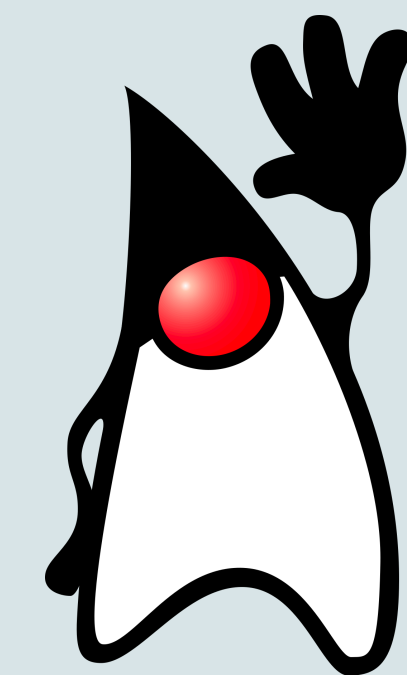




# СБЕРБАНК

---

Корпоративный  
университет



# Базы данных. Продолжение. JRQL

Занятие №11

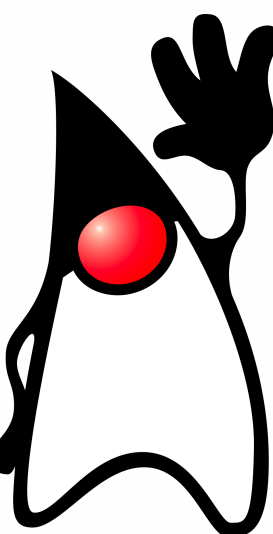


**СБЕРБАНК**

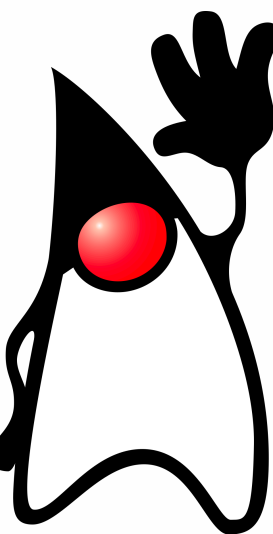
Корпоративный  
университет



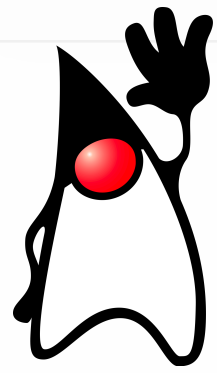
- Чуть-чуть опять про maven (по материалам вопросов)
- Основные команды чистого SQL
- Connection Pool
- Ещё раз Hibernate
- Hibernate JPQL



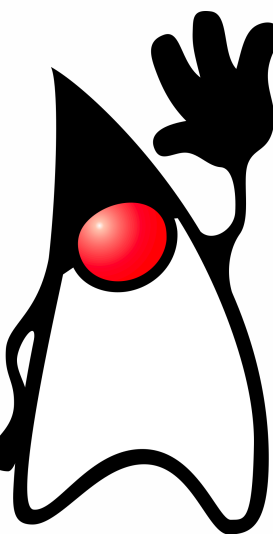
- Активно участвуем. Не стесняйтесь задавать вопрос.
- Но off-topic обсуждаем в Telegram @sb\_ku\_java\_2019\_10
- Не стесняйтесь просто спрашивать в telegram.
- ДЗ - работаем над библиотекой



**Договорились?  
Поехали!**



- **Чуть-чуть опять про maven (по материалам вопросов)**
- Основные команды чистого SQL
- Connection Pool
- Ещё раз Hibernate
- Hibernate JPQL

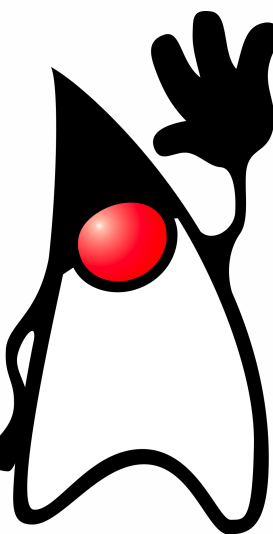


01

# Мaven. Разбор проблем и решений

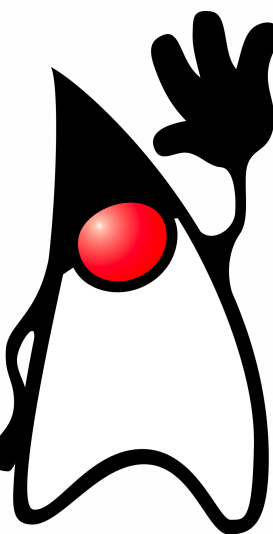
## IDEA или maven не хотят работать.

