

Яндекс.Практикум

PostgreSQL и Go

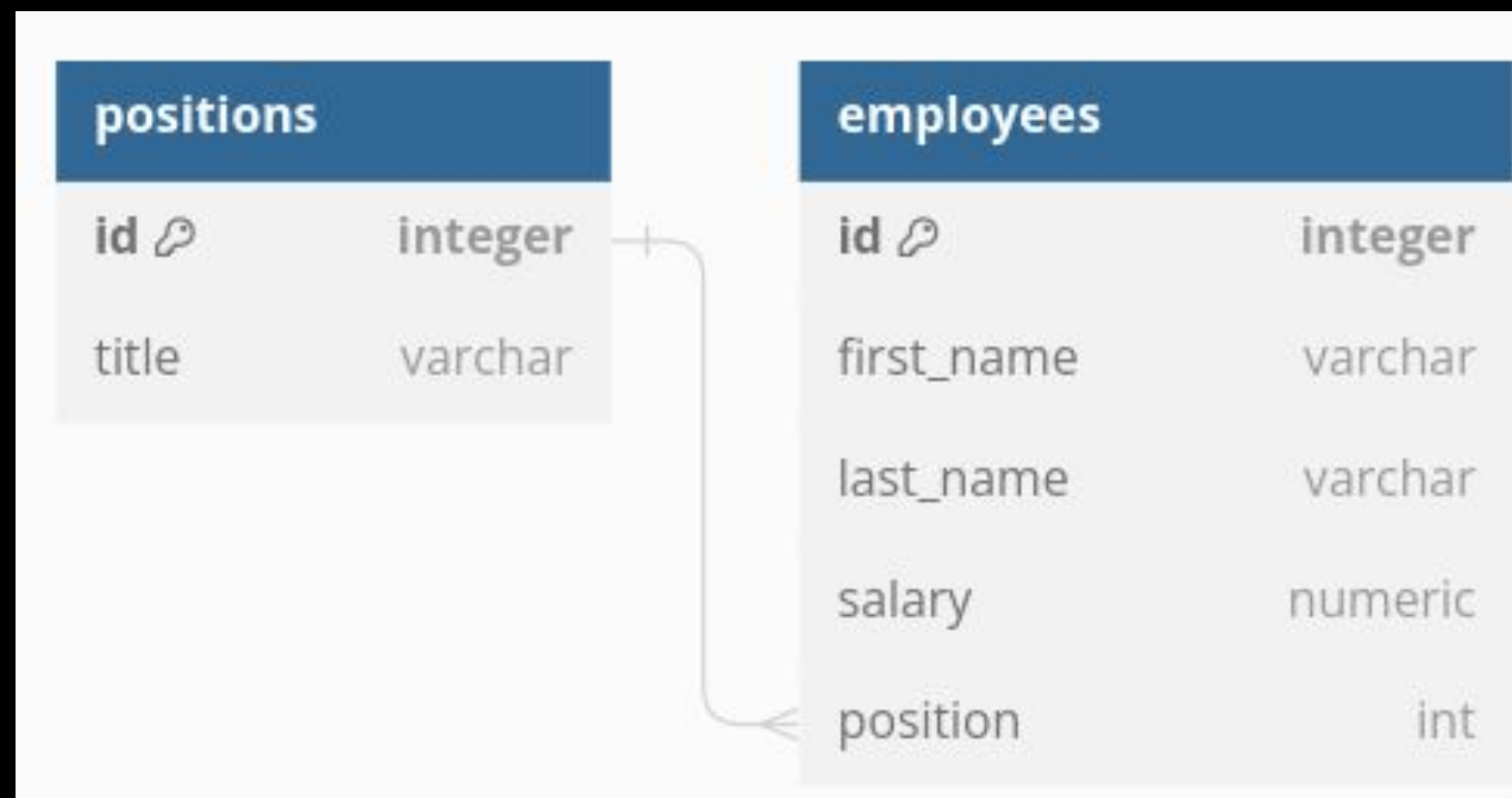
Яндекс Практикум

# Что нас сегодня ждет

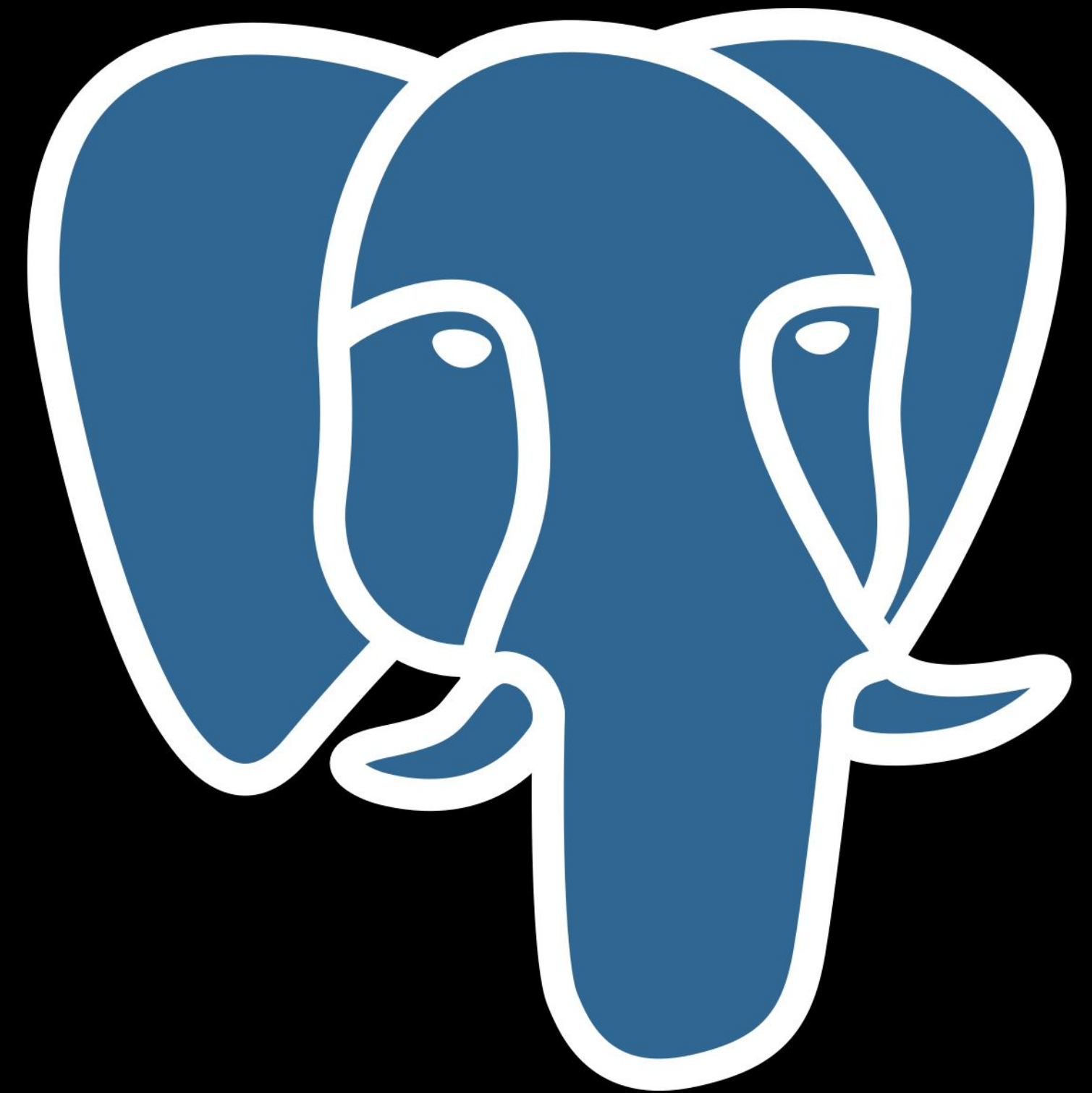
- Поговорим про теорему CAP и ACID-транзакции
- Разберем работу с PostgreSQL из Go-приложения
- Коснемся темы оптимизации запросов

# PostgreSQL

PostgreSQL – реляционная СУБД с открытым исходным кодом.



