Individual Project

Database Management Systems

Dr. Le Gruenwald

CS 3113 – Sec. 001

Fall 2015

Catherine Thomas

Catherine.S.Thomas-1@ou.edu

ID: 113848296

INSERT TABLE OF CONTENTS HERE

Table of Contents

Task 1 Database Design

1.1 ER Diagram

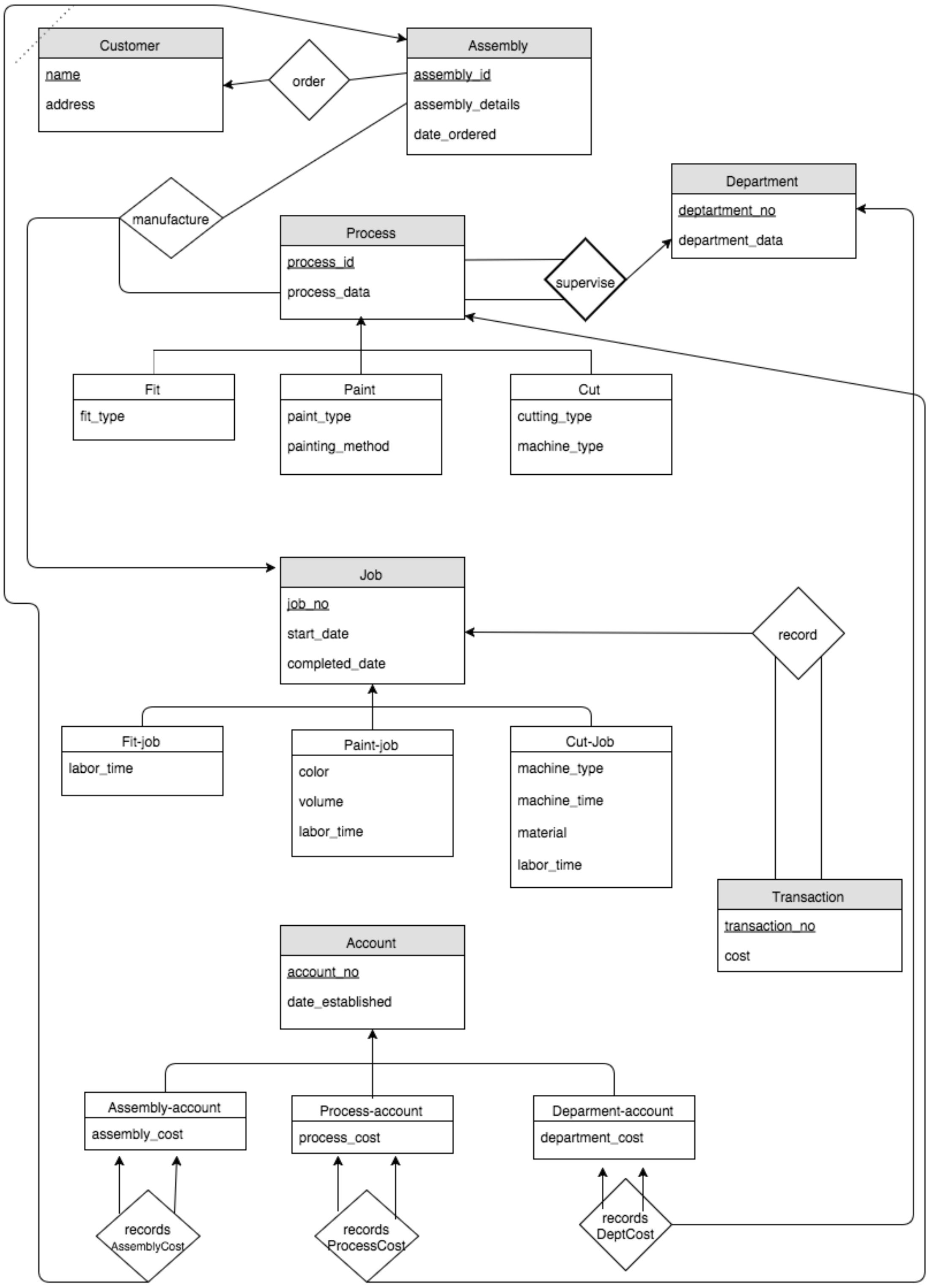


Figure 1: ER Diagram for a job-shop accounting system.

1.2 Relational Database Schema

- customer(cname, address)
- assembly(assembly_id, assembly_details, date_ordered, cname)
- department(department_no, department_data)
- transaction(transaction_no, cost, job_no)
- process(process_id , process_data, department_no)
- fitProcess(process_id, fit_type)
- paintProcess(process_id, paint_type, painting_method)
- cutProcess(process_id, cutting_type, machine_type)
- job(job_no, start_date, completed_date, process_id, assembly_id, labor_time)
- paintjob(job_no, color, volume)
- cutJob(job_no, machine_type, machine_time, material)
- account(account_no, date_established)
- assemblyAccount(account_no, assembly_cost, assembly_id)
- processAccount(account_no, process_cost, process_id)
- departmentAccount(account_no, department_cost, department_id)

Task 2 Data Dictionary

The Data Dictionary was generated through SQL Developer. Tables listed in alphabetical order.

Accounts

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ACCOUNT_NO	NUMBER	No	null	1	null
DATE_ESTABLISHED	VARCHAR2(15 BYTE)	Yes	null	2	null

Assemblies

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENT
ASSEMBLY_ID	NUMBER	No	null	1	null
ASSEMBLY_DETAILS	VARCHAR2(150 BYTE)	Yes	null	2	null
DATE_ORDERED	VARCHAR2(15 BYTE)	Yes	null	3	null
CNAME	VARCHAR2(40 BYTE)	Yes	null	4	

AssemblyAccounts

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ACCOUNT_NO	NUMBER	No	null	1	null
ASSEMBLY_COST	NUMBER	Yes	null	2	null
ASSEMBLY_ID	NUMBER	Yes	null	3	null

Customer

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
CNAME	VARCHAR2(40 BYTE)	No	null	1	null
ADDRESS	VARCHAR2(40 BYTE)	Yes	null	2	null

CutJob

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
JOB_NO	NUMBER	No	null	1	null
MACHINE_TYPE	VARCHAR2(30 BYTE)	Yes	null	2	null
MACHINE_TIME	NUMBER	Yes	null	3	null
MATERIAL	VARCHAR2(30 BYTE)	Yes	null	4	null

CutProcess

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
PROCESS_ID	NUMBER	No	null	1	null
CUTTING_TYPE	VARCHAR2(30 BYTE)	Yes	null	2	null
MACHINE_TYPE	VARCHAR2(30 BYTE)	Yes	null	3	null

Department

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DEPARTMENT_NO	NUMBER	No	null	1	null
DEPARTMENT_DATA	VARCHAR2(150 BYTE)	Yes	null	2	null

DepartmentAccount

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ACCOUNT_NO	NUMBER	No	null	1	null
DEPARTMENT_COST	NUMBER	Yes	null	2	null
DEPARTMENT_NO	NUMBER	Yes	null	3	null

FitProcess

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
PROCESS_ID	NUMBER	No	null	1	null
FIT_TYPE	VARCHAR2(30 BYTE)	Yes	null	2	null

Jobs

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
JOB_NO	NUMBER	No	null	1	null
START_DATE	VARCHAR2(15 BYTE)	Yes	null	2	null
COMPLETED_DATE	VARCHAR2(15 BYTE)	Yes	null	3	null
PROCESS_ID	NUMBER	Yes	null	4	null
ASSEMBLY_ID	NUMBER	Yes	null	5	null
LABOR_TIME	NUMBER	Yes	null	6	null

PaintJob

COLUMN_NAME	DATA_TYPE	NULLABL	DATA_DEFAULT	COLUMN_I	COMMENT

		E		**D**	**S**
JOB_NO	NUMBER	No	null	1	null
COLOR	VARCHAR2(15 BYTE)	Yes	null	2	null
VOLUME_PAINT	NUMBER	Yes	null	3	null

PaintProcess

COLUMN_NAME	**DATA_TYPE**	**NULLABLE**	**DATA_DEFAULT**	**COLUMN_ID**	**COMMENTS**
PROCESS_ID	NUMBER	No	null	1	null
PAINT_TYPE	VARCHAR2(30 BYTE)	Yes	null	2	null
PAINTING_METHOD	VARCHAR2(30 BYTE)	Yes	null	3	null

ProcessAccount

COLUMN_NAME	**DATA_TYPE**	**NULLABLE**	**DATA_DEFAULT**	**COLUMN_ID**	**COMMENTS**
ACCOUNT_NO	NUMBER	No	null	1	null
PROCESS_COST	NUMBER	Yes	null	2	null
PROCESS_ID	NUMBER	Yes	null	3	null

Processes

COLUMN_NAME	**DATA_TYPE**	**NULLABLE**	**DATA_DEFAULT**	**COLUMN_ID**	**COMMENTS**
PROCESS_ID	NUMBER	No	null	1	null
PROCESS_DATA	VARCHAR2(150 BYTE)	Yes	null	2	null
DEPARTMENT_NO	NUMBER	Yes	null	3	null

Transactions

COLUMN_NAME	**DATA_TYPE**	**NULLABLE**	**DATA_DEFAULT**	**COLUMN_ID**	**COMMENTS**
TRANSACTION_NO	NUMBER	No	null	1	null
JOB_NO	NUMBER	Yes	null	2	null
COSTS	NUMBER	Yes	null	3	null

Task 3 Database Storage Structures

3.1 Discussion of Storage Structures for Tables

Table Name	Type of Query	Search Key	Frequency	Storage Structure
Customer	#1 Insert #14 Random	-- cname	30/day 50/day	Dynamic hashing (cname key)
Department	#2 Insert #12 Random #13 Random	-- department_no department_no	Infrequent 100/day 20/day	Dynamic hashing (department_no key)
Assembly	#3 Insert #14 Random	-- assembly_id	50/day 50/day	Dynamic hashing (assembly_id key)
Process	#4 Insert #12 Random #13 Random #11 Random	-- process_id process_id process_id	Infrequent 100/day 20/day 20/day	Dynamic hashing (process_id key)
FitProcess	#4 Insert	--	Infrequent	Heap
PaintProcess	#4 Insert	--	Infrequent	Heap
CutProcess	#4 Insert	--	Infrequent	Heap
Account	#5 Insert #8 Random	-- account_no	10/day 50/day	Dynamic hashing (account_no key)
AssemblyAccount	#5 Insert #8 Random #9 Random	-- account_no assembly_id	10/day 50/day 200/day	Dynamic hashing (assembly_id)
ProcessAccount	#5 Insert #8 Random	-- account_no	10/day 50/day	Dynamic hashing (account_no)
DepartmentAccount	#5 Insert #8 Random	-- account_no	10/day 50/day	Dynamic hashing (account_no)
Job	#6 Insert #7 Random #10 Random #13 Random #14 Random #11 Random	-- job_no assembly_id completed_date job_no completed_date	50/day 50/day 100/day 20/day 50/day 20/day	Dynamic hashing (assembly_id)
CutJob	#15 Range #7 Insert #13 Random	Job_no -- job_no	1/month 50/day 20/day	Index sequential on job_no
PaintJob	#16 Random #7 Insert #13 Random #14 Random	Job_no -- job_no color	1/week 50/day 20/day 50/day	Dynamic Hashing (color)
Transaction	#8 Insert	--	50/day	Heap

Figure 2: Table showing entities with corresponding queries

As can be seen in Figure 2, when looking over the storage structures that we discussed in class, a **dynamic hashing** scheme should be used for those tables requiring random search using the most frequented key as the hash key. Tables that require a range search, such as CutJob, should be stored as an **index sequential file** on the search key. Tables that only ever have the insert query called should be stored as a **heap**.

3.2 Discussion of Storage Structures For Tables (Oracle 12c)

As shown, the bulk of our tables should be stored with a dynamic hash due to the high number of random access calls. Unfortunately, although Oracle 12c does offer hash clusters, as a user I do not have the administrative privileges to create such a cluster on the OU Oracle databases. The heap-organized table is the default table in Oracle.

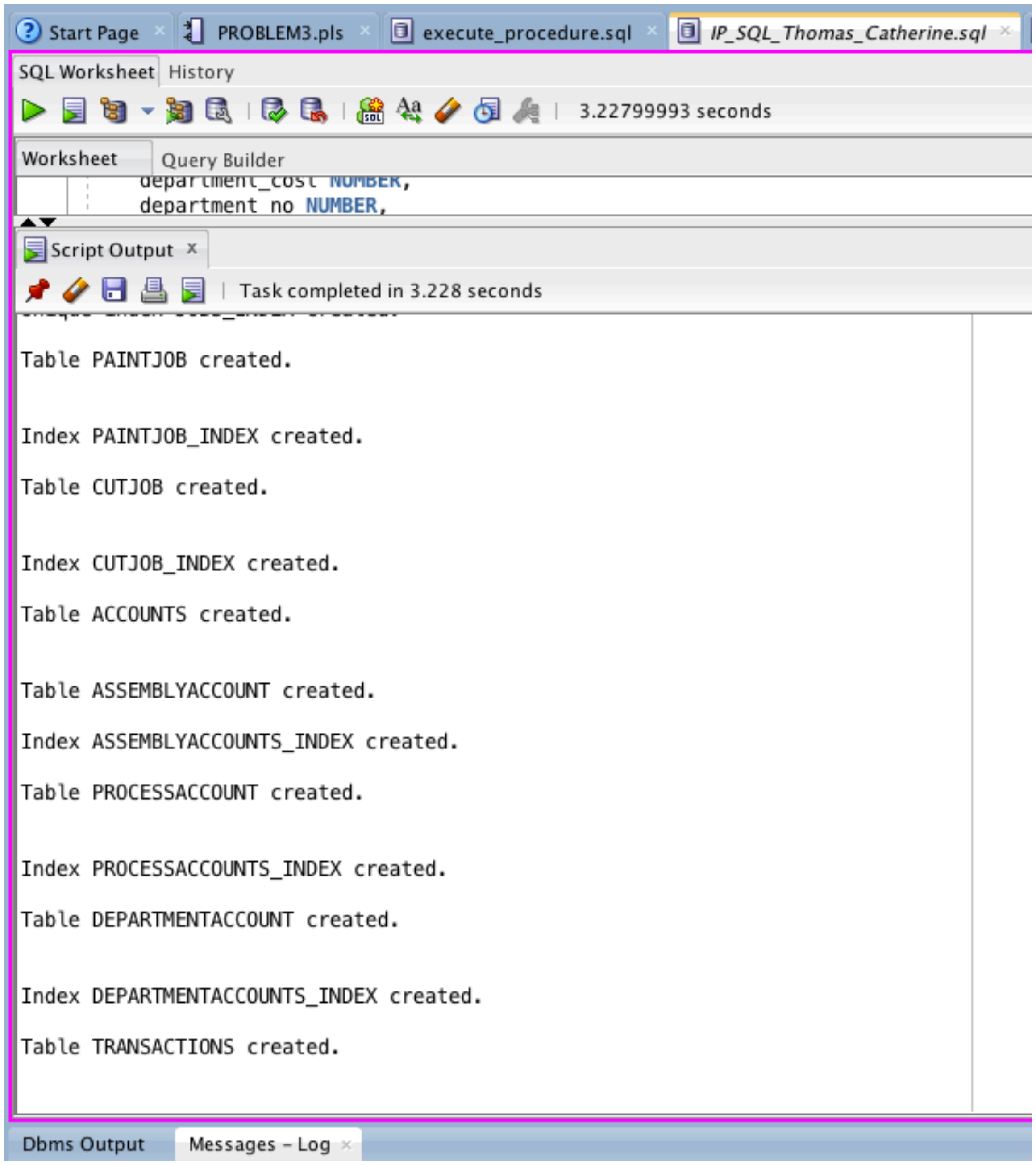
As an alternative to hashing, I can instead create indexes. The default index-organized table in Oracle is based on the B-tree. This would provide both fast random access on the search key as well as fast range access. Thus, almost all tables listed in Figure 2 except for those specified to be stored as a heap will instead be stored as an index-organized file with the specified key. In the case that the primary key is also the key to be indexed, this will NOT be explicitly stated in my SQL statements as Oracle by default creates an index on the primary key.

Task 4 SQL Program

4.1 SQL Statements

```
DROP TABLE Transactions;
DROP TABLE PaintJob;
DROP TABLE CutJob;
DROP TABLE Jobs;
DROP TABLE AssemblyAccount;
DROP TABLE DepartmentAccount;
DROP TABLE ProcessAccount;
DROP TABLE Accounts;
DROP TABLE FitProcess;
DROP TABLE PaintProcess;
DROP TABLE CutProcess;
DROP TABLE Processes;
DROP TABLE Department;
DROP TABLE Assemblies;
DROP TABLE Customer;

CREATE TABLE Customer (
  cname VARCHAR2(40) NOT NULL,
  address VARCHAR2(40),
  PRIMARY KEY (cname));

CREATE TABLE Department (
  department_no NUMBER NOT NULL,
  department_data VARCHAR2(150),
  PRIMARY KEY (department_no));

CREATE TABLE Assemblies (
  assembly_id NUMBER NOT NULL,
  assembly_details VARCHAR2(150),
  date_ordered VARCHAR2(15),
  cname VARCHAR2(40),
  PRIMARY KEY (assembly_id),
  FOREIGN KEY (cname) REFERENCES Customer(cname));

CREATE TABLE Processes (
  process_id NUMBER NOT NULL,
  process_data VARCHAR2(150),
  department_no NUMBER,
  PRIMARY KEY (process_id),
  FOREIGN KEY (department_no) REFERENCES Department(department_no));

CREATE TABLE FitProcess (
  process_id NUMBER NOT NULL,
  fit_type VARCHAR2(30),
  FOREIGN KEY (process_id) REFERENCES Processes(process_id)
  ON DELETE CASCADE);

CREATE TABLE PaintProcess (
  process_id NUMBER NOT NULL,
  paint_type VARCHAR2(30),
  painting_method VARCHAR2(30),
  FOREIGN KEY (process_id) REFERENCES Processes(process_id)
  ON DELETE CASCADE);

CREATE TABLE CutProcess (
  process_id NUMBER NOT NULL,
  cutting_type VARCHAR2(30),
  machine_type VARCHAR2(30),
  FOREIGN KEY (process_id) REFERENCES Processes(process_id)
  ON DELETE CASCADE);

CREATE TABLE Jobs (
  job_no NUMBER NOT NULL,
  start_date VARCHAR2(15),
  completed_date VARCHAR2(15),
  process_id NUMBER,
  assembly_id NUMBER,
  labor_time NUMBER,
  PRIMARY KEY (job_no),
  FOREIGN KEY (process_id) REFERENCES Processes(process_id),
  FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id));

  CREATE UNIQUE INDEX jobs_index ON Jobs(assembly_id);

  CREATE TABLE PaintJob (
    job_no NOT NULL,
    color VARCHAR2(15),
    volume_paint NUMBER,
    FOREIGN KEY (job_no) REFERENCES Jobs(job_no)
    ON DELETE CASCADE);

  CREATE INDEX paintjob_index ON PaintJob(color);

  CREATE TABLE CutJob (
    job_no NOT NULL,
    machine_type VARCHAR2(30),
    machine_time NUMBER,
    material VARCHAR2(30),
    FOREIGN KEY (job_no) REFERENCES Jobs(job_no)
    ON DELETE CASCADE);

  CREATE INDEX cutjob_index ON CutJob(job_no);

  CREATE TABLE Accounts (
    account_no NUMBER NOT NULL,
    date_established VARCHAR2(15),
    PRIMARY KEY (account_no));

  CREATE TABLE AssemblyAccount (
    account_no NUMBER NOT NULL,
    assembly_cost NUMBER,
    assembly_id NUMBER,
    FOREIGN KEY (account_no) REFERENCES Accounts(account_no) ON DELETE
CASCADE,
    FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id));

    CREATE INDEX assemblyaccounts_index ON AssemblyAccount(assembly_id);

  CREATE TABLE ProcessAccount (
    account_no NUMBER NOT NULL,
    process_cost NUMBER,
    process_id NUMBER,
    FOREIGN KEY (account_no) REFERENCES Accounts(account_no) ON DELETE
CASCADE,
    FOREIGN KEY (process_id) REFERENCES Processes(process_id));

    CREATE INDEX processaccounts_index ON ProcessAccount(account_no);

  CREATE TABLE DepartmentAccount (
    account_no NUMBER NOT NULL,
    department_cost NUMBER,
    department_no NUMBER,
    FOREIGN KEY (account_no) REFERENCES Accounts(account_no) ON DELETE
CASCADE,
    FOREIGN KEY (department_no) REFERENCES Department(department_no));

    CREATE INDEX departmentaccounts_index ON DepartmentAccount(account_no);

  CREATE TABLE Transactions (
    transaction_no NUMBER NOT NULL,
    job_no NUMBER,
    costs NUMBER,
    PRIMARY KEY (transaction_no),
    FOREIGN KEY (job_no) REFERENCES Jobs(job_no));
```

