

# ToiletPaper #116

## SELECT n:m-tables with JPQL Projection

Author: Matthias Mair / Senior Software Architect / Business Division Automotive Bavaria

### ✘ Problem

JPA is a widely used interface for database access and object-relational mapping (ORM). Using the annotations `@ManyToMany` and `@JoinTable`, n:m relationships between database tables can be written comfortably, quickly and legibly. But only because of this description, the Join table does not yet have its own Java class, which means that all objects of the Join table's "left" part must first be retrieved before the "right" part can be aggregated using lazy loading. The information of the Join table can only be used after this step.

### ✓ Solution

Using the keyword **NEW** and specifying a constructor, JPQL can create one object per record for SELECT queries. This means that table columns are projected onto an object and opens the possibility to obtain the Join table with ONE database access.

### ➔ Example

A project has several employees and one employee can work in several projects.

```
1) @Entity
public class Project {
    @Id @Column(name = "ID")
    private long id;

    @ManyToMany @JoinTable(name = "PROJECT_EMPLOYEE",
        joinColumns = @JoinColumn(name = "PROJECT_ID", referencedColumnName = "ID"),
        inverseJoinColumns = @JoinColumn(name = "EMPLOYEE_ID", referencedColumnName = "ID"))
    private List<Employee> employees;
}
@Entity
public class Employee {
    @Id @Column(name = "ID")
    private long id;

    @ManyToMany(mappedBy = "employees")
    private List<Project> projects;
}

2) public class JoinTable {
    private long projectId, employeeId;

    public JoinTable(long projectId, long employeeId) {
        this.projectId = projectId; this.employeeId = employeeId;
    }
}
// Spring Data
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
@Repository
public interface ProjectRepository extends JpaRepository<Project, Long> {
    @Query("SELECT NEW com.jambit.toiletpaper.JoinTable(p.id, e.id) FROM Project p " +
        "JOIN p.employees e WHERE p.id = ?1")
    List<JoinTable> getAllJoinTableById(long id);
}
```

### + Further Aspects

- <https://thoughts-on-java.org/jpql/>