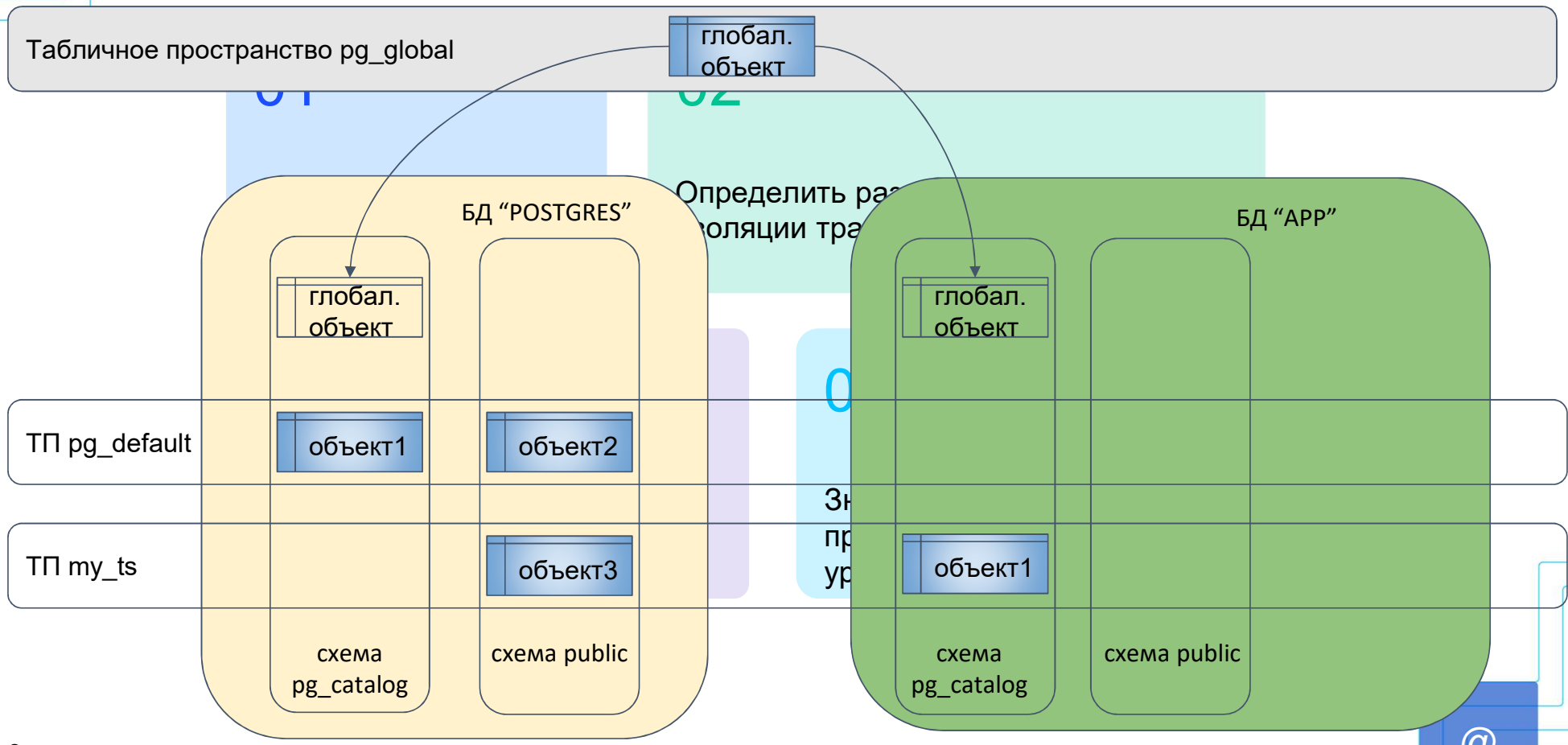
БЛОК 3. АРХИТЕКТУРА

ИЗОЛЯЦИЯ И МНОГОВЕРСИОННОСТЬ

{ }

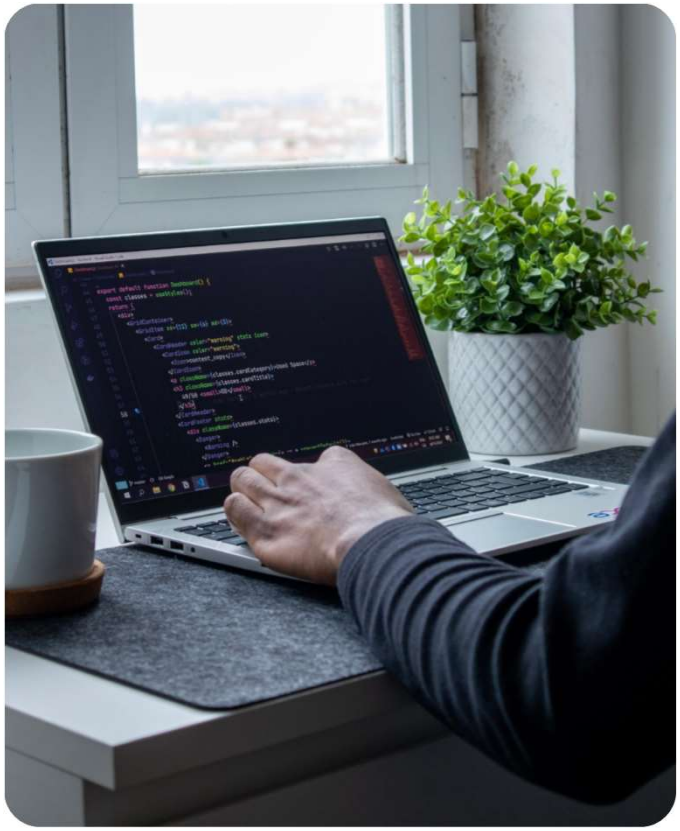
begin

ЦЕЛЬ УРОКА


Табличное пространство pg_global
глобал. объект
БД “POSTGRES”
БД “APP”
глобал. объект
глобал. объект
ТП pg_default
ТП my_ts
объект1
объект2
объект3
объект1
схема pg_catalog
схема public
схема pg_catalog
схема public

СОДЕРЖАНИЕ УРОКА

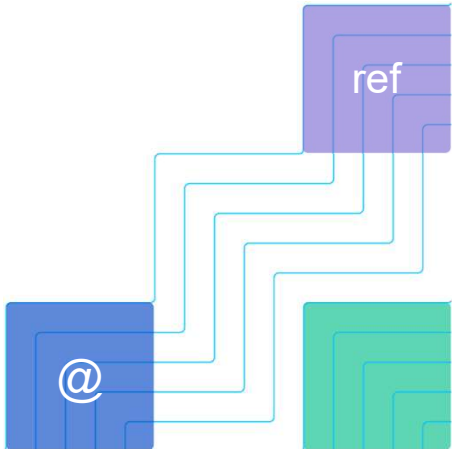
1. ACID
2. MVCC
3. Уровни изоляции транзакций



МНОЖЕСТВЕННАЯ ПАРАЛЛЕЛЬНАЯ НАГРУЗКА

СУБД решает много задач и в том числе
