

EntityManager API (JPA 1.0):

```
void persist(Object o); // Neuen Datensatz anlegen, wenn er noch nicht existiert

<T> T merge(T entity); // Existierenden Datensatz updaten, oder nicht exist. neu anlegen
// abhängig von DB Engine u. DB Treiber ist ein partieller Update
// mit detachten geänderten Daten machbar oder nicht

void remove(Object o); // Existierenden Datensatz löschen

<T> T find(Class<T> entityClass, Object primaryKey);
// Existierenden Datensatz per Primary Key finden,
// gibt im Fehlerfall (z.B. Exception) NULL zurück

Query createQuery(String qlString);
// Generiert aus dem JPA Query Language String eine Query

Query createNamedQuery(String name);
// Generiert eine NamedQuery mit dem name

Query createNativeQuery(String sqlString);
// Generiert aus dem SQL String eine native Query

Query createNativeQuery(String sqlString, Class resultClass);
// Generiert aus dem SQL String eine native Query für die resultClass

Query createNativeQuery(String sqlString, String resultSetMapping);
// Generiert aus dem SQL String die native Query mit resultSetMapping

void remove(Object o); // Existierenden Datensatz löschen

<T> T getReference(Class<T> entityClass, Object primaryKey);
// Existierenden Datensatz per Primary Key finden,
// generiert im Fehlerfall EntityNotFoundException

void lock(Object entity, LockModeType lockMode);
// Existierenden Datensatz locken

void flush(); // PersistenceContext (PC) in die Datenbank schreiben

void setFlushMode(FlushModeType flushMode);
// Den Flush-Modus setzen, d.h. die Art, wie geflusht wird
FlushModeType getFlushMode(); // Gibt den FlushModeType zurück, z.B. automatisch

void refresh(Object entity); // Aktualisierendes Einlesen eines Datensatzes in den PC

void clear(); // Löschen/Auslagern des PersistenceContexts (PC)

boolean contains(Object entity); // Liefert true, wenn die Entity im PC liegt, sonst false

Object getDelegate(); // Gibt aktuellen JPA Provider zurück, z.B. HibernateSessionFactory

void close(); // Schließen des aktuellen PersistenceContexts (PC)

EntityTransaction getTransaction();
// Gibt die aktuell laufende Transaktion zurück
```