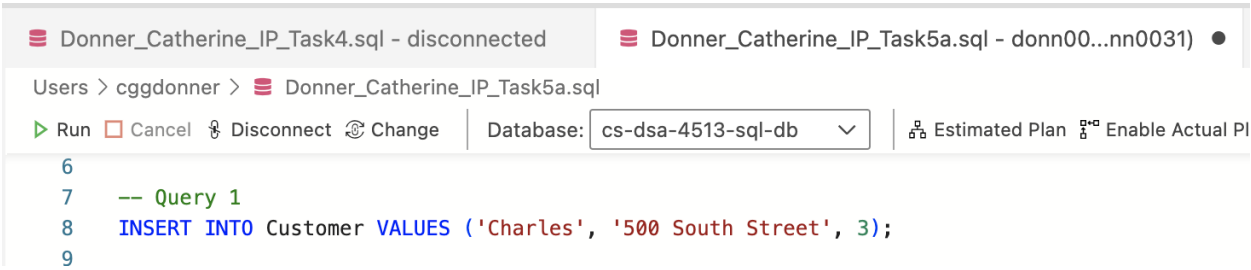
TASK 5

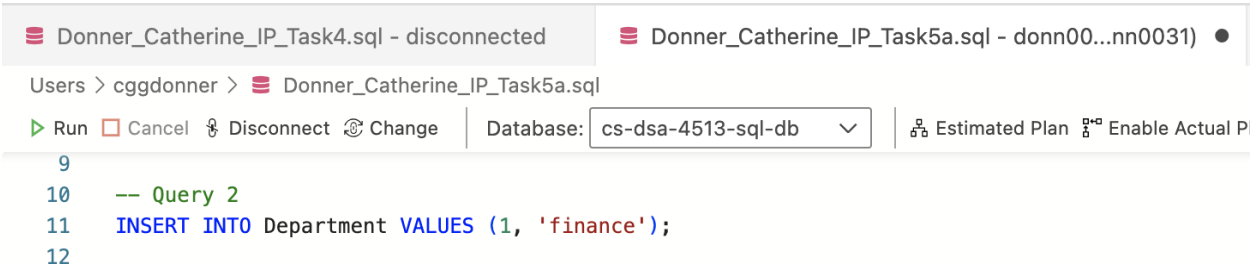
5.1 SQL Statements for implementing all queries (1-14 and error checking)

Note that these are queries with sample values and were implemented using JBDC program with ? to substitute values

