

はまる！JPA (LTシヨート版)

(株)カサリアル 多田真敏

OSS勉強会 #21 Java EE

2017年12月11日(火)

自己紹介

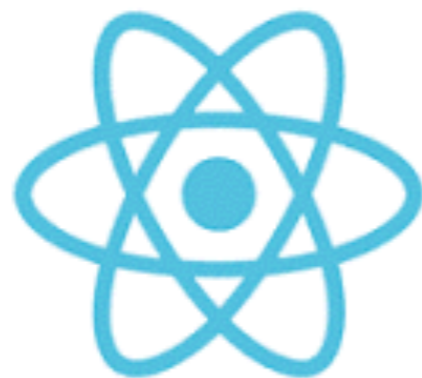
- ▶ 多田真敏 (@suke_masa)
 - ▶ 研修トレーナー@カサレアル
 - ▶ Spring / Java EE
 - ▶ Pivotal認定講師 (国内では数人)
 - ▶ JSUGスタッフ、GUGJ運営メンバー



**AUTHORIZED
TRAINING
PARTNER**

カサレアル会社紹介

- ▶ 色々なプログラミング言語の研修をやっています！



どんなORMマッパーを使っていますか？

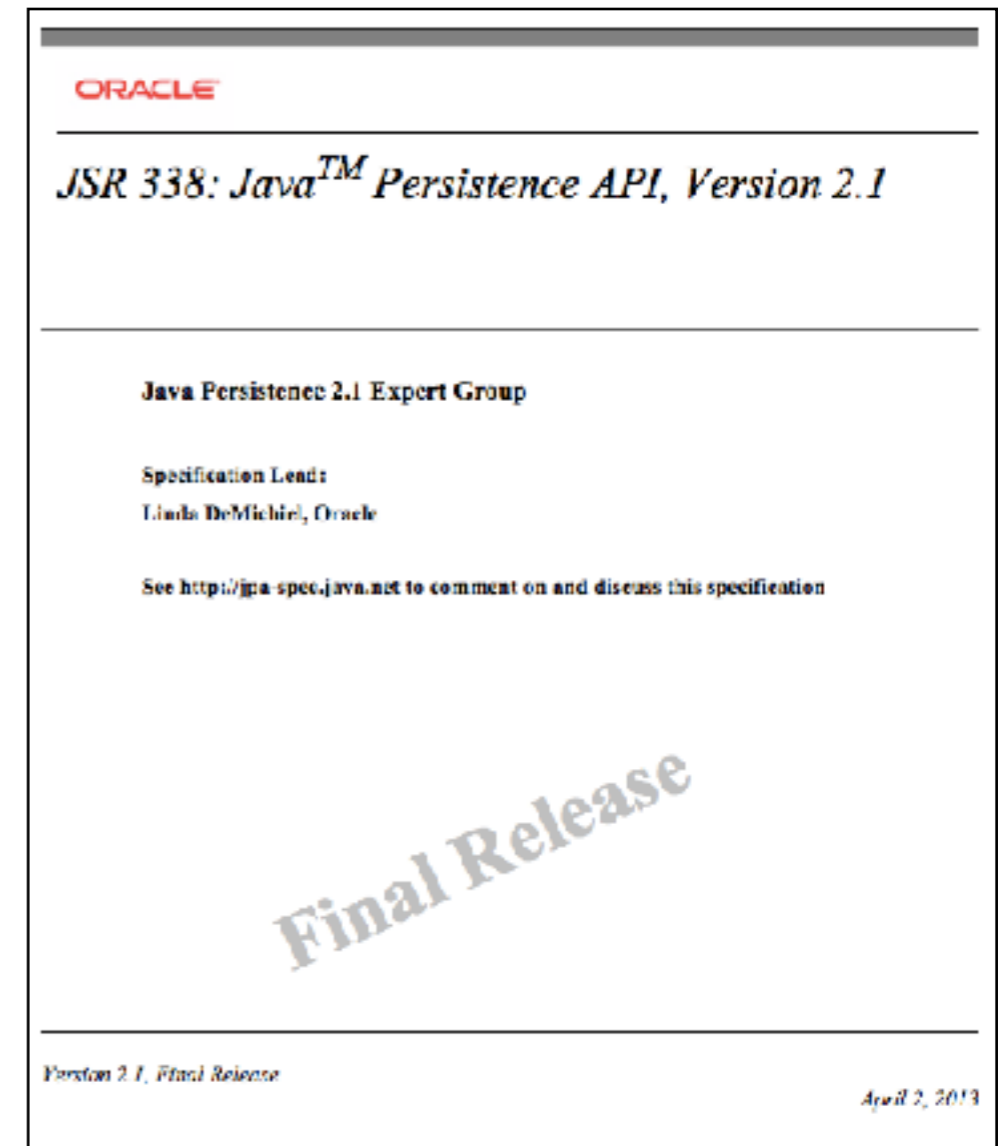
- ① JPA
- ② MyBatis
- ③ Hibernate (非JPA)
- ④ 漢なら生JDBC
- ⑤ その他