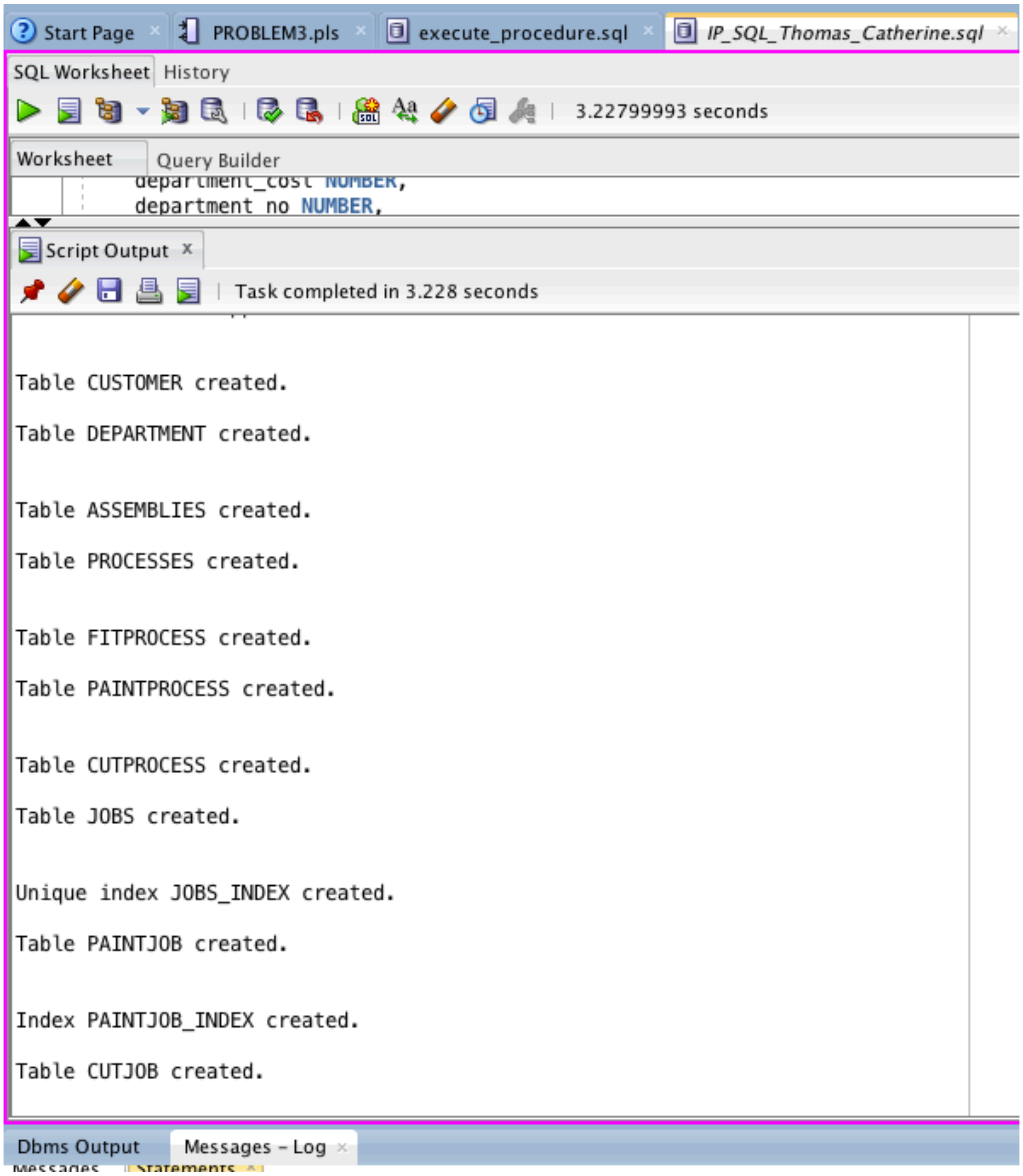
4.2 ScreenShot of SQL Table Creation



```
        department_cost NUMBER,
        department_no NUMBER,
```

```
Table PAINTJOB created.


Index PAINTJOB_INDEX created.

Table CUTJOB created.


Index CUTJOB_INDEX created.

Table ACCOUNTS created.


Table ASSEMBLYACCOUNT created.

Index ASSEMBLYACCOUNTS_INDEX created.

Table PROCESSACCOUNT created.


Index PROCESSACCOUNTS_INDEX created.

Table DEPARTMENTACCOUNT created.


Index DEPARTMENTACCOUNTS_INDEX created.

Table TRANSACTIONS created.
```

4.2.a Screenshot of table creation Part 1



```
department_cost NUMBER,
department_no NUMBER,
```

```
Table CUSTOMER created.

Table DEPARTMENT created.


Table ASSEMBLIES created.

Table PROCESSES created.


Table FITPROCESS created.

Table PAINTPROCESS created.


Table CUTPROCESS created.

Table JOBS created.


Unique index JOBS_INDEX created.

Table PAINTJOB created.


Index PAINTJOB_INDEX created.

Table CUTJOB created.
```

4.2.b Screenshot of table creation Part 2

Task 5 Java Source Program

5.1 Java Source Code

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.io.IOException;

/**
 * Individual Project for DBMS.
 * @author catherinethomas
 * @studentID 113848296
 * @date November 22, 2015
 */
public class IP_JAVA_Thomas_Catherine {
	String userName = "thom8296";
	String password = "FMzd8Ft2";
	String hostName = "oracle.cs.ou.edu";
	String serviceName = "pdborcl.cs.ou.edu";
	String portNumber = "1521";

	/**
	 * Get Connection
	 * Establishes a connection with an Oracle database.
	 * Uses class variables userName, password, hostname,
	 * serviceName, and portNumber to do so.
	 * @return Connection to an oracle database.
	 */
	public Connection getConnection(){
		try{
			Class.forName("oracle.jdbc.driver.OracleDriver");
		    Connection conn = null;
		    conn = DriverManager.getConnection("jdbc:oracle:thin:@" + 
hostName
		              + ":1521/" + serviceName,
		              userName, password);
		    return conn;
		} catch(Exception e){
			System.out.println(e);
			return null;
		}
	}

	/**
```

```
         * Option 1
         * Enters a new customer into the database.
         * @cname Name of the customer.
         * @address Address of the customer.
         * @conn Connection with Oracle database
         */
        public void option1(String cname, String address, Connection conn){
                //Write SQL
                try{
                        String sql = "INSERT INTO Customer (cname, address) VALUES
(\'"+cname+"\',\'"+address+"\')";
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);
                        System.out.println("Insert customer was a success!\n");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 1. \n"
                                        + e);
                }
        }

        /**
         * Option 2
         * Enters a new department into the database.
         * @param departmentNumber Number of new department. Must be unique.
         * @param departmentData String representation of the department's data.
         * @param conn Connection with Oracle database.
         */
        public void option2(int departmentNumber, String departmentData,
Connection conn){
                try{
                        String sql = "INSERT INTO Department (department_no,
department_data) "
                                        + "VALUES
("+departmentNumber+",'"+departmentData+"')";
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);
                        System.out.println("Insert department was a success!\n");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 2. \n" +
e);
                }
        } //end option2

        /**
         * Option 3
         * Creates a new assembly.
         * @param cname Name of customer associated with assembly.
         * @param assemblyDetails Details of assembly
         * @param assemblyid ID number of assembly
         * @param dateOrdered Date the assembly was ordered
         * @param conn Connection with oracle database
```

```
	 */
	public void option3(String cname, String assemblyDetails, int
assemblyid, String dateOrdered, Connection conn){
		try{
			String sql = "INSERT INTO Assemblies (assembly_id,
assembly_details, cname, date_ordered) "
					+ "VALUES (" + assemblyid +
",'"+assemblyDetails+"','"+cname+"','"+dateOrdered+"')";
			Statement stmt = conn.createStatement();
			stmt.executeQuery(sql);
			System.out.println("Insert assembly was a success!\n");
			stmt.close();
		} catch(Exception e){
			System.out.println("An error occurred on option 3. \n" +
e);
		}
	} //end option3

	/**
	 * Option 4
	 * Creates a new process.
	 * @param processId ID number of new process
	 * @param processData Data of new process
	 * @param departmentNo Number of department that supervises this process
	 * @param type Can be cut, paint, or fit.
	 * @param methodType Can be cut type, paint type, or fit type
	 * @param optional For a paint process, this can be painting method, or
for
	 * a cut process this can represent machine type.
	 * @param conn Connection with oracle database
	 */
	public void option4(int processId,String processData,int departmentNo,
String type, String methodType,
			String optional, Connection conn){
		try{
			//Insert into Processes table
			String sql = "INSERT INTO Processes (process_id,
process_data,department_no) "
					+ "VALUES
("+processId+",'"+processData+"',"+departmentNo+")";
			Statement stmt = conn.createStatement();
			stmt.executeQuery(sql);

			//Insert into specific type table
			sql = "";
			if(type.equalsIgnoreCase("fit")){
				//Was a fit process, construct appropriate SQL
				sql = "INSERT INTO FitProcess (process_id, fit_type)
VALUES ("+processId+",'"+methodType+"')";
			} else if (type.equalsIgnoreCase("paint")){
				//Paint process, construct appropriate SQL
				sql = "INSERT INTO PaintProcess (process_id,
paint_type, painting_method) "
```

```
                                        + "VALUES
("+processId+",'"+methodType+"','"+optional+"')";
                    } else if (type.equalsIgnoreCase("cut")){
                          //Cut process, construct appropriate SQL
                          sql = "INSERT INTO CutProcess (process_id,
cutting_type, machine_type) "
                                        + "VALUES
("+processId+",'"+methodType+"','"+optional+"')";
                    }

                    //Execute SQL
                    stmt.executeQuery(sql);
                    System.out.println("Insert process was a success!\n");
                    stmt.close();
             } catch (Exception e){
                    System.out.println("An error occurred on option 4. \n" +
e);
             }
       } //end option4

       /**
        * Option 5
        * Enters a new account and associates it with a department, process, or
assembly.
        * @param accountNumber Unique ID nomber for the account.
        * @param type Can be department, process, or assembly.
        * @param foreignId ID for the department, process, or assembly that the
account
        * is associated with.
        * @param dateStarted Date the account was set up.
        * @param conn Connection to an Oracle database
        */
       public void option5(int accountNumber,String type, int foreignId,
String dateStarted, Connection conn){
             try{
                    //Insert into Accounts table
                    String sql = "INSERT INTO Accounts (account_no,
date_established) "
                                 + "VALUES
("+accountNumber+",'"+dateStarted+"')";
                    Statement stmt = conn.createStatement();
                    stmt.executeQuery(sql);

                    //Insert into specific type table
                    sql = "";
                    if(type.equalsIgnoreCase("department")){
                          //Was a dept account, construct appropriate SQL
                          sql = "INSERT INTO DepartmentAccount (account_no,
department_cost, department_no)"
                                        + " VALUES
("+accountNumber+",0,"+foreignId+")";
                    } else if (type.equalsIgnoreCase("assembly")){
                          //Assembly account, construct appropriate SQL
```

```
                                sql = "INSERT INTO AssemblyAccount (account_no, 
assembly_cost, assembly_id)"
                                                + " VALUES 
("+accountNumber+",0,"+foreignId+")";
                        } else if (type.equalsIgnoreCase("process")){
                                //Process account, construct appropriate SQL
                                sql = "INSERT INTO ProcessAccount (account_no, 
process_cost, process_id)"
                                                + " VALUES 
("+accountNumber+",0,"+foreignId+")";
                        }

                        //Execute sql
                        stmt.executeQuery(sql);
                        System.out.println("Insert account was a success!\n");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 5. \n" + 
e);
                }
        } //end option5

        /**
         * Option 6
         * Enters a new job into the system.
         * @param jobNo Unique identifier of job
         * @param assemblyId ID of associated assembly
         * @param processId ID of associated process
         * @param startDate Start date of the job
         * @param conn Connection to an Oracle database
         */
        public void option6(int jobNo, int assemblyId, int processId, String 
startDate, Connection conn){
                try{
                        String sql = "INSERT INTO Jobs (job_no, assembly_id, 
process_id, start_date)"
                                        + "VALUES 
("+jobNo+","+assemblyId+","+processId+",'"+startDate+"')";
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);
                        System.out.println("Create job was a success!");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 6. \n" + 
e);
                }
        } //end option6

        /**
         * Option 7
         * Updates a job upon its completion. Adds information
         * relevant to the type of job
         * @param jobNo ID of job to be updated.
         * @param completedDate Date the job was completed.
         * @param laborTime Total labor time done on the job.
         * @param jobType Can be fit, paint, or cut.
         * @param volumeOrTime Represents paint volume or machine time.
         * @param colorOrMType Represents paint color or machine type.
         * @param material Material used for cut jobs.
         * @param conn Connection to Oracle database.
         */
        public void option7(int jobNo, String completedDate, int laborTime,
String jobType,
                        int volumeOrTime, String colorOrMType, String material,
Connection conn){
                try{
                        String sql = "UPDATE Jobs SET
completed_date='"+completedDate+"', labor_time ="+laborTime
                                        +" WHERE job_no="+jobNo;
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);

                        sql = "";
                        if(jobType.equalsIgnoreCase("cut")){
                                sql = "INSERT INTO CutJob (job_no, machine_type,
machine_time, material)"
                                        + "VALUES
("+jobNo+",'"+colorOrMType+"',"+volumeOrTime+",'"+material+"')";
                        } else if(jobType.equalsIgnoreCase("paint")){
                                sql = "INSERT INTO PaintJob (job_no, color,
volume_paint)"
                                        + "VALUES
("+jobNo+",'"+colorOrMType+"',"+volumeOrTime+")";
                        }

                        stmt.executeQuery(sql);
                        System.out.println("Update job was a success!");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 7. \n" +
e);
                }
        } //end option7

        /**
         * Option 8
         * Creates a new transaction associated with a job
         * and updates all accounts associated with this transaction.
         * @param transactionNo Unique identifier for transaction
         * @param jobNo Job this transaction will be associated with
         * @param cost Cost of transaction
         * @param conn Connection to Oracle database
         */
        public void option8(int transactionNo, int jobNo, int cost, Connection
conn){
                try{
                    //Create transaction
                    String sql = "INSERT INTO Transactions (transaction_no, 
job_no, costs)"
                                   + " VALUES 
("+transactionNo+","+jobNo+","+cost+")";
                    Statement stmt = conn.createStatement();
                    stmt.executeQuery(sql);

                    //Update process account
                    //The updated process account is for the process used by a 
job.
                    sql = "UPDATE ProcessAccount SET process_cost = 
process_cost +"+cost+" "
                                   + "WHERE EXISTS (SELECT * FROM Jobs WHERE 
Jobs.process_id = ProcessAccount.process_id "
                                   + "AND Jobs.job_no="+jobNo+")";
                    stmt.executeQuery(sql);

                    //Update department account
                    //The updated department account is for the department that 
manages that process.
                    sql = "UPDATE DepartmentAccount SET department_cost = 
department_cost +"+cost+" "
                                   + "WHERE EXISTS (SELECT * FROM Jobs,Processes 
WHERE Jobs.process_id = Processes.process_id "
                                   + "AND Jobs.job_no="+jobNo+" AND 
DepartmentAccount.department_no = Processes.department_no)";
                    stmt.executeQuery(sql);

                    //Update assembly account
                    //The updated assembly account is for the assembly that 
requires the job.
                    sql = "UPDATE AssemblyAccount SET assembly_cost = 
assembly_cost +"+cost+" "
                                   + "WHERE EXISTS (SELECT * FROM Jobs WHERE 
Jobs.assembly_id = AssemblyAccount.assembly_id "
                                   + "AND Jobs.job_no="+jobNo+")";
                    stmt.executeQuery(sql);

             } catch (Exception e){
                    System.out.println("An error occurred in option 8.\n" + e);
             }
      } //end option8

      /**
       * Option 9
       * Retrieves the cost for a given assembly
       * @param assemblyId ID of assembly whose cost will be retrieved
       * @param conn Connection to Oracle database
       */
      public void option9(int assemblyId, Connection conn){
             try{
                    String sql = "SELECT assembly_cost FROM AssemblyAccount 
```

```
WHERE assembly_id="+assemblyId;
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    if(rs.next()){
                          System.out.println("The cost for Assembly
#"+assemblyId+" is: "+rs.getString("assembly_cost")); //Check
                    } else {
                          System.out.println("No results found.");
                    }
                    stmt.close();
              } catch (Exception e){
                    System.out.println("An error occurred on option 9. \n" +
e);
              }
      } //end option9

      /**
       * Option 10
       * Retrieves the labor time for a given assembly
       * @param assemblyId ID of assembly whose labor time will be retrieved
       * @param conn Connection to Oracle database
       */
      public void option10(int assemblyId, Connection conn){
              try{
                    String sql = "SELECT SUM(labor_time) sum FROM Jobs WHERE
assembly_id="+assemblyId;
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    if(rs.next()){
                          System.out.println("The labor time for Assembly
#"+assemblyId+" is: "+rs.getString("sum"));
                    }
                    stmt.close();
              } catch (Exception e){
                    System.out.println("An error occurred on option 10. \n" +
e);
              }
      }// end option10

      /**
       * Option 11
       * Retrieves total labor time in a department for jobs completed on a
certain
       * date.
       * @param departmentNo Number of department whose total labor will be
calculated.
       * @param date Date to find total labor time.
       * @param conn Connection to Oracle database.
       */
      public void option11(int departmentNo, String date, Connection conn){
              try{
                    String sql = "SELECT SUM(labor_time) sum FROM
Jobs,Processes"
```

```
                                + " WHERE
Processes.department_no="+departmentNo+" AND "
                                             +
"Jobs.process_id=Processes.process_id AND "
                                             +
"Jobs.completed_date='"+date+"'";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    if(rs.next()){
                          System.out.println("The labor time for Department
#"+departmentNo+" "
                                      + "on " +date+ " is:
"+rs.getString("sum"));
                    }
                    stmt.close();
             } catch (Exception e){
                    System.out.println("An error occurred on option 11. \n" +
e);
             }
      } //end option11

      /**
       * Option 12
       * Retrieves processes associated with a given assembly, orders them
       * by date, and also gives the department associated with each process.
       * @param assemblyId ID of assembly whose processes will be found.
       * @param conn Connection to Oracle database.
       */
      public void option12(int assemblyId, Connection conn){
             try{
                    String sql = "SELECT Processes.process_id, Jobs.start_date,
Processes.department_no FROM Jobs, Processes "
                                + "WHERE Jobs.process_id =
Processes.process_id AND "
                                + "Jobs.assembly_id="+assemblyId+" ORDER BY
start_date";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    while(rs.next()){
                          System.out.println("Process id: " +
rs.getString("process_id") + " Start date: " + rs.getString("start_date")
                                      + " Dept. No.: " +
rs.getString("department_no") );
                    }
                    stmt.close();
             } catch (Exception e){
                    System.out.println("An error occurred on option 12. \n" +
e);
             }
      } //end option12

      /**
       * Retrieves all jobs, together with type information
```

```
         * and assembly id, completed on a specified day
         * in a certain department.
         * @param departmentNo Number of department with associated jobs.
         * @param date Day the jobs were completed.
         * @param conn Connection to Oracle database.
         */
        public void option13(int departmentNo, String date, Connection conn){
                try{

                        //Get Cut Jobs
                        String sql = "SELECT Jobs.job_no, Jobs.assembly_id,
CutJob.machine_type,"
                                        + " CutJob.machine_time, CutJob.material"
                                        + "  FROM Jobs,Processes,CutJob"
                                        + " WHERE
Processes.department_no="+departmentNo+" AND "
                                        + "Jobs.process_id=Processes.process_id AND "
                                        + "Jobs.completed_date='"+date+"' AND "
                                        + "CutJob.job_no = Jobs.job_no";
                        Statement stmt = conn.createStatement();
                        ResultSet rs = stmt.executeQuery(sql);
                        if(rs.next()){
                                System.out.println("Job No.:
"+rs.getString("job_no")+" "
                                                + " Date: " +date+ " Assembly ID:
"+rs.getString("assembly_id")
                                                + " Machine Type: " +
rs.getString("machine_type")
                                                + " Machine Time: " +
rs.getString("machine_time")
                                                + " Material: " +
rs.getString("material"));
                        }

                        //Get Paint Jobs
                        sql = "SELECT Jobs.job_no, Jobs.assembly_id,
PaintJob.volume_paint,"
                                        + " PaintJob.color"
                                        + "  FROM Jobs,Processes,PaintJob"
                                        + " WHERE
Processes.department_no="+departmentNo+" AND "
                                        + "Jobs.process_id=Processes.process_id AND "
                                        + "Jobs.completed_date='"+date+"' AND "
                                        + "PaintJob.job_no = Jobs.job_no";
                        rs = stmt.executeQuery(sql);
                        if(rs.next()){
                                System.out.println("Job No.:
"+rs.getString("job_no")+" "
                                                + " Date: " +date+ " Assembly ID:
"+rs.getString("assembly_id")
                                                + " Volume Paint: " +
rs.getString("volume_paint")
                                                + " Color: " + rs.getString("color"));
```

