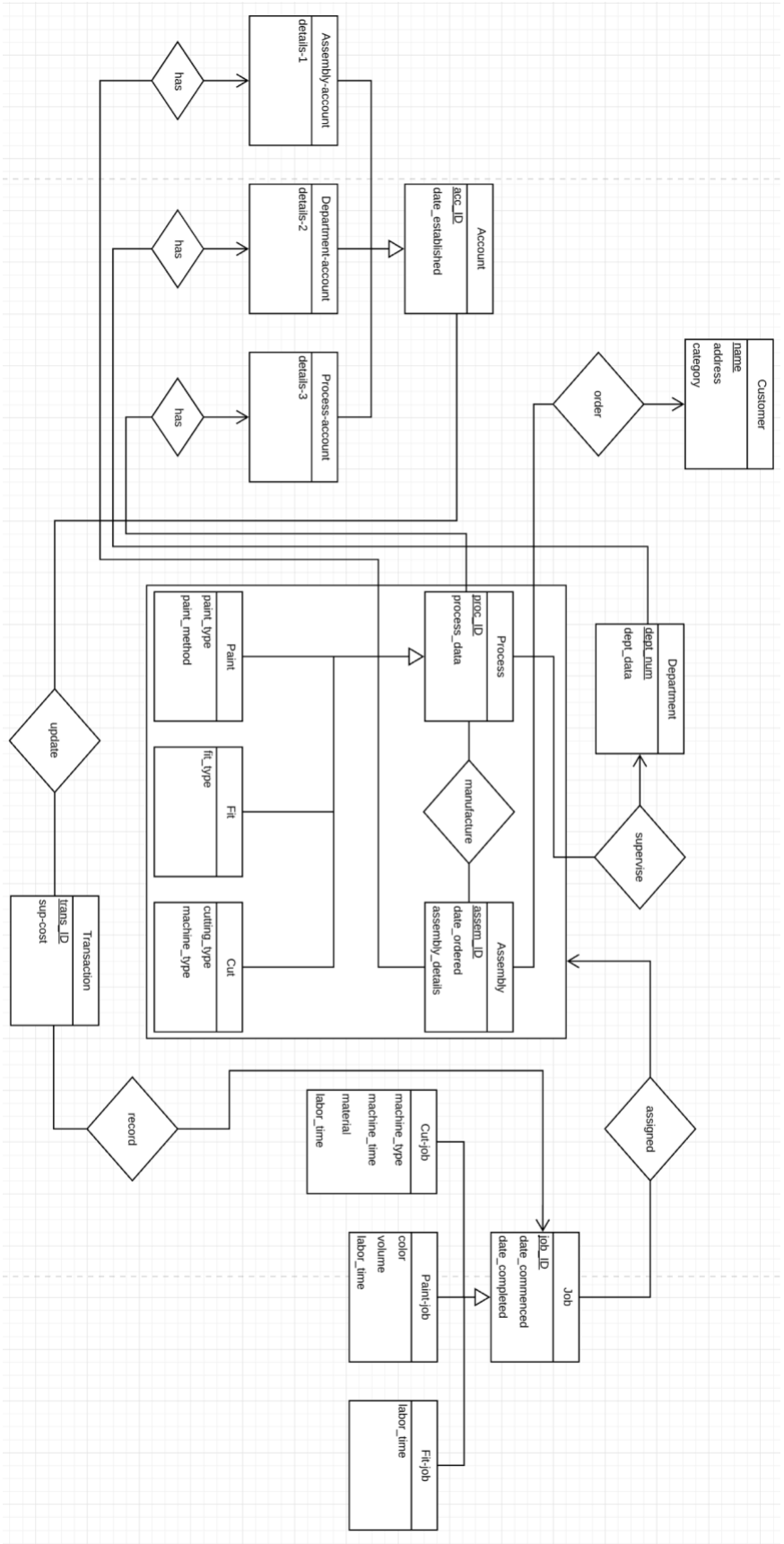
COURSE NAME:	Database Management Systems
COURSE NUMBER:	4513
SECTION NUMBER:	001
SEMESTER – YEAR:	Fall - 2019
INSTRUCTOR NAME:	Dr. Le Gruenwald
AUTHOR – ID – EMAIL:	Zaki Refai – 113380402 – zhrefai@ou.edu
TITLE:	Individual Project

Task 1.

1.1 ER Diagram:



1.2 Relational Database Schema

- Account(accout_ID, date_established)
- Assembly_Account(account_ID, details-1)
- Department_Account(account_ID, details-2)
- Process_Account(account_ID, details-3)
- Customer(customer_name, customer_address, customer_category)
- Assembly(assem_ID, date_ordered, assembly_details, customer_name, account_ID)
- Department(dept_ID, dept_data, account_ID)
- Process(proc_ID, process_data, dept_ID, account_ID)
- Paint(proc_ID, paint_type, paint_method)
- Fit(proc_ID, fit_type)
- Cut(proc_ID, cutting_type, machine_type)
- Job(job_ID, date_commenced, date_completed, assembly_ID, proc_ID)
- Cut_Job(job_ID, machine_type, machine_time, material, labor_time, assembly_ID, proc_ID)
- Paint_job(job_ID, color, volume, labor_time, assembly_ID, proc_ID)
- Fit_job(job_ID, labor_time, assembly_ID, proc_ID)
- Transaction(trans_ID, sup_cost, job_ID)
- Manufacture(proc_ID, assem_ID)
- Updates(acc_ID, trans_ID)

Task 2. Data Dictionary

Table dbo.Account

	Column	Data Type	Size	Nullable	Constraint
PK	account_ID	int	4		
	date_established	date	3		

Referenced by:

dbo.Assembly_Account (*account_ID*)
dbo.Department_Account (*account_ID*)
dbo.Process_Account (*account_ID*)
dbo.Assembly (*account_ID*)
dbo.Department (*account_ID*)
dbo.Process (*account_ID*)
dbo.Updates (*account_ID*)

Table dbo.Assembly

	Column	Data Type	Size	Nullable	Constraint
PK	assembly_ID	int	4		
	date_ordered	date	3		
	assembly_details	varchar(64)	66		
FK	customer_name	varchar(64)	66	Yes	
FK	account_ID	int	4	Yes	

Referenced by:

dbo.Job (*assembly_ID*)
dbo.Cut_Job (*assembly_ID*)
dbo.Paint_Job (*assembly_ID*)
dbo.Fit_Job (*assembly_ID*)
dbo.Manufacture (*assembly_ID*)

References:

dbo.Customer (*customer_name*)
dbo.Account (*account_ID*)

Table dbo.Assembly_Account

	Column	Data Type	Size	Nullable	Constraint
PK, FK	acc_ID	int	4		
	details-1	int	4		

References:

dbo.Account (*account_ID)*

Table dbo.Customer

	Column	Data Type	Size	Nullable	Constraint
PK	name	varchar(64)	66		
	address	varchar(64)	66		
	category	int	4		1 to 10

Referenced by:

dbo. Assembly (*customer_name*)

Table dbo.Cut

	Column	Data Type	Size	Nullable	Constraint
PK,FK	proc_ID	int	4		
	cutting_type	varchar(64)	66		
	machine_type	varchar(64)	66		

References:

dbo.Process (*proc_ID*)

Table dbo.Cut_Job

	Column	Data Type	Size	Nullable	Constraints
PK,FK	job_ID	int	4		
	machine_type	varchar(64)	66		
	machine_time	decimal	5		Decimal(6,3)
	material	varchar(64)	66		
	labor_time	decimal	5		Decimal(6,3)
FK	assembly_ID	int	4		
FK	proc_ID	int	4		

References:

dbo.Job (*job_ID*)
dbo.Assembly (*assembly_ID*)
dbo.Process (*proc_ID)*

Table dbo.Department

	Column	Data Type	Size	Nullable	Constraint
PK	dept_ID	int	4		
	dept_data	varchar(64)	66		
FK	account_ID	int	4	Yes	

Referenced by:

dbo.Process (*dept_ID*)

References:

dbo.Account (*account_ID*)

Table dbo.Department_Account

	Column	Data Type	Size	Nullable	Constraint
PK, FK	accout_ID	int	4		
	details-2	int	4		

Referenced by:

dbo.Account (*account_ID*)

Table dbo.Fit

	Column	Data Type	Size	Nullable	Constraint
PK, FK	proc_ID	int	4		
	fit_type	varchar(64)	66		

References:

dbo.Process (*proc_ID*)

Table dbo.Fit_Job

	Column	Data Type	Size	Nullable	Constraint
PK, FK	job_ID	int	4		
	labor_time	decimal	5		Decimal(6,3)
FK	assembly_ID	int	4		
FK	proc_ID	int	4		

References:

dbo.Job (job_ID)
dbo.Assembly (*assembly_ID*)
dbo.Process (*proc_ID*)

Table dbo.Job

	Column	Data Type	Size	Nullable	Constraint
PK	job_ID	int	4		
	date_commenced	date	3		
	date_completed	date	3	Yes	
FK	assembly_ID	int	4		
FK	proc_ID	int	4		

Referenced by:

dbo.Cut_Job (*job_ID*)
dbo.Paint_Job (*job_ID*)
dbo.Fit_Job (*job_ID*)
dbo.Transactions (*job_ID*)

Refernces:

dbo.Assembly (*assembly_ID*)
dbo.Process (*proc_ID*)

Table dbo.Manufacture

	Column	Data Type	Size	Nullable	Constraint
PK, FK	proc_ID	int	4		
PK, FK	assem_ID	int	4		

References:

dbo.Process (*proc_ID*)
dbo.Assembly (*assembly_ID*)

Table dbo.Paint

	Column	Data Type	Size	Nullable	Constraint
PK, FK	proc_ID	int	4		
	paint_type	varchar(64)	66		
	paint_method	varchar(64)	66		

References:

dbo.Process (*proc_ID*)

Table dbo.Paint_Job

	Column	Data Type	Size	Nullable	Constraint
PK, FK	proc_ID	int	4		
	color	varchar(64)	66		
	volume	decimal	5		Decimal(6,3)
	labor_time	decimal	5		Decimal(6,3)
FK	assembly_ID	int	4		
FK	proc_ID	int	4		

References:

dbo.Job (*job_ID*)
dbo.Assembly (*assembly_ID*)
dbo.Process (*proc_ID*)

Table dbo.Process

	Column	Data Type	Size	Nullable	Constraint
PK	proc_ID	int	4		
	process_data	varchar(64)	66		
FK	dept_ID	int	4	Yes	
FK	account_ID	int	4	Yes	

Referenced by:

dbo.Piant (*proc_ID*)
dbo.Fit (*proc_ID*)
dbo.Cut (*proc_ID*)
dbo.Job (*proc_ID*)
dbo.Cut_Job (*proc_ID*)
dbo.Paint_Job (*proc_ID*)
dbo.Fit_Job (*proc_ID*)

dbo.Manufacture (*proc_ID*)

References:

dbo.Department (*dept_ID*)
dbo.Account (*account_ID*)

Table dbo.Process-Account

	Column	Data Type	Size	Nullable	Constraint
PK, FK	accout_ID	int	4		
	details_3	int	4		

References:

dbo.Account (*account_ID*)

Table dbo.Transactions

	Column	Data Type	Size	Nullable	Constraint
PK	trans_ID	int	4		
	sup_cost	int	4		
FK	job_ID	int	4		

Referenced by:

dbo.Updates(*trans_ID*)

References:

dbo.Job (*job_ID*)

Table dbo.Updates

	Column	Data Type	Size	Nullable	Constraint
PK, FK	accout_ID	int	4		
PK, FK	trans_ID	int	4		

References:

dbo.Transaction (*trans_ID*)
dbo.Account (*account_ID*)

Task 3.

3.1 Discussion of storage structures for tables

Table Name	Query # and Type	Search Key	Query Frequency	Selected File Organization	Justification
Account	5: Insertion 8: Update 9: Retrieval	Account ID	10/day 50/day 200/day	Dynamic hashing	Queries are very frequent per day
Assembly_Account	8: Update 9: Retrieval	Account ID	50/day 200/day	Dynamic hashing	Queries are very frequent per day
Process_Account	8: Update	Account ID	50/day	Heap	Just inserting into table
Department_Account	8: Update	Account ID	50/day	Heap	Just inserting into table
Customer	1: Insertion 3: Update 13: Retrieval	Customer Name	30/day 40/day 100/day	Sequential File	Range search on category, so sequential file
Assembly	3: Insertion 6: Insertion 8: Update 9: Retrieval 11: Retrieval	Assembly ID	40/day 50/day 50/day 200/day 100/day	Dynamic hashing	Queries are very frequent
Department	2: Insertion 8:Update 10: Retrieval 12: Retrieval	Department ID	infrequemt 50/day 20/day 20/day	Dynamic hashing	Queries are infrequent, but random search
Process	4: Insertion 8: Update 11: Retrieval	Process ID	infrequent 50/day 100/day	Dynamic hashing	Queries are very frequent
Fit	4: Insertion	Process ID	infrequent	Sequential File	Insertion is infrequent
Cut	4: Insertion	Process ID	infrequent	Sequential File	Insertion is infrequent
Paint	4: Insertion	Process ID	infrequent	Sequential File	Insertion is infrequent
Job	6: Insertion 7: Update 12: Retrieval	Job ID	50/day 50/day 20/day	Dynamic hashing	Queires are frequent
Cut_Job	7: Insertion 10: Retrieval 14: Update	Job ID	50/day 20/day 1/month	Sequential File	Queries are infrequent
Fit_Job	7: Insertion 10: Retrieval	Job ID	50/day 20/day	Sequential File	Queries are infrequent
Paint_Job	7: Insertion 10: Retrieval	Job ID	50/day 20/day	Sequential File	Queries are infrequent

	15: Update		**1/week**		
Transactions	**8: Insertion**	**Transaction ID**	**50/day**	**Heap file**	**Just inserting**
Updates	**8: Insertion**	**Transaction ID Account ID**	**50/day**	**Heap file**	**Just inserting**
Manufacture	**6: Insertion**	**Process ID Assembly ID**	**50/day**	**Heap file**	**Just inserting**

3.2 Discussion of choices of storage structures for each relational table

We do not have super access to change the actual strucutres of the table in Azure SQL. SO we can only use the created default tables with clustered primary indexes.

