```
import java.util.Scanner;
import java.sql.*;
import java.io.*;
public class CSCI3170Proj {

	public static String dbAddress = "jdbc:mysql://projgw.cse.cuhk.edu.hk:2312/db00";
	public static String dbUsername = "Group00";
	public static String dbPassword = "CSCI3170";

	public static Connection connectToOracle(){
		Connection con = null;
		try{
			Class.forName("com.mysql.jdbc.Driver");
			con = DriverManager.getConnection(dbAddress, dbUsername, dbPassword);
		} catch (ClassNotFoundException e){
			System.out.println("[Error]: Java MySQL DB Driver not found!!");
			System.exit(0);
		} catch (SQLException e){
			System.out.println(e);
		}
		return con;
	}

	public static void createTables(Connection mySQLDB) throws SQLException{
		String categorySQL = "CREATE TABLE category (";
		categorySQL += "c_id INT PRIMARY KEY NOT NULL,";
		categorySQL += "c_name VARCHAR(20) NOT NULL,";
		categorySQL += "CHECK (c_id BETWEEN 1 AND 9))";

		String manufacturerSQL = "CREATE TABLE manufacturer (";
		manufacturerSQL += "m_id INT PRIMARY KEY NOT NULL,";
		manufacturerSQL += "m_name VARCHAR(20) NOT NULL,";
		manufacturerSQL += "m_addr VARCHAR(50) NOT NULL,";
		manufacturerSQL += "m_phone INT NOT NULL,";
		manufacturerSQL += "CHECK (m_id BETWEEN 1 AND 99),";
		manufacturerSQL += "CHECK (m_phone BETWEEN 10000000 AND 99999999))";

		String salespersonSQL = "CREATE TABLE salesperson (";
		salespersonSQL += "s_id INT PRIMARY KEY NOT NULL,";
		salespersonSQL += "s_name VARCHAR(20) NOT NULL,";
		salespersonSQL += "s_addr VARCHAR(50) NOT NULL,";
		salespersonSQL += "s_phone INT NOT NULL,";
		salespersonSQL += "s_experience INT NOT NULL,";
		salespersonSQL += "CHECK (s_id BETWEEN 1 AND 99),";
		salespersonSQL += "CHECK (s_phone BETWEEN 10000000 AND 99999999),";
		salespersonSQL += "CHECK (s_experience BETWEEN 1 AND 9))";

		String partSQL = "CREATE TABLE part (";
		partSQL += "p_id INT PRIMARY KEY NOT NULL,";
		partSQL += "p_name VARCHAR(20) NOT NULL,";
		partSQL += "p_price INT NOT NULL,";
		partSQL += "m_id INT NOT NULL,";
		partSQL += "c_id INT NOT NULL,";
		partSQL += "p_quantity INT NOT NULL,";
		partSQL += "p_warranty INT NOT NULL,";
		partSQL += "FOREIGN KEY (m_id) REFERENCES manufacturer(m_id),";
		partSQL += "FOREIGN KEY (c_id) REFERENCES category(c_id),";
		partSQL += "CHECK (p_id BETWEEN 1 AND 999),";
		partSQL += "CHECK (p_price BETWEEN 1 AND 99999),";
		partSQL += "CHECK (p_warranty BETWEEN 1 AND 99),";
		partSQL += "CHECK (p_quantity BETWEEN 0 AND 99))";

		String transactionSQL = "CREATE TABLE transaction (";
		transactionSQL += "t_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,";
		transactionSQL += "p_id INT NOT NULL,";
		transactionSQL += "s_id INT NOT NULL,";
		transactionSQL += "t_date DATE NOT NULL,";
		transactionSQL += "FOREIGN KEY (p_id) REFERENCES part(p_id),";
		transactionSQL += "FOREIGN KEY (s_id) REFERENCES salesperson(s_id),";
		transactionSQL += "CHECK (t_id BETWEEN 1 AND 9999))";

		Statement stmt  = mySQLDB.createStatement();
		System.out.print("Processing...");
```

```
                //System.err.println("Creating Category Table.");
                stmt.execute(categorySQL);

                //System.err.println("Creating Manufacturer Table.");
                stmt.execute(manufacturerSQL);

                //System.err.println("Creating Salesperson Table.");
                stmt.execute(salespersonSQL);

                //System.err.println("Creating Part Table.");
                stmt.execute(partSQL);

                //System.err.println("Creating Transaction Table.");
                stmt.execute(transactionSQL);
                System.out.println("Done! Database is initialized!");
                stmt.close();
        }

        public static void deleteTables(Connection mySQLDB) throws SQLException{
                Statement stmt  = mySQLDB.createStatement();
                System.out.print("Processing...");
                stmt.execute("SET FOREIGN_KEY_CHECKS = 0;");
                stmt.execute("DROP TABLE IF EXISTS category");
                stmt.execute("DROP TABLE IF EXISTS manufacturer");
                stmt.execute("DROP TABLE IF EXISTS part");
                stmt.execute("DROP TABLE IF EXISTS salesperson");
                stmt.execute("DROP TABLE IF EXISTS transaction");
                stmt.execute("SET FOREIGN_KEY_CHECKS = 1;");
                System.out.println("Done! Database is removed!");
                stmt.close();
        }

        public static void loadTables(Scanner menuAns, Connection mySQLDB) throws SQLException{

                String categorySQL = "INSERT INTO category (c_id, c_name) VALUES (?,?)";
