|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №2  
" Транзакции. Уровни изоляции. Уровни изоляции в PostgreSQL Read Committed, Repeatable Read. Точки сохранения."**

по дисциплине «Технологии обработки транзакций клиент-серверных приложений»

**Тема практической работы:**

**Студент группы** ИКБО-16-19 Мурадов Н.Н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись студента)

**Принял преподаватель** Китанин С.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

Содержание

[ПРАКТИЧЕСКАЯ РАБОТА №2 " Транзакции. Уровни изоляции. Уровни изоляции в PostgreSQL Read Committed, Repeatable Read. Точки сохранения." 3](#_Toc97992485)

[Цель работы 3](#_Toc97992486)

[Постановка задачи 3](#_Toc97992487)

[Ход работы 5](#_Toc97992488)

[Вывод 12](#_Toc97992489)

# ПРАКТИЧЕСКАЯ РАБОТА №2 " Транзакции. Уровни изоляции. Уровни изоляции в PostgreSQL Read Committed, Repeatable Read. Точки сохранения."

## Цель работы

Целью данной работы является работа с транзакциями, уровнями изоляциями и точками сохранения в PostgreSQL.

## Постановка задачи

1. Начните транзакцию (командой BEGIN) и создайте новый заказ в таблице sales\_order c сегодняшней датой. Добавьте два предмета в таблицу item, связанных с созданным заказом.

Представьте, что пользователь не подтвердил заказ и все введенные данные необходимо отменить. Выполните отмену транзакции и проверьте, что никакой добавленной вами информации действительно не осталось.

2. Теперь представьте сценарий, в котором нужно отменить не все данные, а только последний из добавленных предметов. Для этого повторите все действия из предыдущего упражнения, но перед добавлением каждого предмета создавайте точку сохранения (с одним и тем же именем). После ввода второго предмета выполните откат к точке сохранения. Проверьте, что заказ и первый предмет остались.

3. В рамках той же транзакции добавьте еще один предмет и зафиксируйте транзакцию. Обратите внимание на то, что после этой операции отменить внесенные транзакцией изменения будет уже невозможно.

Уровень изоляции Read Committed

4. Перед началом выполнения задания проверьте, что в таблице sales\_order нет заказов на сумму total 1 000 рублей.

4.1. В первом сеансе начните транзакцию (командой BEGIN). Выполните обновление таблицы sales\_order: увеличьте total в два раза в тех строках, где сумма равна 1 000 рублей.

4.2. Во втором сеансе (откройте новое окно psql) вставьте в таблицу sales\_order новый заказ на 1 000 рублей и зафиксируйте транзакцию.

4.3. В первом сеансе повторите обновление таблицы sales\_order и зафиксируйте транзакцию.

Осталась ли сумма добавленного заказа равной 1 000 рублей? Почему это не так?

Уровень изоляции Repeatable Read

5. Повторите предыдущее упражнение, но начните транзакцию в первом сеансе с уровнем изоляции транзакций Repeatable Read. Объясните различие полученных результатов.

6. Выполните указанные действия в двух сеансах:

6.1. В первом сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 20 000 рублей.

6.2. Во втором сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 30 000 рублей.

6.3. В первом сеансе добавьте новый заказ на 30 000 рублей и снова вычислите количество заказов с суммой 20 000 рублей.

6.4. Во втором сеансе добавьте новый заказ на 20 000 рублей и снова вычислите количество заказов с суммой 30 000 рублей.

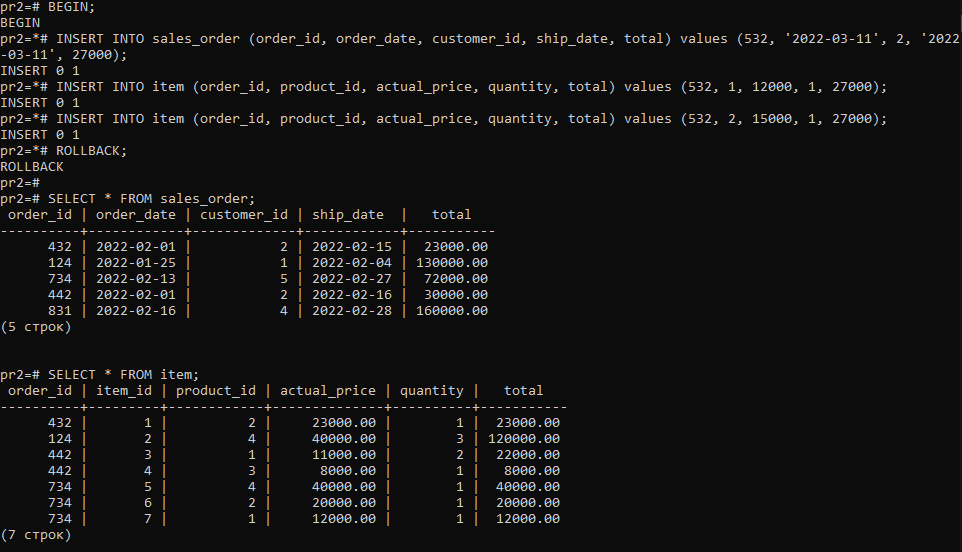
6.5. Зафиксируйте транзакции в обоих сеансах.

