**Lab 2 - Работа с распределенными (двухфазными) транзакциями**

Для работы с распределенными (двухфазными) транзакциями предлагается использовать СУБД PostgreSQL (<http://www.postgresql.org/download/>).

Для удобства работы с СУБД PostgreSQL можно использовать pgAdmin (<http://www.pgadmin.org/>) или аналогичное графическое приложение.

Для выполнения распределенной транзакции Вам необходимо реализовать свой Менеджер (координатор) распределенных транзакций (TM). TM будет отправлять СУБД команды SQL с помощью которых и будет осуществляться управление распределенной транзакцией, на основе протокола двухфазной фиксации ([Two-Phase Commit Protocol](https://en.wikipedia.org/wiki/Two-phase_commit_protocol)).

TM представляет из себя простое запускаемое приложение (которое может быть реализован на любом из языков программирования), в котором необходимо выполнить набор SQL-запросов к СУБД PostgreSQL.

Процедуру подключения к СУБД PostgreSQL и то как выполнять SQL-запросы из соответствующего языка, читайте в документации или литературе к выбранному языку программирования.

Для имитации распределенной системы, в PostgreSQL необходимо создать две отдельных БД (схемы). Это можно сделать в рамках одного сервера PostgreSQL.

Одна БД будет содержать данные (таблицу) по бронированию билетов на самолет, вторая - бронирование отеля.

***DB1: Fly Booking***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Booking ID** | **Client Name** | **Fly Number** | **From** | **To** | **Date** |
| XXX | Nik | KLM 1382 | KBP | AMS | 01/05/2015 |

***DB2: Hotel Booking***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Booking ID** | **Client Name** | **Hotel Name** | **Arrival** | **Departure** |
| YYY | Nik | Hilton | 01/05/2015 | 07/05/2015 |

В TM Вам необходимо реализовать функциональность, которая заносит информацию о брони на самолет и отель для определенного клиента, используя протокол двухфазной фиксации.

Ниже приводится пример SQL-запросов для выполнения двухфазной транзакции.

1. Here's the sequence that two-phase commits:

BEGIN;

update mytable set a\_col='something' where red\_id=1000;

PREPARE TRANSACTION 'foobar';

COMMIT PREPARED 'foobar';

2. Here's the sequence that rolls back, leaving the table unchanged:

BEGIN;

update mytable set a\_col='something' where red\_id=1000;

PREPARE TRANSACTION 'foobar';

ROLLBACK PREPARED 'foobar';

В вашем приложении постарайтесь промоделировать оба случая.

**Дополнительная информация:**

1. Краткая инструкция по установке и настройке PostgreSQL <https://github.com/jboss-developer/jboss-developer-shared-resources/blob/master/guides/CONFIGURE_POSTGRESQL.md>
2. Информация по командам PostgreSQL для выполнения двухфазной транзакции <http://www.postgresql.org/docs/9.4/static/sql-prepare-transaction.html>
3. В случае возникновения блокировки из-за неудачной транзакции (небыли выполнены команды [COMMIT PREPARED](http://www.postgresql.org/docs/9.4/static/sql-commit-prepared.html) или [ROLLBACK PREPARED](http://www.postgresql.org/docs/9.4/static/sql-rollback-prepared.html)), их можно увидеть выполнив SQL-запрос:

*select \* from pg\_prepared\_xacts;*

из pgAdmin.

Чтобы снять блокировки необходимо из pgAdmin выполнить SQL-запрос

*ROLLBACK PREPARED '<gid>';*

или

*COMMIT PREPARED '<gid>';*

1. В случае возникновения сообщения вида

*XAException occurred. XAException contents and details are: The cause is org.postgresql.util.PSQLException: ERROR: prepared transactions are disabled*

*Error code is: XAER\_RMERR (-3). Exception is: Error preparing transaction*

*Hint: Set max\_prepared\_transactions to a nonzero value.*

модифицируйте файл C:\Program Files\PostgreSQL\9.3\data\postgresql.conf установив возможность для выполнения двухфазных транзакций:

max\_prepared\_transactions = 5