Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа №4-6

**Распределенные информационные системы**

Выполнили:

Студенты 3 курса 1 группы ФИТ

Пригодич Вера,

Позняк Полина

**Лабораторная работа №4**

Используя современные технологии построения РИС, например, Web-сервисы SOA (либо XML/SOAP, либо RESTfull), создать ПО опроса источников данных телеизмерений каналов передачи ресурсов. Реализовать асинхронный опрос двух веб сервисов источников ТИ, центральным веб сервисом. Реализовать обработку исключительных ситуаций, связанных с недоступностью или неисправностью удаленных сервисов. Обеспечить синхронизацию времени между этими веб-сервисами. Обеспечить протоколирование работы сервисов. Обеспечить получение сведений о текущем состоянии опрашиваемых веб-сервисов.

Для выполнения лабораторной работы №4 было разработано 3 программы:

* CentralizedServer;
* FirstClient
* SecondClient

В рамках этой лабораторной работы, было реализовано опрашивание центральным сервером клиентов, протоколирование, получение сведений о текущей работе сервисов.

При запуске сервера, программа ожидает подключения клиентов, для обмена данными.

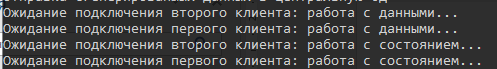


Рисунок 1 – Ожидание сервером подключения клиентов

При запуске клиентов происходит отправка запросов на состояние и данные.

