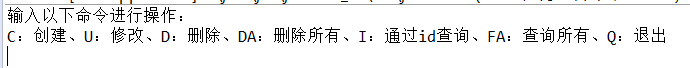
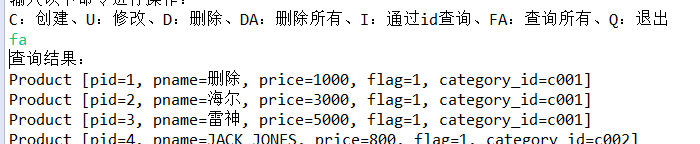
# 综合案例&阶段总结

## 案例需求

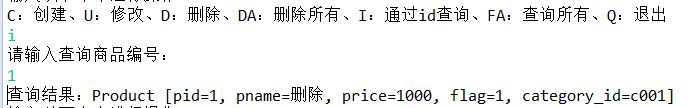
* 运行



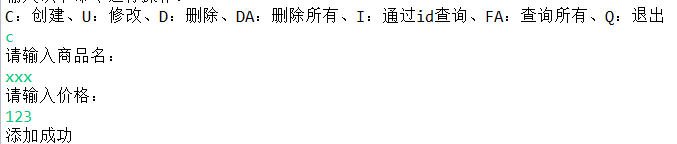
* 查询所有



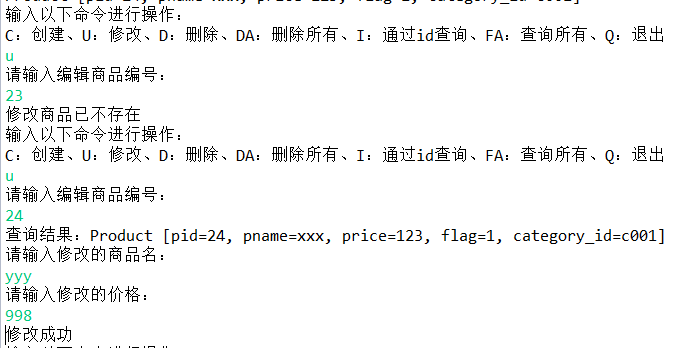
* 通过id查询



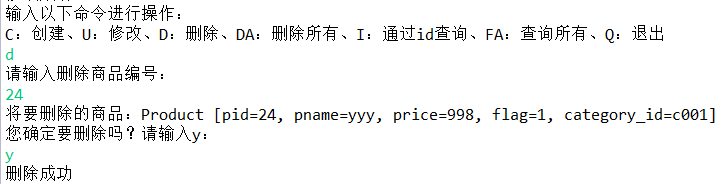
* 添加



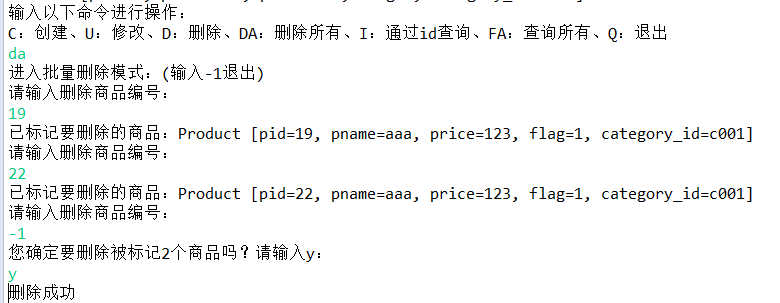
* 修改



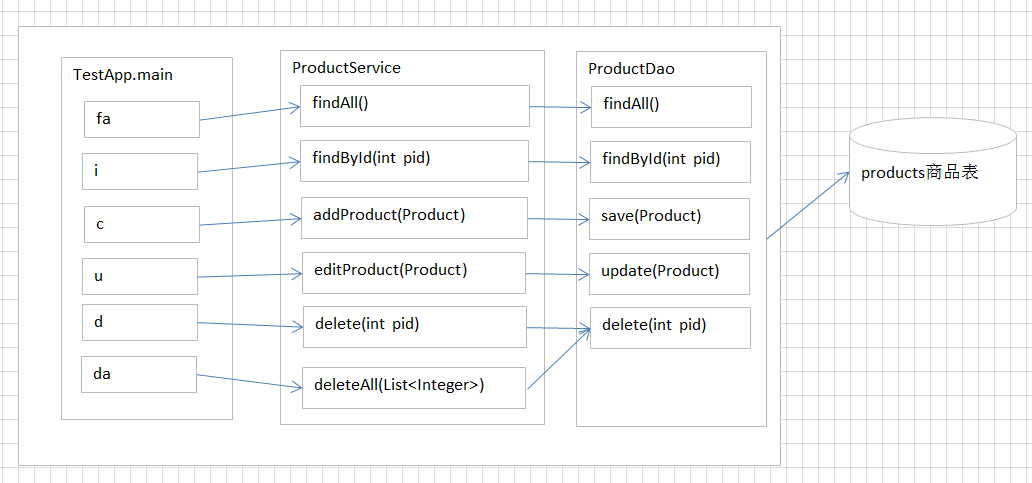
* 通过id删除



* 删除所有



## 案例分析



* 程序将划分层次
  + com.itheima.domain javaBean
  + com.itheima.utils 工具类
  + com.itheima.dao dao层
  + com.itheima.service service层（业务层，一般情况内容非常少，直接调用dao，只有与业务挂钩时才能体现出来的）

