How Database tables are created in Spring JPA – 2025

For the below Entity,

@Table(name = "company")

@Setter @Getter @NoArgsConstructor

**public** **class** Company {

@Id

@GeneratedValue

**private** Long id;

**private** String name;

**public** Company(String name) {

**this**.name = name;

}

@OneToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***LAZY***)

**private** Set<Department> departments = **new** HashSet<>();

@OneToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***LAZY***)

**private** Set<Location> locations = **new** HashSet<>();

}

**Database Tables are created like below**

**create table company (**

**id bigint not null,**

**name varchar(255),**

**primary key (id)**

**)**

**create table company\_departments (**

**company\_id bigint not null,**

**departments\_id bigint not null,**

**primary key (company\_id, departments\_id)**

**)**

**create table company\_locations (**

**company\_id bigint not null,**

**locations\_id bigint not null,**

**primary key (company\_id, locations\_id)**

**)**

**Sequences are created based upon the Primary Entity Name**

**create sequence company\_seq start with 1 increment by 50**

**create sequence dept\_seq start with 1 increment by 50**

**create sequence emp\_seq start with 1 increment by 50**

**create sequence location\_seq start with 1 increment by 50**

**For the below Entity**

@Entity(name = "Department")

@Table(name = "dept")

@Getter @Setter @ToString @NoArgsConstructor

**public** **class** Department {

@Id

@GeneratedValue

**private** Long id;

**private** String name;

**public** Department(String name) {

**this**.name = name;

}

@OneToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***LAZY***)

**private** Set<Employee> employees = **new** HashSet<>();

}

**Database Tables created**

**create table dept (**

**id bigint not null,**

**name varchar(255),**

**primary key (id)**

**)**

**create table dept\_employees (**

**department\_id bigint not null,**

**employees\_id bigint not null,**

**primary key (department\_id, employees\_id)**

**)**

**For the below Entity**

@Entity(name="Employee")

@Table(name = "emp")

@Getter @Setter @ToString @NoArgsConstructor

**public** **class** Employee {

@Id @GeneratedValue

**private** Long id;

**private** String name;

**public** Employee(String name) {

**this**.name = name;

}

}

**Database Tables created**

**create table emp (**

**id bigint not null,**

**name varchar(255),**

**primary key (id)**

**)**