

Chapter-0: Terminology Database

Upcode Software
Engineer Team

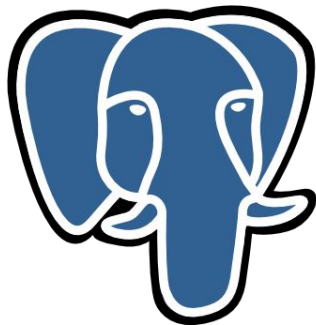


CONTENT

1. **Postgresql**
2. **ACID**
3. **MVCC**
4. **Orm**
5. **Jpa**
6. **Hibernate**

Что такое PostgreSQL?

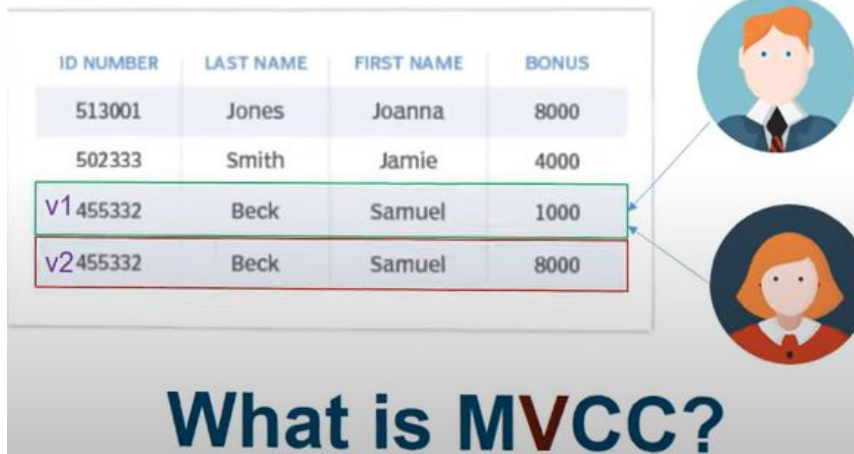
- PostgreSQL (также известная как Postgres) - это бесплатная и открытая система управления реляционными базами данных (СУБД), которая акцентирует внимание на расширяемости и соответствии стандарту SQL.
- PostgreSQL поддерживает транзакции с атомарностью, согласованностью, изолированностью и долговечностью (ACID).



PostgreSQL

Что такое PostgreSQL?

PostgreSQL — это очень популярная объектно-реляционная база данных с открытым исходным кодом, использующая уникальный подход к управлению параллелизмом под названием Multi-Version Concurrency Control (MVCC).





MVCC

- Multi-Version Concurrency Control — это метод управления параллелизмом, используемый в системах управления реляционными базами данных (RDBM). Это позволяет нескольким транзакциям одновременно получать доступ к одним и тем же данным, сохраняя при этом **согласованность и изоляцию** данных.
- Мы **сканируем каждый блок таблицы** из-за важной особенности Postgres: называется Multiversion Concurrency Control (MVCC).



Atomicity, Consistency, Isolation and Durability

Свойства ACID

Atomicity

Каждая транзакция рассматривается как отдельная единица, которая либо завершается полным успехом, либо полностью завершается неудачей.

Consistency

Транзакции вносят изменения в таблицы только заранее определенным и предсказуемым образом.