<https://dbdiagram.io>



# Теорема CAP (теорема Брюера)

Применима к распределенным  
хранилищам данных



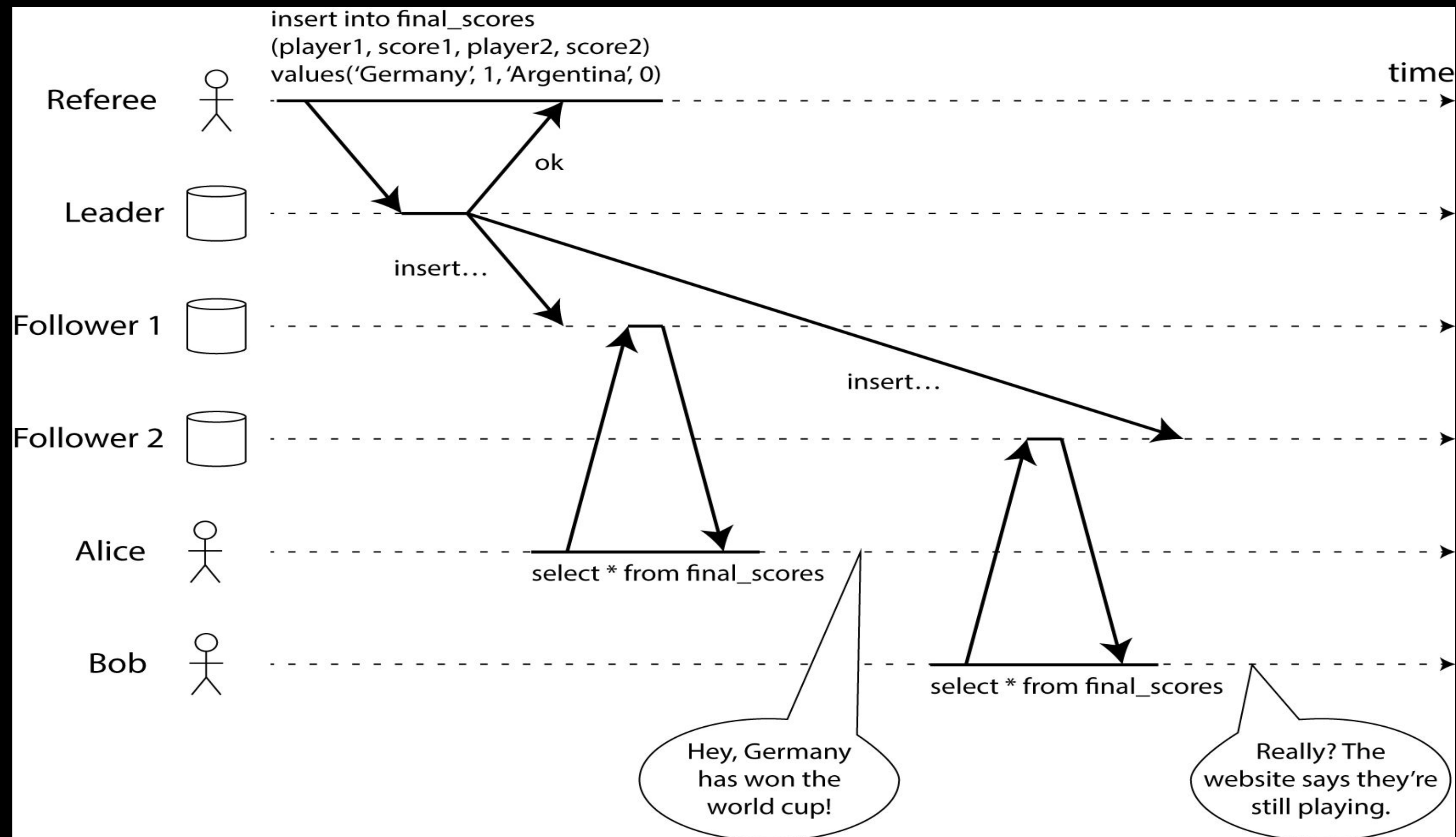
# Теорема CAP. Согласованность

В теореме CAP Consistency = линейризуемость (*linearizability*)

Пусть операция В началась после операции А, которая завершилась успешно.

Тогда операция В должна наблюдать систему в том состоянии, которое возникло после выполнения операции А, или в более НОВОМ СОСТОЯНИИ.

