Spring AU '21 – Hibernate – Morning&Afternoon Session

Name: Sheik Abudhahir K

Date: 21/01/2021

Source code is also Uploaded

1 Employee [id, firstName LastName]
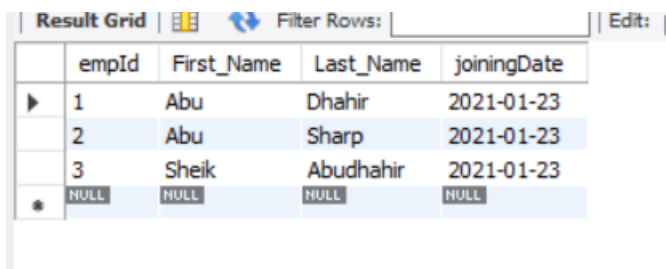
```java
@Entity
@Table(name="Emp") // for diff table name
@Setter
@Getter
public class Employee { //table create with the class name default
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int empId;

    @Column(name="First_Name", length=25)
    private String firstName;

    @Column(name="Last_Name") // if we need another name for columns use annotation
    private String lastName; // by default column take this name

    @Temporal(TemporalType.DATE)
    private Date joiningDate;
    |
```

Output in Mysql workbench

empId	First_Name	Last_Name	joiningDate
1	Abu	Dhahir	2021-01-23
2	Abu	Sharp	2021-01-23
3	Sheik	Abudhahir	2021-01-23
NULL	NULL	NULL	NULL

Employee db

2. OneToOneManpping with EmployeeDetails[phone and email]

No new table is created.

```java
@Entity
@Getter
@Setter
public class PersonalInfo {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int pId;

    private String phone;

    private String email;

    @OneToOne(cascade = CascadeType.ALL)
    private Employee employee;

    public PersonalInfo() {
        super();
    }
}
```

Output: Employee key added as foreign key in personal info table.

pId	email	phone	employee_empId
1	sheik@gmail.com	9080512621	1
2	abu@gmail.com	9994648368	2
3	sharp@gmail.com	9585112121	3
NULL	NULL	NULL	NULL

Employee Table:

empId	First_Name	Last_Name	joiningDate
1	Abu	Dhahir	2021-01-23
2	Abu	Sharp	2021-01-23
3	Sheik	Abudhahir	2021-01-23
NULL	NULL	NULL	NULL

3. OneToMany with Address [without Jointable] [Address has streetName,city,pincode]

Address entity

```java
public class Address {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int Aid;

    private String streetName;

    private String city;

    private String pincode;

    @ManyToOne(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
    private Employee employee = new Employee();
```

One to Many mapping in employee without jointable (using mappedBy)

```java
    @OneToMany(cascade = CascadeType.ALL, mappedBy = "employee")
    private Collection<Address> address = new HashSet<>();
```

Output: Address Table – Employee id is foreign key in address table

Aid	city	pincode	streetName	employee_empId
1	Chennai	600025	72, Besant Nage	1
2	Chennai	600025	183, Anna Nagar	1
3	Chennai	600026	76, Triplicane	2
4	Chennai	600026	12, Rayapuram	2
5	Chennai	600028	14, Adayar	3
NULL	NULL	NULL	NULL	NULL

Employee table:

empId	First_Name	Last_Name	joiningDate
1	Abu	Dhahir	2021-01-23
2	Abu	Sharp	2021-01-23
3	Sheik	Abudhahir	2021-01-23
NULL	NULL	NULL	NULL

4. ManyTomany[Courses] Only 1 Join Table. [Course has CourseName]

Here I mentioned sessions instead of courses

Here I add list of employees mapped with list of session mentioned in employee entity

```
@Entity
@Getter
@Setter
public class HSessions {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String sessionName;

    @ManyToMany(mappedBy = "sessions")
    private List<Employee> employee = new ArrayList<>();
```

In employee entity, I created collection of sessions and used the variable to map in sessions entity

```
0
1    @ManyToMany(cascade = CascadeType.ALL)
2    @JoinTable(name="emp_session")
3    private Collection<HSessions> sessions = new HashSet<>();
4
5
```

Output: Here I give alternate name for joint table (emp_session) only one join table

employee_empId	sessions_id
1	1
1	2
2	3
2	2

Session table

id	sessionName
1	Hibernate
2	Spring
3	HTML
NULL	NULL

Employee table

empId	First_Name	Last_Name	joiningDate
1	Abu	Dhahir	2021-01-23
2	Abu	Sharp	2021-01-23
3	Sheik	Abudhahir	2021-01-23
NULL	NULL	NULL	NULL