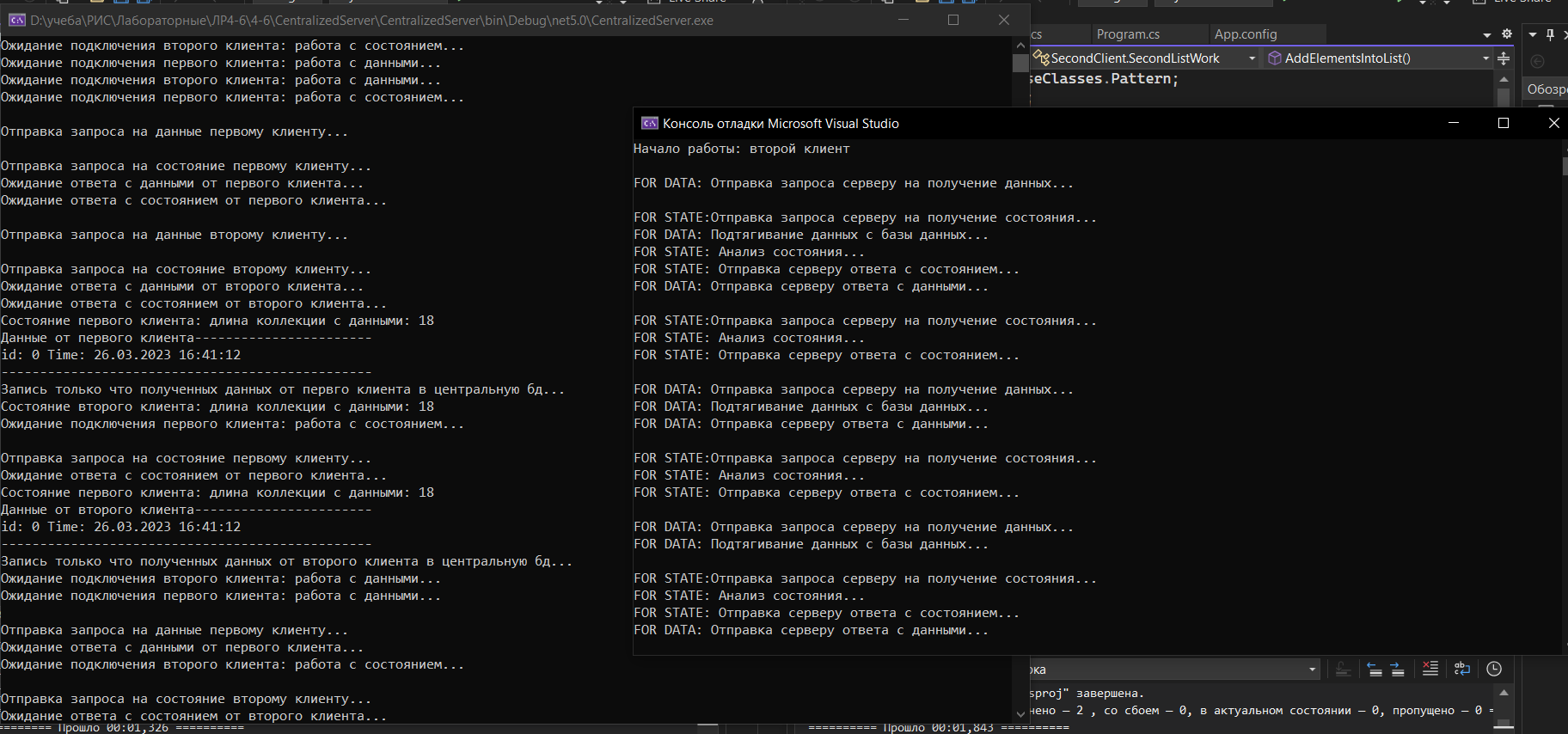


Рисунок 2 – Результат выполнения программы на первом компьютере

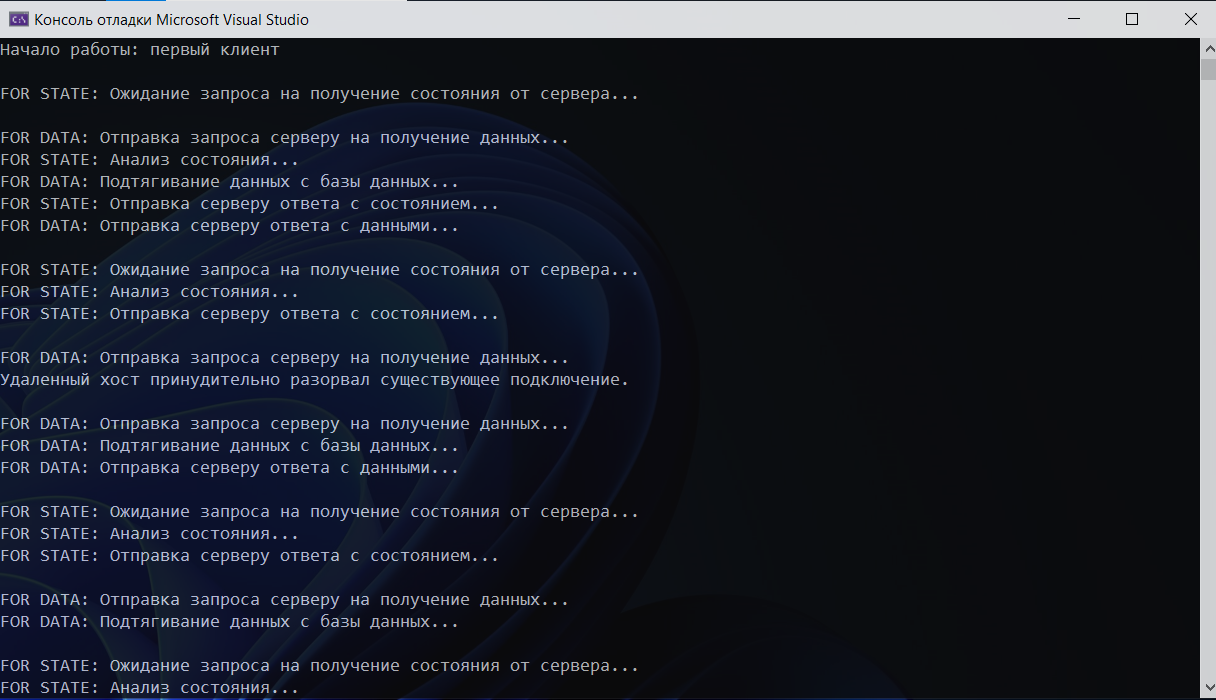


Рисунок 3 – Результат выполнения программы на втором компьютере

Так же была создана обработка исключительных ситуаций, которая обрабатывает все внештатные ситуации, произошедшие во время работы программ.

