Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа №3

«**Распределенная транзакция в Oracle 12C**»

Выполнил:

Студенты 3 курса 4 группы ФИТ

Артём Елизавета  
Решетилова Антонина

Для данной лабораторной работы были использованы 2 виртуальные машины, подключенные к одной сети. Обе виртуальные обнаруживают друг друга.

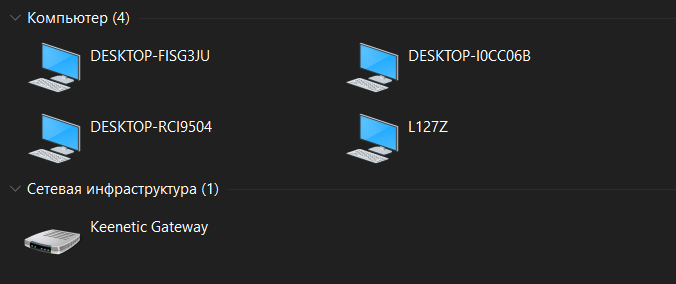


Рисунок 1 ­– Сетевое окружение первой машины

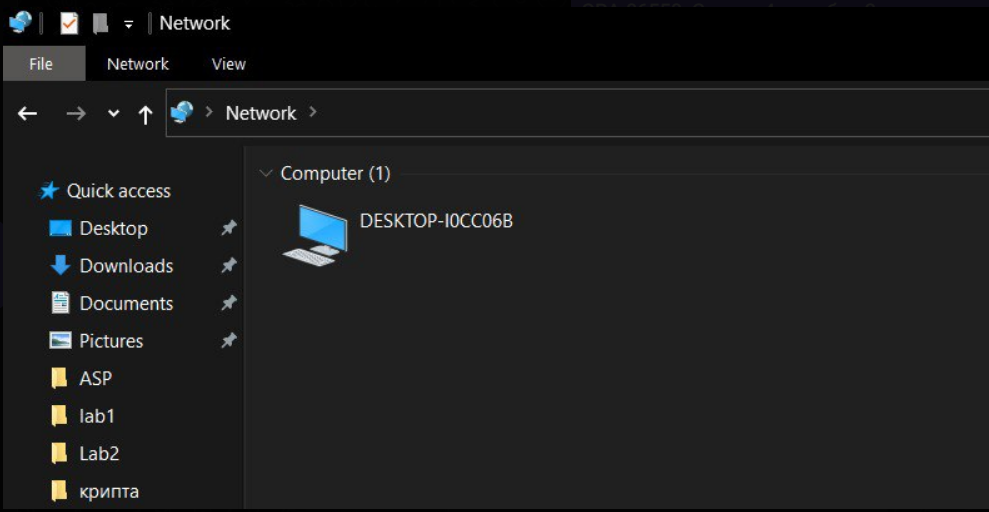


Рисунок 2 ­– Сетевое окружение второй машины

Было проверено, что соединение между машинами установлено, используя команду ping.

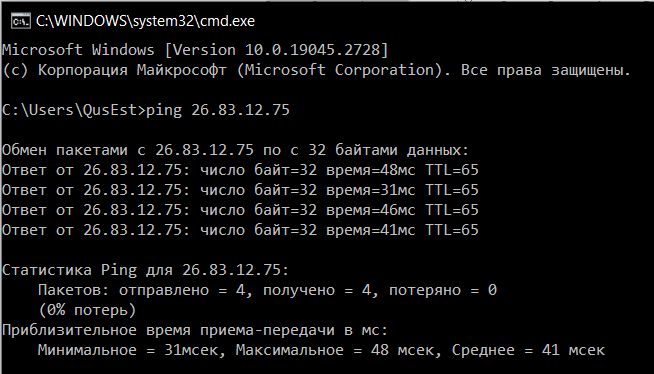


Рисунок 3 ­– ping второй машины

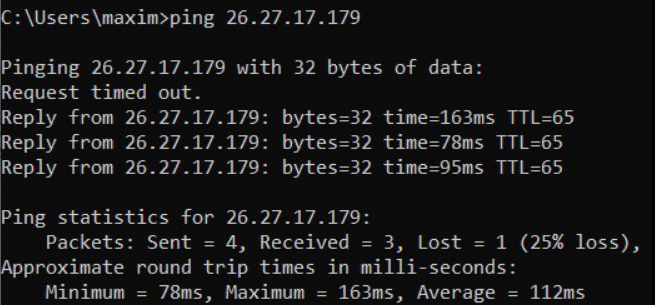


Рисунок 4 ­– ping первой машины

Для выполнения лабораторной работы были использованы два сервера Oracle 12c.

