

Базы данных



План занятия

1. Поговорим про типы баз данных
2. Немного о нормальных формах
3. SQL и его компоненты
4. Подробнее про TCL и изоляцию
5. UML



Типы баз данных

Текстовые файлы

- /etc/passwd и /etc/fstab в *nix-системах
- csv-файлы

Иерархические базы данных

- файловые системы
- DNS
- LDAP



Не реляционные (NoSQL)

1) Document-Based

MongoDB, Orient DB, BaseX

2) Key-Value

Redis, Hazelcast, DynamoDB

3) Column-Based

Cassandra, Hbase

4) Graph-Based

Neo4j

Реляционные

- MySQL
- ORACLE
- PostgreSQL
- SQLite
- И много других

Нормальные формы БД

1

1NF

2

2NF

3

3NF

Отношение находится в 1НФ:
1. Все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Отношение находится во 2НФ:
1. Когда находится в 1НФ
2. Каждый не ключевой атрибут неприводимо зависит от Первичного Ключа (ПК).

Отношение находится в 3НФ:
1. Когда находится во 2НФ
2. Каждый не ключевой атрибут нетранзитивно зависит от ПК (Нет зависимости одних неключевых атрибутов от других).

Нормализация – процесс минимизации избыточности отношения (приведение к НФ)

Основной инструмент работы с базами данных

SQL (structured query language) — декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Каждая СУБД включает в себя стандарт языка SQL и дополнительно свои собственные языковые «расширения» (часто не поддерживают стандарт полностью).

Основные типы SQL запросов по их видам

Команды SQL

DDL

Язык определения данных

ALTER

CREATE

DROP

RENAME

ENABLE TRIGGER

DISABLE TRIGGER

DML

Язык манипулирования
данными

SELECT

DELETE

INSERT

UPDATE

DCL

Язык управления
данными

GRANT

REVOKE

DENY

TCL

Язык управления
транзакциями

BEGIN

COMMIT

ROLLBACK

Data Definition Language

Data Definition Language (DDL) – это группа операторов определения данных. Другими словами, с помощью операторов, входящих в эту группы, мы определяем структуру базы данных и работаем с объектами этой базы, т.е. создаем, изменяем и удаляем их.

В эту группу входят следующие операторы:

CREATE – используется для создания объектов базы данных

ALTER – используется для изменения объектов базы данных;

DROP – используется для удаления объектов базы данных.

Data Manipulation Language

Data Manipulation Language (DML) – это группа операторов для манипуляции данными. С помощью этих операторов мы можем добавлять, изменять, удалять и выгружать данные из базы, т.е. манипулировать ими.

В эту группу входят самые распространённые операторы языка SQL:

SELECT – осуществляет выборку данных;

INSERT – добавляет новые данные;

UPDATE – изменяет существующие данные;

DELETE – удаляет данные.

Data Control Language

Data Control Language (DCL) – группа операторов определения доступа к данным. Иными словами, это операторы для управления разрешениями, с помощью них мы можем разрешать или запрещать выполнение определенных операций над объектами базы данных.

Сюда входят:

GRANT – предоставляет пользователю или группе разрешения на определённые операции с объектом;

REVOKE – отзывает выданные разрешения;

DENY– задаёт запрет, имеющий приоритет над разрешением.

Transaction Control Language

Transaction Control Language (TCL) – группа операторов для управления транзакциями. Транзакция – это команда или блок команд (инструкций), которые успешно завершаются как единое целое, при этом в базе данных все внесенные изменения фиксируются на постоянной основе или отменяются, т.е. все изменения, внесенные любой командой, входящей в транзакцию, будут отменены.

Сюда можно отнести:

BEGIN TRANSACTION – служит для определения начала транзакции;

COMMIT TRANSACTION – применяет транзакцию;

ROLLBACK TRANSACTION – откатывает все изменения, сделанные в контексте текущей транзакции;

SAVE TRANSACTION – устанавливает промежуточную точку сохранения внутри транзакции.

