

# 1.基本課題

## 1.実習課題の説明

基本課題として次の四つのテーブルを利用して在庫管理を行うプログラムを作成する。

(a) 商品テーブル、(b) 仕入れ先テーブル、(c) 在庫テーブル、(d) 売上テーブル

### 問題1

コマンドラインでSQLを操作することで、CREATE TABLE文により各テーブルを生成する。それぞれのテーブルは次のような名前で実装した。

(a) 商品テーブル → product  
(b) 仕入れ先テーブル → purchase  
(c) 在庫テーブル → stock  
(d) 売上テーブル → sales

また、それぞれのテーブルの列名を次のように対応させた

- (a)product(商品テーブル)
  - product\_id(商品id)
  - product\_name(商品名)
  - purchase\_id(仕入れ先id)
  - purchase\_price(仕入単価)
  - selling\_price(販売単価)
- (b)purchase(仕入先テーブル)
  - purchase\_id(仕入先id)
  - supplier\_name(仕入先名)
- (c)stock(在庫テーブル)
  - product\_id(商品id)
  - purchase\_count(仕入個数)
  - selling\_count(販売個数)
  - stock\_count(在庫個数)
- (d)sales(売上テーブル)
  - product\_id(商品id)
  - sales\_amount(売上金額)

### 問題2

以下の書式でINSERT文で所定の値を登録した

```
INSERT INTO テーブル名 (列1, 列2, 列3, ...)  
VALUES
```

```
(値1_1, 値2_1, 値3_1, ...),
(値1_2, 値2_2, 値3_2, ...),
(値1_3, 値2_3, 値3_3, ...),
-- 他の行も同様
;
```

### 問題3(仕入単価が200円以上の商品について、商品名、仕入先名、在庫数を出力せよ)

まずString型の変数urlの値をjdbc:mysql://softeng001.tu.tokuyama.ac.jp/r05db23に書き換え、userNameとpasswordも対応するものに書き換えた。

次に以下のSQL文をexecuteQuery()に挿入した

```
SELECT X.product_name, Y.supplier_name, Z.stock_count
FROM product AS X JOIN stock AS Z ON X.product_id = Z.product_id JOIN
purchase AS Y ON X.purchase_id = Y.purchase_id --結合するためのJOIN ON
WHERE X.purchase_price >= 200; -- WHERE分で仕入単価が200円以上の商品を指定
```

次にgetString()で出力したい列名を取得し,println()で出力した。

### 問題4(売上テーブルに、商品ごとの売上高を格納せよ。商品ごとの売上高と利益を出力せよ)

- ArrayListを二つ用意する。用意したのは以下の二つである。

```
ArrayList<Integer> ar = new ArrayList<Integer>();
ArrayList<Integer> arr = new ArrayList<Integer>();
```

#### while文の中

- executeQuery()に以下のsql文を挿入する

```
SELECT * FROM (product as X JOIN purchase as Y ON
X.purchase_id=Y.purchase_id JOIN stock as Z on
X.product_id=Z.product_id);
```

このSQL文は表の結合を行うが目的である

- getInt()で必要なデータを取得する
- ArrayListのメソッドadd()でデータを保存しておく
- println()で表品ごとの売上高(amount of sales)と利益(profit)を出力した

#### while文を抜けfor文の中

- executeUpdate()に以下のINSERT文を挿入し、売上テーブル(stock)に商品ごとの売上高を格納する

```
INSERT sales(product_id,sales_amount)
VALUES (" + ar.get(i) + ',' + arr.get(i) + "');
```

## 2.実行結果

### 問題1

- 商品テーブル(product)

- 作成

```
mysql> create table product(product_id INT(13),product_name varchar(25),purchase_id int(12),purchase_price int(12),selling_price int(23));
```

- 主キーの設定

```
mysql> ALTER TABLE product
-> ADD CONSTRAINT PK_product PRIMARY KEY (product_id);
Query OK, 0 rows affected (0.75 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

- 表の確認