- IDEA показывает ошибки непонятные? Не находит либы?
  - Maven почему-то не собирает и непонятно почему?
1. Читаем логи!!!! Внимательно читаем
  2. Если не помогло - вызываем детальные логи (параметр -X)  
**`mvn clean package -X`**
  3. Внимательно читаем расширенные логи





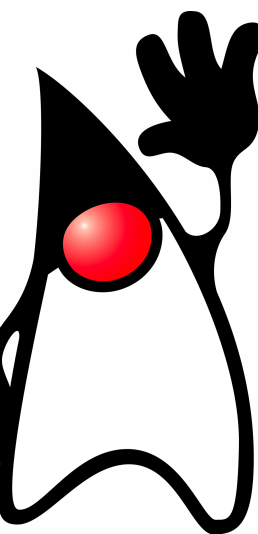
- Ошибки в pom. Проверяем на дублирование или отсутствие обязательных ТЭГОВ
- Возможно была попытка загрузить зависимость, но с ошибкой. Повторная попытка только после timeout. Чистим .m2
- Родитель есть или корневой проект?
- Не подключен корректный репозиторий



# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

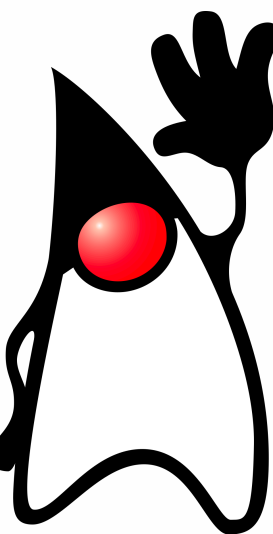
- Чуть-чуть опять про maven (по материалам вопросов)
- **Основные команды чистого SQL**
- Connection Pool
- Ещё раз Hibernate
- Hibernate JPQL



02

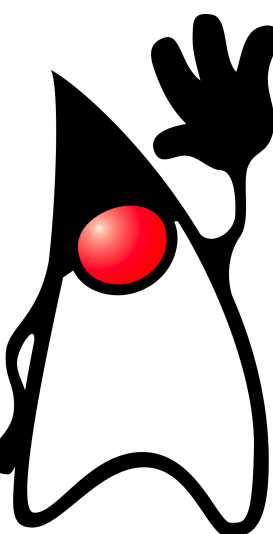
# **Основные команды манипуляции данными чистого SQL**

```
insert into <название таблицы> ( [<Имя столбца>, ... ] ) values  
( <Значение>, ... )
```

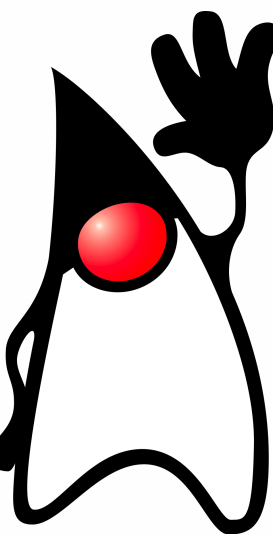


## SELECT

```
[DISTINCT | DISTINCTROW | ALL]  
select_expression, ...  
FROM table_references  
[WHERE where_definition]  
[GROUP BY {unsigned_integer | col_name | formula}]  
[HAVING where_definition]  
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
```

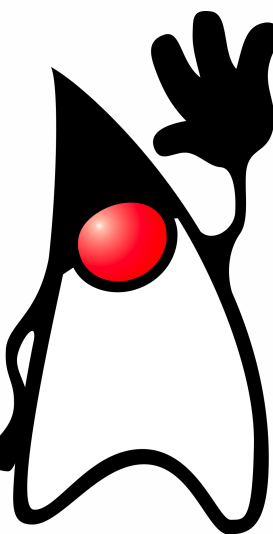


```
UPDATE [top(x)] <объект>  
SET <присваивание1 [ , присваивание2, ... ]>  
[ WHERE <условие> ];
```