# Теорема CAP. Согласованность



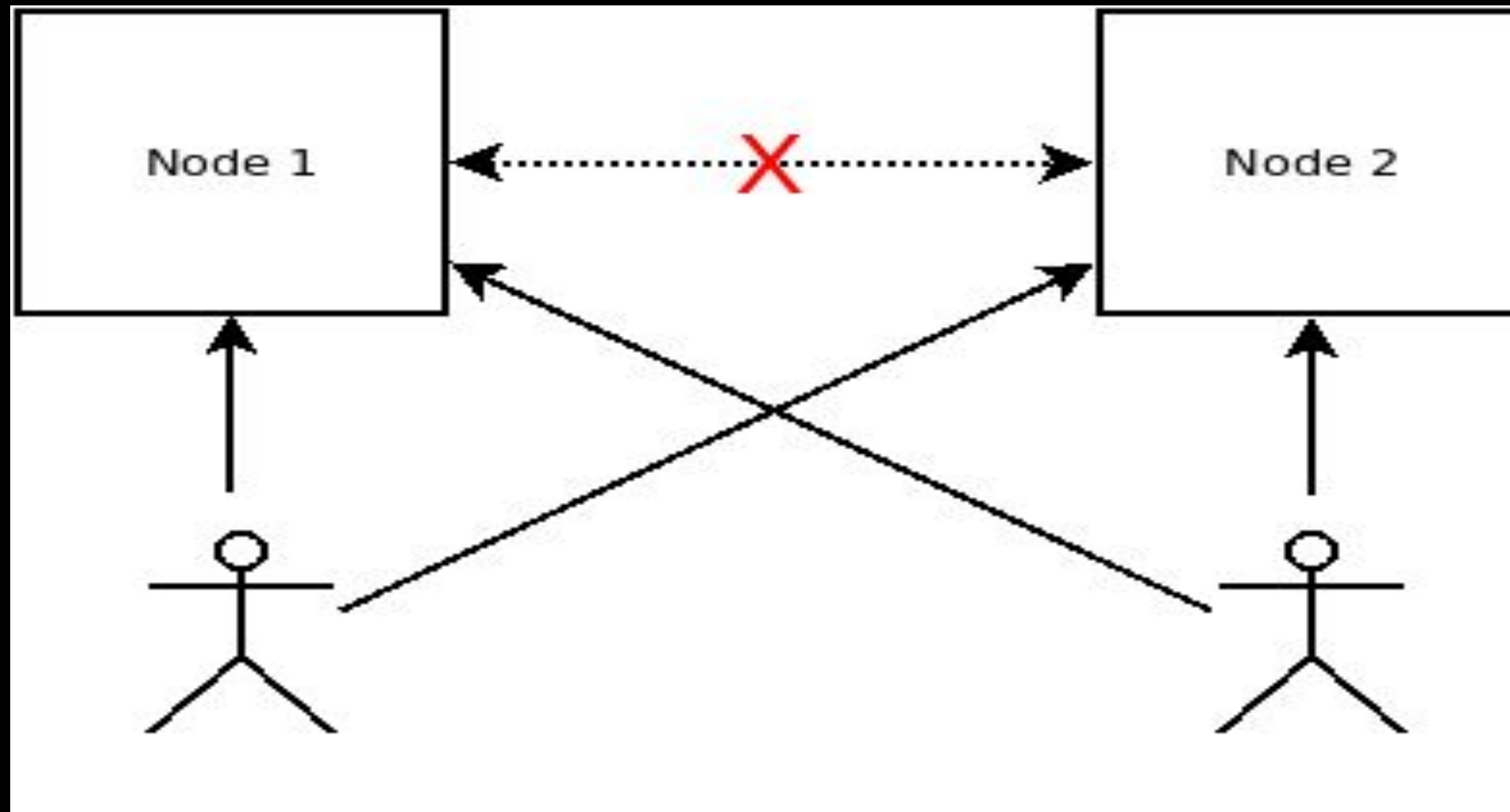
# Теорема CAP. Доступность

На каждый запрос к исправному узлу система должна отправить корректный ответ.

Availability в теореме CAP  $\neq$  High Availability: пусть в БД работает только один узел и он начал перезагрузку, которая длилась час, после чего узел заработал штатно  $\Rightarrow$  чудовищно с т.з. HA, нормально с т.з. теоремы CAP.