We will not need to change the anything in the sql file

Task 4: SQL Statements

```
DROP TABLE Manufacture;
DROP TABLE Updates;
DROP TABLE Transactions;
DROP TABLE Cut_Job;
DROP TABLE Fit_Job;
DROP TABLE Paint_Job;
DROP TABLE Job;
DROP TABLE Assembly;
DROP TABLE Paint;
DROP TABLE Fit;
DROP TABLE Cut;
DROP TABLE Process;
DROP TABLE Department;
DROP TABLE Customer;
DROP TABLE Assembly_Account;
DROP TABLE Process_Account;
DROP TABLE Department_Account;
DROP TABLE Account;

CREATE TABLE Account (
   account_ID INT PRIMARY KEY NOT NULL,
   date_established DATE NOT NULL
);

CREATE TABLE Assembly_Account(
   account_ID INT PRIMARY KEY NOT NULL,
   details_1 INT NOT NULL,

   FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Department_Account(
   account_ID INT PRIMARY KEY NOT NULL,
   details_2 INT NOT NULL,
```

```
    FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Process_Account(
    account_ID INT PRIMARY KEY NOT NULL,
    details_3 INT NOT NULL,

    FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Customer (
    customer_name VARCHAR(64) PRIMARY KEY NOT NULL,
    customer_address VARCHAR(64) NOT NULL,
    customer_category INT NOT NULL CHECK (customer_category between 1 and 10)
);

CREATE TABLE Assembly (
    assembly_ID INT PRIMARY KEY NOT NULL,
    date_ordered DATE NOT NULL,
    assembly_details VARCHAR(64) NOT NULL,
    customer_name VARCHAR(64),
    account_ID INT,

    FOREIGN KEY(customer_name) REFERENCES Customer(customer_name),
    FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Department (
    dept_ID INT PRIMARY KEY NOT NULL,
    dept_data VARCHAR(64) NOT NULL,
    account_ID INT,

    FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Process (
    proc_ID INT PRIMARY KEY NOT NULL,
```

```
    process_data VARCHAR(64) NOT NULL,
    dept_ID INT,
    account_ID INT,

    FOREIGN KEY(dept_ID) REFERENCES Department(dept_ID),
    FOREIGN KEY(account_ID) REFERENCES Account(account_ID)
);

CREATE TABLE Paint (
    proc_ID INT PRIMARY KEY NOT NULL,
    paint_type VARCHAR(64) NOT NULL,
    paint_method VARCHAR(64) NOT NULL
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
);

CREATE TABLE Fit (
    proc_ID INT PRIMARY KEY NOT NULL,
    fit_type VARCHAR(64) NOT NULL,
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
);

CREATE TABLE Cut (
    proc_ID INT PRIMARY KEY NOT NULL,
    cutting_type VARCHAR(64) NOT NULL,
    machine_type VARCHAR(64) NOT NULL,
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
);

CREATE TABLE Job (
    job_ID INT PRIMARY KEY NOT NULL,
    date_commenced DATE NOT NULL,
    date_completed DATE,
    assembly_ID INT NOT NULL,
    proc_ID INT NOT NULL,

    FOREIGN KEY(assembly_ID) REFERENCES Assembly(assembly_ID),
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
```

```
);

CREATE TABLE Cut_Job (
    job_ID INT PRIMARY KEY NOT NULL,
    machine_type VARCHAR(64) NOT NULL,
    machine_time DECIMAL(6,3) NOT NULL,
    material VARCHAR(64) NOT NULL,
    labor_time DECIMAL(6,3) NOT NULL,
    assembly_ID INT NOT NULL,
    proc_ID INT NOT NULL,

    FOREIGN KEY(job_ID) REFERENCES Job(job_ID),
    FOREIGN KEY(assembly_ID) REFERENCES Assembly(assembly_ID),
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
);

CREATE TABLE Paint_Job (
    job_ID INT PRIMARY KEY NOT NULL,
    color VARCHAR(64) NOT NULL,
    volume DECIMAL(6,3) NOT NULL,
    labor_time DECIMAL(6,3)NOT NULL,
    assembly_ID INT NOT NULL,
    proc_ID INT NOT NULL,

    FOREIGN KEY(job_ID) REFERENCES Job(job_ID),
    FOREIGN KEY(assembly_ID) REFERENCES Assembly(assembly_ID),
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)
);

CREATE TABLE Fit_Job(
    job_ID INT PRIMARY KEY NOT NULL,
    labor_time DECIMAL(6,3) NOT NULL,
    assembly_ID INT NOT NULL,
    proc_ID INT NOT NULL,

    FOREIGN KEY(job_ID) REFERENCES Job(job_ID),
    FOREIGN KEY(assembly_ID) REFERENCES Assembly(assembly_ID),
```

```
    FOREIGN KEY(proc_ID) REFERENCES Process(proc_ID)

);

CREATE TABLE Transactions(
    trans_ID INT PRIMARY KEY NOT NULL,
    sup_cost INT NOT NULL,
    job_ID INT NOT NULL,

    FOREIGN KEY(job_ID) REFERENCES Job(job_ID)
);

CREATE TABLE Manufacture(
    proc_ID INT NOT NULL,
    assembly_ID INT NOT NULL,

    PRIMARY KEY (proc_ID, assembly_ID),
    FOREIGN KEY (proc_ID) REFERENCES Process(proc_ID),
    FOREIGN KEY (assembly_ID) REFERENCES Assembly(assembly_ID)
)

CREATE TABLE Updates(
    account_ID INT NOT NULL,
    trans_ID INT NOT NULL,

    PRIMARY KEY (account_ID, trans_ID),
    FOREIGN KEY (account_ID) REFERENCES Account(account_ID),
    FOREIGN KEY (trans_ID) REFERENCES Transactions(trans_ID)
);
```

Task 5: Java Source Code

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

public class Database {
	private static int ERROR_FLAG = 0;
	private static final String[] query_Options = { "(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 to the
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
by adding sup-cost to their current values of details",
			"(9) Retrieve the cost incurred on an assembly-id",
			"(10) Retrieve the total labor time within a department for jobs completed in the department during a
given date",
			"(11) Retrieve the processes through which a given assembly-id has passed so far (in date-
commenced order) and the department responsible for each process",
			"(12) Retrieve the jobs (together with their type information and assembly-id) completed during a
given date in a given department",
			"(13) Retrieve the customers (in name order) whose category is in a given range",
			"(14) Delete all cut-jobs whose job-no is in a given range", "(15) Change the color of a given paint
job",
```

```
                "(16) Import new customers from file", "(17) Export customers to file", "(18) Quit" };

        // Print the query options
        public static void print_queries() {
                for (int i = 0; i < query_Options.length; ++i) {
                        System.out.println(query_Options[i]);
                }
                System.out.println();
                System.out.print("Please select the number corresponding to the query: ");
        }

        // Print all customer information
        public static void print_table(Connection connection, int table) throws SQLException {

                String query = "";

                switch (table) {
                case 1: // Print Customer table
                        query = "SELECT * FROM Customer";
                        try (final Statement statement = connection.createStatement();
                                        final ResultSet resultSet = statement.executeQuery(query)) {
                                System.out.println();
                                System.out.println("Contents of Customer table:");
                                System.out.println("--Customer Name--|--Customer Address--|--Customer Category--");

                                // Printing out table
                                while (resultSet.next()) {
                                        System.out.println(String.format("----%s | %s | %s", resultSet.getString(1),
resultSet.getString(2),
                                                        resultSet.getString(3)));
                                }
                        }
                        break;

                case 2: // Print Department table
                        query = "SELECT * FROM Department";
                        try (final Statement statement = connection.createStatement();
                                        final ResultSet resultSet = statement.executeQuery(query)) {
                                System.out.println();
                                System.out.println("Contents of Department table:");
                                System.out.println("--Department ID--|--Department Data--|--Account ID--");
```

```
                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(String.format("----%s | %s | %s", resultSet.getString(1),
resultSet.getString(2),
                                resultSet.getString(3)));
                    }
                }
                break;

            case 3: // Print Process table
                query = "SELECT * FROM Process";
                try (final Statement statement = connection.createStatement();
                        final ResultSet resultSet = statement.executeQuery(query)) {
                    System.out.println();
                    System.out.println("Contents of Process table:");
                    System.out.println("--Process ID--|--Process Data--|--Department ID--|--Account ID--");
                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(String.format("----%s | %s | %s | %s", resultSet.getString(1),
                                resultSet.getString(2), resultSet.getString(3),
resultSet.getString(4)));
                    }
                }
                break;

            case 4: // Print Assembly table
                query = "SELECT * FROM Assembly";
                try (final Statement statement = connection.createStatement();
                        final ResultSet resultSet = statement.executeQuery(query)) {
                    System.out.println();
                    System.out.println("Contents of Assembly table:");
                    System.out.println(
                            "--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer
Name--|--Account ID--");

                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(
                                String.format("----%s | %s | %s | %s | %s",
resultSet.getString(1), resultSet.getString(2),
```

```
                                                                    resultSet.getString(3),
resultSet.getString(4), resultSet.getString(5)));
                                }
                        }
                        break;

                case 5: // Print the Job table
                        query = "SELECT * FROM Job";
                        try (final Statement statement = connection.createStatement();
                                        final ResultSet resultSet = statement.executeQuery(query)) {
                                System.out.println();
                                System.out.println("Contents of Job table:");
                                System.out.println("--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--
Process ID--|");

                                // Printing out table
                                while (resultSet.next()) {
                                        System.out.println(

                                                                        String.format("----%s | %s | %s | %s | %s",
resultSet.getString(1), resultSet.getString(2),
                                                                    resultSet.getString(3),
resultSet.getString(4), resultSet.getString(5)));
                                }
                        }
                        break;

                case 6: // Print the Account table
                        query = "SELECT * FROM Account";
                        try (final Statement statement = connection.createStatement();
                                        final ResultSet resultSet = statement.executeQuery(query)) {
                                System.out.println();
                                System.out.println("Contents of Account table:");
                                System.out.println("--Account ID--|--Date Established--");

                                // Printing out table
                                while (resultSet.next()) {
                                        System.out.println(String.format("----%s | %s ", resultSet.getString(1),
resultSet.getString(2)));
                                }
                        }
```

```
                    break;

                case 7: // Print the Department_Account table
                    query = "SELECT * FROM Department_Account";
                    try (final Statement statement = connection.createStatement();
                                    final ResultSet resultSet = statement.executeQuery(query)) {
                        System.out.println();
                        System.out.println("Contents of Department_Account table:");
                        System.out.println("--Account ID--|--Details 2--");

                        // Printing out table
                        while (resultSet.next()) {
                            System.out.println(String.format("----%s | %s ", resultSet.getString(1),
resultSet.getString(2)));
                        }
                    }
                    break;

                case 8: // Print the Assembly_Account table
                    query = "SELECT * FROM Assembly_Account";
                    try (final Statement statement = connection.createStatement();
                                    final ResultSet resultSet = statement.executeQuery(query)) {
                        System.out.println();
                        System.out.println("Contents of Assembly_Account table:");
                        System.out.println("--Account ID--|--Details 1--");

                        // Printing out table
                        while (resultSet.next()) {
                            System.out.println(String.format("----%s | %s ", resultSet.getString(1),
resultSet.getString(2)));
                        }
                    }
                    break;

                case 9: // Print the Process_Account table
                    query = "SELECT * FROM Process_Account";
                    try (final Statement statement = connection.createStatement();
                                    final ResultSet resultSet = statement.executeQuery(query)) {
                        System.out.println();
                        System.out.println("Contents of Process_Account table:");
                        System.out.println("--Account ID--|--Details 3--");
```

```
                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(String.format("----%s | %s ", resultSet.getString(1),
resultSet.getString(2)));
                    }
                }
                break;

            case 10: // Print jobs where date_completed in not null
                query = "SELECT * FROM Job WHERE date_completed IS NOT NULL";
                try (final Statement statement = connection.createStatement();
                        final ResultSet resultSet = statement.executeQuery(query)) {
                    System.out.println();
                    System.out.println("Contents of Job table where date_completed is not null:");
                    System.out.println("--Job ID--|--Date Completed--");

                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(

                                String.format("----%s | %s ", resultSet.getString(1),
resultSet.getString(3)));
                    }
                }
                break;

            case 11: // Print Cut Jobs
                query = "SELECT * FROM Cut_Job";
                try (final Statement statement = connection.createStatement();
                        final ResultSet resultSet = statement.executeQuery(query)) {
                    System.out.println();
                    System.out.println("Contents of Cut Job table:");
                    System.out.println("--Job IDs of Cut Jobs--");

                    // Printing out table
                    while (resultSet.next()) {
                        System.out.println(String.format("----%s ", resultSet.getString(1)));
                    }
                }
                break;
```

```
                default:
                        System.out.println("Print Table");
                        break;
                }
        }



// Check if primary key exists
        public static boolean check_key(Connection connection, String ID, String table, String where_condition,
                        int column_number) throws SQLException {
                boolean check = false;

                String query = "SELECT * FROM " + table + " " + where_condition;
                try (final Statement statement = connection.createStatement();
                                final ResultSet resultSet = statement.executeQuery(query)) {

                        while (resultSet.next()) {
                                if (ID.equalsIgnoreCase(resultSet.getString(column_number))) {
                                        check = true;
                                }
                        }
                }

                return check;
        }

        // Selectively print a column of a table based on a condition
        public static void selective_print(Connection connection, String table, String where_condition, int column_number,
                        String label) throws SQLException {

                String query = "SELECT * FROM " + table + " " + where_condition;
                try (final Statement statement = connection.createStatement();
                                final ResultSet resultSet = statement.executeQuery(query)) {

                        // Printing label
                        System.out.println("Existing " + table + " entries");

                        System.out.println(label);
                        while (resultSet.next()) {
```

```
                                        System.out.println(String.format("----%s ", resultSet.getString(column_number)));
                                }
                        }
                }

/*----------------------------------------QUERY FUNCTIONS----------------------------------------*/

                // COMPLETELY DONE: Function enters new customer: Query 1 and Query 3
                public static String enter_new_customer(Connection connection, Scanner scanner, int query_flag)
                                throws SQLException {
                        // Print existing customer names
                        System.out.println();
                        selective_print(connection, "Customer", "", 1, "--Customer Name");
                        // Asking for customer name
                        System.out.println();
                        System.out.print("New Customer Name: ");
                        String customer_name = scanner.nextLine();

                        // If customer name already exists
                        if (check_key(connection, customer_name, "Customer", "", 1)) {
                                System.out.println();
                                System.out.println("Customer Name " + customer_name + " already exists, please enter a new
name");

                                System.out.println();
                                print_queries();

                                ERROR_FLAG = 1;
                        } else {
                                // Asking for customer address
                                System.out.println();
                                System.out.print("Customer's address: ");
                                String customer_address = scanner.nextLine();

                                // Asking for customer category
                                System.out.println();
                                System.out.print("Customer's category (1-10): ");
                                int customer_category = scanner.nextInt();
                                scanner.nextLine();