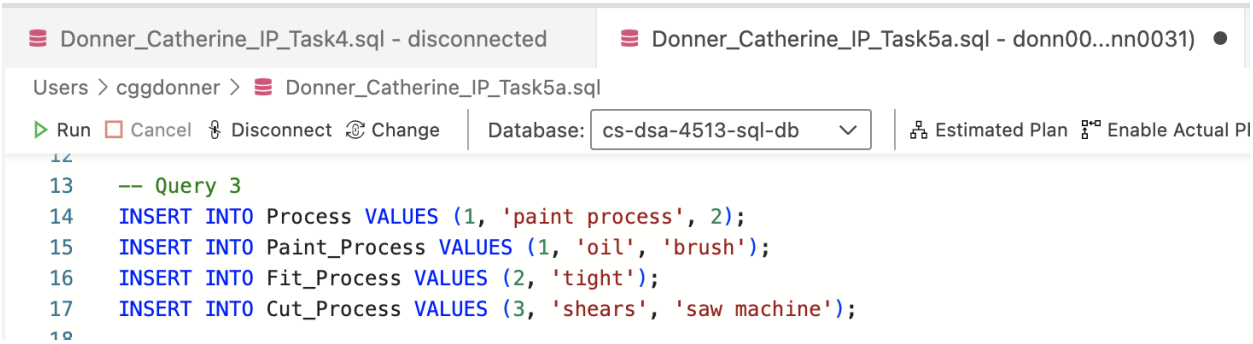
Query 1



Query 2

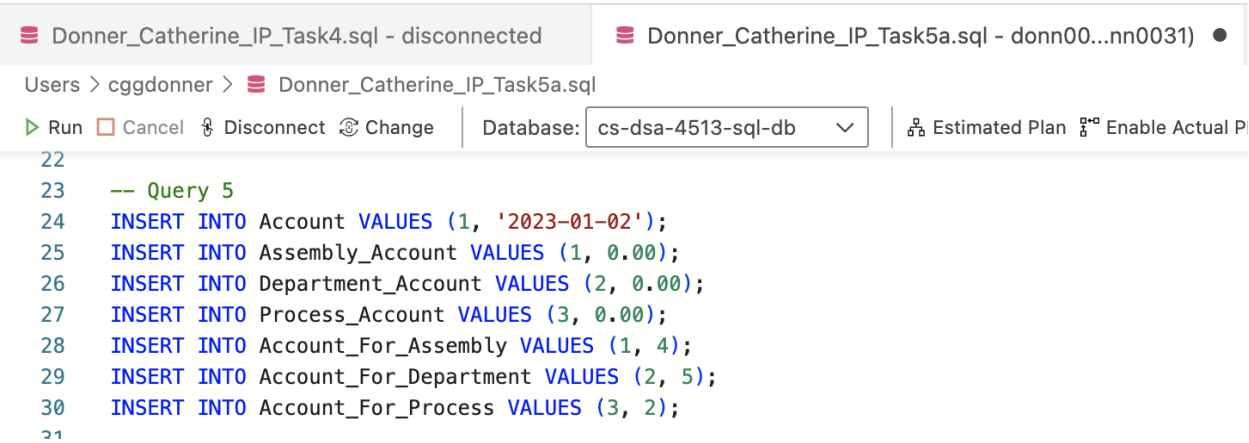


Query 3

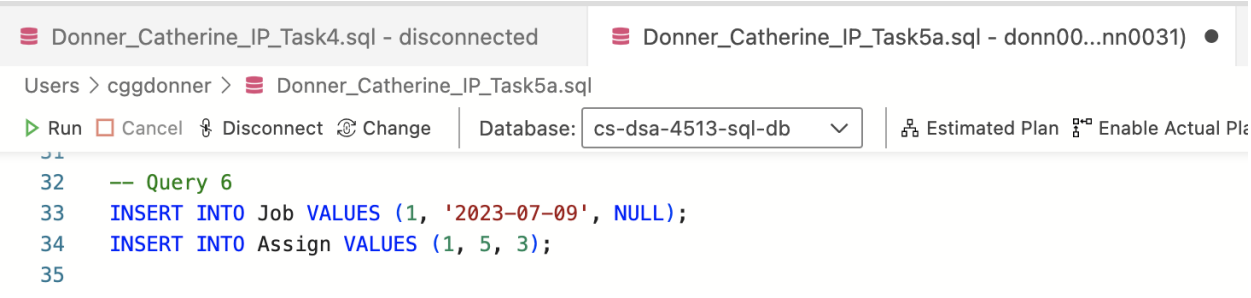


Query 4

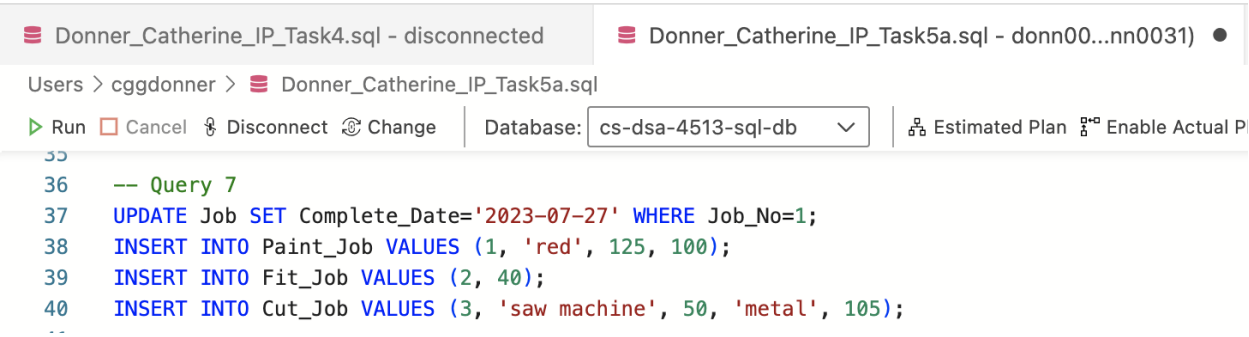
Query 5



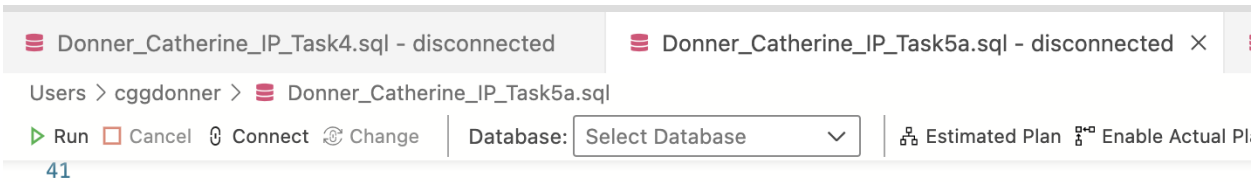
Query 6



Query 7



Query 8




```
41
42    -- Query 8
43    INSERT INTO Cost_Transaction VALUES (1, 90.04, 6, 1);
44    UPDATE Assembly_Account SET Details_1 = (Details_1 + 90.04) WHERE Account_No=1;
45    UPDATE Department_Account SET Details_2 = (Details_2 + 87.92) WHERE Account_No=2;
46    UPDATE Process_Account SET Details_3 = (Details_3 + 430.29) WHERE Account_No=6;
47
```

Query 9


```
48    -- Query 9
49    SELECT SUM(Sup_Cost) FROM Cost_Transaction T
50    JOIN Account_For_Assembly A ON A.Account_No = T.Account_No WHERE A.Assembly_Id=6;
51
```

Query 10

Donner_Catherine_IP_Task4.sql - disconnected ● | Donner_Catherine_IP_Task5a.sql - disconnected ×

Users > cggdonner > Donner_Catherine_IP_Task5a.sql

Run | Cancel | Connect | Change | Database: Select Database | Estimated Plan | Enable Actual

```sql
-- Query 10
WITH AllJobLaborTimes AS (
    SELECT P.Department_Number, Labor_Time
    FROM Paint_Job PJ
    JOIN Job J ON J.Job_No = PJ.Job_No
    JOIN Assign A ON A.Job_No = J.Job_No
    JOIN Process P ON P.Process_Id = A.Process_Id
    WHERE J.Complete_Date = '2023-09-30'

    UNION

    SELECT P.Department_Number, Labor_Time
    FROM Fit_Job FJ
    JOIN Job J ON J.Job_No = FJ.Job_No
    JOIN Assign A ON A.Job_No = J.Job_No
    JOIN Process P ON P.Process_Id = A.Process_Id
    WHERE J.Complete_Date = '2023-09-30'

    UNION

    SELECT P.Department_Number, Labor_Time
    FROM Cut_Job CJ
    JOIN Job J ON J.Job_No = CJ.Job_No
    JOIN Assign A ON A.Job_No = J.Job_No
    JOIN Process P ON P.Process_Id = A.Process_Id
    WHERE J.Complete_Date = '2023-09-30'
)
SELECT Department_Number, SUM(Labor_Time)
FROM AllJobLaborTimes
GROUP BY Department_Number;
```

Query 11

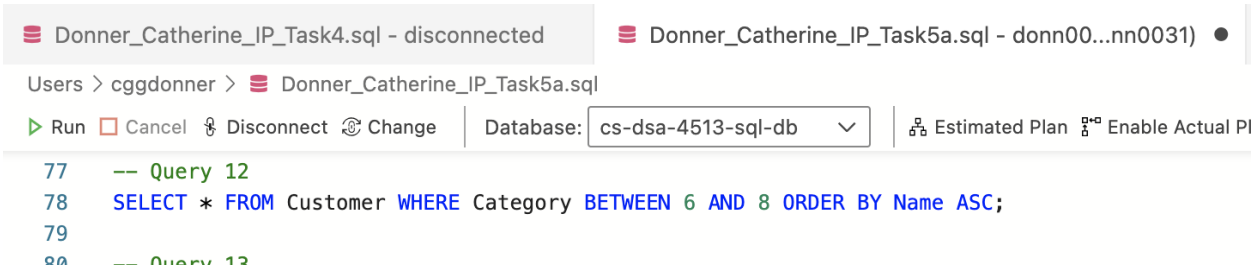
Donner_Catherine_IP_Task4.sql - disconnected | Donner_Catherine_IP_Task5a.sql - donn00...nn0031) ●

Users > cggdonner > Donner_Catherine_IP_Task5a.sql

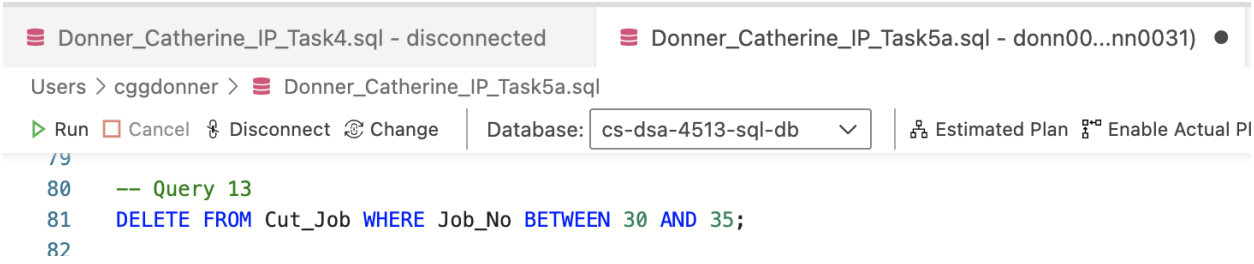
Run | Cancel | Disconnect | Change | Database: cs-dsa-4513-sql-db | Estimated Plan | Enable Actual P

```sql
-- Query 11
SELECT P.*, J.Commence_Date FROM Process P
JOIN Assign A ON A.Process_Id = P.Process_Id
JOIN Job J ON J.Job_No = A.Job_No WHERE A.Assembly_Id=1
ORDER BY J.Commence_Date;
```