ACID:

- Atomicity (Атомарность) – никакая транзакция не будет выполнена частично.
- Consistency (Согласованность) – невозможно произвести неразрешенные операции, нарушить constraints и тд.
- Isolation (Изолированность) – данные в одной транзакции изолированы от других транзакций.
- Durability (Долговечность) – если транзакция прошла, то данные надежно зафиксированы.

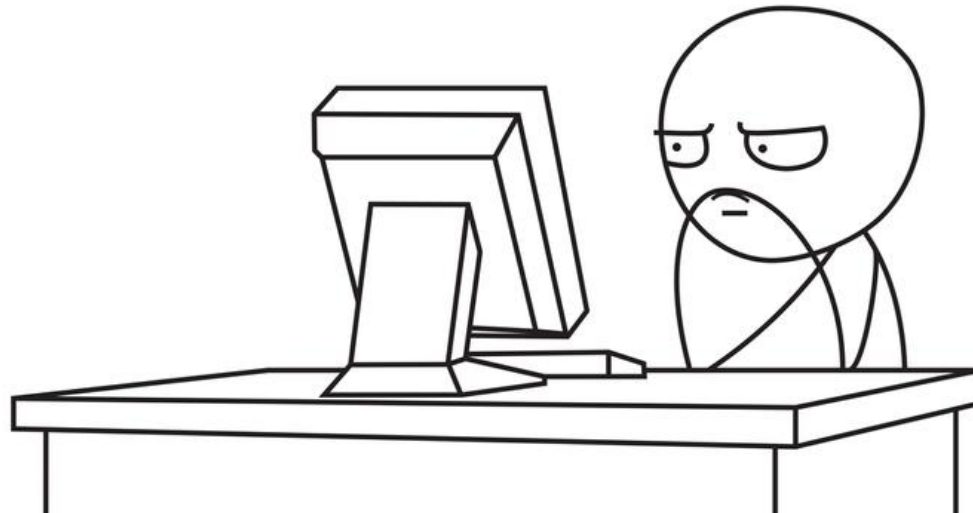
Особенности транзакций в Postgres:

- 1) Абсолютно все запросы в postgres обернуты в транзакции.
- 2) Нельзя создавать транзакции в функциях (хотя неявно она там есть).
- 3) Чтобы прервать транзакцию в функции можно бросить RAISE EXCEPTION.
- 4) Концепция подразумевает что уровнем изоляции и откатами управляет внешний код (SQL-скрипт или код приложения).