# Теорема CAP. Устойчивость к разделению

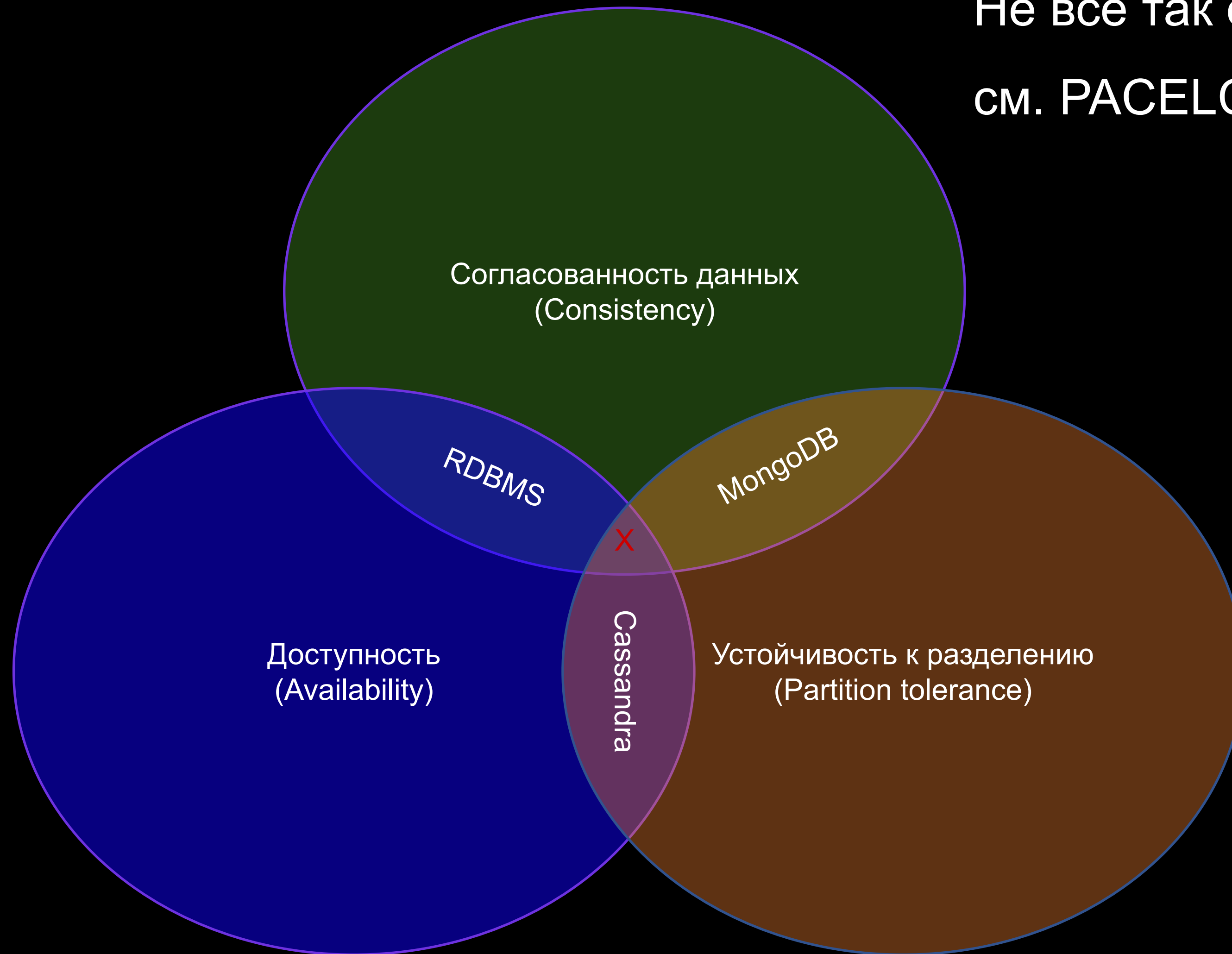
Устойчивость к разделению: допускается потеря случайного числа сообщений от одного узла к другому (разрыв канала связи); при этом БД не отправляет клиенту неверные ответы.





