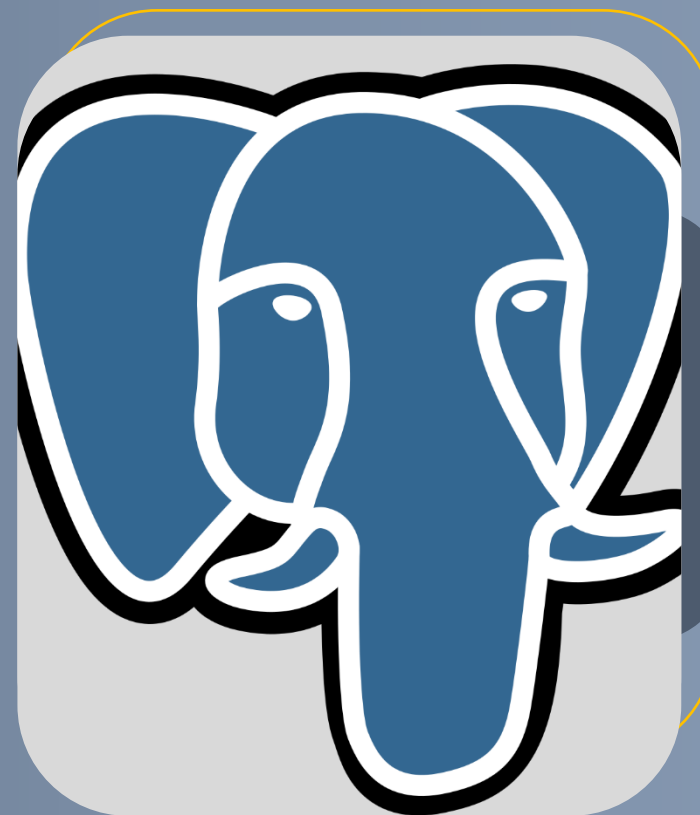




Занятие №7

Понятие транзакции



Практическая работа

1. Необходимо получить список игроков, их роль, гражданство, сумма минут проведенных на поле и количество полученных карточек за весь футбольный сезон.

Поля: **nickname**, **role**, **citizenship**, **sum_minutes**, **sum_cards**

Отсортировать:

- *голкипер, защитник, центральный полузащитник, нападающий;*

- *от большего к меньшему количеству минут на поле.*

2. Создать одно простое представление, и одно материализованное представление.

3. Изменить количество минут у игрока «A.Becker» в игре № 1 с 90 минут на 120 и посмотреть результат в обоих представлениях.

	nickname character varying (50) 🔒	role character varying (20) 🔒	citizenship character varying 🔒	sum_minutes bigint 🔒	sum_cards bigint 🔒
1	A.Becker	GOALKEEPER	Brazil	3370	1
2	Mignolet	GOALKEEPER	Belgia	0	0
3	Karius	GOALKEEPER	Germany	0	0
4	Virgil	DEFENDER	Netherland	3385	1
5	Robertson	DEFENDER	Scotland	3219	4
6	Alexander-Arnold	DEFENDER	British	2435	3
7	Matip	DEFENDER	Germany	1535	3
8	Gomez	DEFENDER	British	1060	2
9	Lovren	DEFENDER	Croatia	986	1
10	Moreno	DEFENDER	Spain	155	0
11	Clyne	DEFENDER	British	108	0
12	Wijnaldum	MIDFIELDER	Netherland	2746	5
13	Fabinho	MIDFIELDER	Brazil	2002	6
14	Henderson	MIDFIELDER	British	1982	2
15	Milner	MIDFIELDER	British	1786	6
16	Keita	MIDFIELDER	Guinea	1394	0
17	Shaqiri	MIDFIELDER	Switzerland	1058	3
18	Lallana	MIDFIELDER	British	465	1
19	Oxlade-Chamberlain	MIDFIELDER	British	19	0
20	Camacho	MIDFIELDER	Portugal	1	0
21	Jones	MIDFIELDER	British	0	0
22	M.Salah	FORWARD	Egypt	3233	1
23	Mane	FORWARD	Senegal	3084	2
24	Firmino	FORWARD	Brazil	2617	0

OLAP и OLTP системы

Оперативная обработка онлайн (OLTP - *Online Transactional Processing*)

Для того чтобы крупные / средние компании могли выполнять свои административные / коммерческие задачи или задачи по продажам, обязательно иметь системы OLTP, которые должны заботиться об огромном количестве транзакций, которые происходят каждый день.

Аналитическая обработка онлайн (OLAP - *Online Analytical Processing*)

На уровне OLAP происходит очень мало транзакций, и они помогают бизнесу принимать лучшие решения.

Специальный набор требований - ACID

ACID — специальный набор требований, предъявляемых к транзакционной системе.