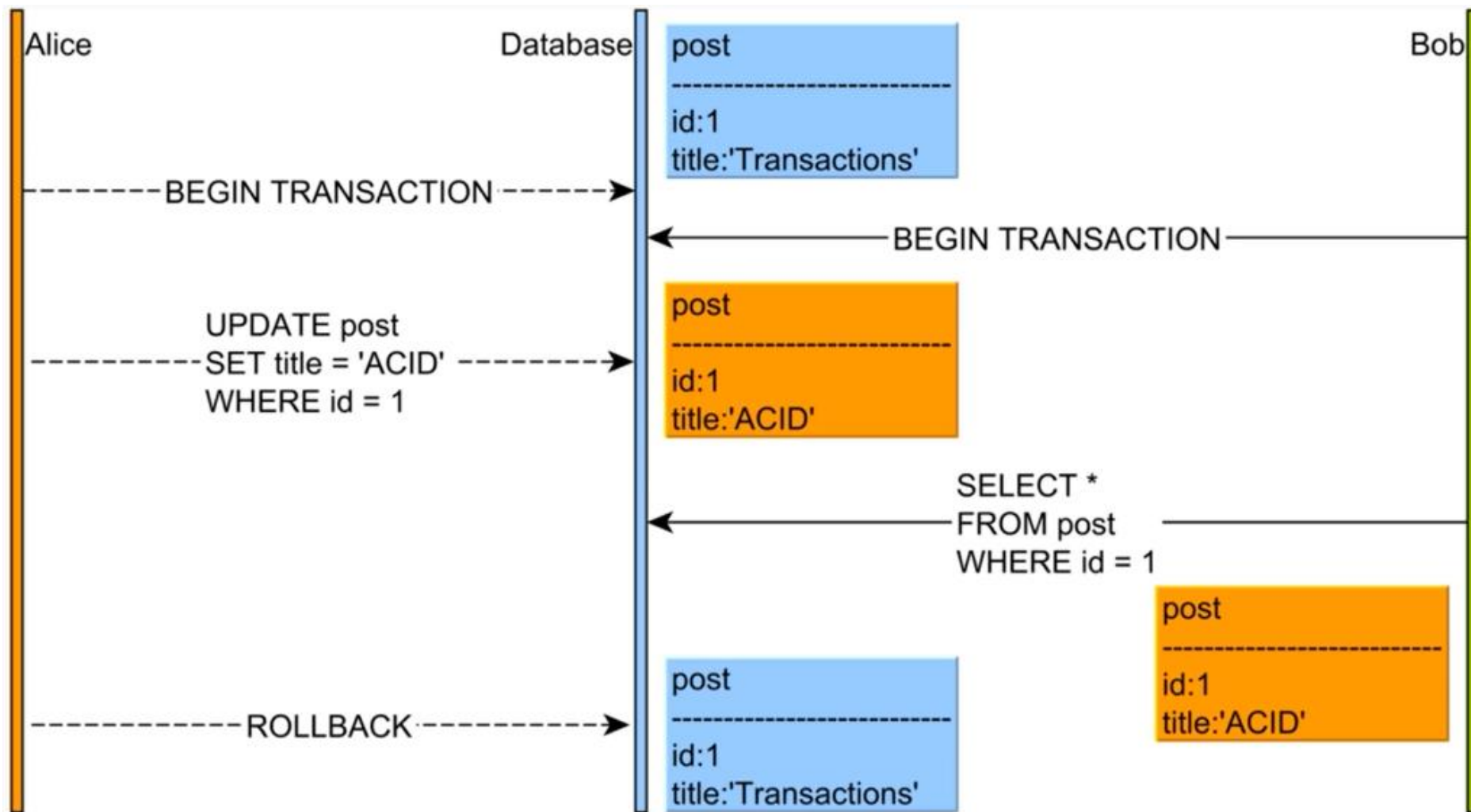
Изоляция транзакций

Потенциальные проблемы параллельности:

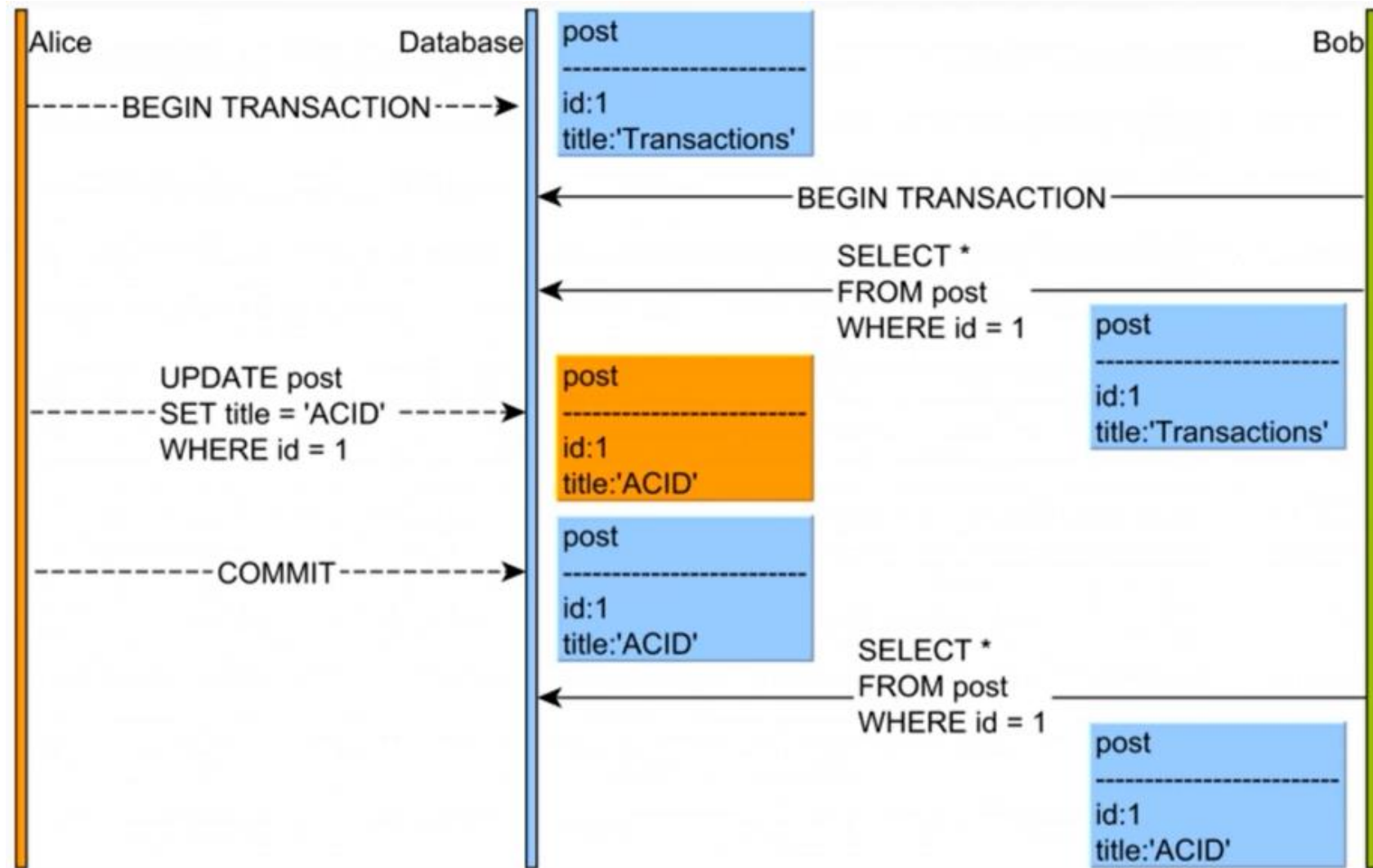
- «Грязное» чтение – чтение частичных изменений.
- Неповторяемое чтение – повторное чтение показывает, что данные были изменены после первого чтения.
- Фантомное чтение – повторное чтение показывает другой результирующий набор.
- Аномалия сериализации – результат параллельно выполняемых транзакций может не согласовываться с результатом этих же транзакций выполняемых по очереди.



