```
---- deleting the table ---
CallableStatement p = c.prepareCall("drop table details");
boolean e = p.execute();
---- deleting the database ---
CallableStatement p = c.prepareCall("drop database goodfriday");
boolean e = p.execute();
c.close();
}
}
```

ORM :- Object Relational Mapping.

ORM is one of the technique to perform CRUD operation with database using object oriented programming Language.

In java we have many ORM frameworks:-

i) Toplinks ii) Ibatis iii) Hibernate etc.

In python we have .Django

HIBERNATE:-

Hibernate is one of the ORM java frameworks which simplifies the development of java application to Interact with database.

Advantages of Hibernate:-

* In hibernate we can deal with objects
* We can implement OOP's concept
* No need to write SQL Queries.
* It is a Open source
* In hibernate table will get created automatically
* Hibernate is one of the high performance oriented java framework because by default hibernate uses Cache Memory

Mapping:-

Establishing the relationship between two tables by using primary key and foreign key.

A types of Mapping:-

1. One to One
2. One to many
3. Many to one
4. Many to many.

① ONE to ONE.

```
@Entity
@Data
public class Person
{
 private int id;
 @Id
 private String pName;
 @Column (nullable = false)
 private int age;
 @OneToMany
 private List<Bank> bank;
}
```

```
@Entity
@Data
public class Bank
{
 @Id
 private String bName;
 @Column (unique = true)
 private long aNo;
 private double balance;
}
```

```
public class MainClass
{
 P.S.V.M (String[] args)
 {
  EntityManagerFactory e = Persistence.createEntityManagerFactory
                           ("mapping");
  EntityManager m = e.createEntityManager();
  EntityTransaction t = m.getTransaction();
  //.....
  Bank b1 = new Bank();
  b1.setANo(7896441);
  b1.setBalance(99.9);
  b1.setBName("Canara");
  Bank b2 = new Bank();
  b2.setANo(99999991);
  b2.setBalance(85.25);
  b2.setBName("hdfc");
```

```
ArrayList<Bank> al= new ArrayList<Bank>();
al.add(b1);
al.add(b2);
person p= new person();
p.setAge(25);
p.setId(1);
p.setPName("mohan sir");
p.setBank(al);

t.begin();
m.persist(b1);
m.persist(b2);
m.persist(p);
t.commit();
}}
```

NOTE :-
first persist the child then parent table

person table

pName	age	id
Mohan Sir	25	1

Bank table

bName	aNo	balance
Canara	...	99.9

Person pName bankId
mohan Sir ca

ONE TO ONE :-

```
@Entity
@Data
public class Aadhar{
@Id
private long ANo;
private String address;
@Column(unique = true)
private long pNo;
}
```

```
@Entity
@Data
public class Person{
private int id;
@Id
private String pName;
@Column(nullable = false)
private int age;
@OneToOne
private Aadhaar adr;
}
```

```
public class MainClass
{
  P.S.V.M(String[] args)
  {
    EntityManagerFactory e = Persistence.CreateEntityManagerFactory
                                ("mapping");
    EntityManager m = e.createEntityManager();
    EntityTransaction t = m.getTransaction();

    Aadhar a = new Aadhar();
    a.setAddress("bengaluru");
    a.setANo(987654321);
    a.setPNo(9898989898);
```

```
Person p = new Person();
p.setAge(25);
p.setPName("Umesh Anna");
p.setId(1);
p.setAdr(a);

t.begin();
m.persist(p);
m.persist(a);
t.commit();
}
}
```

Many to One :-

```
@Entity
@Data
public class Bank
{
@Id
private String bName;
private double balance;
@ManyToOne
private Person per;
}
```

```
@Entity
@Data
public class Person
{
@Id
private String name;
private int age;
}
```

```
public class MainClass
{
p.s.v.m (String[] args)
{
EntityManagerFactory e = Persistence.createEntityManager-
                         -Factory("Mapping");
EntityManager m = e.createEntityManager();
EntityTransaction t = m.getEntityTransaction();
Person p = new Person();
p.setAge(36);
p.setName("ramesh");
Bank b1 = new Bank();
b1.setBName("canara");
b1.setBalance(8.9);
b1.setPer(p);
```

```
Bank b2 = new Bank();
b2.setBName("SBI");
b2.setBalance(23.4);
b2.setPer(p);
Bank b3 = new Bank()
b3.setBName("ICICI");
b3.setBalance(9543523);
b3.setPer(p);

t.begin();
t.beo m.persist(b1);
m.persist(b2);
m.persist(b3);
m.persist(p);
t.commit();
}
}
```

name	age
ramesh	36

bName	balance
canara	8.2
ICICI	9343.23
SBI	93.4

per_name
ramesh
ramesh
ramesh.

