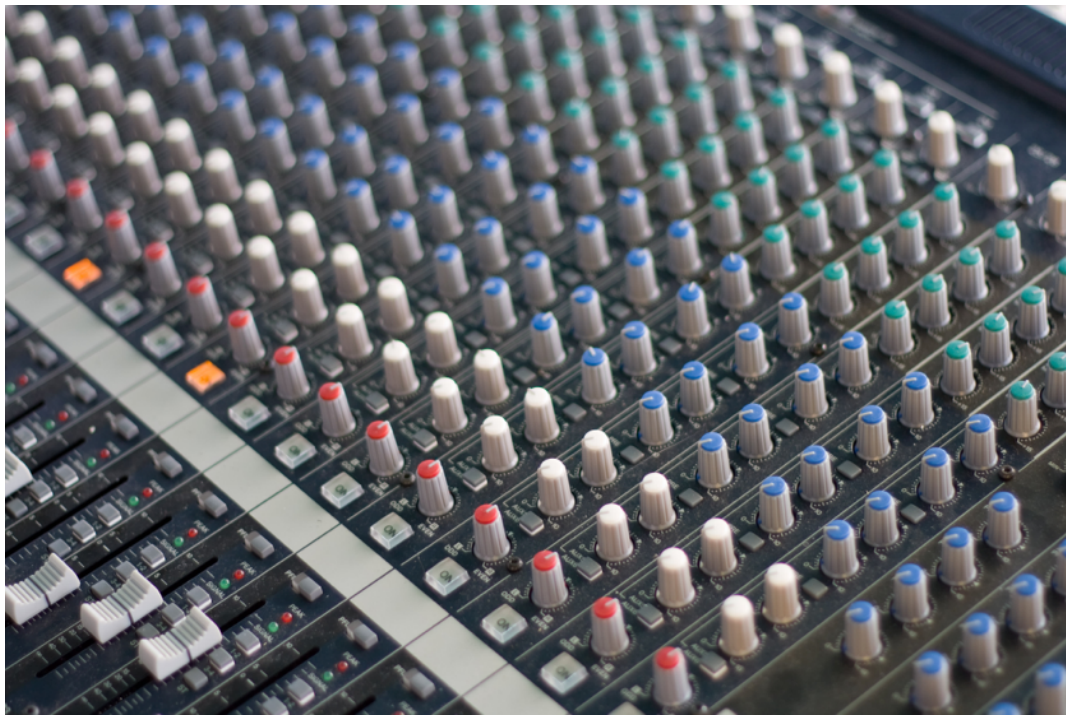# @Query Annotation

Reuse existing JPQL in your data access layer

Advanced query functionality

Eager loading control ("fetch")

```
@Query("select m from Model m where m.name = ?1")
List<Model> queryByName(String name);
```

@Query Options

Named Parameters

```
@Query("select m from Model m where m.name = :modelname")
List<Model> queryByName(@Param("modelname") String name);
```

Enhanced JPQL Syntax

```
@Query("select m from Model m where m.name like %?1")
List<Model> queryByName(String name);
```

@Query Options

Native Queries

```
@Query(value = "select * from Model where name = ?0", nativeQuery = true)
List<Model> queryByName(String name);
```

Modifiable Queries

```
@Modifying
@Query("update Model m set m.name = ?1")
int updateByName(String name);
```

JPA NamedQueries

App startup validates queries rather than at runtime

```java
@Entity
@NamedQuery(
  name="Model.namedFindAllModelsByType",
  query="select m from Model m where m.modelType.name = :name")
public class Model {…}
```
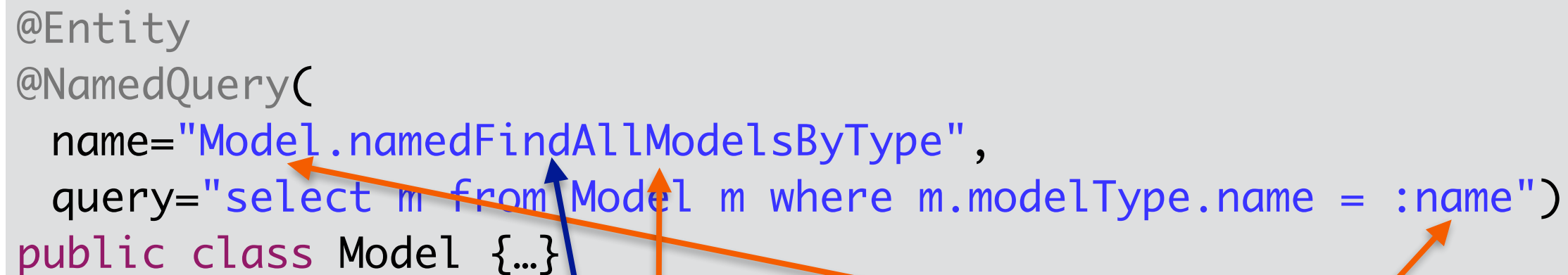
```java
@Repository
public interface ModelJpaRepository extends JpaRepository<Model, Long> {
  List<Model> namedFindAllModelsByType(@Param("name") String type);
}
```

```java
  @Query(name="Model.namedFindAllModelsByType")
  List<Model> findAllModelsByType(@Param("name") String type);
```

Native SQL Support

Native Queries

```java
@Query(value = "select * from Model where name = ?0", nativeQuery = true)
List<Model> queryByName(String name);
```

@NamedNativeQuery

```java
@NamedNativeQuery(name = "Manufacturer.getAllThatSellAcoustics",
query = "SELECT m.id, m.name, m.foundedDate, m.averageYearlySales, m.location_id as headquarters_id, m.active "
        + "FROM Manufacturer m "
        + "LEFT JOIN Model mod ON (m.id = mod.manufacturer_id) "
        + "LEFT JOIN ModelType mt ON (mt.id = mod.modeltype_id) "
        + "WHERE (mt.name = ?)", resultClass = Manufacturer.class)
public class Manufacturer { … }
```

Tip: Where Should I Put My Queries?

Entity
orm.xml

JpaRepository

NamedQuery
NamedNativeQuery

Query DSL
@Query

JpaRepository Query Precedence

Methods with @Query annotation take highest precedence

Methods that match a named or native named query "name"

Methods that follow the query DSL keyword naming structure

Summary

@Query annotation

JPA NamedQueries

SQL NamedNativeQueries