                                // Adding information from user into sql statement
```

```
                        try (final Statement statement = connection.createStatement();) {
                                statement.executeUpdate(
                                                "INSERT INTO Customer(customer_name, customer_address,
customer_category)\n" + "VALUES ('"
                                                                + customer_name + "', '" + customer_address + "',
'" + customer_category + "')\n");

                                System.out.println();
                                System.out.println("Successfully added new customer\n");

                                // Print queries if flag is set
                                if (query_flag == 1) {
                                        print_queries();
                                }
                        } catch (SQLException e) {
                                System.out.println("\nERROR: " + e.getMessage() + "\n");
                                System.out.print("Please select a number corresponding to the query: ");
                                ERROR_FLAG = 1; // cycles to query 3 if customer input is incorrect
                        }

                }
                return customer_name;
        }

// COMPLETELY DONE: Function adds a new department: also part of Query 4 and 5
        public static int enter_new_department(Connection connection, Scanner scanner, int query_flag) throws SQLException {

                // Print existing department ID
                System.out.println();
                selective_print(connection, "Department", "", 1, "--Department ID");
                // Enter a department ID
                System.out.println();
                System.out.print("New Department ID: ");
                int department_ID = scanner.nextInt();
                scanner.nextLine();

                // If department name already exists
                if (check_key(connection, Integer.toString(department_ID), "Department", "", 1)) {
                        System.out.println();
                        System.out.println("Department ID " + department_ID + " already exists, please enter a new ID");
```

```
                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                } else {
                        // Enter department data
                        System.out.println();
                        System.out.print("Department Data: ");
                        String department_data = scanner.nextLine();

                        // Adding new department into database
                        try (final Statement statement = connection.createStatement();) {
                                statement.executeUpdate("INSERT INTO Department(dept_ID, dept_data)\n" + "VALUES
("'" + department_ID
                                                + "', '" + department_data + "')\n");

                                System.out.println();
                                System.out.println("Successfully added new department\n");

                                // Print queries if flag is set
                                if (query_flag == 1) {
                                        print_queries();
                                }

                        } catch (SQLException e) {
                                System.out.println("\nERROR: " + e.getMessage() + "\n");
                                System.out.print("Please select a number corresponding to the query: ");
                                ERROR_FLAG = 1; // cycles to query 4
                        }
                }

                return department_ID;
        }

// COMPLETELY DONE: Function adds a new assembly: Part of Query 3
        public static int enter_new_assembly(Connection connection, Scanner scanner, String customer_name, int query_flag)
                        throws SQLException {

                // Print existing Assembly ID
                System.out.println();
                selective_print(connection, "Assembly", "", 1, "--Assembly ID");
```

```
                // Asking for assembly_ID
                System.out.println();
                System.out.print("New Assembly ID: ");
                int assembly_ID = scanner.nextInt();
                scanner.nextLine();

                // If department name already exists
                if (check_key(connection, Integer.toString(assembly_ID), "Assembly", "", 1)) {
                        System.out.println();
                        System.out.println("Department ID " + assembly_ID + " already exists, please enter a new ID");

                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                } else {
                        // Asking for date_ordered
                        System.out.println();
                        System.out.print("Assembly Ordered (YYYY-MM-DD): ");
                        String assembly_date_ordered = scanner.nextLine();

                        // Asking for assembly_details
                        System.out.println();
                        System.out.print("Assembly Details: ");
                        String assembly_details = scanner.nextLine();

                        // Adding new assembly to database
                        try (final Statement statement = connection.createStatement();) {
                                statement.executeUpdate(
                                                "INSERT INTO Assembly(assembly_ID, date_ordered,
assembly_details, customer_name)\n"
                                                                + "VALUES ('" + assembly_ID + "', '" +
assembly_date_ordered + "', '" + assembly_details
                                                                + "', '" + customer_name + "')");
                                System.out.println();
                                System.out.println("Successfully added new assembly\n");

                                // Print queries if flag is set
                                if (query_flag == 1) {
                                        print_queries();
                                }
```

```
                } catch (SQLException e) {
                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                        System.out.print("Please select a number corresponding to the query: ");
                        ERROR_FLAG = 1;
                }
        }

        return assembly_ID;
    }

// COMPLETELY DONE: Function enters a new assembly: Part of Query 5
    public static int query_3(Connection connection, Scanner scanner) throws SQLException {
        int assembly_ID = 0;
        String customer_name = "";

        // Query for new or current customer for new assembly
        System.out.println();
        System.out.print("Is customer for this assembly new or currently in database (new/current): ");
        String customer_query = scanner.nextLine();

        // If customer is new, enter new customer, if current, ask for name
        if (customer_query.equalsIgnoreCase("new") || customer_query.equalsIgnoreCase("new ")) {
            // Entering new customer and getting customer_name
            customer_name = enter_new_customer(connection, scanner, 0);
        } else if (customer_query.equalsIgnoreCase("current") || customer_query.equalsIgnoreCase("customer ")) {

            // Print existing customer names
            System.out.println();
            selective_print(connection, "Customer", "", 1, "--Customer Name");

            // Asking for customer name
            System.out.println();
            System.out.print("Existing Customer's Name: ");
            customer_name = scanner.nextLine();

            // If customer name already exists
            if (check_key(connection, customer_name, "Customer", "", 1)) {

            } else {
                System.out.println();
```

```
                                        System.out.println("ERROR: Customer name does not exists, please pick an existing
customer");

                                        System.out.println();
                                        print_queries();

                                        ERROR_FLAG = 1;
                                }

                        } else {
                                System.out.println();
                                System.out.println("Please enter 'new' for new customer or 'current' for current customer");

                                System.out.println();
                                print_queries();
                                ERROR_FLAG = 1;
                        }

                        if (ERROR_FLAG == 0) {
                                // Enter new assembly with customer name
                                assembly_ID = enter_new_assembly(connection, scanner, customer_name, 1);
                        }

                        return assembly_ID;
                }

// COMPLETELY DONE: Function adds new process: Part of Query 4
                public static int enter_new_process(Connection connection, Scanner scanner, int department_ID, int query_flag)
                                throws SQLException {

                        // Print existing Assembly ID
                        System.out.println();
                        selective_print(connection, "Process", "", 1, "--Process ID");
                        // Asking for process ID
                        System.out.println();
                        System.out.print("Process ID: ");
                        int process_ID = scanner.nextInt();
                        scanner.nextLine();

                        // If department name already exists
                        if (check_key(connection, Integer.toString(process_ID), "Process", "", 1)) {
```

```
                        System.out.println();
                        System.out.println("Process ID " + process_ID + " already exists, please enter a new ID");

                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                } else {
                        // Asking for process data
                        System.out.println();
                        System.out.print("Process Data: ");
                        String process_data = scanner.nextLine();

                        // Adding new assembly to database
                        try (final Statement statement = connection.createStatement();) {
                                statement.executeUpdate("INSERT INTO Process(proc_ID, process_data, dept_ID)\n" +
"VALUES ('"
                                                + process_ID + "', '" + process_data + "', '" + department_ID + "')");
                                System.out.println();
                                System.out.println("Successfully added new process\n");

                                // Print queries if flag is set
                                if (query_flag == 1) {
                                        print_queries();
                                }
                        } catch (SQLException e) {
                                System.out.println("\nERROR: " + e.getMessage() + "\n");
                                System.out.print("Please select a number corresponding to the query: ");
                                ERROR_FLAG = 1;
                        }
                }
                return process_ID;
        }

// COMPLETELY DONE: Function enters a new process: Part of Query 5
        public static int query_4(Connection connection, Scanner scanner) throws SQLException {
                int process_ID = 0;
                int department_ID = 0;
                String department_query = "";
                // Query for new or current department
                System.out.println();
```

```
                System.out.print("Is department for this process new or currently in database (new/current): ");
                department_query = scanner.nextLine();

                // If department is new, enter new department, if current, ask for dept_ID
                if (department_query.equalsIgnoreCase("new") || department_query.equalsIgnoreCase("new ")) {
                        department_ID = enter_new_department(connection, scanner, 0);
                } else if (department_query.equalsIgnoreCase("current") || department_query.equalsIgnoreCase("current ")) {
                        // Print existing department ID
                        System.out.println();
                        selective_print(connection, "Department", "", 1, "--Department ID");

                        // Asking for department
                        System.out.println();
                        System.out.print("Existing Department ID: ");
                        department_ID = scanner.nextInt();
                        scanner.nextLine();

                        // If department ID already exists
                        if (check_key(connection, Integer.toString(department_ID), "Department", "", 1)) {
                                // Do nothing
                        } else {
                                System.out.println();
                                System.out.println("ERROR: Department ID does not exists, please pick an existing
department ID");

                                System.out.println();
                                print_queries();

                                ERROR_FLAG = 1;
                        }

                } else {
                        System.out.println();
                        System.out.println("Please enter 'new' for new department or 'current' for current department");

                        System.out.println();
                        print_queries();
                        ERROR_FLAG = 1;
                }

                if (ERROR_FLAG == 0) {
```

```
                    // Entering new process with department ID
                    process_ID = enter_new_process(connection, scanner, department_ID, 1);
                }
                return process_ID;
            }

// COMPLETELY DONE: Function adds a new account: Query 5
            public static int enter_new_account(Connection connection, Scanner scanner, String type, int type_ID,
                    int query_flag) throws SQLException {
                int account_ID = 0;

                // Print existing Account ID
                System.out.println();
                selective_print(connection, "Account", "", 1, "--Account ID");
                // Asking for account ID
                System.out.println();
                System.out.print("New Account ID: ");
                account_ID = scanner.nextInt();
                scanner.nextLine();

                // If account ID already exists
                if (check_key(connection, Integer.toString(account_ID), "Account", "", 1)) {
                    System.out.println();
                    System.out.println("Account ID " + account_ID + " already exists, please enter a new ID");

                    System.out.println();
                    print_queries();

                    ERROR_FLAG = 1;
                } else {
                    // Asking for account date established
                    System.out.println();
                    System.out.print("Account Date Established (YYYY-MM-DD): ");
                    String date_established = scanner.nextLine();

                    if (type.equalsIgnoreCase("Department") || type.equalsIgnoreCase("Department ")) {
                        // Asking for details 2
                        System.out.println();
                        System.out.print("Details-2: ");
                        String details_2 = scanner.nextLine();

                        // Adding new account to database
```

```
                    try (final Statement statement = connection.createStatement();) {
                        statement.executeUpdate("INSERT INTO Account(account_ID,
date_established)\n" + "VALUES ('"
                                + account_ID + "', '" + date_established + "')");
                        System.out.println();
                        System.out.println("Successfully added new account\n");

                        statement.executeUpdate("INSERT INTO Department_Account(account_ID,
details_2)\n" + "VALUES ('"
                                + account_ID + "', '" + details_2 + "')");
                        System.out.println("Successfully added new department account\n");

                        statement.executeUpdate("UPDATE Department\n"
                                + "SET account_ID = (SELECT account_ID FROM
Department_Account WHERE account_ID = '"
                                + account_ID + "')\n" + "WHERE dept_ID = " + type_ID);

                        System.out.println("Successfully updated department with new account\n");

                        // Print queries if flag is set
                        if (query_flag == 1) {
                            print_queries();
                        }
                    } catch (SQLException e) {
                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                        System.out.print("Please select a number corresponding to the query: ");
                    }
                } else if (type.equalsIgnoreCase("Assembly") || type.equalsIgnoreCase("Assembly ")) {
                    // Asking for details 1
                    System.out.println();
                    System.out.print("Details-1: ");
                    String details_1 = scanner.nextLine();

                    // Adding new account to database
                    try (final Statement statement = connection.createStatement();) {
                        statement.executeUpdate("INSERT INTO Account(account_ID,
date_established)\n" + "VALUES ('"
                                + account_ID + "', '" + date_established + "')");
                        System.out.println();
                        System.out.println("Successfully added new account\n");
```

```
                                        statement.executeUpdate("INSERT INTO Assembly_Account(account_ID,
details_1)\n" + "VALUES ('"
                                                        + account_ID + "', '" + details_1 + "')");
                                        System.out.println("Successfully added new assembly account\n");

                                        statement.executeUpdate("UPDATE Assembly\n"
                                                        + "SET account_ID = (SELECT account_ID FROM
Assembly_Account WHERE account_ID = "
                                                        + account_ID + ")\n" + "WHERE assembly_ID = " + type_ID);

                                        System.out.println("Successfully updated assembly with new account\n");

                                        // Print queries if flag is set
                                        if (query_flag == 1) {
                                                print_queries();
                                        }
                                } catch (SQLException e) {
                                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                                        System.out.print("Please select a number corresponding to the query: ");
                                }
                        } else if (type.equalsIgnoreCase("Process") || type.equalsIgnoreCase("Process ")) {
                                // Asking for details 3
                                System.out.println();
                                System.out.print("Details-3: ");
                                String details_3 = scanner.nextLine();

                                // Adding new account to database
                                try (final Statement statement = connection.createStatement();) {
                                        statement.executeUpdate("INSERT INTO Account(account_ID,
date_established)\n" + "VALUES ('"
                                                        + account_ID + "', '" + date_established + "')");
                                        System.out.println();
                                        System.out.println("Successfully added new account\n");

                                        statement.executeUpdate("INSERT INTO Process_Account(account_ID,
details_3)\n" + "VALUES ('"
                                                        + account_ID + "', '" + details_3 + "')");
                                        System.out.println("Successfully added new process account\n");

                                        statement.executeUpdate("UPDATE Process\n"
```

```
                                                + "SET account_ID = (SELECT account_ID FROM
Process_Account WHERE account_ID = "
                                                + account_ID + ")\n" + "WHERE proc_ID = " + type_ID);

                                        System.out.println("Successfully updated process with new account\n");

                                        // Print queries if flag is set
                                        if (query_flag == 1) {
                                                print_queries();
                                        }
                                } catch (SQLException e) {
                                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                                        System.out.print("Please select a number corresponding to the query: ");
                                }
                        }
                }

                return account_ID;
        }

// COMPLETELY DONE: Function enters a new account
        public static int query_5(Connection connection, Scanner scanner) throws SQLException {
                int account_ID = 0;
                // Query for account type
                System.out.println();
                System.out.print("What type of account would you like (Department, Assembly, Process): ");
                String account_type = scanner.nextLine();

                if (account_type.equalsIgnoreCase("Department") || account_type.equalsIgnoreCase("Department ")) {
                        String department_query = "";
                        int department_ID = 0;
                        // Query for new or current department
                        System.out.println();
                        System.out.print("Is department for this account new or currently in database (new/current): ");
                        department_query = scanner.nextLine();