Isolation

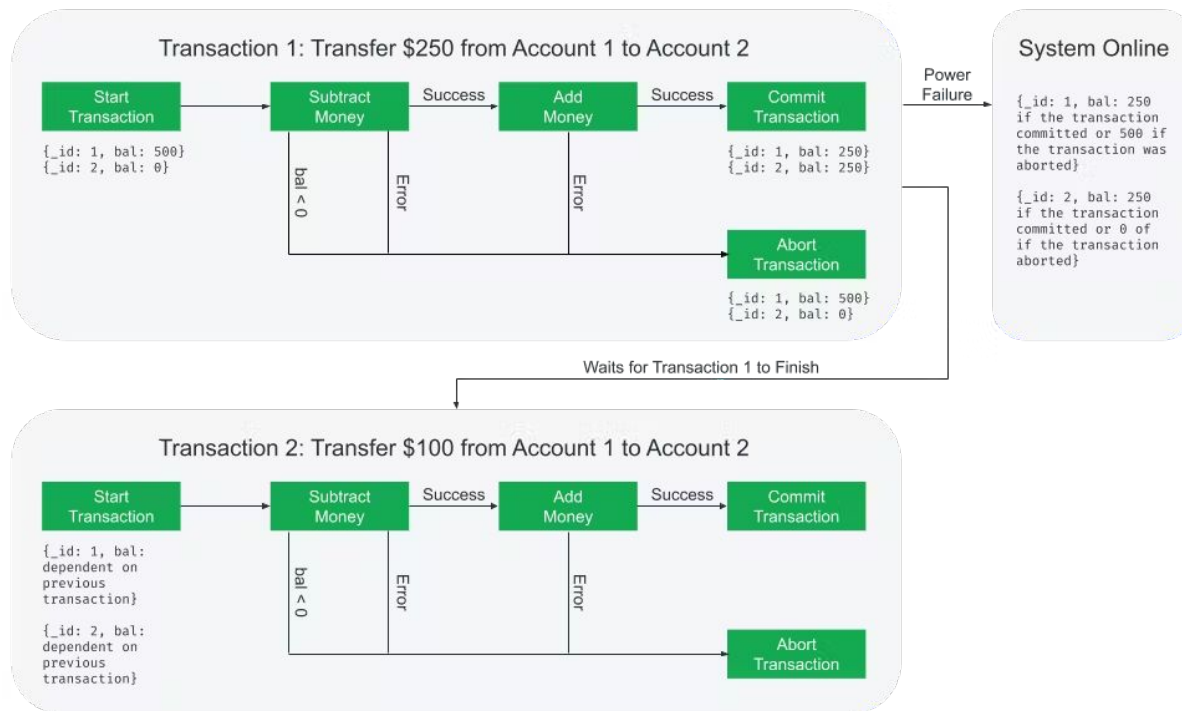
Параллельные транзакции не взаимодействуют и не влияют друг на друга.

Durability

После того как транзакция была зафиксирована, она останется зафиксированной даже в случае сбоя системы.

Atomicity, Consistency, Isolation and Durability

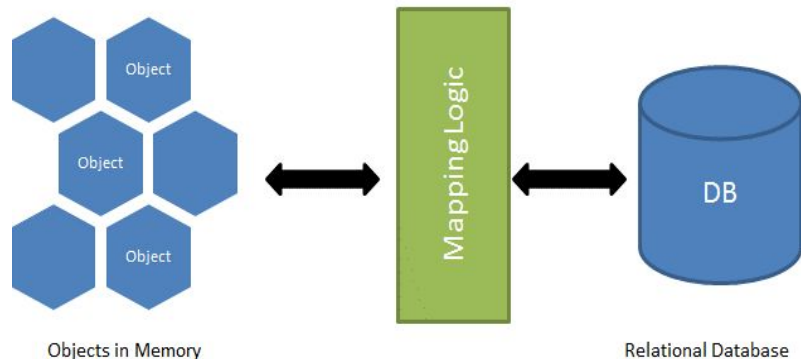
Диаграмма демонстрирует, как свойства ACID влияют на поток перевода денег с одного банковского счета на другой.



Object Relational Mapping

- ORM (Object-Relational Mapping) - это технология, которая позволяет разработчикам работать с объектами в объектно-ориентированном стиле, в то время как данные хранятся в реляционных базах данных.
- ORM устанавливает соответствие (отображение) между объектами и таблицами базы данных, что позволяет автоматически сохранять, извлекать, обновлять и удалять данные без явного написания SQL-запросов.

O/R Mapping





Object Relational Mapping

Транзакция

- **Транзакция** - это логическая единица работы, которая состоит из одного или нескольких операций базы данных. Она должна быть выполнена целиком или не выполнена вообще.



Object Relational Mapping

Кэширование

- **Кэширование** - это механизм, который используется для временного хранения данных в памяти, чтобы улучшить производительность доступа к данным.
- В контексте ORM, кэширование позволяет **уменьшить количество запросов к базе данных**, сохраняя часто используемые данные в памяти.



Object Relational Mapping

Отложенная выборка

- Стратегия отложенной выборки (lazy loading) - это подход, используемый в ORM (Object-Relational Mapping), который позволяет откладывать загрузку связанных данных до момента, когда они действительно понадобятся.