Many-to-Many:-

```
@Entity
@Data
public class Person
{
  @Id
  private String name;
  private int age;
  @ManyToMany
  private List<TouristPlace> touristPlaces;
}
```

```
@Entity
@Data
public class TouristPlace
{
  @Id
  private String place;
  private int pincode;
}
```

```
public class MainClass
{
  P.S.V.M(String[] args)
  {
    EntityManagerFactory e = Persistence.createEntityManagerFactory
                             ("Mapping");
    EntityManager m = e.createEntityManager();
    EntityTransaction t = m.getTransaction();

    TouristPlace place = new TouristPlace();
    place.setPincode(56255);
    place.setPlace("mysore");
```

```
TouristPlace place2 = new TouristPlace();
place2.setPincode(8505);
place2.setPlace("bangalore");

ArrayList<TouristPlace> arrayList = new ArrayList<TouristPlace>();
arrayList.add(place);
arrayList.add(place2);

Person person = new Person();
person.setAge(58);
person.setName("dbas");
person.setTouristPlaces(arrayList);

person person2 = new Person();
person2.setAge(35);
person2.setName("rocky");
person2.setTouristPlaces(arrayList);

t.begin();
m.persist(place);
m.persist(place2);
m.persist(person);
m.persist(person2);
t.commit();
}
```

person_table

name	age
dbas	58
rocky	35

tourist place table

place	pincode
beng	8525
mysore	561255

person_tourist place table

person_name	touristPlaces_place
dbas	mysore
dbas	bangalore
rocky	mysore
rocky	Bangalore

Dynamic Input :-

```
public class Tester
{
  PSVM(String[] args) throws Exception.
  {
    class.forName("com.mysql.cj.jdbc.Driver");
    Connection c = DriverManager.getConnection("jdbc:mysql://
                   localhost:3306/Saturday","root","root");

    //Insertion:-
    CallableStatement p= c.prepareCall("insert into details
                                  value(?,?,?)");

    Scanner sc = new Scanner(System.in);
    SOP("enter id");
    p.setInt(1, sc.nextInt());
    SOP("enter name");
    p.setString(2, sc.next());
    SOP("enter salary");
    p.setDouble(3, sc.nextDouble());
    int e = p.executeQuery();
    SOP(e);
    //Update.
    CallableStatement p= c. prepareCall("Update details set name=
                          where id=?);
    Scanner sc = new Scanner(System.in);
    SOP("Enter id");
    p.setInt(2, sc.nextInt());
    SOP("enter new name");
    p.setString(1, sc.next());
    int e = p.executeUpdate();
    SOP(e);
    // Delete.
    CallableStatement p= c. prepareCall("delete from details
                                  where id=?");
    Scanner sc =new Scanner(System.in);
    SOP("enter id");
```

```
p.setInt(1, sc.nextInt());
int e = p.executeUpdate();
SOP(e);
```

Fetching particular value:-

```
CallableStatement p= c.prepareCall("select * from details where
                                    id = ?");
Scanner sc= new Scanner(System.in);
SOP("enter Id");
p.setInt(1,sc.nextInt());
ResultSet r = p.executeQuery();
r.next();
SOP(r.getInt(1) + " " + r.getString(2) + " " + r.get
c.close();
}
}
```

Currently we are using 1.8 JDK → JDBC → 4.2
Latest version of jbc → jdbc. 4.3.

Maven Project:-

It is a open-source build automation which is used to manage the projects and widely used to develop the java applications.

* POM.xml → project object model extensible markup language
pom.xml will give complete configuration information of the project.

* Latest version of JDBC → 4.3
JAVA → 21.

Currently we are using JDBC → 4.2 JDK → 1.8.

Dependency version is 8.0.33 of mysql.

Drawbacks of JDBC:-

1. In JDBC we cannot deal with objects
2. We cannot implement OOPs concept effectively
3. We have to write SQL queries manually
4. Can't able to achieve loose coupling.

09/1/24

Servlets:- [Component of Server]

How to create servlet project ?

* Go to file → new → maven project → click on next (don't select & check create a simple project checkbox).
* In catalog dropdown select all catalogs and in filter text field type eg. apache.maven and select maven-archetype-webapp (1.0/1.4)
* Click on next after selecting filter and enter groupid (company name) and artifact id (project name) (lowercase and avoid special char except . and -/-) and click on finish.
* If it is pausing at 33% while creating servlet project go to console and type (Y.)

How to download Apache tomcat Server:-

→ Go to browser type apache tomcat download, go to official website of apache tomcat (click on 1st link) under download option click on tomcat 9 and Under core click on zip ( )

How to add Server to eclipse:-

→ Go to window → show view → servers → Click on no servers are available link → select apache version 9.0 click on next | Add, go to browse → downloads → select extracted folder → select folder and click on finish

sol also.

NOTE:-

Once we create servlet project we should follow below mentioned steps.

Step 1:- deploy the project to server.

Steps:- Select the project → Right click on the project → go to properties → Targeted Runtimes → select the server and click on apply & close.

Step 2:- Add 8.jars Servlet API (4.0.1) dependency

Step 3:- Select the project → Right click on project → build path → Configure build path → select (or check) all the checkboxes.

JRE System Library [Java SE 1.8]
Maven Dependencies
Server Runtime [Apache tomcat v7.0]
and click on apply & close

Step 4:- Update the project
select the project → Right click on proj → go to maven → Update project → select force update of snapshots checkbox and click on okay.

NOTE:- [to establish the connection from html file to java class]

```
<web-app>
  <display-name> Archetype created Web Application </display-name>
  <servlet>
    <servlet-name> </servlet-name>
    <servlet-class> </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name> </servlet-name>
    <url-pattern> </url-pattern>
  </servlet-mapping>
</web-app>
```