

## JDBC API - ZALETY

- ➔ Czyste i proste przetwarzanie SQL
- ➔ Dobra wydajność z dużymi danymi
- ➔ Bardzo dobry do małych aplikacji
- ➔ Prosta składnia, łatwa do nauczenia



## JDBC API - WADY

- ➔ Złożone, jeśli jest używane w dużych projektach
- ➔ Dużo kodu
- ➔ Trudno wdrożyć koncepcję MVC
- ➔ Zapytanie jest specyficzne dla systemu DBMS



## ORM

Mapowanie obiektowo-relacyjne (ang. Object-Relational Mapping ORM) – sposób odwzorowania obiektowej architektury systemu informatycznego na bazę danych (lub inny element systemu) o relacyjnym charakterze.



## JPA

Java Persistence API (skrót JPA) – oficjalny standard mapowania obiektowo-relacyjnego (ORM) firmy Sun Microsystems dla języka programowania Java.



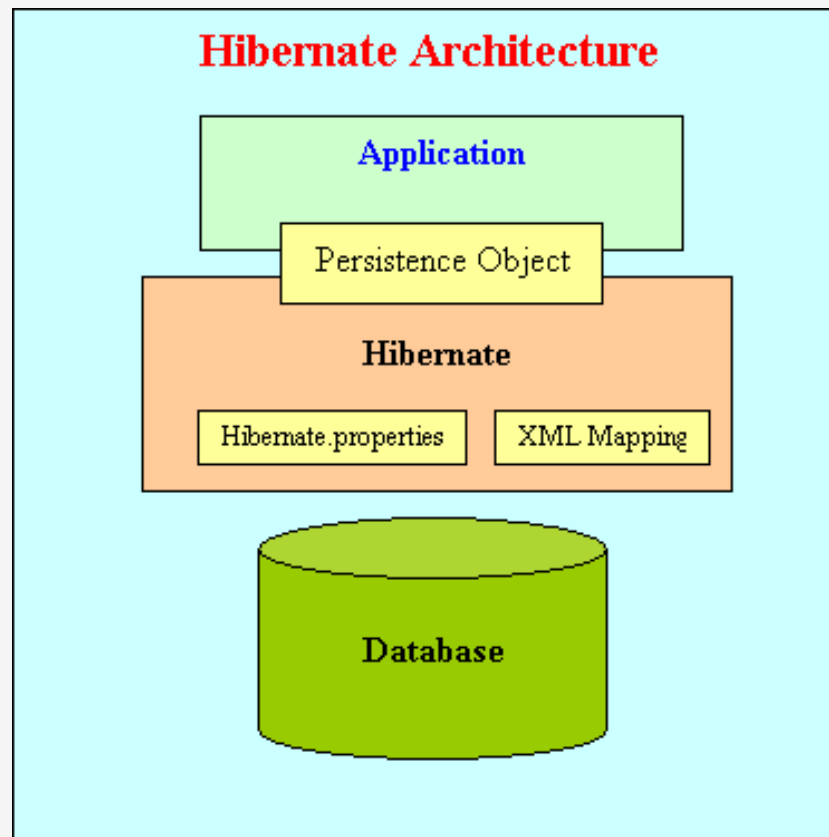
# JPA

Implementacje JPA:

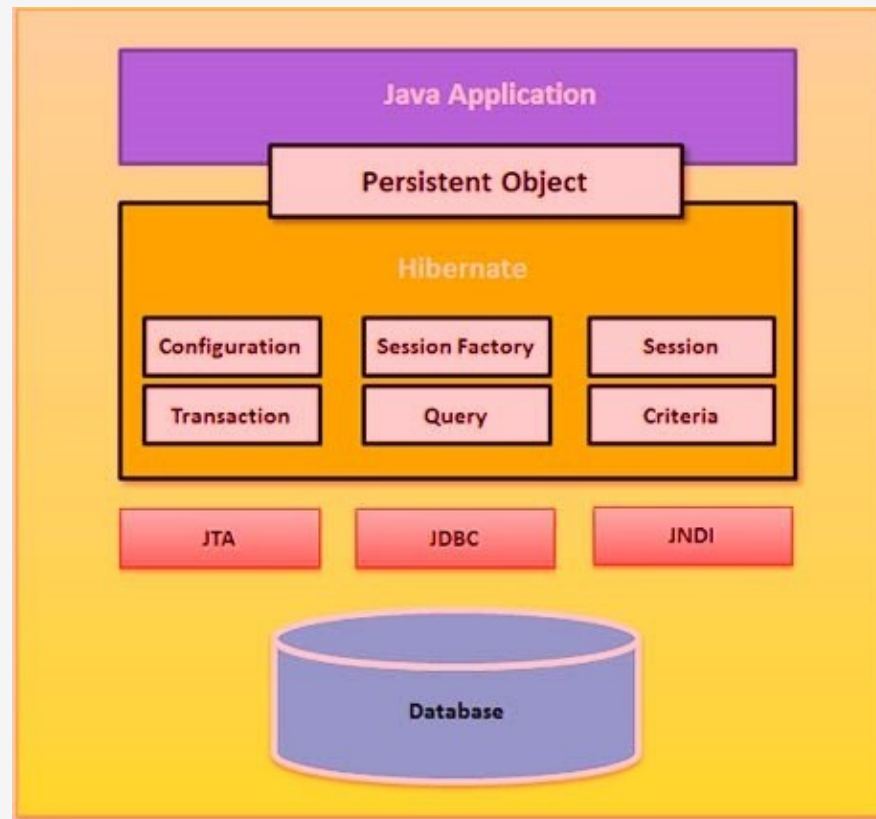
- Hibernate
- TopLink
- Kodo
- Open JPA



# Hibernate Architecture



# Hibernate Core



## Configuration

Obiekt Configuration jest pierwszym obiektem Hibernate tworzonym w dowolnej aplikacji korzystającej z tej biblioteki. Zwykle jest tworzony tylko raz podczas inicjowania aplikacji. Reprezentuje plik konfiguracji lub właściwości wymagany przez Hibernate.





## SessionFactory

Obiekt Configuration służy do tworzenia obiektu SessionFactory, który z kolei konfiguruje Hibernate dla aplikacji przy użyciu dostarczonego pliku konfiguracyjnego i umożliwia utworzenie instancji obiektu Session. SessionFactory jest obiektem bezpiecznym dla wątków i jest używany przez wszystkie wątki aplikacji.



## SessionFactory

SessionFactory jest obiektem ciężkim. Zwykle jest tworzony podczas uruchamiania aplikacji i przechowywany do późniejszego użycia. Potrzebny byłby jeden obiekt SessionFactory na bazę danych przy użyciu oddzielnego pliku konfiguracyjnego. Tak więc, jeśli używasz wielu baz danych, będziesz musiał utworzyć wiele obiektów SessionFactory.



## Session

Sesja służy do uzyskania fizycznego połączenia z bazą danych. Obiekt Session jest lekki i zaprojektowany do tworzenia za każdym razem, gdy potrzebna jest interakcja z bazą danych. Zmapowane obiekty są zapisywane i pobierane za pośrednictwem obiektu Session. Obiekty sesji nie powinny być otwarte przez długi czas, ponieważ zazwyczaj nie są bezpieczne i powinny być tworzone i niszczone w razie potrzeby.



## Transaction

Odpowiada jednej transakcji bazodanowej. Jest związana z konkretną sesją, tworzona jest przy pomocy:

```
Transaction tx = session.beginTransaction();
```

```
tx.commit();
```

```
tx.rollback();
```



## Query

Obiekt ten służy do wykonywania zapytań do bazy danych napisanych w HQL lub SQL.



## POJO

POJO (Plain Old Java Object) – to klasy modelu. Posiadają jedynie pola, gettery i settery. Ich zadaniem nie jest wykonywanie skomplikowanych operacji, a jedynie przechowywanie informacji.