                        // If department is new, enter new department, if current, ask for dept_ID
                        if (department_query.equalsIgnoreCase("new") || department_query.equalsIgnoreCase("new ")) {
                                department_ID = enter_new_department(connection, scanner, 0);
                        } else if (department_query.equalsIgnoreCase("current") ||
department_query.equalsIgnoreCase("current ")) {
```

```
                                        // Print existing department ID
                                        System.out.println();
                                        selective_print(connection, "Department", "", 1, "--Department ID");

                                        // Asking for department id
                                        System.out.println();
                                        System.out.print("Existing Department ID: ");
                                        department_ID = scanner.nextInt();
                                        scanner.nextLine();

                                        // If department ID already exists
                                        if (check_key(connection, Integer.toString(department_ID), "Department", "", 1)) {
                                                // Do nothing
                                        } else {
                                                System.out.println();
                                                System.out.println("ERROR: Department ID does not exists, please pick an
existing department ID");

                                                System.out.println();
                                                print_queries();

                                                ERROR_FLAG = 1;
                                        }

                                } else {
                                        System.out.println();
                                        System.out.println("Please enter 'new' for new department or 'current' for current
department");

                                        System.out.println();
                                        print_queries();
                                        ERROR_FLAG = 1;
                                }

                                if (ERROR_FLAG == 0) {
                                        // Entering new process with department ID
                                        account_ID = enter_new_account(connection, scanner, account_type, department_ID, 1);
                                }

                        } else if (account_type.equalsIgnoreCase("Assembly") || account_type.equalsIgnoreCase("Assembly ")) {
                                String assembly_query = "";
```

```
                int assembly_ID = 0;
                // Query for new or current assembly
                System.out.println();
                System.out.print("Is assembly for this account new or currently in database (new/current): ");
                assembly_query = scanner.nextLine();

                // If assembly is new, enter new assembly, if current, ask for dept_ID
                if (assembly_query.equalsIgnoreCase("new") || assembly_query.equalsIgnoreCase("new ")) {
                        // Entering in new assembly so go to query 3
                        assembly_ID = query_3(connection, scanner);
                } else if (assembly_query.equalsIgnoreCase("current") || assembly_query.equalsIgnoreCase("current
")) {
                        // Print existing Assembly ID
                        System.out.println();
                        selective_print(connection, "Assembly", "", 1, "--Assembly ID");

                        // Asking for Assembly id
                        System.out.println();
                        System.out.print("Existing Assembly ID: ");
                        assembly_ID = scanner.nextInt();
                        scanner.nextLine();

                        // If assembly ID already exists
                        if (check_key(connection, Integer.toString(assembly_ID), "Assembly", "", 1)) {
                                // Do nothing
                        } else {
                                System.out.println();
                                System.out.println("ERROR: Assembly ID does not exists, please pick an
existing assembly ID");

                                System.out.println();
                                print_queries();

                                ERROR_FLAG = 1;
                        }
                } else {
                        System.out.println();
                        System.out.println("Please enter 'new' for new assembly or 'current' for current assembly");

                        System.out.println();
                        print_queries();
```

```
                    ERROR_FLAG = 1;
                }

                if (ERROR_FLAG == 0) {
                    // Entering new process with department ID
                    account_ID = enter_new_account(connection, scanner, account_type, assembly_ID, 1);
                }

            } else if (account_type.equalsIgnoreCase("Process") || account_type.equalsIgnoreCase("Process ")) {
                String process_query = "";
                int process_ID = 0;
                // Query for new or current process
                System.out.println();
                System.out.print("Is process for this account new or currently in database (new/current): ");
                process_query = scanner.nextLine();

                // If assembly is new, enter new assembly, if current, ask for dept_ID
                if (process_query.equalsIgnoreCase("new") || process_query.equalsIgnoreCase("new ")) {
                    // Entering in new process so go to query 4
                    process_ID = query_4(connection, scanner);
                } else if (process_query.equalsIgnoreCase("current") || process_query.equalsIgnoreCase("current ")) {
                    // Print existing Process ID
                    System.out.println();
                    selective_print(connection, "Process", "", 1, "--Process ID");

                    // Asking for Assembly id
                    System.out.println();
                    System.out.print("Existing Process ID: ");
                    process_ID = scanner.nextInt();
                    scanner.nextLine();

                    // If prcoess ID already exists
                    if (check_key(connection, Integer.toString(process_ID), "Process", "", 1)) {
                        // Do nothing
                    } else {
                        System.out.println();
                        System.out.println("ERROR: Process ID does not exists, please pick an existing
process ID");

                        System.out.println();
                        print_queries();
```

```
                                        ERROR_FLAG = 1;
                                }
                        } else {
                                System.out.println();
                                System.out.println("Please enter 'new' for new process or 'current' for current process");

                                System.out.println();
                                print_queries();
                                ERROR_FLAG = 1;
                        }

                        if (ERROR_FLAG == 0) {
                                // Entering new process with department ID
                                account_ID = enter_new_account(connection, scanner, account_type, process_ID, 1);
                        }
                } else {
                        System.out.println();
                        System.out.println(
                                        "Please enter account type: 'Department' for department, 'Assembly' for
Assmebly, or 'Process' for Process");

                        System.out.println();
                        print_queries();
                }

                return account_ID;
        }

// COMPLETELY DONE: Function adds a new job: Query 6
        public static int enter_new_job(Connection connection, Scanner scanner, int query_flag) throws SQLException {
                int job_ID = 0;
                // Print existing job ID
                System.out.println();
                selective_print(connection, "Job", "", 1, "--Job ID");
                // Query a job_ID
                System.out.println();
                System.out.print("New Job ID: ");
                job_ID = scanner.nextInt();
                scanner.nextLine();
```

```
                // If job ID already exists
                if (check_key(connection, Integer.toString(job_ID), "Job", "", 1)) {
                        System.out.println();
                        System.out.println("Job ID " + job_ID + " already exists, please enter a new ID");

                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                } else {
                        // Query date_commenced
                        System.out.println();
                        System.out.print("Date Commenced (YYYY-MM-DD): ");
                        String date_commenced = scanner.nextLine();

                        // Printing assembly table
                        print_table(connection, 4);
                        // Query assembly_ID
                        System.out.println();
                        System.out.print("Assembly ID: ");
                        int assembly_ID = scanner.nextInt();
                        scanner.nextLine();

                        // Printing process table
                        print_table(connection, 3);
                        // Query process_ID
                        System.out.println();
                        System.out.print("Process ID: ");
                        int process_ID = scanner.nextInt();
                        scanner.nextLine();

                        // Adding new job to database
                        try (final Statement statement = connection.createStatement();) {
                                statement.executeUpdate("INSERT INTO Job(job_ID, date_commenced, assembly_ID,
proc_ID)\n" + "VALUES ('"
                                                + job_ID + "', '" + date_commenced + "', '" + assembly_ID + "', '" +
process_ID + "')");
                                System.out.println();
                                System.out.println("Successfully added new job\n");

                                // Print queries if flag is set
```

```
                    if (query_flag == 1) {
                        print_queries();
                    }
                } catch (SQLException e) {
                    System.out.println("\nERROR: " + e.getMessage() + "\n");
                    System.out.print("Please select a number corresponding to the query: ");
                    ERROR_FLAG = 1;
                }
            }

            return job_ID;
}

// COMPLETELY DONE: Function adds new job type and updates job: Query 7
public static int enter_new_job_type(Connection connection, Scanner scanner, int query_flag) throws SQLException {
            int job_ID = 0;
            int job_type_ID = 0;

            // Printing job table
            print_table(connection, 5);
            // Query a job_ID
            System.out.println();
            System.out.print("Existing Job ID of completed job: ");
            job_ID = scanner.nextInt();
            scanner.nextLine();

            // If job ID already exists
            if (check_key(connection, Integer.toString(job_ID), "Job", "", 1)) {
                // Query a date_completed of job_ID
                System.out.println();
                System.out.print("Date Completed of Job (YYYY-MM-DD): ");
                String date_completed = scanner.nextLine();

                String assembly_ID = "";
                String process_ID = "";

                // Get assembly_ID and process_ID from job_ID
                String query = "SELECT * FROM Job WHERE job_ID = " + job_ID;
                try (final Statement statement = connection.createStatement();
                        final ResultSet resultSet = statement.executeQuery(query)) {
```

```
                resultSet.next();
                assembly_ID = resultSet.getString(4);
                process_ID = resultSet.getString(5);
        }

        // Query type of Job
        System.out.println();
        System.out.print("What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): ");
        String job_type = scanner.nextLine();

        // Query cut job information
        if (job_type.equalsIgnoreCase("Cut") || job_type.equalsIgnoreCase("Cut ")) {
                // Query machine_type
                System.out.println();
                System.out.print("Machine Type: ");
                String machine_type = scanner.nextLine();

                // Query machine_time
                System.out.println();
                System.out.print("Machine Time: ");
                Double machine_time = scanner.nextDouble();
                scanner.nextLine();

                // Query material
                System.out.println();
                System.out.print("Material: ");
                String material = scanner.nextLine();

                // Query labor_time
                System.out.println();
                System.out.print("Labor Time: ");
                Double labor_time = scanner.nextDouble();
                scanner.nextLine();

                // Adding new cut job to database
                try (final Statement statement = connection.createStatement();) {
                        statement.executeUpdate("UPDATE Job\n" + "SET date_completed = '" +
date_completed + "'\n"
                                        + "WHERE job_ID = " + job_ID);
                        System.out.println();
                        System.out.println("Successfully updated job with new information\n");
```

```
                        statement.executeUpdate(
                                "INSERT INTO Cut_Job(job_ID, machine_type,
machine_time, material, labor_time, assembly_ID, proc_ID)\n"
                                + "VALUES ('" + job_ID + "', '" +
machine_type + "', '" + machine_time + "', '"
                                + material + "', '" + labor_time + "', '" +
assembly_ID + "', '" + process_ID
                                + "')");
                        System.out.println("Successfully added new cut-job account\n");

                        // Print queries if flag is set
                        if (query_flag == 1) {
                                print_queries();
                        }
                } catch (SQLException e) {
                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                        System.out.print("Please select a number corresponding to the query: ");
                }

        } else if (job_type.equalsIgnoreCase("Paint") || job_type.equalsIgnoreCase("Paint ")) {
                // Query color
                System.out.println();
                System.out.print("Color: ");
                String color = scanner.nextLine();

                // Query volume
                System.out.println();
                System.out.print("Volume: ");
                Double volume = scanner.nextDouble();
                scanner.nextLine();

                // Query labor_time
                System.out.println();
                System.out.print("Labor Time: ");
                Double labor_time = scanner.nextDouble();
                scanner.nextLine();

                // Adding new cut job to database
                try (final Statement statement = connection.createStatement();) {
```

```
                    statement.executeUpdate("UPDATE Job\n" + "SET date_completed = '" +
date_completed + "'\n"
                                    + "WHERE job_ID = " + job_ID);
                    System.out.println();
                    System.out.println("Successfully updated job with new information\n");

                    statement.executeUpdate(
                                    "INSERT INTO Paint_Job(job_ID, color, volume, labor_time,
assembly_ID, proc_ID)\n"
                                                    + "VALUES ('" + job_ID + "', '" + color +
"', '" + volume + "', '" + labor_time
                                                    + "', '" + assembly_ID + "', '" +
process_ID + "')");
                    System.out.println("Successfully added new paint-job account\n");

                    // Print queries if flag is set
                    if (query_flag == 1) {
                            print_queries();
                    }
            } catch (SQLException e) {
                    System.out.println("\nERROR: " + e.getMessage() + "\n");
                    System.out.print("Please select a number corresponding to the query: ");
            }

        } else if (job_type.equalsIgnoreCase("Fit") || job_type.equalsIgnoreCase("Fit ")) {
                // Query labor_time
                System.out.println();
                System.out.print("Labor Time: ");
                Double labor_time = scanner.nextDouble();
                scanner.nextLine();

                // Adding new cut job to database
                try (final Statement statement = connection.createStatement();) {
                    statement.executeUpdate("UPDATE Job\n" + "SET date_completed = '" +
date_completed + "'\n"
                                    + "WHERE job_ID = " + job_ID);
                    System.out.println();
                    System.out.println("Successfully updated job with new information\n");

                    statement.executeUpdate(
```

```
                                                "INSERT INTO Fit_Job(job_ID, labor_time, assembly_ID,
proc_ID)\n" + "VALUES ('" + job_ID
                                                        + "', '" + labor_time + "', '" + assembly_ID
+ "', '" + process_ID + "')");
                                        System.out.println("Successfully added new paint-job account\n");

                                        // Print queries if flag is set
                                        if (query_flag == 1) {
                                                print_queries();
                                        }
                                } catch (SQLException e) {
                                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                                        System.out.print("Please select a number corresponding to the query: ");
                                }

                        } else {
                                System.out.println();
                                System.out.println("Please enter job type: 'Cut' for Cut, 'Paint' for Paint, or 'Fit' for Fit");
                                System.out.println();
                                print_queries();
                        }
                } else {
                        System.out.println();
                        System.out.println("Job ID " + job_ID + " does not exist, please enter an existing ID");

                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                }

                return job_type_ID;
        }

// COMPLETELY DONE: Function enters new transaction: Query 8
        public static int enter_new_transaction(Connection connection, Scanner scanner, int query_flag)
                        throws SQLException {

                // Print existing job ID
                System.out.println();
                selective_print(connection, "Transactions", "", 1, "--Transaction ID");
```

```
                    // Query a job_ID
                    System.out.println();
                    System.out.print("New Transaction ID: ");
                    int transaction_ID = scanner.nextInt();
                    scanner.nextLine();

                    // If transaction ID already exists
                    if (check_key(connection, Integer.toString(transaction_ID), "Transactions", "", 1)) {
                            System.out.println();
                            System.out.println("Transaction ID " + transaction_ID + " does not exists, please enter an existing
ID");

                            System.out.println();
                            print_queries();

                            ERROR_FLAG = 1;
                    } else {

                            // Query for sup-cost
                            System.out.println();
                            System.out.print("Sup Cost: ");
                            int sup_cost = scanner.nextInt();
                            scanner.nextLine();

                            // Print existing job ID
                            System.out.println();
                            selective_print(connection, "Job", "", 1, "--Job ID");
                            // Query for job_ID
                            System.out.println();
                            System.out.print("Existing Job ID for transaction: ");
                            int job_ID = scanner.nextInt();
                            scanner.nextLine();

                            // If transaction ID already exists
                            if (check_key(connection, Integer.toString(transaction_ID), "Transactions", "", 1)) {
                                    System.out.println();
                                    System.out
                                                    .println("Transaction ID " + transaction_ID + " does not exists, please
enter an existing ID");

                                    System.out.println();
```

```
                    print_queries();

