

N대1 양방향 실습

- 스프링 부트, 마리아DB를 사용하여 N:1 관계 양방향 예제를 작성해보자.

http://ojc.asia/bbs/board.php?bo_table=LecSpring&wr_id=524

(마리아 DB 설치는 위 URL에서 참조)

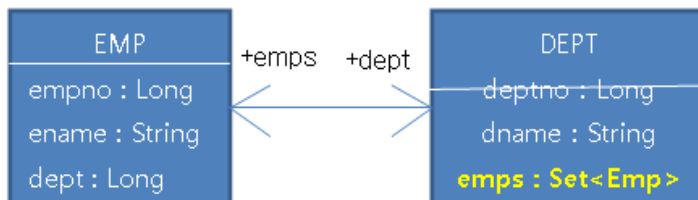
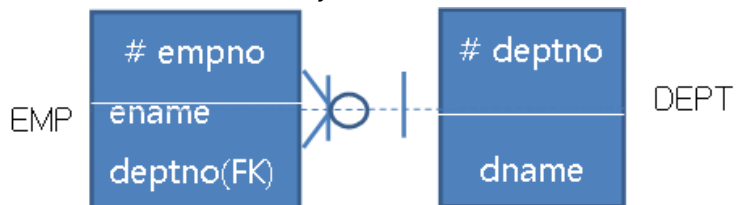
STS -> Spring Starter Project

Name : manytoone

Type : Maven

Package : demo

다음 창에서 SQL -> JPA, MySQL 선택



application.properties

```
spring.datasource.platform=mysql
spring.datasource.url=jdbc:mysql://localhost/manytoone?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
```

Dept.java(도메인 오브젝트)

```

package demo.model;

@Entity
@Table(name = "dept")
public class Dept {
    private Long deptno;
    private String dname;
    private Set<Emp> emps;
    public Dept() { }
    public Dept(String dname) {
        this.dname = dname;
    }
    public Dept(String dname, Set<Emp> emps) {
        this.dname = dname;
        this.emps = emps;
    }
    @Id
    @GeneratedValue
    public Long getDeptno() { return deptno; }
    public void setDeptno(Long deptno) {
        this.deptno = deptno;
    }

    public String getDname() { return dname; }
    public void setDname(String dname) {
        this.dname = dname;
    }
}

//Emp 엔티티의 dept 속성(필드)를 매핑
//mappedBy에서 반대쪽(many쪽),Owner가 되는 쪽의 매핑되는 속성지정
@OneToMany(mappedBy = "dept", cascade = CascadeType.ALL)
public Set<Emp> getEmps() { return emps; }
public void setEmps(Set<Emp> emps) {
    this.emps = emps; }
public String toString() {
    String s = String.format("DEPT[deptno = %d, dname = '%s']\n", deptno, dname);
    if (emps != null) {
        for(Emp e : emps) {
            s += String.format("EMP[empno = %d, ename = '%s', deptno = '%s']\n",
                                e.getEmpno(), e.getEname(),
                                e.getDept()==null?"":e.getDept().deptno);
        }
    }
    return s;
}

```

```

    }
    }
    return s;
}
}

```

만약 위 예문에서 List를 이용한다면 아래와 같다.

@OneToMany(mappedBy="dept", cascade=CascadeType.ALL)

@OrderBy("dname DESC")

@OrderColumn(name="emp_index")

private **List**<Emp> emps;

➔ EMP테이블에 emp_index 칼럼이 자동 생성된다.