Соответствует ли результат ожиданиями? Можно ли сериализовать эти транзакции (иными словами, можно ли представить такой порядок последовательного выполнения этих транзакций, при котором результат совпадет с тем, что получился при параллельном выполнении)?

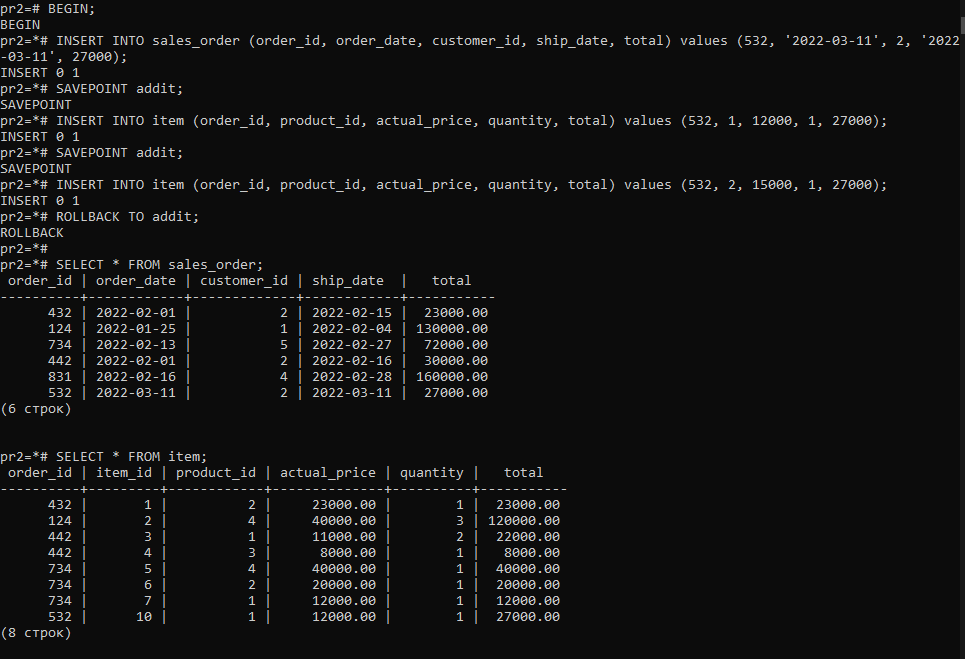
## Ход работы

1. Начните транзакцию (командой BEGIN) и создайте новый заказ в таблице sales\_order c сегодняшней датой. Добавьте два предмета в таблицу item, связанных с созданным заказом.

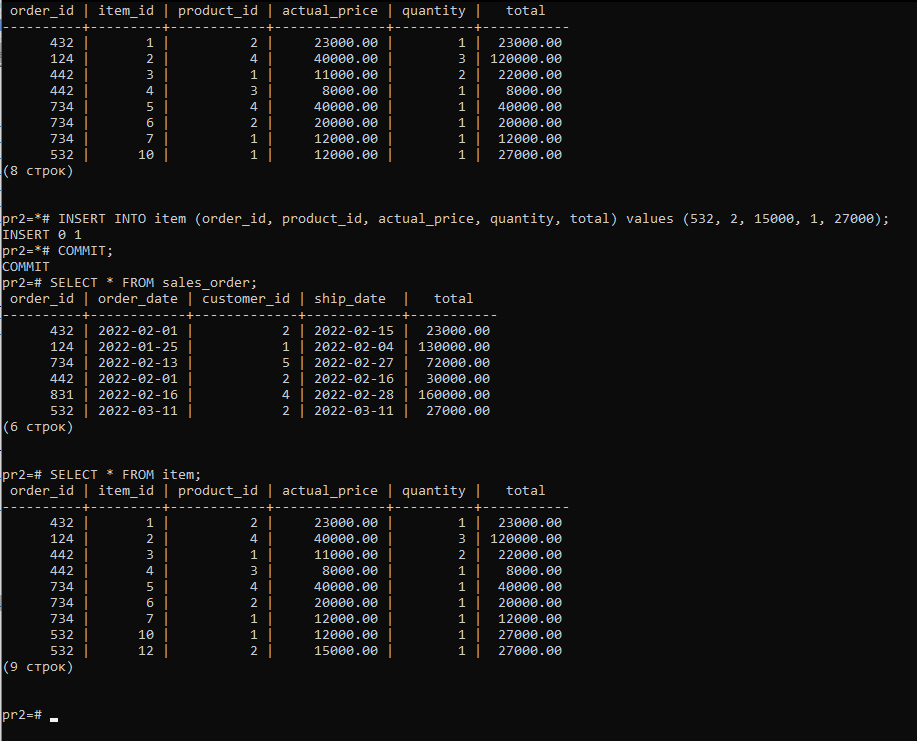
Представьте, что пользователь не подтвердил заказ и все введенные данные необходимо отменить. Выполните отмену транзакции и проверьте, что никакой добавленной вами информации действительно не осталось.