```
mysql> desc product;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_id     | int(13)       | NO   | PRI | 0       |      |
| product_name   | varchar(25)   | YES  |     | NULL    |      |
| purchase_id    | int(12)       | YES  |     | NULL    |      |
| purchase_price | int(12)       | YES  |     | NULL    |      |
| selling_price  | int(23)       | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- 仕入れ先テーブル(purchase)

```
mysql> create table purchase( purchase_id int primary key, supplier_name varchar(20) );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'primary key, supplier_name varchar(20) )' at line 1
mysql> create table purchase( purchase_id int , supplier_name varchar(20) );
Query OK, 0 rows affected (0.33 sec)
```

```
mysql> ALTER TABLE purchase
-> ADD CONSTRAINT pk_purchase PRIMARY KEY (purchase_id);
Query OK, 0 rows affected (0.76 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc purchase;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| purchase_id    | int(11)       | NO   | PRI | 0       |      |
| supplier_name  | varchar(20)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

- 在庫テーブル(stock)

```
mysql> create table purchase( purchase_id int primary key, supplier_name varchar(20) );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'primary key, supplier_name varchar(20) )' at line 1
mysql> create table purchase( purchase_id int , supplier_name varchar(20) );
Query OK, 0 rows affected (0.33 sec)
```

```
mysql> ALTER TABLE purchase
-> ADD CONSTRAINT pk_purchase PRIMARY KEY (purchase_id);
Query OK, 0 rows affected (0.76 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc purchase;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| purchase_id    | int(11)       | NO   | PRI | 0       |      |
| supplier_name  | varchar(20)   | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

- 売上テーブル(sales)

```
mysql> create table sales( product_id int(13) , sales_amount int(12));
Query OK, 0 rows affected (0.34 sec)

mysql> ALTER TABLE sales ADD CONSTRAINT pk_product PRIMARY KEY (product_id);
Query OK, 0 rows affected (0.75 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc sales;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_id     | int(13)   | NO   | PRI | 0        |       |
| sales_amount   | int(12)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## 問題2

- 商品テーブル
  - INSERT文

```
mysql> INSERT INTO product(product_id,product_name,purchase_id,purchase_price,selling_price)
-> values
-> (1,'cookie dough ice cream',100,210,240),(2,'vanilla ice cream',100,200,220),(3,'chocolate ice cream',100,180,200),(4,'matcha ice cream',300,205,235),(5,'apple sherbet',200,170,180),(6,'strawberry ice cream',400,190,220);
```

- 表示

```
mysql> select * from product;
+-----+-----+-----+-----+-----+
| product_id | product_name          | purchase_id | purchase_price | selling_price |
+-----+-----+-----+-----+-----+
| 1          | cookie dough ice cream | 100         | 210            | 240           |
| 2          | vanilla ice cream      | 100         | 200            | 220           |
| 3          | chocolate ice cream    | 100         | 180            | 200           |
| 4          | matcha ice cream       | 300         | 205            | 235           |
| 5          | apple sherbet          | 200         | 170            | 180           |
| 6          | strawberry ice cream    | 400         | 190            | 220           |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

- 仕入れ先テーブル

```
mysql> insert into purchase(purchase_id,supplier_name)
-> values (100,'east foods'),(200,'west foods'),(300,'south foods'),(400,'north foods');
Query OK, 4 rows affected (0.06 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> select * from purchase;
+-----+-----+
| purchase_id | supplier_name |
+-----+-----+
| 100         | east foods    |
| 200         | west foods    |
| 300         | south foods   |
| 400         | north foods   |
+-----+-----+
4 rows in set (0.00 sec)
```

- 在庫テーブル

- INSERT文

```
mysql> INSERT INTO stock
-> VALUES
-> (1,50,20,2),(2,30,40,10),(3,60,60,10),(4,10,5,40);
Query OK, 4 rows affected (0.05 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO stock
-> VALUES
-> (5,50,20,30),(6,40,30,10);
Query OK, 2 rows affected (0.05 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> select * from stock;
```