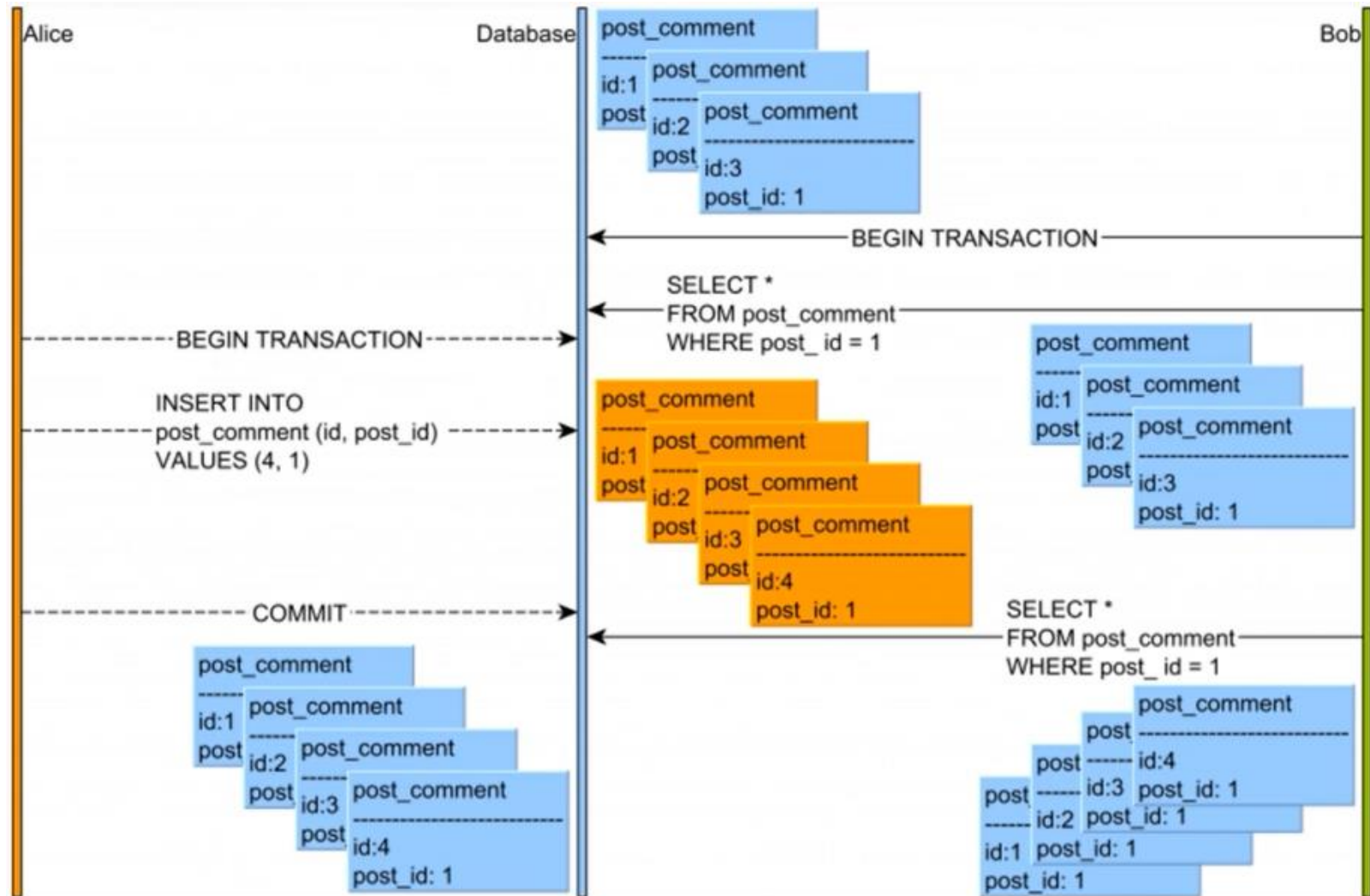
«Грязное» чтение



Неповторяемое чтение



Фантомное чтение



Аномалия сериализации

class	value
1	10
1	20
2	100
2	200

Транзакция А	Транзакция В
<pre>SELECT SUM(value) FROM vals WHERE class = 1; ----- 30</pre>	<pre>SELECT SUM(value) FROM vals WHERE class = 2; ----- 300</pre>
<pre>INSERT INTO mytab VALUES (30, 2)</pre>	<pre>INSERT INTO mytab VALUES (300, 1)</pre>
<pre>COMMIT;</pre>	<pre>COMMIT;</pre>