Query 12

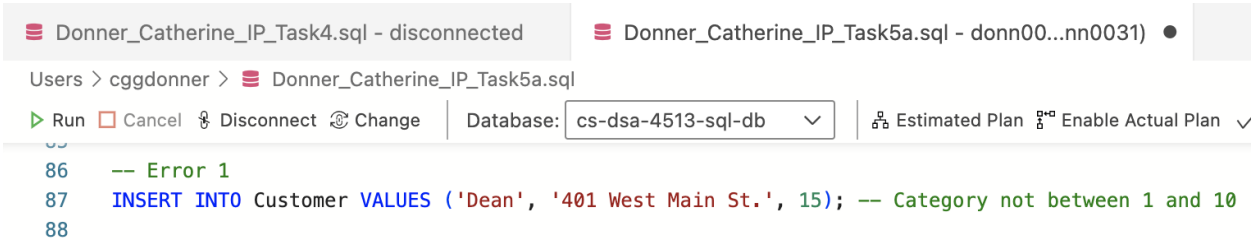


Query 13

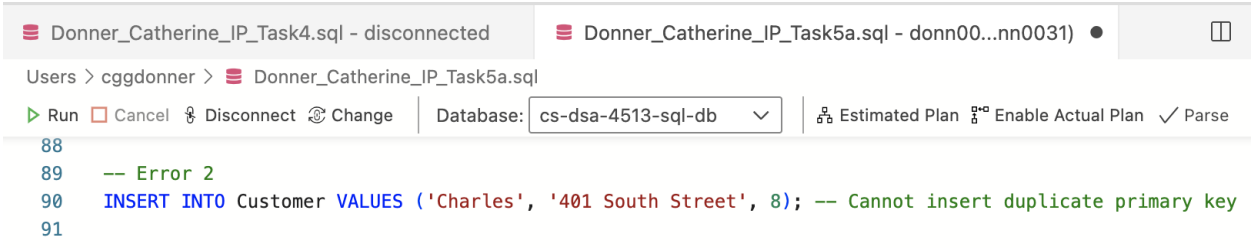


Query 14

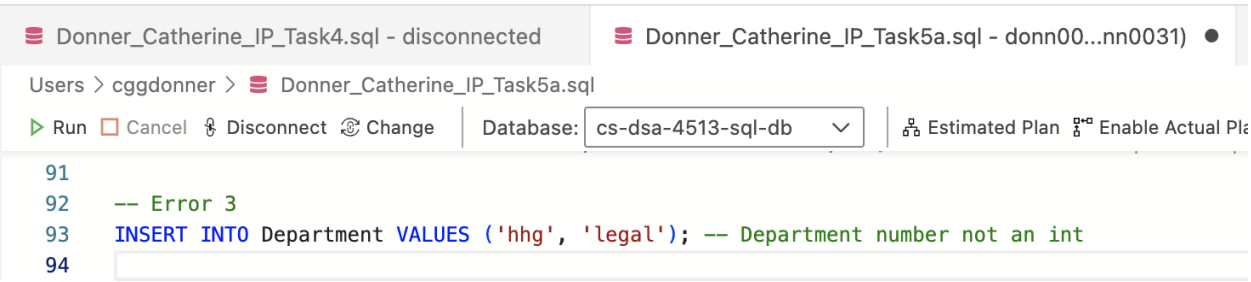
Error 1: Category Check Constraint Violation



