

Name: Yogesh Kumar

StudentID: 613836

Total inc review: 98

# Midterm 2022-06

## CS544 Enterprise Architecture

### Theory Section

A. [3 pts] Describe what surrogate keys are (in the context of relational databases):

3 Surrogate keys are those that never change & are always consistent & does not have any meaning to the Business always preferred as P.K in Relational databases.

B. [3 pts] Describe the difference between the Transient and Detached entity states

3 A Detached entity is one who has identity in the P.B but not in Persistence context. while transient entity is not yet in managed context neither in D.B.

C. [3 pts] Explain how Bi-Directional associations are mapped (what do you need to stop them from being 2 uni-directional associations)

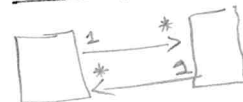
3 Bi-Directional associations ~~use~~ mappedBy to give up the property. If it was not the case then both the mapping will create two unidirection mapping.



not mapped by.

D. [3 pts] What does the @MappedSuperClass annotation do?

3 Does not create a table in the Database for parent all the properties then transferred to the child classes. Its weird because in DO there is a class & not in D.B so its considered Bad.



E. [3 pts] What annotations do you need to map a table that has a composite key?

3 @EmbeddedId & Embeddable Class.

F. [3 pts] Explain what the N+1 problem is in Hibernate

3 N+1 problem states the to load a chunk of data it use many small select, instead of one big select - problem is not when but how.

G. [3 pts] Describe what Optimistic Concurrency is

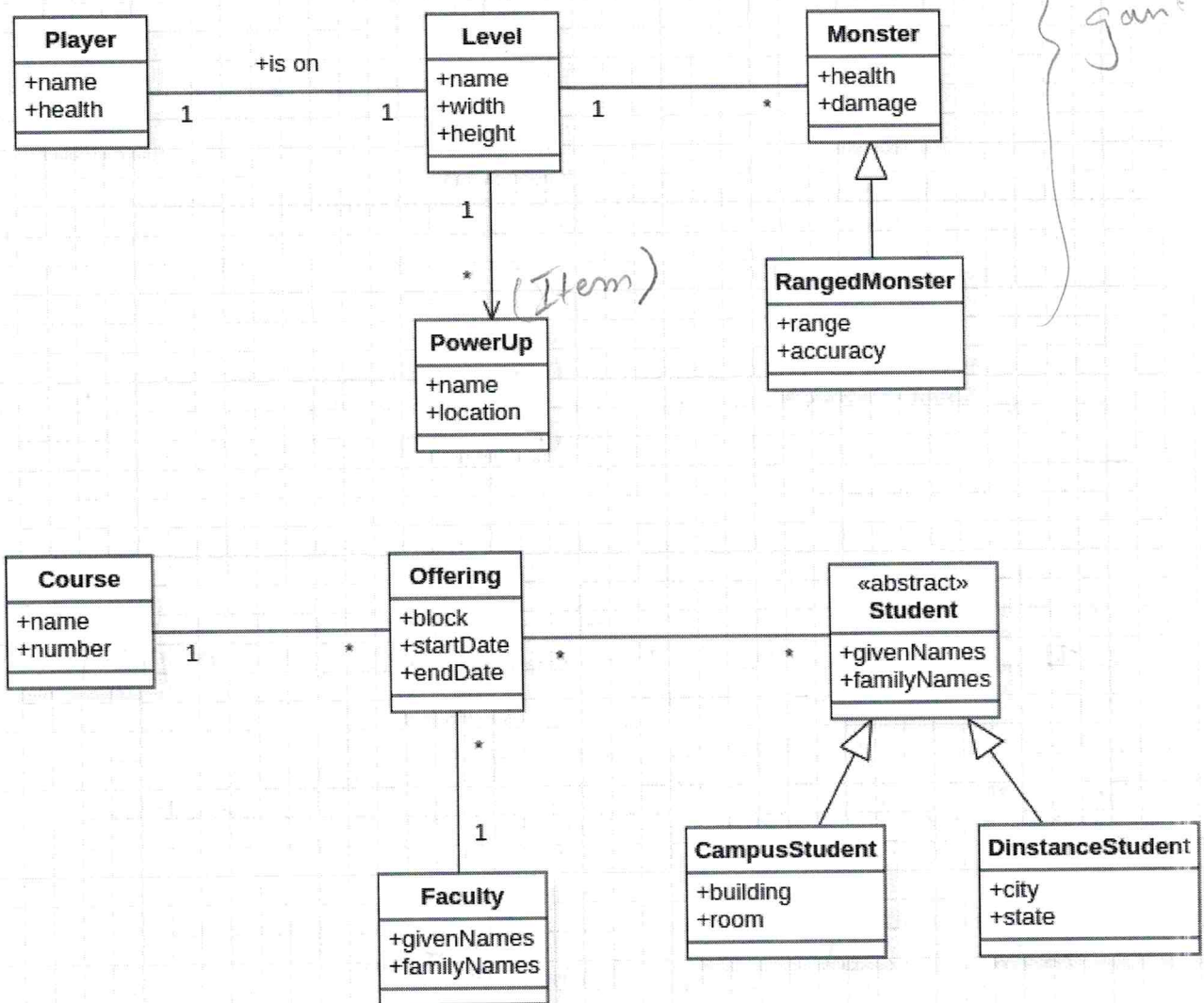
3 Optimistic Concurrency is a solution to the lost update problem where it tries to solve it through version where First update win last update fails.