# DELETE

```
DELETE FROM <Имя Таблицы> WHERE <Условие отбора записей>
```





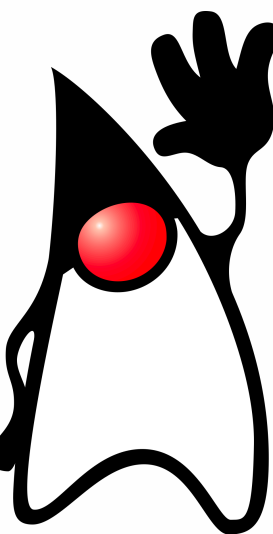
**FROM**

Table1

{**INNER** | {**LEFT** | **RIGHT** | **FULL**} **OUTER** | **CROSS** } **JOIN**

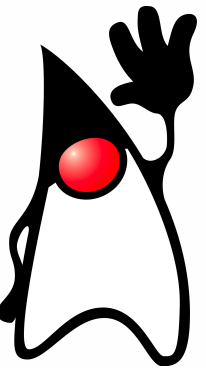
Table2

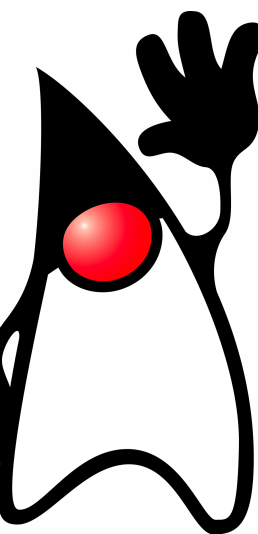
{**ON** <condition> | **USING** (field\_name [, ... n])}



# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

- Чуть-чуть опять про maven (по материалам вопросов)
- Основные команды чистого SQL
-  • **Connection Pool**
- Ещё раз Hibernate
- Hibernate JPQL



03

# Connection Pool

Connection Pool реализует паттерн Object Pool.  
Ускоряет и автоматизирует работу с базой данных.

Популярные connection pool:

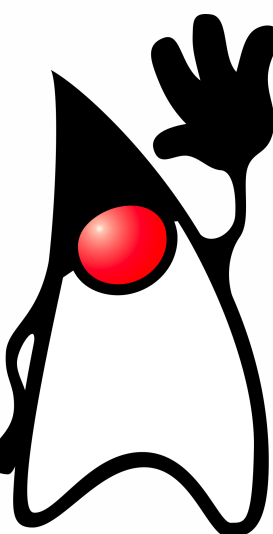
- c3p0

- Oracle UCP

- Tomcat build-in connection Pool

- HikariCP

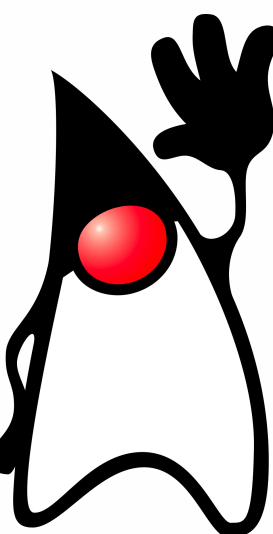
- Apache DBCP



[Сайт пула.](#)

Fast, simple, reliable. HikariCP is a "zero-overhead" production ready JDBC connection pool. At roughly 130Kb, the library is very light.

[Обсуждение на Хабре](#) (см. комментарии Паньгина)



## Параметры Essentials:

dataSourceClassName

This is the name of the DataSource class provided by the JDBC driver.

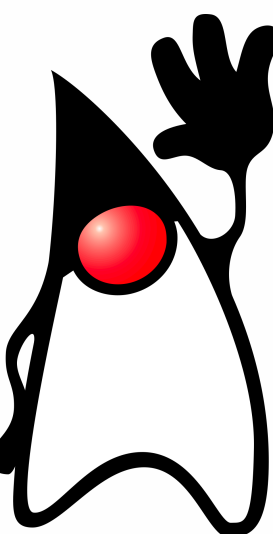
или

jdbcUrl

This property directs HikariCP to use "DriverManager-based" configuration.

username

password



## Параметры Frequently used:

### connectionTimeout

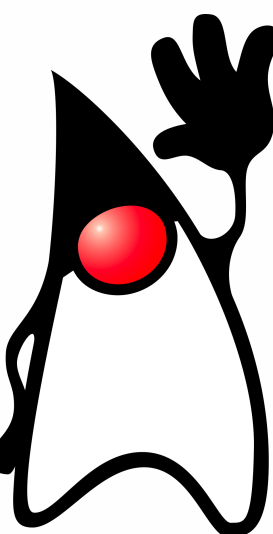
This property controls the maximum number of milliseconds that a client (that's you) will wait for a connection from the pool.

