагается сначала разорвать соединение);
       2. В поле (*Memo*) пользователь вручную заносит текст SQL-запроса. Применить следующие запросы:
          1. Отобрать конкретный тип (группу) товара по условию;
          2. Отобрать поступления товара от конкретного поставщика по условию;
          3. Отобрать продажи конкретному покупателю по условию;
          4. Внести в базу новую продажу товара;
          5. Переименовать конкретного поставщика;
       3. При нажатии на кнопку *(Button3)* *Клиент* отправляет текст запроса *Серверу*;
       4. Полученный набор записей от *Сервера* должен отобразиться в компоненте-сетке (*StringGrid*);
       5. Производить заполнение сетки данных (*StringGrid*) программно, путем прохождения цикла по сформированному набору данных;
12. Разработка модели сценария сетевого протокола между *Клиентом* и *Сервером*:
    1. Запуск *Сервера*. Подключение *Сервера* к базе данных.
    2. Открытие сетевого порта связи с *Клиентом*, который моделируется использованием сигнальных файлов, приведённых ниже;
    3. Включение *Сервера* в режим ожидания. В этом режиме *Сервер* постоянно сканирует рабочую папку на наличие сигнального файла – *flag.txt*;
    4. Запуск *Клиента*. Подключение *Клиента* к *Серверу.* При этом *Клиент* создает в рабочей папке сигнальный файл – *flag.txt*, за которым следит *Сервер*. В сигнальный файл *Клиент* записывает свое имя;
    5. *Сервер* при обнаружениисигнального файла – *flag.txt* – сигнализирует о подключении *Клиента*. При этом *Сервер* начинает сканирование рабочей папки на наличие файла запроса – *sql.txt*;
    6. На форме *Клиента* записывается текст SQL-запроса и нажимается кнопка (*Button3 – «Отправить»*). При этом Клиент создает файл – *sql.txt* – и записывает в него текст SQL-запроса из поля (*Memo*);
    7. *Клиент* переходит в режим ожидания ответа от *Сервера*, постоянно сканируя рабочую папку на наличие файла *data.txt*;
    8. *Сервер,* при обнаружении файла *sql.txt* в рабочей папке, читает из него текст SQL-запроса и выполняет его. После чего *Сервер* удаляет файл *sql.txt*;
    9. После выполнения запроса *Сервер* создает в рабочей папке файл *data.txt* с результатами работы. Если был запрос на управление данными, то в файле *data.txt* возвращается слово «*успех*». Если был запрос на выборку данных, то в файле *data.txt* возвращается набор записей в текстовом виде.
    10. *Клиент*, обнаружив файл *data.txt*, читает из него набор записей и отображает его в компоненте сетка (*StringGrid*). После чего *Клиент* удаляет файл *data.txt*;
    11. При отключении от *Сервера* *Клиент* удаляет сигнальный файл - *flag.txt.* При исчезновении файла - *flag.txt -* *Сервер* сигнализирует об отключении *Клиента*;
13. Сделать выводы об эффективности использования сервера, протоколе взаимодействия, успешности выполнения запросов, компонентах, связанных с данными, объектном доступе к базам данных.

**Отчет должен содержать:**

1. Основные сведения о клиент-серверных технологиях и распределенных базах данных;
2. Основные сведения о сетевых протоколах и функционировании серверов баз данных;
3. Схему базы данных в виде UML-диаграммы
4. Блок-схемы алгоритмов работы методов классов
5. Скриншот экрана приложения с результатом работы программы
6. Текст исходных кодов программы
7. Краткие выводы
8. Перечень использованных источников