Object Relational Mapping

Пакетная обработка

Пакетная обработка (Batch Processing) в ORM - это подход, который позволяет выполнять операции сразу над несколькими объектами или записями в базе данных, пакетная обработка позволяет группировать операции и выполнять их одновременно.



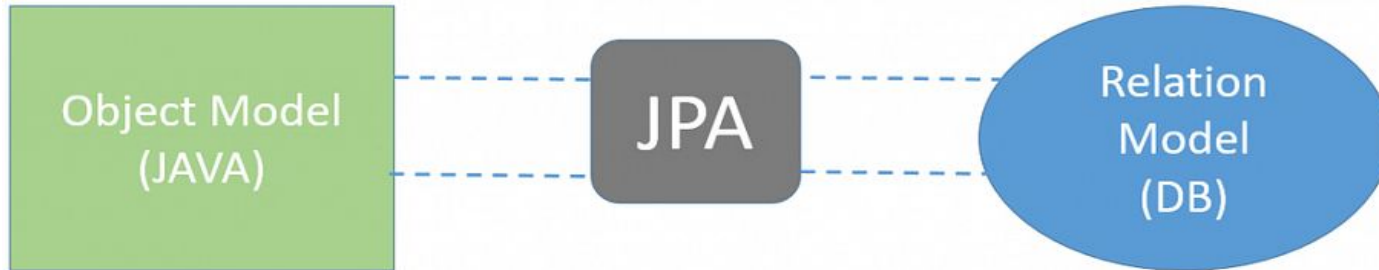
Object Relational Mapping

Пул соединений

- **Пул соединений** - это пул предварительно созданных и готовых к использованию соединений с базой данных.
- Вместо того, чтобы каждый раз устанавливать новое соединение при выполнении операций с базой данных, механизмы объединения пулов соединений позволяют переиспользовать уже установленные соединения из пула.

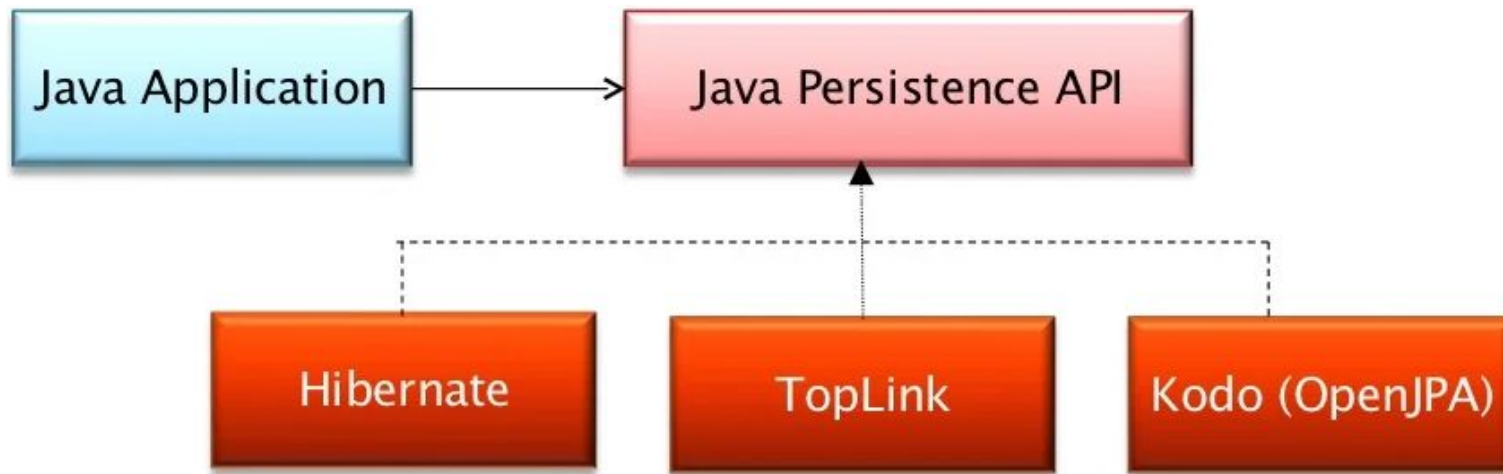
Java Persistence API

- JPA (Java Persistence API) - это спецификация Java EE, которая определяет стандартный способ работы с объектами в Java-приложениях и их сохранения в реляционных базах данных.



