GEDOPLAN

Java Persistence in Action
Features jenseits des Entry-Levels

Dirk Weil, GEDOPLAN GmbH

GEDOPLAN

Dirk Weil

- GEDOPLAN GmbH, Bielefeld
- Java EE seit 1998
- Konzeption und Realisierung
- Seminare
- Vorträge
- Veröffentlichungen

GEDOPLAN

- Legacy-Herausforderungen
- Tuning
- Providerspezifisches

GEDOPLAN

Composite IDs

- Basics:
  - ID-Klasse

```
@Embeddable
public class BranchId implements Serializable
{
  private int companyId;
  private int branchNo;
```

GEDOPLAN

Composite IDs

- Basics:
  - `@EmbeddedId`
    oder mehrere `@Id`

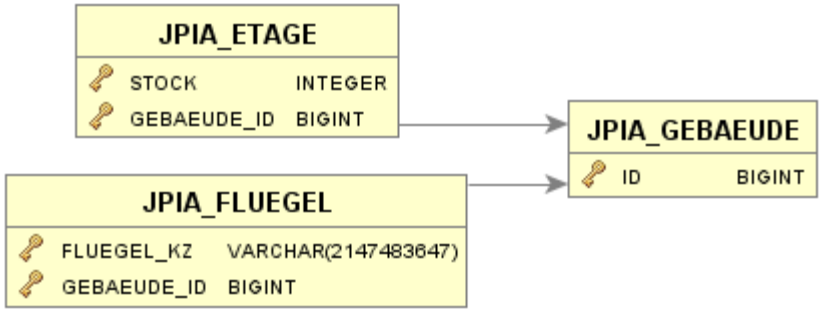
```
@Entity
public class SuperMarket
{
  @EmbeddedId
  private BranchId id;
```

```
@Entity
@IdClass(BranchId.class)
public class PetrolStation
{
  @Id
  private int    companyId;

  @Id
  private int    branchNo;
```

GEDOPLAN

Composite IDs

- PK/FK-Überlappung



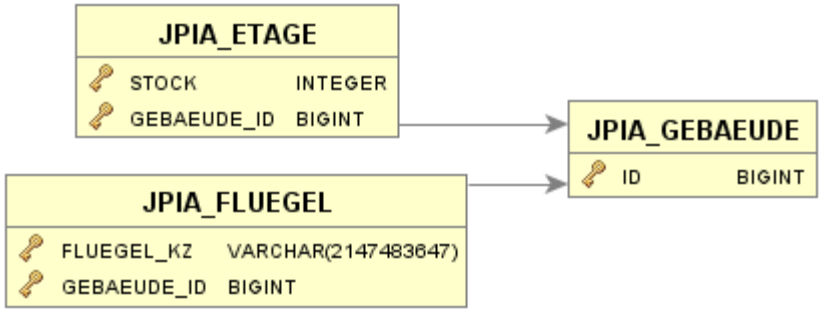

```
@Embeddable
public class EtageId
  implements Serializable
{
  @Column(name = "GEBAEUDE_ID")
  private Long gebaeudeId;
  private int  stock;
```

```
@Entity
public class Etage
{
  @EmbeddedId
  private EtageId  id;

  @MapsId("gebaeudeId")
  @ManyToOne
  private Gebaeude gebaeude;
```