                String manufacturerSQL = "INSERT INTO manufacturer (m_id, m_name, m_addr,
m_phone) VALUES (?,?,?,?)";
                String partSQL = "INSERT INTO part (p_id, p_name, p_price, m_id, c_id,
p_warranty, p_quantity) VALUES (?,?,?,?,?,?,?)";
                String salespersonSQL = "INSERT INTO salesperson (s_id, s_name, s_addr, s_phone,
s_experience) VALUES (?,?,?,?,?)";
                String transactionSQL = "INSERT INTO transaction (t_id, p_id, s_id, t_date)
VALUES (?,?,?,STR_TO_DATE(?,'%d/%m/%Y'))";

                String filePath = "";
                String targetTable = "";

                while(true){
                        System.out.println("");
                        System.out.print("Type in the Source Data Folder Path: ");
                        filePath = menuAns.nextLine();
                        if((new File(filePath)).isDirectory()) break;
                }

                System.out.print("Processing...");
                //System.err.println("Loading Category");
                try{
                        PreparedStatement stmt = mySQLDB.prepareStatement(categorySQL);
                        String line = null;
                        BufferedReader dataReader = new BufferedReader(new
FileReader(filePath+"/category.txt"));

                        while ((line = dataReader.readLine()) != null) {
                                String[] dataFields = line.split("\t");
                                stmt.setInt(1, Integer.parseInt(dataFields[0]));
                                stmt.setString(2, dataFields[1]);
                                stmt.addBatch();
                        }
                        stmt.executeBatch();
                        stmt.close();
                }catch (Exception e){
                        System.out.println(e);
                }
```

```
                //System.err.println("Loading Manufacturer");
                try{
                        PreparedStatement stmt = mySQLDB.prepareStatement(manufacturerSQL);
                        String line = null;
                        BufferedReader dataReader = new BufferedReader(new
FileReader(filePath+"/manufacturer.txt"));

                        while ((line = dataReader.readLine()) != null) {
                                String[] dataFields = line.split("\t");
                                stmt.setInt(1, Integer.parseInt(dataFields[0]));
                                stmt.setString(2, dataFields[1]);
                                stmt.setString(3, dataFields[2]);
                                stmt.setInt(4, Integer.parseInt(dataFields[3]));
                                stmt.addBatch();
                        }
                        stmt.executeBatch();
                        stmt.close();
                }catch (Exception e){
                        System.out.println(e);
                }

                //System.err.println("Loading Part");
                try{
                        PreparedStatement stmt = mySQLDB.prepareStatement(partSQL);

                        String line = null;
                        BufferedReader dataReader = new BufferedReader(new
FileReader(filePath+"/part.txt"));

                        while ((line = dataReader.readLine()) != null) {
                                String[] dataFields = line.split("\t");
                                stmt.setInt(1, Integer.parseInt(dataFields[0]));
                                stmt.setString(2, dataFields[1]);
                                stmt.setInt(3, Integer.parseInt(dataFields[2]));
                                stmt.setInt(4, Integer.parseInt(dataFields[3]));
                                stmt.setInt(5, Integer.parseInt(dataFields[4]));
                                stmt.setInt(6, Integer.parseInt(dataFields[5]));
                                stmt.setInt(7, Integer.parseInt(dataFields[6]));
                                stmt.addBatch();
                        }

                        stmt.executeBatch();
                        stmt.close();
                }catch (Exception e){
                        System.out.println(e);
                }

                //System.err.println("Loading Salesperson");
                try{
                        PreparedStatement stmt = mySQLDB.prepareStatement(salespersonSQL);
                        String line = null;
                        BufferedReader dataReader = new BufferedReader(new
FileReader(filePath+"/salesperson.txt"));

                        while ((line = dataReader.readLine()) != null) {