Java Persistence API

- JPA предоставляет набор аннотаций и интерфейсов для работы с объектно-реляционным отображением (ORM).





JPA основные интерфейсы(1/2)

- **EntityManager**: является основным интерфейсом для выполнения операций по сохранению, чтению, обновлению и удалению сущностей в базе данных.
- **EntityTransaction**: представляет транзакцию базы данных и предоставляет методы для управления транзакционными операциями, такими как `begin()`, `commit()`, `rollback()`.
- **Query**: позволяет создавать и выполнять запросы к базе данных. Он предоставляет методы для создания запросов на основе языка запросов, таких как JPQL (Java Persistence Query Language) или Criteria API.



JPRA основные интерфейсы(2/2)

- **TypedQuery**: является расширением интерфейса Query и предоставляет типизированные методы для работы с запросами. Он позволяет указать ожидаемый тип результата запроса и предоставляет более типобезопасные операции.
- **CriteriaQuery**: представляет типобезопасный способ создания запросов с использованием Criteria API. Он позволяет строить запросы в программном коде с помощью объектов-критериев, представляющих различные части запроса, такие как выборка, условия и сортировка.
- **CriteriaBuilder**: предоставляет методы для создания объектов-критериев (CriteriaQuery, Predicate, Order и других) и выполнения операций сравнения, логических операций и математических операций в запросах.



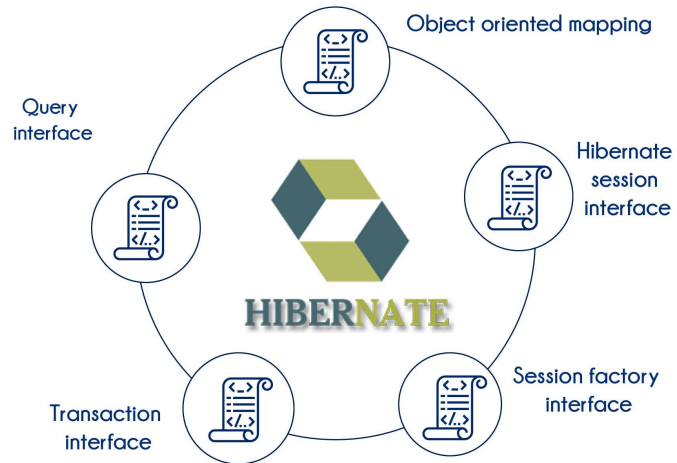
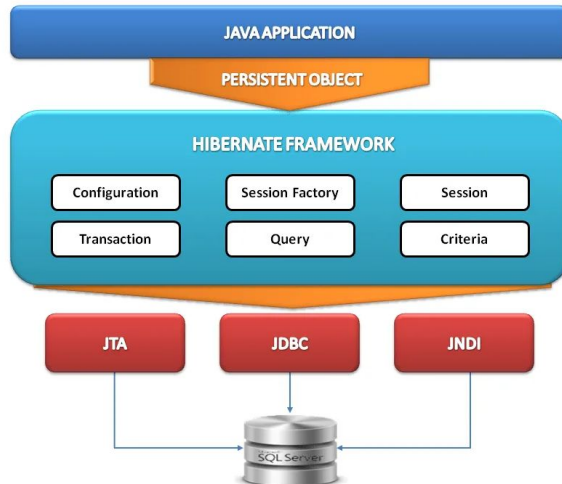
Java Persistence Language

- **Java Persistence Query Language (JPQL)** - это язык запросов, который позволяет разработчикам выполнять операции с сущностями в базе данных, используя объектно-ориентированный подход. JPQL предоставляет удобный способ взаимодействия с базой данных в Java Persistence API (JPA).

```
String hqlQuery = "FROM Product p WHERE p.price > :price" ;  
List<Product> products = session.createQuery(hqlQuery)  
    .setParameter("price", 100)  
    .getResultList();
```

Hibernate

- **Hibernate** - Этот фреймворк отвечает за решение проблем несоответствия объектно-реляционного импеданса.
- В Java есть спецификация под названием Java Persistence API (JPA), которая описывает управление объектами в реляционной базе данных.
- **Hibernate** — это всего лишь реализация JPA.