GEDOPLAN

Composite IDs

- PK/FK-Überlappung



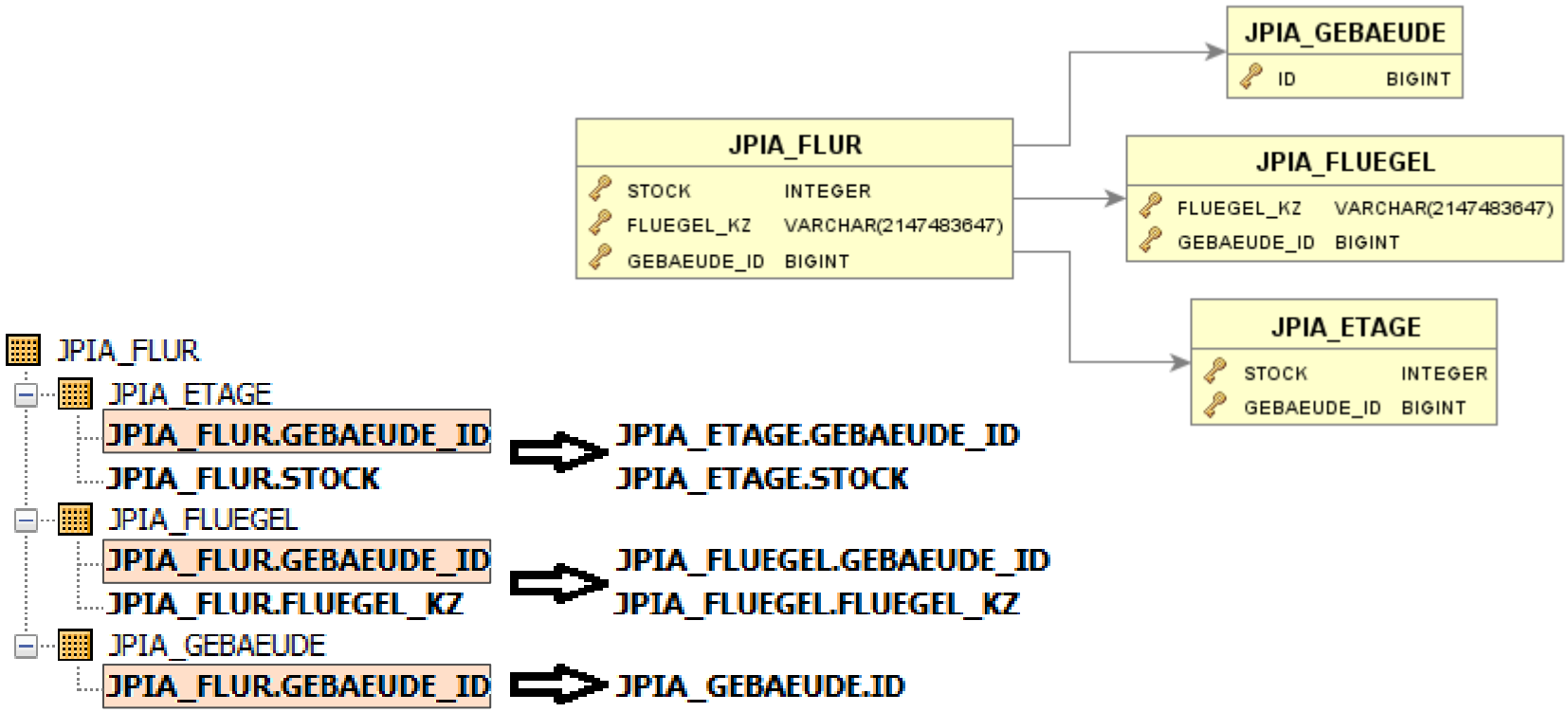

```
public class FluegelId
  implements Serializable
{
  private Long   gebaeude;
  private String fluegelKz;
```

```
@Entity
@IdClass(FluegelId.class)
public class Fluegel
{
  @Id @ManyToOne
  private Gebaeude gebaeude;
  @Id @Column(name = "FLUEGEL_KZ")
  private String   fluegelKz;
```

GEDOPLAN

Composite IDs

- PK/FK-Überlappung



GEDOPLAN

Composite IDs

- Mehrfach-Mapping auf gleiche Spalte

```
…
@Id @ManyToOne
private Gebaeude  gebaeude;

@ManyToOne
@JoinColumns({
  @JoinColumn(name = "GEBAEUDE_ID", referencedColumnName = "GEBAEUDE_ID",
              insertable = false, updatable = false),
  @JoinColumn(name = "FLUEGEL_KZ", referencedColumnName = "FLUEGEL_KZ",
              insertable = false, updatable = false)
})
private Fluegel   fluegel;
…
```

GEDOPLAN

Composite IDs

- Mehrfach-Mapping auf gleiche Spalte
  - s. Demo-Klassen **`Flur`** und **`Raum`**
  - Nachteile:
    - Überlappende Relationsattribute können nicht unabhängig voneinander gesetzt werden
    - **`null`** ist i. A nicht möglich

GEDOPLAN

Optimistic Locking

- Standardverfahren:
  - Versionsattribut
  - setzt zusätzliche Spalte voraus

```
@Entity
public class Person
{
  @Version
  private long version;
```