                    ERROR_FLAG = 1;
                } else {

                    // Adding new job to database
                    try (final Statement statement = connection.createStatement();) {
                        statement.executeUpdate("INSERT INTO Transactions(trans_ID, sup_cost,
job_ID)\n" + "VALUES ('"
                                + transaction_ID + "', '" + sup_cost + "', '" + job_ID + "')");
                        System.out.println();
                        System.out.println("Successfully added new transaction\n");

                    } catch (SQLException e) {
                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                        System.out.print("Please select a number corresponding to the query: ");
                        ERROR_FLAG = 1;
                    }

                    // Starting looping to query for accounts that are affected by transaction
                    if (ERROR_FLAG == 0) {

                        // Start querying for account types and IDs to update
                        System.out.print(
                                "What type of account(s) would you like to update
(Department, Assembly, Process): ");
                        String type_query = scanner.nextLine();

                        while (!type_query.equalsIgnoreCase("Quit")) {

                            if (type_query.equalsIgnoreCase("Department") ||
type_query.equalsIgnoreCase("Department ")) {
                                // Print existing Department Account ID
                                System.out.println();
                                selective_print(connection, "Department_Account", "", 1, "--
Department Account ID");

                                // Query for department ID
                                System.out.println();
                                System.out.print("Existing Department_Account ID: ");
                                int department_account_ID = scanner.nextInt();
```

```
                scanner.nextLine();

                // Update department_account
                try (final Statement statement =
connection.createStatement();) {
                    statement.executeUpdate("INSERT INTO
Updates(account_ID, trans_ID)\n" + "VALUES ('"
                            + department_account_ID + "',
'" + transaction_ID + "')");

                    statement.executeUpdate("UPDATE
Department_Account\n"
                            + "SET details_2 += (SELECT
sup_cost FROM Transactions WHERE trans_ID = "
                            + transaction_ID + ") WHERE
account_ID = " + department_account_ID);

                    System.out.println();
                    System.out.println("Successfully updated
department account\n");

                } catch (SQLException e) {
                    System.out.println("\nERROR: " + e.getMessage()
+ "\n");
                    System.out.print("Update failed");
                    break;
                }

                System.out.print(
                        "What type of account would you like to
update (Department, Assembly, Process, Quit): ");
                // Re-query for next account or quit
                type_query = scanner.nextLine();
            } else if (type_query.equalsIgnoreCase("Assembly")
                    || type_query.equalsIgnoreCase("Assembly ")) {
                // Print existing Assembly Account ID
                System.out.println();
                selective_print(connection, "Assembly_Account", "", 1, "--
Assembly Account ID");

                // Query for assembly ID
```

```
                System.out.println();
                System.out.print("Existing Assembly_Account ID: ");
                int assembly_account_ID = scanner.nextInt();
                scanner.nextLine();

                // Update assembly_account
                try (final Statement statement =
connection.createStatement();) {
                    statement.executeUpdate("INSERT INTO
Updates(account_ID, trans_ID)\n" + "VALUES ('"
                            + assembly_account_ID + "', '"
+ transaction_ID + "')");

                    statement.executeUpdate("UPDATE
Assembly_Account\n"
                            + "SET details_1 += (SELECT
sup_cost FROM Transactions WHERE trans_ID = "
                            + transaction_ID + ") WHERE
account_ID = " + assembly_account_ID);

                    System.out.println();
                    System.out.println("Successfully updated
assembly account\n");

                } catch (SQLException e) {
                    System.out.println("\nERROR: " + e.getMessage()
+ "\n");

                    System.out.print("Update failed");
                    break;
                }

                System.out.print(
                        "What type of account would you like to
update (Department, Assembly, Process, Quit): ");

                // Re-query for next account or quit
                type_query = scanner.nextLine();
            } else if (type_query.equalsIgnoreCase("Process") ||
type_query.equalsIgnoreCase("Process ")) {

                // Print existing Process Account ID
                System.out.println();
```

```
                                        selective_print(connection, "Process_Account", "", 1, "--
Process Account ID");

                                        // Query for process ID
                                        System.out.println();
                                        System.out.print("Existing Process_Account ID: ");
                                        int process_account_ID = scanner.nextInt();
                                        scanner.nextLine();

                                        // Update process_account
                                        try (final Statement statement =
connection.createStatement();) {
                                                statement.executeUpdate("INSERT INTO
Updates(account_ID, trans_ID)\n" + "VALUES ('"
                                                        + process_account_ID + "', '"
+ transaction_ID + "')");

                                                statement.executeUpdate("UPDATE
Process_Account\n"
                                                        + "SET details_3 += (SELECT
sup_cost FROM Transactions WHERE trans_ID = "
                                                        + transaction_ID + ") WHERE
account_ID = " + process_account_ID);

                                                System.out.println();
                                                System.out.println("Successfully updated process
account\n");

                                        } catch (SQLException e) {
                                                System.out.println("\nERROR: " + e.getMessage()
+ "\n");
                                                System.out.print("Update failed");
                                                break;
                                        }

                                        System.out.print(
                                                        "What type of account would you like to
update (Department, Assembly, Process, Quit): ");
                                        // Re-query for next account or quit
                                        type_query = scanner.nextLine();
                                } else {
```

```
                                                System.out.println();
                                                System.out.print(
                                                        "Please 'Department' for department
account, 'Assembly' for assembly account, 'Process' for process account, or 'Quit' to stop adding affected accounts");
                                                }

                                            }

                                            System.out.println();
                                            print_queries();
                                        }
                                    }
                                }

                                return transaction_ID;
                            }

// COMPLETELY DONE: Function retrieves cost incurred on assembly ID
                            public static int query_9(Connection connection, Scanner scanner, int query_flag) throws SQLException {
                                int account_ID = 0;

                                // Print assembly table
                                print_table(connection, 4);
                                // Query for assembly ID
                                System.out.println();
                                System.out.print("Existing Assembly ID: ");
                                int assembly_ID = scanner.nextInt();
                                scanner.nextLine();

                                // If Assembly ID already exists
                                if (check_key(connection, Integer.toString(assembly_ID), "Assembly", "", 1)) {

                                    // Query for total cost on assembly ID
                                    try (final Statement statement = connection.createStatement();) {
                                        final ResultSet resultSet = statement.executeQuery("SELECT AA.details_1\n"
                                                + "FROM Assembly AS A JOIN Assembly_Account AS AA ON
A.account_ID = AA.account_ID\n"
                                                + "WHERE A.assembly_ID = " + assembly_ID);
                                        resultSet.next();
                                        // Getting cost-incurred
                                        System.out.println();
```

```
                                        System.out.println("Cost incurred on Assembly ID " + assembly_ID + ": " +
resultSet.getString(1));

                                        // Print queries if flag is set
                                        if (query_flag == 1) {
                                                System.out.println();
                                                print_queries();
                                        }
                                } catch (SQLException e) {
                                        System.out.println("\nERROR: " + e.getMessage()
                                                        + ", either because there is no account associated with this assembly,
or there are no costs on this assembly yet\n");
                                        System.out.print("Please select a number corresponding to the query: ");
                                        ERROR_FLAG = 1;
                                }
                        } else {
                                System.out.println();
                                System.out.println("Assembly ID " + assembly_ID + " does not exist, please enter an existing ID");

                                System.out.println();
                                print_queries();

                                ERROR_FLAG = 1;
                        }

                        return account_ID;
                }

// COMPLETELY DONE: Function returns total labor time:
                public static void query_10(Connection connection, Scanner scanner, int query_flag) throws SQLException {

                        // Print table where date_completed is not null
                        print_table(connection, 10);
                        // Query for job_ID
                        System.out.println();
                        System.out.print("Existing Date Completed: ");
                        String date_completed = scanner.nextLine();

                        // Check to make sure that the date completed exists exists
                        if (check_key(connection, date_completed, "Job", "WHERE date_completed IS NOT NULL", 3)) {
                                // Job_ID exists in table, now check dept_ID
```

```
                    // Print table where date_completed is not null
                    print_table(connection, 2);
                    // Query for dept_ID
                    System.out.println();
                    System.out.print("Department ID: ");
                    String dept_ID = scanner.nextLine();

                    // Check to make sure that department ID exists
                    if (check_key(connection, dept_ID, "Department", "", 1)) {

                        // Query for total labor time in a department
                        try (final Statement statement = connection.createStatement();) {
                            final ResultSet resultSet = statement.executeQuery(
                                    "SELECT SUM(ISNULL(C.labor_time, 0)) +
SUM(ISNULL(PJ.labor_time, 0)) + SUM(ISNULL(F.labor_time, 0)) AS total_labor_time\n"
                                            + "FROM Job J LEFT OUTER JOIN
Cut_Job C \n"
                                            + "ON J.job_ID = C.job_ID LEFT
OUTER JOIN Paint_Job PJ \n"
                                            + "ON J.job_ID = PJ.job_ID LEFT
OUTER JOIN Fit_Job F \n"
                                            + "ON J.job_ID = F.job_ID JOIN Process
P \n"
                                            + "ON J.proc_ID = P.proc_ID JOIN
Department D \n" + "ON P.dept_ID = D.dept_ID \n"
                                            + "WHERE J.date_completed = '" +
date_completed + "' AND D.dept_ID = " + dept_ID);
                            resultSet.next();
                            // Getting cost-incurred
                            System.out.println();
                            System.out.println("Total labor time on Department ID " + dept_ID + ": " +
resultSet.getString(1));

                        } catch (SQLException e) {
                            System.out.println("\nERROR: " + e.getMessage() + "\n");
                            System.out.print("Please select a number corresponding to the query: ");
                            ERROR_FLAG = 1;
                        }

                    } else {
                        // Key does not exist in table
```

```
                                        System.out.println();
                                        System.out.println(
                                                        "Department ID chosen does not exist, please enter query 10 again
and enter Department ID from table above");
                                }

                        } else {
                                // Key does not exist in table
                                System.out.println();
                                System.out.println(
                                                "Date completed does not exist, please enter query 10 again and enter Job ID
from table above");
                        }

                        System.out.println();
                        print_queries();
                }

// COMPLETELY DONE: Function retrieves processes through an assembly-id and
                // department responsible
                public static void query_11(Connection connection, Scanner scanner, int query_flag) throws SQLException {
                        // Print existing assembly ID
                        System.out.println();
                        selective_print(connection, "Assembly", "", 1, "--Assembly ID");
                        // Query for assembly ID
                        System.out.println();
                        System.out.print("Existing Assembly ID: ");
                        int assembly_ID = scanner.nextInt();
                        scanner.nextLine();

                        // Check to make sure that the date completed exists exists
                        if (check_key(connection, Integer.toString(assembly_ID), "Assembly", "", 1)) {
                                // Query for total cost on assembly ID
                                try (final Statement statement = connection.createStatement();) {
                                        final ResultSet resultSet = statement.executeQuery("SELECT J.proc_ID,
J.date_commenced, D.dept_ID "
                                                        + "FROM Job J JOIN Process P ON J.proc_id = P.proc_ID JOIN
Department D ON P.dept_ID = D.dept_ID "
                                                        + "WHERE J.assembly_ID = " + assembly_ID + "\n" + "ORDER BY
J.date_commenced");

                                        System.out.println("--Process ID--|--Date Commenced--|--Department ID--");
```

```
                                        // Printing out table
                                        while (resultSet.next()) {
                                                System.out.println(String.format("----%s | %s | %s", resultSet.getString(1),
resultSet.getString(2),
                                                                                        resultSet.getString(3)));
                                        }

                                        // Print queries if flag is set
                                        if (query_flag == 1) {
                                                System.out.println();
                                                print_queries();
                                        }
                                } catch (SQLException e) {
                                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                                        System.out.print("Please select a number corresponding to the query: ");
                                        ERROR_FLAG = 1;
                                }

                        } else {
                                System.out.println();
                                System.out.println("Assembly ID " + assembly_ID + " does not exist, please enter an existing ID");

                                System.out.println();
                                print_queries();

                                ERROR_FLAG = 1;
                        }
                }

// COMPLETELY DONE: Retrieve the jobs completed during a given date in a given
                // department
                public static void query_12(Connection connection, Scanner scanner, int query_flag) throws SQLException {
                        // Print existing department ID
                        System.out.println();
                        selective_print(connection, "Department", "", 1, "--Department ID");
                        // Enter a department ID
                        System.out.println();
                        System.out.print("Existing Department ID: ");
                        int department_ID = scanner.nextInt();
                        scanner.nextLine();
```

```
                // If department name already exists
                if (check_key(connection, Integer.toString(department_ID), "Department", "", 1)) {

                        // Printing only dates
                        System.out.println();
                        System.out.println("Dates of completed jobs");
                        selective_print(connection, "Job", "WHERE date_completed IS NOT NULL", 3, "--Date Completed");

                        // Enter a date completed
                        System.out.println();
                        System.out.print("Existing Date Completed: ");
                        String date_completed = scanner.nextLine();

                        // Query for total cost on assembly ID
                        try (final Statement statement = connection.createStatement();) {
                                final ResultSet resultSet1 = statement.executeQuery("SELECT CJ.job_ID,
CJ.assembly_ID\n"
                                                + "FROM Cut_Job CJ JOIN Job J ON CJ.job_ID = J.job_ID\n"
                                                + "JOIN Process P ON J.proc_id = P.proc_ID\n" + "JOIN Department D
ON P.dept_ID = D.dept_ID\n"
                                                + "WHERE D.dept_ID = " + department_ID + " AND J.date_completed
= '" + date_completed + "'");

                                System.out.println();
                                System.out.println("Cut_Jobs:");
                                System.out.println(
                                                "--Job ID--|--Machine Type--|--Machine Time--|--Material|--Labor Time--
|--Assembly ID--");
                                // Printing out table
                                while (resultSet1.next()) {
                                        System.out.println(String.format("----%s | %s | %s | %s | %s | %s",
resultSet1.getString(1),
                                                        resultSet1.getString(2), resultSet1.getString(3),
resultSet1.getString(4),
                                                        resultSet1.getString(5), resultSet1.getString(6)));
                                }

                                final ResultSet resultSet2 = statement
                                                .executeQuery("SELECT PJ.job_ID, color, volume, labor_time,
PJ.assembly_ID\n"
```

```
                                                        + "FROM Paint_Job AS PJ JOIN Job AS J ON
PJ.job_ID = J.job_ID\n"
                                                        + "JOIN Process P ON J.proc_id = P.proc_ID\n"
                                                        + "JOIN Department D ON P.dept_ID =
D.dept_ID\n" + "WHERE D.dept_ID = " + department_ID
                                                        + " AND J.date_completed =  '" + date_completed
+ "'");