```
                    }

                    //Get Fit Jobs
                    sql = "SELECT Jobs.job_no, Jobs.assembly_id"
                                    + "  FROM Jobs,Processes"
                                    + " WHERE
Processes.department_no="+departmentNo+" AND "
                                    + "Jobs.process_id=Processes.process_id AND "
                                    + "Jobs.completed_date='"+date+"' AND NOT
EXISTS "
                                    + "(SELECT * FROM CutJob,PaintJob WHERE
CutJob.job_no=Jobs.job_no "
                                    + "OR PaintJob.job_no = Jobs.job_no)";
                    rs = stmt.executeQuery(sql);
                    if(rs.next()){
                          System.out.println("Job No.:
"+rs.getString("job_no")+" "
                                          + " Date: " +date+ " Assembly ID:
"+rs.getString("assembly_id"));
                    }

                    stmt.close();
              } catch (Exception e){
                    System.out.println("An error occurred on option 13. \n" +
e);
              }
        } //end option 13

        /**
         * Retrieves customers' whose assemblies are painted
         * RED with the given paint method.
         * @param paintMethod String stating method (ex: spray, prime, base)
         * @param conn Connection to Oracle database
         */
        public void option14(String paintMethod, Connection conn){
              try{
                    String sql = "SELECT C.cname, C.address "
                                    + "FROM Customer C, Assemblies A, Jobs J,
PaintJob PJ, PaintProcess PP"
                                    + " WHERE C.cname = A.cname"
                                    + " AND A.assembly_id = J.assembly_id"
                                    + " AND J.job_no = PJ.job_no"
                                    + " AND PJ.color = 'RED'"
                                    + " AND J.process_id = PP.process_id"
                                    + " AND PP.painting_method='"+paintMethod+"'";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    while(rs.next()){
                          System.out.println("Customer name: " +
rs.getString("cname") + " Address: " + rs.getString("address"));
                    }
                    stmt.close();
              } catch (Exception e){
```

```
                        System.out.println("An error occurred on option 14.\n" +
e);
                }
        } //end option14

        /**
         * Deletes all cutjobs who dates are within a given range
         * @param jobNo1 Start number
         * @param jobNo2 End number
         * @param conn Connection to Oracle database
         */
        public void option15(int jobNo1, int jobNo2, Connection conn){
                if(jobNo1 > jobNo2){
                        System.out.println("Invalid input.\n");
                        return;
                }

                try{
                        String sql = "DELETE FROM CutJob"
                                        + " WHERE job_no BETWEEN " + jobNo1 + " AND "
+ jobNo2;
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);
                        System.out.println("Delete cut jobs was a success!");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 15.\n" +
e);
                }
        } //end option15

        /**
         * Changes the color of a given paint job.
         * @param job_no ID of paint job.
         * @param color New color.
         * @param conn Connection to Oracle database.
         */
        public void option16(int job_no, String color, Connection conn){
                try{
                        String sql = "UPDATE PaintJob SET color='"+color+"' WHERE
job_no="+job_no;
                        Statement stmt = conn.createStatement();
                        stmt.executeQuery(sql);
                        System.out.println("Update color successful!");
                        stmt.close();
                } catch (Exception e){
                        System.out.println("An error occurred on option 16.\n" +
e);
                }
        } //end option16

        /**
         * Prints the average cost of
```

```
	 * assemblies, processes, and departments.
	 * @param conn Connection to Oracle database.
	 */
	public void option17(Connection conn){
		try{
			//Find assembly average
			String sql = "SELECT AVG(assembly_cost) assemblyCost FROM 
AssemblyAccount";
			Statement stmt = conn.createStatement();
			ResultSet rs = stmt.executeQuery(sql);
			if(rs.next()){
				System.out.println("Avg. Assembly Cost: " + 
rs.getString("assemblyCost"));
			}
			
			sql = "SELECT AVG(process_cost) processCost FROM 
ProcessAccount";
			rs = stmt.executeQuery(sql);
			if(rs.next()){
				System.out.println("Avg. Process Cost: " + 
rs.getString("processCost"));
			}
			
			sql = "SELECT AVG(department_cost) deptCost FROM 
DepartmentAccount";
			rs = stmt.executeQuery(sql);
			if(rs.next()){
				System.out.println("Avg. Department Cost: " + 
rs.getString("deptCost"));
			}
			
			stmt.close();
		} catch(Exception e){
			System.out.println("An error occurred on option 17.\n" + 
e);
		}
	}
	
	/**
	 * Reads in a files of customer names and addresses
	 * and inserts them into the Customer table.
	 * @param filename Name of file with customer data
	 * @param conn Connection to Oracle database
	 * @throws FileNotFoundException Input file was not found
	 */
	public void option18(String filename, Connection conn) throws 
FileNotFoundException{
		Scanner in = new Scanner(new FileReader(filename));
		
		try{
			Statement stmt = conn.createStatement();
			String sql;
```

```
				//Read in file line by line
				while(in.hasNextLine()){
					String[] input = in.nextLine().split(",", 2);
					sql = "INSERT INTO Customer (cname, address) VALUES
(\'"+input[0]+"\',\'"+input[1]+"\')";
					stmt.executeQuery(sql);
				}
				stmt.close();
				in.close();
				System.out.println("Import customers was a success!");
			} catch (Exception e){
				System.out.println("An error occurred on option 18.\n" +
e);
			}
		} //end option18

		/**
		 * Prints names of customers whose assemblies are RED
		 * done with a given paint method to a file.
		 * @param filename Name of output file
		 * @param method Paint method to be search for
		 * @param conn Connection to Oracle database
		 */
		public void option19(String filename, String method, Connection conn){
			try{
				//Execute query
				String sql = "SELECT C.cname, C.address "
						+ "FROM Customer C, Assemblies A, Jobs J,
PaintJob PJ, PaintProcess PP"
						+ " WHERE C.cname = A.cname"
						+ " AND A.assembly_id = J.assembly_id"
						+ " AND J.job_no = PJ.job_no"
						+ " AND PJ.color = 'RED'"
						+ " AND J.process_id = PP.process_id"
						+ " AND PP.painting_method='"+method+"'";
				Statement stmt = conn.createStatement();
				ResultSet rs = stmt.executeQuery(sql);


				//Create file to write to
				File file = new File(filename);

				// if file doesnt exists, then create it
				if (!file.exists()) {
					file.createNewFile();
				}
				FileWriter fw = new FileWriter(file.getAbsoluteFile());
				BufferedWriter bw = new BufferedWriter(fw);

				//Write to file
				while(rs.next()){
					bw.write("Customer name: " + rs.getString("cname") +
" Address: " + rs.getString("address"));
```

```
                    }
                    stmt.close();
                    bw.close();
                    System.out.println("Write to file was a success!");
            } catch (Exception e){
                    System.out.println("An error occurred on option 14.\n" +
e);
            }
    } //end option19

    /**
     * Prints all query options.
     */
    public void printOptions(){
            System.out.println(
                            "(1) Enter a new customer\n" +
                            "(2) Enter a new department\n" +
                            "(3) Enter a new assembly with its customer-name,
assembly-details, assembly-id, and date ordered\n" +
                            "(4) Enter a new process-id and its department
together with its type and information relevant tothe type\n" +
                            "(5) Create a new account and associate it with the
process, assembly, or department to "
                                    + "which it is applicable\n" +
                            "(6) Enter a new job, given its job-no, assembly-id,
process-id, and date the job commenced\n" +
                            "(7) At the completion of a job, enter the date it
completed and the information relevant to the" +
                                    " type of job\n" +
                            "(8) Enter a transaction-no, and its sup-cost and
update all the costs (details) of the affected accounts\n" +
                            "(9) Retrieve the cost incurred on an assembly-id \n"
+
                            "(10) Retrieve the labor time recorded on an
assembly-id \n" +
                            "(11) Retrieve the total labor time within a
department for jobs completed in the "
                                    + "department during a given date\n" +
                            "(12) Retrieve the processes through which a given
assembly-id has passed so far (in "
                                    + "date commenced order) and the department
responsible for each process \n" +
                            "(13) Retrieve the jobs (together with their type
information and assembly-id) "
                                    + "completed during a given date in a given
department \n" +
                            "(14) Retrieve the customers (in name order) whose
assemblies are painted"
                                    + " RED using a given painting method \n" +
                            "(15) Delete all cut-jobs whose job-no is in some
range \n" +
                            "(16) Change the color of a given paint job \n" +
                            "(17) Retrieve the average cost of all accounts \n" +
                            "(18) Import: enter new customers from a data file
until the file is empty \n" +
                            "(19) Export: Retrieve the customers (in name order)
whose assemblies are painted RED"
                                    + " using a given painting method and output
them to a data file \n" +
                            "(20) Quit\n" +
                            "Input option #: ");
    } //end printOptions

    /**
     * Prints the Customer table.
     * @param conn Connection to Oracle database.
     */
    public void printCustomers(Connection conn){
            try{
                    String sql = "SELECT * FROM Customer";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);

                    System.out.println("CUSTOMER TABLE\n"
                                    + "cname, address\n"
                                    + "-------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("cname") + ", " +
rs.getString("address"));
                    }
                    System.out.println("-------------------------------\n");
                    stmt.close();
            } catch(Exception e){
                    System.out.println("An error occured printing
customers.\n" + e);
            }
    } //end printCustomers

    /**
     * Prints the Department table.
     * @param conn Connection to Oracle database.
     */
    public void printDepartments(Connection conn){
            try{
                    String sql = "SELECT * FROM Department";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);

                    System.out.println("DEPARTMENT TABLE\n"
                                    + "department_no, department_data\n"
                                    + "-------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("department_no") +
", " + rs.getString("department_data"));
                    }
                    System.out.println("-------------------------------\n");
                    stmt.close();
            } catch(Exception e){
                    System.out.println("An error occured printing
departments.\n" + e);
                    return;
            }
    } //end printDepartment

    /**
     * Prints the Assemblies table.
     * @param conn Connection to Oracle database.
     */
    public void printAssemblies(Connection conn){
            try{
                    String sql = "SELECT * FROM Assemblies";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);

                    System.out.println("ASSEMBLIES TABLE\n"
                                    + "assembly_id, assembly_details,
date_ordered, cname\n"
                                    + "-------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("assembly_id") + ",
" + rs.getString("assembly_details")
                                                + ", " + rs.getString("date_ordered") +
", " + rs.getString("cname"));
                    }
                    System.out.println("-------------------------------\n");
                    stmt.close();
            } catch(Exception e){
                    System.out.println("An error occured printing
departments.\n" + e);
                    return;
            }
    } //end printDepartment

    /**
     * Prints all the Processes table.
     * @param conn Connection to Oracle database.
     */
    public void printProcesses(Connection conn){
            try{
                    //Get processes
                    String sql = "SELECT * FROM Processes";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);

                    System.out.println("PROCESSES TABLE\n"
                                    + "process_id, process_data, department_no\n"
                                    + "-------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("process_id") + ", "
+ rs.getString("process_data")
                                        + ", " +
rs.getString("department_no"));
                    }
                    System.out.println("-------------------------------\n");

                    //Get Fit Processes
                    sql = "SELECT * FROM FitProcess";
                    rs = stmt.executeQuery(sql);

                    System.out.println("FIT PROCESS TABLE\n"
                                 + "process_id, fit_type\n"
                                 + "-------------------------------");
                    while(rs.next()){
                           System.out.println(rs.getString("process_id") + ", "
+ rs.getString("fit_type"));
                    }
                    System.out.println("-------------------------------\n");

                    //Get Paint Processes
                    sql = "SELECT * FROM PaintProcess";
                    rs = stmt.executeQuery(sql);

                    System.out.println("PAINT PROCESS TABLE\n"
                                 + "process_id, paint_type, painting_method\n"
                                 + "-------------------------------");
                    while(rs.next()){
                           System.out.println(rs.getString("process_id") + ", "
+ rs.getString("paint_type")
                                        + ", " +
rs.getString("painting_method"));
                    }
                    System.out.println("-------------------------------\n");

                    //Get Paint Processes
                    sql = "SELECT * FROM CutProcess";
                    rs = stmt.executeQuery(sql);

                    System.out.println("CUT PROCESS TABLE\n"
                                 + "process_id, cutting_type, machine_type\n"
                                 + "-------------------------------");
                    while(rs.next()){
                           System.out.println(rs.getString("process_id") + ", "
+ rs.getString("cutting_type")
                                        + ", " + rs.getString("machine_type"));
                    }
                    System.out.println("-------------------------------\n");

                    stmt.close();
             } catch(Exception e){
                    System.out.println("An error occured printing processes.\n"
+ e);
                    return;
```

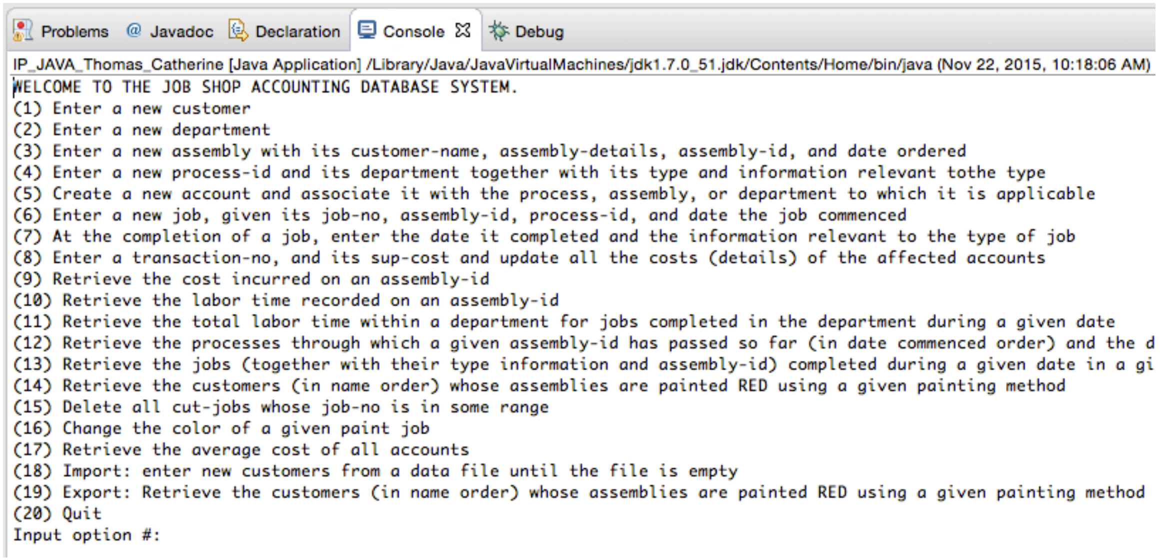
```
			}
		} //end printProcesses

		/**
		 * Prints all the Account tables.
		 * @param conn Connection to Oracle database.
		 */
		public void printAccounts(Connection conn){
			try{
				//Get accounts
				String sql = "SELECT * FROM Accounts";
				Statement stmt = conn.createStatement();
				ResultSet rs = stmt.executeQuery(sql);

				System.out.println("ACCOUNTS TABLE\n"
						+ "account_no, date_established\n"
						+ "-------------------------------");
				while(rs.next()){
					System.out.println(rs.getString("account_no") + ", "
+ rs.getString("date_established"));
				}
				System.out.println("-------------------------------\n");

				//Get Assembly accounts
				sql = "SELECT * FROM AssemblyAccount";
				rs = stmt.executeQuery(sql);

				System.out.println("ASSEMBLY ACCOUNT TABLE\n"
						+ "account_no, assembly_id, assembly_cost\n"
						+ "-------------------------------");
				while(rs.next()){
					System.out.println(rs.getString("account_no") + ", "
+ rs.getString("assembly_id")
							+ ", " +
rs.getString("assembly_cost"));
				}
				System.out.println("-------------------------------\n");

				//Get process accounts
				sql = "SELECT * FROM ProcessAccount";
				rs = stmt.executeQuery(sql);

				System.out.println("PROCESS ACCOUNT TABLE\n"
						+ "account_no, process_id, process_cost\n"
						+ "-------------------------------");
				while(rs.next()){
					System.out.println(rs.getString("account_no") + ", "
+ rs.getString("process_id")
							+ ", " + rs.getString("process_cost"));
				}
				System.out.println("-------------------------------\n");

				//Get department accounts
```

```
                    sql = "SELECT * FROM DepartmentAccount";
                    rs = stmt.executeQuery(sql);

                    System.out.println("DEPARTMENT ACCOUNT TABLE\n"
                                + "account_no, department_no,
department_cost\n"
                                + "--------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("account_no") + ", "
+ rs.getString("department_no")
                                    + ", " +
rs.getString("department_cost"));
                    }
                    System.out.println("--------------------------------\n");

                    stmt.close();
            } catch(Exception e){
                    System.out.println("An error occured printing accounts.\n"
+ e);
                    return;
            }
    } //end printAccounts

    /**
     * Prints all the Job tables.
     * @param conn Connection to Oracle database.
     */
    public void printJobs(Connection conn){
            try{
                    //Get jobs
                    String sql = "SELECT * FROM Jobs";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);

                    System.out.println("JOBS TABLE\n"
                                + "job_no, start_date, completed_date,
process_id, assembly_id, labor_time\n"
                                + "--------------------------------");
                    while(rs.next()){
                            System.out.println(rs.getString("job_no") + ", " +
rs.getString("start_date")
                                    + ", " +
rs.getString("completed_date")+ ", " + rs.getString("process_id")
                                    + ", " + rs.getString("assembly_id") +
", " + rs.getString("labor_time"));
                    }
                    System.out.println("--------------------------------\n");

                    //Get Paint Jobs
                    sql = "SELECT * FROM PaintJob";
                    rs = stmt.executeQuery(sql);

                    System.out.println("PAINT JOB TABLE\n"
                                + "job_no, color, volume\n"
                                + "------------------------------");
                    while(rs.next()){
                          System.out.println(rs.getString("job_no") + ", " +
rs.getString("color")
                                      + ", " + rs.getString("volume_paint"));
                    }
                    System.out.println("------------------------------\n");

                    //Get Cut Jobs
                    sql = "SELECT * FROM CutJob";
                    rs = stmt.executeQuery(sql);

                    System.out.println("CUT JOB TABLE\n"
                                + "job_no, machine_type, machine_time,
material\n"
                                + "------------------------------");
                    while(rs.next()){
                          System.out.println(rs.getString("job_no") + ", " +
rs.getString("machine_type")
                                      + ", " + rs.getString("machine_time")+
", " + rs.getString("material"));
                    }
                    System.out.println("------------------------------\n");

                    stmt.close();
              } catch(Exception e){
                    System.out.println("An error occured printing jobs.\n" +
e);
                    return;
              }
        } //end printJobs

        public void printTransaction(Connection conn){
              try{
                    String sql = "SELECT * FROM Transactions";
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery(sql);
                    System.out.println("TRANSACTIONS TABLE\n"
                                + "transaction_no, job_no, cost\n"
                                + "------------------------------");
                    while(rs.next()){
                          System.out.println(rs.getString("transaction_no") +
", " + rs.getString("job_no")
                                      + ", " + rs.getString("costs"));
                    }
                    System.out.println("------------------------------\n");
                    stmt.close();
              } catch (Exception e){
                    System.out.println("An error occured printing
transactions\n" + e);
                    return;
              }
```

```
	} //end printTransactions

	/**
	 * Main method. Runs a loop providing query options and receiving
	 * user input.
	 * @param args None.
	 * @throws IOException For invalid user input.
	 */
	public static void main(String[] args) throws IOException{
		IP_JAVA_Thomas_Catherine app = new IP_JAVA_Thomas_Catherine();
		Connection conn = app.getConnection(); //Establish database
connection

		//buffer to read in user input
		BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

		//Start app
		System.out.println("WELCOME TO THE JOB SHOP ACCOUNTING DATABASE
SYSTEM.");
		String option = "0";
		while(!option.equals("20")){
			app.printOptions();
			option = br.readLine(); // get option choice

			if (option.equals("1")){
				//Get user input
				System.out.println("Input customer name:");
				String cname = br.readLine();
				System.out.println("Input customer address:");
				String address = br.readLine();

				//Create customer
				app.option1(cname, address, conn);

				//Print customer table
				app.printCustomers(conn);
			} else if (option.equals("2")){
				//Get user input
				System.out.println("Input department number:");
				int number = Integer.parseInt(br.readLine());
				System.out.println("Input department data:");
				String data = br.readLine();

				//Create department
				app.option2(number, data, conn);

				//Print department table
				app.printDepartments(conn);
			} else if (option.equals("3")){
				//Get user input
				System.out.println("Input assembly id (number):");
				int id = Integer.parseInt(br.readLine());
```

```
                                System.out.println("Input assembly details:");
                                String details = br.readLine();
                                System.out.println("Input date ordered mm-dd-
yyyy:");
                                String date = br.readLine();
                                System.out.println("Input customer name:");
                                String cname = br.readLine();

                                //Create assembly
                                app.option3(cname, details, id, date, conn);