なぜJPAを選びましたか？

- ① 標準技術だから
- ② 高機能だから
- ③ 要件に合っていたから
- ④ その他

JPAとは？

- ▶ Java EEの一部として標準化されたデータアクセス技術の仕様
- ▶ 実体はJSR 338という文書
- ▶ 決まっているのはインタフェース・アノテーション・例外がほとんど



JPA実装ライブラリ

① Hibernate



- ▶ O/Rマッパーの草分け的存在
- ▶ Spring Data JPAは基本的にコレ、JBoss内包

② EclipseLink

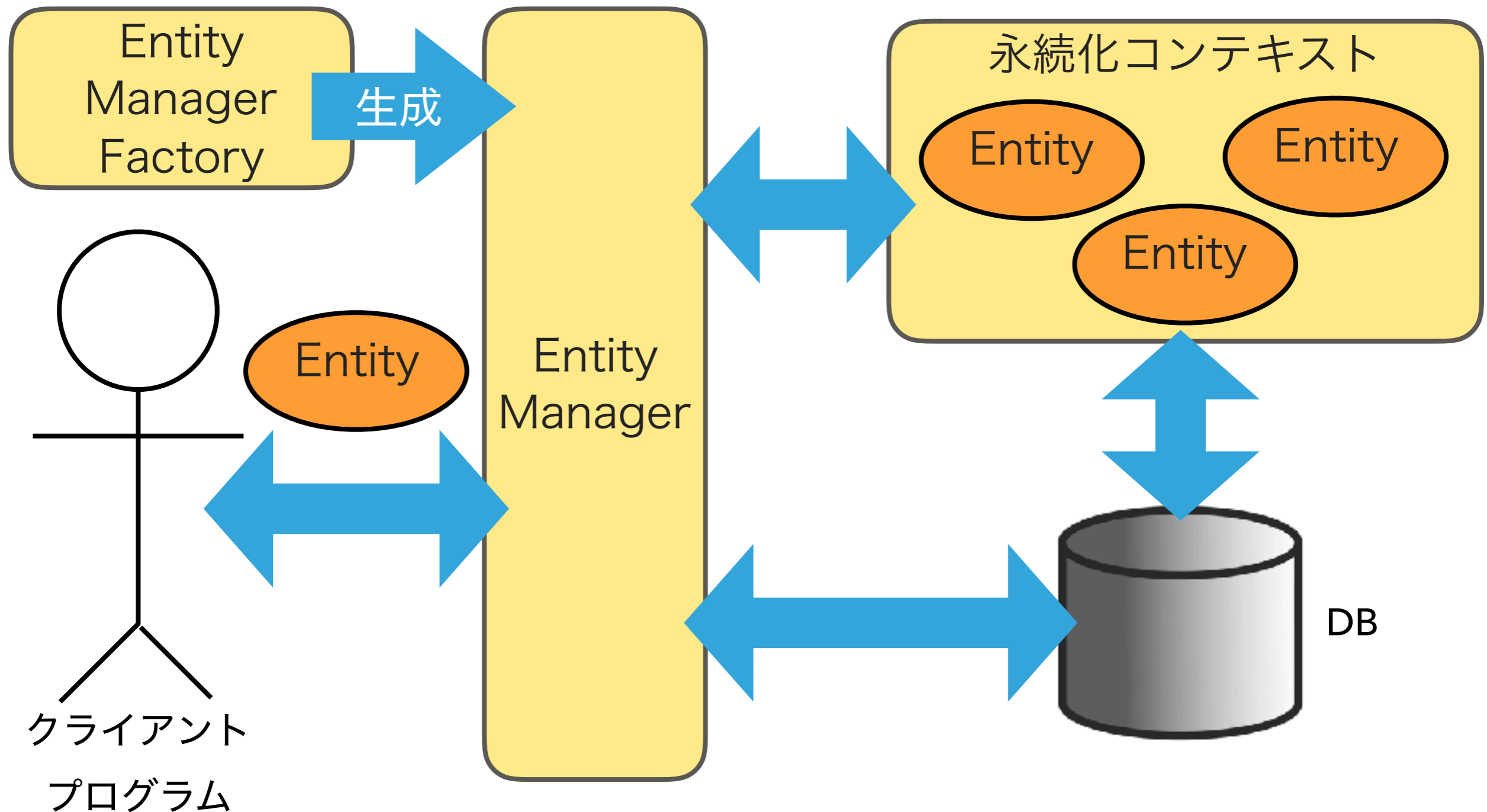


- ▶ JPAの参照実装
- ▶ GlassFish / WebLogicなどに内包

今回はコレ

※他の実装はJPA 2.1(Java EE 7)に対応していない

JPAのアーキテクチャ

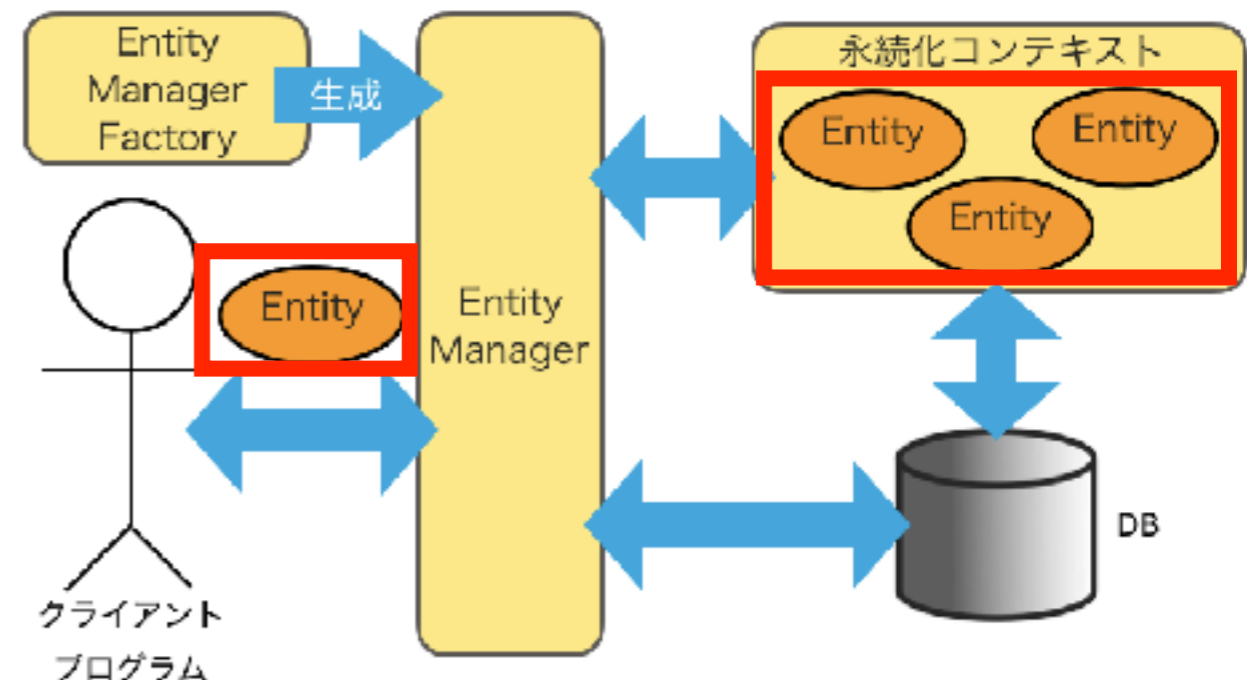


エンティティ

