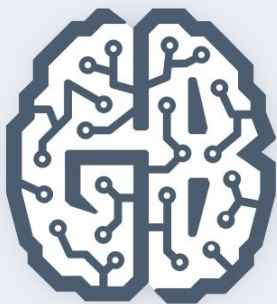


## Урок 6



# Базы данных

Транзакции. Переменные. Временные  
таблицы. Prepared-выражения.  
Представления.

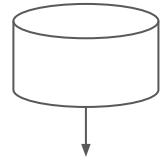
# Транзакции

- Транзакции
- Ключевые слова COMMIT и ROLLBACK
- Точки сохранения
- Режим автозавершения транзакций
- Принцип ACID
- Уровни изоляции



# Транзакции

Состояние  
базы до  
транзакции



SELECT



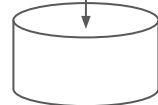
UPDATE



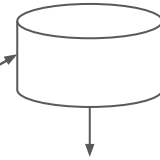
DELETE



INSERT



Состояние  
базы после  
транзакции



SELECT



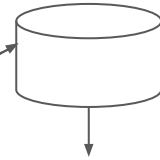
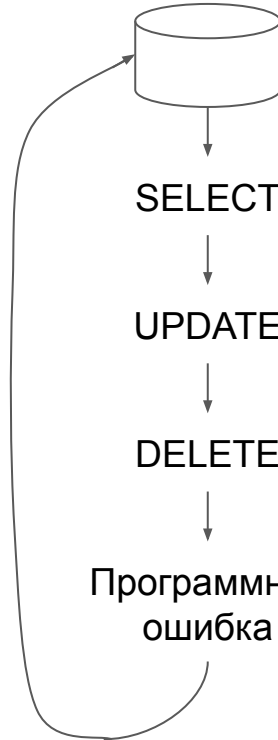
UPDATE



DELETE



Программная  
ошибка



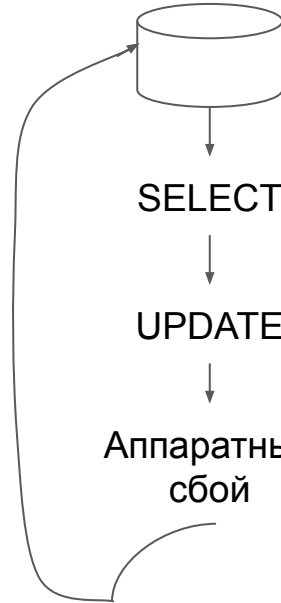
SELECT



UPDATE

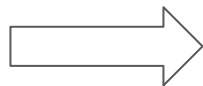


Аппаратный  
сбой



# Перемещение денежных средств

id	user_id	total
1	4	5000
2	3	0
3	2	200
4	NULL	25000



id	user_id	total
1	4	3000
2	9	0
3	2	200
4	4	27000



# Перевод средств

- Убедиться, что остаток на счете клиента больше 2000 рублей
- Вычесть 2000 рублей со счета клиента
- Добавить 2000 к счету интернет-магазина



# Необратимые команды

- CREATE INDEX
- DROP INDEX
- CREATE TABLE
- DROP TABLE
- TRUNCATE TABLE
- ALTER TABLE
- RENAME TABLE
- CREATE DATABASE
- DROP DATABASE
- ALTER DATABASE



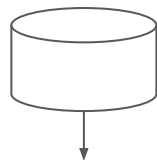
# Неявное завершение транзакций

- ALTER TABLE
- BEGIN
- CREATE INDEX
- CREATE TABLE
- CREATE DATABASE
- DROP DATABASE
- DROP INDEX
- DROP TABLE
- DROP DATABASE
- LOAD MASTER DATA
- LOCK TABLES
- RENAME
- SET AUTOCOMMIT=1
- START TRANSACTION
- TRUNCATE TABLE



# Точки сохранения

Состояние  
базы до  
транзакции

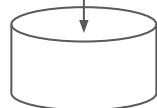


SELECT

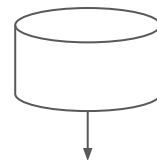
SAVEPOINT

DELETE

INSERT



Состояние  
базы после  
транзакции



SELECT

SAVEPOINT

DELETE

Программная  
ошибка





# Команды для точек сохранения

- SAVEPOINT
- ROLLBACK TO SAVEPOINT



# ACID

- Atomicity — атомарность
- Consistency — согласованность
- Isolation — изолированность
- Durability — сохраняемость



# Уровни изоляции

- READ UNCOMMITTED
- READ COMMITTED
- REPEATABLE READ
- SERIALIZABLE



## Урок 6



# Базы данных

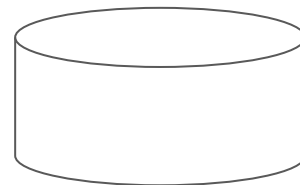
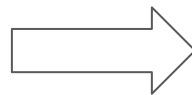
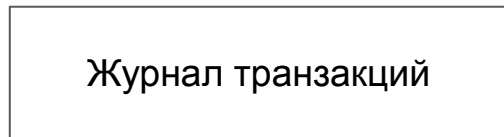
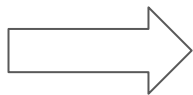
Транзакции. Переменные. Временные  
таблицы. Prepared-выражения.  
Представления.