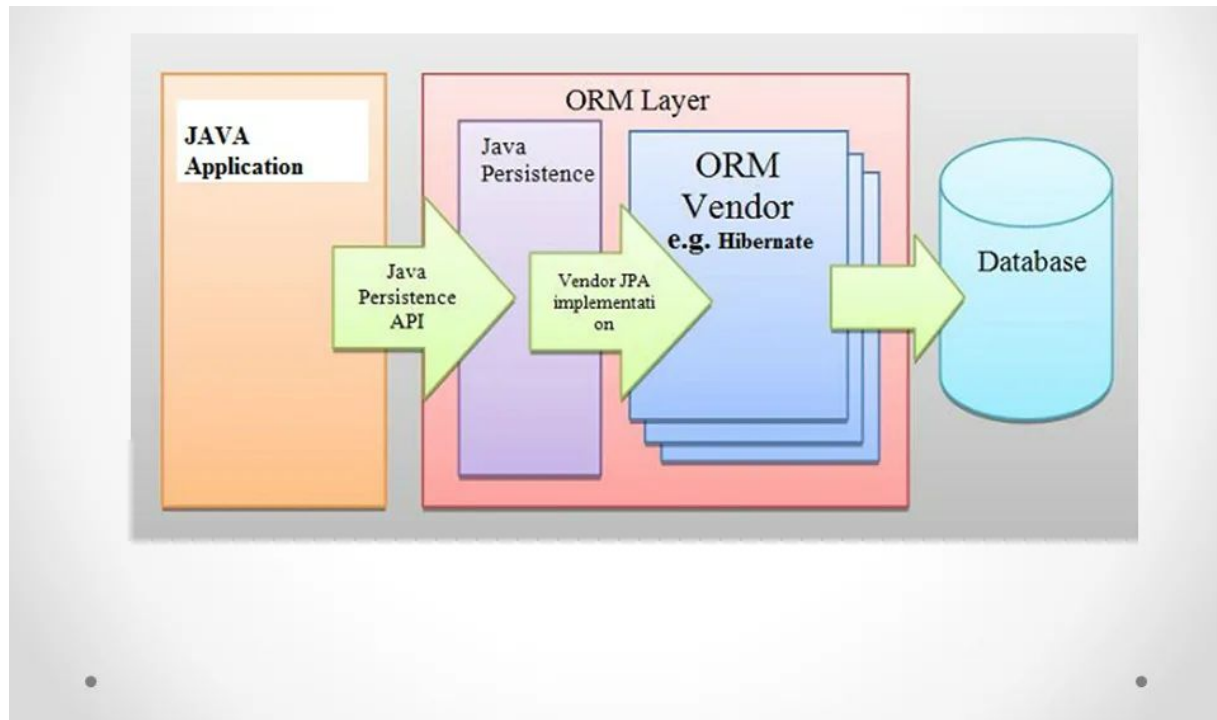


Hibernate Query Language

- **Hibernate Query Language (HQL)** - это язык запросов, используемый в фреймворке Hibernate. HQL является объектно-ориентированным языком запросов, похожим на SQL, HQL работает с постоянными объектами и их свойствами.
- Запросы на HQL транслируются Hibernate в обычные SQL-запросы, которые затем выполняются в базе данных

```
String hql = "FROM Employee E WHERE E.id > 10 " +  
            "ORDER BY E.firstName DESC, E.salary DESC " ;  
Query query = session.createQuery(hql);  
List results = query.list();
```

JPA и Hibernate





Ссылки

- <https://docs.oracle.com/javaee/7/api/javax/persistence/package-summary.html>
- <https://medium.com/@sendvjs/spring-data-jpa-af0f0c0c78a2>
- <https://practicum.yandex.ru/blog/cto-takoe-subd-postgresql/>
- [PostgreSQL MVCC: Общие сведения об управлении многоверсионным параллелизмом в PostgreSQL | Сатиш Мишра | Терпимая \(medium.com\)](#)
- [MVCC](#)