

Phase 2

Day 3

12 Dec 2023

Limitation of JDBC

1. Using jdbc we can't store java object as well as we can't retrieve java object. we need to convert object to sql query format and vice-versa.
2. Jdbc use sql query language. SQL is database dependent query language.
3. JDBC doesn't support relationship. One – to- many (PK—FK)
4. JDBC throw checked exception. We need to handle using try-catch or throws mandatory. And exception hierarchy is database dependent.

ORM : Object Relation Mapping

Orm is a concept.

Programming Java

Object

Database

Relation

Class Employee

Id,name,salary – variable

Employee

id,name,salary – columns

Employee emp = new Employee();

Emp.setId(100);

Emp.setName("Ravi");

Emp.setSalary(12000);

100, Ravi, 12000

Mapping

Employee = Employee

Id = ID PK

Name = name

Salary = salary with their data types.

This mapping we can do with

- 1. Xml file**
- 2. Using annotation**

Implementation of ORM tool is

- 1. Hibernate**
- 2. JPA**
- 3. iBatis**

Hibernate : Hibernate is an open source orm tool provided by jboss. Using hibernate we can achieve orm features.

Create a table in Database.

Employee--→ eid(PK), ename, Salary

In orm java bean class is known as **entity** class. using annotation

@Entity //

@Table(name="Employee") optional if table name and entity class name same.

class Employee {

@Id the column which contains pk

@Column(name="eid")

private int eid;

@Column(name="ename")

private String name;

@Column(name="salary")

private salary;

}

In Hibernate we use hibernate.cfg.xml file. Which contains database details.

Like **drivername, url,username and password.** It is equal to resource file

Old Version we are writing mapping details using hibernate mapping file.

.hbm.xml

New version mapping file replace by annotation ie **@Entity and @Id**

We will create Simple Hibernate Program to do CRUD Operation on Employee table.

Hibernate project must be created using maven.

Table create in database ie employee

Then create java bean class with annotation @entity and @id

Now we will create configuration file which contains database details.

Hiberanate.cfg.xml file

Database details drivername, user,username, password

+

```
org.hibernate.dialect.MySQLDialect  
    it is responsible to convert java object into sql and  
    vice-versa.
```

+

Mapping : entity class with annotation

Using Configuration pre defined class we need to load hibernate.cfg.xml file.

SessionFactory : SessionFactory is an interface which help to create more than one **session** object. SessionFactory is like a Connection con in JDBC.

Session : Session is an interface provided by hibernate which contains more than one method which help to do CRUD Operation on entity class without depending upon sql query. It is like a Statement or PreparedStatement in JDBC.

Transaction : generally we do transaction on DML query

Insert / delete and update.

If all insert query, delete query as well as update query execute successfully we can say commit. If any query go wrong we can say rollback.

If we do any operation using JDBC by default they are auto commit.

But if we do any DML operation using ORM tool like hibernate they are not auto commit. So using hibernate api we need to commit or rollback base upon our requirements.