                                //Print table
                                app.printAssemblies(conn);
                        } else if(option.equals("4")){
                                //Get user input
                                System.out.println("Input process id (number):");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input process data:");
                                String details = br.readLine();
                                System.out.println("Input department number:");
                                int deptNo = Integer.parseInt(br.readLine());
                                System.out.println("Input the type of process (cut,
paint, fit):");
                                String type = br.readLine();
                                System.out.println("Depending on the type of
process, input the fit type, paint type, or cut type:");
                                String methodtype = br.readLine();
                                System.out.println("If cut process, input machine
type. If paint process, input painting method:");
                                String optional = br.readLine();

                                //Create process
                                app.option4(id, details, deptNo, type, methodtype,
optional, conn);

                                //Print processes
                                app.printProcesses(conn);
                        } else if(option.equals("5")){
                                //Get user input
                                System.out.println("Input account id (number):");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input date established mm-dd-
yyyy:");
                                String date = br.readLine();
                                System.out.println("Input account type (department,
assembly, process):");
                                String type = br.readLine();
                                System.out.println("Input ID/no. of corresponding
department, assembly, or process:");
                                int foreign = Integer.parseInt(br.readLine());

                                //Create account
                                app.option5(id, type, foreign, date,conn);
                                //Print accounts
                                app.printAccounts(conn);
                        } else if(option.equals("6")){
                                //Get user input
                                System.out.println("Input job number:");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input date started mm-dd-
yyyy:");
                                String date = br.readLine();
                                System.out.println("Input associated assembly id:");
                                int assemblyId = Integer.parseInt(br.readLine());
                                System.out.println("Input associated process id:");
                                int processId = Integer.parseInt(br.readLine());

                                //Create job
                                app.option6(id, assemblyId, processId, date, conn);

                                //Print job table
                                app.printJobs(conn);
                        } else if(option.equals("7")){
                                //Get user input
                                System.out.println("Input job number:");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input date completed mm-dd-
yyyy:");
                                String date = br.readLine();
                                System.out.println("Input labor time:");
                                int laborTime = Integer.parseInt(br.readLine());
                                System.out.println("Input job type (paint, cut,
fit):");
                                String type = br.readLine();
                                System.out.println("Input color (paint job) or
machine type (cut job) or 'none':");
                                String colorOrMType = br.readLine();
                                System.out.println("Input volume (paint job) or
machine time (cut job) or 0:");
                                int volOrTime = Integer.parseInt(br.readLine());
                                System.out.println("Input material (cut job) or
'none':");
                                String material = br.readLine();

                                //Update job
                                app.option7(id, date, laborTime, type, volOrTime,
colorOrMType, material, conn);

                                //Print job table
                                app.printJobs(conn);
                        } else if(option.equals("8")){
                                System.out.println("Input transaction number:");
                                int no = Integer.parseInt(br.readLine());
                                System.out.println("Input job number:");
                                int job = Integer.parseInt(br.readLine());
                                System.out.println("Input cost:");
                                int cost = Integer.parseInt(br.readLine());

                                //Create transaction
                                app.option8(no, job, cost, conn);

                                //Print transactions and accounts
                                app.printTransaction(conn);
                                app.printAccounts(conn);
                        } else if(option.equals("9")){
                                //Get user input
                                System.out.println("Input assembly id:");
                                int id = Integer.parseInt(br.readLine());

                                //Retrieve cost
                                app.option9(id, conn);
                        } else if(option.equals("10")){
                                //Get user input
                                System.out.println("Input assembly id:");
                                int id = Integer.parseInt(br.readLine());

                                //Retrieve labor time
                                app.option10(id, conn);
                        } else if(option.equals("11")){
                                //Get user input
                                System.out.println("Input department no:");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input completion date mm-dd-
        yyyy:");

                                String date = br.readLine();

                                //Retrieve labor time
                                app.option11(id, date, conn);
                        } else if(option.equals("12")){
                                //Get user input
                                System.out.println("Input assembly id:");
                                int id = Integer.parseInt(br.readLine());

                                //Retrieve data
                                app.option12(id, conn);
                        } else if(option.equals("13")){
                                //Get user input
                                System.out.println("Input department no:");
                                int id = Integer.parseInt(br.readLine());
                                System.out.println("Input completion date mm-dd-
        yyyy:");

                                String date = br.readLine();

                                //Retrieve data
                                app.option13(id, date, conn);
                        } else if(option.equals("14")){
                                //Get user input
                                System.out.println("Paint method (ex. spray, brush,
sponged):");
                                        String method = br.readLine();

                                        //Get customers
                                        app.option14(method, conn);
                              } else if(option.equals("15")){
                                        //Get user input
                                        System.out.println("Start job no.:");
                                        int start = Integer.parseInt(br.readLine());
                                        System.out.println("End job no.:");
                                        int end = Integer.parseInt(br.readLine());

                                        //Retrieve data
                                        app.option15(start, end, conn);

                                        //Print Jobs
                                        app.printJobs(conn);
                              } else if(option.equals("16")){
                                        //Get user input
                                        System.out.println("Enter paint job no.:");
                                        int number = Integer.parseInt(br.readLine());
                                        System.out.println("Enter new color:");
                                        String color = br.readLine();

                                        app.option16(number, color, conn);

                                        //Print Jobs
                                        app.printJobs(conn);
                              } else if(option.equals("17")){
                                        app.option17(conn);
                              } else if(option.equals("18")){
                                        //Get user input
                                        System.out.println("Enter name of Customer data file
(should be csv with cname, address):");
                                        String filename = br.readLine();

                                        //Read in file
                                        app.option18(filename, conn);

                                        //Print customers
                                        app.printCustomers(conn);
                              } else if(option.equals("19")){
                                        //Get user input
                                        System.out.println("Enter name of output file:");
                                        String filename = br.readLine();
                                        System.out.println("Enter name of paint method:");
                                        String method = br.readLine();

                                        //Read in file
                                        app.option19(filename, method, conn);
                              } else if (option.equals("20")){
                                        System.out.println("Program exiting.");
                              } else {
                                System.out.println("Invalid option entered.\n");
                        }
                } //end main loop
        } //end main method
} //end project file
```

5.2 Screenshot of Successful Compilation



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/bin/java (Nov 22, 2015, 10:18:06 AM)
WELCOME TO THE JOB SHOP ACCOUNTING DATABASE SYSTEM.
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date ordered
(4) Enter a new process-id and its department together with its type and information relevant tothe type
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts
(9) Retrieve the cost incurred on an assembly-id
(10) Retrieve the labor time recorded on an assembly-id
(11) Retrieve the total labor time within a department for jobs completed in the department during a given date
(12) Retrieve the processes through which a given assembly-id has passed so far (in date commenced order) and the d
(13) Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a gi
(14) Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method
(15) Delete all cut-jobs whose job-no is in some range
(16) Change the color of a given paint job
(17) Retrieve the average cost of all accounts
(18) Import: enter new customers from a data file until the file is empty
(19) Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method
(20) Quit
Input option #:
```

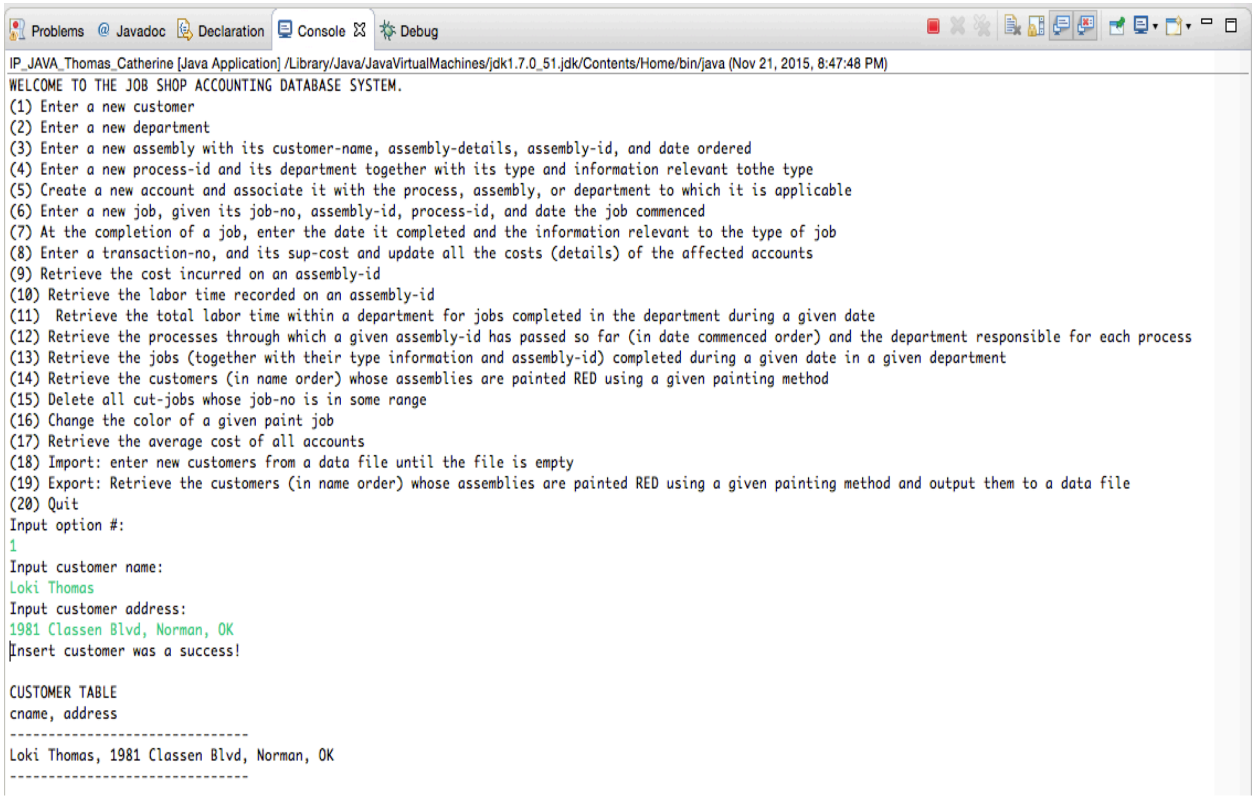
5.2 Screenshot of Java program compilation

Task 6 Java Program Execution

All Queries will have one screenshot showing the first time it ran with subsequent calls shown by output copy-pasted from the Java Console. For brevity's sake, the 'menu' of all 20 Query options will ONLY be shown on the screenshot of Query 1. This menu and how it was implemented can also be seen in the Java source code of Task 5.

6.1 Testing of Query 1

6.1.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/bin/java (Nov 21, 2015, 8:47:48 PM)
WELCOME TO THE JOB SHOP ACCOUNTING DATABASE SYSTEM.
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date ordered
(4) Enter a new process-id and its department together with its type and information relevant tothe type
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts
(9) Retrieve the cost incurred on an assembly-id
(10) Retrieve the labor time recorded on an assembly-id
(11)  Retrieve the total labor time within a department for jobs completed in the department during a given date
(12) Retrieve the processes through which a given assembly-id has passed so far (in date commenced order) and the department responsible for each process
(13) Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
(14) Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method
(15) Delete all cut-jobs whose job-no is in some range
(16) Change the color of a given paint job
(17) Retrieve the average cost of all accounts
(18) Import: enter new customers from a data file until the file is empty
(19) Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method and output them to a data file
(20) Quit
Input option #:
1
Input customer name:
Loki Thomas
Input customer address:
1981 Classen Blvd, Norman, OK
Insert customer was a success!

CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
-------------------------------
```

6.1.a Screenshot of query 1

6.1.b

```
Input option #:
1
Input customer name:
Gary Thomas
Input customer address:
18707 E 94th St
Insert customer was a success!

CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
```

```
-------------------------------
```

6.1.c

```
Input option #:
1
Input customer name:
Stefanie McDonald
Input customer address:
Space Needle, Seattle, Washington
Insert customer was a success!

CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
-------------------------------
```

6.1.d

```
Input option #:
1
Input customer name:
Cedar Floyd
Input customer address:
Over The Rainbow, FoCo, CO
Insert customer was a success!

CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
Cedar Floyd, Over The Rainbow, FoCo, CO
-------------------------------
```

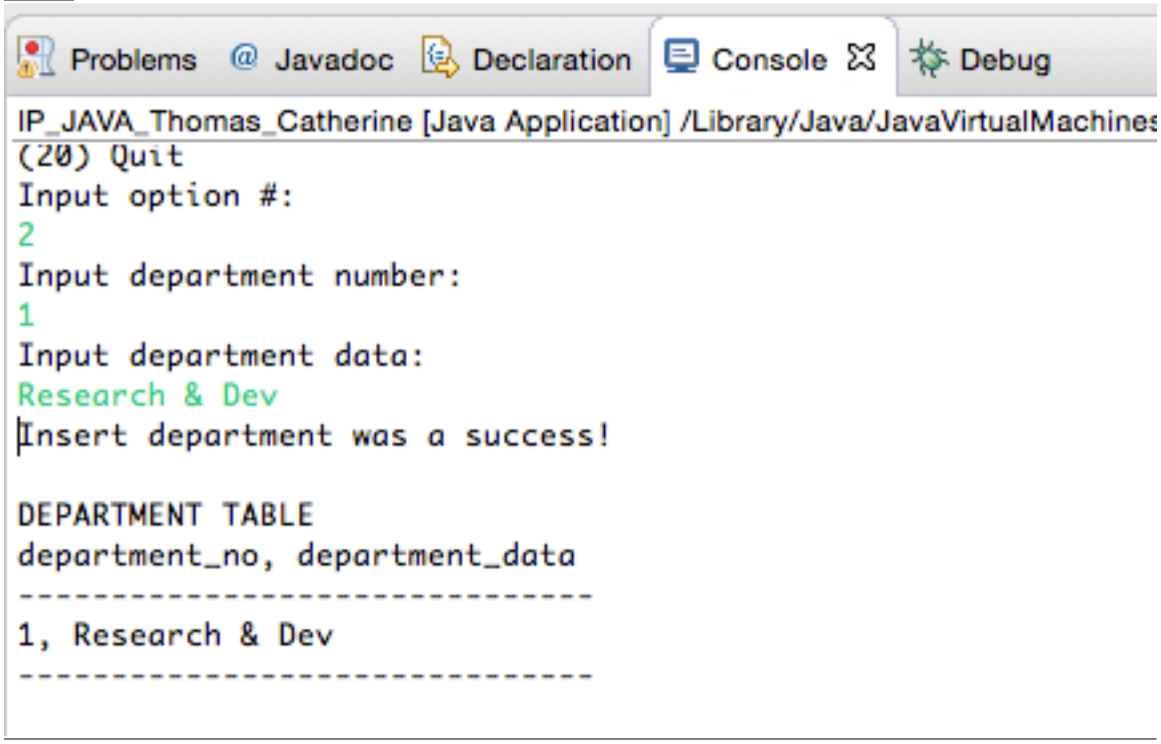
6.1.e

```
Input option #:
1
Input customer name:
George Washington
Input customer address:
Mt. Rainier, Seattle, WA
```

```
Insert customer was a success!

CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
Cedar Floyd, Over The Rainbow, FoCo, CO
George Washington, Mt. Rainier, Seattle, WA
-------------------------------
```

6.2 Testing of Query 2

6.2.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines
(20) Quit
Input option #:
2
Input department number:
1
Input department data:
Research & Dev
Insert department was a success!

DEPARTMENT TABLE
department_no, department_data
-------------------------------
1, Research & Dev
-------------------------------
```

6.2.a Screenshot of query 2

6.2.b

```
Input option #:
2
Input department number:
2
Input department data:
Painting
Insert department was a success!
```

```
DEPARTMENT TABLE
department_no, department_data
-------------------------------
1, Research & Dev
2, Painting
-------------------------------
```

6.2.c

```
Input option #:
2
Input department number:
3
Input department data:
Human Relations
Insert department was a success!

DEPARTMENT TABLE
department_no, department_data
-------------------------------
1, Research & Dev
2, Painting
3, Human Relations
-------------------------------
```

6.2.d

```
Input option #:
2
Input department number:
4
Input department data:
Accounting
Insert department was a success!

DEPARTMENT TABLE
department_no, department_data
-------------------------------
1, Research & Dev
2, Painting
3, Human Relations
4, Accounting
-------------------------------
```

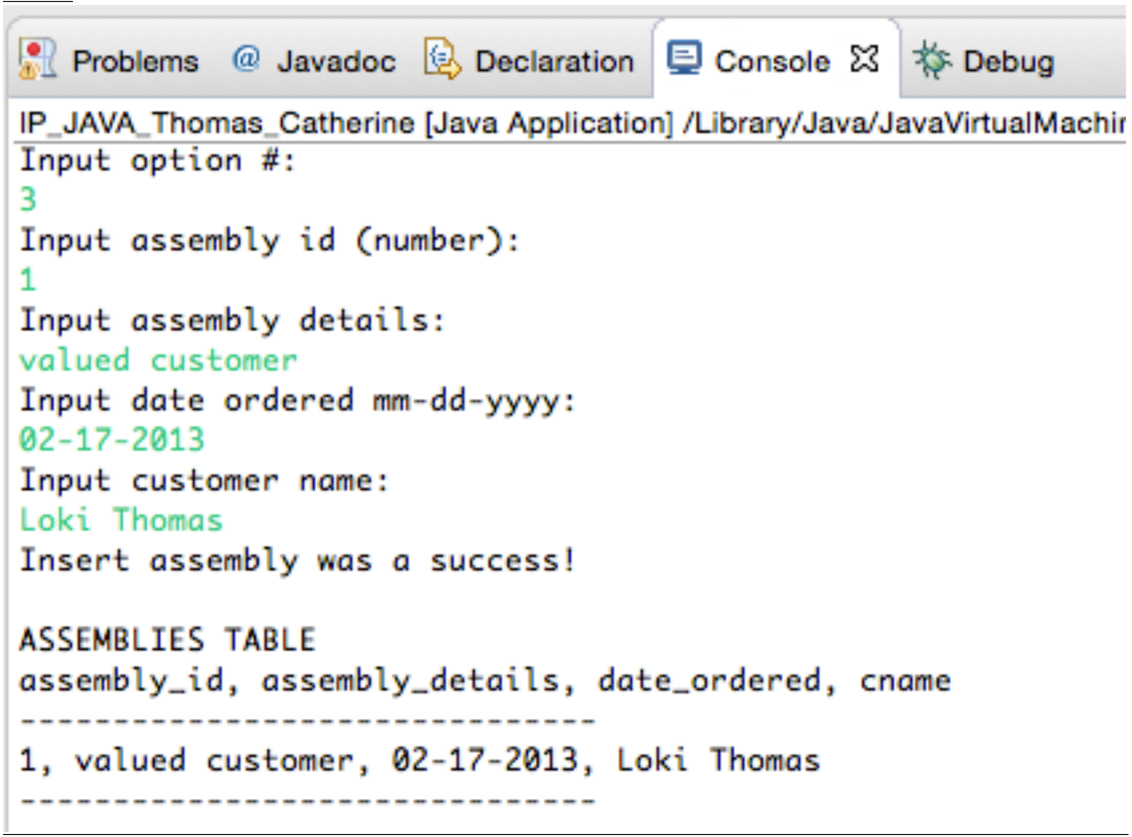
6.2.e

```
Input option #:
```

```
2
Input department number:
5
Input department data:
Information Technology
Insert department was a success!

DEPARTMENT TABLE
department_no, department_data
-------------------------------
1, Research & Dev
2, Painting
3, Human Relations
4, Accounting
5, Information Technology
-------------------------------
```

6.3 Testing of Query 3

6.3.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachir
Input option #:
3
Input assembly id (number):
1
Input assembly details:
valued customer
Input date ordered mm-dd-yyyy:
02-17-2013
Input customer name:
Loki Thomas
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
-------------------------------
```

6.3.a Screenshot of query 3

6.3.b

```
Input option #:
3
Input assembly id (number):
2
Input assembly details:
should be done quickly
Input date ordered mm-dd-yyyy:
03-14-2014
Input customer name:
Loki Thomas
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
-------------------------------
```

6.3.c

```
Input option #:
3
Input assembly id (number):
3
Input assembly details:
complicated work
Input date ordered mm-dd-yyyy:
05-05-2005
Input customer name:
George Washington
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
-------------------------------
```

6.3.d

```
Input option #:
3
```

```
Input assembly id (number):
4
Input assembly details:
vibrant coloring
Input date ordered mm-dd-yyyy:
04-04-2004
Input customer name:
Stefanie McDonald
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
-------------------------------
```

6.3.e

```
Input option #:
3
Input assembly id (number):
5
Input assembly details:
cherry wood
Input date ordered mm-dd-yyyy:
06-06-2015
Input customer name:
George Washington
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
-------------------------------
```

6.3.f

```
Input option #:
```

```
3
Input assembly id (number):
7
Input assembly details:
new customer
Input date ordered mm-dd-yyyy:
01-03-2015
Input customer name:
Gary Thomas
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
-------------------------------
```

6.3.g

```
Input option #:
3
Input assembly id (number):
6
Input assembly details:
renovating van
Input date ordered mm-dd-yyyy:
07-13-2011
Input customer name:
Cedar Floyd
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
```

```
6, renovating van, 07-13-2011, Cedar Floyd
-------------------------------
```

6.3.h

