

Date : 06/02/2021  
Spring Boot 9AM  
Mr. RAGHU

-----  
Data JPA : Association Mapping  
One-To-One (1...1)

=> Parent table one row is connected to Child table one row (at max).  
1...1

ex: Passenger -----<> BusTicket

=>\*\*\* Use Many-To-One and apply unique Condition (unique=true)

1...1 = (unique) \*...1

Q) Why cant we take 1...\* and apply unique for 1...1?

A) YES. You can take.

1...1 = Non-Collection

\*...1 = Non-Collection

1...\* = Collection. So, we should use List that holds one value.  
Becomes memory waste.

-----  
\*) SQL Joins : (TASK)

Equal/Equy Join

Cross Join

Self Join

Inner Join

Outer Joins

Left Join

Right Join

Full Join

-----code-----1...1

1.Model

```
package in.nareshit.raghu.model;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
@Data
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@Entity
```

```
public class BusTicket {
```

```
    @Id
```

```
    private Integer btid;
```

```
    private String seatNo;
```

```

        private String bfrom;
        private String bto;
        private Double bcost;
    }

```

-----

```

package in.nareshit.raghu.model;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
public class Passenger {
    @Id
    private Integer pid;
    private String pname;
    private String gender;
    private Integer age;

    @ManyToOne //1...1
    @JoinColumn(name="tckt_id_fk",unique = true)
    private BusTicket ticket;
}

```

## 2. Repo

```

package in.nareshit.raghu.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import in.nareshit.raghu.model.BusTicket;

public interface BusTicketRepo
    extends JpaRepository<BusTicket, Integer> {

}

-----
package in.nareshit.raghu.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import in.nareshit.raghu.model.Passenger;

public interface PassengerRepo
    extends JpaRepository<Passenger, Integer> {

}

```

### 3.Runner

```
package in.nareshit.raghu.runner;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

import in.nareshit.raghu.model.BusTicket;
import in.nareshit.raghu.model.Passenger;
import in.nareshit.raghu.repo.BusTicketRepo;
import in.nareshit.raghu.repo.PassengerRepo;

@Component
public class DataInsertRunner implements CommandLineRunner {
    @Autowired
    private BusTicketRepo brepo;

    @Autowired
    private PassengerRepo prepo;

    @Override
    public void run(String... args) throws Exception {
        BusTicket b1 = new BusTicket(110, "A01", "HYD", "CHN",
500.0);
        BusTicket b2 = new BusTicket(111, "A02", "HYD", "DHL",
600.0);
        BusTicket b3 = new BusTicket(112, "A03", "HYD", "MUM",
700.0);

        brepo.save(b1);
        brepo.save(b2);
        brepo.save(b3);

        Passenger p1 = new Passenger(85, "ABD", "M", 50, b1);
        Passenger p2 = new Passenger(86, "MNO", "F", 36, b2);
        Passenger p3 = new Passenger(87, "PQR", "M", 28, b3);

        prepo.save(p1);
        prepo.save(p2);
        prepo.save(p3);
    }
}
```

### 4. yaml

```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/boot9am
    username: root
    password: root
  jpa:
    show-sql: true
    hibernate:
```

```
ddl-auto: create
database-platform: org.hibernate.dialect.MySQL8Dialect
```