- ▶ 基本的にテーブルと1対1で作る
 - ▶ 「SELECT文と1対1」ではない！
- ▶ 主キーフィールド必須

@Entity

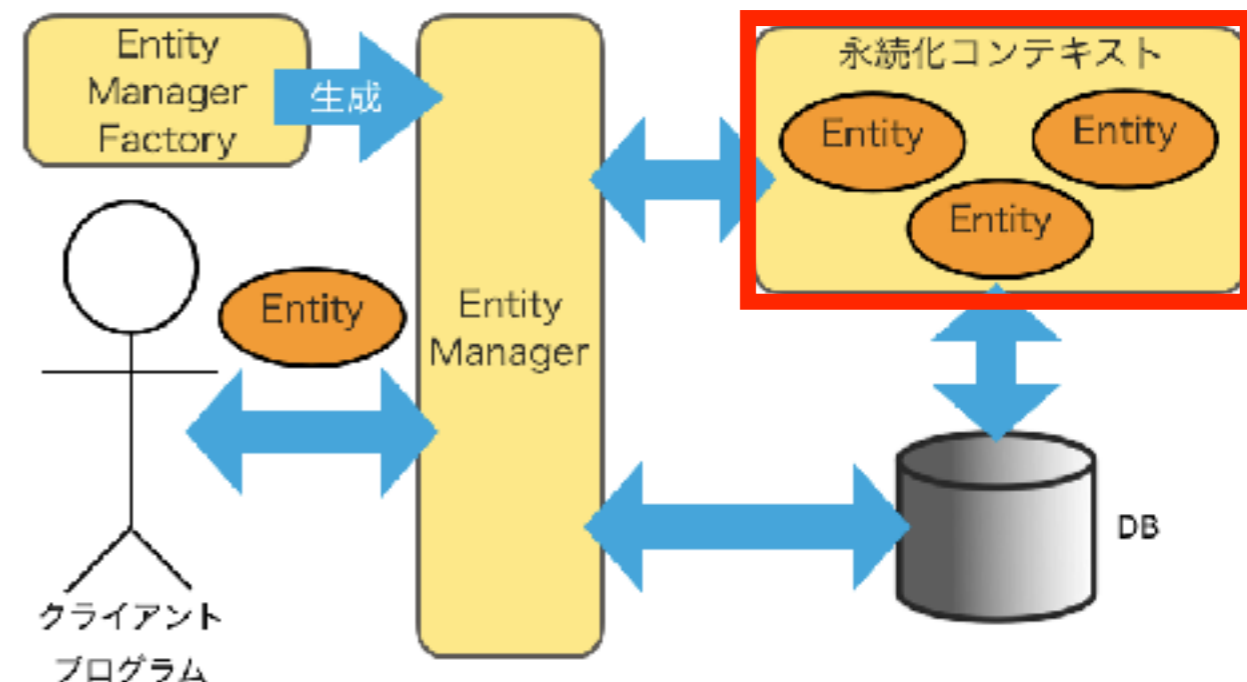
```
public class Product {  
    @Id  
    private Integer id;  
    private String name;  
    // setter/getter  
}
```