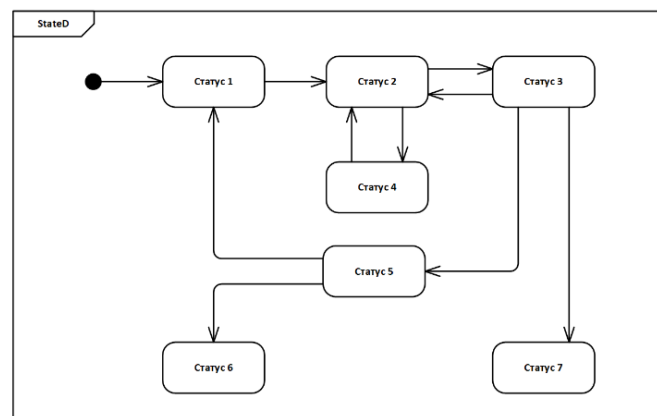
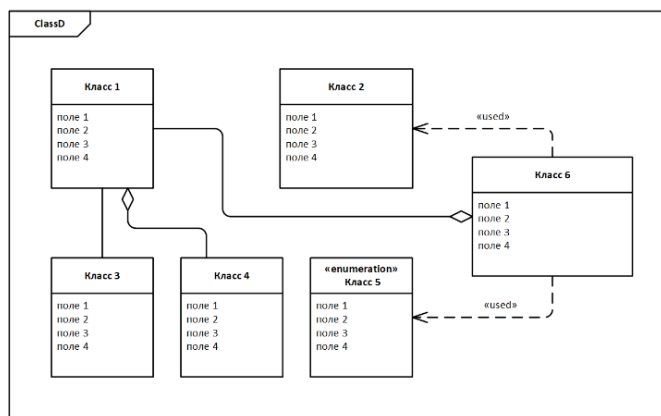
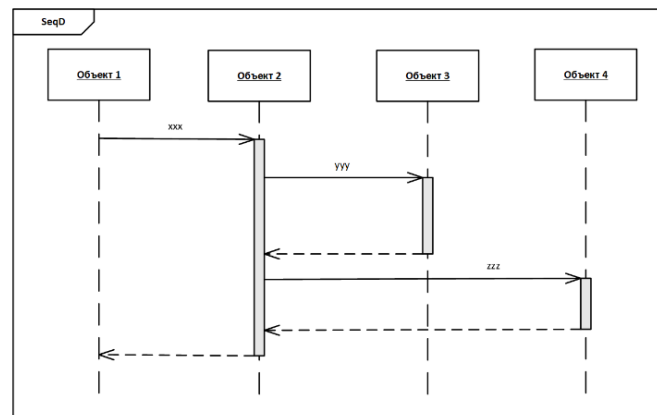
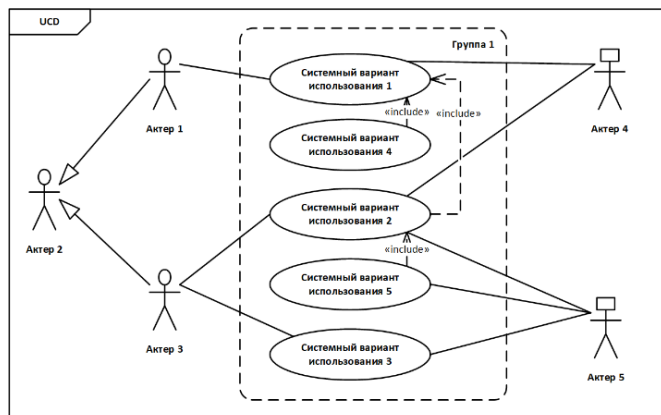
Уровни изоляции транзакций

Уровень изоляции	«Грязное» чтение	Неповторяемое чтение	Фантомное чтение	Аномалия сериализации
Read uncommitted (Чтение незафиксированных данных)	Допускается, но не в PG	Возможно	Возможно	Возможно
Read committed (Чтение зафиксированных данных)	Невозможно	Возможно	Возможно	Возможно
Repeatable read (Повторяемое чтение)	Невозможно	Невозможно	Допускается, но не в PG	Возможно
Serializable (Сериализуемость)	Невозможно	Невозможно	Невозможно	Невозможно

Важный момент: уровни изоляции не исключают само возникновение конфликтов (неповторяемое, фантомное чтение и т.д.) а просто ловят их, генерируют ошибку и откатывают транзакцию которая наткнулась на конфликт

Unified Modeling Language (UML)

