

Dt : 17/3/2025

Construct Application demonstrating Function in JDBC:

step-1 : Construct Function to retrieve Employee TotSal based in emp-Id

create or replace Function RetrieveTotSal72

(id varchar2) return number as ts number;

begin

select totalsal into ts from EmpSalary72 where eid=id;

return ts;

end;

/

step-2 : Construct JDBC Application to execute function.

Program : DBCon13.java

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon13 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try(s){
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection
                ("jdbc:oracle:thin:@localhost:1521:xe","system","tiger");
            CallableStatement cs = con.prepareCall
                ("call ?:=RetrieveTotSal72(?)");
            System.out.println("Enter the Emp-Id to retrieve TotSal:");
            String eId = s.nextLine();
            cs.setString(2, eId);
            cs.registerOutParameter(1, Types.FLOAT);
```

```

        cs.execute();
        System.out.println("*****Details*****");
        System.out.println("Emp-Id:"+eId);
        System.out.println("TotSal:"+cs.getFloat(1));
        con.close();
    }catch(Exception e) {
        e.printStackTrace();
    }
}
}

```

**o/p:**

**Enter the Emp-Id to retrieve TotSal:**

**T121**

**\*\*\*\*\*Details\*\*\*\*\***

**Emp-Id:T121**

**TotSal:114300.0**

-----

**Assignment:**

**Construct and Execute Function to retrieve Student Percentage based on RollNo.**

-----

**faq:**

**define registerOutParameter()-method?**

**=>registerOutParameter()-method is from 'CallableStatement' and which specify the type of data recored(loaded) to the Parameter-Index-field of CallableStatement-Object**

**syntax:**

**cs.registerOutParameter(1,Types.FLOAT);**

**Note:**

*"Types" in JDBC is a class from java.sql package and which specify the SQL-TYPE used in registerOutParameter()-method*

---

*\*imp*

*Transaction Management in JDBC:*

*define Transaction?*

*=>The set-of-statements which are executed on a single resource or multiple resources using ACID properties is known as Transaction.*

*A - Atomicity*

*C - Consistency*

*I - Isolation*

*D - Durability*

*A - Atomicity*

*=>The process in which the statements in Transaction are executed at-a-time or not-at-all,is known as Atomicity.*

*C - Consistency*

*=>The process in which the selected state of resources remain same until the Transaction is complemented,is known as Consistency.*

*I - Isolation*

*=>The process in which multiple users are executed independently is known as Isolation.*

## **D - Durability**

**=>The process in which recording the state of transaction and making it available for Customers,is known as Durability**

---

**faq:**

**define Transaction Management?**

**=>The process of controlling the Transaction from starting to ending is known as Transaction Management.**

**=>We use the following methods in Transaction Management:**

**(a)getAutoCommit()**

**(b)setAutoCommit()**

**(c)setSavepoint()**

**(d)releaseSavepoint()**

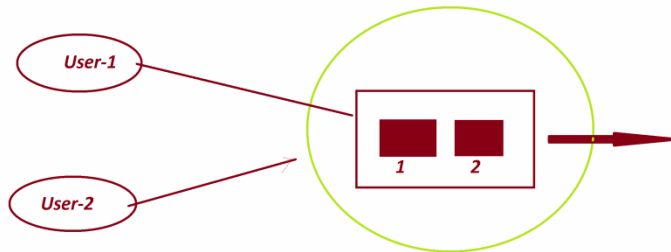
**(e)commit()**

**(f)rollback()**

---

**Transaction : Booking Ticket using BookMyShow**

- 1.Login Process
- 2.Region
- 3.Movie
- 4.Date
- 5.Show Time
- 6.No of Tickets
- 7.Availability
- 8.Select the Seats
- 9.Payment
  - (a)Card
  - (b)IB
  - (c)UPI
- 10.Conf....
  - (a)Mobile
  - (b)MailId
- 11.Logout



Venkatesh Ma...