

```
package jdbc;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
import org.springframework.stereotype.Component;
import org.springframework.transaction.TransactionStatus;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.transaction.interceptor.TransactionAspectSupport;

@Component
public class CategoryManager {
    @Autowired
    JdbcTemplate jdbcTemplate;

    @Autowired
    NamedParameterJdbcTemplate namedJdbcTemplate;

    public void showCategoryCount() {
        Integer count = jdbcTemplate.queryForObject("select count(*) from categories", Integer.class);
        System.out.println("Count = " + count);
    }

    public void listCategories() {
        List<Category> categories = jdbcTemplate.query("select * from categories",
            // RowMapper (mapRow(rs, int)) is implemented using Lambda Expression
            (rs, rowNum) -> new Category(rs.getString("catcode"), rs.getString("catdesc")));

        for (Category c : categories)
            System.out.printf("%-10s %s\n", c.getCode(), c.getDescription());
    }

    public void listCategoryCodes() {
        List<String> codes = jdbcTemplate.query("select catcode from categories",
            // RowMapper is implemented using Lambda Expression
            (rs, rowNum) -> rs.getString("catcode"));

        for (String code : codes)
            System.out.println(code);
    }

    public void updateCatDesc(String code, String desc) {
        int count = jdbcTemplate.update("update categories set catdesc = ? where catcode = ?", desc, code);
        System.out.println("No. of rows updated : " + count);
    }

    public void updateCatDescWithNames(String code, String desc) {
        var params = new MapSqlParameterSource().addValue("code", code).addValue("desc", desc);

        int count = namedJdbcTemplate.update("update categories set catdesc = :desc where catcode = :code", params);
        System.out.println("No. of rows updated : " + count);
    }

    @Transactional(propagation = Propagation.REQUIRED)
    public void swapDescriptions(String code1, String code2) {
        TransactionStatus status = TransactionAspectSupport.currentTransactionStatus();
        System.out.println("Transaction Begins");

        // get description for code1
        String desc1 = jdbcTemplate.queryForObject("select catdesc from categories where catcode = ?", String.class,
            code1);

        int count = jdbcTemplate.update(
            "update categories set catdesc = (select catdesc from categories where catcode = ?) where catcode = ?",
            code2, code1);
        if (count == 0) {
            // throw new RuntimeException();
            status.setRollbackOnly();
        } else {
            System.out.println("First Update Done");

            count = jdbcTemplate.update("update categories set catdesc = ? where catcode = ?", desc1, code2);
            if (count == 0)
                status.setRollbackOnly();
            else

```

```
        System.out.println("Second update done!");
    }
    System.out.println("New Transaction : " + status.isNewTransaction());
    System.out.println("Rollback ? " + status.isRollbackOnly());
    System.out.println("Function Done!");
}

//
// @Transactional(propagation = Propagation.REQUIRES_NEW)
// public void sale(int id, int qty) {
//
//     try {
//         double price = jdbcTemplate.queryForObject("select price from products where prodid = ?", Double.class, id);
//
//         int count = jdbcTemplate.update("insert into sales(prodid,qty,amount,transdate) values(?,?,?,getdate())",
//             id, qty, qty * price);
//
//         if (count == 0) {
//             throw new IllegalArgumentException("Unable to insert into sales");
//         }
//
//         count = jdbcTemplate.update("update products set qoh = qoh - ? where prodid = ?", qty, id);
//         if (count == 0)
//             throw new IllegalArgumentException("Unable to update products");
//
//         System.out.println("Done");
//     } catch (Exception ex) {
//         throw ex;
//     }
// }

public void printCategoryAvgPrice() {
    List<CatAvgPrice> categories = jdbcTemplate.query
        ("select catcode, avg(price) as avgprice from products group by catcode",
        // RowMapper (mapRow(rs, int)) is implemented using Lambda Expression
        (rs, rowNum) -> new CatAvgPrice(rs.getString("catcode"), rs.getDouble("avgprice")));

    for (var c : categories)
        System.out.printf("%-10s  %8.2f\n", c.getCode(), c.getAvgPrice());
}
}
```