

# Midterm 2022-06

## CS544 Enterprise Architecture

### Theory Section

A. [3 pts] Explain what a surrogate key is in the context of databases, and why they are preferred

- a key that doesn't have any business meaning that identifies each unique entry/row of data.
- they are preferred because they don't create confusion with our business and always maintain uniqueness as they can be generated on uniquely inserted

B. [3 pts] What does the JPA @Transient annotation do?

- tell to database not to store that property into DB.

C. [3 pts] Give 3 circumstances that cause Hibernate to push changes to the database

- on persist() - insert new element to persistent context & DB
- on flush() - explicitly push changes to DB before committing the TX.
- on merge() - insert/update an element to persistent context & DB.

D. [3 pts] What does Hibernate need in order to persist a real list? Explain the difference with Bag.

- to persist a real list Hibernate needs order column because List are indexed unlike bag which are not indexed.

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

- @MappedSuperClass does map a concrete superclass to its child classes where all properties of the superclass goes to child tables.

F. [3 pts] What causes the N+1 problem?

- Joining collections and lazy loading behaviour of hibernate on collections.

G. [3 pts] Explain what a Filter does in a Web Container

- adding functionality to our web container pre-and post-request.
- filter requests before forwarding to our server and filter results after they are processed and replied back to the client.

H. [3 pts] Explain what optimistic concurrency is:

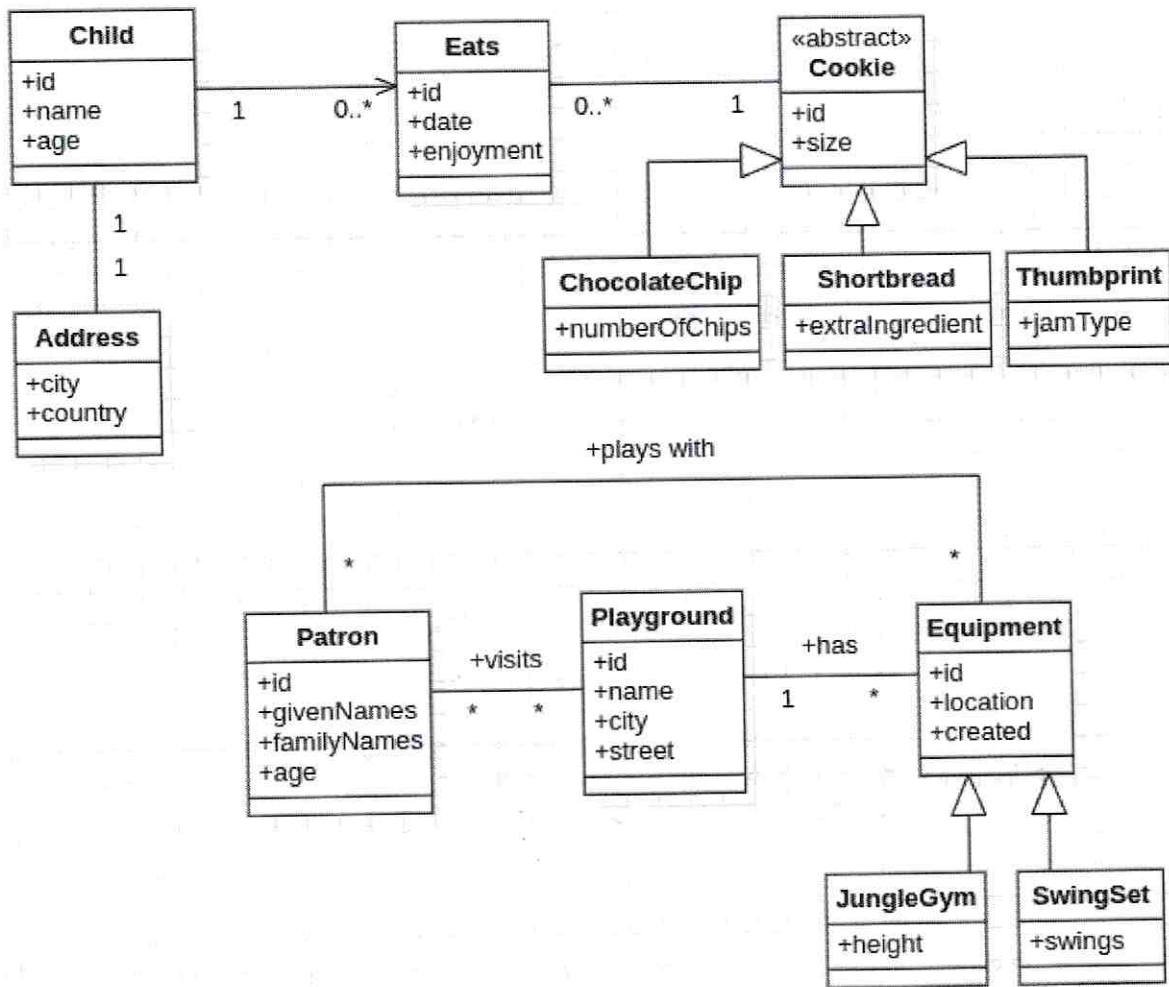
- introducing versioning to every commit in a transaction to mitigate lost-update concurrency problem.