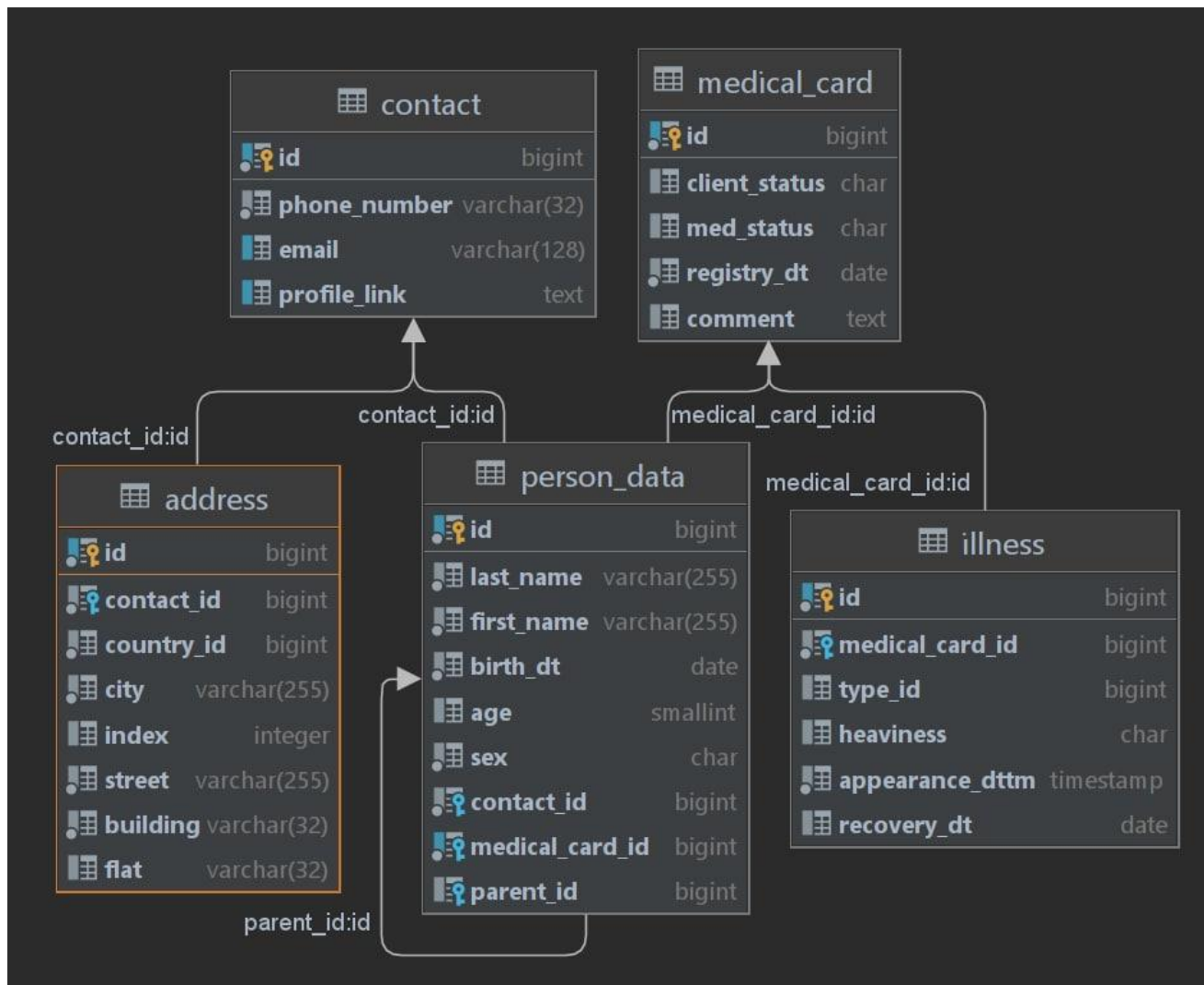
Язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.



Домашнее задание:

1. Повторить структуру БД по предоставленной UML диаграмме.
2. Заполнить тестовыми данными таблицы.
3. Написать запрос, выводящий общие записи из 2 таблиц.
4. Написать запрос, удаляющий повторяющиеся (без учета id) строки с таблице.
5. Написать запрос, выводящий первые 50% записей из таблицы (первые значит те где наименьший id).
6. Написать запрос, выводящий список ФЛ с их родителями, у которых мед. статус отсутствует.
7. Создать представление из нескольких таблиц.
8. Создать хранимую процедуру для извлечения данных из таблицы.
9. Создать хранимую процедуру, которая будет модифицировать таблицу.
10. Создать триггер, реагирующий на добавление данных.

Структура базы данных для домашнего задания



Полезная литература

Упражнения sql

- <https://sql-ex.ru/>

Нормализация БД (1NF, 2NF, 3NF)

- <https://info-comp.ru/database-normalization>

Основные команды

- <https://tproger.ru/translations/sql-recap/>

Полная документация postgresSQL

- <https://postgrespro.ru/docs/>

UML

- <https://evergreens.com.ua/ru/articles/uml-diagrams.html>



Всем спасибо!

Лига –
лучший
старт
карьеры!

Мы в Лиге!

Умножай
знания –
верь в мечту!

Каждый
день – новый
челлендж!



Владислав Сыров

Разработчик



АКАДЕМИЯ