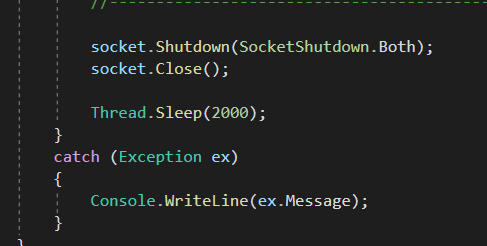


Рисунок 4 – Реализация обработки исключительных ситуаций

На следующем рисунке показана обработка ошибки, при неудавшемся подключении клиента к серверу.

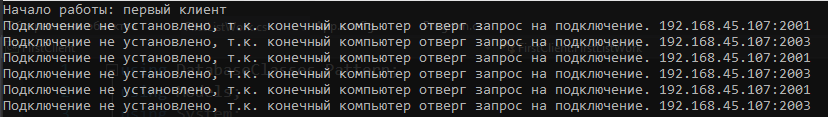


Рисунок 5 – Обработка ошибки неудавшегося подключения клиента к серверу

**Лабораторная работа №5**

Реализовать вытягивающую репликацию данных из территориальных источников (двух) в центральную базу данных показаний счетчиков головной организации. Для чего предварительно разработать программу обеспечивающую заполнение б/д источников модельными данными (для 10 объектов) генерируемыми случайным образом. Диапазон генерируемых значений определяется по номеру студента в учебном журнале гпы. Обеспечить протоколирование работы программы генерации данных и механизма репликации.

Для выполнения лабораторной работы №5 необходимо было разрешить соединение по протоколу TCP/IP с других компьютеров к Microsoft SQL Server. Была создана учетная запись (рис. 1), необходимая для авторизации клиента SQL Sever.

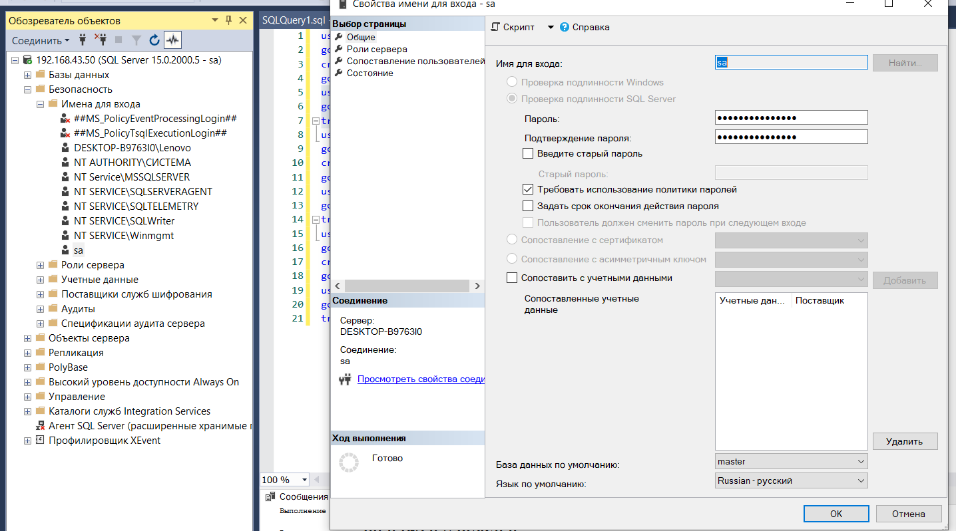


Рисунок 6 – Учетная запись

Оба компьютера могут устанавливать соединение с данной учетной записью.

