

## 1 : 1 단방향 식별관계, 대상테이블에 외래키 실습

STS -> Spring Stater Project , name : onetoone-3

SQL : JPA, MySQL 선택

[http://ojc.asia/bbs/board.php?bo\\_table=LecSpring&wr\\_id=524](http://ojc.asia/bbs/board.php?bo_table=LecSpring&wr_id=524)

(마리아 DB 설치하는 위 URL에서 참조)

### src/main/resources/application.properties

```
spring.datasource.platform=mysql
spring.datasource.url=jdbc:mysql://localhost/onetoone_3?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=1111
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.sql-script-encoding=UTF-8
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
```

### demo.model.Emp.java

```
@Entity
public class Emp {
    @Id
    @GeneratedValue
    private Long empno;
    private String ename;
    public Emp(String ename) {
        this.ename = ename;
    }

    public Long getEmpno() {
        return empno;
    }

    public void setEmpno(Long empno) {
        this.empno = empno;
    }

    public String getEname() {
        return ename;
    }
}
```

```

    }
    public void setName(String ename) {
        this.ename = ename;
    }
    @Override
    public String toString() {
        return String.format(
            "Emp[empno=%d, ename='%s']",
            empno, ename);
    }
}

```

#### demo.model.Addr.java

```

package demo.model;
@Entity
public class Addr {
    @Id private Long id;
    private String address;

    //Addr 테이블에 empno 칼럼이 키로 생성
    @OneToOne(cascade=CascadeType.ALL)
    @MapsId
    @JoinColumn(name="empno")
    private Emp emp;
    public Addr(String address, Emp emp) {
        this.address = address; this.emp = emp;
    }
    public Emp getEmp() { return emp; }
    public void setEmp(Emp emp) {
        this.emp = emp;
    }
    public Addr(String address) {
        this.address = address; }
    public String getAddress() {
        return address; }
    public void setAddress(String address) {
        this.address = address;
    }
}

```

#### **demo.repository.EmpRepository.java**

```
package demo.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import demo.model.Emp;
public interface EmpRepository extends JpaRepository<Emp, Long> { }
```

#### **demo.repository.AddrRepository.java**

```
package demo.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import demo.model.Addr;
public interface AddrRepository extends JpaRepository<Addr, Long>{ }
```

