Phase 2

SQL Self Learning using My SQL

JDBC: Connecting MySQL or Oracle database using Java Technologies.

We can insert, delete , update and retrieve records from database.

Hibernate : ORM (Object Relation Mapping ) : Advanced of JDBC to store, retrieve, update and insert record using Java Technologies.

Servlet and JSP : Using these two modules we can create the web application using Java technologies.

In Virtual open the terminal and type

mysql –u root –p

Simplilearn

mysql >

Database is use to the data in table format.

We can connect the mysql database two ways

1. Using command prompt
2. Workbench (GUI base ).

After connected mysql database using command prompt

show databases; This command is use to display all databases present in mysql database.

Create database databaseName;

create database maybank

use databasename; This command is use to switch in database.

Syntax to create the table

Employee --🡪Table

Three column id name salary

Number String float

create table tableName(colummName datatype, columnName datatype);

create table employee(id int primary key,name varchar(10), salary float);

command to insert the record in Table

insert into employee values(1,’Raj’,12000);

To view the records from a table

select \* from employee

retrieve the record using conditions

select \* from employee where id=1;

select \* from employee where name = ‘Raj’;

select \* from employee where salary > 14000;

update query

update employee set salary = 22000 where id=1;

delete query

delete from employee where id=1;

JDBC : Java Database Connectivity : JDBC is a API (Application programming interfaces) which help to connect the RDBMS (oracle or Mysql) using Java technology.

Relational Database Management System.

JDBC always throw checked exception. So we have to handle this exception using try-catch or throws.

1 we have to load the Driver : Driver is a pre-defined class provider by vendor in the form of jar file which help to connect the database.

Java provided pre-defined class Class. which contains pre-defined method forName() and it is a static method.

Class.forName(“driverName”);

com.mysql.jdbc.Driver :5.x

com.mysql.cj.jdbc.Driver :8.x

Establish the connection

Now we have to create the reference of PreparedStatement. Which provide set of method which help to do insert, delete, update and retrieve.

Don’t write any business logic and database logic in main method.

Maven Project :

Maven is known as build tool. This tool responsible to compile the program, run the program, creating jar or war. Downloading the dependencies for the project.

Pom.xml ( project object model) file. This file is known as a maven deployment description file.

Database Table -------🡪 Employee ---🡪 ID,Name,Salary (columns)

Java (JavaBean) -🡪Employee--🡪id,name,salary (variable)

DAO Layer (Data Access Object ) : This class is responsible to write pure database logic.

EmployeeDao : this class contains set method which help to do database operation on table. Like insert, delete, update and retrieve.

Service layer : This layer is responsible to write business logic. We can write business logic before interact with database or after retrieve records from database.

EmployeeService : pure business logic

Class Employee {

Id,name,age

Setter and getter methods

}

class Information {

void passInformation(Employee emp) {

}

}

1st person

Information in = new Information();

Employee emp = new Employee();

emp.setId(1);

In.passInformation(emp); hold only id

2nd person

Information in = new Information();

Employee emp = new Employee();

emp.setId(1);

emp.setName(“Ravi”);

In.passInformation(emp);

3rd person

Information in = new Information();

Employee emp = new Employee();

emp.setId(1);

emp.setName(“Ravi”);

emp.setAge(21);

In.passInformation(emp);

**ORM : Object Relation Mapping**

**ORM is a concept.**

**Limitation of JDBC**

1. **Using JDBC we can’t store Java object into database or we can’t retrieve Java object from a database. In DAO layer we have to convert Java object into sql query format and sql query format into Java object.**
2. **JDBC use SQL Language. SQL is database dependent language. So when we move from one database to another database we have to change the query.**
3. **JDBC through checked exception. All exception hierarchy is database dependent.**
4. **JDBC doesn’t relationship is a and has relationship.**

**Object Relation Mapping**

**JavaBean or Entity class**

**@Entity**

**class Employee { Employee table**

**@Id**

**id,name,salary id,name,salary**

**}**

**Mapping**

**Employee ---🡪Employee**

**Id -🡪ID PK**

**Name🡪name**

**Salary 🡪salary**

**Old Version we are doing mapping using xml file**

**New version we are doing mapping using annotation**

**In ORM we provide all database details in xml configuration file.**

**driverName**

**url**

**username**

**password**

**mapping class details.**

**Hibernate and JPA (Java Persistence API) are implementation of ORM.**

**Hibernate is a third party framework which support ORM features.**

25-04-2022

We have to the Entity class means class bean class on that class we have to use the annotation @Entity and @Id the column which contains primary key.

If we do any operation like Insert, Delete and Update using JDBC it will auto commit.

But if we want to do the operation through Hibernate we have to use Transaction. Through Hibernate it will not auto commit.

26-04-2022

If we are planning retrieve multiple records using hibernate. Hibernate provide their own query language

Ie HQL (Hibernate Query Language).

Sql (Structured Query Language) : using sql we can insert, delete, update and retrieve.

Select \* from employee; (employee is table name it is not a case sensitive).

We are retrieving all column from a table.

Select \* from employee where id=100;

Select \* from employee where name = ‘Ravi’;

Select \* from employee where salary > 12000;

HQL use only for retrieve purpose

Hql (Hibernate Query Language)

Select emp from Employee emp (Employee is java bean class name and emp is object consider)

Select emp from Employee emp where emp.id = 100; (emp is object and id is variable of java bean class)

Master / main ---🡪

Demo.java (some code)

Push this code in remote repository

Git clone URL

Demo.java (some code)

Master branch or main branch (if you do directly changes on my code with master branch or main branch)

Git pull

Create new folder (new repository)

Inside that folder

Git clone https://github.com/Kaleakash/maybank\_2022\_java\_full\_stack\_batch.git

int a; your code

int a; my code

SQL relationship

One to many : Trainer --🡪 Student (PK and FK)

One to One : Person 🡪 Passport (one to One)

Many to one : Employee -🡪 Department

Many to Many : Employees -🡪 SkillSet

TrainerAndStudent

TId TName tech SId SName age

1 Raj Java 100 Seeta 21

1 Raj Java 101 Reeta 22

1 Raj Java 102 Meeta 23

Trainer

PK

TId TName tech

1 Raj Java

2 Ravi Python

Student

PK FK

SId SName Age TSId

100 Seeta 21 1

101 Reeta 22 1

102 Meeta 23 1

103 Leeta 24 2

FK : Foreign key: Foreign key is use to connect the pk of another table. And it allow only those values which present in primary key and it can allow duplicate. Primary key doesn’t allow duplicate.

PK : Primary key : if column contains primary key it doesn’t allow duplicate.

create table trainer(tid int primary key,tname varchar(10), tech varchar(10));

Hibernate Relationship