                                String[] dataFields = line.split("\t");
                                stmt.setInt(1, Integer.parseInt(dataFields[0]));
                                stmt.setString(2, dataFields[1]);
                                stmt.setString(3, dataFields[2]);
                                stmt.setInt(4, Integer.parseInt(dataFields[3]));
                                stmt.setInt(5, Integer.parseInt(dataFields[4]));
                                stmt.addBatch();
                        }
                        stmt.executeBatch();
                        stmt.close();
                }catch (Exception e){
                        System.out.println(e);
                }

                //System.err.println("Loading Transaction");
                try{
                        PreparedStatement stmt = mySQLDB.prepareStatement(transactionSQL);
                        String line = null;
```

```
                        BufferedReader dataReader = new BufferedReader(new
FileReader(filePath+"/transaction.txt"));

                        while ((line = dataReader.readLine()) != null) {
                                String[] dataFields = line.split("\t");
                                stmt.setInt(1, Integer.parseInt(dataFields[0]));
                                stmt.setInt(2, Integer.parseInt(dataFields[1]));
                                stmt.setInt(3, Integer.parseInt(dataFields[2]));
                                stmt.setString(4, dataFields[3]);

                                //System.err.println("Record: " + i);
                                stmt.addBatch();
                                //i++;
                        }
                        stmt.executeBatch();
                        stmt.close();
                }catch (Exception e){
                        System.out.println(e);
                }

                System.out.println("Done! Data is inputted to the database!");
        }

        public static void showTables(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String[] table_name = {"category", "manufacturer", "part", "salesperson",
"transaction"};

                System.out.println("Number of records in each table:");
                for (int i = 0; i < 5; i++){
                        Statement stmt  = mySQLDB.createStatement();
                        ResultSet rs = stmt.executeQuery("select count(*) from "+table_name[i]);

                        rs.next();
                        System.out.println(table_name[i]+": "+rs.getString(1));
                        rs.close();
                        stmt.close();
                }
        }

        public static void adminMenu(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String answer = null;

                while(true){
                        System.out.println();
                        System.out.println("-----Operations for administrator menu-----");
                        System.out.println("What kinds of operation would you like to perform?");
                        System.out.println("1. Create all tables");
                        System.out.println("2. Delete all tables");
                        System.out.println("3. Load from datafile");
                        System.out.println("4. Show number of records in each table");
                        System.out.println("5. Return to the main menu");
                        System.out.print("Enter Your Choice: ");
                        answer = menuAns.nextLine();


if(answer.equals("1")||answer.equals("2")||answer.equals("3")||answer.equals("4")||answer.equals(
"5"))
                                break;
                        System.out.println("[Error]: Wrong Input, Type in again!!!");
                }

                if(answer.equals("1")){
                        createTables(mySQLDB);
                }else if(answer.equals("2")){
                        deleteTables(mySQLDB);
                }else if(answer.equals("3")){
                        loadTables(menuAns, mySQLDB);
                }else if(answer.equals("4")){
                        showTables(menuAns, mySQLDB);
                }
        }

        public static void searchParts(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String ans = null, keyword = null, method = null, ordering = null;
                String searchSQL = "";
```

```
                String searchSQL =   ;
                PreparedStatement stmt = null;

                searchSQL += "SELECT P.p_id, P.p_name, M.m_name, C.c_name, P.p_quantity,
P.p_warranty, P.p_price ";
                searchSQL += "FROM part P, manufacturer M, category C ";
