

```
package jdbc;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.transaction.interceptor.TransactionAspectSupport;

@Component
public class CategoryCRUD implements CommandLineRunner {
    @Autowired
    JdbcTemplate jdbcTemplate;

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

    public void addCategory(String code, String description) {
        jdbcTemplate.update("insert into categories values(?,?)", code, description);
        System.out.println("Added Successfully!");
    }

    public void listCategories() {
        var cats = jdbcTemplate.queryForList("select catcode as code, catdesc as description from categories");

        for (var cat : cats) {
            System.out.printf("%-10s %-30s\n",
                cat.get("code"), cat.get("description"));
        }
    }

    public void listCatObjects() {
        List<Category> categories =
            jdbcTemplate.query("select * from categories order by catcode",
                // 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());
    }

    @Transactional(propagation = Propagation.REQUIRED,
        rollbackFor = {InterruptedException.class})
    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!");
    }
}
```

```
}

@Override
public void run(String... args) throws Exception {
    // showCategoryCount();
    // addCategory("cat111", "Category 111");
    // listCategories();
    // listCatObjects();
    // swapDescriptions("cat111", "cat123");
}
}
```