永続化コンテキスト

- ▶ エンティティのキャッシュのようなもの
 - ▶ エンティティが永続化コンテキストに含まれているかどうか非常に重要

- ▶ **トランザクション単位**で生成・破棄される



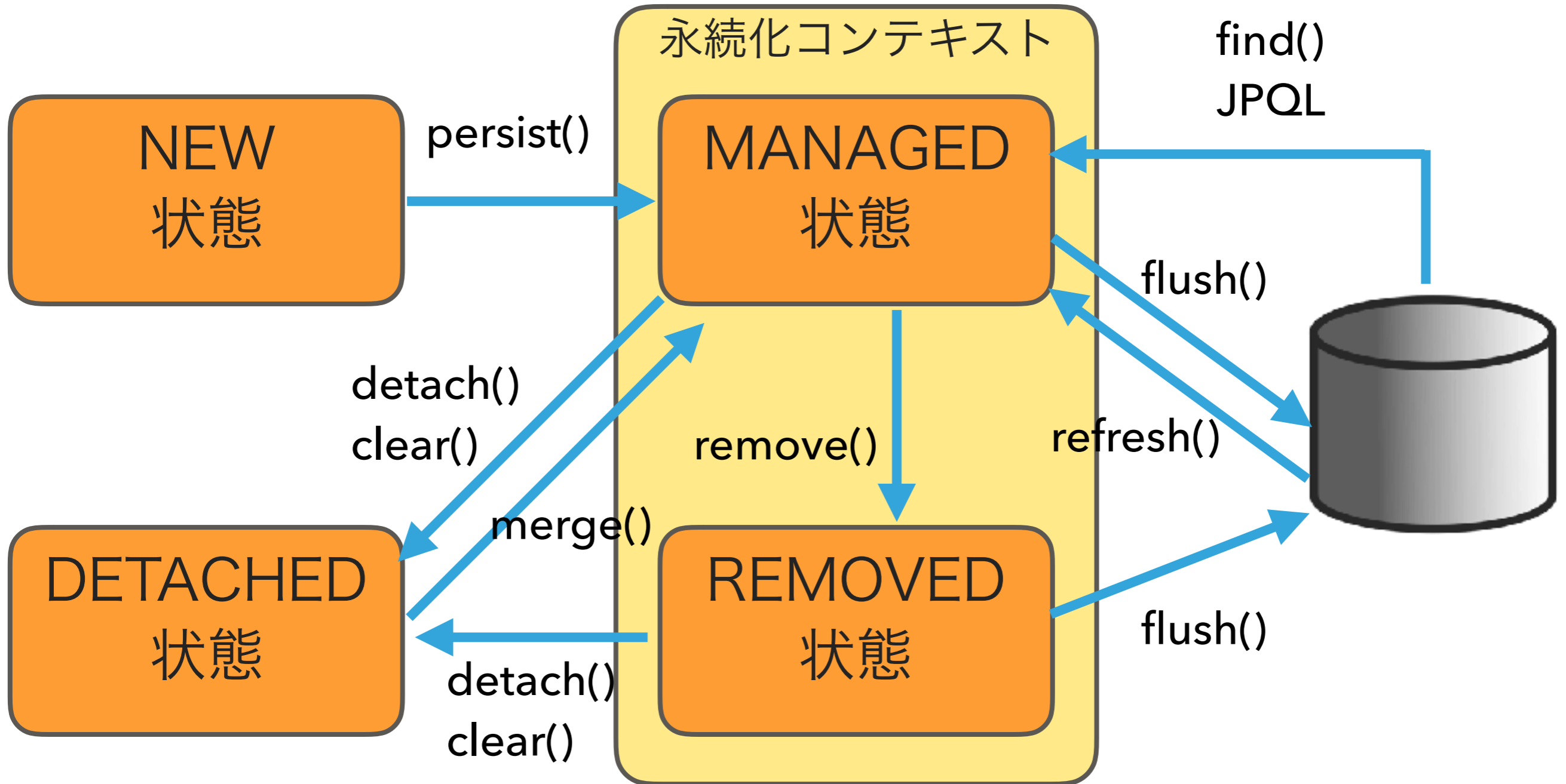
エンティティの「状態」

状態	説明
NEW	newされたばかり
MANAGED	永続化コンテキストで管理されている
DETACHED	永続化コンテキストから分離されている
REMOVED	削除が予約されている

EntityManagerのメソッド

メソッド	説明
find()	主キー検索
persist()	1件の永続化
remove()	1件の削除
merge()	永続化コンテキスト管理下にする
detach()	永続化コンテキスト管理外にする
flush()	変更をDBへ反映
clear()	永続化コンテキストをクリア
refresh()	エンティティの内容をDB最新に上書き

★重要★ エンティティの状態遷移



主キー検索

```
Manufacturer m =  
    em.find(Manufacturer.class, 19985678);  
System.out.println(m.getName());
```

```
[EL Fine]: sql: Connection(50826412)--SELECT  
MANUFACTURER_ID, ADDRESSLINE1, ADDRESSLINE2,  
CITY, EMAIL, FAX, NAME, PHONE, REP, STATE, ZIP  
FROM MANUFACTURER WHERE (MANUFACTURER_ID = ?)  
bind => [19985678]
```