# Теорема CAP (теорема Брюера)

Не все так однозначно,  
см. PACELC



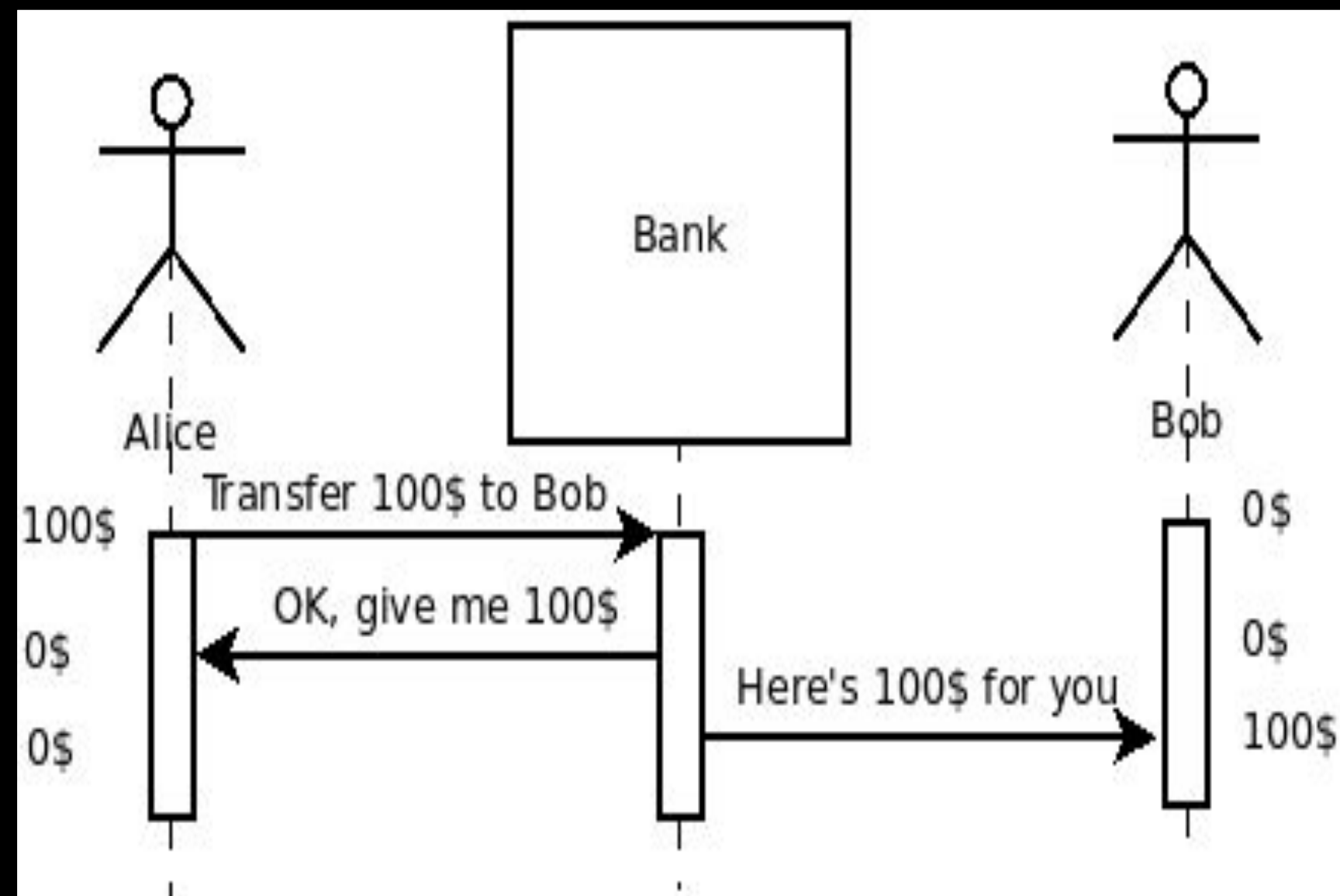
# Транзакции

Транзакция – группировка нескольких операций с данными в БД в одну логическую единицу для:

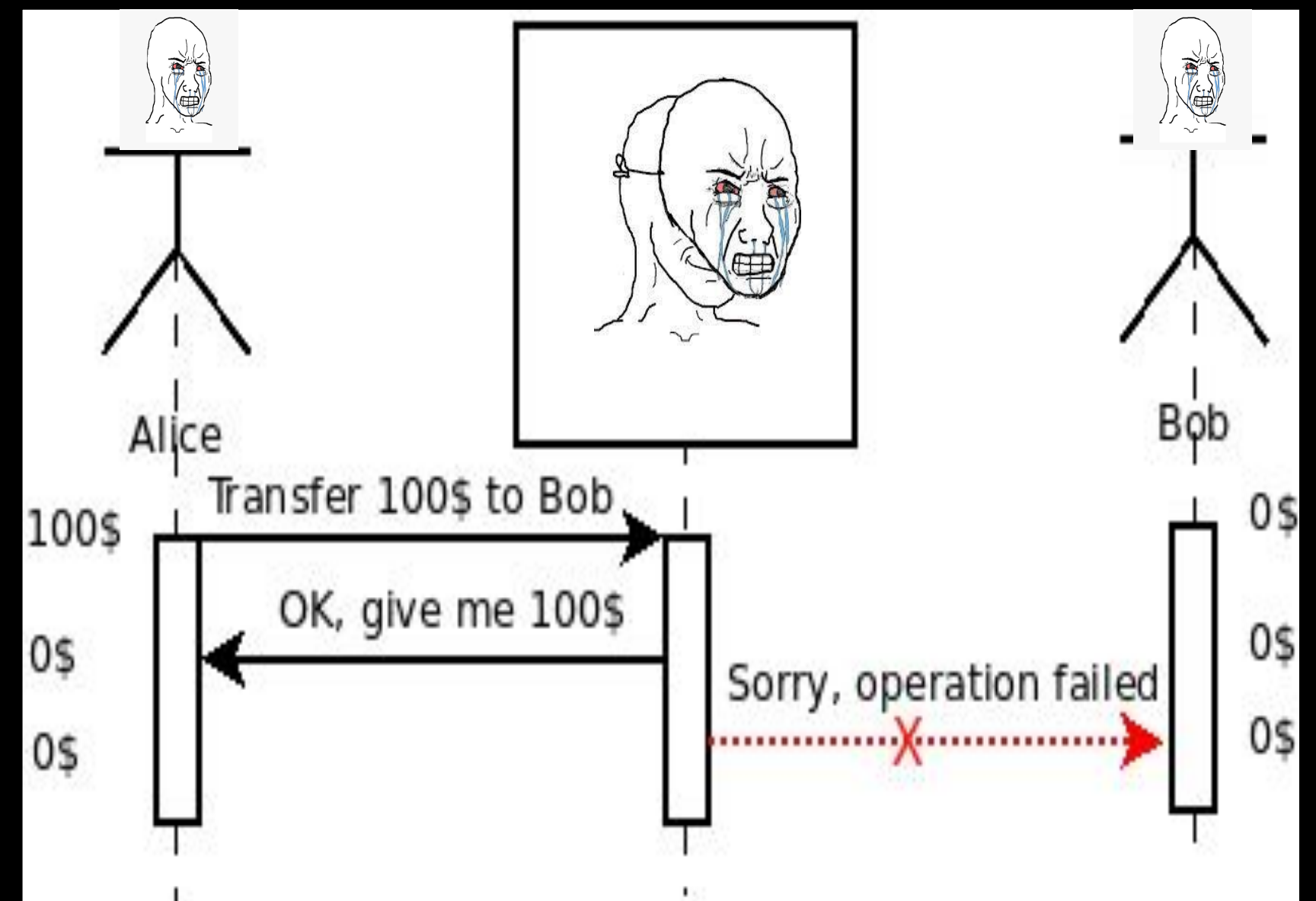
- обеспечения согласованности данных даже в случае отказа системы во время выполнения операций из одной транзакции
- возможности параллельного доступа к одним и тем же данным