```
Input option #:
3
Input assembly id (number):
8
Input assembly details:
flexible with assembly
Input date ordered mm-dd-yyyy:
08-08-2008
Input customer name:
Loki Thomas
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
6, renovating van, 07-13-2011, Cedar Floyd
8, flexible with assembly, 08-08-2008, Loki Thomas
-------------------------------
```

6.3.i

```
Input option #:
3
Input assembly id (number):
9
Input assembly details:
pet friendly
Input date ordered mm-dd-yyyy:
12-25-2015
Input customer name:
Gary Thomas
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
```

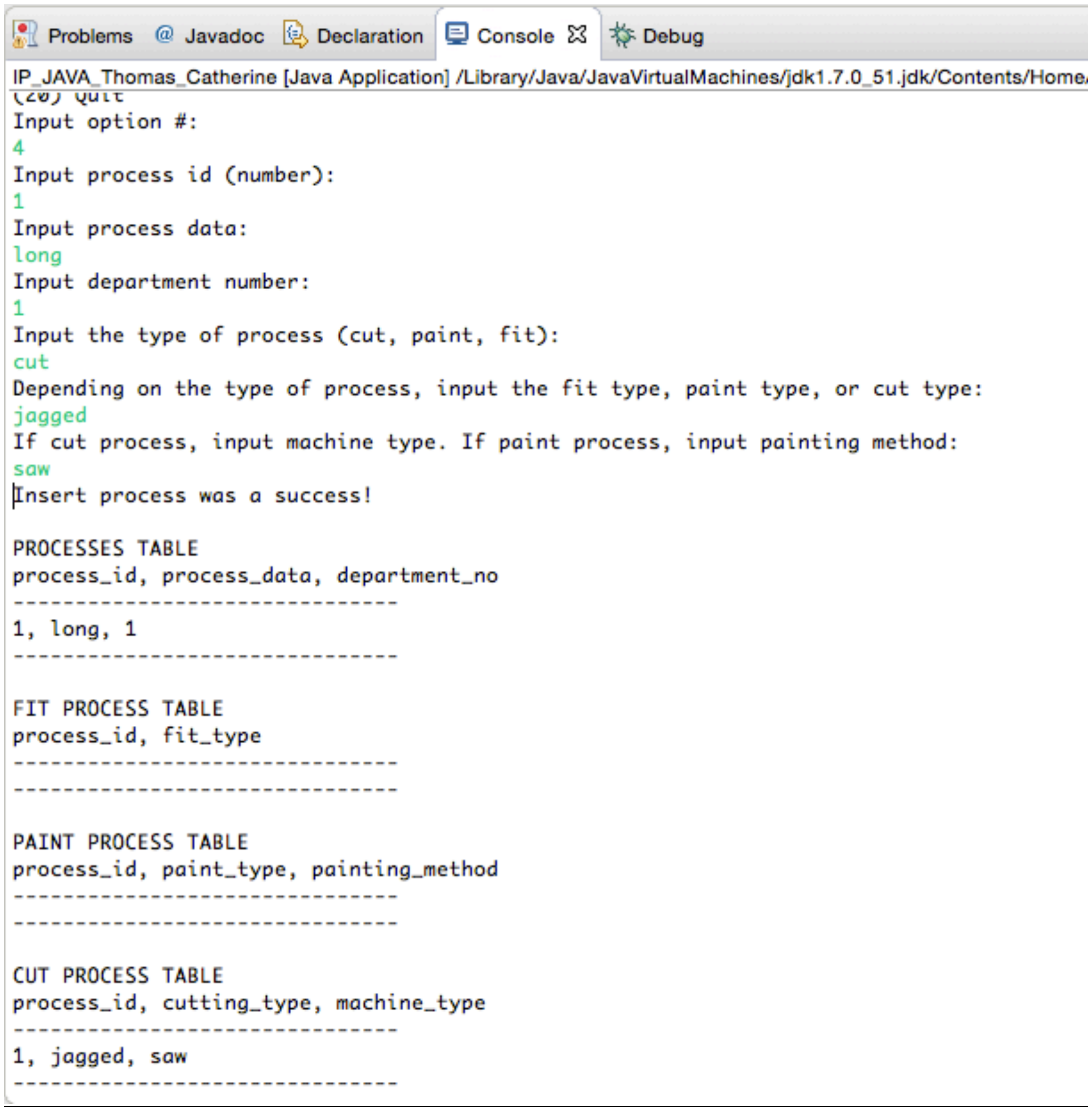
```
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
6, renovating van, 07-13-2011, Cedar Floyd
8, flexible with assembly, 08-08-2008, Loki Thomas
9, pet friendly, 12-25-2015, Gary Thomas
-------------------------------
```

6.3.j

```
Input option #:
3
Input assembly id (number):
10
Input assembly details:
complicated
Input date ordered mm-dd-yyyy:
07-07-2012
Input customer name:
Stefanie McDonald
Insert assembly was a success!

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
6, renovating van, 07-13-2011, Cedar Floyd
8, flexible with assembly, 08-08-2008, Loki Thomas
9, pet friendly, 12-25-2015, Gary Thomas
10, complicated, 07-07-2012, Stefanie McDonald
-------------------------------
```

6.4 Testing of Query 4

6.4.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/
(20) Quit
Input option #:
4
Input process id (number):
1
Input process data:
long
Input department number:
1
Input the type of process (cut, paint, fit):
cut
Depending on the type of process, input the fit type, paint type, or cut type:
jagged
If cut process, input machine type. If paint process, input painting method:
saw
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
1, long, 1
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
-------------------------------
```

6.4.a Screenshot of query 4

6.4.b

```
Input option #:
4
Input process id (number):
2
Input process data:
scissors cutting
Input department number:
2
Input the type of process (cut, paint, fit):
cut
Depending on the type of process, input the fit type, paint type,
or cut type:
straight
If cut process, input machine type. If paint process, input
painting method:
scissors
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
1, long, 1
2, scissors cutting, 2
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
-------------------------------
```

6.4.c

```
Input option #:
4
Input process id (number):
3
Input process data:
spray paint
Input department number:
3
Input the type of process (cut, paint, fit):
paint
Depending on the type of process, input the fit type, paint type,
or cut type:
```

```
solid
If cut process, input machine type. If paint process, input
painting method:
spray
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
-------------------------------
```

6.4.d

```
Input option #:
4
Input process id (number):
4
Input process data:
fit
Input department number:
4
Input the type of process (cut, paint, fit):
fit
Depending on the type of process, input the fit type, paint type,
or cut type:
tight
If cut process, input machine type. If paint process, input
painting method:
none
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
-------------------------------
```

6.4.e

```
Input option #:
4
Input process id (number):
5
Input process data:
brush
Input department number:
5
```

```
Input the type of process (cut, paint, fit):
paint
Depending on the type of process, input the fit type, paint type,
or cut type:
spots
If cut process, input machine type. If paint process, input
painting method:
brush
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
5, spots, brush
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
-------------------------------
```

6.4.f

```
Input option #:
4
Input process id (number):
```

```
6
Input process data:
loose
Input department number:
1
Input the type of process (cut, paint, fit):
fit
Depending on the type of process, input the fit type, paint type,
or cut type:
loose
If cut process, input machine type. If paint process, input
painting method:
none
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
6, loose, 1
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
6, loose
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
5, spots, brush
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
```

```
2, straight, scissors
-------------------------------
```

6.4.g

```
Input option #:
4
Input process id (number):
7
Input process data:
crooked
Input department number:
3
Input the type of process (cut, paint, fit):
cut
Depending on the type of process, input the fit type, paint type,
or cut type:
crooked
If cut process, input machine type. If paint process, input
painting method:
handsaw
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
6, loose, 1
7, crooked, 3
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
6, loose
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
```

```
3, solid, spray
5, spots, brush
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
7, crooked, handsaw
-------------------------------
```

6.4.h

```
Input option #:
4
Input process id (number):
8
Input process data:
stripes
Input department number:
3
Input the type of process (cut, paint, fit):
paint
Depending on the type of process, input the fit type, paint type,
or cut type:
striped
If cut process, input machine type. If paint process, input
painting method:
sponged
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
6, loose, 1
7, crooked, 3
8, stripes, 3
-------------------------------
```

```
FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
6, loose
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
5, spots, brush
8, striped, sponged
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
7, crooked, handsaw
-------------------------------
```

6.4.i

```
Input option #:
4
Input process id (number):
9
Input process data:
cutting edge
Input department number:
2
Input the type of process (cut, paint, fit):
cut
Depending on the type of process, input the fit type, paint type,
or cut type:
straight
If cut process, input machine type. If paint process, input
painting method:
laser
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
```

```
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
6, loose, 1
7, crooked, 3
8, stripes, 3
9, cutting edge, 2
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
6, loose
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
5, spots, brush
8, striped, sponged
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
1, jagged, saw
2, straight, scissors
7, crooked, handsaw
9, straight, laser
-------------------------------
```

6.4.j

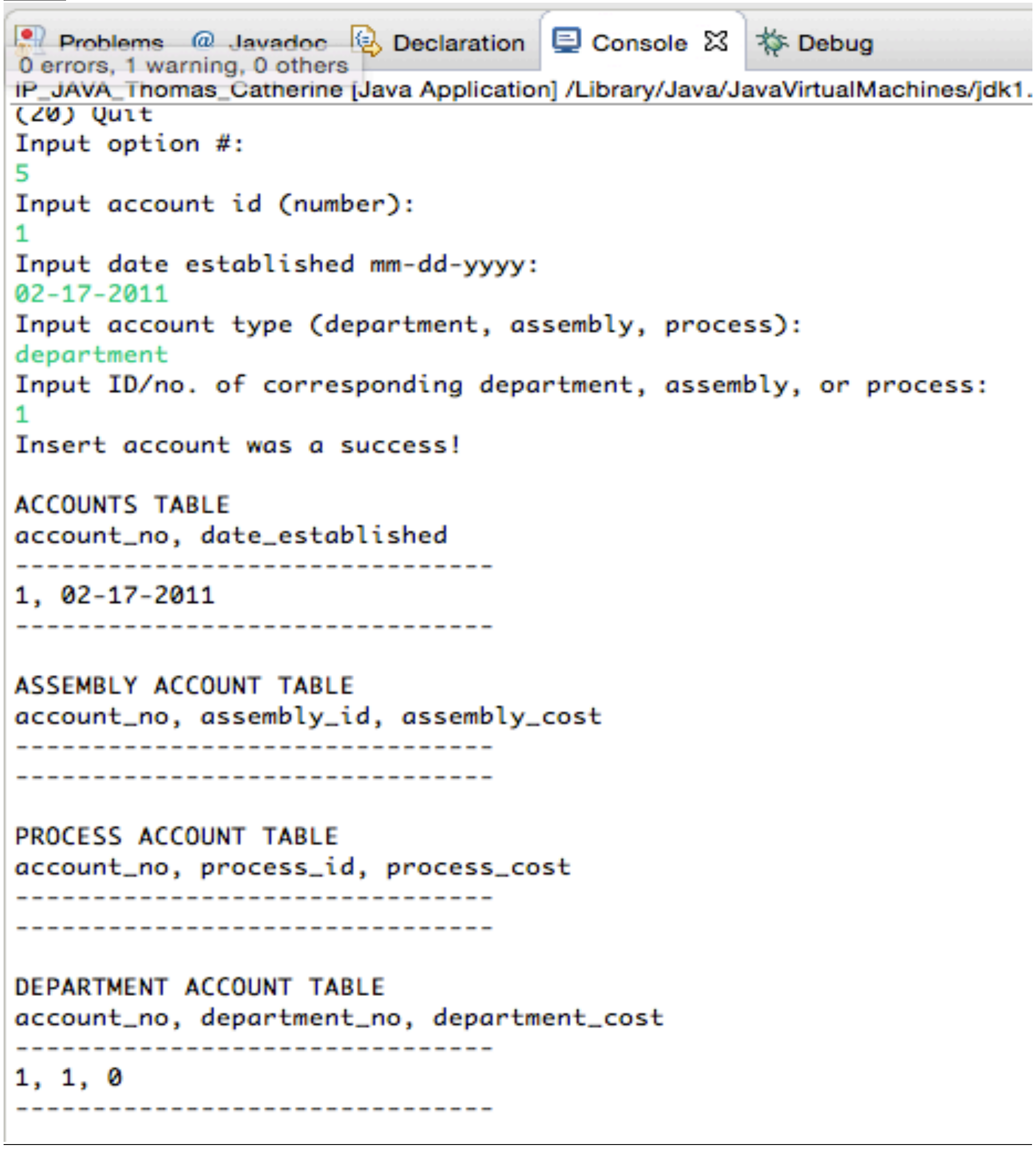
```
Input option #:
4
Input process id (number):
10
Input process data:
saw
Input department number:
```

```
5
Input the type of process (cut, paint, fit):
cut
Depending on the type of process, input the fit type, paint type,
or cut type:
circle
If cut process, input machine type. If paint process, input
painting method:
saw
Insert process was a success!

PROCESSES TABLE
process_id, process_data, department_no
-------------------------------
3, spray paint, 3
1, long, 1
2, scissors cutting, 2
4, fit, 4
5, brush, 5
6, loose, 1
7, crooked, 3
8, stripes, 3
9, cutting edge, 2
10, saw, 5
-------------------------------

FIT PROCESS TABLE
process_id, fit_type
-------------------------------
4, tight
6, loose
-------------------------------

PAINT PROCESS TABLE
process_id, paint_type, painting_method
-------------------------------
3, solid, spray
5, spots, brush
8, striped, sponged
-------------------------------

CUT PROCESS TABLE
process_id, cutting_type, machine_type
-------------------------------
```

```
1, jagged, saw
2, straight, scissors
7, crooked, handsaw
9, straight, laser
10, circle, saw
-------------------------------
```

6.5 Testing of Query 5

6.5.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.
(20) Quit
Input option #:
5
Input account id (number):
1
Input date established mm-dd-yyyy:
02-17-2011
Input account type (department, assembly, process):
department
Input ID/no. of corresponding department, assembly, or process:
1
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
1, 02-17-2011
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
-------------------------------
```

6.5.a Screenshot of Query 5

6.5.b

```
Input option #:
5
Input account id (number):
2
Input date established mm-dd-yyyy:
09-25-2012
Input account type (department, assembly, process):
process
Input ID/no. of corresponding department, assembly, or process:
1
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
1, 02-17-2011
2, 09-25-2012
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
-------------------------------
```

6.5.c

```
Input option #:
5
Input account id (number):
3
Input date established mm-dd-yyyy:
08-12-1234
Input account type (department, assembly, process):
```

```
assembly
Input ID/no. of corresponding department, assembly, or process:
1
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
-------------------------------
```

6.5.d

```
Input option #:
5
Input account id (number):
4
Input date established mm-dd-yyyy:
07-07-2007
Input account type (department, assembly, process):
department
Input ID/no. of corresponding department, assembly, or process:
2
Insert account was a success!
```

```
ACCOUNTS TABLE
account_no, date_established
-------------------------------
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
4, 2, 0
-------------------------------
```

6.5.e

```
Input option #:
5
Input account id (number):
6
Input date established mm-dd-yyyy:
08-17-2014
Input account type (department, assembly, process):
assembly
Input ID/no. of corresponding department, assembly, or process:
3
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
```

```
6, 08-17-2014
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
4, 2, 0
-------------------------------
```

6.5.f

```
Input option #:
5
Input account id (number):
7
Input date established mm-dd-yyyy:
06-06-2011
Input account type (department, assembly, process):
assembly
Input ID/no. of corresponding department, assembly, or process:
2
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
6, 08-17-2014
```

```
7, 06-06-2011
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
4, 2, 0
-------------------------------
```

6.5.g

```
Input option #:
5
Input account id (number):
8
Input date established mm-dd-yyyy:
07-27-2012
Input account type (department, assembly, process):
process
Input ID/no. of corresponding department, assembly, or process:
2
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
```

```
6, 08-17-2014
7, 06-06-2011
8, 07-27-2012
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
1, 1, 0
4, 2, 0
-------------------------------
```

6.5.h

```
Input option #:
5
Input account id (number):
9
Input date established mm-dd-yyyy:
04-17-2003
Input account type (department, assembly, process):
department
Input ID/no. of corresponding department, assembly, or process:
3
Insert account was a success!
```

```
ACCOUNTS TABLE
account_no, date_established
-------------------------------
6, 08-17-2014
7, 06-06-2011
8, 07-27-2012
9, 04-17-2003
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 0
1, 1, 0
4, 2, 0
-------------------------------
```

6.5.i

```
Input option #:
5
Input account id (number):
10
Input date established mm-dd-yyyy:
06-15-2009
Input account type (department, assembly, process):
```

```
process
Input ID/no. of corresponding department, assembly, or process:
3
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
6, 08-17-2014
7, 06-06-2011
8, 07-27-2012
9, 04-17-2003
10, 06-15-2009
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
10, 3, 0
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 0
1, 1, 0
4, 2, 0
-------------------------------
```

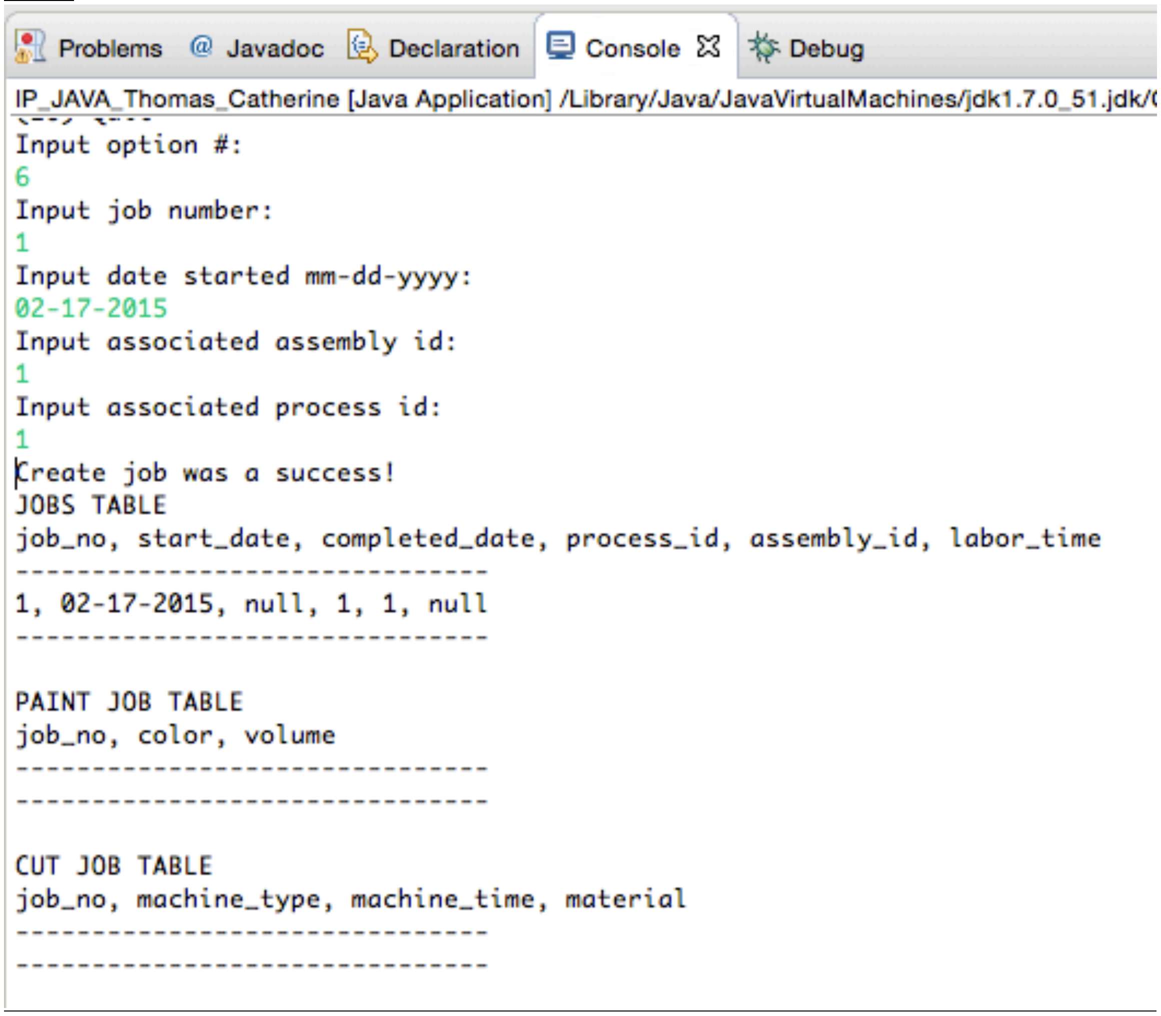
6.5.j

```
Input option #:
5
Input account id (number):
11
Input date established mm-dd-yyyy:
08-14-2006
Input account type (department, assembly, process):
department
Input ID/no. of corresponding department, assembly, or process:
4
Insert account was a success!