Error 2: Primary Key Violation



Error 3: Datatype Mismatch Error



5.2 Java source program and screenshots showing successful compilation

```
// Catherine Donner
// DBMS Individual Project Task 5b
import java.sql.Connection;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
public class Donner_Catherine_IP_Task5b {
        // Database credentials
        // INSERT YOUR OWN CREDENTIALS
        final static String HOSTNAME = "donn0031-sql-server.database.windows.net";
        final static String DBNAME = "cs-dsa-4513-sql-db";
        final static String USERNAME = "donn0031";
        final static String PASSWORD = "RoSaLiNa5000$";
        // Database connection string
```

```
        final static String URL =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trust
ServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
HOSTNAME, DBNAME, USERNAME, PASSWORD);

        // User input prompt

        final static String PROMPT =

        "\nWELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM \n" +

        "(1) Enter a new customer; \n" +

        "(2) Enter a new department; \n" +

        "(3) Enter a new process; \n" +

        "(4) Enter a new assembly; \n" +

        "(5) Create a new account; \n" +

        "(6) Enter a new job; \n" +

        "(7) Enter a completed job; \n" +

        "(8) Enter a new transaction; \n" +

        "(9) Retrieve the total cost from an assembly; \n" +

        "(10) Retrieve the total labor time within a department for jobs completed; \n" +

        "(11) Retrieve the processes and departments responsible for processes; \n" +

        "(12) Retrieve the customers whose category is in a given range; \n" +

        "(13) Delete all cut-jobs with job-no in a given range; \n" +

        "(14) Change the color of a paint job; \n" +

        "(15) Import: Enter new customers from a data file until the file is empty; \n" +

        "(16) Export: Retrieve the customers (in name order) whose category is in a given range
and output them to a data file; \n" +

        "(17) Quit;";

        // Query statements

        // Query 1

        final static String QUERY_1 = "INSERT INTO Customer VALUES (?, ?, ?);";

        // Query 2
```

```
final static String QUERY_2 = "INSERT INTO Department VALUES (?, ?);";
// Query 3
final static String QUERY_3_1 = "INSERT INTO Process VALUES (?, ?, ?);";
final static String QUERY_3_2 = "INSERT INTO Paint_Process VALUES (?, ?, ?);";
final static String QUERY_3_3 = "INSERT INTO Fit_Process VALUES (?, ?);";
final static String QUERY_3_4 = "INSERT INTO Cut_Process VALUES (?, ?, ?);";
// Query 4
final static String QUERY_4_1 = "INSERT INTO Assembly VALUES (?, ?, ?, ?);";
final static String QUERY_4_2 = "INSERT INTO Begin_Manufacture VALUES (?, ?);";
// Query 5
final static String QUERY_5_1 = "INSERT INTO Account VALUES (?, ?);";
final static String QUERY_5_2 = "INSERT INTO Assembly_Account VALUES (?, ?);";
final static String QUERY_5_3 = "INSERT INTO Department_Account VALUES (?, ?);";
final static String QUERY_5_4 = "INSERT INTO Process_Account VALUES (?, ?);";
final static String QUERY_5_5 = "INSERT INTO Account_For_Assembly VALUES (?, ?);";
final static String QUERY_5_6 = "INSERT INTO Account_For_Department VALUES (?, ?);";
final static String QUERY_5_7 = "INSERT INTO Account_For_Process VALUES (?, ?);";
// Query 6
final static String QUERY_6_1 = "INSERT INTO Job VALUES (?, ?, NULL);";
final static String QUERY_6_2 = "INSERT INTO Assign VALUES (?, ?, ?);";
// Query 7
final static String QUERY_7_1 = "UPDATE Job SET Complete_Date=? WHERE Job_No=?;";
final static String QUERY_7_2 = "INSERT INTO Paint_Job VALUES (?, ?, ?, ?);";
final static String QUERY_7_3 = "INSERT INTO Fit_Job VALUES (?, ?);";
final static String QUERY_7_4 = "INSERT INTO Cut_Job VALUES (?, ?, ?, ?, ?);";
// Query 8
final static String QUERY_8_1 = "INSERT INTO Cost_Transaction VALUES (?, ?, ?, ?);";
```

```
final static String QUERY_8_2 = "UPDATE Assembly_Account SET Details_1 = (Details_1 +
?) WHERE Account_No=?;";

final static String QUERY_8_3 = "UPDATE Department_Account SET Details_2 =
(Details_2 + ?) WHERE Account_No=?;";

final static String QUERY_8_4 = "UPDATE Process_Account SET Details_3 = (Details_3 +
?) WHERE Account_No=?;";

// Query 9

final static String QUERY_9 = "SELECT SUM(Sup_Cost) FROM Cost_Transaction T JOIN
Account_For_Assembly A ON A.Account_No = T.Account_No WHERE A.Assembly_Id=?;";

// Query 10

final static String QUERY_10 = "WITH AllJobLaborTimes AS (\n"
                + "    SELECT P.Department_Number, Labor_Time\n"
                + "    FROM Paint_Job PJ\n"
                + "    JOIN Job J ON J.Job_No = PJ.Job_No\n"
                + "    JOIN Assign A ON A.Job_No = J.Job_No\n"
                + "    JOIN Process P ON P.Process_Id = A.Process_Id\n"
                + "    WHERE J.Complete_Date = ?\n"
                + "\n"
                + "    UNION\n"
                + "\n"
                + "    SELECT P.Department_Number, Labor_Time\n"
                + "    FROM Fit_Job FJ\n"
                + "    JOIN Job J ON J.Job_No = FJ.Job_No\n"
                + "    JOIN Assign A ON A.Job_No = J.Job_No\n"
                + "    JOIN Process P ON P.Process_Id = A.Process_Id\n"
                + "    WHERE J.Complete_Date = ?\n"
                + "\n"
                + "    UNION\n"
                + "\n"
```

```
                + "    SELECT P.Department_Number, Labor_Time\n"
                + "    FROM Cut_Job CJ\n"
                + "    JOIN Job J ON J.Job_No = CJ.Job_No\n"
                + "    JOIN Assign A ON A.Job_No = J.Job_No\n"
                + "    JOIN Process P ON P.Process_Id = A.Process_Id\n"
                + "    WHERE J.Complete_Date = ?\n"
                + ")\n"
                + "SELECT Department_Number, SUM(Labor_Time)\n"
                + "FROM AllJobLaborTimes\n"
                + "GROUP BY Department_Number;";
        // Query 11
        final static String QUERY_11 = "SELECT P.*, J.Commence_Date FROM Process P JOIN
Assign A ON A.Process_Id = P.Process_Id JOIN Job J ON J.Job_No = A.Job_No WHERE
A.Assembly_Id=? ORDER BY J.Commence_Date;";
        // Query 12
        final static String QUERY_12 = "SELECT * FROM Customer WHERE Category BETWEEN ?
AND ? ORDER BY Name ASC;";
        // Query 13
        final static String QUERY_13 = "DELETE FROM Cut_Job WHERE Job_No BETWEEN ? AND
?;";
        // Query 14
        final static String QUERY_14 = "UPDATE Paint_Job SET Color=? WHERE Job_No=?;";
        // Initialize application
        public static void main(String[] args) throws SQLException {
                final Scanner sc = new Scanner(System.in);
                String option = "";
                // While application is not quit
                while (!option.equals("17")) {
                        System.out.println(PROMPT);
```

```
                option = sc.next();
                switch (option) {
                        case "1":
                                System.out.println("Please enter customer name
(varchar):"); // enter name
                                sc.nextLine();
                                final String name = sc.nextLine();
                                System.out.println("Please enter customer address
(varchar):"); // enter address
                                //sc.nextLine();
                                final String address = sc.nextLine();
                                System.out.println("Please enter customer category (1-
10):"); // enter category
                                final int category = sc.nextInt();
                                System.out.println("Connecting to the database...");
                                // Insert values
                                try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_1)) {
                                                // Populate Customer with the data
collected from the user
                                                statement.setString(1, name);
statement.setString(2, address);
                                                statement.setInt(3, category);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();
```

```
        System.out.println(String.format("Done. %d rows inserted into Customer.",
rows_inserted));
                                                                    }
                                                                }
                        break;
                        case "2":
                                System.out.println("Please enter department number:");
// enter dept_no
                                final int dept_no = sc.nextInt();
                                System.out.println("Please enter department data:"); //
enter dept_data
                                sc.nextLine();
                                final String dept_data = sc.nextLine();
                                System.out.println("Connecting to the database...");
                                // Insert values
                                try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_2)) {
                                                // Populate Department with the
data collected from the user
                                                statement.setInt(1, dept_no);
statement.setString(2, dept_data);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Department.",
rows_inserted));
```

```
                                                                }
                                                            }
                        break;
                        case "3":
                                System.out.println("Please enter process ID:"); // enter
process_id
                                final int process_id = sc.nextInt();
                                System.out.println("Please enter process data (varchar):");
// enter process_data
                                sc.nextLine();
                                final String process_data = sc.nextLine();
                                System.out.println("Please enter department number:");
// enter dept_no_2
                                final int dept_no_2 = sc.nextInt();
                                System.out.println("Connecting to the database...");
                                // Insert values
                                try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_3_1)) {
                                                // Populate Process with the data
collected from the user
                                                statement.setInt(1, process_id);
statement.setString(2, process_data);
                                                statement.setInt(3, dept_no_2);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();
```

```
        System.out.println(String.format("Done. %d rows inserted into Process.",
rows_inserted));
                                                                    }
                                                                }
                                    System.out.println("Please enter process type: paint, fit, or
cut"); // enter process type to determine where data is inserted
                                    sc.nextLine();
                                    final String process_type = sc.nextLine();
                                    if (process_type.equals("paint")) {
                                        //System.out.println("Please enter paint process
ID:"); // enter process_id_paint
                                        //final int process_id_paint = sc.nextInt();
                                        System.out.println("Please enter paint type
(varchar):"); // enter paint_type
                                        final String paint_type = sc.nextLine();
                                        System.out.println("Please enter painting method
(varchar):"); // enter painting_method
                                        //sc.nextLine();
                                        final String painting_method = sc.nextLine();
                                        System.out.println("Connecting to the
database...");
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final