allows you to run in read-committed

Name: ASHENAFI T. Gebreyesus

StudentID: 614689

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'

Single  
Joined  
Concrete  
DWT  
Double  
Hyst

Name: Ashenshi T. G

StudentID: 614689

## 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 Child {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    private int age;
    @Embedded
    private Address address;
    @OneToMany
    private List<Eats> eats = new ArrayList<>();
}
```

```
@Embeddable
public class Address {
    private String city;
    private String country;
}
```

```
@Entity
public class Eats {
    @Id
    @GeneratedValue
    private Long id;
    @Temporal(TemporalType.DATE)
    private Date date;
    private String enjoyment;
    @ManyToOne
    private Cookie cookie;
}
```

```
@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public class Cookie {
    @Id
    @GeneratedValue
    private Long id;
    private double size;
    @OneToOne(mappedBy = "cookie")
    private List<Eats> eats = new ArrayList<>();
}
```

```
@Entity
public class ChocolateChip extends Cookie {
    private int numberOfChips;
}
```

```
@Entity
public class Shortbread extends Cookie {
    private String extraIngredient;
}
```

```
@Entity
public class Thumbprint extends Cookie {
    private String jamType;
}
```

Child ✓

id bigint (20) auto\_increment ✓  
 name varchar (255) ✓  
 age int ✓  
 city varchar (255) ✓  
 country varchar (255) ✓

Eats ✓

id bigint (20) auto\_increment ✓  
 date DATE ✓  
 enjoyment varchar (255) ✓  
 cookie\_id bigint (20) ✓

Child-Eats ✓

child\_id bigint (20) ✓  
 eats\_id bigint (20) ✓

ChocolateChip ✓

id bigint (20) auto\_increment ✓  
 size double ✓  
 number\_of\_chips int ✓

Shortbread ✓

id bigint (20) auto\_increment ✓  
 size double ✓  
 extra\_ingredient varchar (255) ✓

Thumbprint ✓

id bigint (20) auto\_increment ✓  
 size double ✓  
 jam\_type varchar (255) ✓

(20.5)

-35



Name: Ashwini T.G StudentID: 614689

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

@Entity ✓

public class Patron {

@Id ✓

@GeneratedValue ✓

private Long id;

private String givenNames;

private String familyNames;

private int age;

@ManyToMany (mappedBy = "patrons") ✓

private List<Playground> playgrounds  
= new ArrayList<>();

@ManyToMany (mappedBy = "patrons") ✓

private List<Equipment> equipments  
= new ArrayList<>();

}

@Entity ✓

public class Equipment {

@Id ✓

@GeneratedValue ✓

private Long id;

private String location;

private Date created;

@ManyToOne ✓

private Playground playground;

@ManyToMany ✓

private List<Patron> patrons  
= new ArrayList<>();

}

@Entity ✓ @SecondaryTable (name = "Locations") ✓

public class Playground {

@Id ✓

@GeneratedValue ✓

private Long id;

@

private String name;

@Column (table = "Locations") ✓

private String city;

@Column (table = "Locations") ✓

private String street;

@OneToMany ✓

private List<Patron> patrons  
= new ArrayList<>();

@OneToOne (mappedBy = "Playground") ✓

private List<Equipment> equipments  
= new ArrayList<>();

}

@Entity ✓

public class JungleGym extends Equipment {

private double height;

}

@Entity ✓

public class SwingSet extends Equipment {

private int swings;

}

23

Name: Ashenafi T.G StudentID: 614689

describe Patron;

| Field       | Type         | Null | Key | Default | Extra          |