ACCOUNTS TABLE
account_no, date_established
-------------------------------
6, 08-17-2014
7, 06-06-2011
8, 07-27-2012
9, 04-17-2003
10, 06-15-2009
11, 08-14-2006
1, 02-17-2011
2, 09-25-2012
3, 08-12-1234
4, 07-07-2007
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
3, 1, 0
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
10, 3, 0
2, 1, 0
-------------------------------

DEPARTMENT ACCOUNT TABLE
```

```
account_no, department_no, department_cost
-------------------------------
9, 3, 0
11, 4, 0
1, 1, 0
4, 2, 0
-------------------------------
```

6.6 Testing for Query 6

6.6.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/
Input option #:
6
Input job number:
1
Input date started mm-dd-yyyy:
02-17-2015
Input associated assembly id:
1
Input associated process id:
1
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id, labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.a Screenshot for Query 6

6.6.b

```
Input option #:
6
Input job number:
2
Input date started mm-dd-yyyy:
```

```
09-18-2011
Input associated assembly id:
2
Input associated process id:
2
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.c

```
Input option #:
6
Input job number:
3
Input date started mm-dd-yyyy:
08-17-2007
Input associated assembly id:
3
Input associated process id:
3
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
-------------------------------
```

```
PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.d

```
Input option #:
6
Input job number:
4
Input date started mm-dd-yyyy:
03-14-2006
Input associated assembly id:
1
Input associated process id:
4
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
```

6.6.e

```
Input option #:
6
Input job number:
5
Input date started mm-dd-yyyy:
04-09-2008
Input associated assembly id:
1
Input associated process id:
5
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.f

```
Input option #:
6
Input job number:
6
Input date started mm-dd-yyyy:
07-31-1004
Input associated assembly id:
6
Input associated process id:
7
Create job was a success!
```

```
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.g

```
Input option #:
6
Input job number:
7
Input date started mm-dd-yyyy:
08-10-2010
Input associated assembly id:
4
Input associated process id:
3
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
```

```
7, 08-10-2010, null, 3, 4, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.h

```
Input option #:
6
Input job number:
8
Input date started mm-dd-yyyy:
05-05-2005
Input associated assembly id:
4
Input associated process id:
3
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------
```

```
CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.i

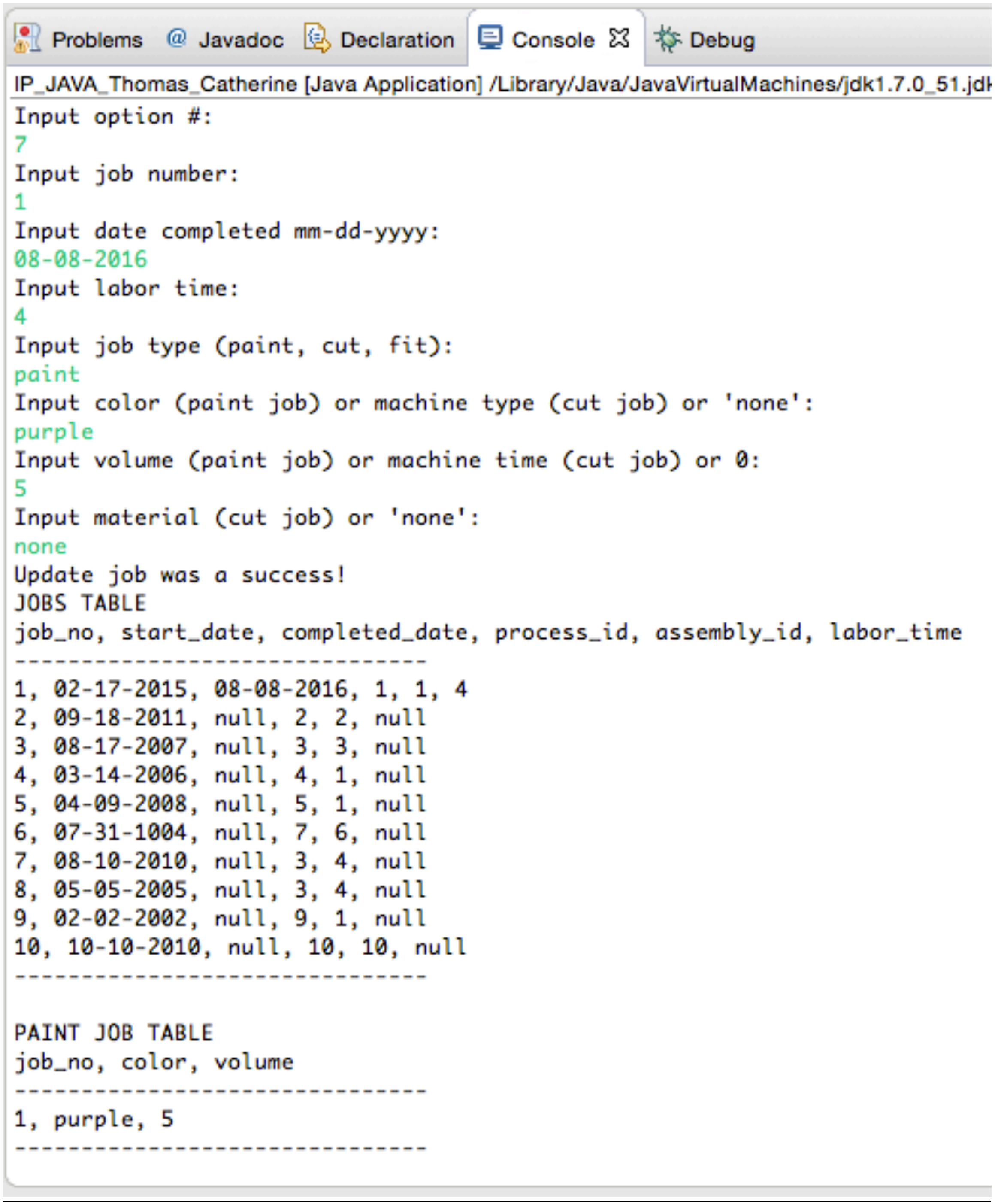
```
Input option #:
6
Input job number:
9
Input date started mm-dd-yyyy:
02-02-2002
Input associated assembly id:
1
Input associated process id:
9
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
9, 02-02-2002, null, 9, 1, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.6.j

```
Input option #:
6
Input job number:
10
Input date started mm-dd-yyyy:
10-10-2010
Input associated assembly id:
10
Input associated process id:
10
Create job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, null, 1, 1, null
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
9, 02-02-2002, null, 9, 1, null
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.7 Testing for Query 7

6.7.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk
Input option #:
7
Input job number:
1
Input date completed mm-dd-yyyy:
08-08-2016
Input labor time:
4
Input job type (paint, cut, fit):
paint
Input color (paint job) or machine type (cut job) or 'none':
purple
Input volume (paint job) or machine time (cut job) or 0:
5
Input material (cut job) or 'none':
none
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id, labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, null, 2, 2, null
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
9, 02-02-2002, null, 9, 1, null
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
1, purple, 5
-------------------------------
```

6.7.a Screenshot for Query 7

6.7.b

```
Input option #:
7
Input job number:
2
Input date completed mm-dd-yyyy:
09-10-2016
```

```
Input labor time:
10
Input job type (paint, cut, fit):
fit
Input color (paint job) or machine type (cut job) or 'none':
none
Input volume (paint job) or machine time (cut job) or 0:
0
Input material (cut job) or 'none':
none
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, null, 4, 1, null
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
9, 02-02-2002, null, 9, 1, null
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
1, purple, 5
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.7.c

```
Input option #:
7
Input job number:
4
Input date completed mm-dd-yyyy:
08-17-2016
```

```
Input labor time:
17
Input job type (paint, cut, fit):
paint
Input color (paint job) or machine type (cut job) or 'none':
RED
Input volume (paint job) or machine time (cut job) or 0:
10
Input material (cut job) or 'none':
none
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, null, 3, 4, null
9, 02-02-2002, null, 9, 1, null
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
-------------------------------
```

6.7.d

```
Input option #:
7
Input job number:
8
```

```
Input date completed mm-dd-yyyy:
09-10-2016
Input labor time:
2
Input job type (paint, cut, fit):
cut
Input color (paint job) or machine type (cut job) or 'none':
scissors
Input volume (paint job) or machine time (cut job) or 0:
2
Input material (cut job) or 'none':
carpet
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, null, 9, 1, null
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
-------------------------------
```

6.7.e

```
Input option #:
7
Input job number:
9
Input date completed mm-dd-yyyy:
07-07-2016
Input labor time:
4
Input job type (paint, cut, fit):
fit
Input color (paint job) or machine type (cut job) or 'none':
none
Input volume (paint job) or machine time (cut job) or 0:
0
Input material (cut job) or 'none':
none
An error occurred on option 7.
java.sql.SQLException: SQL statement to execute cannot be empty
or null
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, null, 10, 10, null
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
```

```
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
-------------------------------
```

6.7.f

```
Input option #:
7
Input job number:
10
Input date completed mm-dd-yyyy:
12-12-2016
Input labor time:
1
Input job type (paint, cut, fit):
paint
Input color (paint job) or machine type (cut job) or 'none':
RED
Input volume (paint job) or machine time (cut job) or 0:
1
Input material (cut job) or 'none':
none
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, null, 3, 3, null
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
1, purple, 5
```

```
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
-------------------------------
```

6.7.g

```
Input option #:
7
Input job number:
3
Input date completed mm-dd-yyyy:
12-12-2016
Input labor time:
5
Input job type (paint, cut, fit):
cut
Input color (paint job) or machine type (cut job) or 'none':
handsaw
Input volume (paint job) or machine time (cut job) or 0:
7
Input material (cut job) or 'none':
leather
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, null, 5, 1, null
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
```

```
job_no, color, volume
-------------------------------
10, RED, 1
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
3, handsaw, 7, leather
-------------------------------
```

6.7.h

```
Input option #:
7
Input job number:
5
Input date completed mm-dd-yyyy:
11-11-2015
Input labor time:
5
Input job type (paint, cut, fit):
paint
Input color (paint job) or machine type (cut job) or 'none':
RED
Input volume (paint job) or machine time (cut job) or 0:
2
Input material (cut job) or 'none':
none
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, null, 7, 6, null
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
```

```
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
3, handsaw, 7, leather
-------------------------------
```

6.7.i

```
Input option #:
7
Input job number:
6
Input date completed mm-dd-yyyy:
09-09-2017
Input labor time:
8
Input job type (paint, cut, fit):
paint
Input color (paint job) or machine type (cut job) or 'none':
RED
Input volume (paint job) or machine time (cut job) or 0:
7
Input material (cut job) or 'none':
none
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
```

```
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, null, 3, 4, null
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
3, handsaw, 7, leather
-------------------------------
```

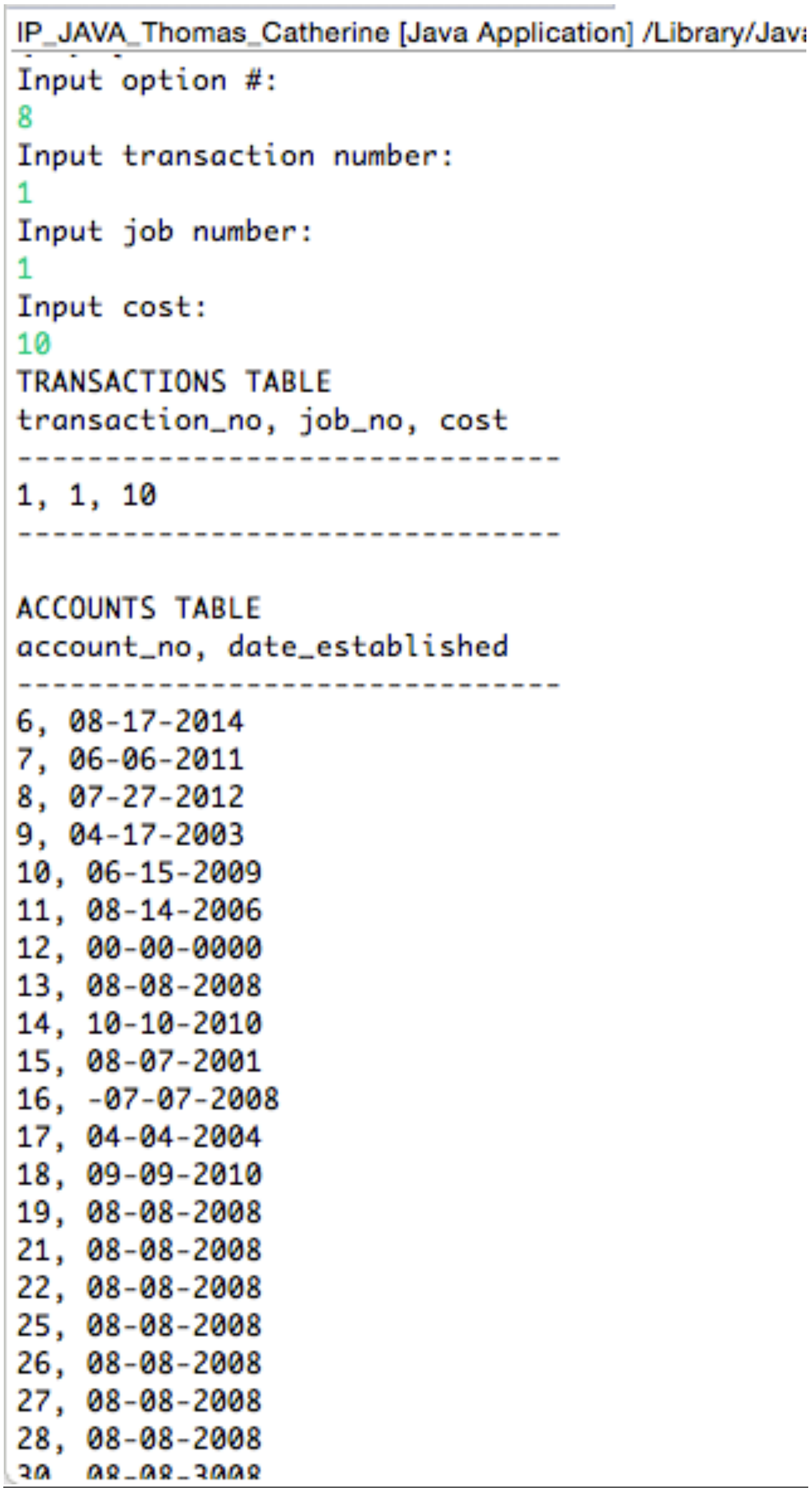
6.7.j

```
Input option #:
7
Input job number:
7
Input date completed mm-dd-yyyy:
11-11-2011
Input labor time:
9
Input job type (paint, cut, fit):
cut
Input color (paint job) or machine type (cut job) or 'none':
saw
Input volume (paint job) or machine time (cut job) or 0:
4
Input material (cut job) or 'none':
wood
```

```
Update job was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
3, handsaw, 7, leather
7, saw, 4, wood
-------------------------------
```

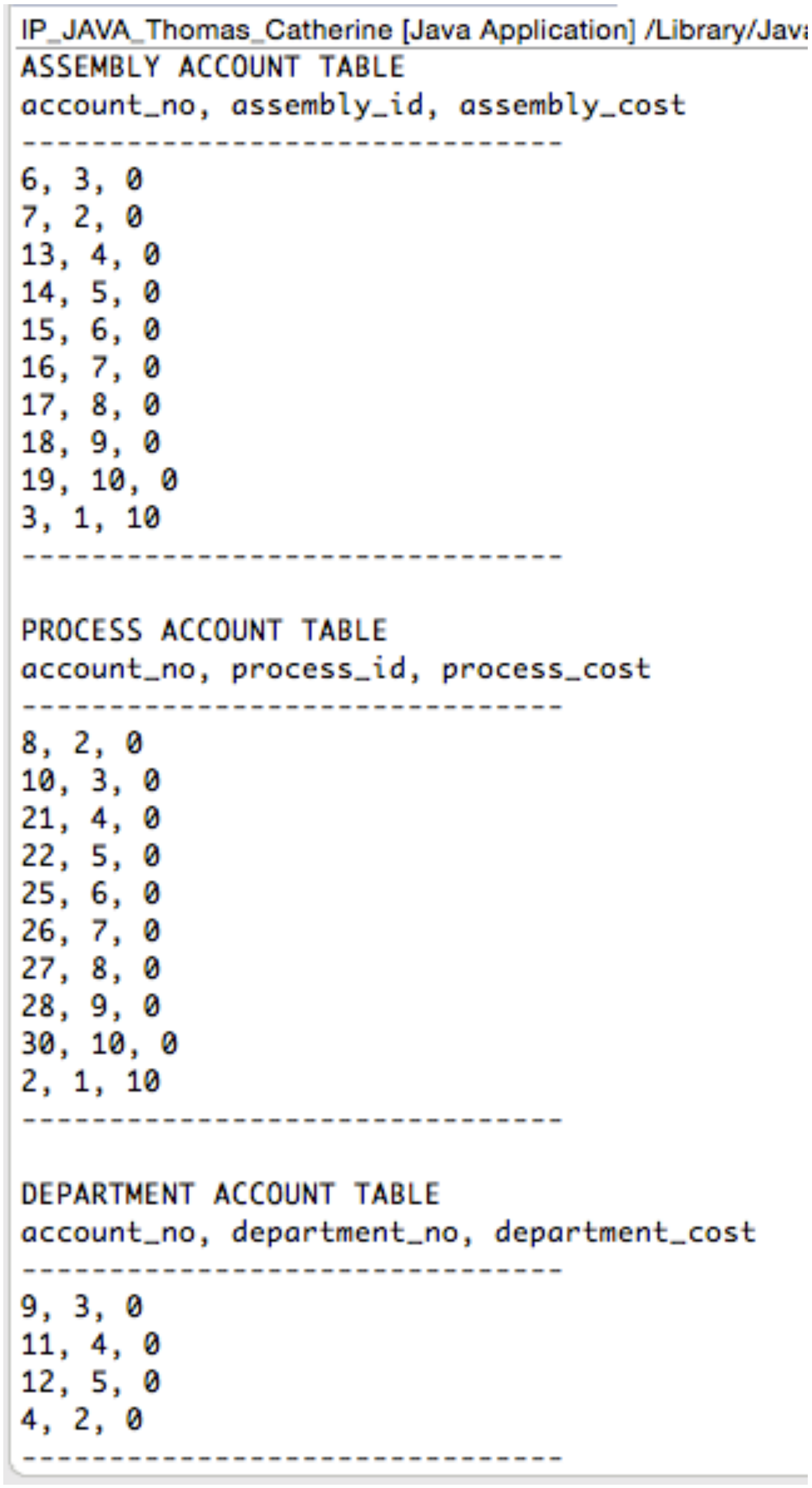
6.8 Testing for Query 8

6.8.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Jav
Input option #:
8
Input transaction number:
1
Input job number:
1
Input cost:
10
TRANSACTIONS TABLE
transaction_no, job_no, cost
------------------------------
1, 1, 10
------------------------------

ACCOUNTS TABLE
account_no, date_established
------------------------------
6, 08-17-2014
7, 06-06-2011
8, 07-27-2012
9, 04-17-2003
10, 06-15-2009
11, 08-14-2006
12, 00-00-0000
13, 08-08-2008
14, 10-10-2010
15, 08-07-2001
16, -07-07-2008
17, 04-04-2004
18, 09-09-2010
19, 08-08-2008
21, 08-08-2008
22, 08-08-2008
25, 08-08-2008
26, 08-08-2008
27, 08-08-2008
28, 08-08-2008
30, 08-08-2008
```

6.8.a1 Screen shot of Query 8



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Jav
ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 0
7, 2, 0
13, 4, 0
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 10
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
10, 3, 0
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 10
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 0
11, 4, 0
12, 5, 0
4, 2, 0
-------------------------------
```

6.8.a2 Screen shot of Query 8

6.8.b

```
Input option #:
8
Input transaction number:
2
Input job number:
3
Input cost:
20
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 0
13, 4, 0
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 10
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
10, 3, 20
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 10
```

```
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 20
11, 4, 0
12, 5, 0
31, 1, 0
4, 2, 0
-------------------------------
```

6.8.c

```
Input option #:
8
Input transaction number:
3
Input job number:
1
Input cost:
15
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 0
13, 4, 0
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 25
-------------------------------
```

```
PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 0
10, 3, 20
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 20
11, 4, 0
12, 5, 0
31, 1, 15
4, 2, 0
-------------------------------
```

6.8.d

```
Input option #:
8
Input transaction number:
4
Input job number:
2
Input cost:
30
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
-------------------------------

ASSEMBLY ACCOUNT TABLE
```

```
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 30
13, 4, 0
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 25
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 30
10, 3, 20
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 20
11, 4, 0
12, 5, 0
31, 1, 15
4, 2, 30
-------------------------------
```

6.8.e

```
Input option #:
8
Input transaction number:
5
```

```
Input job number:
7
Input cost:
200
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 30
13, 4, 200
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 25
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 30
10, 3, 220
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------
```

```
DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 220
11, 4, 0
12, 5, 0
31, 1, 15
4, 2, 30
-------------------------------
```

6.8.f

```
Input option #:
8
Input transaction number:
6
Input job number:
2
Input cost:
46
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
6, 2, 46
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 76
13, 4, 200
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 25
-------------------------------
```