одну из непростых проблем:

→ как обеспечить параллельную работу множества
сессий (concurrency),
которые модифицируют данные
так, чтобы они не мешали друг другу
ни с точки зрения чтения ни с точки зрения записи
и обеспечивали целостность данных - т.н. consistency
и их надежность - т.н. durability?



ACID

→ Ответ — транзакционные системы, OLTP — Online Transaction Processing

DURABILITY
Долговечность

ATOMICITY
Атомарность

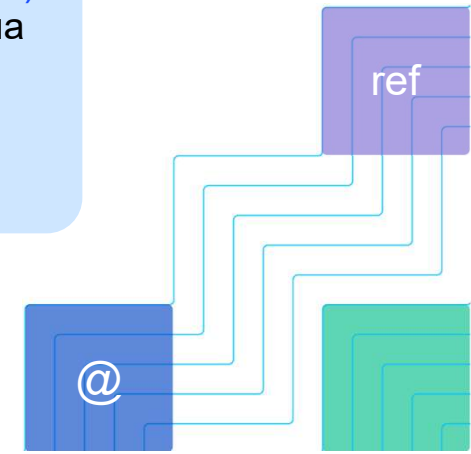
CONSISTENCY
Согласованность

ISOLATION
Изолированность

TRANSACTION

Транзакция:

- множество операций, выполняемое приложением
- которое переводит базу данных из одного корректного состояния в другое корректное состояние (согласованность)
- при условии, что транзакция выполнена полностью (атомарность)
- и без помех со стороны других транзакций (изолированность)



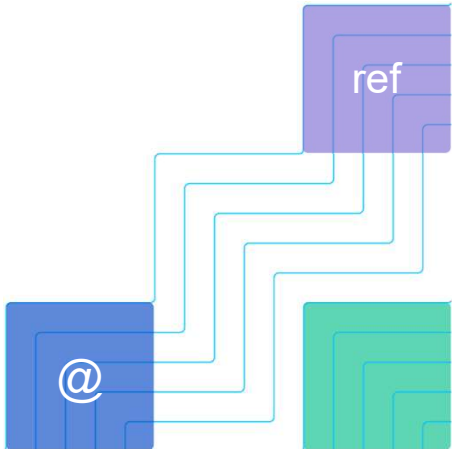
ACID — А ГДЕ ЖЕ НАДЕЖНОСТЬ?

ARIES — Algorithms for Recovery and Isolation Exploiting Semantics

- logging
- undo
- redo
- checkpoints

MVCC — Multiversion Concurrency Control

- copy-on-write
- каждый пользователь работает со снимком БД
- вносимые пользователем изменения не видны другим до фиксации транзакции



POSTGRESQL MVCC

Tuple multi versioning

Многоверсионность

Данные не удаляются в процессе обработки транзакций

Создаются новые версии записей

Дорого для операции update

POSTGRESQL MVCC, XMIN & XMAX, CMIN & CMAX

В каждой таблице есть четыре скрытые колонки

XMIN

Идентификатор транзакции которая создала данную версию записи

XMAX

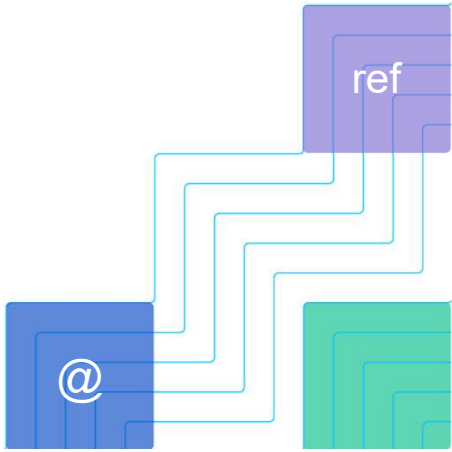
Идентификатор транзакции которая удалила данную версию записи

CMIN

Порядковый номер команды в транзакции, добавившей запись

CMAX

Номер команды в транзакции, удалившей запись



POSTGRESQL MVCC DML

INSERT

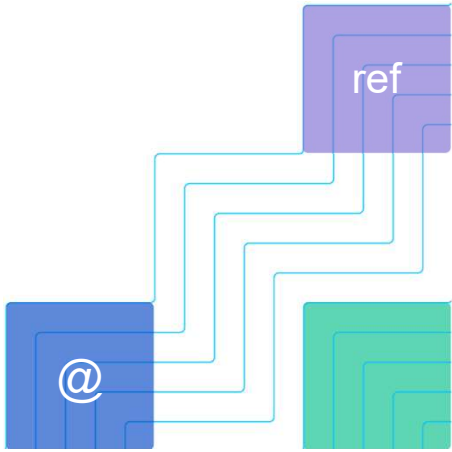
Добавляется новая запись с xmin=txid_current() и xmax=0

