

## 8. Übung "Java Persistence API (JPA)"

Verwenden Sie für die nachfolgende Übung die **Eclipse IDE für Java EE Developer**.

### 1. JPA-Projekt mit Eclipse Link

Erstellen Sie in Eclipse ein neues JPA-Projekt ( ohne EclipseLink-JPA-Implementierung und DB - Connection). Fügen Sie dem Java Build Path des Projektes die externen EclipseLink-Bibliotheken

eclipselink.jar

javax.persistence\_2.1.1.v201509150925.jar

sowie den Oracle-Driver

ojdbc8.jar

aus dem OPAL-Ordner der Übung8 hinzu.

Eine **Anleitung zur Erstellung und Konfiguration eines JPA-Projektes** in Eclipse finden Sie im Dokument AnleitungJPAmitEclipseLink.pdf im OPAL-Ordner der Übung 8.

### 2. JPA Konfigurationsdatei persistence.xml

Ergänzen Sie in der Konfigurationsdatei persistence.xml den Provider für EclipseLink sowie die Konfiguration für die DB-Connection zum Oracle-Server wie in der obigen Anleitung beschrieben.

### 3. JPA Entities für die Tabellen EMP und DEPT

- a. Erzeugen Sie im SQL-Developer zunächst zwei Sequenzen mit folgenden Eigenschaften

Sequenz	Startwert	Inkrementwert
emp_seq	8000	1
dept_seq	50	1

- b. Erstellen Sie anschließend im JPA-Projekt ein Package mit zwei JPA-Entity-Klassen für die Tabellen EMP und DEPT.

Erzeugen Sie die Entity-Klassen im Kontextmenue des Packages über New→JPA Entity.

Übernehmen Sie dabei alle Tabellenspalten in Attribute.

Legen Sie Key und Access Type (Field) fest!

#### Hinweise zu 3b:

- Erzeugen Sie **JPA Entity-Klassen** (anstelle von einfachen Java-Klassen), da bei deren Erstellung gleich einige Annotationen, Konstruktor sowie Getter und Setter angelegt werden.
  - Verwenden Sie bei der Eingabe der Tabellenspalten ausschließlich primitive Datentypen, also z.B. STRING anstelle von DATE, da die später zu benutzende find() -Methode nur auf primitiven Datentypen funktioniert.
- c. Die Primärschlüsselwerte der Entity-Objekte sollen über die unter a. erstellten Sequenzen erzeugt werden. Fügen Sie den entsprechenden Attributen der Klassen die erforderlichen Annotationen für die Generierung der Primärschlüsselwerte aus der zugehörigen Sequenz hinzu.
- d. Lassen Sie eine toString() -Methode für jede Entity-Klasse generieren.

- e. Integrieren Sie die zu persistierenden Klassen in die Persistence-Unit der Konfigurationsdatei persistence.xml

```
<persistence-unit name="Uebung8_JPA">
<provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
<class>package.Emp</class>
<class>package.Dept</class>
...
</persistence-unit>
```

wobei "package" durch den Packagebezeichner Ihres JPA-Projektes zu ersetzen ist.

#### 4. ManyToOne-Beziehung

Bilden Sie die N:1-Beziehung zwischen den Tabellen EMP und DEPT durch eine unidirektionale ManyToOne-Annotation in der Klasse EMP ab, indem Sie die DEPTNO der EMP-Klasse durch ein Attribut vom Objekttyp DEPT ersetzen. Passen Sie die toString() -Methode an!

#### 5. Testklasse

Erzeugen Sie anschließend eine Testklasse mit Entity Manager und Transaktionsverwaltung für das Entity-Management!

#### 6. Ausgabemethoden mittels JPQL

Programmieren Sie in der Testklasse für die Tabellen EMP und DEPT je eine Ausgabemethode, die mittels JPQL die Datensätze der Tabelle liest, als Objektinstanzen in einer Liste speichert und diese ausgibt.

In der Ausgabemethode für die Tabelle DEPT sollen außerdem zu jeder Abteilung alle zugehörigen Angestellten über eine parametrisierte Query gesucht und angezeigt werden.

Testen Sie die Methoden!

#### 7. ID-basiertes Daten-Retrieval

- Geben Sie die Daten des Angestellten 7369 aus.
- Geben Sie die Daten der Abteilung des Angestellten 7369 aus.
- Fügen Sie der Abteilung des Angestellten 7369 den nachfolgenden neuen Mitarbeiter hinzu:

ENAME	JOB	MGR	HIREDATE	SAL	COMM
Morrison	Developer	7788	06.08.2014	2600	0

Speichern Sie die neue Objektinstanz mit Hilfe des Entity Managers in der zugehörigen DB-Tabelle und überprüfen Sie das Ergebnis.

#### 8. Änderung von Entitäten

##### a. Update

Ändern Sie für den neuen Angestellten aus Aufgabe 7c die Abteilung auf 40 und das Gehalt auf 3200\$.

##### b. Delete

Geben Sie alle Angestellten mit Namen Morrison aus.

Löschen Sie diesen Datensatz/diese Datensätze!

Überprüfen Sie das Ergebnis mit Hilfe der Ausgabemethoden!

#### 9. Named Query

Mit Hilfe einer vordefinierten Named Query findAllEmployeesByManager sollen sämtliche Mitarbeiter eines Managers gesucht werden können. Die MGR des Managers soll dabei erst zur Laufzeit festgelegt werden.