                searchSQL += "WHERE P.m_id = M.m_id AND P.c_id = C.c_id ";

                while(true){
                        System.out.println("Choose the Search criterion:");
                        System.out.println("1. Part Name");
                        System.out.println("2. Manufacturer Name");
                        System.out.print("Choose the search criterion: ");
                        ans = menuAns.nextLine();
                        if(ans.equals("1")||ans.equals("2")) break;
                }
                method = ans;

                while(true){
                        System.out.print("Type in the Search Keyword:");
                        ans = menuAns.nextLine();
                        if(!ans.isEmpty()) break;
                }
                keyword = ans;

                while(true){
                        System.out.println("Choose ordering:");
                        System.out.println("1. By price, ascending order");
                        System.out.println("2. By price, descending order");
                        System.out.print("Choose the search criterion: ");
                        ans = menuAns.nextLine();
                        if(ans.equals("1")||ans.equals("2")) break;
                }
                ordering = ans;

                if(method.equals("1")){
                        searchSQL += " AND P.p_name LIKE ? ";
                }else if(method.equals("2")){
                        searchSQL += " AND M.m_name LIKE ? ";
                }

                if(ordering.equals("1")){
                        searchSQL += " ORDER BY P.p_price ASC";
                }else if(ordering.equals("2")){
                        searchSQL += " ORDER BY P.p_price DESC";
                }

                stmt = mySQLDB.prepareStatement(searchSQL);
                stmt.setString(1, "%" + keyword + "%");

                String[] field_name = {"ID", "Name", "Manufacturer", "Category", "Quantity",
"Warranty", "Price"};
                for (int i = 0; i < 7; i++){
                         System.out.print("| " + field_name[i] + " ");
                }
                System.out.println("|");

                ResultSet resultSet = stmt.executeQuery();
                while(resultSet.next()){
                        for (int i = 1; i <= 7; i++){
                                System.out.print("| " + resultSet.getString(i) + " ");
                        }
                        System.out.println("|");
                }
                System.out.println("End of Query");
                resultSet.close();
                stmt.close();
        }

        public static void sellProducts(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String updateProductSQL = "UPDATE part set p_quantity = p_quantity - 1 WHERE p_id
= ? and p_quantity > 0";
                String insertRecordSQL = "INSERT INTO transaction VALUES (NULL, ?, ?,
CURDATE())";
                String remainQuantitySQL = "SELECT p_id, p_name, p_quantity FROM part WHERE p_id
```

```
                String remainQuantitySQL = "SELECT p_id, p_name, p_quantity FROM part WHERE p_id
= ?";

                String p_id = null, s_id = null;

                while(true){
                        System.out.print("Enter The Part ID: ");
                        p_id = menuAns.nextLine();
                        if(!p_id.isEmpty()) break;
                }

                while(true){
                        System.out.print("Enter The Salesperson ID: ");
                        s_id = menuAns.nextLine();
                        if(!s_id.isEmpty()) break;
                }

                PreparedStatement stmt = mySQLDB.prepareStatement(updateProductSQL);
                stmt.setString(1, p_id);

                int retVal = stmt.executeUpdate();
                if(retVal == 0){
                        System.err.println("[Error]: This Product is currently out of stock");
                        return;
                }
                stmt.close();

                PreparedStatement stmt2 = mySQLDB.prepareStatement(insertRecordSQL);
                stmt2.setString(1, p_id);
                stmt2.setString(2, s_id);
                stmt2.executeUpdate();
                stmt2.close();

                PreparedStatement stmt3 = mySQLDB.prepareStatement(remainQuantitySQL);
                stmt3.setString(1, p_id);
                ResultSet resultSet = stmt3.executeQuery();
                resultSet.next();
                System.out.println("Product: "+ resultSet.getString(2) + "(id: " +