### idleTimeout

This property controls the maximum amount of time that a connection is allowed to sit idle in the pool.

### maxLifetime

This property controls the maximum lifetime of a connection in the pool.





Параметры Frequently used:

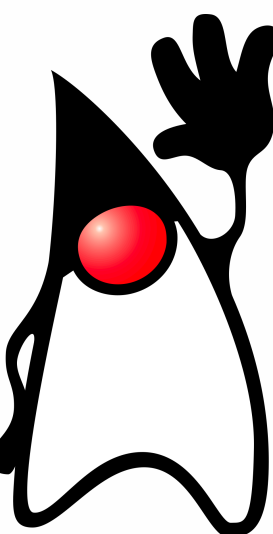
autoCommit

minimumIdle

This property controls the minimum number of idle connections that HikariCP tries to maintain in the pool. .

maximumPoolSize

This property controls the maximum size that the pool is allowed to reach, including both idle and in-use connections..



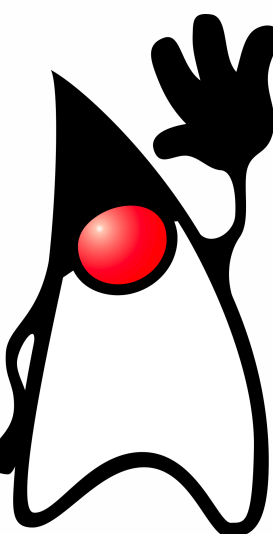
Параметры Frequently used:

`poolName`

This property represents a user-defined name for the connection pool.

`registerMbeans`

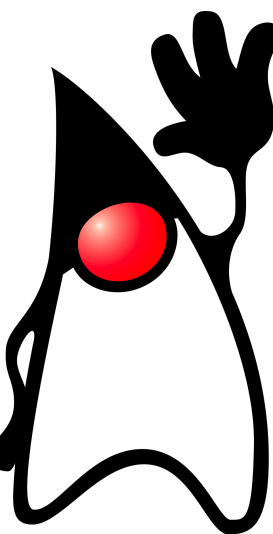
This property controls whether or not JMX Management Beans ("MBeans") are registered or not. Default: false.



Мониторинг.

JMX-bean (jconsole)

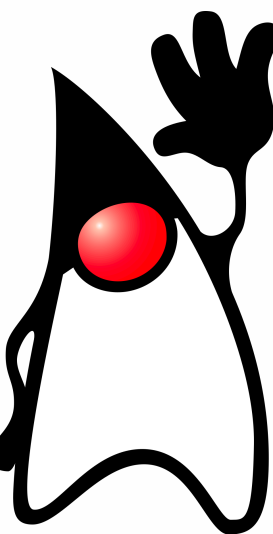
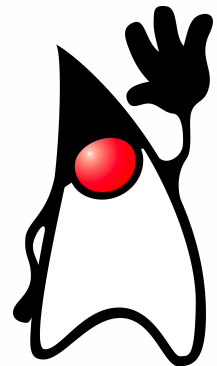
Встроенное логирование (lsf4j)



# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

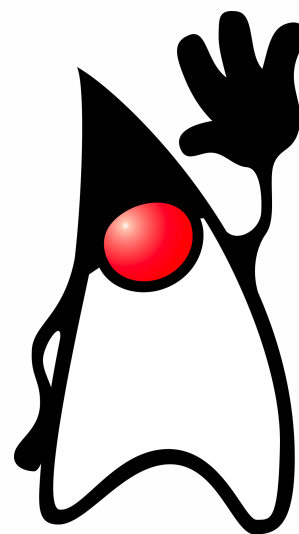
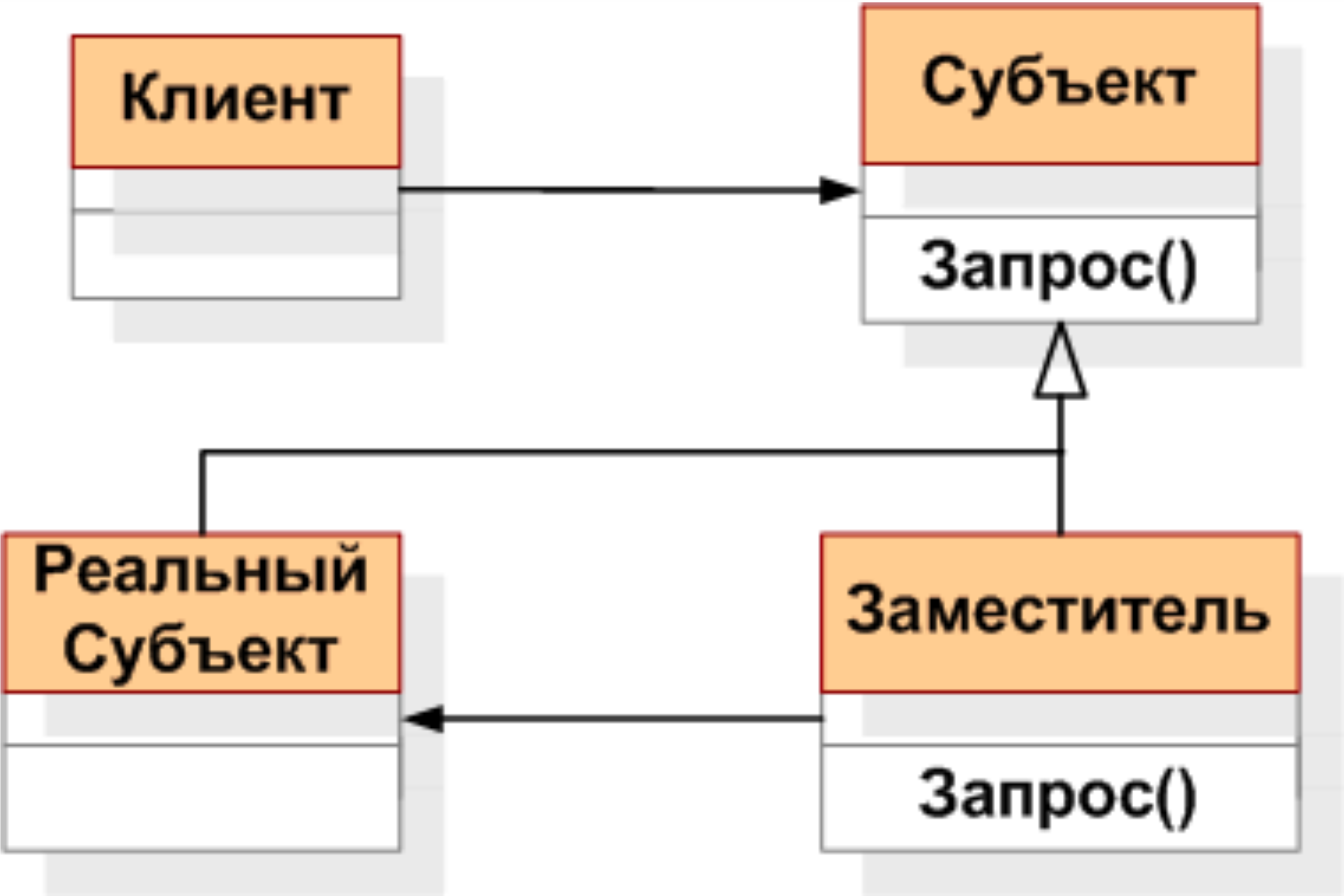
- Чуть-чуть опять про maven (по материалам вопросов)
- Основные команды чистого SQL
- Connection Pool
- **Ещё раз Hibernate**
- Hibernate JPQL