На каждой машине были создан пользователь с именем, обозначающим трехсимвольный идентификатор студента.

create user c##aev identified by qwerty;

Листинг 1 ­– создание первого пользователя

create user c##RAM identified by qwerty;

Листинг 2 ­– создание второго пользователя

Пользователям были выданы привилегии, разрешающие соединение с сервером, полный цикл работы с таблицами, создание DBLINK.

GRANT ALL PRIVILEGES TO AEV;

GRANT ALL PRIVILEGES TO RAM;

Листинг 3 ­– выдача привилегий пользователям

Далее необходимо было создать новое подключение от данных пользователей и осуществлять все последующие действия в их схемах.

Для подключения к удаленной базе данных используется объект DBLINK типа USER1-USER1 между пользователями, расположенными на двух серверах.

CREATE PUBLIC DATABASE LINK AEV\_DB

CONNECT TO AEV

IDENTIFIED BY qwerty

USING '26.27.17.179:1521/xe';

CREATE PUBLIC DATABASE LINK RAM\_DB

CONNECT TO RAM

IDENTIFIED BY qwerty

USING '26.83.12.75:1521/xe';

Листинг 4 ­– создание database link

Необходимо было от имени пользователя-владельца DBLINK разработать SQL-скрипт, выполняющий распределенные транзакции, изменяющие две таблицы на разных серверах.

Были рассмотрены следующие случаи распределенных транзакций:

1. INSERT/INSERT

BEGIN

INSERT INTO AEV\_T VALUES(1);

INSERT INTO RAM\_T@RAM\_DB VALUES(1);

COMMIT;

END;

SELECT \* FROM AEV\_T;

SELECT \* FROM RAM\_T@RAM\_DB;

Листинг 5 ­– распределенные транзакции типа INSERT/INSERT

1. INSERT/UPDATE

BEGIN

INSERT INTO AEV\_T VALUES(2);

UPDATE RAM\_T@RAM\_DB set x = 2 where x = 1;

COMMIT;

END;

SELECT \* FROM AEV\_T;

SELECT \* FROM RAM\_T@RAM\_DB;

Листинг 6 ­– распределенные транзакции типа INSERT/UPDATE

1. UPDATE/INSERT

BEGIN

UPDATE AEV\_T SET X = 3 WHERE X = 1;

INSERT INTO RAM\_T@RAM\_DB VALUES(3);

COMMIT;

END;

SELECT \* FROM AEV\_T;

SELECT \* FROM RAM\_T@RAM\_DB;

Листинг 7 ­– распределенные транзакции типа UPDATE/INSERT

Результат выполнения данных распределенных транзакций выглядит следующим образом.

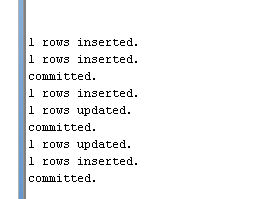


Рисунок 5 – вывод консоли

Необходимо было смоделировать распределенную транзакцию, у которой нарушается ограничение целостности на стороне уделенного сервера. В данном случае нарушается ограничение целостность primary key, так как данное значение уже есть в таблице после операций вставки в предыдущих распределенных транзакциях.

BEGIN

INSERT INTO AEV\_T VALUES(4);

INSERT INTO RAM\_T@RAM\_DB VALUES(3);

commit;

END;

Листинг 8 ­– нарушение ограничение целостности

Т.к. второй пользователь добавляет данные, не соответствующие ограничениям целостности, то при фиксации транзакции первым пользователем, второму пользователю должно выдать ошибку.

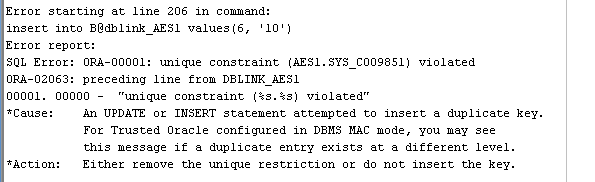


Рисунок 6 – Ошибка у второго пользователя

Необходимо было смоделировать распределенную транзакцию, которая заблокируется и будет ожидать, освобождение ресурса на удаленном сервере.

В данном случае так как ресурс занят, пока не произойдёт фиксация или откат транзакции, другой пользователь не сможет изменить данные в таблице.

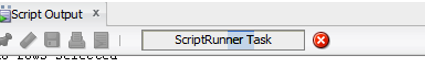


Рисунок 7 – ожидание второго пользователя завершения транзакции

**Вывод:** были созданы соединения удалённые соединения для серверов oracle рассмотрены следующие случаи распределенных транзакций: INSERT/INSERT, INSERT/UPDATE, UPDATE/INSERT и продемонстрировано успешное выполнение транзакций, смоделирована распределенная транзакция, у которой нарушается ограничение целостности на стороне уделенного сервера и распределенная транзакция, которая заблокируется и будет ожидать, освобождение ресурса на удаленном сервере