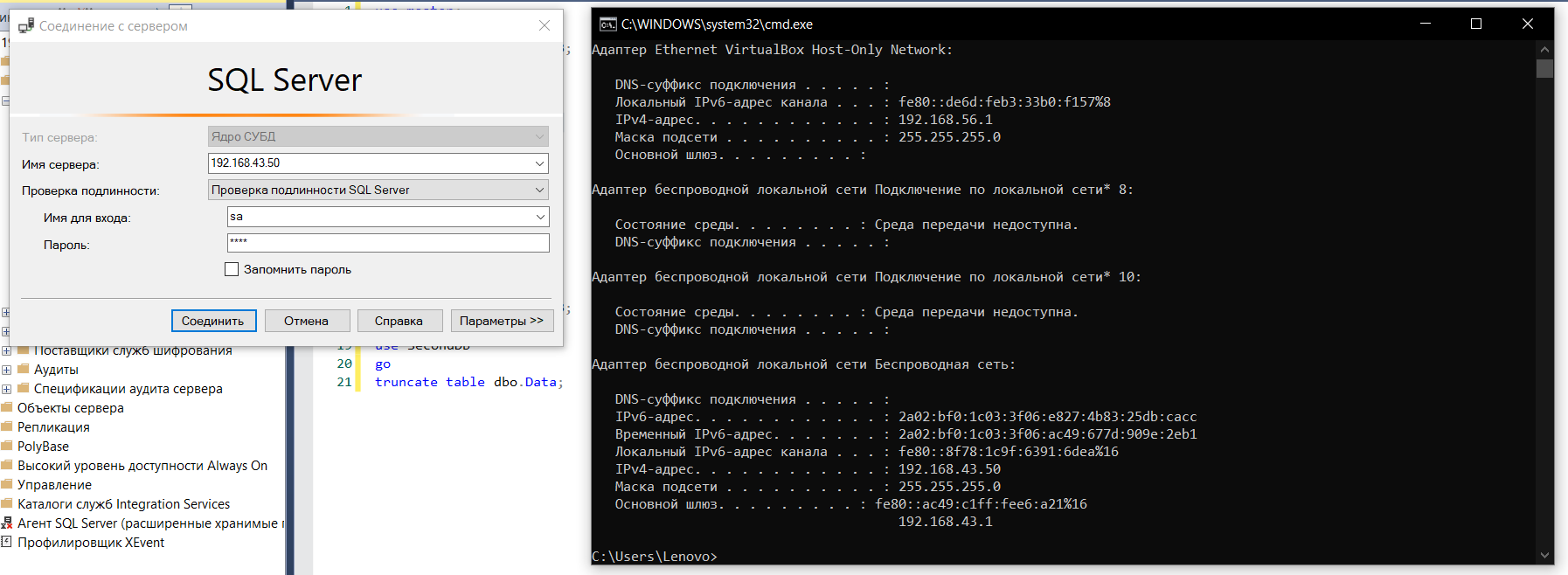


Рисунок 7 – Подключение первого компьютера

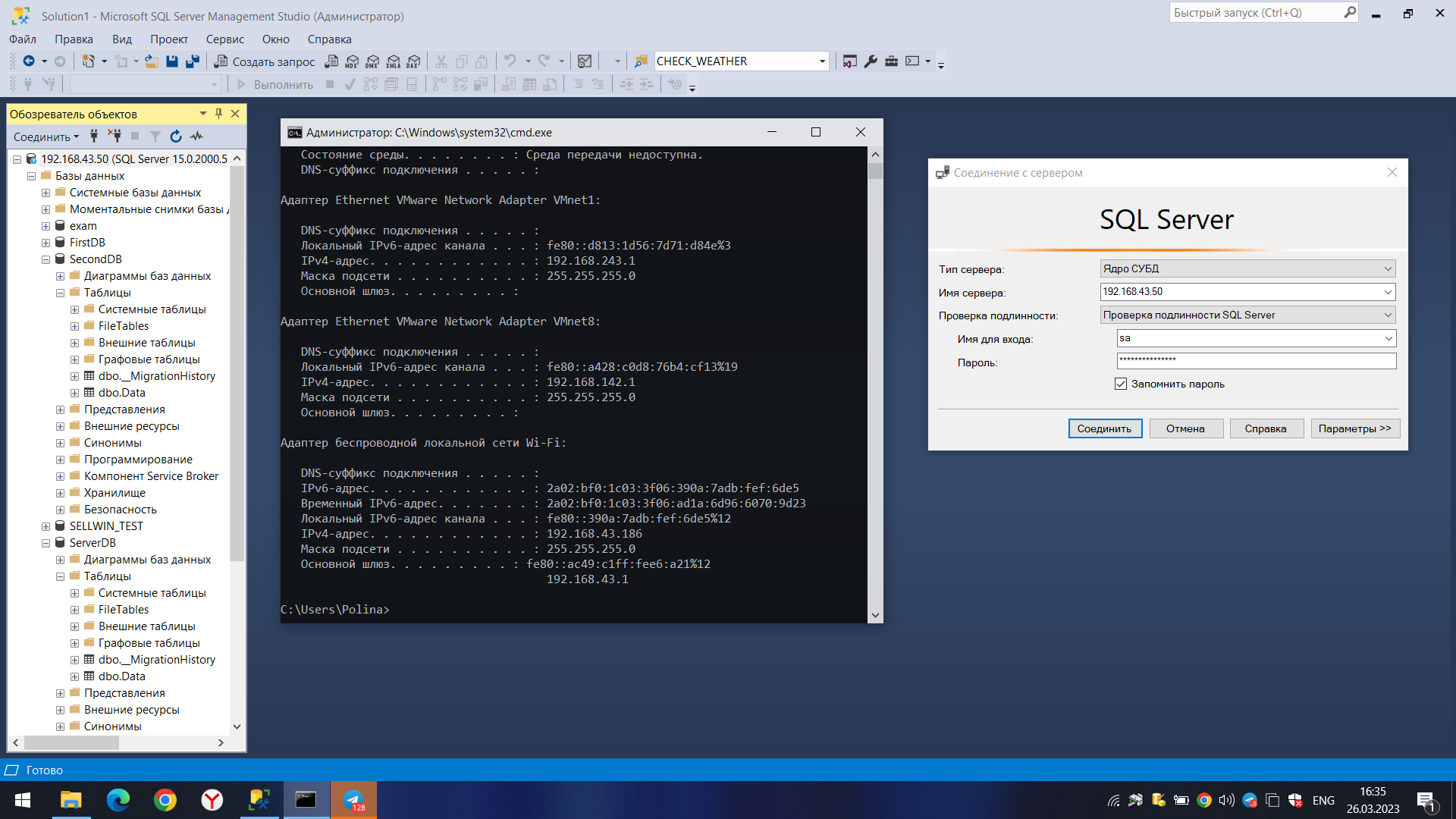


Рисунок 8 – Подключение второго компьютера

Были использованы 3 базы данных: база данных сервера, база данных первого клиента и база данных второго клиента (рис. 4).

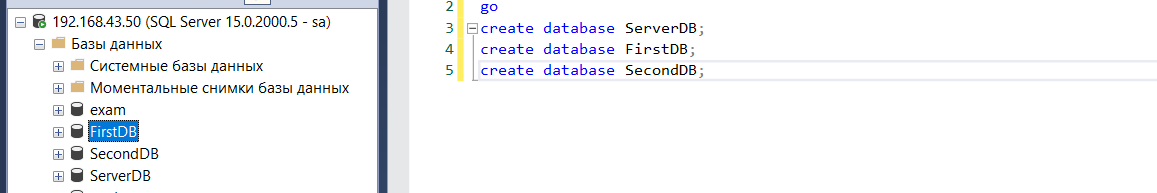


Рисунок 9 – Базы данных

Для того, чтобы реализовать случайное заполнение данными, была разработана ещё одна программа, которая генерирует случайные данные и записывает их в базы данных.

Для этого был реализован паттерн UnitOfWork, с помощью которого в отдельном потоке создаются записи в базе данных.

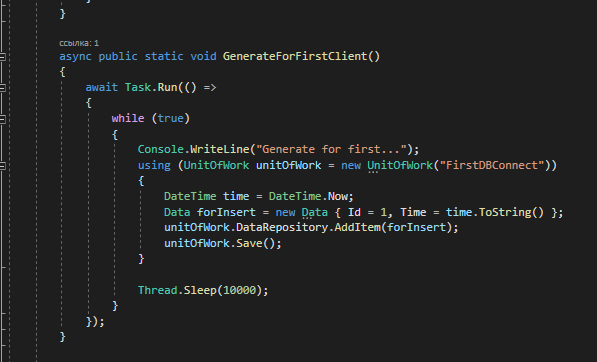


Рисунок 10 – Листинг кода добавления записи в БД

Программа генерирует данные для клиентов после чего записывает их в БД (рис. 11).