Happy End Searching

新規追加

```
EntityTransaction tx =  
    em.getTransaction();  
tx.begin();  
Manufacturer m = new Manufacturer();  
m.setManufacturerId(1);  
m.setName("HOGE");  
em.persist(m);  
em.flush();  
tx.commit();
```

更新時の注意点

- ▶ 対象のエンティティがMANAGED状態でないと、更新が反映されない
 - ▶ 方法①：find()で検索したものを変更する
 - ▶ 方法②：newしたエンティティをmerge()する

更新

```
EntityTransaction tx = ...  
tx.begin();  
Manufacturer m =  
    em.find(Manufacturer.class, 1);  
m.setName("FUGA");  
em.flush();  
tx.commit();
```

削除

```
EntityTransaction tx = ...  
tx.begin();  
Manufacturer m = new Manufacturer();  
m.manufacturerId(1);  
m.setName("FUGA");  
em.remove(m);  
em.flush();  
tx.commit();
```

削除すると . . .

Exception in thread "main"

```
java.lang.IllegalArgumentException: Entity  
must be managed to call remove:  
com.example.entity.Manufacturer[ manufactu  
rerId=1 ], try merging the detached and  
try the remove again.
```

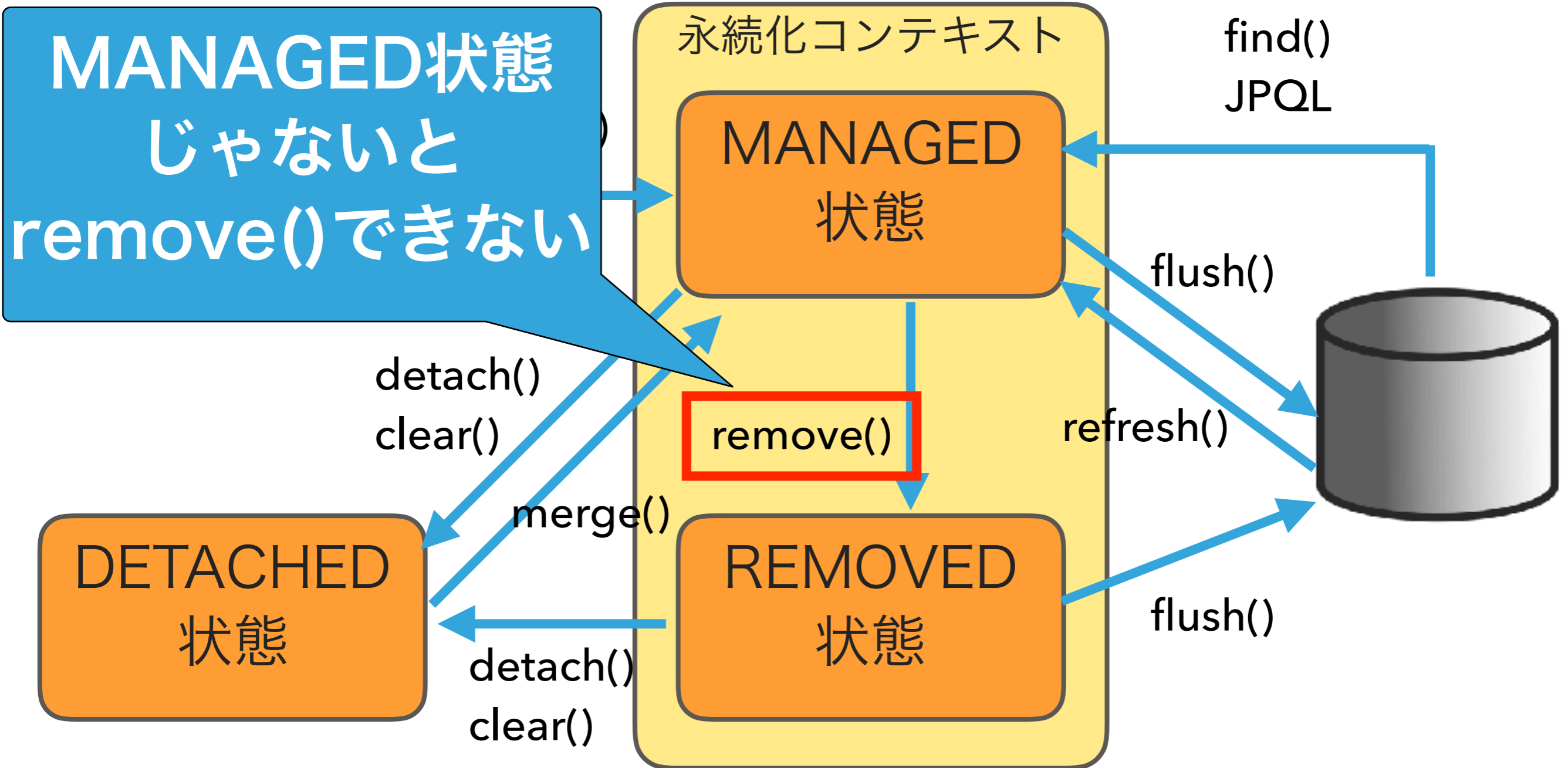
例外! ?