Atomicity (атомарность) — выражается в том, что транзакция должна быть выполнена в целом или не выполнена вовсе.

Consistency (согласованность) — гарантирует, что по мере выполнения транзакций, данные переходят из одного согласованного состояния в другое, то есть транзакция не может разрушить взаимной согласованности данных.

Isolation (изолированность) — локализация пользовательских процессов означает, что конкурирующие за доступ к БД транзакции физически обрабатываются последовательно, изолированно друг от друга, но для пользователей это выглядит, как будто они выполняются параллельно.

Durability (долговечность) — устойчивость к ошибкам — если транзакция завершена успешно, то те изменения в данных, которые были ею произведены, не могут быть потеряны ни при каких обстоятельствах.

Понятие транзакции

Транзакция — это набор операций по работе с базой данных (БД), объединенных в одну атомарную пачку.

Суть транзакции в том, что она объединяет последовательность действий в одну операцию «всё или ничего». Промежуточные состояния внутри последовательности не видны другим транзакциям, и если что-то помешает успешно завершить транзакцию, ни один из результатов этих действий не сохранится в базе данных.

Причины невыполнения команд

Отказ СУБД

- Аппаратная проблема с сервером СУБД
- Программная ошибка в СУБД или операционной системе
- Не хватает места на диске для записи данных

Отказ приложения пользователя

- Аппаратная проблема на клиенте
- Программная ошибка в приложении или операционной системе
- Пользователь прервал работу приложения

Потеря сетевого соединения клиента и сервера СУБД

Транзакция

```
UPDATE accounts SET balance = balance - 15000  
WHERE account_number = 1234567
```

```
UPDATE accounts SET balance = balance + 15000  
WHERE account_number = 9876543
```

Автоматическая фиксация транзакций

Автоматическая фиксация транзакций

- Специальный режим работы СУБД
- После выполнения каждой команды SQL автоматически фиксируется транзакция
- Нет возможности отменить изменения

Использование режима автоматической фиксации транзакций

- PostgreSQL – включено по умолчанию
- Oracle – выключено по умолчанию

Транзакции в PostgreSQL

Postgres Pro на самом деле обрабатывает каждый SQL-оператор как транзакцию. Если вы не вставите команду BEGIN, то каждый отдельный оператор будет неявно окружён командами BEGIN и COMMIT (в случае успешного завершения). Группу операторов, окружённых командами BEGIN и COMMIT иногда называют *блоком транзакции*.

Завершение транзакций

Команды COMMIT или ROLLBACK

- Явное подтверждение или отмена транзакции

Запуск команд создания, изменения или удаления таблиц

- Создание, изменение или удаление таблицы не может быть отменено
- Перед запуском команды предыдущая транзакция фиксируется

Сбой сервера СУБД

- Откат всех незавершенных транзакций после восстановления сервера

Сбой клиента СУБД

- Откат незавершенной транзакции после тайм-аута

Итоги по теме транзакций

Транзакции в базах данных

- Последовательность команд, которые должны быть выполнены все полностью, или не выполнены совсем

Начала транзакции

- Команда BEGIN
- Автоматическая фиксация транзакций

Завершение транзакции

- Команда COMMIT – фиксация транзакции
- Команда ROLLBACK – откат транзакции

Пример транзакции в PostgreSQL

BEGIN;

UPDATE table_name **SET** field_1 = NULL **WHERE** id = 5;

DELETE FROM other_table **WHERE** id = 17;

COMMIT; - выполнит группу операций как одно целое, запрос на изменения поля и удаления записи не вступят в силу, пока мы не подтвердим операции, ключевым словом **COMMIT**.

Пример транзакции в MySQL

START TRANSACTION;

UPDATE table_name **SET** field_1 = NULL **WHERE** id = 5;

DELETE FROM other_table **WHERE** id = 17;

COMMIT; - выполнит группу операций как одно целое, запрос на изменения поля и удаления записи не вступят в силу, пока мы не подтвердим операции, ключевым словом **COMMIT**.

Отмена транзакции ROLLBACK

BEGIN;

UPDATE table_name **SET** field_1 = NULL **WHERE** id = 5;

DELETE FROM other_table **WHERE** id = 17;

ROLLBACK; - отмена транзакции, т.е. запрос на изменения поля и удаления записи не вступят в силу, и таблицы останутся без изменений.

Контрольные точки SAVEPOINT

BEGIN;

UPDATE table_name **SET** field_1 = NULL **WHERE** id = 5;

DELETE FROM other_table **WHERE** id = 17;

SAVEPOINT point_name;

DELETE FROM test_table **WHERE** id = 75;

ROLLBACK TO SAVEPOINT point_name;

COMMIT; - в результате данной транзакции будет выполнено только два запроса: изменение записи в таблице **table_name** и удаление записи из таблицы **other_table**.