#### **demo.Onetoone3Application.java**

```
@SpringBootApplication
public class Onetoone3Application implements CommandLineRunner {
    public static void main(String[] args) {
        SpringApplication.run(Onetoone3Application.class,
args);
    }

    @Autowired
    EmpRepository empRepository;

    @Autowired
    AddrRepository addrRepository;

    @Transactional
    @Override
    public void run(String...args) {
        List<Emp> emps = new ArrayList();

        emps.add(new Emp("김길동"));
        emps.add(new Emp("나길동"));
        emps.add(new Emp("다길동"));

        empRepository.save(emps);

        //ALL Emp Display
    }
}
```

```

        for(Emp e : empRepository.findAll()) {
            System.out.println(e.toString());
        }

        Addr addr1 = new Addr("대전");
        Emp e1 = new Emp("대전길동");
        addr1.setEmp(e1);

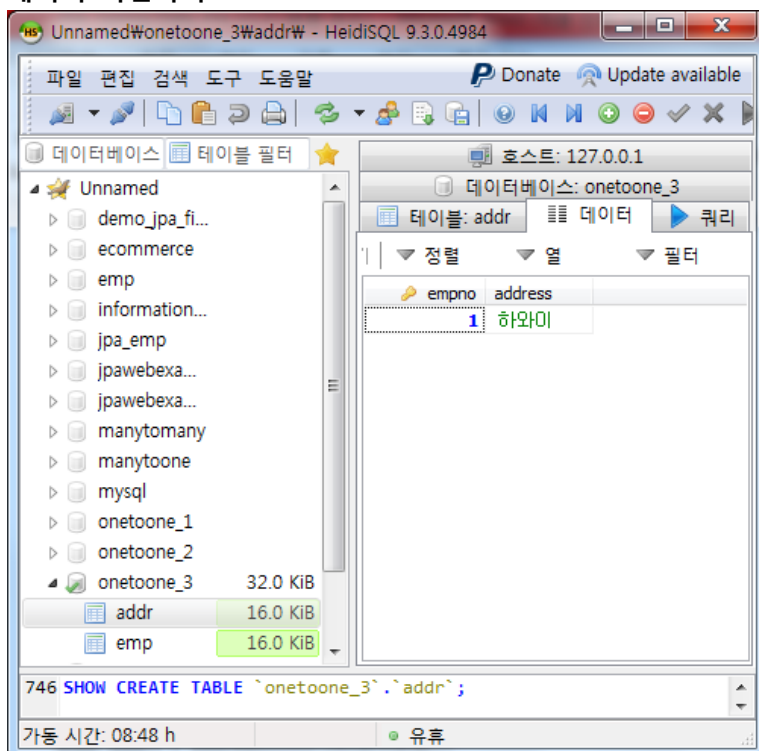
        Addr addr2 = new Addr("하와이");
        addr2.setEmp(empRepository.get(0));

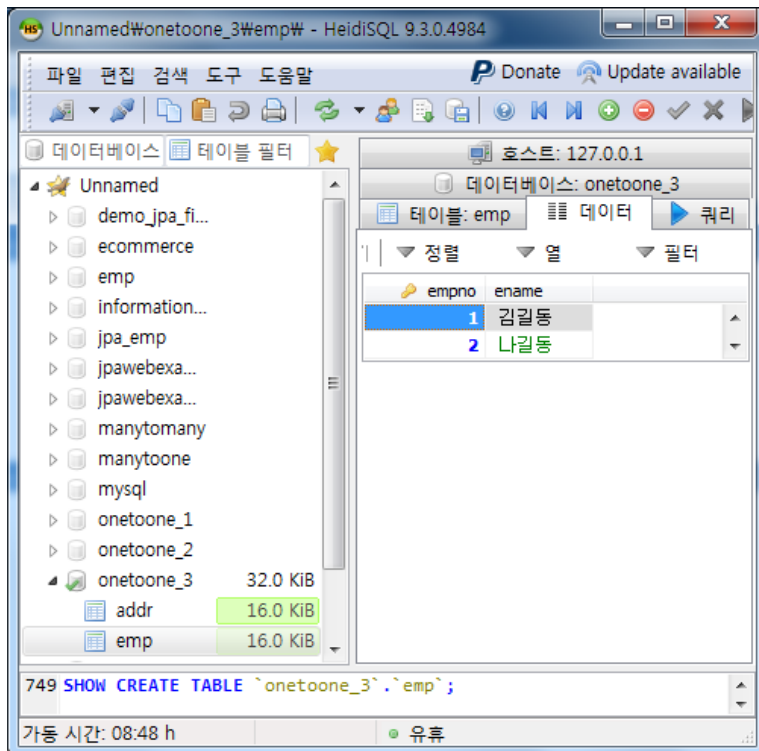
        empRepository.save(addr1);
        empRepository.save(addr2);

        //Addr에서 4번 "대전"을 삭제
        //CascadeType.ALL 때문에 4번 "대전길동" 사원도 삭제됨
        Addr addr3 = empRepository.findOne(4L);
        empRepository.delete(addr3);
    }
}

```

## 데이터 확인하기





### [실행 결과]

Hibernate: alter table addr drop foreign key FK\_nqlgci8i855oq6mcrad87w5ks

Hibernate: drop table if exists addr

Hibernate: drop table if exists emp

Hibernate: create table addr (empno bigint not null, address varchar(255), primary key (empno))

Hibernate: create table emp (empno bigint not null auto\_increment, ename varchar(255), primary key (empno))

Hibernate: alter table addr add constraint FK\_nqlgci8i855oq6mcrad87w5ks foreign key (empno) references emp (empno)

Hibernate: insert into emp (ename) values (?)

Hibernate: insert into emp (ename) values (?)

Hibernate: insert into emp (ename) values (?)

Hibernate: select emp0\_.empno as empno1\_1\_, emp0\_.ename as ename2\_1\_ from emp emp0\_

//메인 출력

Emp[empno=1, ename='김길동']

Emp[empno=2, ename='나길동']

Emp[empno=3, ename='다길동']

Hibernate: insert into emp (ename) values (?)

Hibernate: insert into addr (address, empno) values (?, ?)

Hibernate: insert into addr (address, empno) values (?, ?)

Hibernate: delete from addr where empno=?

Hibernate: delete from emp where empno=?