**开发步骤：**

1. 开发前准备工作
2. 创建数据库及表
3. 创建项目，添加jar包
4. 分层创建包及添加工具类
5. 实现功能：

## 代码实现

### 准备数据

create database day07;

CREATE TABLE products(

pid int PRIMARY KEY auto\_increment ,

pname VARCHAR(50),

price INT,

flag VARCHAR(2), #是否上架标记为：1表示上架、0表示下架

category\_id VARCHAR(32)

);

#商品

INSERT INTO products(pname,price,flag,category\_id) VALUES('联想',5000,'1','c001');

INSERT INTO products(pname,price,flag,category\_id) VALUES('海尔',3000,'1','c001');

INSERT INTO products(pname,price,flag,category\_id) VALUES('雷神',5000,'1','c001');

INSERT INTO products(pname,price,flag,category\_id) VALUES('JACK JONES',800,'1','c002');

INSERT INTO products(pname,price,flag,category\_id) VALUES('真维斯',200,'1','c002');

INSERT INTO products(pname,price,flag,category\_id) VALUES('花花公子',440,'1','c002');

INSERT INTO products(pname,price,flag,category\_id) VALUES('劲霸',2000,'1','c002');

INSERT INTO products(pname,price,flag,category\_id) VALUES('香奈儿',800,'1','c003');

INSERT INTO products(pname,price,flag,category\_id) VALUES('相宜本草',200,'1','c003');

### 控制台输入

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

while(true){

System.out.println("请输入以下命令进行操作：");

System.out.println("C:创建、U:修改、D:删除、DA:删除所有、QI：根据id查询、FA：查询所有、Q:退出");

String s = input.nextLine();

switch (s.toUpperCase()) {

case "C":

addProduct(input);

break;

case "U":

update(input);

break;

case "D":

deleteById(input);

break;

case "DA":

deleteAll(input);

break;

case "QI":

findById(input);

break;

case "FA":

findAll();

break;

case "Q":

System.exit(0);

break;

default:

System.out.println("输入有误，请重新输入！");

break;

}

}

}

### 查询所有

#### dao层

//查询所有

public List<Product> findAll() throws SQLException{

QueryRunner qr = new QueryRunner(C3P0Utils.getDataSource());

return qr.query("SELECT \* FROM products", new BeanListHandler<Product>(Product.class));

}

#### service层

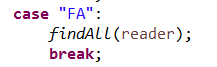
//查询所有

public List<Product> findAll() throws SQLException{

return pd.findAll();

}

#### 入口



private static void findAll(BufferedReader reader) {

ProductService productService = new ProductService();

List<Product> list = productService.findAll();

System.out.println("查询结果：");

for (Product product : list) {

System.out.println(product);

}

}

### 通过id查询详情

#### dao层

/\*\*

\* 查询详情

\* @return

\* @throws SQLException

\*/

public Product findById(Integer pid) throws SQLException{

QueryRunner queryRunner = new QueryRunner(C3P0Utils.getDataSource());

String sql = "SELECT \* FROM products where pid = ?";

Object[] params = {pid};

return queryRunner.query(sql, new BeanHandler<Product>(Product.class), params);

}

#### service层

/\*\*

\* 通过id查询

\* @param pid

\* @return

\* @throws SQLException

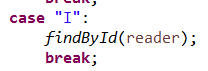
\*/

public Product findById(Integer pid) throws SQLException{

return productDao.findById(pid);

}

#### 入口



private static void findById(BufferedReader reader) {

System.out.println("请输入查询商品编号：");

String pidStr = reader.readLine();

Integer pid = Integer.parseInt(pidStr);

ProductService productService = new ProductService();

Product product = productService.findById(pid);

System.out.println("查询结果：" + product);

}

### 添加

#### dao层

/\*\*

\* 添加

\* @return

\* @throws SQLException

\*/

public void save(Product product) throws SQLException{

QueryRunner queryRunner = new QueryRunner(C3P0Utils.getDataSource());

String sql = "INSERT INTO products(pname,price,flag,category\_id) VALUES(?,?,?,?)";

Object[] params = {product.getPname(),product.getPrice(),product.getFlag(),product.getCategory\_id()};

queryRunner.update(sql, params);