Emp.java

```

package demo.model;

@Entity
@Table(name = "emp")
public class Emp {
    private Long empno;
    private String ename;
    private Dept dept;

    public Emp() {}
    public Emp(String ename) {
        this.ename = ename;
    }
    public Emp(String ename, Dept dept) {
        this.ename = ename;
        this.dept = dept;
    }
    @Id
    @GeneratedValue
    public Long getEmpno() {
        return empno;
    }
}

```

```

public void setEmpno(Long empno) {
    this.empno = empno;
}
public String getEname() {
    return ename;
}
public void setEname(String ename) {
    this.ename = ename;
}
@ManyToOne
//deptno는 현재 엔티티 테이블의 조인칼럼명
//생략시 조인칼럼명은 dept_deptno가 된다.
@JoinColumn(name="deptno")
public Dept getDept() {
    return dept;
}
public void setDept(Dept dept) {
    this.dept = dept;
}
public String toString() {
    String s = String.format("EMP[empno = %d, ename = '%s', deptno=%d]%n", empno,
    ename, getDept().getDeptno());
    return s;
}
}

```

DeptRepository.java(레포지토리 클래스)

```

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

```

ManyTooneApplication.java(스프링부트 메인)

```

package demo;

import java.util.HashSet;

```

```

import java.util.Set;

import javax.transaction.Transactional;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import demo.model.Dept;
import demo.model.Emp;
import demo.repository.DeptRepository;

```

@SpringBootApplication

```

public class ManytooneApplication implements CommandLineRunner {
    private static final Logger logger = LoggerFactory.getLogger(ManytooneApplication.class);

    @Autowired
    private DeptRepository deptRepository;

    public static void main(String[] args) {
        SpringApplication.run(ManytooneApplication.class, args);
    }
}

```

//아래에서 @Transactional이 생략되면 org.hibernate.LazyInitializationException: failed to lazily initialize a collection of role: demo.model.Dept.emps, could not initialize proxy 오류발생

@Override

@Transactional

```

public void run(String...strings) throws Exception {
    //Dept에 1번부서가 만들어지고, Emp의 deptno는 1로 입력됨
    Dept d1 = new Dept("교육팀");
    Emp e1 = new Emp("김교육", d1);
    Emp e2 = new Emp("나교육", d1);

    d1.setEmps(new HashSet<Emp>() {
        {
            add(e1);

```

```

        add(e2);
    }
});

//Dept에 2번 부서가 만들어지고 Emp의 deptno는 NULL로 입력
Dept d2 = new Dept("개발팀", new HashSet<Emp>() {
    {
        add(new Emp("김개발"));
        add(new Emp("나개발"));
    }
});

//Dept, Emp 저장
deptRepository.save(new HashSet<Dept>() {
    {
        add(d1);
        add(d2);
    }
});

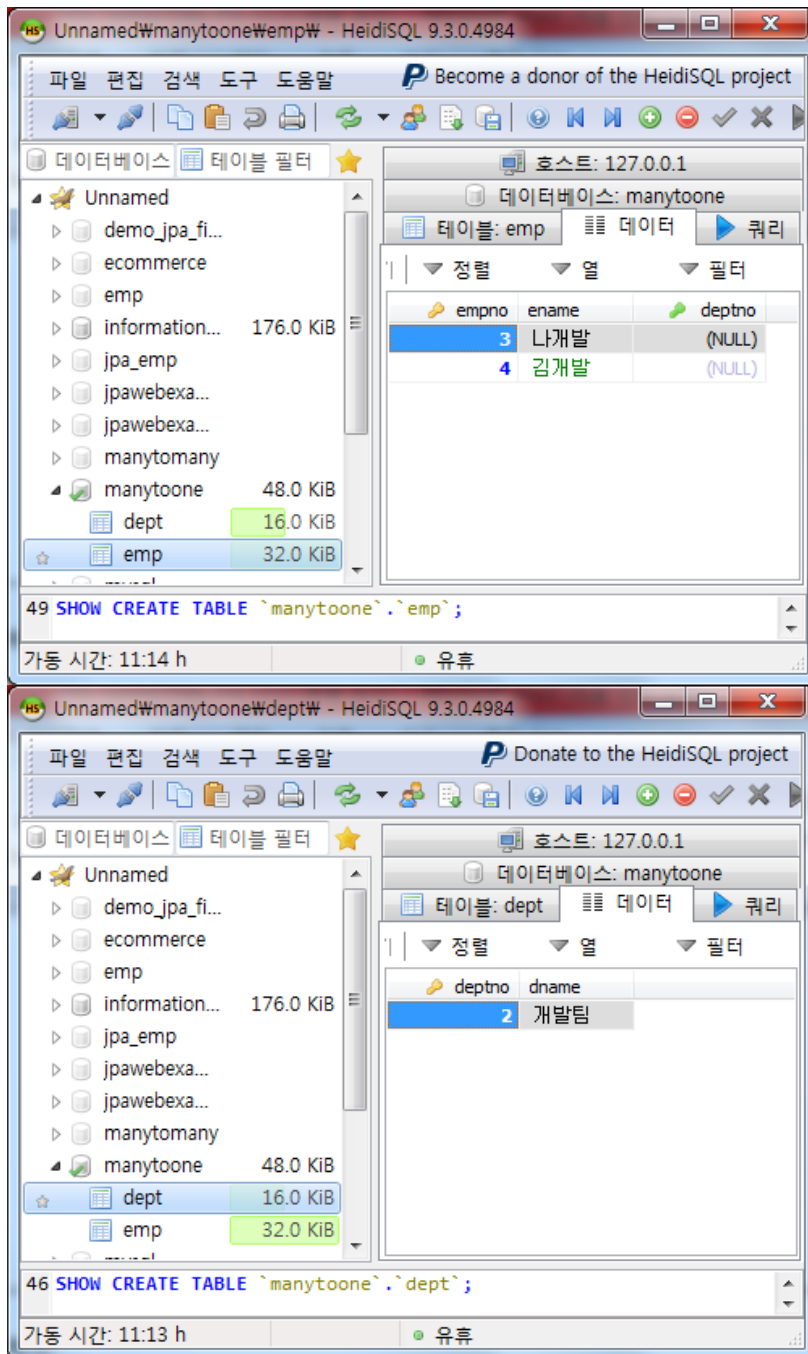
for(Dept d : deptRepository.findAll()) {
    logger.info(d.toString());
}

//----- 1번 부서 및 부서원 로드
Dept d3 = deptRepository.findOne(1L);
Set<Emp> emps = d3.getEmps();
for(Emp e : emps) {
    logger.info(e.toString());
}

//----- 1번부서 삭제, 1번부서원들도 같이 삭제된다.
deptRepository.delete(d3);
}
}