- 表示

product_id	purchase_count	selling_count	stock_count
1	50	20	2
2	30	40	10
3	60	60	10
4	10	5	40
5	50	20	30
6	40	30	10

6 rows in set (0.00 sec)

### 問題3

```
i20hirota@Virtual-PC026:~/DB_enshu$ javac mondai3.java
i20hirota@Virtual-PC026:~/DB_enshu$ java mondai3
Connected
procut: cookie dough ice cream, supplier: east foods, stcok: 2
procut: vanilla ice cream, supplier: east foods, stcok: 10
procut: matcha ice cream, supplier: south foods, stcok: 40
Disconnected
```

ソースコードはpに記載

### 問題4



```
i20hirota@Virtual-PC014:~/DB_enshu$ javac mondai4.java
i20hirota@Virtual-PC014:~/DB_enshu$ java mondai4
Connected
procut: 1, amount of sales: 4800, profit: 600
procut: 2, amount of sales: 8800, profit: 800
procut: 3, amount of sales: 12000, profit: 1200
procut: 5, amount of sales: 3600, profit: 200
procut: 4, amount of sales: 1175, profit: 150
procut: 6, amount of sales: 6600, profit: 900
Disconnected
```

```
mysql> select * from sales;
Empty set (0.01 sec)

mysql> select * from sales;
+-----+-----+
| product_id | sales_amount |
+-----+-----+
|          1 |          4800 |
|          2 |          8800 |
|          3 |         12000 |
|          4 |          1175 |
|          5 |          3600 |
|          6 |          6600 |
+-----+-----+
6 rows in set (0.00 sec)
```

## エラー解析

- INSERT INTOを使うとき、挿入するものが文字列出会った場合'(シングルクォーテーション)で囲まなければならないが、忘れることが多くエラーが起きた。
- 主キーをCREATEでテーブルを作るついでに設定しようとしたが、エラーが起きてしまったため、ALTER TABLEで後から主キーへと設定した。
- JAVAでSQLを用いるとき、`source .bash_profile`を忘れてエラーが起きることがあった。
- 問題4のJAVAプログラムを作成するときに多くエラーが起きたため、以下に記す。

- INSERTは結果を返さないSQL文のため`executeQuery()`ではなく、`executeUpdate()`に挿入しなければならないが、誤って`executeQuery()`に挿入してしまった。
- `executeUpdate()`を結果を表示するwhile文の中で実行しようするとエラーが起きてしまったため、`ArrayList`に一旦保存してwhile文の実行が終わってからINSERTを実行することでエラーを避けた。
- 配列に保存しようとしたが、配列を作成するには配列の数を指定しなければならないため、数を指定しなくて良い`ArrayList`を用いた。

## 2.応用課題

### 1.データベース定義と機能(問題5)

データベース定義として以下の二つを定義する。

1. 仕入先テーブルに曜日情報を追加する
2. 収支テーブルを追加する
  - 主キー = 曜日
  - 属性 = { 曜日, 収入, 支出 }

新しい機能 → 曜日ごとの支出が分かる

### 2.実行結果(問題6と問題7)

#### 問題6

1. 仕入れ先テーブル(purchase)に曜日情報を追加する  
まず

```
ALTER TABLE purchase ADD day varchar(20);
```

で、仕入先テーブルにday(曜日)を追加し、その後

```
UPDATE purchase  
SET day = ...  
WHERE 条件;
```

で、データを更新した。実際に操作したものを以下に示す

```
mysql> use r05db23  
Database changed  
mysql> ALTER TABLE purchase  
-> ADD day varchar(20);  
Query OK, 0 rows affected (0.75 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```



```
mysql> use r05db23
Database changed
mysql> UPDATE purchase SET day = 'Sunday' WHERE purchase_id = 100;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE purchase SET day = 'Monday' WHERE purchase_id = 200;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE purchase SET day = 'Monday' WHERE purchase_id = 300;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE purchase SET day = 'Tuesday' WHERE purchase_id = 400;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from purchase;
+-----+-----+-----+
| purchase_id | supplier_name | day      |
+-----+-----+-----+
|          100 | east foods    | Sunday   |
|          200 | west foods    | Monday   |
|          300 | south foods   | Monday   |
|          400 | north foods   | Tuesday  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