}

#### service层

/\*\*

\* 添加商品

\* @param product

\* @throws SQLException

\*/

public void addProduct(Product product) throws SQLException{

productDao.save(product);

}

#### 入口



private static void add(BufferedReader reader) {

System.out.println("请输入商品名：");

String pname = reader.readLine();

System.out.println("请输入价格：");

String priceStr = reader.readLine();

int price = Integer.parseInt(priceStr);

String flag = "1";

String category\_id = "c001";

Product product = new Product(pname, price, flag, category\_id);

ProductService productService = new ProductService();

productService.addProduct(product);

System.out.println("添加成功");

}

### 修改

#### dao层

/\*\*

\* 修改

\* @return

\* @throws SQLException

\*/

public void update(Product product) throws SQLException{

QueryRunner queryRunner = new QueryRunner(C3P0Utils.getDataSource());

String sql = "update products set pname=?,price=?,flag=?,category\_id=? where pid=?";

Object[] params = {product.getPname(),product.getPrice(),product.getFlag(),

product.getCategory\_id(),product.getPid()};

queryRunner.update(sql, params);

}

#### service层

/\*\*

\* 修改商品

\* @param product

\* @throws SQLException

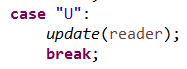
\*/

public void editProduct(Product product) throws SQLException{

productDao.update(product);

}

#### 入口



private static void update(BufferedReader reader) {

System.out.println("请输入编辑商品编号：");

String pidStr = reader.readLine();

Integer pid = Integer.parseInt(pidStr);

ProductService productService = new ProductService();

Product product = productService.findById(pid);

if(product == null){

System.out.println("修改商品已不存在");

return;

}

System.out.println("查询结果：" + product);

System.out.println("请输入修改的商品名：");

String pname = reader.readLine();

product.setPname(pname);

System.out.println("请输入修改的价格：");

String priceStr = reader.readLine();

int price = Integer.parseInt(priceStr);

product.setPrice(price);

productService.editProduct(product);

System.out.println("修改成功");

}

### 通过id删除

#### dao层

/\*\*

\* 删除

\* @return

\* @throws SQLException

\*/

public void delete(Integer pid) throws SQLException{

QueryRunner queryRunner = new QueryRunner();

String sql = "delete from products where pid=?";

Object[] params = {pid};

queryRunner.update(C3P0Utils.getConnection() , sql, params);

}

#### service层

public void delete(Integer pid) {

Connection conn = null;

try {

conn = JdbcUtils.getConnection();

//开启

conn.setAutoCommit(false);

productDao.delete(pid);

//提交

DbUtils.commitAndClose(conn);

} catch (Exception e) {

//回滚

DbUtils.rollbackAndCloseQuietly(conn);

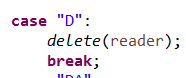
//通知调用者

throw new RuntimeException(e);

}

}

#### 入口



private static void delete(BufferedReader reader) {

System.out.println("请输入删除商品编号：");

String pidStr = reader.readLine();

Integer pid = Integer.parseInt(pidStr);

ProductService productService = new ProductService();

Product product = productService.findById(pid);

System.out.println("将要删除的商品：" + product);

System.out.println("您确定要删除吗？请输入y：");

String yes = reader.readLine();

if("y".equalsIgnoreCase(yes)){

productService.delete(pid);

System.out.println("删除成功");

} else {

System.out.println("操作取消");

}

}

### 批量删除

#### service层

/\*\*

\* 删除所有商品

\* @param product

\* @throws SQLException

\*/

public void deleteAll(List<Integer> ids) throws SQLException{

Connection conn = null;

try {

conn = C3P0Utils.getConnection();

conn.setAutoCommit(false);

for(Integer id : ids) {

productDao.delete(id);

}

DbUtils.commitAndClose(conn);

} catch (Exception e) {

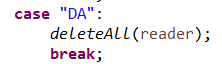
DbUtils.rollbackAndCloseQuietly(conn);

throw new RuntimeException("批量删除失败");

}

}

#### 入口



private static void deleteAll(BufferedReader reader) {

List<Integer> idList = new ArrayList<Integer>();

System.out.println("进入批量删除模式：(输入-1退出)");

ProductService productService = new ProductService();

while(true){

System.out.println("请输入删除商品编号：");

String pidStr = reader.readLine();

Integer pid = Integer.parseInt(pidStr);

if(pid == -1){

break;

}

Product product = productService.findById(pid);

if(product != null){

System.out.println("已标记要删除的商品：" + product);

idList.add(pid);

} else {

System.out.println("删除商品不存在");

}

}

System.out.println("您确定要删除被标记"+idList.size()+"个商品吗？请输入y：");

String yes = reader.readLine();

if("y".equalsIgnoreCase(yes)){

productService.deleteAll(idList);

System.out.println("删除成功");

} else {

System.out.println("操作取消");

}

}