# Внутренняя реализация транзакций

- Взаимоблокировка
- Журнал транзакций
- Управление режимом сохранения транзакций
- MVCC
- Связь MVCC с уровнями изоляции



# Журнал транзакций



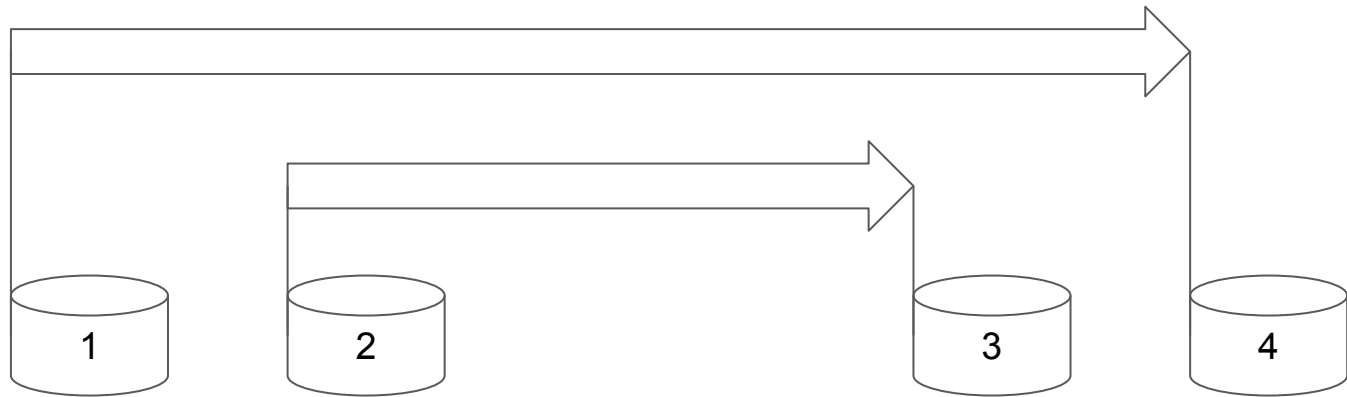
# Управление режимом сохранения журнала транзакций

`innodb_flush_log_at_trx_commit`

- 0 — сохранение журнала раз в секунду
- 1 — сохранение после каждой транзакции
- 2 — сохранение журнала раз в секунду и после каждой транзакции



# MVCC





# Уровни изоляции и MVCC

- READ UNCOMMITTED
- READ COMMITTED
- REPEATABLE READ
- SERIALIZABLE



## Урок 6



# Базы данных

Транзакции. Переменные. Временные  
таблицы. Prepared-выражения.  
Представления.

# Переменные, временные таблицы и динамические запросы

- Пользовательские переменные
- Системные переменные
- Временные таблицы
- Динамические запросы



# Типы системных переменных

- GLOBAL — глобальные
- SESSION — сеансовые



# Временные таблицы

```
CREATE TEMPORARY TABLE table_name (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(255),  
  ...  
);
```



## Урок 6



# Базы данных

Транзакции. Переменные. Временные  
таблицы. Prepared-выражения.  
Представления.

# Представления

- Создание представлений
- Вертикальные и горизонтальные представления
- Вставка записей в представление
- Обновление представлений
- Управление представлениями



# Представление таблицы products

id	name	catalog_id
1	Intel Core i3-8100	1
2	Intel Core i5-7400	1

id	name	description	price	catalog_id
1	Intel Core i3-8100	Процессор для...	7890.00	1
2	Intel Core i5-7400	Процессор для...	12700.00	1





# Алгоритм формирования конечного запроса

- MERGE
- TEMPTABLE
- UNDEFINED



# Домашнее задание

- В базе данных shop и sample присутствуют одни и те же таблицы, учебной базы данных. Переместите запись id = 1 из таблицы shop.users в таблицу sample.users. Используйте транзакции.
- Создайте представление, которое выводит название name товарной позиции из таблицы products и соответствующее название каталога name из таблицы catalogs.



# Домашнее задание

- **(по желанию)** Пусть имеется таблица с календарным полем `created_at`. В ней размещены разряженные календарные записи за август 2018 года '2018-08-01', '2016-08-04', '2018-08-16' и 2018-08-17. Составьте запрос, который выводит полный список дат за август, выставляя в соседнем поле значение 1, если дата присутствует в исходном таблице и 0, если она отсутствует.
- **(по желанию)** Пусть имеется любая таблица с календарным полем `created_at`. Создайте запрос, который удаляет устаревшие записи из таблицы, оставляя только 5 самых свежих записей.