                                System.out.println();
                                System.out.println("Paint_Jobs: ");
                                System.out.println("--Job ID--|--Color--|--Volume--|--Labor Time--|--Assembly ID--");
                                // Printing out table
                                while (resultSet2.next()) {
                                        System.out.println(String.format("----%s | %s | %s | %s | %s",
resultSet2.getString(1),
                                                        resultSet2.getString(2), resultSet2.getString(3),
resultSet2.getString(4),
                                                        resultSet2.getString(5)));
                                }

                                final ResultSet resultSet3 = statement.executeQuery("SELECT FJ.job_ID, FJ.assembly_ID
"
                                                + "FROM Fit_Job FJ JOIN Job J ON FJ.job_ID = J.job_ID JOIN
Process P "
                                                + "ON J.proc_id = P.proc_ID JOIN Department D ON P.dept_ID =
D.dept_ID " + "WHERE D.dept_ID = "
                                                + department_ID + " AND J.date_completed = '" + date_completed +
"'");

                                System.out.println();
                                System.out.println("Fit_Jobs: ");
                                System.out.println("--Job ID--|--Labor Time--|--Assembly ID--");
                                // Printing out table
                                while (resultSet3.next()) {
                                        System.out.println(String.format("----%s | %s | %s", resultSet3.getString(1),
                                                        resultSet3.getString(2), resultSet3.getString(3)));
                                }

                                // Print queries if flag is set
                                if (query_flag == 1) {
                                        System.out.println();
```

```
                                        print_queries();
                                }
                        } catch (SQLException e) {
                                System.out.println("\nERROR: " + e.getMessage() + "\n");
                                System.out.print("Please select a number corresponding to the query: ");
                                ERROR_FLAG = 1;
                        }
                } else {
                        System.out.println();
                        System.out.println("Department ID " + department_ID + " does not exist, please enter an existing ID");

                        System.out.println();
                        print_queries();

                        ERROR_FLAG = 1;
                }
        }

        // COMPELTELY DONE: Retrieve customer whose category is in a given range
        public static void query_13(Connection connection, Scanner scanner, int query_flag) throws SQLException {
                // Query for category number
                System.out.println();
                System.out.print("First Category Number: ");
                int category_1 = scanner.nextInt();
                scanner.nextLine();
                System.out.println();
                System.out.print("Second Category Number: ");
                int category_2 = scanner.nextInt();
                scanner.nextLine();

                // Query for total cost on assembly ID
                try (final Statement statement = connection.createStatement();) {
                        final ResultSet resultSet = statement
                                        .executeQuery("SELECT customer_name, customer_address,
customer_category\n" + "FROM Customer\n"
                                                        + "WHERE customer_category BETWEEN " + category_1 +
" AND " + category_2 + "\n"
                                                        + "ORDER BY (customer_name)");

                        System.out.println("--Customer Name--|--Customer Address--|--Category Number--");
                        // Printing out table
```

```
                while (resultSet.next()) {
                        System.out.println(String.format("----%s | %s | %s", resultSet.getString(1),
resultSet.getString(2),
                                                        resultSet.getString(3)));
                }

                // Print queries if flag is set
                if (query_flag == 1) {
                        System.out.println();
                        print_queries();
                }
        } catch (SQLException e) {
                System.out.println("\nERROR: " + e.getMessage() + "\n");
                System.out.print("Please select a number corresponding to the query: ");
                ERROR_FLAG = 1;
        }
    }

// COMPLETELY DONE: Delete all cut jobs whose job no is in a given range
    public static void query_14(Connection connection, Scanner scanner, int query_flag) throws SQLException {
        // Print all cut jobs
        print_table(connection, 11);

        // Query for range
        System.out.println();
        System.out.print("First Range Number: ");
        int range_1 = scanner.nextInt();
        scanner.nextLine();
        System.out.println();
        System.out.print("Second Range Number: ");
        int range_2 = scanner.nextInt();
        scanner.nextLine();

        // Deleting Cut jobs in range
        try (final Statement statement = connection.createStatement();) {
            statement.executeUpdate("DELETE Cut_Job\n" + "WHERE Cut_Job.job_ID BETWEEN " + range_1 +
" AND " + range_2);

            System.out.println("Deleted Cut Jobs in range " + range_1 + " and " + range_2);

            // Print queries if flag is set
```

```
                    if (query_flag == 1) {
                        System.out.println();
                        print_queries();
                    }
                } catch (SQLException e) {
                    System.out.println("\nERROR: " + e.getMessage() + "\n");
                    System.out.print("Please select a number corresponding to the query: ");
                    ERROR_FLAG = 1;
                }
            }

            // COMPLETELY DONE Change color of a given paint job
            public static void query_15(Connection connection, Scanner scanner, int query_flag) throws SQLException {

                // Print existing Paint_Job ID
                System.out.println();
                selective_print(connection, "Paint_Job", "", 1, "--Paint_Job ID");
                // Asking for Paint_Job ID
                System.out.println();
                System.out.print("New Paint_Job ID: ");
                int job_ID = scanner.nextInt();
                scanner.nextLine();

                // If job ID already exists
                if (check_key(connection, Integer.toString(job_ID), "Paint_Job", "", 1)) {
                    // Asking for new Paint_Job color
                    System.out.println();
                    System.out.print("New Color: ");
                    String color = scanner.nextLine();

                    // Deleting Cut jobs in range
                    try (final Statement statement = connection.createStatement();) {
                        statement.executeUpdate(
                                "UPDATE Paint_Job\n" + "SET color = '" + color + "'\n" + "WHERE
Paint_Job.job_ID = " + job_ID);

                        System.out.println();
                        System.out.println("Updated Paint_Job " + job_ID + " to new color");

                        // Print queries if flag is set
                        if (query_flag == 1) {
```

```
                                System.out.println();
                                print_queries();
                        }
                } catch (SQLException e) {
                        System.out.println("\nERROR: " + e.getMessage() + "\n");
                        System.out.print("Please select a number corresponding to the query: ");
                        ERROR_FLAG = 1;
                }
        } else {
                System.out.println();
                System.out.println("Paint_Job ID " + job_ID + " does not exist, please enter an existing ID");

                System.out.println();
                print_queries();

                ERROR_FLAG = 1;
        }

}

// COMPLETELY DONE: Import new customers
        public static void query_16(Connection connection, Scanner scanner, int query_flag)
                        throws SQLException, IOException {

                // Query for file path
                System.out.println();
                System.out.print("Please enter the file path: ");
                String file_path = scanner.nextLine();

                // Attaching src to file name
                file_path = "src/" + file_path;

                // Executing statements when reading in file lines
                try (final Statement statement = connection.createStatement();) {
                        // File line
                        try {
                                // Declaring variable
                                String file_line = "";
                                File f = new File(file_path);
                                FileReader fr = new FileReader(f);
                                // Reading in file lines
```

```
                BufferedReader buffer = new BufferedReader(fr);

                try {
                        // Reading in line by line
                        while ((file_line = buffer.readLine()) != null) {
                                // Split string on commas
                                String[] customer = file_line.split(",");

                                statement.executeUpdate(
                                                "INSERT INTO Customer(customer_name,
customer_address, customer_category)\n"
                                                                + "VALUES ('" + customer[0]
+ "','" + customer[1] + "','" + customer[2] + "')");
                        }

                        System.out.println("Successfully added new customers");

                } catch (SQLException e) {
                        System.out.println();
                        System.out.println("ERROR: " + e.getMessage() + ", 1");
                        System.out.println();
                        System.out.print("Please select a number corresponding to the query: ");
                } finally {
                        try {
                                // Close buffer
                                buffer.close();
                        } catch (IOException e1) {
                                System.out.println();
                                System.out.println("ERROR: " + e1.getMessage());
                                System.out.println();
                                System.out.print("Please select a number corresponding to the query:
");
                        }
                }

                // Print out queries
                if (query_flag == 1) {
                        System.out.println();
                        print_queries();
                }
        } catch (FileNotFoundException e) {
```

```
                                System.out.println();
                                System.out.println("ERROR: " + e.getMessage());
                                System.out.println();
                                System.out.print("Please select a number corresponding to the query: ");
                        }
                }

        }

// COMPLETELY DONE: Export all customers to new file
        public static void query_17(Connection connection, Scanner scanner, int query_flag)
                        throws SQLException, IOException {

                // Query for range 1
                System.out.println();
                System.out.print("Enter range start: ");
                int range_1 = scanner.nextInt();
                scanner.nextLine();

                // Query for range 2
                System.out.println();
                System.out.print("Enter range end: ");
                int range_2 = scanner.nextInt();

                scanner.nextLine();
                // Query for output file
                System.out.println();
                System.out.print("Enter output file name: ");
                String file_output = scanner.nextLine();

                try (final Statement statement = connection.createStatement();) {
                        ResultSet resultSet = statement
                                        .executeQuery("SELECT * " + "FROM Customer\n" + "WHERE
customer_category BETWEEN " + range_1
                                                        + " AND " + range_2 + "\n" + "ORDER BY customer_name");

                        FileWriter wrt = new FileWriter(file_output + ".csv");
                        StringBuilder builder = new StringBuilder();
                        File output = new File(file_output + ".csv");

                        while (resultSet.next()) {
                                if (resultSet.isBeforeFirst()) {
```

```
                                        builder.append(resultSet.getString(1));
                                        builder.append(",");
                                        builder.append(resultSet.getString(2));
                                        builder.append(",");
                                        builder.append(resultSet.getString(3));
                                        builder.append("\n");
                                } else {
                                        builder.append(resultSet.getString(1));
                                        builder.append(",");
                                        builder.append(resultSet.getString(2));
                                        builder.append(",");
                                        builder.append(resultSet.getString(3));
                                        builder.append("\n");
                                }
                        }

                        wrt.append(builder);
                        wrt.flush();
                        wrt.close();

                        System.out.println();
                        System.out
                                        .println("Successfully added all customers in category range to file: " +
output.getAbsolutePath());

                        // Print out queries
                        if (query_flag == 1) {
                                System.out.println();
                                print_queries();
                        }

                } catch (SQLException e) {
                        System.out.println();
                        System.out.println("ERROR: " + e.getMessage() + ", 1");
                        System.out.println();
                        System.out.print("Please select a number corresponding to the query: ");
                }
        }

// All of the SQL and querying for each option
        public static void queries(String url) throws SQLException, IOException {
```

```
try (final Connection connection = DriverManager.getConnection(url)) {
        // Successful connection printout
        final String schema = connection.getSchema();
        System.out.println("Successful connection - Schema:" + schema);

        // Printing out header
        System.out.println("WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM");
        System.out.println("================================================");
        // Print out query options
        print_queries();
        // Scanner to take in input
        Scanner scanner = new Scanner(System.in);

        int usr_input = 0;
        // Loop until user has quit the program
        while (scanner.hasNext()) {

                // Take in user input for option
                usr_input = scanner.nextInt();
                scanner.nextLine();

                // Using switch for different query options
                switch (usr_input) {

                case 1: // Query 1
                        enter_new_customer(connection, scanner, 1);
                        ERROR_FLAG = 0;
                        break;
                case 2: // Query 2
                        enter_new_department(connection, scanner, 1);
                        ERROR_FLAG = 0;
                        break;
                case 3: // Query 3
                        query_3(connection, scanner);
                        ERROR_FLAG = 0;
                        break;
                case 4: // Query 4
                        query_4(connection, scanner);
                        ERROR_FLAG = 0;
                        break;
```

```
case 5: // Query 5
        query_5(connection, scanner);
        ERROR_FLAG = 0;
        break;
case 6: // Query 6
        enter_new_job(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 7: // Query 7
        enter_new_job_type(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 8: // Query 8
        enter_new_transaction(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 9: // Query 9
        query_9(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 10: // Query 10
        query_10(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 11: // Query 11
        query_11(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 12:
        query_12(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 13:
        query_13(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 14:
        query_14(connection, scanner, 1);
        ERROR_FLAG = 0;
        break;
case 15:
```

```
                    query_15(connection, scanner, 1);
                    ERROR_FLAG = 0;
                    break;
                case 16:
                    query_16(connection, scanner, 1);
                    ERROR_FLAG = 0;
                    break;
                case 17:
                    query_17(connection, scanner, 1);
                    ERROR_FLAG = 0;
                    break;
                case 18:
                    System.out.println("You have exited the program");
                    System.exit(0);

                default: // if the user did not input a valid number

                    System.out.println();
                    System.out.println("Please enter a number from 1 to 18\n");
                    print_queries();
                    System.out.println();
                    System.out.print("Please select the number corresponding to the query: ");
                    // Print out query options
                    continue;

                }
            }
        }
    }

    public static void signin() throws SQLException, IOException {
        // Connect to database
        final String hostName = "113380402-sql-server.database.windows.net";
        final String dbName = "cs-dsa-4513-sql-db";
        final String user = "ZAKI-113380402";
        final String password = "Sims31997^";
        final String url = String.format(
                "jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;host
NameInCertificate=*.database.windows.net;loginTimeout=30;",
                hostName, dbName, user, password);
```

```
            // Handle queries
            queries(url);
    }

    public static void main(String[] args) throws SQLException, IOException {
            signin();
    }
```

Task 6: Java Program Execution

6.1 Screenshots of Query 1

```
Successful connection - Schema:dbo
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 to the 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 by adding sup-cos
(9) Retrieve the cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the departm
(12) Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given de
(13) Retrieve the customers (in name order) whose category is in a given range
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import new customers from file
(17) Export customers to file
(18) Quit

Please select the number corresponding to the query:
```

```
Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name

New Customer Name: Zaki Refai

Customer's address: 1063 Greystone Crest

Customer's category (1-10): 4

Successfully added new customer
```

	customer_name	customer_address	customer_category
1	Zaki Refai	1063 Greystone Crest	4

```
Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name
----Zaki Refai

New Customer Name: Sammy Najib

Customer's address: 750 Imhoff Rd

Customer's category (1-10): 6

Successfully added new customer
```

	customer_name	customer_address	customer_category
1	Sammy Najib	750 Imhoff Rd	6
2	Zaki Refai	1063 Greystone Crest	4

```
Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name
----Sammy Najib
----Zaki Refai

New Customer Name: Trevor Bryant

Customer's address: 750 W Imhoff Rd

Customer's category (1-10): 8

Successfully added new customer
```

	customer_name	customer_address	customer_category
1	Sammy Najib	750 Imhoff Rd	6
2	Trevor Bryant	750 W Imhoff Rd	8
3	Zaki Refai	1063 Greystone Crest	4

```
Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name
----Sammy Najib
----Trevor Bryant
----Zaki Refai