はまりポイント①

- ▶ 削除だけできません問題



[再確認] エンティティの状態遷移



merge()して削除

```
EntityTransaction tx = ..  
tx.begin();  
Manufacturer m = new Manufacturer();  
m.manufacturerId(1);  
m.setName("FUGA");  
em.merge(m);  
em.remove(m);  
em.flush();  
tx.commit();
```

merge()すれば
MANAGED状態に
なるはず！

merge()して削除すると・・・

Exception in thread "main"

```
java.lang.IllegalArgumentException: Entity  
must be managed to call remove:  
com.example.entity.Manufacturer[ manufactu  
rerId=1 ], try merging the detached and  
try the remove again.
```

同じ例外！？
MANAGED状態になってない！？

はまりポイント②

- ▶ merge() できません問題



Javadocを確認

merge

`<T> T merge(T entity)`

Merge the state of the given entity into the current persistence context.

Parameters:

`entity` - entity instance

Returns:

the managed instance that the state was merged to

merge()に
戻り値がある！？

仕様書を確認

▶ 3.2.7.1 Merging Detached Entity State

- ▶ エンティティXがDETACHED状態の場合、
 - ▶ Xと同じ識別子を持つMANAGED状態のX'にコピーされる
 - ▶ またはMANAGED状態のX'が新規生成される

引数のコピーがMANAGED状態になり、
それが戻り値となる

→ 引数自体はMANAGED状態にならない！

戻り値をremove() !

```
EntityTransaction tx =  
tx.begin();  
Manufacturer m = new M  
m.manufacturerId(1);  
m.setName("FUGA");  
Manufacturer managedManu = em.merge(m);  
em.remove(managedManu);  
em.flush();  
tx.commit();
```

merge()の戻り値を
remove()
→削除できる!

他にもはまりポイントはいっぱい！

- ▶ 状態がらみ
- ▶ リレーションがらみ
- ▶ JPQLがらみ
- ▶ ...



ちよつと補足

- ▶ JPAをディスってる訳ではありません
 - ▶ 多田はほぼいつもJPA
 - ▶ 正しく使えば、コード量の少なさ = 生産性は圧倒的
 - ▶ ただ、ハマりポイントは多いので注意してください！
という話です

結論

- ▶ 「標準技術だから」という理由**だけ**での採用は危険！
- ▶ 仕組みの理解が何より大事！
- ▶ メリット・デメリット勘案して決めましょう！

JPAを使っていい条件

- ① DBを新規に設計できる
- ② 集合演算やFROM句での副問合せなど、複雑なSQLは要件的に少ない
- ③ 「パーフェクトJava EE」を読破した人がプロジェクトに1人以上いる

1つでも当てはまらない項目があれば、他のORMマッパーを使った方がいいかも 😊💧

[告知] 12/19 GlassFish勉強会

▶ Java EEユーザーは是非ご参加ください！

▶ 12月19日(火) 19:00-20:30

▶ 終了後に忘年会もやります！

▶ <https://glassfish.doorkeeper.jp/events/67871>