```

데이터 확인하기



[DB쪽 생성 스크립트 및 결과]

Hibernate: alter table emp drop foreign key FK_gbxI70x5ckxun8hi19v4n6dfb

Hibernate: drop table if exists dept

Hibernate: drop table if exists emp

Hibernate: create table dept (deptno bigint not null auto_increment, dname varchar(255), primary key (deptno))

Hibernate: create table emp (empno bigint not null auto_increment, ename varchar(255), deptno

```
bigint, primary key (empno))
Hibernate: alter table emp add constraint FK_gbxI70x5ckxun8hi19v4n6dfb foreign key (deptno)
references dept (deptno)
Hibernate: insert into dept (dname) values (?)
Hibernate: insert into emp (deptno, ename) values (?, ?)
Hibernate: insert into emp (deptno, ename) values (?, ?)
Hibernate: insert into dept (dname) values (?)
Hibernate: insert into emp (deptno, ename) values (?, ?)
Hibernate: insert into emp (deptno, ename) values (?, ?)
Hibernate: select dept0_.deptno as deptno1_0_, dept0_.dname as dname2_0_ from dept dept0_
```

//메인 출력

```
DEPT[deptno = 1, dname = '교육팀']
EMP[empno = 1, ename = '김교육', deptno = '1']
EMP[empno = 2, ename = '나교육', deptno = '1']
```

```
DEPT[deptno = 2, dname = '개발팀']
EMP[empno = 3, ename = '나개발', deptno = '']
EMP[empno = 4, ename = '김개발', deptno = '']
```

```
EMP[empno = 1, ename = '김교육', deptno=1]
EMP[empno = 2, ename = '나교육', deptno=1]
```

```
Hibernate: delete from emp where empno=?
Hibernate: delete from emp where empno=?
Hibernate: delete from dept where deptno=?
```