



.

- : ,
- : ,

:

mapping compositekey primary and non foreign

## DDL

: itemno + itemtype .

### ItemInfo

```
CREATE TABLE `db_example2`.`iteminfo` (  
  `itemtype` CHAR(1) NOT NULL,  
  `itemno` INT NOT NULL,  
  `itemname` VARCHAR(45) NULL,  
  PRIMARY KEY (`itemtype`, `itemno`));
```

### ItemStatics

```
CREATE TABLE `itemstatics` (  
  `itemtype` char(1) NOT NULL,  
  `itemno` int(11) NOT NULL,  
  `viewcnt` int(11) DEFAULT NULL,  
  `salecnt` int(11) DEFAULT NULL,  
  `statictype` varchar(45) NOT NULL,  
  PRIMARY KEY (`itemtype`,`itemno`,`statictype`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```



ItemStatic itemtype itemno ,3 .

.

JPA ,

ItemId(itemtype,itemno) , .

JPA

.

## Entity

```

import javax.persistence.Column;
import javax.persistence.Embeddable;

@Embeddable
public class ItemId implements Serializable {

    @Column(columnDefinition="char(1)") // Type char(1) -> string
    private String      itemtype;
    private int         itemno;

    public ItemId() {

    }

    public ItemId(String itemtype, int itemno) {
        this.itemtype = itemtype;
        this.itemno = itemno;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = itemtype.hashCode() + itemno;
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        ItemId other = (ItemId) obj;
        if (itemtype == null) {
            if (other.itemtype != null)
                return false;
        } else if (!itemtype.equals(other.itemtype))
            return false;
        if (itemno != other.itemno)
            return false;
        return true;
    }

    public String getItemtype() {
        return itemtype;
    }

    public void setItemtype(String itemtype) {
        this.itemtype = itemtype;
    }

    public int getItemno() {
        return itemno;
    }

    public void setItemno(int itemno) {
        this.itemno = itemno;
    }

}

```

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

```
@Entity
public class ItemStatics {
    @EmbeddedId
    ItemId staticid;

    int viewcnt;
}
```

```
import javax.persistence.Column;
import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.JoinColumn;
import javax.persistence.JoinColumns;
import javax.persistence.ManyToOne;
import javax.persistence.MapId;
```

```
@Entity
public class ItemInfo {

    @EmbeddedId
    ItemId itemid;

    String itemname;

    @MapId("staticid") //references EmbeddedId's property
    @JoinColumns({
        @JoinColumn( insertable = false, updatable = false,
            name = "itemtype",
            referencedColumnName = "itemtype"),
        @JoinColumn( insertable = false, updatable = false,
            name = "itemno",
            referencedColumnName = "itemno")
    })
    @ManyToOne
    private ItemStatics statics;
}
```

( )

```
@Autowired
ItemInfoRepository itemRepo;

public void allTest() {
    itemRepo.findAll().forEach(item -> {
        String itemString = String.format("%s %d",item.getItemname() ,item.getStatics().
getViewcnt() );
        System.out.println(itemString);
    });
}
```

( ),

.

>

ItemInfo				ItemStatics					
	itemtype	itemno	itemname		itemtype	itemno	viewcnt	salecnt	statictype
	C	1	T1		C	1	100	100	A
	C	2	T2		C	2	200	200	A
	C	3	T3		C	3	200	200	C
	NULL	NULL	NULL		NULL	NULL	NULL	NULL	NULL

>

T1 100  
T2 200  
T3 200