CS544 EA
Hibernate

Collection: List

List

- Bag with the ability to keep an arbitrary order
  - **Built-in order** (not based on properties): no @OrderBy
  - List can **contain duplicates**

- A shopping list is a typical example
  - Built-in although often arbitrary order
  - May contain duplicates

List Implementation

- Java has the java.util.List interface
  - Java.util.ArrayList most common implementation

- Lists are indexed collections
  - Needs FK and an **additional indexed sequence column**
  - Same problem with Bi-Direct as additional column for map!

FK column

Additional column
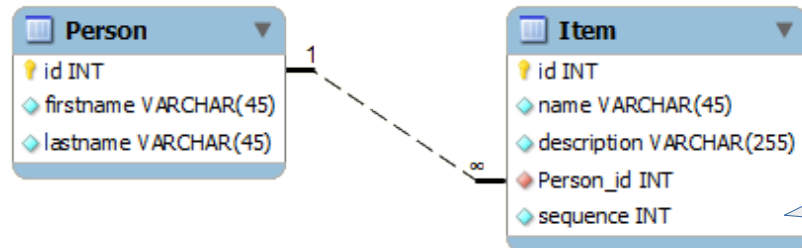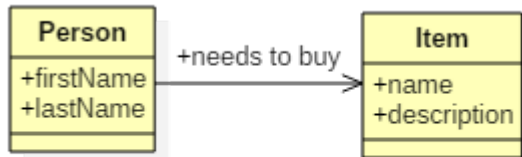
3
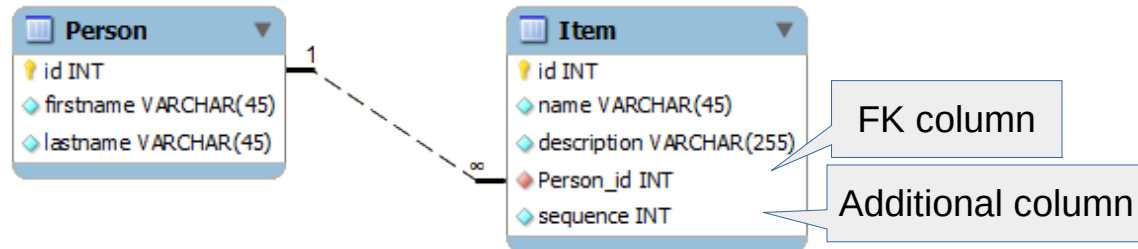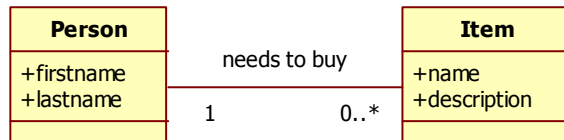
Code List Uni-Direct

- **@OrderColumn** for additional indexed column
  - Is what makes it a 'real' list instead of bag

```
@Entity
public class Person {
        @Id
        @GeneratedValue
        private Long id;
        private String firstname;
        private String lastname;
        @OneToMany
        @JoinColumn(name="Person_id")
        @OrderColumn(name="sequence")
        private List<Item> items
                = new ArrayList<>();
```

```
@Entity
public class Item {
        @Id
        @GeneratedValue
        private Long id;
        private String name;
        private String description;
```

@JoinColumn to avoid join table!

Optional name for order column

Person
+firstName
+lastName

+needs to buy →

Item
+name
+description

Person ▼
🔑 id INT
◇ firstname VARCHAR(45)
◇ lastname VARCHAR(45)

1

Item ▼
🔑 id INT
◇ name VARCHAR(45)
◇ description VARCHAR(255)
🔴 Person_id INT
◇ sequence INT

∞

Items_ORDER is default

4

Code List Bi-Direct

- **Emulation of mappedBy**: insertable, updateable

```
@Entity
public class Person {
    @Id
    @GeneratedValue
    private Long id;
    private String firstname;
    private String lastname;
    @OneToMany
    @JoinColumn(name="Person_id")
    @OrderColumn(name="sequence")
    private List<Item> items
        = new ArrayList<>();
```

```
@Entity
public class Item {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private String description;
    @ManyToOne
    @JoinColumn(name="Person_id", insertable=false, updatable=false)
    private Person person;
```

Same as uni-direct

Using @JoinColumn insertable, updatable to give up control of association

Person
+firstName
+lastName

+needs to buy

Item
+name
+description

Person
- id INT
- firstname VARCHAR(45)
- lastname VARCHAR(45)

1

∞

Item
- id INT
- name VARCHAR(45)
- description VARCHAR(255)
- Person_id INT
- sequence INT

5