H. [3 pts] Describe what Auto Commit Mode is and how it relates to Hibernate

3 Auto Commit Mode is most implemented in All the db where we can easily write a Query & it will be automatically committed by 1 of 6 default, hibernate turns it off because When there are more transactions it creates more overhead & less Isolation.

Name: Yogesh Kumar StudentID: 613836

These are the UML diagrams of the domains for the 2 mapping questions:



You can use these UML diagrams to get an overview of what the code looks like, which is useful when writing queries. Hint: use dates directly in you query string, like: '2022-06-03'

Name: Yogesh kumar

StudentID: 613836

## Exercises:

1. [24 pts] Based on the following classes with annotations write what the tables names, column names, and data types will be (also include if a column is auto increment).

```
@Entity
public class Player {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private int health;
    @OneToOne(mappedBy = "player")
    private Level level;
}
```

```
@Entity
public class Level {
    @Id
    private Long id;
    private String name;
    private int width;
    private int height;
    @OneToOne
    @JoinColumn(name="id")
    private Player player;
    @OneToMany
    private List<PowerUp> items =
        new ArrayList<>();
    @OneToMany(mappedBy = "level")
    private List<Monster> monsters =
        new ArrayList<>();
}
```

```
@Entity(name="Item")
public class PowerUp {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private String location;
}
```

```
@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public class Monster {
    @Id
    @GeneratedValue
    private Long id;
    private int health;
    private int damage;
    @ManyToOne
    private Level level;
}
```

```
@Entity
public class RangedMonster extends Monster {
    @Column(name="shootDistance")
    private int range;
    private double accuracy;
}
```

Player

id - BigInt - auto-increment  
name - varchar  
health - int

Level

id - BigInt - auto-increment  
name - varchar  
width - int  
height - int

Item

id - BigInt - auto-increment  
name - Varchar  
location - Varchar

Monster

id - BigInt - auto-increment  
health - int  
damage - int  
level\_id : BigInt

Level - Item

level\_id - BigInt  
items\_id - BigInt

Ranged Monster

shootDistance - int  
accuracy - double

Name: Yogesh Kumar StudentID: 613836

2. [24 pts] Add annotations to the following classes to map to the tables shown on the next page.