ID	NAME	VORNAME	VERSION
1	Maier	Sepp	2
2	Mustermann	Max	1

GEDOPLAN

Optimistic Locking

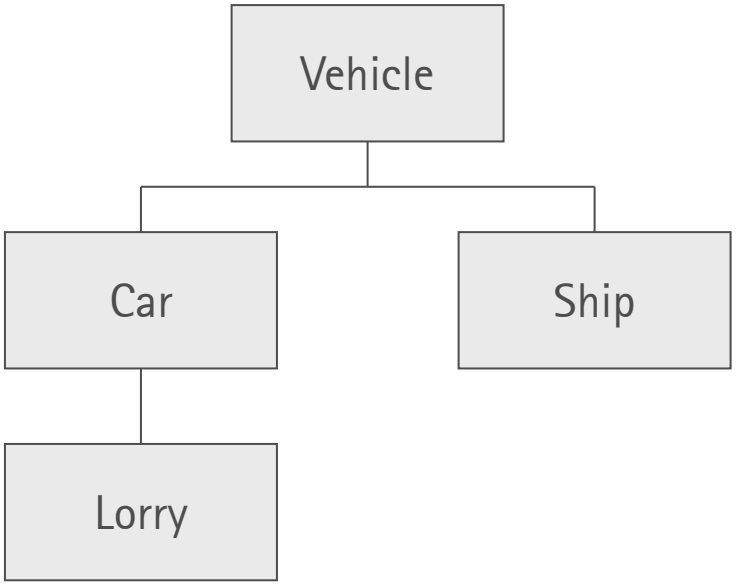
- Providerabhängig weitere Verfahren ohne explizite Versionsspalte

verglichene Spalten	EclipseLink	Hibernate	OpenJPA
alle	✓	✓	✓
veränderte	✓	✓	
weitere	✓		✓

GEDOPLAN

Inheritance

- Mögliche Strategie SINGLE_TABLE
  - benötigt Diskriminator-Spalte



JPIA_VEHICLE	
ID	VARCHAR(2147483647)
NAME	VARCHAR(2147483647)
BRZ	DOUBLE(17,0)
NOOFDOORS	INTEGER
PAYLOAD	DOUBLE(17,0)
DTYPE	VARCHAR(31)

ID	NAME	BRZ	NOOFDOORS	PAYLOAD	DTYPE
C.02	Golf	(null)	5	(null)	Car
C.01	Smart	(null)	3	(null)	Car
L.01	Actros	(null)	2	35.0	Lorry
S.01	Norwegian Breakaway	146600.0	(null)	(null)	Ship
S.02	Disney Fantasy	130000.0	(null)	(null)	Ship