UPDATE

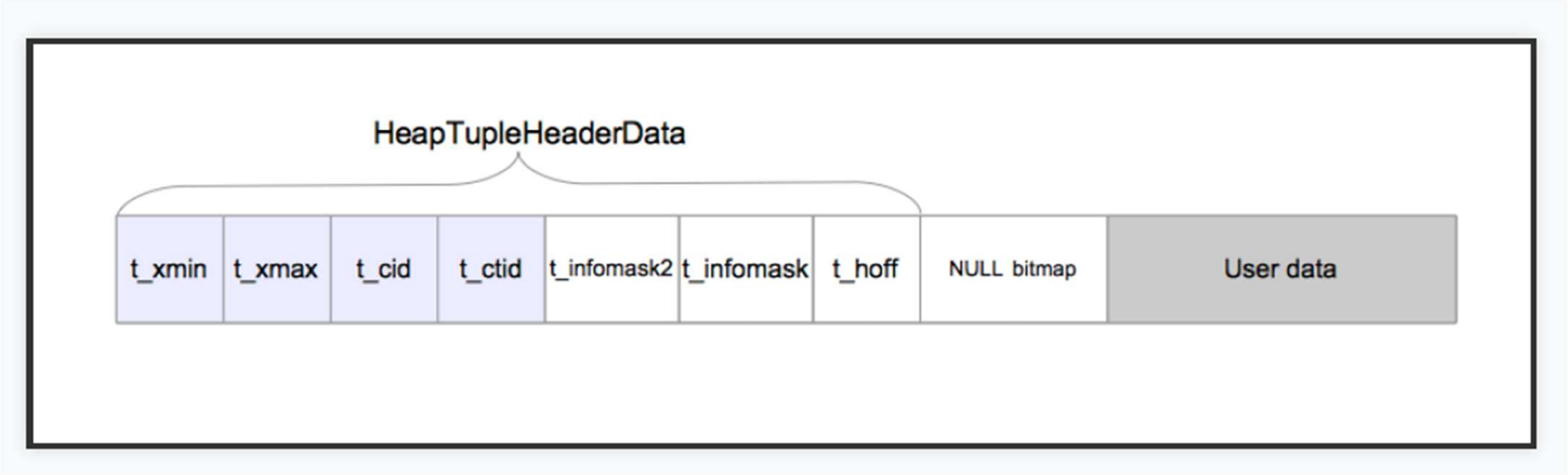
В старой версии записи xmax=txid_current(), то есть делается delete
добавляется новая запись с xmin=txid_current() и xmax=0, то есть делается insert

DELETE

В старой версии записи xmax=txid_current()



MVCC


HeapTupleHeaderData
t_xmin
t_xmax
t_cid
t_ctid
t_infomask2
t_infomask
t_hoff
NULL bitmap
User data

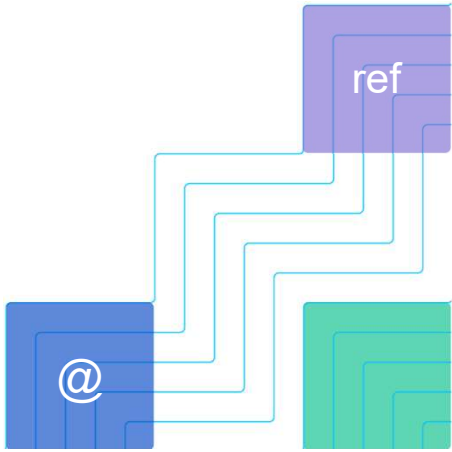
MVCC

Дополнительные атрибуты строки:
infomask содержит ряд битов, определяющих свойства данной версии.
xmin_commited, xmin_aborted
xmax_commited, xmax_aborted
ctid является ссылкой на следующую, более новую, версию той же строки.

У самой новой, актуальной, версии строки ctid ссылается на саму эту версию.

Номера ctid имеют вид (x,y): здесь x — номер страницы, y —порядковый номер указателя в массиве.

Посмотрим на практике.



