**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РФ»**

**Департамент анализа данных и машинного обучения**

**Отчет по практике №3**

по дисциплине «системы управления базами данных»

Студента группы ПМ23-1

Факультета информационных технологий и анализа больших данных

Тищенко И.С.

Преподаватель

Карамышев Е.П.

Москва 2025

2. Потерянное обновление

Да, имеет место. Например:

- Транзакция 1: UPDATE ... SET range = 2100

- Транзакция 2: UPDATE ... SET range = 2500

Сохранится только последнее изменение.

Решение: Использовать уровень изоляции Serializable или блокировку SELECT FOR UPDATE .

3. Фантомное чтение с INSERT

Если вторая транзакция вставляет новую строку, удовлетворяющую условию первой (например, INSERT INTO aircrafts\_tmp VALUES ('NEW', 6500)), то первая транзакция увидит эту строку при повторном запросе. Это пример фантомного чтения .

4. Эксперименты с SELECT FOR UPDATE

- Подмножество: Вторая транзакция будет ждать завершения первой.

- Надмножество: Блокировка на пересекающиеся строки.

- Пересечение/Нет пересечения: Неблокирующее выполнение для непересекающихся строк .

5. FOR SHARE vs FOR UPDATE

- FOR SHARE блокирует строки для чтения с возможностью обновления другими транзакциями.

- FOR UPDATE блокирует строки для эксклюзивного обновления .

6. Несколько параллельных транзакций

При добавлении 3-4 транзакций возможны:

- Взаимоблокировки при попытке захвата ресурсов в разном порядке.

- Увеличение числа конфликтов на уровне Serializable .

7. Уровень Serializable для бронирования

Да, Serializable предотвращает аномалии, но может вызвать ошибки сериализации. Альтернатива — явные блокировки (SELECT FOR UPDATE) .

8. Индексы и Serializable

Индексы позволяют избежать предикатных блокировок. В примере с таблицей modes обновления разных строк через индекс не конфликтуют, поэтому транзакции фиксируются успешно.

.

10. Смена условий в SELECT

Если транзакции меняют flight\_id в SELECT и INSERT, возникает конфликт сериализации, так как каждая транзакция влияет на данные другой, нарушая последовательную согласованность .