# Транзакции. Согласованность

Что хотели:

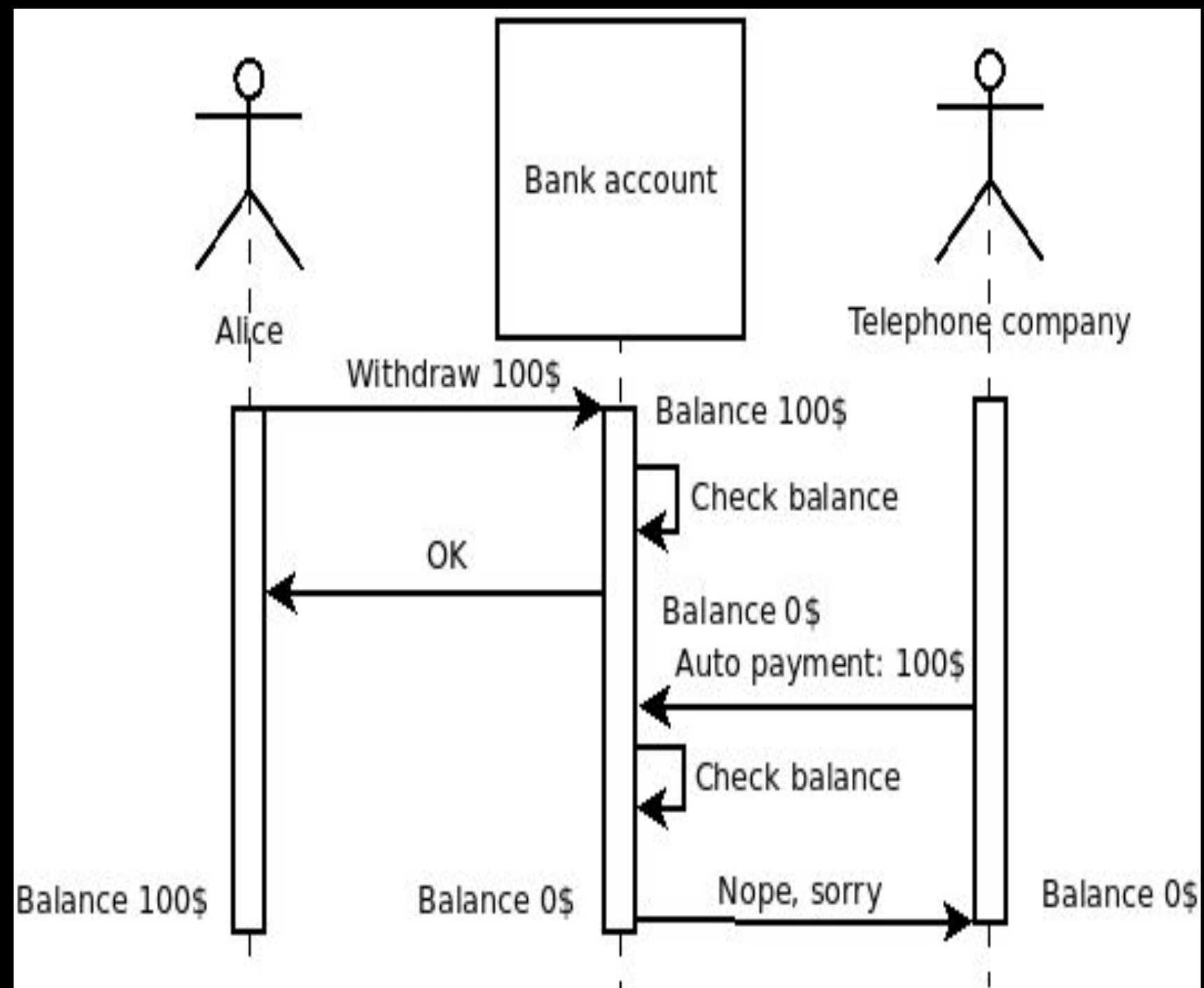


Что получили без согласованности:

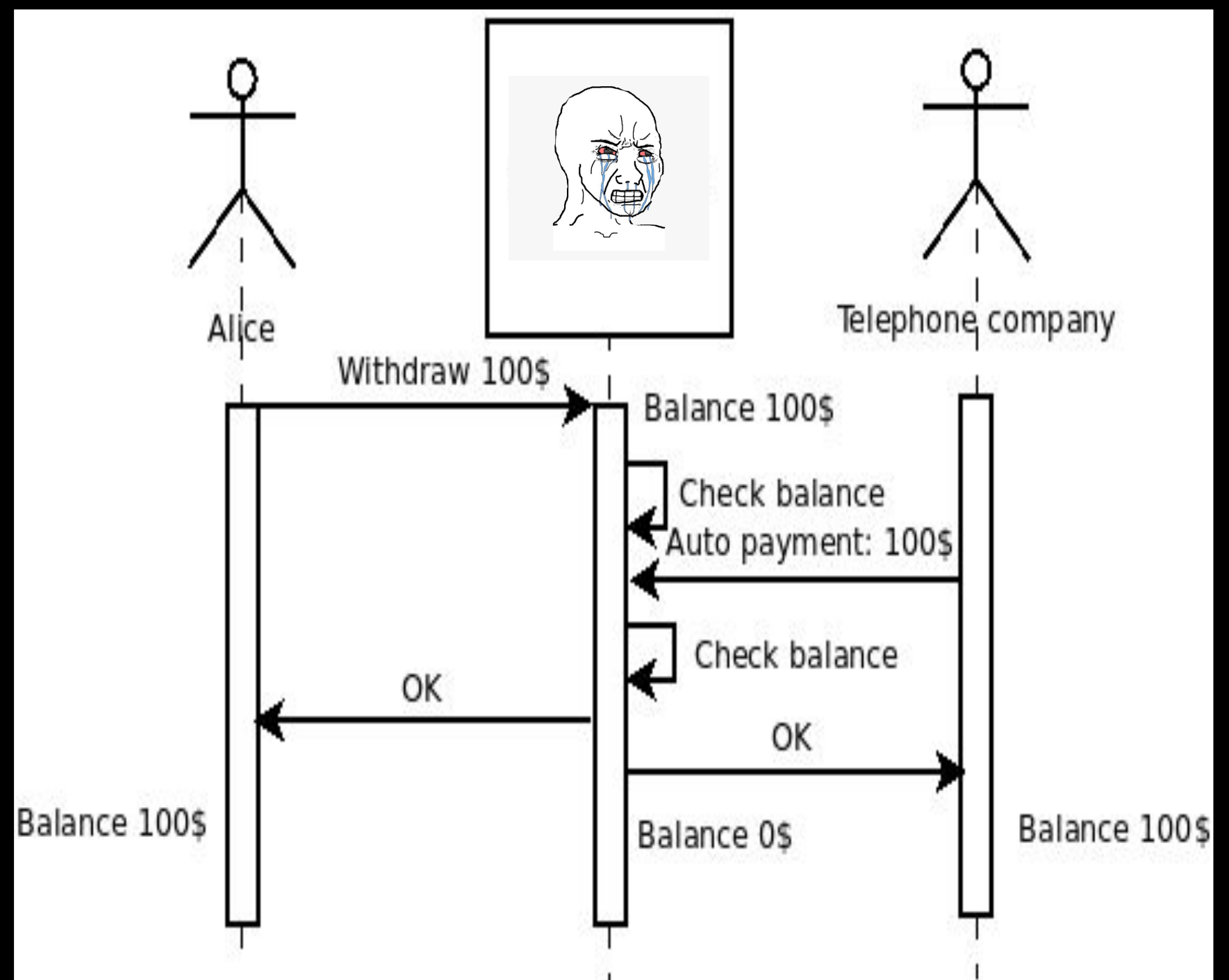


# Транзакции. Изолированность

Что хотели:



Что получили без  
изолированности:



# Транзакции. ACID

Выполнение требований ACID гарантирует надежную и предсказуемую работу транзакционной системы.

- **A - Atomicity**, транзакция не может быть зафиксирована частично, все или ничего, COMMIT/ROLLBACK
- **C - Consistency**, при завершении транзакции данные должны оставаться в валидном состоянии (e.g. зарплата всегда положительна)
- **I - Isolation**, параллельное выполнение транзакций должно приводить к такому же результату, как если бы транзакции выполнялись последовательно (*сериализуемость*)
- **D - durability**, записанные данные не будут утрачены в результате технического сбоя

# Работа с PostgreSQL

См. скрипт урока

Спасибо за внимание!

Вопросы?