УРОВНИ ИЗОЛЯЦИИ ТРАНЗАКЦИЙ

Стандарт SQL допускает четыре уровня изоляции, которые определяются в терминах аномалий, которые допускаются при конкурентном выполнении транзакций на этом уровне:

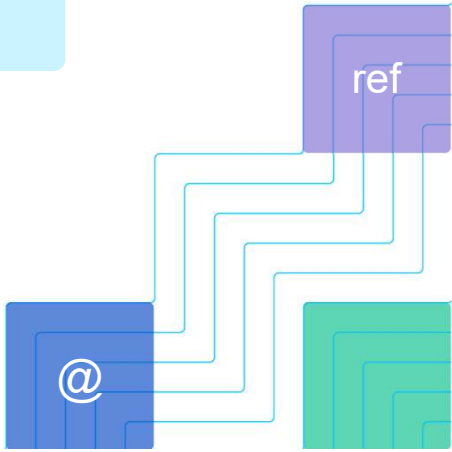
«Грязное» чтение (dirty read). Транзакция T1 может читать строки измененные, но еще не зафиксированные, транзакцией T2. Отмена изменений (ROLLBACK) в T2 приведет к тому, что T1 прочитает данные, которых никогда не существовало.

Неповторяющееся чтение (non-repeatable read). После того, как транзакция T1 прочитала строку, транзакция T2 изменила или удалила эту строку и зафиксировала изменения (COMMIT). При повторном чтении этой же строки транзакция T1 видит, что строка изменена или удалена.

УРОВНИ ИЗОЛЯЦИИ ТРАНЗАКЦИЙ

Фантомное чтение (phantom read). Транзакция T1 прочитала набор строк по некоторому условию. Затем транзакция T2 добавила строки, также удовлетворяющие этому условию. Если транзакция T1 повторит запрос, она получит другую выборку строк.

Аномалия сериализации — результат успешной фиксации группы транзакций оказывается несогласованным при всевозможных вариантах исполнения этих транзакций по очереди.



УРОВНИ ИЗОЛЯЦИИ ТРАНЗАКЦИЙ

	«грязное» чтение	неповторяю-щееся чтение	фантомное чтение	аномалия сериализации
Read Uncommitted	Допускается, но не в PG	да	да	да
Read Committed	-	да	да	да
Repeatable Read	-	-	Допускается, но не в PG	да
Serializable	-	-	-	-

на всех уровнях не допускается потеря зафиксированных изменений

https://habr.com/ru/company/otus/blog/501294/
https://postgrespro.ru/docs/postgrespro/13/transaction-iso

ИТОГИ ЗАНЯТИЯ

01

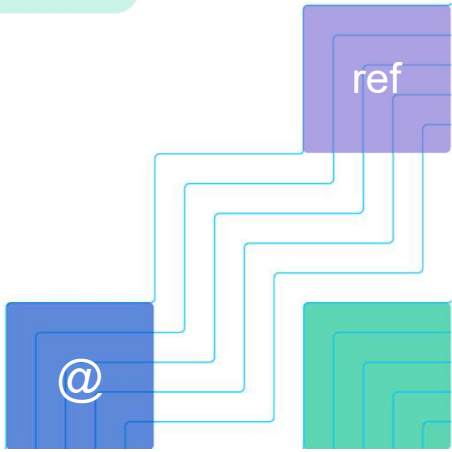
Поняли различие разных уровней изоляции транзакций

02

Поняли работу системы многоверсионности MVCC

03

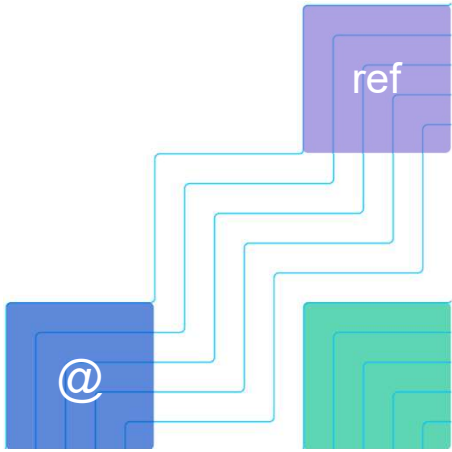
Узнали как подключиться и проверить как настроены уровни изоляции транзакций



ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ

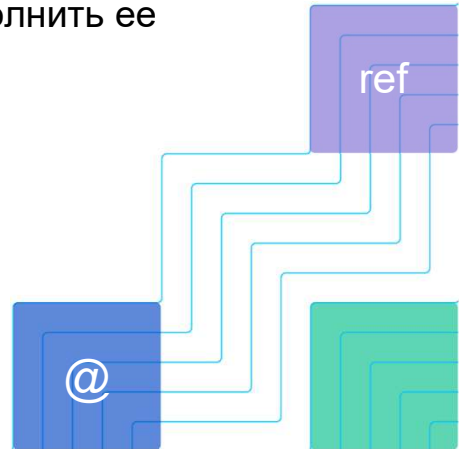
Цель задания:
проверить на практике работу с уровнями изоляции транзакций и увидеть разницу в их работе с данными из разных пользовательских сессий.

Постарайтесь выполнить задание самостоятельно, если какой-то пункт не получится, в материалах к уроку приложена полная инструкция по выполнению работы.



ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ. ПОШАГОВАЯ ИНСТРУКЦИЯ (Ч. 1)

1. Открыть консоль из под пользователя Линукс postgres (если мы открыли новую консоль под student выполнить sudo su postgres)
2. Открыть вторую консоль из под пользователя Линукс postgres зажав кнопку shift и нажать на иконку консоли (выполнить sudo su postgres)
3. Запустить везде psql из под пользователя postgres
4. Сделать в первой сессии новую таблицу и наполнить ее данными
5. Посмотреть текущий уровень изоляции



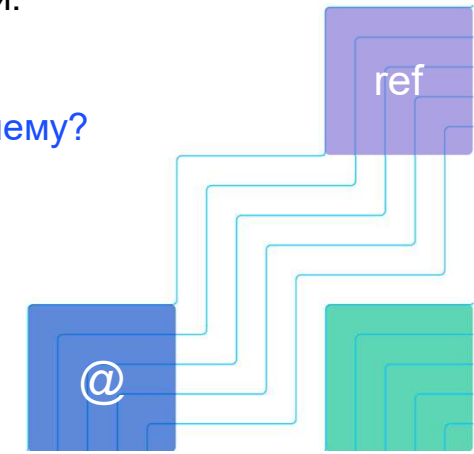
ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ. ПОШАГОВАЯ ИНСТРУКЦИЯ (Ч. 2)

6 Начать новую транзакцию в обоих сессиях с дефолтным (не меняя) уровнем изоляции. В первой сессии добавить новую запись. Сделать запрос на выбор всех записей во второй сессии

? Видите ли вы новую запись и если да то почему? После задания можете сверить правильный ответ с эталонным

7 Завершить транзакцию в первом окне. Сделать запрос на выбор всех записей второй сессии.

? Видите ли вы новую запись и если да то почему?



ЗАДАНИЕ ДЛЯ САМОПРОВЕРКИ. ПОШАГОВАЯ ИНСТРУКЦИЯ (Ч. 3)

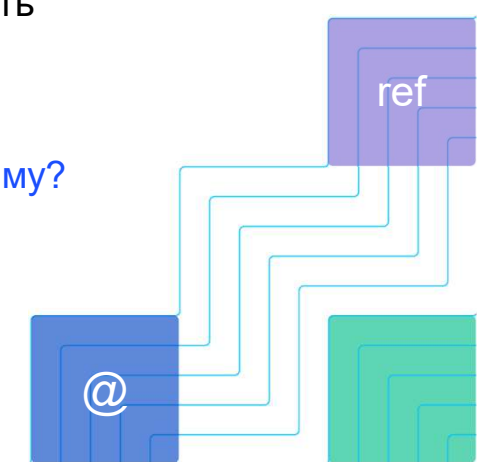
8 Завершите транзакцию во второй сессии. Начать новые транзакции, но уже на уровне repeatable read в ОБЕИХ сессиях

9 В первой сессии добавить новую запись. Сделать запрос на выбор всех записей во второй сессии.

? Видите ли вы новую запись и если да то почему?

10 Завершить транзакцию в первом окне. Сделать запрос во выбор всех записей второй сессии.

? Видите ли вы новую запись и если да то почему?



СПАСИБО

На следующем занятии мы рассмотрим тему:

- Вакуум и автовакуум

{ }

begin