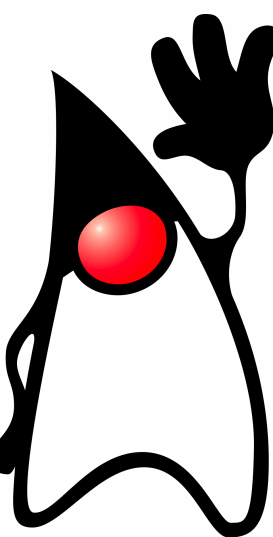
04

**Hibernate. Освежим**

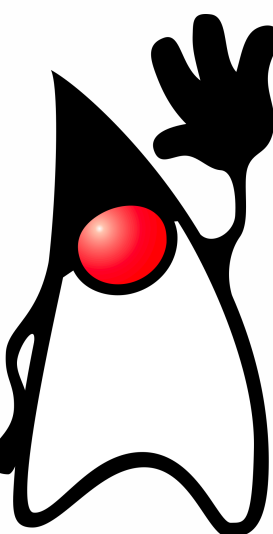
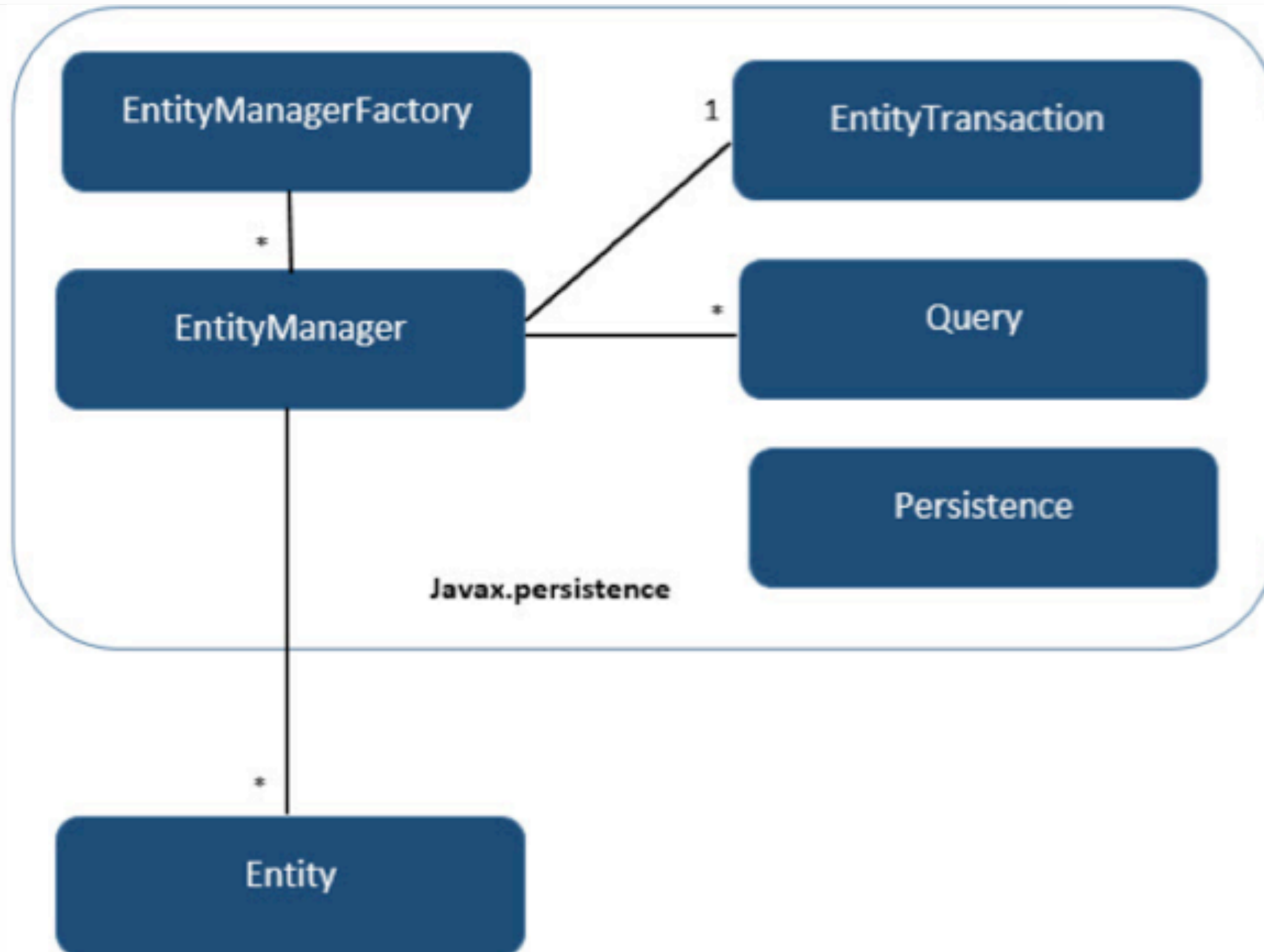
# Шаблон Заместитель (Proxy)