## 2. 収支テーブルを追加する

**CREATE TABLE**を使い収支テーブル(newbalance)を作成した。以下に示す。

```
mysql> use r05db23
Database changed
mysql> create table balance (day varchar(20) ,income INT(12),expense INT(12));
Query OK, 0 rows affected (0.30 sec)
```

※テーブル名を最初**balance**にしていたが、予約語であるとのちに判明したため、**newbalance**に変更した

```
mysql> ALTER TABLE newbalance
-> ADD PRIMARY KEY (day);
Query OK, 0 rows affected (0.79 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc newbalance;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| day        | varchar(20)   | NO   | PRI |          |       |
| income     | int(12)        | YES  |     | NULL    |       |
| expense    | int(12)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 問題7

JAVAプログラムでnewbalanceに収入と支出をそれぞれ格納した。

問題4とほとんど同じプログラムであるため、詳細は割愛する。プログラムはpに記載する。

- JAVAプログラムの実行

```
i20hirota@Virtual-PC044:~$ source .bash_profile
i20hirota@Virtual-PC044:~$ cd DB_enshu/
i20hirota@Virtual-PC044:~/DB_enshu$ javac mondai7.java
i20hirota@Virtual-PC044:~/DB_enshu$ java mondai7
Connected
day: Monday, income: 555, expense: 615
day: Sunday, income: 410, expense: 460
day: Tuesday, income: 190, expense: 220
Disconnected
```

- 機能の確認

```
mysql> use r05db23
Database changed
mysql> select * from newbalance;
+-----+-----+-----+
| day      | income | expense |
+-----+-----+-----+
| Monday   | 555    | 615     |
| Sunday   | 410    | 460     |
| Tuesday  | 190    | 220     |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

# 所感

## ソースコード

### 問題3のプログラム

```
import java.sql.*;
class mondai3{
    public static void main(String[] args)
    {
        Connection conn = null;
        String url = "jdbc:mysql://softeng001.tu.tokuyama.ac.jp/r05db23";

        String userName = "guest23";
        String password = "tokuya23";
        try
        {
            conn = DriverManager.getConnection(url, userName, password);
            System.out.println("Connected");
            tryQuery(conn);
        }
        catch(SQLException se)
        {
            se.printStackTrace();
            System.err.println("SQLException: " + se.getMessage ());
            System.err.println("SQLState: "+ se.getSQLState ());
            System.err.println("VendorCode: "+ se.getErrorCode ());

        }

        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            if(conn != null)
            {
                try
                {
                    conn.close();
                    System.out.println("Disconnected");
                }
                catch(SQLException se)
                {
                    se.printStackTrace();
                }
            }
        }
    }

    public static void tryQuery(Connection conn) throws Exception
    {

```

```
Statement s = conn.createStatement();
// s.executeUpdate("UPDATE friends SET name='sumire' WHERE id=1");

ResultSet rs = s.executeQuery("SELECT X.product_name,
Y.supplier_name, Z.stock_count FROM product AS X JOIN stock AS Z ON
X.product_id = Z.product_id JOIN purchase AS Y ON X.purchase_id =
Y.purchase_id WHERE X.purchase_price >= 200; ");

while (rs.next())
{
    String product_name = rs.getString("product_name");
    String supplier_name = rs.getString("supplier_name");
    String stock_count = rs.getString("stock_count");

    System.out.println("procut: " + product_name + ", supplier: " +
supplier_name + ", stcok: " + stock_count );
}
s.close();
}
```

