1대1 양방향, 주테이블에 외래키 실습

STS -> Spring Stater Project, name : onetoone-2

SQL : JPA, MySQL 선택

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

(마리아 DB 설치는 위 URL에서 참조)

**application.properties**

spring.datasource.platform=mysql

spring.datasource.url=jdbc:mysql://localhost**/onetoone\_2**?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;

**@OneToOne(cascade = CascadeType.ALL)**

**@JoinColumn(name = "addr\_id")**

private Addr addr;

public Emp(String ename, Addr addr) {

this.ename = ename; this.addr = addr;

}

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;

}

public Addr getAddr() {

return addr;

}

public void setAddr(Addr addr) {

this.addr = addr;

}

public String toString() {

return String.format(

"Emp[empno=%d, ename='%s', address='%s']",

empno, ename, addr.getAddress());

}

}

**demo.model.Addr.java**

@Entity **//양방향인경우 CascadeType.ALL**

public class Addr {

@Id @GeneratedValue

private Long id; private String address;

**@OneToOne(cascade=CascadeType.ALL, mappedBy="addr")**

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 Long getId() {return id; }

public void setId(Long id) { this.id = id;}

public String getAddress() { return address;}

public void setAddress(String address) { this.address = address; }

public String toString() {

return String.format(

"Addr[id=%d, address='%s', ename='%s']",

id, address, emp.getEname());

}

}

**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.Onetoone2Application.java**

@SpringBootApplication

public class Onetoone2Application implements CommandLineRunner {

public static void main(String[] args) {

SpringApplication.run(Onetoone2Application.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", new Addr("서울")));

emps.add(new Emp("나길동", new Addr("제주")));

emps.add(new Emp("다길동", new Addr("뉴욕")));

**//Emp가 Owner이고 addr 필드값이 설정되므로**

**//Emp 테이블에서 외래키 필드에 값이 채워진다.**

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");

**//Emp쪽이 Owning Side(주인), 외래키가 있다.**

**//아래코드가 빠지면 대전길동의 addr\_id는 NULL이된다.**

e1.setAddr(addr1);

**//양방향 관계이므로 아래 코드가 빠지면 데이터는 한 건도 저장되지 않는다.**

**//insert into addr(address) values (?),insert into emp(addr\_id, ename) values (?, ?)**

addr1.setEmp(e1);

empRepository.save(e1);

Addr addr2 = new Addr("하와이");

Emp e2 = new Emp("하와이길동");

**//아래코드는 연관관계 때문에 두 테이블에 데이터가 저장되지만**

**//Emp쪽의 외래키인 addr이 설정되지 않아**

**//즉 e2.setAddr(addr2) 코드가 없어 "하와이길동"의 addr\_id는 NULL이 된다.**

**//insert into addr (address)values (?),insert into emp (addr\_id, ename)values (?, ?)**

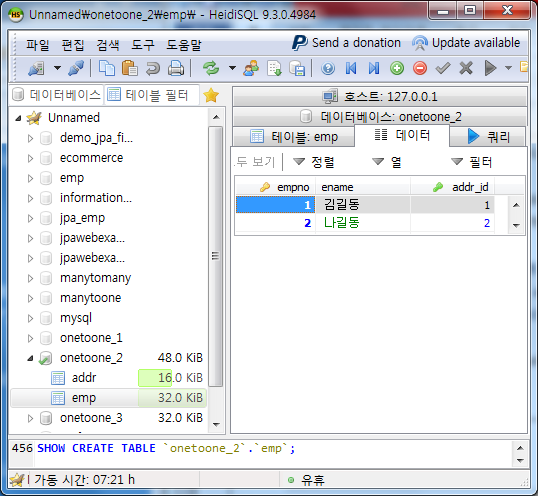
addr1.setEmp(e2);

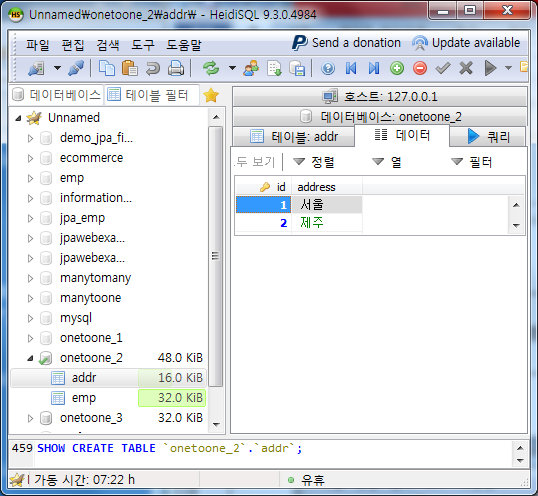
addrRepository.save(addr2);

}

}

**데이터 확인하기**





**[실행 결과]**

Hibernate: alter table emp drop foreign key FK\_b1bolhhce7t698wamy15o3j47

Hibernate: drop table if exists addr

Hibernate: drop table if exists emp

Hibernate: create table addr (id bigint not null auto\_increment, address varchar(255), primary key (id))

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

Hibernate: alter table emp add constraint FK\_b1bolhhce7t698wamy15o3j47 foreign key (addr\_id) references addr (id)

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

Hibernate: insert into emp (addr\_id, ename) values (?, ?)

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

Hibernate: insert into emp (addr\_id, ename) values (?, ?)

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

Hibernate: insert into emp (addr\_id, ename) values (?, ?)

Hibernate: select emp0\_.empno as empno1\_1\_, emp0\_.addr\_id as addr\_id3\_1\_, emp0\_.ename as ename2\_1\_ from emp emp0\_

**//메인 출력**

Emp[empno=1, ename='김길동', address='서울']

Emp[empno=2, ename='나길동', address='제주']

Emp[empno=3, ename='다길동', address='뉴욕']

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

Hibernate: insert into emp (addr\_id, ename) values (?, ?)

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

Hibernate: insert into emp (addr\_id, ename) values (?, ?)