GEDOPLAN

Inheritance

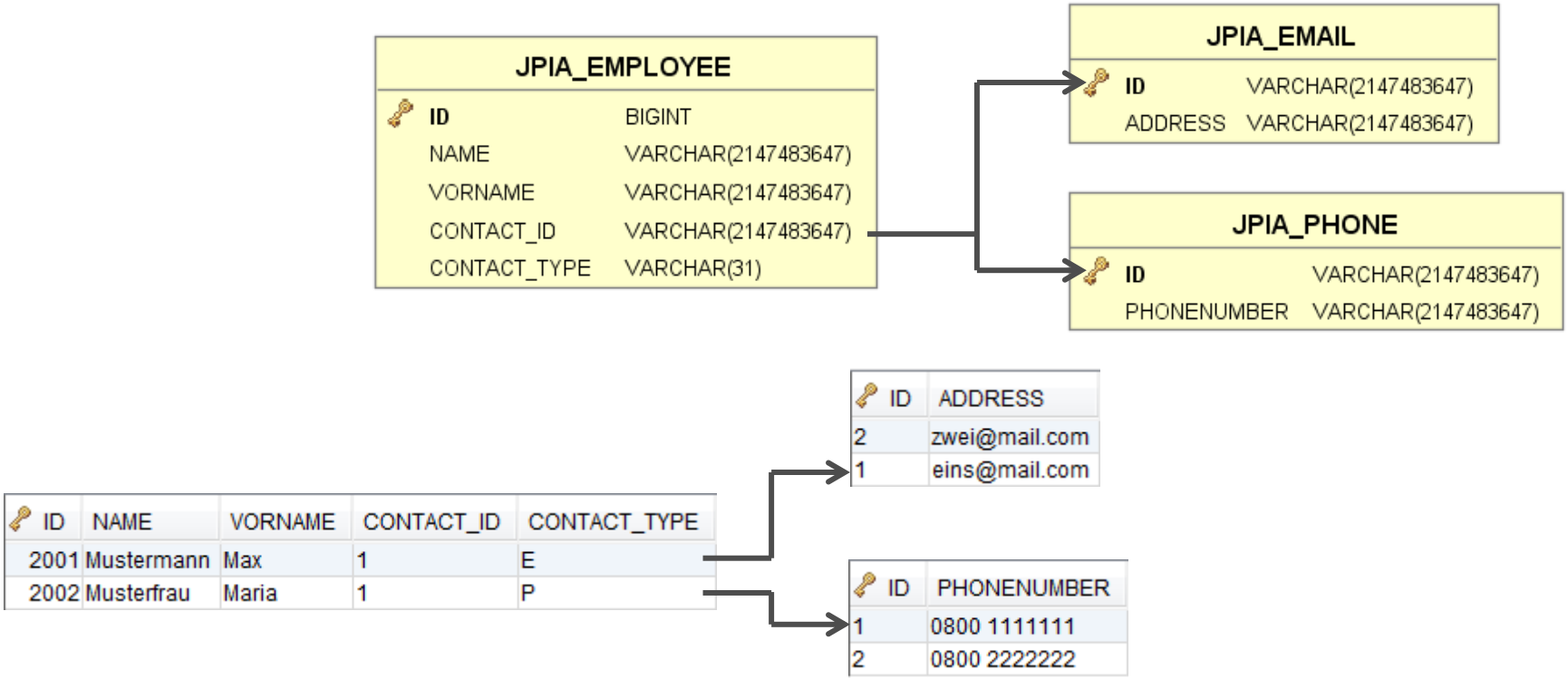
- Provider-abhängig auch ohne Diskriminator-Spalte

	Typ-Ermittlung für gelesene Zeilen	Query-Filter
EclipseLink	`@ClassExtractor`	`@Customizer`
Hibernate	`@DiscriminatorFormula`	
OpenJPA	`@DiscriminatorStrategy`	

GEDOPLAN

Polymorphe Assoziationen

- Foreign Key mit multiplem Ziel



GEDOPLAN

Polymorphe Assoziationen

- Nur Provider-abhängig möglich

	Annotation für Relationsattribut
EclipseLink	`@VariableOneToOne`
Hibernate	`@Any`, `@AnyMetaData`

GEDOPLAN

Custom Types

- Unnatürliche DB-Werte, z. B.:
  `true` → `'J'`
  `false` → `'N'`

	Annotation
EclipseLink	`@ObjectTypeConverter`, `@Converter`
Hibernate	`@Type`
OpenJPA	`@ExternalValues`, `@Type`

- Provider-abhängiges Mapping
  - s. Demo-Klasse **`Projekt`**
- Alternative: Life Cycle Methods

GEDOPLAN

Custom Types

- Seit JPA 2.1: Converster

```
@Entity
public class Projekt
{
  @Convert(converter = JNConverter.class)
  private boolean aktiv;
```

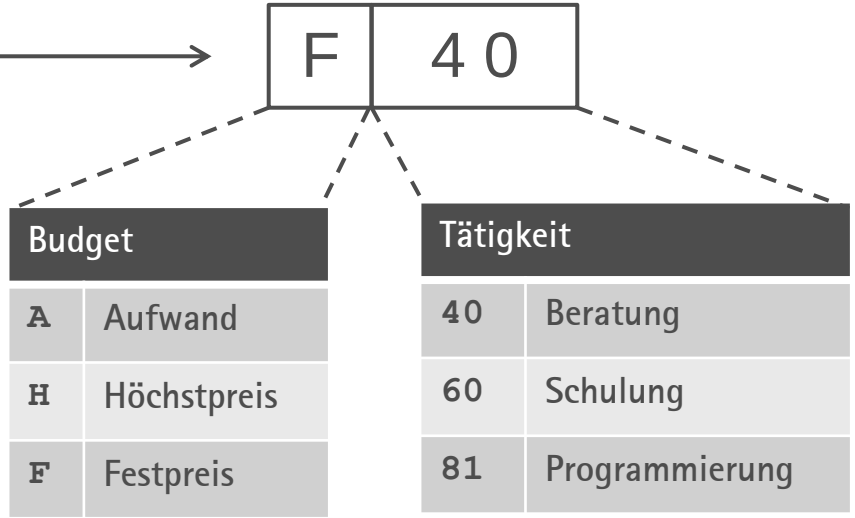
```
@Converter
public class JNConverter implements AttributeConverter<Boolean, String> {
  public String convertToDatabaseColumn(Boolean attributeValue) {
    if (attributeValue == null) return null;
    return attributeValue ? "J" : "N";
  }

  public Boolean convertToEntityAttribute(String columnValue) {
    if (columnValue == null) return null;
    return columnValue.equals("J");
```