## 問題4のプログラム

```
import java.sql.*;
import java.util.*;
class mondai4{
    public static void main(String[] args)
    {
        Connection conn = null;
        String url = "jdbc:mysql://softeng001.tu.tokuyama.ac.jp/r05db23";

        String userName = "guest23";
        String password = "tokuya23";
        try
        {
            conn = DriverManager.getConnection(url, userName, password);
            System.out.println("Connected");
            tryQuery(conn);
        }
        catch(SQLException se)
        {
            se.printStackTrace();
            System.err.println("SQLException: " + se.getMessage ());
            System.err.println("SQLState: "+ se.getSQLState ());
            System.err.println("VendorCode: "+ se.getErrorCode ());
        }

        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            if(conn != null)
            {
                try
                {
                    conn.close();
                    System.out.println("Disconnected");
                }
                catch(SQLException se)
                {
                    se.printStackTrace();
                }
            }
        }
    }

    public static void tryQuery(Connection conn) throws Exception
    {
        Statement s = conn.createStatement();
```



```
// s.executeUpdate("UPDATE friends SET name='sumire' WHERE id=1");

ArrayList<Integer> ar = new ArrayList<Integer>();
ArrayList<Integer> arr = new ArrayList<Integer>();

ResultSet rs = s.executeQuery("SELECT * FROM (product as X JOIN
purchase as Y ON X.purchase_id=Y.purchase_id JOIN stock as Z on
X.product_id=Z.product_id);");

while (rs.next())
{
    int id = rs.getInt("product_id");
    int selling_price = rs.getInt("selling_price");
    int purchase_price = rs.getInt("purchase_price");
    int count = rs.getInt("selling_count");

    ar.add(id);
    arr.add(selling_price*count);
    System.out.println("product: " + id + ", amount of sales: " +
selling_price*count + ", profit: " + (selling_price*count -
purchase_price*count) );
}

for(int i=0;i<ar.size();i++){
    s.executeUpdate("INSERT sales(product_id,sales_amount) VALUES (" +
ar.get(i) + ',' + arr.get(i) + ");");
}
s.close();
}
```

## 問題7のプログラム

```
import java.sql.*;
import java.util.*;
class mondai7{
    public static void main(String[] args)
    {
        Connection conn = null;
        String url = "jdbc:mysql://softeng001.tu.tokuyama.ac.jp/r05db23";

        String userName = "guest23";
        String password = "tokuya23";
        try
        {
            conn = DriverManager.getConnection(url, userName, password);
            System.out.println("Connected");
            tryQuery(conn);
        }
        catch(SQLException se)
        {
            se.printStackTrace();
            System.err.println("SQLException: " + se.getMessage ());
            System.err.println("SQLState: "+ se.getSQLState ());
            System.err.println("VendorCode: "+ se.getErrorCode ());
        }

        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            if(conn != null)
            {
                try
                {
                    conn.close();
                    System.out.println("Disconnected");
                }
                catch(SQLException se)
                {
                    se.printStackTrace();
                }
            }
        }
    }

    public static void tryQuery(Connection conn) throws Exception
    {
        Statement s = conn.createStatement();
```

```
// s.executeUpdate("UPDATE friends SET name='sumire' WHERE id=1");

ArrayList<String> ar = new ArrayList<String>();
ArrayList<Integer> arr = new ArrayList<Integer>();
ArrayList<Integer> arrr = new ArrayList<Integer>();
ResultSet rs = s.executeQuery("SELECT day,sum(purchase_price) as
income,sum(selling_price) as expense FROM (product as X JOIN purchase as Y
ON X.purchase_id=Y.purchase_id JOIN stock as Z on
X.product_id=Z.product_id) GROUP BY day;");

while (rs.next())
{
    String day = rs.getString("day");
    int income = rs.getInt("income");
    int expense = rs.getInt("expense");

    ar.add(day);
    arr.add(income);
    arrr.add(expense);
    System.out.println("day: " + day + ", income: " + income + ",
expense: " + expense );
}

for(int i=0;i<ar.size();i++){
    s.executeUpdate("INSERT INTO newbalance(day,income,expense) VALUES
('" + ar.get(i) + "', " + arr.get(i) + ", " + arrr.get(i) + ");");
}

s.close();
}
```