PreparedStatement statement = connection.prepareStatement(QUERY_3_2)) {
                                                    // Populate Paint_Process
with the data collected from the user
```

```
                                                    statement.setInt(1,
process_id); statement.setString(2, paint_type);
                                                    statement.setString(3,
painting_method);

        System.out.println("Dispatching the query...");
                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Paint_Process.",
rows_inserted));
                                                                }
                                                            }
                                    } else if (process_type.equals("fit")) {
                                            //System.out.println("Please enter fit process ID:");
// enter process_id_fit
                                            //final int process_id_fit = sc.nextInt();
                                            System.out.println("Please enter fit type
(varchar):"); // enter fit_type
                                            //sc.nextLine();
                                            final String fit_type = sc.nextLine();
                                            System.out.println("Connecting to the
database...");
                                            // Insert values
                                            try (final
                                                    Connection connection =
DriverManager.getConnection(URL)) { try (
                                                            final
PreparedStatement statement = connection.prepareStatement(QUERY_3_3)) {
                                                    // Populate Fit_Process with
the data collected from the user
```

```
                                                    statement.setInt(1,
process_id); statement.setString(2, fit_type);

        System.out.println("Dispatching the query...");

                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Fit_Process.",
rows_inserted));

                                                                }

                                                            }

                                    } else if (process_type.equals("cut")) {

                                        //System.out.println("Please enter cut process
ID:"); // enter process_id_cut

                                        //final int process_id_cut = sc.nextInt();

                                        System.out.println("Please enter cutting type
(varchar):"); // enter cutting_type

                                        //sc.nextLine();

                                        final String cutting_type = sc.nextLine();

                                        System.out.println("Please enter machine type
(varchar):"); // enter machine_type

                                        //sc.nextLine();

                                        final String machine_type = sc.nextLine();

                                        System.out.println("Connecting to the
database...");

                                        // Insert values

                                        try (final

                                                Connection connection =
DriverManager.getConnection(URL)) { try (

                                                                final
PreparedStatement statement = connection.prepareStatement(QUERY_3_4)) {
```

```
                                                    // Populate Cut_Process with
the data collected from the user
                                                    statement.setInt(1,
process_id); statement.setString(2, cutting_type);
                                                    statement.setString(3,
machine_type);

        System.out.println("Dispatching the query...");
                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Cut_Process.",
rows_inserted));
                                                                }
                                                            }
                                    } else {
                                            System.out.println("Make sure the input process
type is paint, fit, or cut with no spaces, special characters, or capital letters. Please try again.");
// error exception
                                    }
                                break;
                                case "4":
                                    System.out.println("Please enter assembly ID:"); // enter
assembly_id
                                    final int assembly_id = sc.nextInt();
                                    System.out.println("Please enter date ordered
(MM/DD/YYYY):"); // enter date_ordered
                                    sc.nextLine();
                                    final String date_ordered = sc.nextLine();
                                    System.out.println("Please enter assembly details
(varchar):"); // enter assembly_details
                                    //sc.nextLine();
```

```
                                    final String assembly_details = sc.nextLine();
                                    System.out.println("Please enter customer name
(varchar):"); // enter cust_name
                                    //sc.nextLine();
                                    final String cust_name = sc.nextLine();
                                    System.out.println("Connecting to the database...");
                                    // Insert values
                                    try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_4_1)) {
                                                    // Populate Assembly with the data
collected from the user
                                                    statement.setInt(1, assembly_id);
statement.setString(2, date_ordered);
                                                    statement.setString(3,
assembly_details); statement.setString(4, cust_name);
                                                    System.out.println("Dispatching the
query...");
                                                            final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Assembly.",
rows_inserted));
                                                            }
                                                    }
                                    //System.out.println("Please enter assembly ID:"); // enter
assembly_id_2
                                    //final int assembly_id_2 = sc.nextInt();
                                    System.out.println("Please enter process ID to associate
with assembly:"); // enter process_id_2
```

```
                                    final int process_id_2 = sc.nextInt();
                                    System.out.println("Connecting to the database...");
                                    // Insert values
                                    try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_4_2)) {
                                                    // Populate Begin_Manufacture with
the data collected from the user
                                                    statement.setInt(1, assembly_id);
statement.setInt(2, process_id_2);
                                                    System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Begin_Manufacture.",
rows_inserted));
                                                        }
                                                }
                            break;
                            case "5":
                                    System.out.println("Please enter account number:"); //
enter acct_no
                                    final int acct_no = sc.nextInt();
                                    System.out.println("Please enter establish date
(MM/DD/YYYY):"); // enter est_date
                                    sc.nextLine();
                                    final String est_date = sc.nextLine();
                                    System.out.println("Connecting to the database...");
```

```
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_5_1)) {
                                                // Populate Account with the data
collected from the user
                                                statement.setInt(1, acct_no);
statement.setString(2, est_date);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Account.",
rows_inserted));
                                                        }
                                                }
                                        System.out.println("Please enter account type: assembly,
department, or process"); // enter account type to determine where data is inserted
                                        //sc.nextLine();
                                        final String account_type = sc.nextLine();
                                        if (account_type.equals("assembly")) {
                                                //System.out.println("Please enter assembly
account number:"); // enter acct_no_assemb
                                                //final int acct_no_assemb = sc.nextInt();
                                                System.out.println("Please enter assembly account
details (float):"); // enter details_1
                                                //sc.nextLine();
                                                final float details_1 = sc.nextFloat();
```

```
                                        System.out.println("Connecting to the
database...");

                                        // Insert values

                                        try (final

                                                Connection connection =
DriverManager.getConnection(URL)) { try (

                                                        final
PreparedStatement statement = connection.prepareStatement(QUERY_5_2)) {

                                                // Populate
Assembly_Account with the data collected from the user

                                                statement.setInt(1, acct_no);
statement.setFloat(2, details_1);


        System.out.println("Dispatching the query...");

                                                        final int
rows_inserted = statement.executeUpdate();


        System.out.println(String.format("Done. %d rows inserted into Assembly_Account.",
rows_inserted));

                                                        }

                                                }

                                        //System.out.println("Please enter assembly
account number:"); // enter acct_no_assemb_2

                                        //final int acct_no_assemb_2 = sc.nextInt();

                                        System.out.println("Please enter assembly ID to
associate account with:"); // enter acct_assembly_id

                                        final int acct_assembly_id = sc.nextInt();

                                        // Insert values

                                        try (final

                                                Connection connection =
DriverManager.getConnection(URL)) { try (
```

```
                                                                    final
PreparedStatement statement = connection.prepareStatement(QUERY_5_5)) {
                                                                // Populate
Account_For_Assembly with the data collected from the user
                                                                statement.setInt(1, acct_no);
statement.setInt(2, acct_assembly_id);

        System.out.println("Dispatching the query...");
                                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into
Account_For_Assembly.", rows_inserted));
                                                                        }
                                                                    }
                                    } else if (account_type.equals("department")) {
                                            //System.out.println("Please enter department
