CS544 EA
# Hibernate

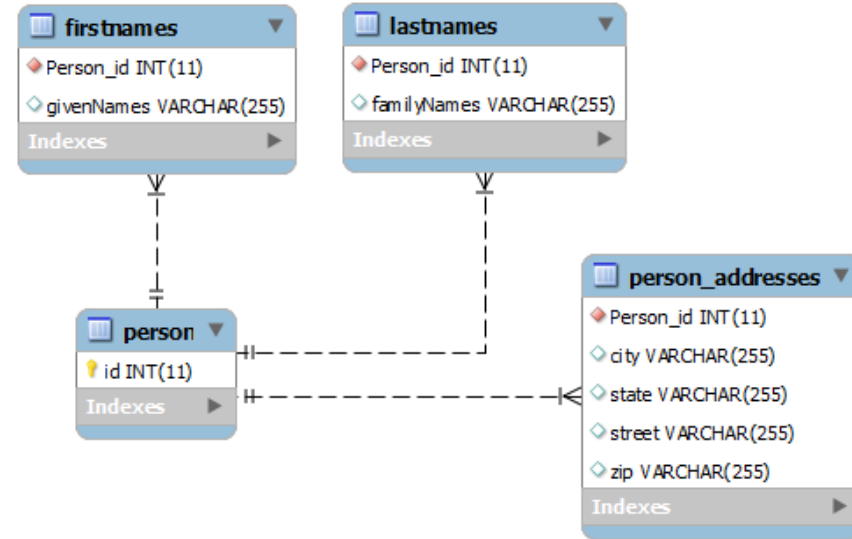## Element Collections

# Element Collections

- To map **collections of primitive values**
  - Or of embeddable classes (discussed later)


- Does not make sense form OO perspective
  - Still good to know about

# @ElementCollection

```java
@Entity
public class Person {
    @Id @GeneratedValue
    private int id;
    @ElementCollection
    @CollectionTable(name = "firstNames")
    private List<String> givenNames = new ArrayList<>();
    @ElementCollection
    @CollectionTable(name = "lastNames")
    private List<String> familyNames = new ArrayList<>();
    @ElementCollection
    private List<Address> addresses = new ArrayList<>();

    ...
```

Optional @CollectionTable:
to specify table name

Default table name is:
Classname_propertyname



firstnames
- Person_id INT(11)
- givenNames VARCHAR(255)
- Indexes

lastnames
- Person_id INT(11)
- familyNames VARCHAR(255)
- Indexes

person
- id INT(11)
- Indexes

person_addresses
- Person_id INT(11)
- city VARCHAR(255)
- state VARCHAR(255)
- street VARCHAR(255)
- zip VARCHAR(255)
- Indexes

# Map

- Maps need a key column
  - Here Pet is a @Embeddable class (more on this later)

```java
@Entity
public class Person {
    @Id @GeneratedValue
    private int id;
    private String name;
    @ElementCollection
    @MapKeyColumn(name = "name")
    private Map<String, Pet> Pets = new HashMap<>();
    ...
```

Optional: specify the column name.
Default name is: propertyname_KEY

```java
@Embeddable
public class Pet {
    private int age;
    private String species;
    ...
```

**person_pets**
- Person_id INT(11)
- age INT(11)
- species VARCHAR(255)
- pets_KEY VARCHAR(255)

Indexes