resultSet.getString(1) + ") Remaining Quality: " + resultSet.getString(3));

                stmt3.close();
        }

        public static void staffMenu(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String answer = "";

                while(true){
                        System.out.println();
                        System.out.println("-----Operations for salesperson menu-----");
                        System.out.println("What kinds of operation would you like to perform?");
                        System.out.println("1. Search for parts");
                        System.out.println("2. Sell a part");
                        System.out.println("3. Return to the main menu");
                        System.out.print("Enter Your Choice: ");
                        answer = menuAns.nextLine();

                        if(answer.equals("1")||answer.equals("2")||answer.equals("3"))
                                break;
                        System.out.println("[Error]: Wrong Input, Type in again!!!");
                }

                if(answer.equals("1")){
                        searchParts(menuAns, mySQLDB);
                }else if(answer.equals("2")){
                        sellProducts(menuAns, mySQLDB);
                }
        }

        public static void countSalespersonRecord(Scanner menuAns, Connection mySQLDB) throws
SQLException{
                String recordSQL = "SELECT S.s_id, S.s_name, S.s_experience, COUNT(T.t_id) ";
                recordSQL += "FROM transaction T, salesperson S ";
                recordSQL += "WHERE T.s_id = S.s_id AND S.s_experience >= ? AND S.s_experience <=
? ";
                recordSQL += "GROUP BY S.s_id, S.s_name, S.s_experience ";
```

```
                recordSQL += "ORDER BY S.s_id DESC";

                String expBegin = null, expEnd = null;

                while(true){
                        System.out.print("Type in the lower bound for years of experience: ");
                        expBegin = menuAns.nextLine();
                        if(!expBegin.isEmpty()) break;
                }

                while(true){
                        System.out.print("Type in the upper bound for years of experience: ");
                        expEnd = menuAns.nextLine();
                        if(!expEnd.isEmpty()) break;
                }

                PreparedStatement stmt  = mySQLDB.prepareStatement(recordSQL);
                stmt.setInt(1, Integer.parseInt(expBegin));
                stmt.setInt(2, Integer.parseInt(expEnd));

                ResultSet resultSet = stmt.executeQuery();

                System.out.println("Transaction Record:");

                System.out.println("| ID | Name | Years of Experience | Number of Transaction
|");
                while(resultSet.next()){
                        for (int i = 1; i <= 4; i++){
                                System.out.print("| " + resultSet.getString(i) + " ");
                        }
                        System.out.println("|");
                }
                System.out.println("End of Query");
        }

        public static void showPopularPart(Scanner menuAns, Connection mySQLDB) throws
SQLException{
                String ans;
                int booknum = 0, i = 0;
                String sql = "SELECT P.p_id, P.p_name, count(*) "+
                                        "FROM part P, transaction T "+
                                        "WHERE P.p_id = T.p_id "+
                                        "GROUP BY P.p_id, P.p_name "+
                                        "ORDER BY count(*) DESC";

                while(true){
                        System.out.print("Type in the number of parts: ");
                        ans = menuAns.nextLine();
                        if(!ans.isEmpty()) break;
                }

                booknum = Integer.parseInt(ans);
                Statement stmt  = mySQLDB.createStatement();
                ResultSet resultSet = stmt.executeQuery(sql);
                System.out.println("| Part ID | Part Name | No. of Transaction |");
                while(resultSet.next() && i < booknum){
                        System.out.println( "| " + resultSet.getString(1) + " " +
                                                                "| " + resultSet.getString(2) + "
" +
                                                                "| " + resultSet.getString(3) + "
" +
                                                                "|");
                        i++;
                }
                System.out.println("End of Query");
                stmt.close();
        }