account number:"); // enter acct_no_dept
                                            //final int acct_no_dept = sc.nextInt();
                                            System.out.println("Please enter department
account details (float):"); // enter details_2
                                            //sc.nextLine();
                                            final float details_2 = sc.nextFloat();
                                            System.out.println("Connecting to the
database...");
                                            // Insert values
                                            try (final
                                                    Connection connection =
DriverManager.getConnection(URL)) { try (
                                                                    final
PreparedStatement statement = connection.prepareStatement(QUERY_5_3)) {
```

```
                                                        // Populate
Department_Account with the data collected from the user

                                                        statement.setInt(1, acct_no);
statement.setFloat(2, details_2);

        System.out.println("Dispatching the query...");

                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Department_Account.",
rows_inserted));

                                                                }

                                                            }

                                        //System.out.println("Please enter department
account number:"); // enter acct_no_dept_2

                                        //final int acct_no_dept_2 = sc.nextInt();

                                        System.out.println("Please enter department
number to associate account with:"); // enter acct_dept_no

                                        final int acct_dept_no = sc.nextInt();

                                        System.out.println("Connecting to the
database...");

                                        // Insert values

                                        try (final

                                                Connection connection =
DriverManager.getConnection(URL)) { try (

                                                                final
PreparedStatement statement = connection.prepareStatement(QUERY_5_6)) {

                                                        // Populate
Account_For_Department with the data collected from the user

                                                        statement.setInt(1, acct_no);
statement.setInt(2, acct_dept_no);

        System.out.println("Dispatching the query...");
```

```
                                                                    final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into
Account_For_Department.", rows_inserted));
                                                                    }
                                                                }
                                    } else if (account_type.equals("process")) {
                                            //System.out.println("Please enter process account
number:"); // enter process_acct_no
                                            //final int process_acct_no = sc.nextInt();
                                            System.out.println("Please enter process account
details (float):"); // enter details_3
                                            //sc.nextLine();
                                            final float details_3 = sc.nextFloat();
                                            System.out.println("Connecting to the
database...");
                                            // Insert values
                                            try (final
                                                    Connection connection =
DriverManager.getConnection(URL)) { try (
                                                                    final
PreparedStatement statement = connection.prepareStatement(QUERY_5_4)) {
                                                            // Populate Process_Account
with the data collected from the user
                                                            statement.setInt(1, acct_no);
statement.setFloat(2, details_3);

        System.out.println("Dispatching the query...");
                                                                    final int
rows_inserted = statement.executeUpdate();
```

```
        System.out.println(String.format("Done. %d rows inserted into Process_Account.",
rows_inserted));
                                                                        }
                                                                }
                                        //System.out.println("Please enter process account
number:"); // enter process_acct_no_2
                                        //final int process_acct_no_2 = sc.nextInt();
                                        System.out.println("Please enter process ID to
associate account with:"); // enter acct_process_id
                                        final int acct_process_id = sc.nextInt();
                                        System.out.println("Connecting to the
database...");
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                                final
PreparedStatement statement = connection.prepareStatement(QUERY_5_7)) {
                                                        // Populate
Account_For_Process with the data collected from the user
                                                        statement.setInt(1, acct_no);
statement.setInt(2, acct_process_id);

        System.out.println("Dispatching the query...");
                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Account_For_Process.",
rows_inserted));
                                                                        }
                                                                }
```

```
                    } else {
                        System.out.println("Make sure the input account type is assembly, department, or process with no spaces, special characters, or capital letters. Please try again."); // error exception
                    }
                break;
                case "6":
                    System.out.println("Please enter job number:"); // enter job_no
                    final int job_no = sc.nextInt();
                    System.out.println("Please enter commencement date of job (MM/DD/YYYY):"); // enter commence_date
                    sc.nextLine();
                    final String commence_date = sc.nextLine();
                    System.out.println("Connecting to the database...");
                    // Insert values
                    try (final
                            Connection connection = DriverManager.getConnection(URL)) { try (
                                final PreparedStatement statement = connection.prepareStatement(QUERY_6_1)) {
                        // Populate Job with the data collected from the user
                        statement.setInt(1, job_no); statement.setString(2, commence_date);
                        System.out.println("Dispatching the query...");
                            final int rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Job.", rows_inserted));
```

```
                                                                }
                                                            }
                                    System.out.println("Please enter assembly ID to associate
job with:"); // enter assign_assemb_id
                                    final int assign_assemb_id = sc.nextInt();
                                    System.out.println("Please enter process ID to associate
job with:"); // enter assign_process_id
                                    final int assign_process_id = sc.nextInt();
                                    try (final
                                            Connection connection =
DriverManager.getConnection(URL)) { try (
                                                    final PreparedStatement
statement = connection.prepareStatement(QUERY_6_2)) {
                                                // Populate Assign with the data
collected from the user
                                                statement.setInt(1, job_no);
statement.setInt(2, assign_assemb_id);
                                                statement.setInt(3,
assign_process_id);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Assign.",
rows_inserted));
                                                }
                                            }
                                break;
                                case "7":
                                    System.out.println("Please enter job number for
completed job:"); // enter job_no
```

```
                                    final int job_no_update = sc.nextInt();
                                    System.out.println("Please enter completion date of job
(MM/DD/YYYY):"); // enter complete_date
                                    sc.nextLine();
                                    final String complete_date = sc.nextLine();
                                    System.out.println("Connecting to the database...");
                                    // Insert values
                                    try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_7_1)) {
                                                    // Update Job with the data
collected from the user
                                                    statement.setString(1,
complete_date); statement.setInt(2, job_no_update);
                                                    System.out.println("Dispatching the
query...");
                                                        final int
rows_updated = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows updated in Job.", rows_updated));
                                                        }
                                                }
                                    System.out.println("Please enter job type: paint, fit, or
cut"); // enter completed job type to determine where data is inserted
                                    //sc.nextLine();
                                    final String job_type = sc.nextLine();
                                    if (job_type.equals("paint")) {
                                            //System.out.println("Please enter paint job
number:"); // enter process_id_paint
```

```
                                        //final int job_no_paint = sc.nextInt();
                                        System.out.println("Please enter paint color
(varchar):"); // enter paint_color
                                        //sc.nextLine();
                                        final String paint_color = sc.nextLine();
                                        System.out.println("Please enter volume (int):"); //
enter volume
                                        final int volume = sc.nextInt();
                                        System.out.println("Please enter labor time of job
(int):"); // enter labor_time_paint
                                        final int labor_time_paint = sc.nextInt();
                                        System.out.println("Connecting to the
database...");
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final
PreparedStatement statement = connection.prepareStatement(QUERY_7_2)) {
                                                // Populate Paint_Job with
the data collected from the user
                                                statement.setInt(1,
job_no_update); statement.setString(2, paint_color);
                                                statement.setInt(3, volume);
statement.setInt(4, labor_time_paint);

        System.out.println("Dispatching the query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Paint_Job.",
rows_inserted));
```

```
                                                                    }
                                                                }
