МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

**Лабораторная работа №8**

**по курсу «Проектирование баз данных»**

Выполнила: Прудникова А. А.

Группа: М8О-114СВ-24

Преподаватель: Моргунов Е. П.

Москва, 2024

## Задание 2

Воспользуемся таблицей «Самолеты» (aircrafts) или ее копией. Предположим, что мы решили удалить из таблицы те модели, дальность полета которых менее 2 000 км. В таблице представлена одна такая модель — Cessna 208 Caravan, имеющая дальность полета 1 200 км. Для выполнения удаления мы организовали транзакцию. Однако параллельная транзакция, которая, причем, началась раньше, успела обновить таблицу таким образом, что дальность полета самолета Cessna 208 Caravan стала составлять 2 100 км, а вот для самолета Bombardier CRJ-200 она, напротив, уменьшилась до 1 900 км. Таким образом, в результате выполнения операций обновления в таблице по-прежнему присутствует строка, удовлетворяющая первоначальному условию, т. е. значение атрибута range, у которой меньше 2000.

Наша задача: проверить, будет ли в результате выполнения двух транзакций удалена какая-либо строка из таблицы.

Модифицируйте сценарий выполнения транзакций: в первой транзакции вместо фиксации изменений выполните их отмену с помощью команды ROLLBACK и посмотрите, будет ли удалена строка и какая конкретно.

### Запрос

-----терминал 1-----

BEGIN;

SELECT \*

FROM aircrafts\_tmp

WHERE range < 2000;

UPDATE aircrafts\_tmp

SET range = 2100

WHERE aircraft\_code = 'CN1';

UPDATE aircrafts\_tmp

SET range = 1900

WHERE aircraft\_code = 'CR2';

ROLLBACK;

-----терминал 2-----

BEGIN;

DELETE FROM aircrafts\_tmp WHERE range < 2000;

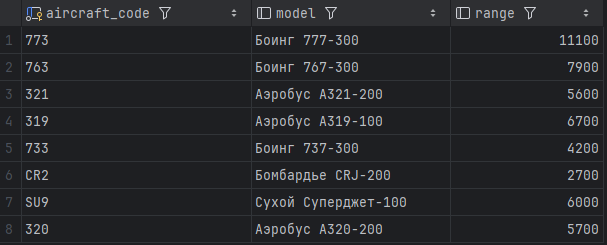
COMMIT;

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

SELECT \*

FROM aircrafts\_tmp;

### Результат



### Решение

Видим, что была удалена строка с самолетом Сессна 208 Караван, поскольку были приняты изменения только из второго терминала.

## Задание 3

Когда говорят о таком феномене, как потерянное обновление, то зачастую в качестве примера приводится операция UPDATE, в которой значение какого-то атрибута изменяется с применением одного из действий арифметики. Например:

UPDATE aircrafts\_tmp

SET range = range + 200

WHERE aircraft\_code = 'CR2';

При выполнении двух и более подобных обновлений в рамках параллельных

транзакций, использующих, например, уровень изоляции Read Committed, будут учтены все такие изменения (что и было показано в тексте главы). Очевидно, что потерянного обновления не происходит.

Предположим, что в одной транзакции будет просто присваиваться новое значение, например, так:

UPDATE aircrafts\_tmp

SET range = 2100

WHERE aircraft\_code = 'CR2';

А в параллельной транзакции будет выполняться аналогичная команда:

UPDATE aircrafts\_tmp

SET range = 2500

WHERE aircraft\_code = 'CR2';

Очевидно, что сохранится только одно из значений атрибута range. Можно ли

говорить, что в такой ситуации имеет место потерянное обновление? Если оно

имеет место, то что можно предпринять для его недопущения? Обоснуйте ваш

ответ.

### Запрос

-----терминал 1-----

BEGIN;

UPDATE aircrafts\_tmp

SET range = 2100

WHERE aircraft\_code = 'CR2';

COMMIT;

-----терминал 2-----

BEGIN;

UPDATE aircrafts\_tmp

SET range = 2500

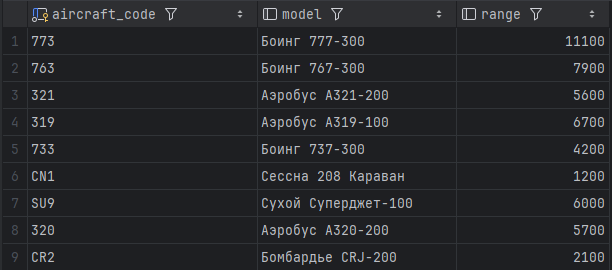
WHERE aircraft\_code = 'CR2';

COMMIT;

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

SELECT \* FROM aircrafts\_tmp;

### Результат



### Решение

В данной ситуации происходит феномен потерянного обновления, так как в результате транзакций сохранилось лишь одно из значений (2100). Оно возникает, когда две или более параллельных транзакции изменяют одно и то же значение без должной синхронизации, и одно из обновлений затирает результаты другого. Чтобы избежать потерянных обновлений, можно использовать блокировки. Например, можно использовать команду SELECT ... FOR UPDATE перед изменением данных. Это заблокирует строку и не позволит другим транзакциям изменить её, пока текущая транзакция не завершится. Также можно изменить уровень изоляции на Repeatable Read или Serializable, являющиеся более строгими при параллельных процессах.