



# Quick Overview

---



# What is Hibernate

- ORM framework– Object Relational Mapping
  - Popular
  - Easy to manage data manipulation
  - For Java
  - Open source
-

# Only Hibernate to ORM??

**No.....but it is famous and widely use**



# Hibernate

- **Simplify the repository layer**
  - **Clean code**
  - **Optimize with generated SQL commands**
-

# What is JPA

- **Java Persistence API**
- **Provide ORM facility for developers**
- **Between in OO Model and Relational Model**
- **Need a implementation**

**<https://www.oracle.com/technical-resources/articles/javase/persistenceapi.html>**

---

# JPL

- **JPL – Jakarta Persistence Layer**
- **Introduces after migrate of Java EE to Jakarta EE**
- **Package name different from JPA**

***javax.persistence*** → ***jakarta.persistence***

---

# Hibernate Flavors

**Native  
Hibernate**

**Hibernate  
JPA**





# Native Hibernate

## JPA

## JDBC



# Native Hibernate

**Several functionalities**

**Limited use**

**Entry point – Session Factory**

---

# Hibernate JPA

**Widely use**

**Entry point – Entity Manager Factory**

---

# Key Terms

**Bootstrapping - Start**

**hbm2ddd – Hibernate mapping to data definition language**

**Hibernate Dialect – Select the matching DBMS to generate SQL commands**

**Session – The time duration to deal with the DBMS**

---

# Key Annotations

**@Entity – Nominate as an entity**

**@Table – Given table name**

**@Id – Nominate the Primary Key**

**@Column – Given custom column name**

---

# Automatic Schema Generation

- **none**
  - **create only**
  - **create drop**
  - **validate**
  - **create**
  - **update**
- 



# Native Hibernate vs JPA

Native Hibernate	JPA
<b>get</b>	<b>find</b>
<b>load</b>	<b>getReference</b>
save	persist
evict	detach
merge	merge
update	-
delete	remove

**All JPA commands are valid in Native Hibernate**

---

## Eager Fetching

**Hit the DB in real-time**

## Lazy Fetching

**Not real-time hit the DB and  
maintain a proxy – represent real  
object record**

**Use performance increase and apply  
in like delete, update operations**

---

## Transient Stage

- **The object created just use "new "k/w**
- **Not associated with any session /entity manager**
- **Not belongs with any DB operation**

## Persistent Stage

- **Associate with the session / entity Manager**
  - **Changes are tracked (Dirty checking)**
-



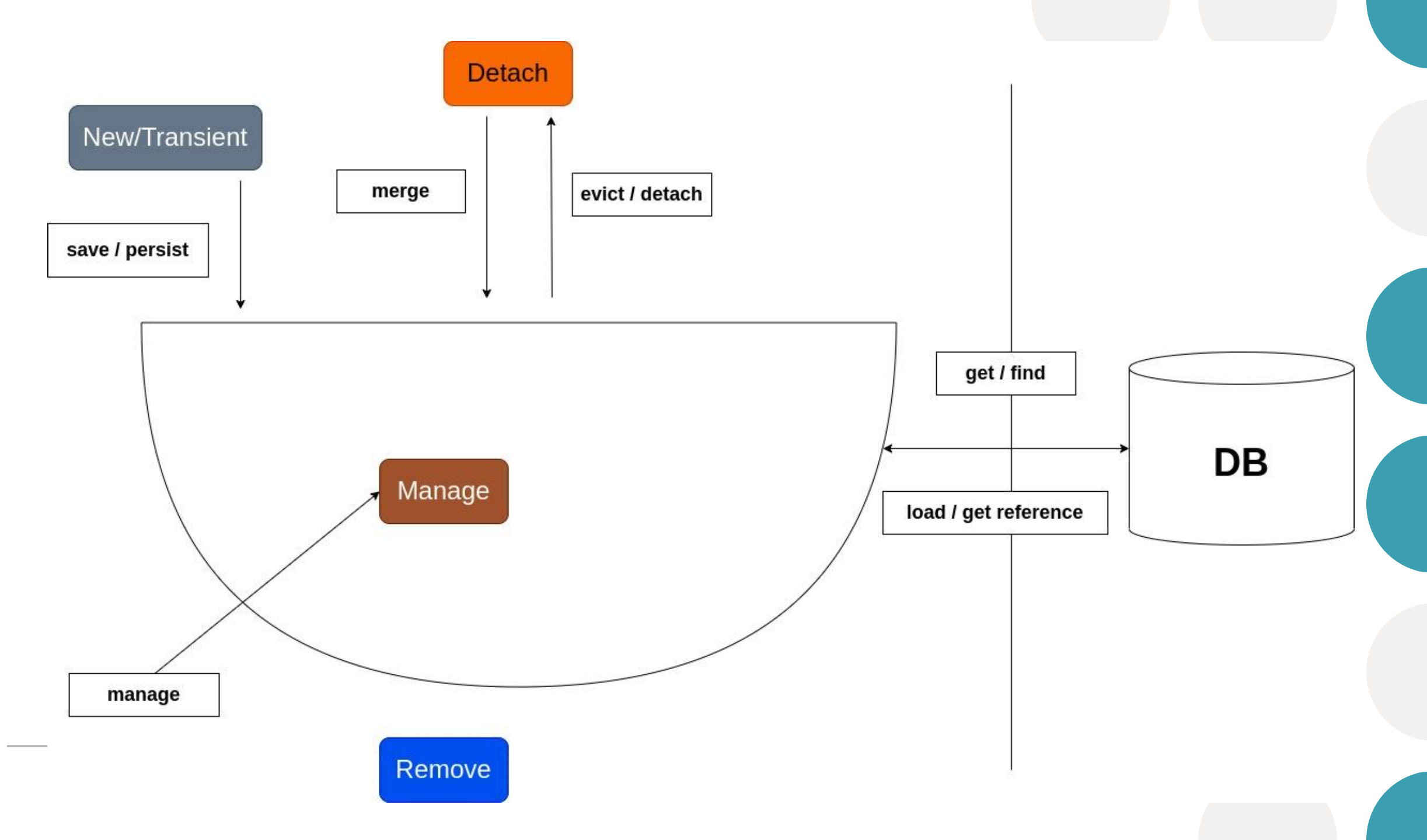
## Detached Stage

- **Not actively managed by Hibernate**
- **Changes are not persisted**
- **Possibility to re-associate**

## Removed Stage

- **No longer associated with the DB**
  - **Cannot attempt to re-associate**
- 





- **Eligible to dirty checking when manage stage**
  - **Listening about the entities that on the manage state via reflection**
  - **No other states eligible for dirty checking**
  - **When apply Hibernate commands, entities goes to the inside of the persistence context to eligible dirty checking**
  - **Applied transactions when in the context and hit the DB**
-

# Thanks

