**Лабораторная работа №10**

**Моделирование многопользовательского приоритетного доступа к базе данных**

**Задание**

1. Изучить принципы организации, схемы и структуры базы данных, 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, принципы работы, сведения о политиках безопасности в распределенных базах данных;
6. В качестве рабочей базы данных взять базу данных из Лабораторной работы № 4
7. В качестве исходной модели приложений взять разработки Лабораторной работы № 9;
8. Разработать в базе данных следующие дополнительные служебные таблицы:
   1. Таблица для хранения данных *Клиента* – *Имя, Пароль, Приоритет*
   2. Таблица для хранения очереди запросов от *Клиентов* с указанием их приоритетов
9. **Разработать 3 экземпляра программы *Клиента* и указать для них разные приоритеты;**
10. Разработать для исходной модели приложений (Лабораторной работы № 9) средства идентификации *Клиента* и средства разграничения прав доступа на основе приоритетов *Клиентов*;
11. При подключении к *Серверу* должна происходить идентификация *Клиента* с запросом его имени и пароля;
12. Каждому *Клиенту* присваивается уровень приоритета (1, 2, 3 и т.д.). Обработка запросов от *Клиентов* (после их идентификации) должна производиться в очередности, зависимой от приоритета;
13. Реализовать следующие дополнительные (к Лабораторной работе № 9) функции программы *Сервер*:
    1. Хранить в отдельной таблице имена, пароли и приоритеты *Клиентов*. Имя, пароль и приоритет *Клиента* записываются в базу данных при его первом подключении (регистрации);
    2. Создавать и хранить очередь запросов от *Клиентов* и выполнять запросы в порядке их приоритетов;
    3. Поместить на форму *Сервера* компонент сетку (*StringGrid*) для хранения имен и приоритетов подключенных *Клиентов*;
14. Реализовать следующие дополнительные (к Лабораторной работе № 9) функции программы *Клиент*:
    1. Поместить на форму три поля для внесения имени, пароля и приоритета;
    2. Имя, пароль и приоритет должны передаваться серверу при подключении к нему;
15. Разработать дополнительную (к Лабораторной работе № 9) модель сценария взаимодействия между *Клиентом* и *Сервером*:
    1. Три *Клиента* подключаются к *Серверу.* При этом открываются три «порта» связи с *Клиентами* – то есть, – три комплекта сигнальных файлов: *flag.txt, sql.txt, data.txt.* Предусмотретьдополнительный файл для передачи данных идентификации от каждого *Клиента*;
    2. При подключении к *Серверу* *Клиент* сначала отправляет свои идентификационные данные – имя и пароль. Для упрощения предполагается, что все клиенты уже зарегистрированы в базе данных;
    3. Если *Клиент* на зарегистрирован (его данные не найдены в служебной таблице), то *Сервер* выдает сообщение об этом и требует отключения *Клиента*;
    4. Если *Клиент* зарегистрирован, то *Сервер* взаимодействует с *Клиентом* в штатном режиме;
    5. *Сервер* постоянно сканирует таблицу с запросами от *Клиентов* и выполняет запросы в соответствии с реализованным протоколом из Лабораторной работы № 9;
    6. При этом могут приходить новые запросы от *Клиентов,* которые должны ставиться в очередь;
    7. После выполнения очередного запроса *Сервер* удаляет его текст из служебной таблицы и выполняет следующий запрос с наибольшим приоритетом;
16. 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. Основные сведения о политиках безопасности в базах данных
4. Схему базы данных в виде UML-диаграммы
5. Блок-схемы алгоритмов работы методов классов
6. Скриншот экрана приложения с результатом работы программы
7. Текст исходных кодов программы
8. Краткие выводы
9. Перечень использованных источников