                                        } else if (job_type.equals("fit")) {
                                                //System.out.println("Please enter fit job
number:"); // enter fit_job_no
                                                //final int fit_job_no = sc.nextInt();
                                                System.out.println("Please enter labor time of job
(int):"); // enter labor_time_fit
                                                final int labor_time_fit = sc.nextInt();
                                                System.out.println("Connecting to the
database...");
                                                // Insert values
                                                try (final
                                                        Connection connection =
DriverManager.getConnection(URL)) { try (
                                                                final
PreparedStatement statement = connection.prepareStatement(QUERY_7_3)) {
                                                        // Populate Fit_Job with the
data collected from the user
                                                        statement.setInt(1,
job_no_update); statement.setInt(2, labor_time_fit);

        System.out.println("Dispatching the query...");
                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Fit_Job.",
rows_inserted));
                                                                }
                                                            }
                                        } else if (job_type.equals("cut")) {
```

```
                                        //System.out.println("Please enter cut job
number:"); // enter cut_job_no
                                        //final int cut_job_no = sc.nextInt();
                                        System.out.println("Please enter machine type
(varchar):"); // enter machine_type_job
                                        //sc.nextLine();
                                        final String machine_type_job = sc.nextLine();
                                        System.out.println("Please enter machine time
used (int):"); // enter machine_time_used
                                        final int machine_time_used = sc.nextInt();
                                        System.out.println("Please enter material used
(varchar):"); // enter material_used
                                        sc.nextLine();
                                        final String material_used = sc.nextLine();
                                        System.out.println("Please enter labor time of job
(int):"); // enter labor_time_cut
                                        final int labor_time_cut = sc.nextInt();
                                        System.out.println("Connecting to the
database...");
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final
PreparedStatement statement = connection.prepareStatement(QUERY_7_4)) {
                                                        // Populate Cut_Job with the
data collected from the user
                                                        statement.setInt(1,
job_no_update); statement.setString(2, machine_type_job);
                                                        statement.setInt(3,
machine_time_used); statement.setString(4, material_used);
```

```
                                                    statement.setInt(5,
labor_time_cut);

        System.out.println("Dispatching the query...");

                                                                final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Cut_Job.",
rows_inserted));

                                                                }

                                                            }

                                    } else {

                                            System.out.println("Make sure the input job type is
paint, fit, or cut with no spaces, special characters, or capital letters. Please try again."); // error
exception

                                    }

                                break;

                                case "8":

                                    System.out.println("Please enter transaction number:"); //
enter trans_no

                                    final int trans_no = sc.nextInt();

                                    System.out.println("Please enter sup cost (float):"); //
enter sup_cost

                                    final float sup_cost = sc.nextFloat();

                                    System.out.println("Please enter job number associated
with transaction:"); // enter job_no_trans

                                    final int job_no_trans = sc.nextInt();

                                    System.out.println("Please enter account number
associated with transaction:"); // enter acct_no_trans

                                    final int acct_no_trans = sc.nextInt();

                                    System.out.println("Connecting to the database...");
```

```
                                        // Insert values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_8_1)) {
                                                // Populate Cost_Transaction with
the data collected from the user
                                                statement.setInt(1, trans_no);
statement.setFloat(2, sup_cost);
                                                statement.setInt(3, job_no_trans);
statement.setInt(4, acct_no_trans);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Cost_Transaction.",
rows_inserted));
                                                        }
                                                }
                                        System.out.println("Is the account that will be updated an
assembly, department, or process account?"); // enter type of account to determine where
data is inserted
                                        sc.nextLine();
                                        final String acct_to_update = sc.nextLine();
                                        if (acct_to_update.equals("assembly")) {
                                        System.out.println("Connecting to the database...");
                                        // Update values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
```

```
                                                    final PreparedStatement
statement = connection.prepareStatement(QUERY_8_2)) {
                                                    // Update Assembly_Account with
the data collected from the user
                                                    statement.setFloat(1, sup_cost);
statement.setInt(2, acct_no_trans); // Reversed indices due to update
                                                    System.out.println("Dispatching the
query...");
                                                        final int
rows_updated = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows updated in Assembly_Account.",
rows_updated));
                                                        }
                                                    }
                                } else if (acct_to_update.equals("department")) {
                                System.out.println("Connecting to the database...");
                                // Update values
                                try (final
                                        Connection connection =
DriverManager.getConnection(URL)) { try (
                                                    final PreparedStatement
statement = connection.prepareStatement(QUERY_8_3)) {
                                                    // Update Department_Account
with the data collected from the user
                                                    statement.setFloat(1, sup_cost);
statement.setInt(2, acct_no_trans); // Reversed indices due to update
                                                    System.out.println("Dispatching the
query...");
                                                        final int
rows_updated = statement.executeUpdate();
```

```
        System.out.println(String.format("Done. %d rows updated in Department_Account.",
rows_updated));
                                                                    }
                                                                }
                                    } else if (acct_to_update.equals("process")) {
                                    System.out.println("Connecting to the database...");
                                    // Update values
                                    try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_8_4)) {
                                                    // Update Process_Account with the
data collected from the user
                                                    statement.setFloat(1, sup_cost);
statement.setInt(2, acct_no_trans); // Reversed indices due to update
                                                    System.out.println("Dispatching the
query...");
                                                                final int
rows_updated = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows updated in Process_Account.",
rows_updated));
                                                                }
                                                            }
                                    } else {
                                            System.out.println("Make sure the input account
type is assembly, department, or process with no spaces, special characters, or capital letters.
Please try again."); // error exception
                                    }
                                break;
```

```
                    case "9":
                        System.out.println("Please enter assembly ID to display
total cost for:"); // enter assemb_id_cost
                        final int assemb_id_cost = sc.nextInt();
                        System.out.println("Connecting to the database...");
                        try (final Connection connection =
DriverManager.getConnection(URL)) {
                            // Display total cost
                            try (final PreparedStatement statement =
connection.prepareStatement(QUERY_9)) {
                                statement.setInt(1,
assemb_id_cost);
                                System.out.println("Dispatching the
query...");
                                final ResultSet resultSet =
statement.executeQuery();
                                System.out.println("Total cost for
assembly ID:");
                                System.out.println("total cost ");
                                while (resultSet.next()) {

        System.out.println(String.format("%s ",

        resultSet.getString(1)));
                                }
                            }
                        }
                    break;
                    // Added more records for this query
                    case "10":
```

```
                                        System.out.println("Please enter completion date of job
(MM/DD/YYYY):"); // enter complete_date_labor_time
                                        sc.nextLine();
                                        final String complete_date_labor_time = sc.nextLine();
                                        System.out.println("Connecting to the database...");
                                        try (final Connection connection =
DriverManager.getConnection(URL)) {
                                                // Display total labor time for ALL jobs in
department
                                                try (final PreparedStatement statement =
connection.prepareStatement(QUERY_10)) {
                                                        statement.setString(1,
complete_date_labor_time); statement.setString(2, complete_date_labor_time);
                                                        statement.setString(3,
complete_date_labor_time);
                                                        System.out.println("Dispatching the
query...");
                                                        final ResultSet resultSet =