@Embeddable

```
public class Course {  
  
    private String name;  
  
    private int number;  
}
```

@Entity

```
public class Offering {
```

@Id

@GeneratedValue

```
private Long id;
```

@Temporal(TemporalType.DATE)

```
private Date startDate;
```

@Temporal(TemporalType.DATE)

```
private Date endDate;
```

@Embedded

```
private Course course;
```

@ManyToOne

```
private Faculty faculty;
```

@ManyToMany(mappedBy="courses")

```
private List<Student> students =  
    new ArrayList<>();
```

```
}
```

@Entity

```
public class CampusStudent  
    extends Student {
```

```
private String building;
```

```
private String room;
```

```
}
```

@Entity

```
public class Faculty {
```

@Id

@GeneratedValue

```
private Long id;
```

```
private String givenNames;
```

```
private String familyNames;
```

@OneToMany(mappedBy="faculty")

```
private List<Offering> offerings =  
    new ArrayList<>();
```

```
}
```

@Entity

```
public abstract class Student {
```

@Id

@GeneratedValue

```
private Long id;
```

```
private String givenNames;
```

```
private String familyNames;
```

@ManyToMany

@OrderBy(name="courses\_ORDER")

```
private List<Offering> courses =  
    new ArrayList<>();
```

```
}
```

@Entity

```
public class DistanceStudent  
    extends Student {
```

```
private String city;
```

```
private String state;
```

```
}
```

24



Name: Yogesh KumarStudentID: 613836

describe Offering;

| Field      | Type         | Null | Key | Default | Extra          |
|------------|--------------|------|-----|---------|----------------|
| id         | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| name       | varchar(255) | YES  |     | NULL    |                |
| number     | int(11)      | NO   |     | NULL    |                |
| endDate    | date         | YES  |     | NULL    |                |
| startDate  | date         | YES  |     | NULL    |                |
| faculty_id | bigint(20)   | YES  | MUL | NULL    |                |

describe Faculty;

| Field       | Type         | Null | Key | Default | Extra          |
|-------------|--------------|------|-----|---------|----------------|
| id          | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| familyNames | varchar(255) | YES  |     | NULL    |                |
| givenNames  | varchar(255) | YES  |     | NULL    |                |

describe Student;

| Field       | Type         | Null | Key | Default | Extra          |
|-------------|--------------|------|-----|---------|----------------|
| DTYPE       | varchar(31)  | NO   |     | NULL    |                |
| id          | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| familyNames | varchar(255) | YES  |     | NULL    |                |
| givenNames  | varchar(255) | YES  |     | NULL    |                |
| building    | varchar(255) | YES  |     | NULL    |                |
| room        | varchar(255) | YES  |     | NULL    |                |
| city        | varchar(255) | YES  |     | NULL    |                |
| state       | varchar(255) | YES  |     | NULL    |                |

describe Student\_Offering;

| Field         | Type       | Null | Key | Default | Extra |
|---------------|------------|------|-----|---------|-------|
| students_id   | bigint(20) | NO   | PRI | NULL    |       |
| courses_id    | bigint(20) | NO   | MUL | NULL    |       |
| courses_ORDER | int(11)    | NO   | PRI | NULL    |       |

Name: Yogesh Kumar

StudentID: 613836

3. [12 pts] Based on the game domain write queries to retrieve:

a. All players whose health is greater than 50 and are on the level named "Beach"

4  
Select Distinct P from Player join P.level as L  
where P.health > 50 and L.name = "Beach"

b. All PowerUp items on the level named "Mountains"

4  
Select distinct pu from Level as L join L.items as pu  
where L.name = "Mountains" and

c. All levels that have a RangedMonster with health greater than 100

4  
Select distinct L from Level as L join L.monsters as m  
where m.health > 100 and type(m) = 'Ranged Monster'

4. [12 pts] Based on the university domain write queries to retrieve:

a. All Students with the familyNames "Smith"

4  
Select distinct s from Student as s where s.familyNames = "Smith"

b. All Faculty teaching the course "Enterprise Applications"

4  
Select distinct f from Faculty as f join f.offerings as o  
where o.course.name = "Enterprise Application".

c. All Offerings with a startDate after 2022-01-01 that has CampusStudents with the givenNames "John"

3  
Select distinct o from Offering join o.students as s  
where o.startDate > "2022-01-01" and type(s) = "campusStudents"