GEDOPLAN

Packed Column Values

- Mehrere Werte in einer DB-Spalte

```
@Entity
public class Projekt
{
  …
  private String projektTyp;
```



GEDOPLAN

Packed Column Values

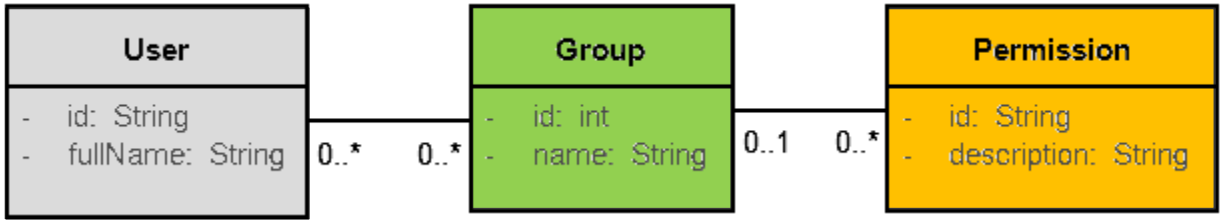
- (Un-)Boxing mittels Life Cycle Methods

```
@Entity
public class Projekt {
  …
  private String                projektTyp;
  @Transient private BudgetTyp  budgetTyp;
  @Transient private Taetigkeit taetigkeit;

  @PrePersist @PreUpdate
  private void preStore() {
    projektTyp = budgetTyp.getKuerzel() + taetigkeit.getKuerzel();
  }
```

GEDOPLAN

Fetch Tuning

- ToMany-Fetching: "N+1"-Problem



  - z.B. Lesen der User inkl. Groups + Permissions

```
SELECT ... FROM USER
SELECT ... FROM USER_GROUP t0, GROUP t1 WHERE t0.USER_ID=? AND …
SELECT ... FROM PERMISSION WHERE GROUP_ID=?
SELECT ... FROM PERMISSION WHERE GROUP_ID=?
SELECT ... FROM USER_GROUP t0, GROUP t1 WHERE t0.USER_ID=? AND …
SELECT ... FROM USER_GROUP t0, GROUP t1 WHERE t0.USER_ID=? AND …
```

GEDOPLAN

Fetch Tuning

- Lösungsansatz 1: Join Fetching

```
CriteriaBuilder criteriaBuilder = entityManager.getCriteriaBuilder();
CriteriaQuery<User> criteriaQuery = criteriaBuilder.createQuery(User.class);

Root<User> u = criteriaQuery.from(User.class);
u.fetch(User_.groups, JoinType.LEFT).fetch(Group_.permissions, JoinType.LEFT);

criteriaQuery.select(u).distinct(true);
List<User> users = entityManager.createQuery(criteriaQuery).getResultList();
```

  - erzeugt 1 (!) SELECT
  - Achtung: Volumen!

GEDOPLAN

Fetch Tuning

- Lösungsansatz 2: Batch Fetching

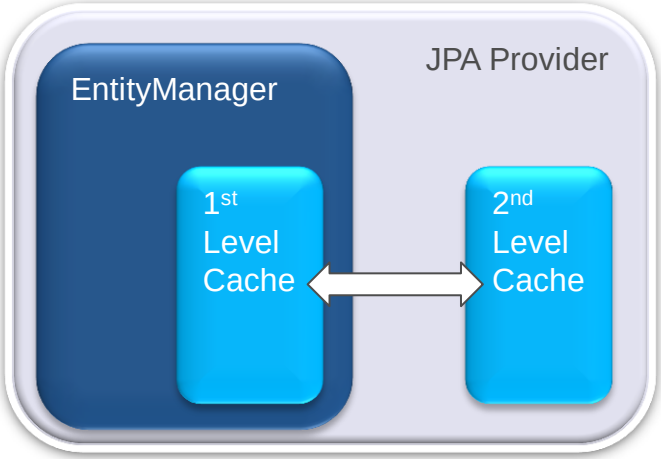
```
@ManyToMany(fetch = FetchType.LAZY, …)
@BatchFetch(value = BatchFetchType.IN, size = 10)
private Set<Group>          groups;
```