        public static void showTotalSales(Scanner menuAns, Connection mySQLDB) throws
SQLException{
                String sql = "SELECT M.m_id, M.m_name, SUM(P.p_price) as total_sum "+
                                        "FROM transaction T, part P, manufacturer M " +
                                        "WHERE T.p_id = P.p_id AND P.m_id = M.m_id " +
                                        "GROUP BY M.m_id, M.m_name " +
```

```
                                        "ORDER by total_sum DESC";

                Statement stmt  = mySQLDB.createStatement();
                ResultSet resultSet = stmt.executeQuery(sql);

                System.out.println("| Manufacturer ID | Manufacturer Name | Total Sales Value
|");
                while(resultSet.next()){
                        System.out.println(     "| " + resultSet.getString(1) + " " +
                                                        "| " + resultSet.getString(2) + "
" +
                                                        "| " + resultSet.getString(3) + "
" +
                                                        "|");
                }
                System.out.println("End of Query");
                stmt.close();
        }

        public static void managerMenu(Scanner menuAns, Connection mySQLDB) throws SQLException{
                String answer = "";

                while(true){
                        System.out.println();
                        System.out.println("-----Operations for manager menu-----");
                        System.out.println("What kinds of operation would you like to perform?");
                        System.out.println("1. Count the no. of sales record of each salesperson
under a specific range on years of experience");
                        System.out.println("2. Show the total sales value of each manufacturer");
                        System.out.println("3. Show the N most popular part");
                        System.out.println("4. Return to the main menu");
                        System.out.print("Enter Your Choice: ");
                        answer = menuAns.nextLine();


if(answer.equals("1")||answer.equals("2")||answer.equals("3")||answer.equals("4"))
                                break;
                        System.out.println("[Error]: Wrong Input, Type in again!!!");
                }

                if(answer.equals("1")){
                        countSalespersonRecord(menuAns, mySQLDB);
                }else if(answer.equals("2")){
                        showTotalSales(menuAns, mySQLDB);
                }else if(answer.equals("3")){
                        showPopularPart(menuAns, mySQLDB);
                }
        }

        public static void main(String[] args) {
                Scanner menuAns = new Scanner(System.in);
                System.out.println("Welcome to sales system!");

                while(true){
                        try{
                                Connection mySQLDB = connectToOracle();
                                System.out.println();
                                System.out.println("-----Main menu-----");
                                System.out.println("What kinds of operation would you like to
perform?");
                                System.out.println("1. Operations for administrator");
                                System.out.println("2. Operations for salesperson");
                                System.out.println("3. Operations for manager");
                                System.out.println("4. Exit this program");
                                System.out.print("Enter Your Choice: ");

                                String answer = menuAns.nextLine();

                                if(answer.equals("1")){
                                        adminMenu(menuAns, mySQLDB);
                                }else if(answer.equals("2")){
                                        staffMenu(menuAns, mySQLDB);
                                }else if(answer.equals("3")){
                                        managerMenu(menuAns, mySQLDB);
```

```
                                }else if(answer.equals("4")){
                                        break;
                                }else{
                                        System.out.println("[Error]: Wrong Input, Type in
again!!!");
                                }
                        }catch (SQLException e){
                                System.out.println(e);
                        }
                }

                menuAns.close();
                System.exit(0);
        }
}
```