```
PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 76
10, 3, 220
21, 4, 0
22, 5, 0
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 220
11, 4, 0
12, 5, 0
31, 1, 15
4, 2, 76
-------------------------------
```

6.8.g

```
Input option #:
8
Input transaction number:
7
Input job number:
5
Input cost:
12
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
6, 2, 46
```

```
7, 5, 12
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 76
13, 4, 200
14, 5, 0
15, 6, 0
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 37
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 76
10, 3, 220
21, 4, 0
22, 5, 12
25, 6, 0
26, 7, 0
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 220
11, 4, 0
12, 5, 12
31, 1, 15
4, 2, 76
-------------------------------
```

6.8.h

```
Input option #:
8
Input transaction number:
8
Input job number:
6
Input cost:
60
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
6, 2, 46
7, 5, 12
8, 6, 60
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 76
13, 4, 200
14, 5, 0
15, 6, 60
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 37
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 76
10, 3, 220
21, 4, 0
22, 5, 12
25, 6, 0
```

```
26, 7, 60
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 280
11, 4, 0
12, 5, 12
31, 1, 15
4, 2, 76
-------------------------------
```

6.8.i

```
Input option #:
8
Input transaction number:
9
Input job number:
8
Input cost:
34
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
6, 2, 46
7, 5, 12
8, 6, 60
9, 8, 34
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
```

```
7, 2, 76
13, 4, 234
14, 5, 0
15, 6, 60
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 37
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 76
10, 3, 254
21, 4, 0
22, 5, 12
25, 6, 0
26, 7, 60
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 314
11, 4, 0
12, 5, 12
31, 1, 15
4, 2, 76
-------------------------------
```

6.8.j

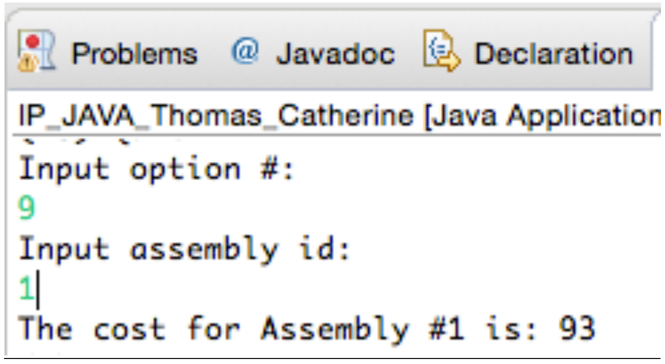
```
Input option #:
8
Input transaction number:
10
Input job number:
4
Input cost:
```

```
56
TRANSACTIONS TABLE
transaction_no, job_no, cost
-------------------------------
1, 1, 10
2, 3, 20
3, 1, 15
4, 2, 30
5, 7, 200
6, 2, 46
7, 5, 12
8, 6, 60
9, 8, 34
10, 4, 56
-------------------------------

ASSEMBLY ACCOUNT TABLE
account_no, assembly_id, assembly_cost
-------------------------------
6, 3, 20
7, 2, 76
13, 4, 234
14, 5, 0
15, 6, 60
16, 7, 0
17, 8, 0
18, 9, 0
19, 10, 0
3, 1, 93
-------------------------------

PROCESS ACCOUNT TABLE
account_no, process_id, process_cost
-------------------------------
8, 2, 76
10, 3, 254
21, 4, 56
22, 5, 12
25, 6, 0
26, 7, 60
27, 8, 0
28, 9, 0
30, 10, 0
2, 1, 25
```

```
-------------------------------

DEPARTMENT ACCOUNT TABLE
account_no, department_no, department_cost
-------------------------------
9, 3, 314
11, 4, 56
12, 5, 12
31, 1, 15
4, 2, 76
-------------------------------
```

6.9 Testing for Query 9

<u>**6.9.a**</u>



```
Input option #:
9
Input assembly id:
1
The cost for Assembly #1 is: 93
```

6.9.a Screenshot for Query 9

<u>**6.9.b**</u>

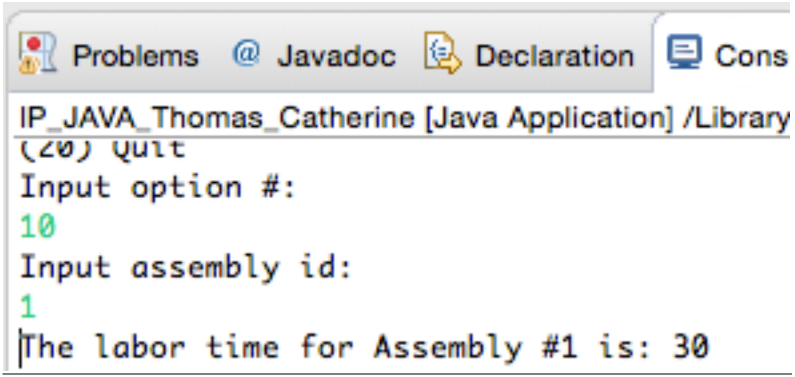
```
Input option #:
9
Input assembly id:
4
The cost for Assembly #4 is: 234
```

<u>**6.9.c**</u>

```
Input option #:
9
Input assembly id:
3
The cost for Assembly #3 is: 20
```

6.10 Testing for Query 10

6.10.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library
(20) Quit
Input option #:
10
Input assembly id:
1
The labor time for Assembly #1 is: 30
```

6.10.a Screenshot for Query 10

6.10.b

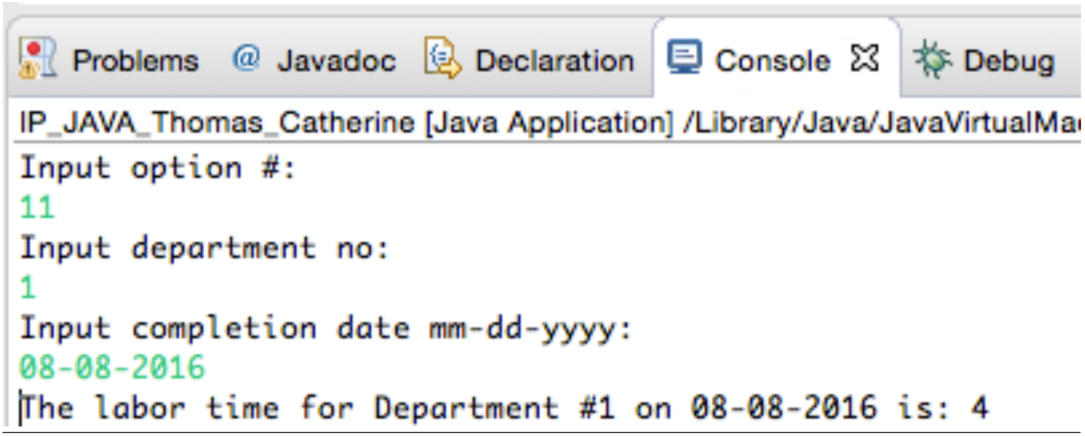
```
Input option #:
10
Input assembly id:
4
The labor time for Assembly #4 is: 11
```

6.10.c

```
Input option #:
10
Input assembly id:
6
The labor time for Assembly #6 is: 8
```

6.11 Testing for Query 11

6.11.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMa
Input option #:
11
Input department no:
1
Input completion date mm-dd-yyyy:
08-08-2016
The labor time for Department #1 on 08-08-2016 is: 4
```

6.11.b

```
Input option #:
```

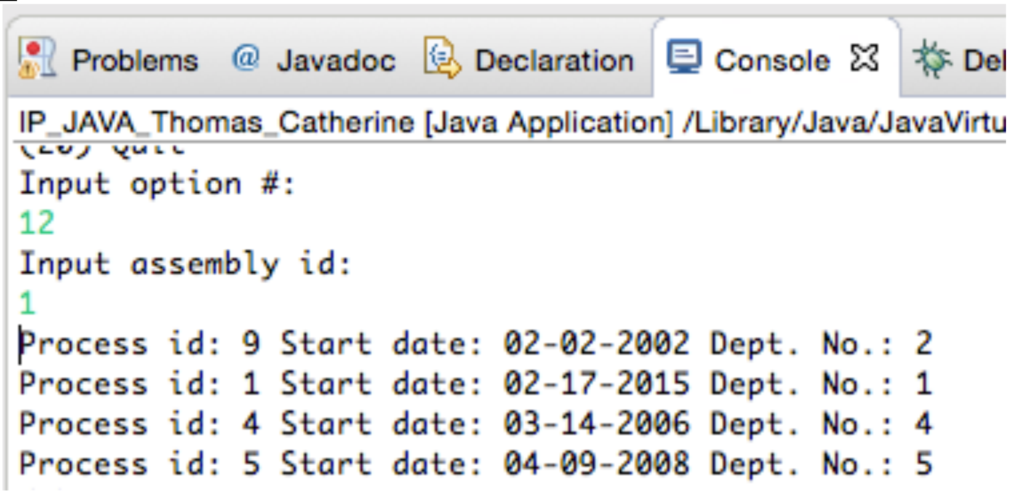
```
11
Input department no:
1
Input completion date mm-dd-yyyy:
02-02-2016
The labor time for Department #1 on 02-02-2016 is: null
```

6.11.c

```
Input option #:
11
Input department no:
3
Input completion date mm-dd-yyyy:
11-11-2011
The labor time for Department #3 on 11-11-2011 is: 9
```

6.12 Testing for Query 12

6.12.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtu
Input option #:
12
Input assembly id:
1
Process id: 9 Start date: 02-02-2002 Dept. No.: 2
Process id: 1 Start date: 02-17-2015 Dept. No.: 1
Process id: 4 Start date: 03-14-2006 Dept. No.: 4
Process id: 5 Start date: 04-09-2008 Dept. No.: 5
```

6.12.a Screenshot of Query 12

6.12.b

```
Input option #:
12
Input assembly id:
2
Process id: 2 Start date: 09-18-2011 Dept. No.: 2
```

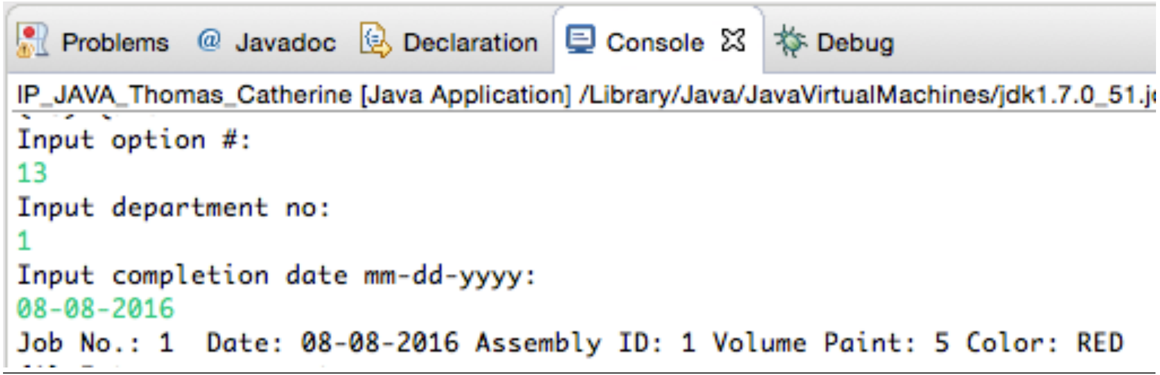
6.12.c

```
Input option #:
12
Input assembly id:
```

```
3
Process id: 3 Start date: 08-17-2007 Dept. No.: 3
```

6.13 Testing for Query 13

<u>6.13.a</u>



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.j
Input option #:
13
Input department no:
1
Input completion date mm-dd-yyyy:
08-08-2016
Job No.: 1  Date: 08-08-2016 Assembly ID: 1 Volume Paint: 5 Color: RED
```

6.13.a Screenshot of Query 13

<u>6.13.b</u>

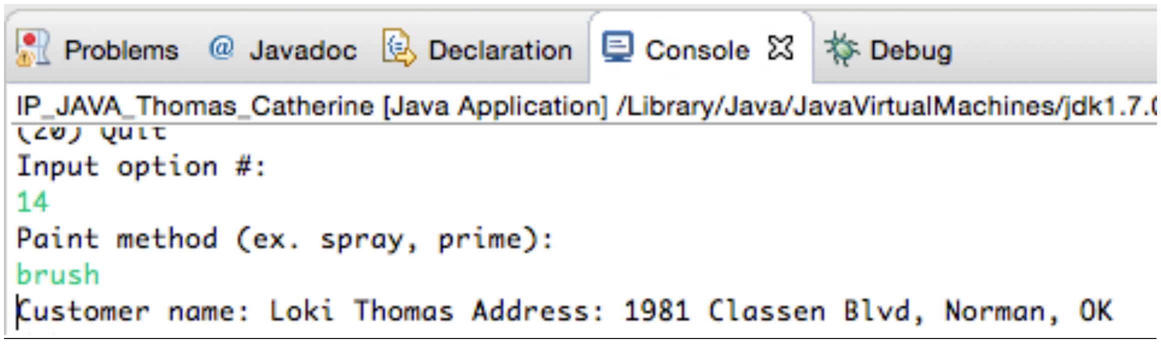
```
Input option #:
13
Input department no:
3
Input completion date mm-dd-yyyy:
11-11-2011
Job No.: 7  Date: 11-11-2011 Assembly ID: 4 Machine Type: saw
Machine Time: 4 Material: wood
```

<u>6.13.c</u>

```
Input option #:
13
Input department no:
1
Input completion date mm-dd-yyyy:
02-17-2015
none
```

6.14 Testing for Query 14

<u>6.14.a</u>



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0
(20) Quit
Input option #:
14
Paint method (ex. spray, prime):
brush
Customer name: Loki Thomas Address: 1981 Classen Blvd, Norman, OK
```

6.14.a Screenshot for Query 14

6.14.b

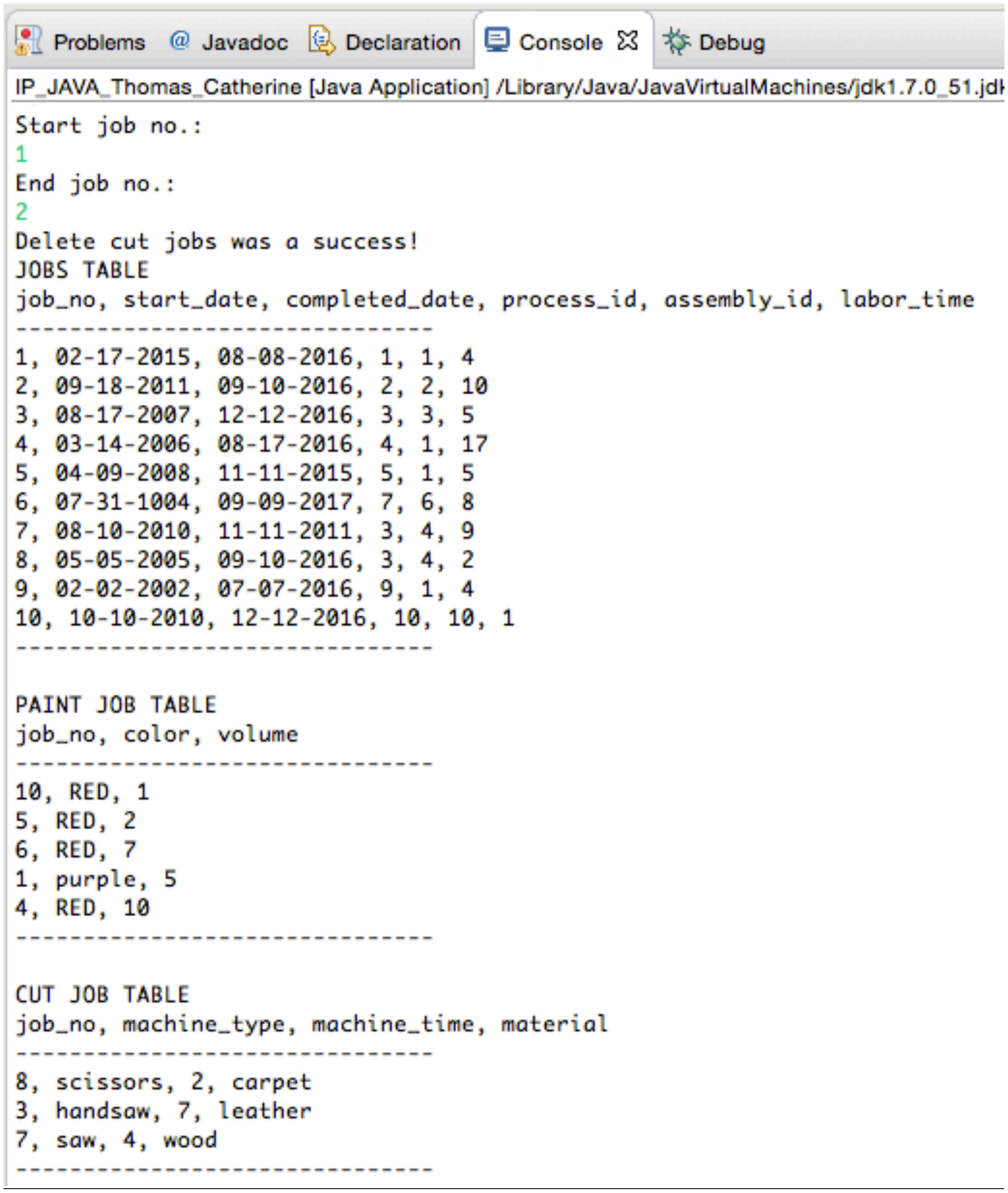
```
Input option #:
14
Paint method (ex. spray, prime):
Sponged
none
```

6.14.c

```
Input option #:
14
Paint method (ex. spray, prime):
spray
none
```

6.15 Testing for Query 15

6.15.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk
Start job no.:
1
End job no.:
2
Delete cut jobs was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id, labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
3, handsaw, 7, leather
7, saw, 4, wood
-------------------------------
```

6.15.a Screenshot for Query 15

6.15.b

```
Input option #:
15
Start job no.:
2
End job no.:
4
```

```
Delete cut jobs was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
```

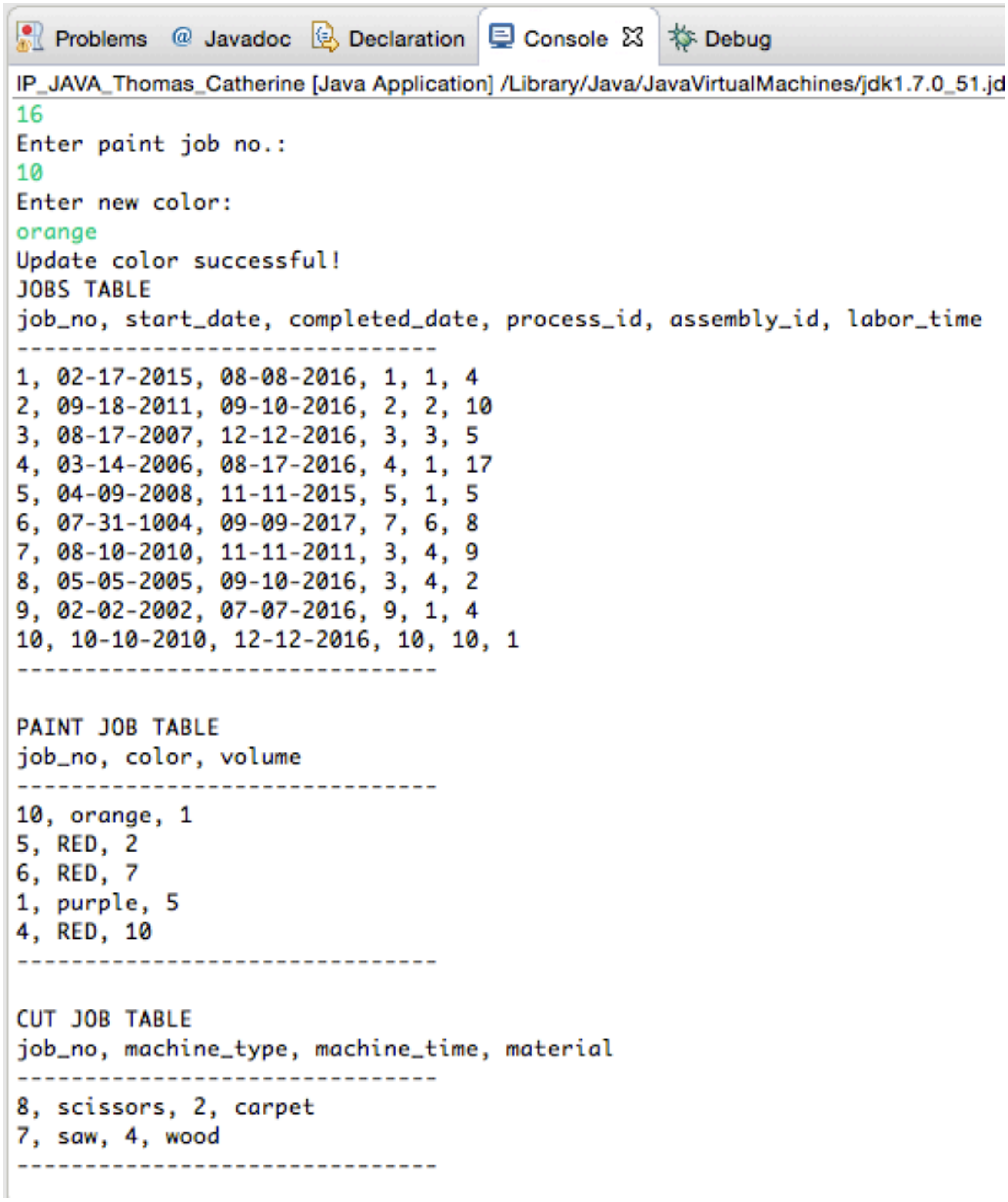
6.15.c