	Annotation	Relationsauflösung
EclipseLink	`@BatchFetch`	`IN, EXISTS, JOIN`
Hibernate	`@BatchSize`	`IN, EXISTS`
OpenJPA	`@EagerFetchMode` (nur für EAGER!)	`JOIN` (='PARALLEL')

GEDOPLAN

Second Level Cache

- JPA 2.0 unterstützt 2nd Level Cache
  - Wirkt applikationsweit
  - Semantik ähnlich HashMap
  - Ladereihenfolge:
    - 1st Level Cache (EntityManager)
    - 2nd Level Cache, falls enabled
    - DB



GEDOPLAN

Second Level Cache

- Konfiguration

```
@Entity
@Cacheable(true)
public class Land
{
   …
```

```
<persistence-unit name="…">
  <provider>…</provider>
  <shared-cache-mode>ENABLE_SELECTIVE</shared-cache-mode>
  …
```

	Cache aktiv für …
`ALL`	alle Entities
`NONE`	keine Klasse
`ENABLE_SELECTIVE`	nur `@Cacheable(true)`
`DISABLE_SELECTIVE`	alle außer `@Cacheable(false)`

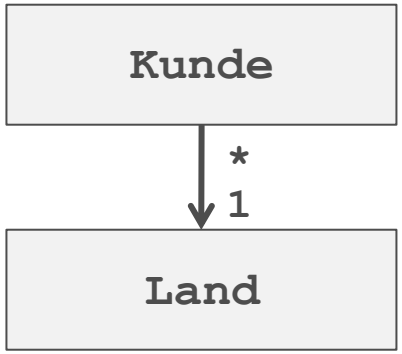
GEDOPLAN

Second Level Cache

- Vorteil bei
  - häufig genutzten
  - (nahezu) konstanten

  Daten

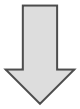
  - s. Demo-Klassen **`Kunde`** und **`Land`**



GEDOPLAN

Query Cache

- Providerspezifisch
  - i. A. Teil des 2nd Level Cache
- Speichert Result Set IDs zu Queries



- Trotz mehrfacher Ausführung nur ein DB-Zugriff

GEDOPLAN

Query Cache

- Aktivierung durch Query Hints

```
TypedQuery<Land> query = entityManager.createQuery(…, Land.class);
query.setHint("eclipselink.cache-usage", "CheckCacheThenDatabase");
List<Land> laender = query.getResultList();
```

  - Map-Semantik
    - Key = Query-Text + Parameter
  - s. Demo-Klasse **`Land`**

GEDOPLAN

Fazit

- JPA out-of-the-box nur für
  - "grüne Wiese"
  - DB-Modelle mittlerer Komplexität
- Legacy-Anforderungen lösbar
  - teilweise nur Provider-spezifisch
- Diverse "Stellschrauben" zur Optimierung
  - teilweise nur Provider-spezifisch

GEDOPLAN

Präsentation / Demo-Projekt

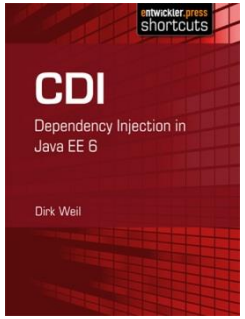
- `http://de.slideshare.net/gedoplan/javaPersistenceInActionFeaturesJenseitsDesEntryLevels`
- `https://github.com/dirkweil/javaPersistenceInAction`

GEDOPLAN

More

- http://www.gedoplan-it-training.de
  Seminare in Berlin, Bielefeld, Inhouse
- http://www.gedoplan-it-consulting.de
  Reviews, Coaching, ...

- http://javaeeblog.wordpress.com/
- http://expertenkreisjava.blogspot.de/



- dirk.weil@gedoplan.de
- @dirkweil