statement.executeQuery();
                                                        System.out.println("Total labor time
for departments for all jobs:");
                                                        System.out.println("department
number | total labor time ");
                                                        while (resultSet.next()) {

        System.out.println(String.format("%s | %s ",
                                                                        resultSet.getString(1),
resultSet.getString(2)));
                                                }
                                        }
                                }
                        break;
```

```
                    case "11":
                        System.out.println("Please enter assembly ID to display processes and departments:"); // enter assemb_11
                        final int assemb_11 = sc.nextInt();
                        System.out.println("Connecting to the database...");
                        try (final Connection connection = DriverManager.getConnection(URL)) {
                            // Display total cost
                            try (final PreparedStatement statement = connection.prepareStatement(QUERY_11)) {
                                statement.setInt(1, assemb_11);
                                System.out.println("Dispatching the query...");
                                final ResultSet resultSet = statement.executeQuery();
                                System.out.println("Processes and departments for assembly ID:");
                                System.out.println("process id | process data | department number | commence date ");
                                while (resultSet.next()) {

        System.out.println(String.format("%s | %s | %s | %s ",
                                        resultSet.getString(1), resultSet.getString(2), resultSet.getString(3), resultSet.getString(4)));
                                }
                            }
                        }
                    break;
                    case "12":
                        System.out.println("Please enter lower category limit:"); // enter lower_category
```

```
                                    final int lower_category = sc.nextInt();
                                    System.out.println("Please enter higher category limit:");
// enter higher_category
                                    final int higher_category = sc.nextInt();
                                    System.out.println("Connecting to the database...");
                                    try (final Connection connection =
DriverManager.getConnection(URL)) {
                                        // Display customer data
                                        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_12)) {
                                            statement.setInt(1, lower_category);
statement.setInt(2, higher_category);
                                            System.out.println("Dispatching the
query...");
                                            final ResultSet resultSet =
statement.executeQuery();
                                            System.out.println("Customers
within the category range:");
                                            System.out.println("name | address
| category ");
                                            while (resultSet.next()) {

        System.out.println(String.format("%s | %s | %s ",
                                                    resultSet.getString(1),
resultSet.getString(2),

        resultSet.getString(3)));
                                                }
                                        }
                                    }
                                break;
```

```
                    // Added more records for this query
                    case "13":
                        System.out.println("Please enter lower job number
limit:"); // enter lower_job_no
                        final int lower_job_no = sc.nextInt();
                        System.out.println("Please enter higher job number
limit:"); // enter higher_job_no
                        final int higher_job_no = sc.nextInt();
                        System.out.println("Connecting to the database...");
                        // Delete values
                        try (final
                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_13)) {
                                // Delete from Cut_Job the records
between the job number limits
                                statement.setInt(1, lower_job_no);
statement.setInt(2, higher_job_no);
                                System.out.println("Dispatching the
query...");
                                        final int rows_deleted
= statement.executeUpdate();

        System.out.println(String.format("Done. %d rows deleted from Cut_Job.",
rows_deleted));
                                        }
                                }
                    break;
                    case "14":
```

```
                                        System.out.println("Please enter job number:"); // enter
job_no_paint
                                        final int job_no_paint = sc.nextInt();
                                        System.out.println("Please enter the color paint that you
want to change to for this job:"); // enter color
                                        sc.nextLine();
                                        final String color = sc.nextLine();
                                        System.out.println("Connecting to the database...");
                                        // Update values
                                        try (final
                                                Connection connection =
DriverManager.getConnection(URL)) { try (
                                                        final PreparedStatement
statement = connection.prepareStatement(QUERY_14)) {
                                                // Change the paint color
                                                statement.setString(1, color);
statement.setInt(2, job_no_paint);
                                                System.out.println("Dispatching the
query...");
                                                        final int
rows_updated = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows updated in Paint_Job.",
rows_updated));
                                                        }
                                                }
                                break;
                                case "15":
                                        System.out.println("Please enter a filename to import
customer data from:");
                        String filename_import = sc.next();
```

```
                    //sc.nextLine();
                    try {
                        File file = new File(filename_import);
                        Scanner fileReader = new Scanner(file);
                        while(fileReader.hasNext()) {
                                String line = fileReader.nextLine();
                                String[] parts = line.split(",");
                                String import_name = parts[0];
                                String import_address = parts[1];
                                int import_category = Integer.parseInt(parts[2]);
                                System.out.println(String.format("Read line %s into
database", line));
                                try (final
                                                        Connection connection =
DriverManager.getConnection(URL)) { try (
                                                                final
PreparedStatement statement = connection.prepareStatement(QUERY_1)) {
                                                                        //
Populate Customer with the data collected from the user

        statement.setString(1, import_name); statement.setString(2, import_address);

        statement.setInt(3, import_category);

        System.out.println("Dispatching the query...");
                                                                final
int rows_inserted = statement.executeUpdate();

        System.out.println(String.format("Done. %d rows inserted into Customer.",
rows_inserted));
                                                                }
```

```
                                                                    }
                                                                }
                                                    fileReader.close();
                                                                }
                                        catch (Exception e) {
                                            System.out.println(e);
                                        }
                                            break;
                                            case "16":
                                                    System.out.println("Please enter a filename to export
customer data to:");
                                        String filename_export = sc.next();
                                        //sc.nextLine();
                                                    System.out.println("Please enter lower category limit:"); //
enter lower_category
                                                    final int lower_category_2 = sc.nextInt();
                                                    System.out.println("Please enter higher category limit:");
// enter higher_category
                                                    final int higher_category_2 = sc.nextInt();
                                                    System.out.println("Connecting to the database...");
                                                    try (final Connection connection =
DriverManager.getConnection(URL)) {
                                                            FileWriter writer = new
FileWriter(filename_export);
                                                            // Display customer data
                                                            try (final PreparedStatement statement =
connection.prepareStatement(QUERY_12)) {
                                                                    statement.setInt(1,
lower_category_2); statement.setInt(2, higher_category_2);
```

```
                                                    System.out.println("Dispatching the
query...");
                                                    final ResultSet resultSet =
statement.executeQuery();
                                                    while (resultSet.next()) {

        writer.write(resultSet.getString(1) + "," + resultSet.getString(2) +
                                                                        "," +
resultSet.getString(3) + "\n");
                                                        }
                                                }
                                                System.out.println(String.format("Data written out
to %s ", filename_export));
                                                writer.close();
                                        } catch (IOException e) {
                                                System.out.println("Error writing to the file: " +
e.getMessage());
                                        }
                                break;
                                // Quit program
                                case "17":
                                        System.out.println("Exited job-shop accounting system.");
                                break;
                                default:
                                        System.out.println(String.format(
                                                        "Unrecognized option: %s\n" +
                                                                "Please try again",
                                                                option));
                                break;
                        }
```

```
            }
            sc.close(); // Close scanner
        }
}
```