  
Рисунок 1 - Процесс выполнения работы.

2. Теперь представьте сценарий, в котором нужно отменить не все данные, а только последний из добавленных предметов. Для этого повторите все действия из предыдущего упражнения, но перед добавлением каждого предмета создавайте точку сохранения (с одним и тем же именем). После ввода второго предмета выполните откат к точке сохранения. Проверьте, что заказ и первый предмет остались.

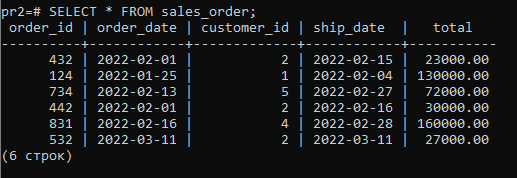
  
Рисунок 2 - Процесс выполнения работы.

3. В рамках той же транзакции добавьте еще один предмет и зафиксируйте транзакцию. Обратите внимание на то, что после этой операции отменить внесенные транзакцией изменения будет уже невозможно.

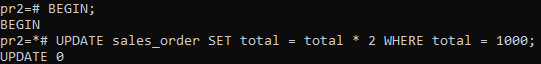
  
Рисунок 3 - Процесс выполнения работы.

**Уровень изоляции Read Committed**

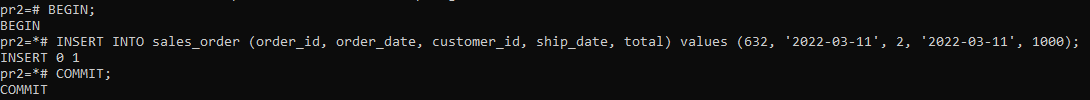
4. Перед началом выполнения задания проверьте, что в таблице sales\_order нет заказов на сумму total 1 000 рублей.

  
Рисунок 4 - Процесс выполнения работы.

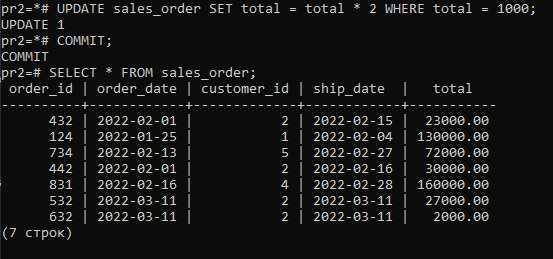
4.1. В первом сеансе начните транзакцию (командой BEGIN). Выполните обновление таблицы sales\_order: увеличьте total в два раза в тех строках, где сумма равна 1 000 рублей.

  
Рисунок 5 - Процесс выполнения работы.

4.2. Во втором сеансе (откройте новое окно psql) вставьте в таблицу sales\_order новый заказ на 1 000 рублей и зафиксируйте транзакцию.

  
Рисунок 6 - Процесс выполнения работы.

4.3. В первом сеансе повторите обновление таблицы sales\_order и зафиксируйте транзакцию.

  
Рисунок 7 - Процесс выполнения работы.

Осталась ли сумма добавленного заказа равной 1 000 рублей? Почему это не так?

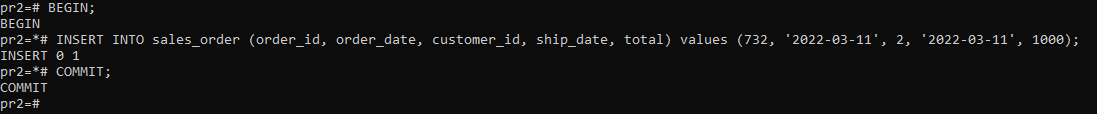
Сумма добавленного заказа второй транзакцией не осталась равной 1000.