```
Input option #:
15
Start job no.:
1
End job no.:
3
Delete cut jobs was a success!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
```

```
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, RED, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
-------------------------------
```

6.16 Testing for Query 16

6.16.a


Problems
Javadoc
Declaration
Console
Debug
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jd

```
16
Enter paint job no.:
10
Enter new color:
orange
Update color successful!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id, labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, orange, 1
5, RED, 2
6, RED, 7
1, purple, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
-------------------------------
```

6.16.a Screenshot for Query 16

6.16.b

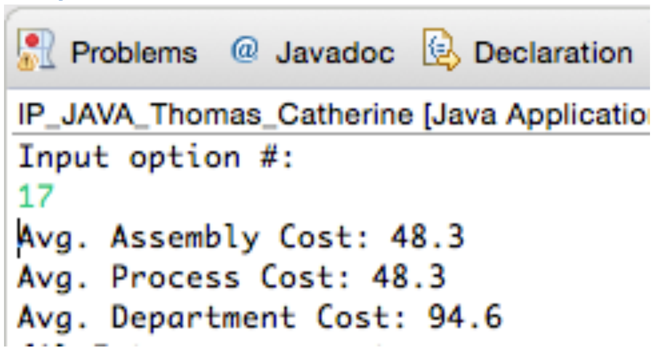
```
Input option #:
16
Enter paint job no.:
1
Enter new color:
RED
```

```
Update color successful!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, orange, 1
5, RED, 2
6, RED, 7
1, RED, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
-------------------------------
```

6.16.c

```
Input option #:
16
Enter paint job no.:
6
Enter new color:
purple
Update color successful!
JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, orange, 1
5, RED, 2
6, purple, 7
1, RED, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
```

6.17 Testing for Query 17

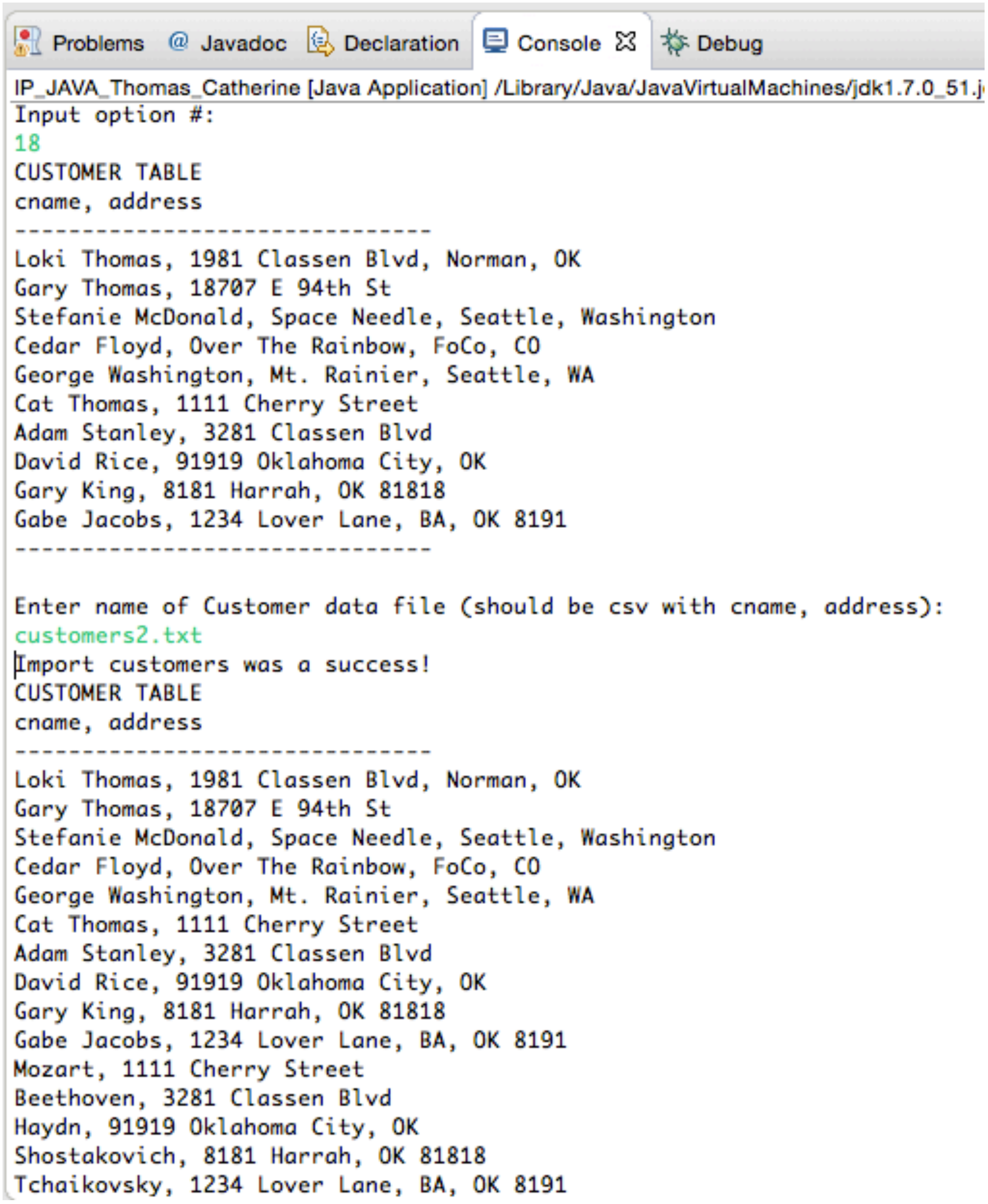


```
Input option #:
17
Avg. Assembly Cost: 48.3
Avg. Process Cost: 48.3
Avg. Department Cost: 94.6
```

6.17 Screenshot for Query 17

6.18 Testing of Import and Export

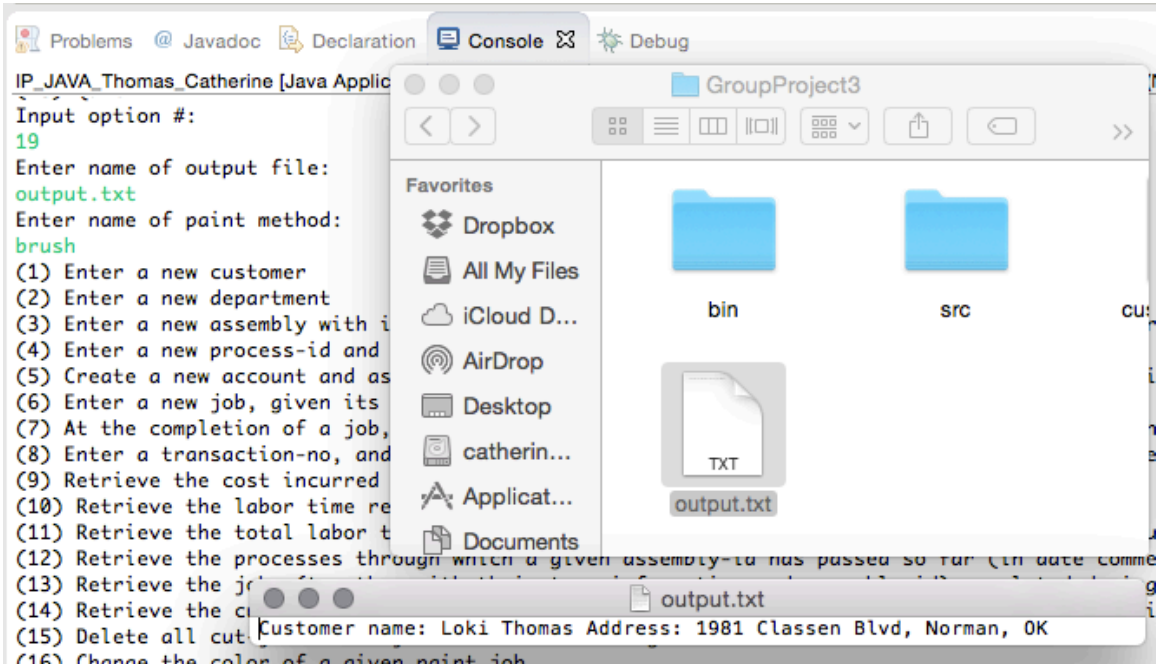
6.18.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.j
Input option #:
18
CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
Cedar Floyd, Over The Rainbow, FoCo, CO
George Washington, Mt. Rainier, Seattle, WA
Cat Thomas, 1111 Cherry Street
Adam Stanley, 3281 Classen Blvd
David Rice, 91919 Oklahoma City, OK
Gary King, 8181 Harrah, OK 81818
Gabe Jacobs, 1234 Lover Lane, BA, OK 8191
-------------------------------

Enter name of Customer data file (should be csv with cname, address):
customers2.txt
Import customers was a success!
CUSTOMER TABLE
cname, address
-------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
Cedar Floyd, Over The Rainbow, FoCo, CO
George Washington, Mt. Rainier, Seattle, WA
Cat Thomas, 1111 Cherry Street
Adam Stanley, 3281 Classen Blvd
David Rice, 91919 Oklahoma City, OK
Gary King, 8181 Harrah, OK 81818
Gabe Jacobs, 1234 Lover Lane, BA, OK 8191
Mozart, 1111 Cherry Street
Beethoven, 3281 Classen Blvd
Haydn, 91919 Oklahoma City, OK
Shostakovich, 8181 Harrah, OK 81818
Tchaikovsky, 1234 Lover Lane, BA, OK 8191
```

6.18.a Screenshot of Import Function

6.18.b



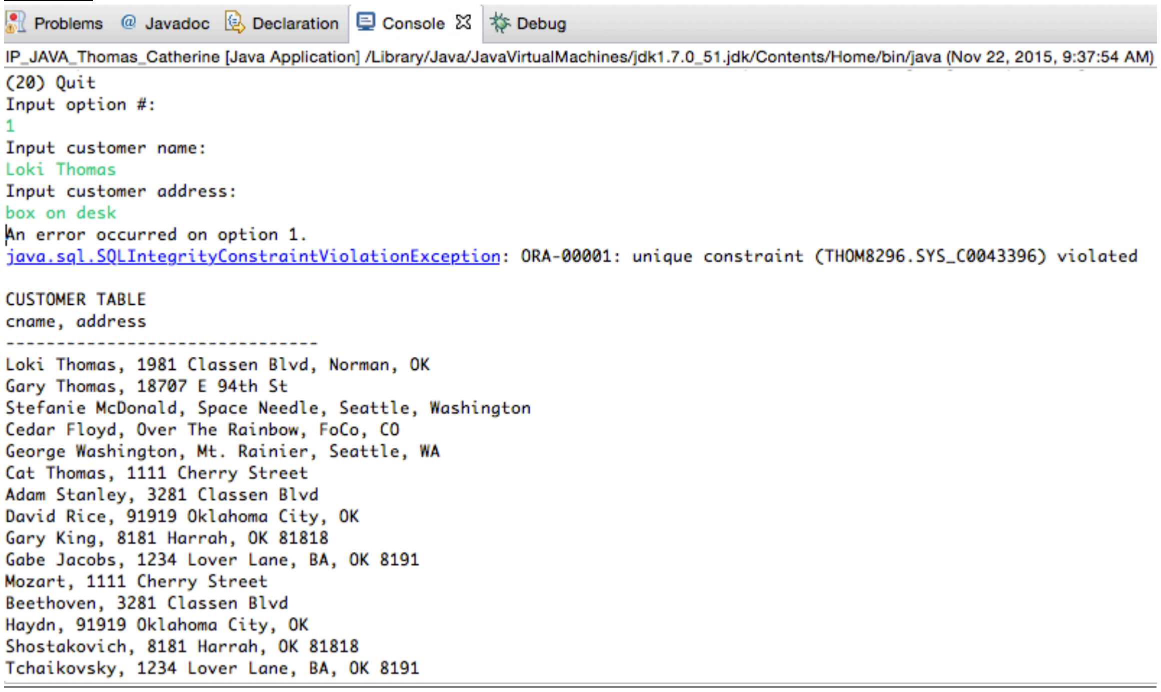
```
IP_JAVA_Thomas_Catherine [Java Applic
Input option #:
19
Enter name of output file:
output.txt
Enter name of paint method:
brush
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new assembly with i
(4) Enter a new process-id and
(5) Create a new account and as
(6) Enter a new job, given its
(7) At the completion of a job,
(8) Enter a transaction-no, and
(9) Retrieve the cost incurred
(10) Retrieve the labor time re
(11) Retrieve the total labor t
(12) Retrieve the processes through which a given assembly-id has passed so far (in date comme
(13) Retrieve the jo
(14) Retrieve the cu
(15) Delete all cut
(16) Change the color of a given paint job
```

```
output.txt
Customer name: Loki Thomas Address: 1981 Classen Blvd, Norman, OK
```

6.18.b Screenshot of Export Function

6.19 Testing Three SQL Errors

6.19.a



```
IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/bin/java (Nov 22, 2015, 9:37:54 AM)
(20) Quit
Input option #:
1
Input customer name:
Loki Thomas
Input customer address:
box on desk
An error occurred on option 1.
java.sql.SQLIntegrityConstraintViolationException: ORA-00001: unique constraint (THOM8296.SYS_C0043396) violated

CUSTOMER TABLE
cname, address
------------------------------
Loki Thomas, 1981 Classen Blvd, Norman, OK
Gary Thomas, 18707 E 94th St
Stefanie McDonald, Space Needle, Seattle, Washington
Cedar Floyd, Over The Rainbow, FoCo, CO
George Washington, Mt. Rainier, Seattle, WA
Cat Thomas, 1111 Cherry Street
Adam Stanley, 3281 Classen Blvd
David Rice, 91919 Oklahoma City, OK
Gary King, 8181 Harrah, OK 81818
Gabe Jacobs, 1234 Lover Lane, BA, OK 8191
Mozart, 1111 Cherry Street
Beethoven, 3281 Classen Blvd
Haydn, 91919 Oklahoma City, OK
Shostakovich, 8181 Harrah, OK 81818
Tchaikovsky, 1234 Lover Lane, BA, OK 8191
```

6.19.a ScreenShot of SQL Error

6.19.b

```
Input option #:
3
Input assembly id (number):
```

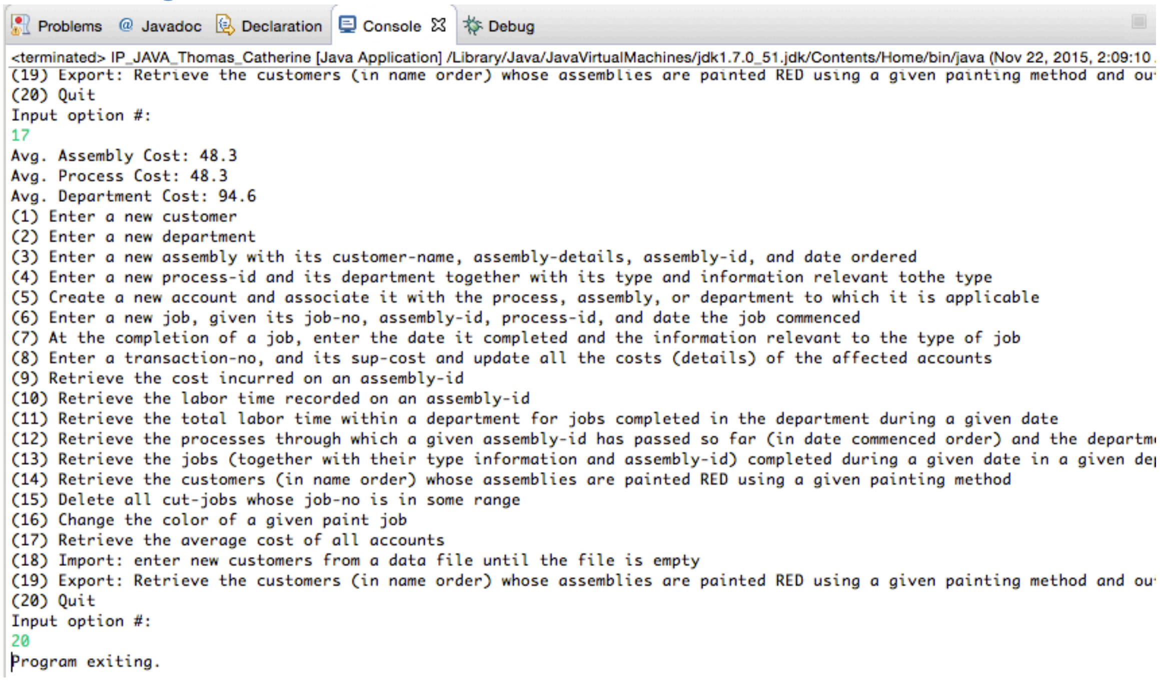
```
15
Input assembly details:
test
Input date ordered mm-dd-yyyy:
08-08-2010
Input customer name:
Bob The Builder
An error occurred on option 3.
java.sql.SQLIntegrityConstraintViolationException: ORA-02291:
integrity constraint (THOM8296.SYS_C0043649) violated - parent
key not found

ASSEMBLIES TABLE
assembly_id, assembly_details, date_ordered, cname
-------------------------------
1, valued customer, 02-17-2013, Loki Thomas
2, should be done quickly, 03-14-2014, Loki Thomas
3, complicated work, 05-05-2005, George Washington
4, vibrant coloring, 04-04-2004, Stefanie McDonald
5, cherry wood, 06-06-2015, George Washington
7, new customer, 01-03-2015, Gary Thomas
6, renovating van, 07-13-2011, Cedar Floyd
8, flexible with assembly, 08-08-2008, Loki Thomas
9, pet friendly, 12-25-2015, Gary Thomas
10, complicated, 07-07-2012, Stefanie McDonald
```

6.19.c

```
Input option #:
7
Input job number:
32
Input date completed mm-dd-yyyy:
08-08-2010
Input labor time:
15
Input job type (paint, cut, fit):
cut
Input color (paint job) or machine type (cut job) or 'none':
saw
Input volume (paint job) or machine time (cut job) or 0:
2
Input material (cut job) or 'none':
none
An error occurred on option 7.
```

```
java.sql.SQLIntegrityConstraintViolationException: ORA-02291:
integrity constraint (THOM8296.SYS_C0043666) violated - parent
key not found

JOBS TABLE
job_no, start_date, completed_date, process_id, assembly_id,
labor_time
-------------------------------
1, 02-17-2015, 08-08-2016, 1, 1, 4
2, 09-18-2011, 09-10-2016, 2, 2, 10
3, 08-17-2007, 12-12-2016, 3, 3, 5
4, 03-14-2006, 08-17-2016, 4, 1, 17
5, 04-09-2008, 11-11-2015, 5, 1, 5
6, 07-31-1004, 09-09-2017, 7, 6, 8
7, 08-10-2010, 11-11-2011, 3, 4, 9
8, 05-05-2005, 09-10-2016, 3, 4, 2
9, 02-02-2002, 07-07-2016, 9, 1, 4
10, 10-10-2010, 12-12-2016, 10, 10, 1
-------------------------------

PAINT JOB TABLE
job_no, color, volume
-------------------------------
10, orange, 1
5, RED, 2
6, purple, 7
1, RED, 5
4, RED, 10
-------------------------------

CUT JOB TABLE
job_no, machine_type, machine_time, material
-------------------------------
8, scissors, 2, carpet
7, saw, 4, wood
-------------------------------
```

6.20 Testing for Quit



```
<terminated> IP_JAVA_Thomas_Catherine [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_51.jdk/Contents/Home/bin/java (Nov 22, 2015, 2:09:10
(19) Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method and ou
(20) Quit
Input option #:
17
Avg. Assembly Cost: 48.3
Avg. Process Cost: 48.3
Avg. Department Cost: 94.6
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date ordered
(4) Enter a new process-id and its department together with its type and information relevant tothe type
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts
(9) Retrieve the cost incurred on an assembly-id
(10) Retrieve the labor time recorded on an assembly-id
(11) Retrieve the total labor time within a department for jobs completed in the department during a given date
(12) Retrieve the processes through which a given assembly-id has passed so far (in date commenced order) and the departm
(13) Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given de
(14) Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method
(15) Delete all cut-jobs whose job-no is in some range
(16) Change the color of a given paint job
(17) Retrieve the average cost of all accounts
(18) Import: enter new customers from a data file until the file is empty
(19) Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method and ou
(20) Quit
Input option #:
20
Program exiting.
```

6.20 Quitting program