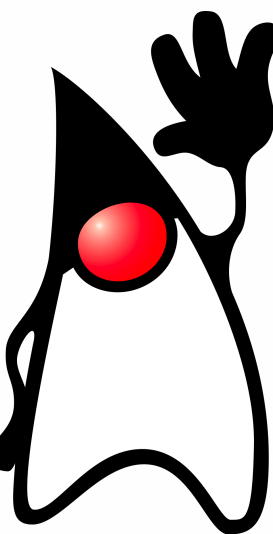
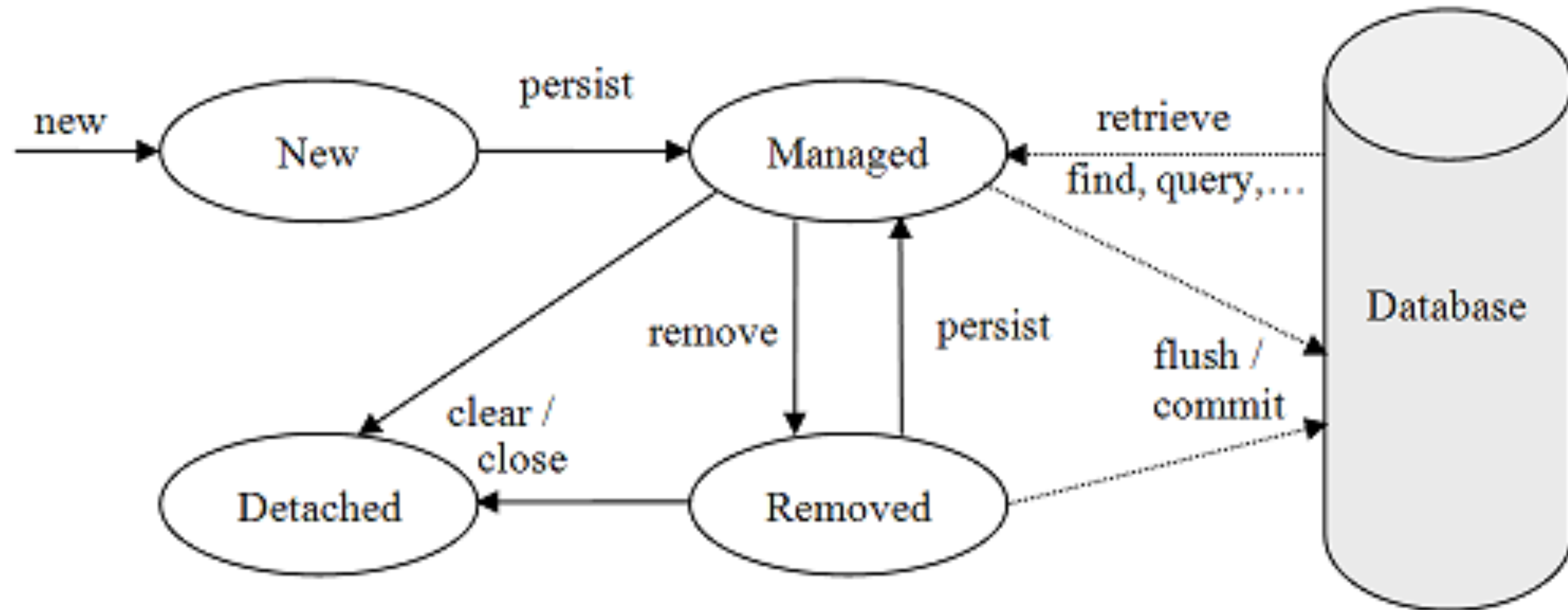


- Сильно тормозит приложение
- Решает все проблемы с базой, можно не учить SQL
- Hibernate - это просто
- Hibernate - это сложно

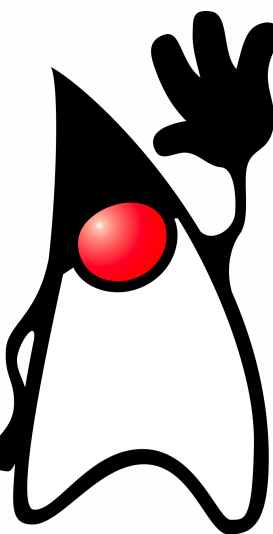








- Persistence Context – кэш первого уровня для объектов, которые сохраняются или загружаются в/из БД в рамках сессии
- В применении к данному контексту сущность может находиться в трех состояниях
- Transient – объект еще ни разу не был присоединен к контексту
- Persistent – объект находится под управлением контекста. Все изменения в нем попадут в БД
- Detached – объект был когда-то прикреплен к контексту, но теперь это не так. Объект переходит в это состояние, если исключить его из контекста, очистить или закрыть сессию



@OneToMany

Две стратегии загрузки дочерних сущностей.