New Customer Name: Sarah Kayali

Customer's address: 1063 Greystone Crest

Customer's category (1-10): 7

Successfully added new customer
```

	customer_name	customer_address	customer_category
1	Sammy Najib	750 Imhoff Rd	6
2	Sarah Kayali	1063 Greystone Crest	7
3	Trevor Bryant	750 W Imhoff Rd	8
4	Zaki Refai	1063 Greystone Crest	4

```
Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Zaki Refai

New Customer Name: Trey Sullivent

Customer's address: Traditions East

Customer's category (1-10): 2

Successfully added new customer
```

	customer_name	customer_address	customer_category
1	Sammy Najib	750 Imhoff Rd	6
2	Sarah Kayali	1063 Greystone Crest	7
3	Trevor Bryant	750 W Imhoff Rd	8
4	Trey Sullivent	Traditions East	2
5	Zaki Refai	1063 Greystone Crest	4

6.2 Screenshots of Query 2

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID

New Department ID: 100

Department Data: CS

Successfully added new department
```

	dept_ID	dept_data	account_ID
1	100	CS	*NULL*

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID
----100

New Department ID: 101

Department Data: MIS

Successfully added new department
```

	dept_ID	dept_data	account_ID
1	100	CS	*NULL*
2	101	MIS	*NULL*

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID
----100
----101

New Department ID: 101

Department ID 101 already exists, please enter a new ID
```

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID
----100
----101

New Department ID: 102

Department Data: AEREO

Successfully added new department
```

	dept_ID	dept_data	account_ID
1	100	CS	*NULL*
2	101	MIS	*NULL*
3	102	AEREO	*NULL*

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID
----100
----101
----102

New Department ID: 103

Department Data: SPACE

Successfully added new department
```

	dept_ID	dept_data	account_ID
1	100	CS	*NULL*
2	101	MIS	*NULL*
3	102	AEREO	*NULL*
4	103	SPACE	*NULL*

```
Please select the number corresponding to the query: 2

Existing Department entries
--Department ID
----100
----101
----102
----103

New Department ID: 104

Department Data: UP

Successfully added new department
```

	dept_ID	dept_data	account_ID
1	100	CS	*NULL*
2	101	MIS	*NULL*
3	102	AEREO	*NULL*
4	103	SPACE	*NULL*
5	104	UP	*NULL*

6.3 Screenshot of Query 3

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Sammy Najib

Existing Assembly entries
--Assembly ID

New Assembly ID: 100

Assembly Ordered (YYYY-MM-DD): 2019-11-19

Assembly Details: Mill

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Sarah jsdjnsj

ERROR: Customer name does not exists, please pick an existing customer
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Trevor Bryant

Existing Assembly entries
--Assembly ID
----100

New Assembly ID: 101

Assembly Ordered (YYYY-MM-DD): 2019-12-11

Assembly Details: Mill

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Sammy Najib

Existing Assembly entries
--Assembly ID
----100
----101

New Assembly ID: 102

Assembly Ordered (YYYY-MM-DD): 2019-04-19

Assembly Details: Saw
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Trevor Bryant

Existing Assembly entries
--Assembly ID
----100
----101
----102

New Assembly ID: 103

Assembly Ordered (YYYY-MM-DD): 2019-09-11

Assembly Details: Mill

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	201 2019-11-19	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Zaki Refai

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103

New Assembly ID: 104

Assembly Ordered (YYYY-MM-DD): 2019-07-15

Assembly Details: Saw

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Trey Sullivant

ERROR: Customer name does not exists, please pick an existing customer
```

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Trey Sullivent

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104

New Assembly ID: 105

Assembly Ordered (YYYY-MM-DD): 2019-08-17

Assembly Details: take

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*
6	105	2019-08-17	take	Trey Sullivent	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Sarah Kayali

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105

New Assembly ID: 106

Assembly Ordered (YYYY-MM-DD): 2019-09-18

Assembly Details: Take

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*
6	105	2019-08-17	take	Trey Sullivent	*NULL*
7	106	2019-09-18	Take	Sarah Kayali	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Zaki Refai

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106

New Assembly ID: 107

Assembly Ordered (YYYY-MM-DD): 2019-11-09

Assembly Details: Saw

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*
6	105	2019-08-17	take	Trey Sullivent	*NULL*
7	106	2019-09-18	Take	Sarah Kayali	*NULL*
8	107	2019-11-09	Saw	Zaki Refai	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Trevor Bryant

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107

New Assembly ID: 108

Assembly Ordered (YYYY-MM-DD): 2019-07-10

Assembly Details: Pull

Successfully added new assembly
```

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*
6	105	2019-08-17	take	Trey Sullivent	*NULL*
7	106	2019-09-18	Take	Sarah Kayali	*NULL*
8	107	2019-11-09	Saw	Zaki Refai	*NULL*
9	108	2019-07-10	Pull	Trevor Bryant	*NULL*

```
Please select the number corresponding to the query: 3

Is customer for this assembly new or currently in database (new/current): current

Existing Customer entries
--Customer Name
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Zaki Refai

Existing Customer's Name: Zaki Refai

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108

New Assembly ID: 109

Assembly Ordered (YYYY-MM-DD): 2019-01-10

Assembly Details: Mill

Successfully added new assembly
```

No Notifications

	assembly_ID	date_ordered	assembly_details	customer_name	account_ID
1	100	2019-11-19	Mill	Sammy Najib	*NULL*
2	101	2019-12-11	Mill	Trevor Bryant	*NULL*
3	102	2019-04-19	Saw	Sammy Najib	*NULL*
4	103	2019-09-11	Mill	Trevor Bryant	*NULL*
5	104	2019-07-15	Saw	Zaki Refai	*NULL*
6	105	2019-08-17	take	Trey Sullivent	*NULL*
7	106	2019-09-18	Take	Sarah Kayali	*NULL*
8	107	2019-11-09	Saw	Zaki Refai	*NULL*
9	108	2019-07-10	Pull	Trevor Bryant	*NULL*
10	109	2019-01-10	Mill	Zaki Refai	*NULL*

6.4 Screenshots of Query 4

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Existing Process entries
--Process ID

Process ID: 100

Process Data: Saw
Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 101

Existing Process entries
--Process ID
----100

Process ID: 101

Process Data: Saw
Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 102

Existing Process entries
--Process ID
----100
----101

Process ID: 102

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 103

Existing Process entries
--Process ID
----100
----101
----102

Process ID: 103

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 104

Existing Process entries
--Process ID
----100
----101
----102
----103

Process ID: 104

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Existing Process entries
--Process ID
----100
----101
----102
----103
----104

Process ID: 105

Process Data: Duster

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*
6	105	Duster	100	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 101

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105

Process ID: 106

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*
6	105	Duster	100	*NULL*
7	106	Mill	101	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 102

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106

Process ID: 107

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*
6	105	Duster	100	*NULL*
7	106	Mill	101	*NULL*
8	107	Mill	102	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 103

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106
----107

Process ID: 108

Process Data: Mill

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*
6	105	Duster	100	*NULL*
7	106	Mill	101	*NULL*
8	107	Mill	102	*NULL*
9	108	Mill	103	*NULL*

```
Please select the number corresponding to the query: 4

Is department for this process new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 104

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106
----107
----108

Process ID: 109

Process Data: Saw

Successfully added new process
```

	proc_ID	process_data	dept_ID	account_ID
1	100	Saw	100	*NULL*
2	101	Saw	101	*NULL*
3	102	Mill	102	*NULL*
4	103	Mill	103	*NULL*
5	104	Mill	104	*NULL*
6	105	Duster	100	*NULL*
7	106	Mill	101	*NULL*
8	107	Mill	102	*NULL*
9	108	Mill	103	*NULL*
10	109	Saw	104	*NULL*

6.5: Screenshots of Query 5

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Department

Is department for this account new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Existing Account entries
--Account ID

New Account ID: 100

Account Date Established (YYYY-MM-DD): 2019-12-09

Details-2: 40

Successfully added new account

Successfully added new department account

Successfully updated department with new account
```

	account_ID	date_established
1	100	2019-12-09

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Department

Is department for this account new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Existing Account entries
--Account ID
----100

New Account ID: 101

Account Date Established (YYYY-MM-DD): 2019-09-09

Details-2: 700

Successfully added new account

Successfully added new department account

Successfully updated department with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Department

Is department for this account new or currently in database (new/current): current

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 101

Existing Account entries
--Account ID
----100
----101

New Account ID: 102

Account Date Established (YYYY-MM-DD): 2019-09-09

Details-2: 900

Successfully added new account

Successfully added new department account

Successfully updated department with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Assembly

Is assembly for this account new or currently in database (new/current): current

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 100

Existing Account entries
--Account ID
----100
----101
----102

New Account ID: 103

Account Date Established (YYYY-MM-DD): 2019-08-09

Details-1: 900

Successfully added new account

Successfully added new assembly account

Successfully updated assembly with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Assembly

Is assembly for this account new or currently in database (new/current): current

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 102

Existing Account entries
--Account ID
----100
----101
----102
----103

New Account ID: 104

Account Date Established (YYYY-MM-DD): 2019-07-09

Details-1: 600

Successfully added new account

Successfully added new assembly account

Successfully updated assembly with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09
5	104	2019-07-09

```
Database [Java Application] /Library/Java/JavaVirtualMachines/jdk-13.jdk/Contents/Home/bin/java (Nov 17, 2019,
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Assembly

Is assembly for this account new or currently in database (new/current): current

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 100

Existing Account entries
--Account ID
----100
----101
----102
----103
----104

New Account ID: 105

Account Date Established (YYYY-MM-DD): 2019-08-05

Details-1: 400

Successfully added new account

Successfully added new assembly account

Successfully updated assembly with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09
5	104	2019-07-09
6	105	2019-08-05

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Process

Is process for this account new or currently in database (new/current): current

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Process ID: 100

Existing Account entries
--Account ID
----100
----101
----102
----103
----104
----105

New Account ID: 106

Account Date Established (YYYY-MM-DD): 2019-03-10

Details-3: 100

Successfully added new account

Successfully added new process account

Successfully updated process with new account
```

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Process

Is process for this account new or currently in database (new/current): current

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Process ID: 101

Existing Account entries
--Account ID
----100
----101
----102
----103
----104
----105
----106

New Account ID: 107

Account Date Established (YYYY-MM-DD): 2019-09-07

Details-3: 100

Successfully added new account

Successfully added new process account

Successfully updated process with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09
5	104	2019-07-09
6	105	2019-08-05
7	106	2019-03-10

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): process

Is process for this account new or currently in database (new/current): current

Existing Process entries
--Process ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Process ID: 103

Existing Account entries
--Account ID
----100
----101
----102
----103
----104
----105
----106
----107

New Account ID: 108

Account Date Established (YYYY-MM-DD): 2019-02-01

Details-3: 300

Successfully added new account

Successfully added new process account

Successfully updated process with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09
5	104	2019-07-09
6	105	2019-08-05
7	106	2019-03-10
8	107	2019-09-07
9	108	2019-02-01

```
Please select the number corresponding to the query: 5

What type of account would you like (Department, Assembly, Process): Assembly

Is assembly for this account new or currently in database (new/current): current

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 102

Existing Account entries
--Account ID
----100
----101
----102
----103
----104
----105
----106
----107
----108

New Account ID: 109

Account Date Established (YYYY-MM-DD): 2019-07-10

Details-1: 200

Successfully added new account

Successfully added new assembly account

Successfully updated assembly with new account
```

	account_ID	date_established
1	100	2019-12-09
2	101	2019-09-09
3	102	2019-09-09
4	103	2019-08-09
5	104	2019-07-09
6	105	2019-08-05
7	106	2019-03-10
8	107	2019-09-07
9	108	2019-02-01
10	109	2019-07-10

6.6: Screenshots for Query 6

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID

New Job ID: 100

Date Commenced (YYYY-MM-DD): 2019-08-09

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 100

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 100

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100

New Job ID: 101

Date Commenced (YYYY-MM-DD): 2019-08-10

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 100

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 101

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101

New Job ID: 102

Date Commenced (YYYY-MM-DD): 2019-10-04

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 100

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 103

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102

New Job ID: 103

Date Commenced (YYYY-MM-DD): 2019-06-10

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 102

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 103
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103

New Job ID: 104

Date Commenced (YYYY-MM-DD): 2019-12-11

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 103

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 105

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104

New Job ID: 105

Date Commenced (YYYY-MM-DD): 2019-04

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 105

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 109

ERROR: Conversion failed when converting date and/or time from character string.
```

```
Please select a number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104

New Job ID: 105

Date Commenced (YYYY-MM-DD): 2019-08-10

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 106

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 109

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105
6	105	2019-08-10	*NULL*	106	109

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105

New Job ID: 106

Date Commenced (YYYY-MM-DD): 2019-06-05

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 106

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 104

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105
6	105	2019-08-10	*NULL*	106	109
7	106	2019-06-05	*NULL*	106	104

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106

New Job ID: 107

Date Commenced (YYYY-MM-DD): 2019-04-20

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 105

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 102

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105
6	105	2019-08-10	*NULL*	106	109
7	106	2019-06-05	*NULL*	106	104
8	107	2019-04-20	*NULL*	105	102

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107

New Job ID: 108

Date Commenced (YYYY-MM-DD): 2019-05-23

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 101

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 100

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105
6	105	2019-08-10	*NULL*	106	109
7	106	2019-06-05	*NULL*	106	104
8	107	2019-04-20	*NULL*	105	102
9	108	2019-05-23	*NULL*	101	100

```
Please select the number corresponding to the query: 6

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106

New Job ID: 107

Date Commenced (YYYY-MM-DD): 2019-04-20

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Assembly ID: 105

Contents of Process table:
--Process ID--|--Process Data--|--Department ID--|--Account ID--
----100 | Saw | 100 | 106
----101 | Saw | 101 | 107
----102 | Mill | 102 | null
----103 | Mill | 103 | 108
----104 | Mill | 104 | null
----105 | Duster | 100 | null
----106 | Mill | 101 | null
----107 | Mill | 102 | null
----108 | Mill | 103 | null
----109 | Saw | 104 | null