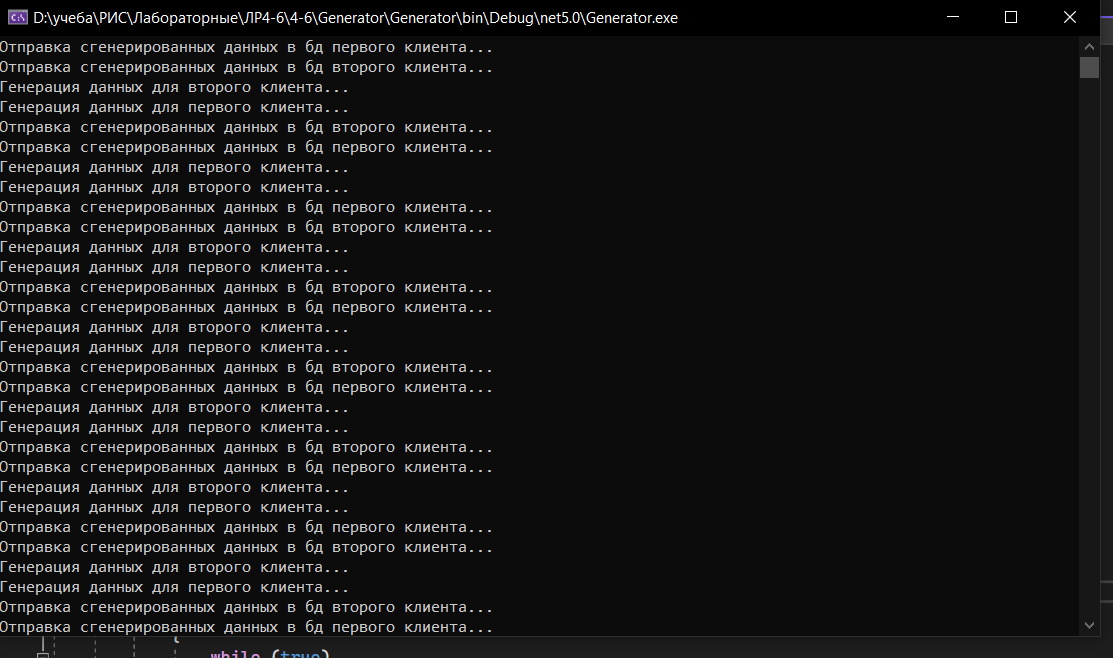


Рисунок 11 – Выполнение генерации данных

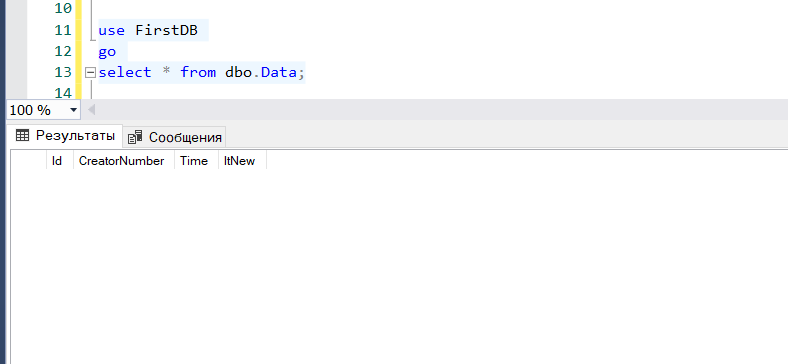


Рисунок 12 – БД первого клиента до выполнения программы

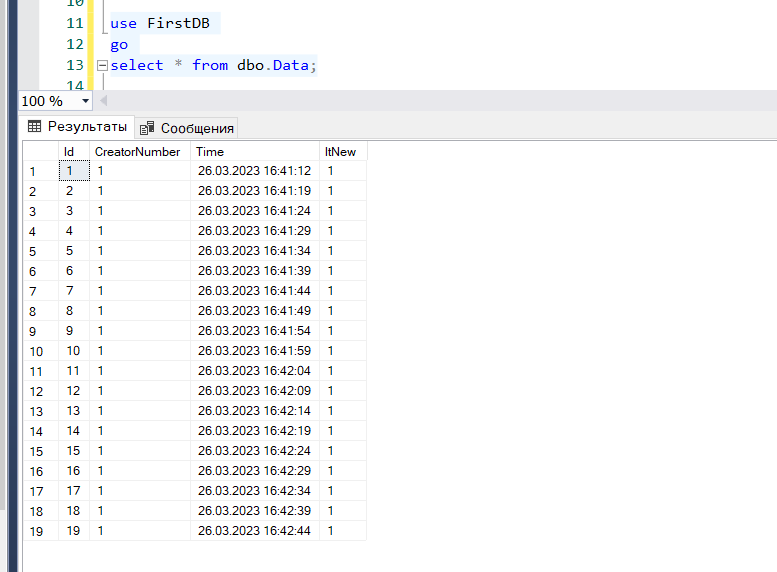


Рисунок 13 – БД первого клиента после выполнения программы

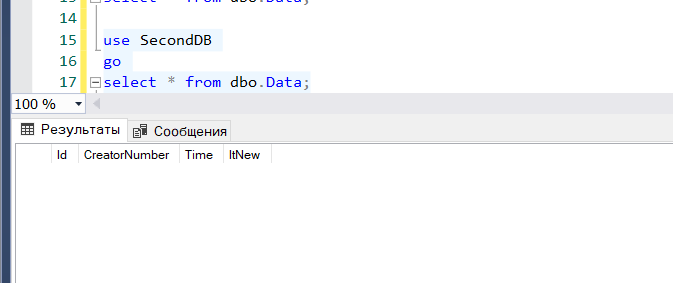


Рисунок 14 – БД второго клиента до выполнения программы

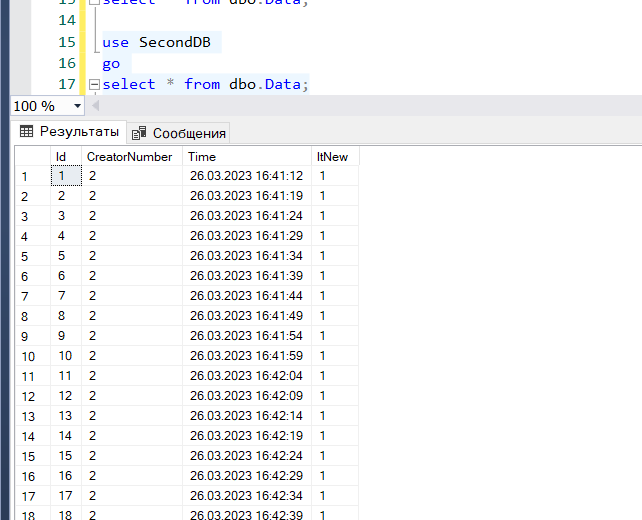


Рисунок 15 – БД второго клиента после выполнения программы

Чтобы реализовать вытягивающую репликацию, в программе сервера было добавлено 2 потока, которые обмениваются данными с клиентами.

Чтобы получить данные для репликации, сервер отправляет запрос клиентам, которые, в свою очередь, отправляют ответ с данными в формате json.

Сервер переводит из формата json, в необходимый тип данных, и проверяет состояние, которое указывает, появились ли новые записи в базе данных клиента.

Если же отправленное состояние базы данных клиентом указывает, что появились новые данные, то сервер создаёт новый объект и из него отправляет данные в свою базу данных.

На рисунке 16 показана детальная информация, о выполнении сервером операций соединения с клиентами и записи данных в базу данных.

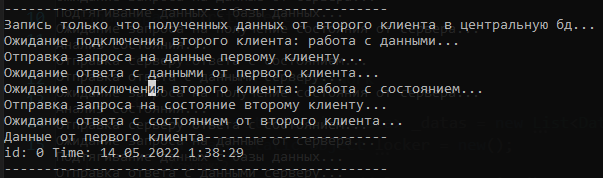


Рисунок 16 – Вывод информации сервером

Если вдруг возникнет исключительная ситуация во время подключения, то сервер будет выводить сообщение об ошибке такое, как показано на рисунке 17.