Fetch = FetchType.EAGER – селектим все и сразу

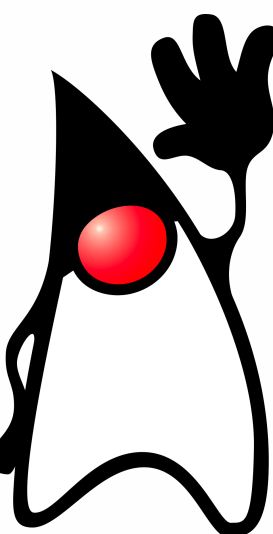
Fetch = FetchType.LAZY - селектим по мере необходимости.

По умолчанию: FetchType.EAGER

Пример: fetchExample()

**Почему это важно?**

В тестах работает, на проде адские тормоза и OOM.



@OneToMany

Внимательно настраивайте связи:

@OneToMany(mappedBy="field")

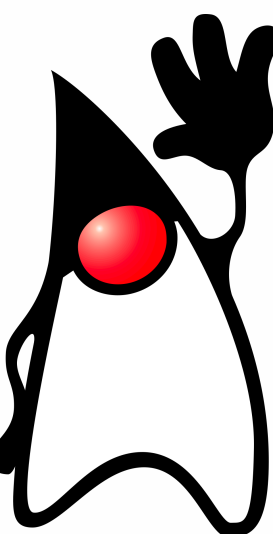
@ManyToOne

@JoinColumn(name = "feild", nullable = false)

Пример: fetchExample()

**Почему это важно?**

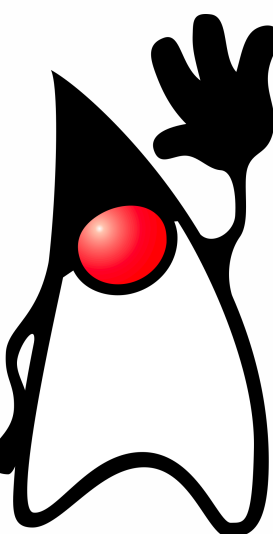
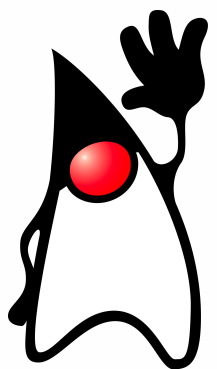
Неверные аннотации приведут к паразитным update-ам и ненужным outer join-ам.



# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

- Чуть-чуть опять про maven (по материалам вопросов)
- Основные команды чистого SQL
- Connection Pool
- Ещё раз Hibernate
- **Hibernate JPQL**



05

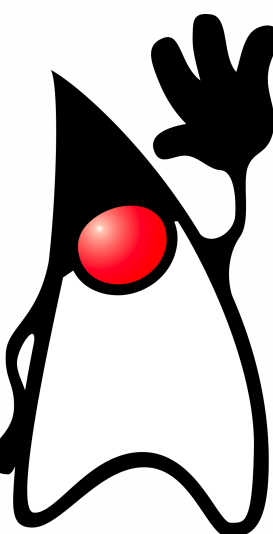
# Hibernate JPQL



Java Persistence Query Language (JPQL)  
“Объектно-ориентированный SQL”

По синтаксису похож на SQL.  
Вместо имен таблиц – классы, вместо колонок – поля классов.

Пример: JPQLexample()



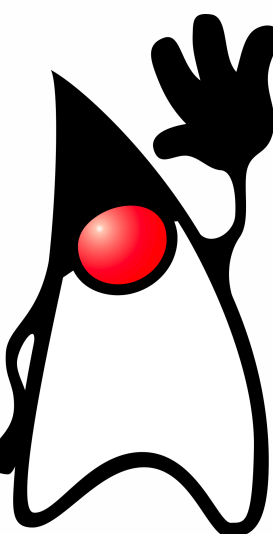
Иногда требуется выполнить «настоящий SQL».

Для этого можно использовать:

```
session.doWork  
session.doReturningWork
```

Пример: `nativeExample()`

Применение SQL следует по возможности избегать, т.к. эти операции идут мимо Persistence Context, могут быть странные «спецэффекты».



# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

**Спасибо за внимание!**

