DBMS LAB-5

Source Code: [GitHub](#)

//db connecter

```java
package com.ved;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;


public class ConnectDB {

    // Database connection constants

    private static final String URL = "jdbc:mysql://localhost:3306/mydb";

    private static final String USERNAME = "root";

    private static final String PASSWORD = "password";


    public static Connection dbConnect() {

        Connection connection = null;

        try {

            // Register JDBC driver

            Class.forName("com.mysql.cj.jdbc.Driver");


            // Open a connection

            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);

            if (connection != null) {

                System.out.println("Successfully connected to the database!");

            }


        } catch (ClassNotFoundException e) {
```

```java
            System.out.println("MySQL JDBC Driver not found.");

            e.printStackTrace();

        } catch (SQLException e) {

            System.out.println("Connection failed! Check output console");

            e.printStackTrace();

        }


        return connection;

    }


    // Test the connection

    public static void main(String[] args) {

        try (Connection conn = dbConnect()) {

            if (conn != null) {

                System.out.println("Database connection is working!");

            }

        } catch (SQLException e) {

            System.out.println("Error closing connection");

            e.printStackTrace();

        }

    }

}
```

// ProductTrigger.java

```java
/*First create Products table-

create table products(product_id int primary key auto_increment,product_name varchar(10),price
decimal(10,2));

create Product_Metadata table-
```

create table Product_Metadata(product_id int,last_modified Timestamp default CURRENT_TIMESTAMP ON uPDATE CURRENT_TIMESTAMP);

```java
*/
package com.ved;


import java.sql.Connection;

import java.sql.Statement;


public class ProductTrigger {

  public static void main(String[] args) {

    try (Connection con = ConnectDB.dbConnect();

      Statement statement = con.createStatement()) {

      // Define the SQL statement to create a trigger

      String createTriggerSQL = """

        CREATE TRIGGER update_last_modified_trigger

        AFTER INSERT ON Products

        FOR EACH ROW

        BEGIN

          INSERT INTO Product_Metadata (product_id)

          VALUES (NEW.product_id)

          ON DUPLICATE KEY UPDATE last_modified = CURRENT_TIMESTAMP;

        END;

        """;

      // Execute the SQL statement to create the trigger

      statement.executeUpdate(createTriggerSQL);

      System.out.println("Product metadata trigger created successfully.");


    } catch (Exception e) {

      System.out.println("Error creating product trigger:");
```

```
            e.printStackTrace();

        }

    }

}
```

// SalaryLogTrigger.java

```
/*

* Run Insert query ffrom SQL CommandLine client

* INSERT INTO Products (product_name, price) VALUES ("phone", 60000);

*/




/*

Create salary Log Table:

create table Salary_Log(log_id int primary key auto_increment,employee_id int,new_salary
decimal(10,2),change_date timestamp default current_timestamp);

*/

package com.ved;


import java.sql.Connection;

import java.sql.Statement;


public class SalaryLogTrigger {

    public static void main(String[] args) {

        try (Connection con = ConnectDB.dbConnect();

            Statement statement = con.createStatement()) {

            // Define the SQL statement to create a trigger
```

```java
        String createTriggerSQL = """

            CREATE TRIGGER update_salary_trigger

            AFTER UPDATE ON employee

            FOR EACH ROW

            BEGIN

              IF NEW.salary != OLD.salary THEN

                INSERT INTO Salary_Log (employee_id, new_salary)

                VALUES (NEW.eid, NEW.salary);

              END IF;

            END;

            """;

        // Execute the SQL statement to create the trigger

        statement.executeUpdate(createTriggerSQL);

        System.out.println("Salary update trigger created successfully.");


    } catch (Exception e) {

        System.out.println("Error creating salary trigger:");

        e.printStackTrace();

    }

  }

}

/*

Not Run Sql command:

UPDATE employee SET salary =60000  WHERE eid =6;

*/
```

MYSQL:-

- Salary Log Related Tables

```sql
create database if not exists mydb;

use mydb;

CREATE TABLE employee (
    eid INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    salary DECIMAL(10,2)
);

insert into employee values(100,'DEV',50000.00);

CREATE TABLE Salary_Log (
    log_id INT PRIMARY KEY AUTO_INCREMENT,
    employee_id INT,
    new_salary DECIMAL(10,2),
    change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (employee_id) REFERENCES employee(eid)
);


-- Create trigger for salary updates
DELIMITER //
CREATE TRIGGER update_salary_trigger
AFTER UPDATE ON employee
FOR EACH ROW
BEGIN
    IF NEW.salary != OLD.salary THEN
        INSERT INTO Salary_Log (employee_id, new_salary)
        VALUES (NEW.eid, NEW.salary);
    END IF;
END;
//
```

```sql
DELIMITER ;


-- Products Related Tables
CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    product_name VARCHAR(10),
    price DECIMAL(10,2)
);


CREATE TABLE Product_Metadata (
    product_id INT,
    last_modified TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (product_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);


-- Create trigger for product updates
DELIMITER //
CREATE TRIGGER update_last_modified_trigger
AFTER INSERT ON Products
FOR EACH ROW
BEGIN
    INSERT INTO Product_Metadata (product_id)
    VALUES (NEW.product_id)
    ON DUPLICATE KEY UPDATE last_modified = CURRENT_TIMESTAMP;
END;
//
DELIMITER ;
```

<terminated> ProductTrigger [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 5:30:08 PM – 5:30:10 PM) [pid: 13976]
Successfully connected to the database!
Product metadata trigger created successfully.