

## TypedQuery vs Query

TypedQuery extends Query, it provides a way to handle query when type of object being returned is specified.

```
Query q = em.createQuery("select p from Product");
```

```
TypedQuery<Product> q = em.createQuery("select p from Product",  
Product.class);
```

If we used Query then we have to handle manually type casting all the things.

```
public List<Object> getAllProducts()  
{  
    String jpql = "SELECT p FROM Product p";  
    Query query = entityManager.createQuery(jpql);  
    return query.getResultList();  
}
```

Only queries allowed are select, delete, and update using entity manager. There is no insert jpql.

We can use

```
TypedQuery<Object[]> q = em.createQuery("select p.name, p.price from  
Product p", Object[].class);
```

```
for(Object[] ob -> q.getResultSet) {  
    sout(ob[0] + " " + ob[1]);  
}
```

// used when we want to skip some relationship

NoResultException is returned for getSingleResult if query returns no value.