

# Transaction Control Language, TCL

## И

# Data Control Language, DCL

# Проблема

- Классическая проблема согласованности работы системы с важными данными – перевод денег с одного счета на другой:
- `UPDATE account SET balance = balance + 10 WHERE id = 1;`
- `UPDATE account SET balance = balance - 10 WHERE id = 2;`
- `UPDATE account SET balance = balance - 10 WHERE id = 2;`
- `UPDATE account SET balance = balance + 10 WHERE id = 1;`

	id	name	balance
>	1	Иванов Иван Иванович	10
	2	Петров Петр Петрович	30

# Определение

- **Транзакция** (Transaction) — группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными.
- Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта.

# Требования к транзакциям

- Транзакционная система, чтобы обеспечить надёжную работу должна выполнять требования ACID (Atomicity, Consistency, Isolation, Durability):

- **Atomicity — Атомарность**

Атомарность гарантирует, что никакая транзакция не будет зафиксирована в системе частично. Будут либо выполнены все её подоперации, либо не выполнено ни одной.

- **Consistency — Согласованность**

Каждая успешная транзакция фиксирует только допустимые результаты. Таким образом система всегда остается в согласованном состоянии.

- **Isolation — Изолированность**

Во время выполнения транзакции параллельные транзакции не должны оказывать влияние на её результат.

- **Durability — Долговечность**

Независимо от проблем на нижних уровнях (к примеру, обесточивание системы или сбой в оборудовании) изменения, сделанные успешно завершённой транзакцией, должны остаться сохранёнными после возвращения системы в работу.

# TCL

- TCL (Transaction Control Language) является подмножеством языка SQL, которое позволяет управлять транзакциями и состоит из команд: COMMIT, ROLLBACK, SAVEPOINT.
- Новую транзакцию начинает сервер базы данных выполняя любую команду из подмножества DML: **INSERT, UPDATE, DELETE;**
- Для того чтобы подтвердить изменения используется команда **COMMIT;**
- Для того чтобы откатить изменения используется команда **ROLLBACK;**
- Для того чтобы создать промежуточную точку сохранения используется команда **SAVEPOINT.**

# Пример

- `/*SET AUTOCOMMIT TO FALSE*/`
- `UPDATE account SET balance = balance + 10 WHERE id = 1;`
- `UPDATE account SET balance = balance - 10 WHERE id = 2;`
- `ROLLBACK;`

- `/*SET AUTOCOMMIT TO FALSE*/`
- `UPDATE account SET balance = balance + 10 WHERE id = 1;`
- `UPDATE account SET balance = balance - 10 WHERE id = 2;`
- `COMMIT;`

- `/*SET AUTOCOMMIT TO FALSE*/`
- `UPDATE account SET balance = balance + 10 WHERE id = 1;`
- `SAVEPOINT sp_name;`
- `UPDATE account SET balance = balance - 10 WHERE id = 2;`
- `ROLLBACK TO SAVEPOINT sp_name;`
- `COMMIT;`

# Разные команды

- `INSERT INTO account VALUES (3, 'Тест', 10000);`
  - `UPDATE account SET balance = balance - 10 WHERE id = 3;`
  - `DELETE FROM account WHERE id=3;`
  - `COMMIT;`
- 
- `INSERT INTO account VALUES (3, 'Тест', 10000);`
  - `INSERT INTO account VALUES (4, 'Тест', 10000);`
  - `SAVEPOINT test;`
  - `INSERT INTO account VALUES (5, 'Тест', 10000);`
  - `UPDATE account SET balance = balance - 10 WHERE id = 3;`
  - `DELETE FROM account WHERE id=3;`
  - `ROLLBACK TO SAVEPOINT test;`
  - `COMMIT;`
- 
- `DELETE FROM account;` vs `TRUNCATE account`

# DCL

- DCL (Data Control Language) является подмножеством языка SQL, которое позволяет устанавливать доступ к данным и состоит из команд: GRANT, REVOKE, DENY.
- Для того чтобы назначить привилегию на выбранное действие пользователю используется команда **GRANT**;
- Для того чтобы отозвать ранее выделенную привилегию используется команда **REVOKE**;
- **DENY** задает запрет, имеющий приоритет над разрешением.



# Применение DCL

- Вы владелец крупной системы, которая работает с огромной базой данных и Вы хотите нанять стороннего разработчика, чтобы он выполнил простенькую задачу, для которой нужен доступ только на чтение и к одной таблице базы данных.
- Очень глупо давать полный доступ ко всем данным, во первых сторонний разработчик может удалить случайно (или преднамеренно) данные, а во вторых может их скачать и продать Вашим конкурентам!
- Для решения этой проблемы используются операторы DCL: Вам необходимо создать отдельную роль базы данных и предоставить ей доступ только на чтение и только к одной таблице.

# Применение DCL на практике

- Допустим необходимо предоставить доступ только на чтение к таблице account:
  - 1) Создаем отдельную роль (readonly);
  - 2) Разрешаем пользователю подключиться к базе данных:  
`GRANT CONNECT ON DATABASE devstudy TO readonly;`
  - 3) Разрешаем использовать схему PUBLIC:  
`GRANT USAGE ON SCHEMA public TO readonly;`
  - 4) Разрешаем пользователю выполнять операторы SELECT над таблицей account.  
`GRANT SELECT ON account TO readonly;`
  - 5) Возможно установить привелегию на отдельный столбец.

# Отзыв привелегий

```
REVOKE CONNECT ON DATABASE devstudy FROM readonly;
```

```
REVOKE USAGE ON SCHEMA public FROM readonly;
```

```
REVOKE SELECT ON account FROM readonly;
```

```
/*Удаление readonly роли*/
```

```
DROP ROLE readonly;
```

# Выводы

1. **Транзакция** (Transaction) — группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными;
2. Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще и тогда она не должна произвести никакого эффекта;
3. Требования ACID (Атомарность, Согласованность, Изолированность и Долговечность) описывают требования, которые должны быть реализованы любой транзакцией для представления последовательности операций как одно логическое целое;
4. TCL (Transaction Control Language) является подмножеством языка SQL, которое позволяет управлять транзакциями и состоит из команд: COMMIT, ROLLBACK, SAVEPOINT.
5. Новую транзакцию начинает сервер базы данных выполняя любую команду из подмножества DML: **INSERT, UPDATE, DELETE**;
6. Для того чтобы подтвердить изменения используется команда **COMMIT**. Для того чтобы откатить изменения используется команда **ROLLBACK**. Для того чтобы создать промежуточную точку сохранения используется команда **SAVEPOINT**;
7. DCL (Data Control Language) является подмножеством языка SQL, которое позволяет устанавливать доступ к данным и состоит из команд: GRANT, REVOKE, DENY.
8. Для того чтобы назначить привилегию на выбранное действие пользователю используется команда **GRANT**;
9. Для того чтобы отозвать ранее выделенную привилегию используется команда **REVOKE**;
10. Назначение привилегий на отдельные таблицы (столбцы) позволяет контролировать доступ к данным на уровне СУБД.