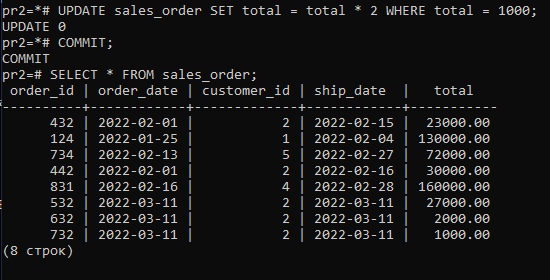
В ходе исполнения транзакций первая транзакция получила данные из таблицы, занесённые второй транзакцией, которая сохранила своё состояние раньше, и смогла их изменить. Это связано с имеющимся уровнем изоляции Read Committed, который допускает аномалию фантомного чтения.

**Уровень изоляции Repeatable Read**

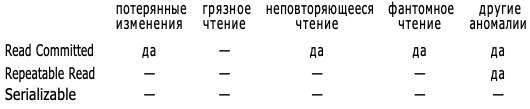
5. Повторите предыдущее упражнение, но начните транзакцию в первом сеансе с уровнем изоляции транзакций Repeatable Read. Объясните различие полученных результатов.

  
Рисунок 8 - Процесс выполнения работы.

  
Рисунок 9 - Процесс выполнения работы.

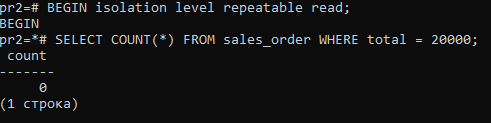
  
Рисунок 10 - Процесс выполнения работы.

В ходе выполнения этих транзакций аномалии фантомного чтения не обнаружена, т.к. уровень изоляции Repeatable Read исключает её, путём изолирования представлений таблиц.

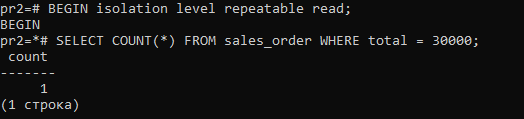
****  
Рисунок 12 - Аномалии при выбранном режиме изоляции в PostgreSQL

6. Выполните указанные действия в двух сеансах:

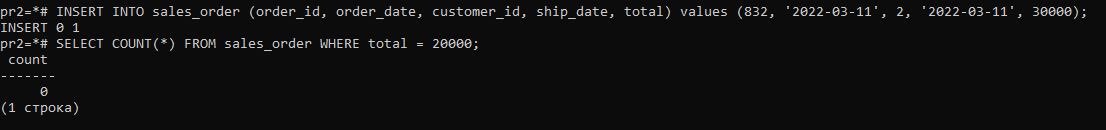
6.1. В первом сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 20 000 рублей.

  
Рисунок 13 - Процесс выполнения работы.

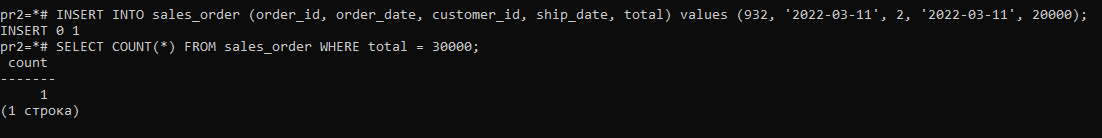
6.2. Во втором сеансе начните новую транзакцию с уровнем изоляции Repeatable Read. Вычислите количество заказов с суммой 30 000 рублей.

  
Рисунок 14 - Процесс выполнения работы.

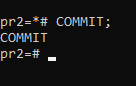
6.3. В первом сеансе добавьте новый заказ на 30 000 рублей и снова вычислите количество заказов с суммой 20 000 рублей.

  
Рисунок 15 - Процесс выполнения работы.

6.4. Во втором сеансе добавьте новый заказ на 20 000 рублей и снова вычислите количество заказов с суммой 30 000 рублей.

  
Рисунок 16 - Процесс выполнения работы.

6.5. Зафиксируйте транзакции в обоих сеансах.

  
Рисунок 17 - Процесс выполнения работы.

Соответствует ли результат ожиданиями? Можно ли сериализовать эти транзакции (иными словами, можно ли представить такой порядок последовательного выполнения этих транзакций, при котором результат совпадет с тем, что получился при параллельном выполнении)?

Нельзя, потому что при последовательном выполнении транзакций вторая транзакция получит обновленную таблицу первой транзакцией, а на данном уровне изоляции это реализовать невозможно.

## Вывод

В ходе выполнения работы были изучены транзакции, точки сохранения состояния транзакций, параллельное выполнение транзакции, аномалии транзакции, уровни изоляций транзакций в PostgreSQL.