|-------------|--------------|------|-----|---------|----------------|
| id          | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| age         | int(11)      | NO   |     | NULL    |                |
| familyNames | varchar(255) | YES  |     | NULL    |                |
| givenNames  | varchar(255) | YES  |     | NULL    |                |

describe Playground;

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

describe Locations;

| Field  | Type         | Null | Key | Default | Extra |
|--------|--------------|------|-----|---------|-------|
| city   | varchar(255) | YES  |     | NULL    |       |
| street | varchar(255) | YES  |     | NULL    |       |
| id     | bigint(20)   | NO   | PRI | NULL    |       |

describe Playground\_Patron;

| Field          | Type       | Null | Key | Default | Extra |
|----------------|------------|------|-----|---------|-------|
| playgrounds_id | bigint(20) | NO   | MUL | NULL    |       |
| patrons_id     | bigint(20) | NO   | MUL | NULL    |       |

describe Equipment;

| Field         | Type         | Null | Key | Default | Extra          |
|---------------|--------------|------|-----|---------|----------------|
| DTYPE         | varchar(31)  | NO   |     | NULL    |                |
| id            | bigint(20)   | NO   | PRI | NULL    | auto_increment |
| created       | date         | YES  |     | NULL    |                |
| location      | varchar(255) | YES  |     | NULL    |                |
| height        | double       | YES  |     | NULL    |                |
| swings        | int(11)      | YES  |     | NULL    |                |
| playground_id | bigint(20)   | YES  | MUL | NULL    |                |

describe Equipment\_Patron;

| Field         | Type       | Null | Key | Default | Extra |
|---------------|------------|------|-----|---------|-------|
| equipments_id | bigint(20) | NO   | MUL | NULL    |       |
| patrons_id    | bigint(20) | NO   | MUL | NULL    |       |

Name: Amenchi T. G StudentID: 614689

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

a. All children in the city of Fairfield with age less than 10

4  
Select C From child C where C.address.city = "Fairfield" and  
C.age < 10 ;

b. All ChocolateChip cookies eaten on '2022-09-17'

4  
Select distinct C From cookies as C join C.eats as E where  
E.date = "2022-09-17" and type(C) = chocolateChip ;

c. All cookies eaten by the child with name 'Jimmy' whose enjoyment was 'supreme' and the type of cookie was Thumbprint

3.5  
Select distinct E.cookie from child as C join ~~C~~.Eats as E  
where C.name = "Jimmy" and E.enjoyment = "supreme" and  
type(E.cookie) = Thumbprint ;

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

a. All Equipment at the playground called 'Treasure Island'

4  
Select E From Equipment E where ~~E.playground.name = "Treasure Island"~~  
E.playground.name = "Treasure Island" ;

b. All Patrons that played with the JungleGym at the Playground called 'Woodland Park'

3.5  
Select distinct P from Patron P join ~~P~~.Equipment as E where  
type(E) = JungleGym and E.playground.name = "Woodland park" ;

c. All Playgrounds that have a SwingSet created before '2000-01-01'

4  
Select E.playground from Equipment E where  
type(E) = SwingSet and E.created < "2000-01-01" ;