

relationships, or map existing relationships to existing tables using the Jdeveloper modeling tools. With the modeling tools you can represent relationships as lines between entities, and change the relationships by changing the line configurations. For more information, see [Modeling with EJB Diagrams](#).

Java Persistence Query Language

Java Persistence Query Language (JPQL) offers a standard way to define relationships between entity beans and dependent classes by introducing abstract schema types and relationships in the deployment descriptor. JPQL also defines queries for navigation using abstract schema names and relationships.

The JPAQL query string consists of two mandatory clauses: SELECT and FROM, and an optional WHERE clause. For example:

```
select d from Departments d where d.department_name = ?1
```

There are two kinds of methods that use JPQL, **finder** methods and **select** methods.

- Finder methods are exposed to the client and return either a single instance, or a collection of entity bean instances.
- Select methods are not exposed to the client, they are used internally to return an instance of `cmp-field` type, or the remote interfaces represented by the `cmr-field`.

JPA Object-Relational Mappings

8

The Java Persistence API lets you declaratively map Java objects to relational database tables in a standard, portable way that works both inside a Java EE 5 application server and outside an EJB container. This approach greatly simplifies Java persistence and provides an **object-relational mapping approach**.

With Oracle TopLink you can configure the JPA behavior of your entities using metadata annotations in your Java source code. At run-time the code is compiled into the corresponding Java class files.

To designate a Java class as a JPA entity, use the `@Entity` annotation, as shown in the example below:

```
@Entity
public class Employee implements Serializable {
    ...
}
```

You can selectively add annotations to override defaults specified in your deployment descriptors.

For more information on JPA Annotations, see the *TopLink JPA Annotation Reference* at <http://www.oracle.com/technetwork/middleware/ias/toplink-jpa-annotations-096251.html>.

How to Use Java Service Facades

A Java service facade implements a lightweight testing environment you can run without an application server. With EJB 3.x, the Java service facade is similar to an