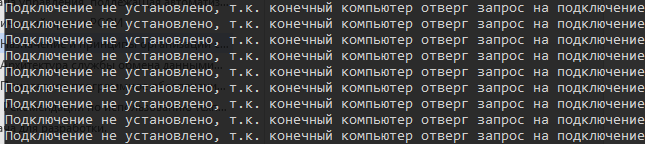


Рисунок 17 – вывод информации, при возникновении ошибок

**Лабораторная работа №6**

Реализовать выталкивающую репликацию данных телеизмерений из централизованной б/д в территориальные (2 б/д по 5 ТИ в каждую). Обеспечить протоколирование работы механизма репликации.

Для этого в программе сервера был добавлен код, который связывается с базами данных клиентов и отправляет им сгенерированные данные.

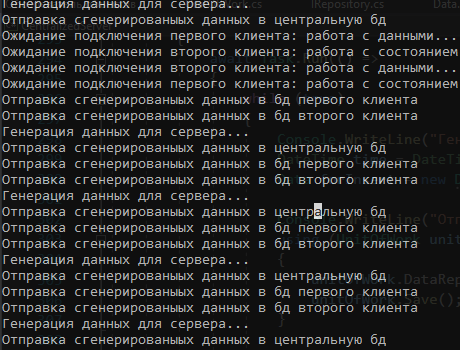


Рисунок 18 – Вывод данных о выталкивающей репликации данных

На рисунке 19 показаны записи, которые были добавлены в базы данных клиентов и сервера. Как видно по результату, то сервер действительно выполняет вытягивающую и выталкивающую репликацию данных.

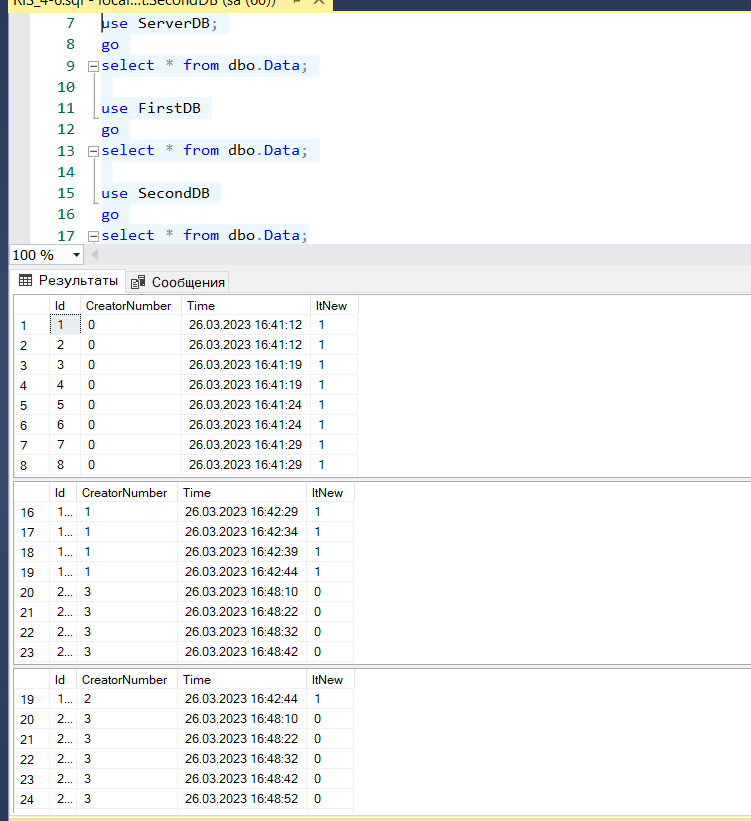


Рисунок 19 – добавление данных сервером в свою базу данных

**Вывод**: в результате лабораторной работы были реализованы два клиента, сервера. Была реализована запись в базу данных клиентами и сервером. Так же была реализована выталкивающая и вытягивающая репликации данных базы данных.