Process ID: 102

Successfully added new job
```

	job_ID	date_commenced	date_completed	assembly_ID	proc_ID
1	100	2019-08-09	*NULL*	100	100
2	101	2019-08-10	*NULL*	100	101
3	102	2019-10-04	*NULL*	100	103
4	103	2019-06-10	*NULL*	102	103
5	104	2019-12-11	*NULL*	103	105
6	105	2019-08-10	*NULL*	106	109
7	106	2019-06-05	*NULL*	106	104
8	107	2019-04-20	*NULL*	105	102
9	108	2019-05-23	*NULL*	101	100
10	109	2019-08-10	*NULL*	108	108

6.7: Screenshots for Query 7

--Tables can be seen with updated date completed, no need to print tables from SQL database

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | null | 100 | 100
----101 | 2019-08-10 | null | 100 | 101
----102 | 2019-10-04 | null | 100 | 103
----103 | 2019-06-10 | null | 102 | 103
----104 | 2019-12-11 | null | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 100

Date Completed of Job (YYYY-MM-DD): 2019-08-10

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Cut

Machine Type: Mill

Machine Time: 5.6

Material: cardboard

Labor Time: 6.5

Successfully updated job with new information

Successfullv added new cut-iob account
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | null | 100 | 101
----102 | 2019-10-04 | null | 100 | 103
----103 | 2019-06-10 | null | 102 | 103
----104 | 2019-12-11 | null | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 101

Date Completed of Job (YYYY-MM-DD): 2019-08-11

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Paint

Color: Purple

Volume: 6.5

Labor Time: 3.4

Successfully updated job with new information

Successfully added new paint-job account
```

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | null | 100 | 103
----103 | 2019-06-10 | null | 102 | 103
----104 | 2019-12-11 | null | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 102

Date Completed of Job (YYYY-MM-DD): 2019-10-06

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Paint

Color: red

Volume: 98.9

Labor Time: 2.3

Successfully updated job with new information

Successfully added new paint-job account
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | null | 102 | 103
----104 | 2019-12-11 | null | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 103

Date Completed of Job (YYYY-MM-DD): 2019-06-20

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Fit

Labor Time: 7.6

Successfully updated job with new information

Successfully added new paint-job account
```

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | null | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 104

Date Completed of Job (YYYY-MM-DD): 2019-09-10

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Cut

Machine Type: Mill

Machine Time: 20.9

Material: Weed

Labor Time: 90.2

Successfully updated job with new information

Successfully added new cut-job account

Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | 2019-09-10 | 103 | 105
----105 | 2019-08-10 | null | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 105

Date Completed of Job (YYYY-MM-DD): 2019-08-20

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Paint

Color: Blue

Volume: 8.7

Labor Time: 9.0

Successfully updated job with new information

Successfully added new paint-job account
```

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | 2019-09-10 | 103 | 105
----105 | 2019-08-10 | 2019-08-20 | 106 | 109
----106 | 2019-06-05 | null | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 106

Date Completed of Job (YYYY-MM-DD): 2019-12-09

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Paint

Color: Green

Volume: 4.5

Labor Time: 3.4

Successfully updated job with new information

Successfully added new paint-job account
```

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | 2019-09-10 | 103 | 105
----105 | 2019-08-10 | 2019-08-20 | 106 | 109
----106 | 2019-06-05 | 2019-12-09 | 106 | 104
----107 | 2019-04-20 | null | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 107

Date Completed of Job (YYYY-MM-DD): 2019-05-20

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Paint

Color: Yello

Volume: 3.4

Labor Time: 5.6

Successfully updated job with new information

Successfully added new paint-job account
```

```
Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | 2019-09-10 | 103 | 105
----105 | 2019-08-10 | 2019-08-20 | 106 | 109
----106 | 2019-06-05 | 2019-12-09 | 106 | 104
----107 | 2019-04-20 | 2019-05-20 | 105 | 102
----108 | 2019-05-23 | null | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 108

Date Completed of Job (YYYY-MM-DD): 2019-12-30

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Fit

Labor Time: 3.0

Successfully updated job with new information

Successfully added new paint-job account

Please select the number corresponding to the query: 7

Contents of Job table:
--Job ID--|--Date Commenced--|--Date Completed--|--Assembly ID--|--Process ID--|
----100 | 2019-08-09 | 2019-08-10 | 100 | 100
----101 | 2019-08-10 | 2019-08-11 | 100 | 101
----102 | 2019-10-04 | 2019-10-06 | 100 | 103
----103 | 2019-06-10 | 2019-06-20 | 102 | 103
----104 | 2019-12-11 | 2019-09-10 | 103 | 105
----105 | 2019-08-10 | 2019-08-20 | 106 | 109
----106 | 2019-06-05 | 2019-12-09 | 106 | 104
----107 | 2019-04-20 | 2019-05-20 | 105 | 102
----108 | 2019-05-23 | 2019-12-30 | 101 | 100
----109 | 2019-08-10 | null | 108 | 108

Existing Job ID of completed job: 109

Date Completed of Job (YYYY-MM-DD): 2019-12-10

What is the type of job Cut, Paint, or Fit (Cut, Paint, Fit): Fit

Labor Time: 6.7

Successfully updated job with new information

Successfully added new paint-job account
```

6.8: Screenshots of Query 8

--This query loops to ask for multiple types of accounts to update with a transaction, so multiple accounts updates in one loop in this example

```
Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100

New Transaction ID: 101

Sup Cost: 200

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 101

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Department

Existing Department_Account entries
--Department Account ID
----100
----101
----102

Existing Department_Account ID: 100

Successfully updated department account

What type of account would you like to update (Department, Assembly, Process, Quit): Department

Existing Department_Account entries
--Department Account ID
----100
----101
----102

Existing Department_Account ID: 101

Successfully updated department account

What type of account would you like to update (Department, Assembly, Process, Quit): Department

Existing Department_Account entries
--Department Account ID
----100
----101
----102

Existing Department_Account ID: 102

Successfully updated department account

What type of account would you like to update (Department, Assembly, Process, Quit): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
----104
----105
----109

Existing Assembly_Account ID: 103

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
----104
----105
----109

Existing Assembly_Account ID: 104

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
```

```
----104
----105
----109

Existing Assembly_Account ID: 105

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
----104
----105
----109

Existing Assembly_Account ID: 109

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 106

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 107

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 106

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 107

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 108

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Quit
```

```
Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100
----101

New Transaction ID: 102

Sup Cost: 400

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 102

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
----104
----105
----109

Existing Assembly_Account ID: 105

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Process

Existing Process_Account entries
--Process Account ID
----106
----107
Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 108

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Quit

(1) Enter a new customer
```

```
Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100
----101
----102

New Transaction ID: 103

Sup Cost: 200

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 109

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Department

Existing Department_Account entries
--Department Account ID
----100
----101
----102

Existing Department_Account ID: 102

Successfully updated department account

What type of account would you like to update (Department, Assembly, Process, Quit): Quit

Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100
----101
----102
----103

New Transaction ID: 104

Sup Cost: 300

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 104

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Assembly

Existing Assembly_Account entries
--Assembly Account ID
----103
----104
----105
----109

Existing Assembly_Account ID: 109

Successfully updated assembly account

What type of account would you like to update (Department, Assembly, Process, Quit): Quit
```

```
Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100
----101
----102
----103
----104

New Transaction ID: 105

Sup Cost: 90

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 105

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Proces

Please 'Department' for department account, 'Assembly' for assembly account, 'Process' for process account, or 'Quit'
Please select the number corresponding to the query: 8

Existing Transactions entries
--Transaction ID
----100
----101
----102
----103
----104
----105

New Transaction ID: 106

Sup Cost: 700

Existing Job entries
--Job ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Job ID for transaction: 107

Successfully added new transaction

What type of account(s) would you like to update (Department, Assembly, Process): Process

Existing Process_Account entries
--Process Account ID
----106
----107
----108

Existing Process_Account ID: 107

Successfully updated process account

What type of account would you like to update (Department, Assembly, Process, Quit): Quit
```

6.9: Screenshot for Query 9

```
Please select the number corresponding to the query: 9

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Existing Assembly ID: 100

Cost incurred on Assembly ID 100: 1000
Please select a number corresponding to the query: 9

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | null
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Existing Assembly ID: 102

Cost incurred on Assembly ID 102: 700

Please select the number corresponding to the query: 9

Contents of Assembly table:
--Assembly ID--|--Date Ordered--|--Assembly Details--|--Customer Name--|--Account ID--
----100 | 2019-11-19 | Mill | Sammy Najib | 105
----101 | 2019-12-11 | Mill | Trevor Bryant | 110
----102 | 2019-04-19 | Saw | Sammy Najib | 109
----103 | 2019-09-11 | Mill | Trevor Bryant | null
----104 | 2019-07-15 | Saw | Zaki Refai | null
----105 | 2019-08-17 | take | Trey Sullivent | null
----106 | 2019-09-18 | Take | Sarah Kayali | null
----107 | 2019-11-09 | Saw | Zaki Refai | null
----108 | 2019-07-10 | Pull | Trevor Bryant | null
----109 | 2019-01-10 | Mill | Zaki Refai | null

Existing Assembly ID: 101

Cost incurred on Assembly ID 101: 700
```

6.10: Screenshots for Query 10

```
Please select the number corresponding to the query: 10

Contents of Job table where date_completed is not null:
--Job ID--|--Date Completed--
----100 | 2019-08-10
----101 | 2019-08-11
----102 | 2019-10-06
----103 | 2019-06-20
----104 | 2019-09-10
----105 | 2019-08-20
----106 | 2019-12-09
----107 | 2019-05-20
----108 | 2019-12-30
----109 | 2019-12-10

Existing Date Completed: 100

Date completed does not exist, please enter query 10 again and enter Job ID from table above
Please select the number corresponding to the query: 10

Contents of Job table where date_completed is not null:
--Job ID--|--Date Completed--
----100 | 2019-08-10
----101 | 2019-08-11
----102 | 2019-10-06
----103 | 2019-06-20
----104 | 2019-09-10
----105 | 2019-08-20
----106 | 2019-12-09
----107 | 2019-05-20
----108 | 2019-12-30
----109 | 2019-12-10

Existing Date Completed: 2019-08-10

Contents of Department table:
--Department ID--|--Department Data--|--Account ID--
----100 | CS | 101
----101 | MIS | 102
----102 | AEREO | null
----103 | SPACE | null
----104 | UP | null

Department ID: 100

Total labor time on Department ID 100: 6.500
Please select the number corresponding to the query: 10

Contents of Job table where date_completed is not null:
--Job ID--|--Date Completed--
----100 | 2019-08-10
----101 | 2019-08-11
----102 | 2019-10-06
----103 | 2019-06-20
----104 | 2019-09-10
----105 | 2019-08-20
----106 | 2019-12-09
----107 | 2019-05-20
----108 | 2019-12-30
----109 | 2019-12-10

Existing Date Completed: 2019-06-20

Contents of Department table:
--Department ID--|--Department Data--|--Account ID--
----100 | CS | 101
----101 | MIS | 102
----102 | AEREO | null
----103 | SPACE | null
----104 | UP | null

Department ID: 100

Total labor time on Department ID 100: null
```

6.11: Screenshots for Query 11

```
Please select the number corresponding to the query: 11

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 100
--Process ID--|--Date Commenced--|--Department ID--
----100 | 2019-08-09 | 100
----101 | 2019-08-10 | 101
----103 | 2019-10-04 | 103

Please select the number corresponding to the query: 11

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 101
--Process ID--|--Date Commenced--|--Department ID--
----100 | 2019-05-23 | 100

Please select the number corresponding to the query: 11

Existing Assembly entries
--Assembly ID
----100
----101
----102
----103
----104
----105
----106
----107
----108
----109

Existing Assembly ID: 102
--Process ID--|--Date Commenced--|--Department ID--
----103 | 2019-06-10 | 103
```

6.12: Screenshots for Query 12

```
Please select the number corresponding to the query: 12

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Dates of completed jobs
Existing Job entries
--Date Completed
----2019-08-10
----2019-08-11
----2019-10-06
----2019-06-20
----2019-09-10
----2019-08-20
----2019-12-09
----2019-05-20
----2019-12-30
----2019-12-10

Existing Date Completed: 2019-08-11

Cut_Jobs:
--Job ID--|--Machine Type--|--Machine Time--|--Material|--Labor Time--|--Assembly ID--

Paint_Jobs:
--Job ID--|--Color--|--Volume--|--Labor Time--|--Assembly ID--

Fit_Jobs:
--Job ID--|--Labor Time--|--Assembly ID--
Please select the number corresponding to the query: 12

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 100

Dates of completed jobs
Existing Job entries
--Date Completed
----2019-08-10
----2019-08-11
----2019-10-06
----2019-06-20
----2019-09-10
----2019-08-20
----2019-12-09
----2019-05-20
----2019-12-30
----2019-12-10
----2019-12-11
----2019-12-11

Existing Date Completed: 2019-12-11

Cut_Jobs:
--Job ID--|--Machine Type--|--Machine Time--|--Material--|--Labor Time--|--Assembly ID--
----110 | Mill | 70.000 | Metal | 6.700 | 103

Paint_Jobs:
--Job ID--|--Color--|--Volume--|--Labor Time--|--Assembly ID--
----111 | Purple | 5.600 | 7.600 | 108

Fit_Jobs:
--Job ID--|--Labor Time--|--Assembly ID--
```

```
Please select the number corresponding to the query: 12

Existing Department entries
--Department ID
----100
----101
----102
----103
----104

Existing Department ID: 101

Dates of completed jobs
Existing Job entries
--Date Completed
----2019-08-10
----2019-08-11
----2019-10-06
----2019-06-20
----2019-09-10
----2019-08-20
----2019-12-09
----2019-05-20
----2019-12-30
----2019-12-10
----2019-12-11
----2019-12-11
----2019-12-11

Existing Date Completed: 2019-12-11

Cut_Jobs:
--Job ID--|--Machine Type--|--Machine Time--|--Material--|--Labor Time--|--Assembly ID--

Paint_Jobs:
--Job ID--|--Color--|--Volume--|--Labor Time--|--Assembly ID--

Fit_Jobs:
--Job ID--|--Labor Time--|--Assembly ID--
----112 | 7.600 | 107
```

6.13: Screenshots for Query 13

```
Please select the number corresponding to the query: 13

First Category Number: 1

Second Category Number: 3
--Customer Name--|--Customer Address--|--Category Number--
----Trey Sullivent | Traditions East | 2


Please select the number corresponding to the query: 13

First Category Number: 3

Second Category Number: 6
├--Customer Name--|--Customer Address--|--Category Number--
----Sammy Najib | 750 Imhoff Rd | 6
----Zaki Refai | 1063 Greystone Crest | 4

Please select the number corresponding to the query: 13

First Category Number: 8

Second Category Number: 10
├--Customer Name--|--Customer Address--|--Category Number--
----Trevor Bryant | 750 W Imhoff Rd | 8
```

6.14: Screenshots for Query 14

--Table of cut Job appears in query, no need to print sql tables after query execution

```
Please select the number corresponding to the query: 14

Contents of Cut Job table:
--Job IDs of Cut Jobs--
----104
----110

First Range Number: 100

Second Range Number: 105
Deleted Cut Jobs in range 100 and 105

Please select the number corresponding to the query: 14

Contents of Cut Job table:
--Job IDs of Cut Jobs--
----100
----104
----110

First Range Number: 100

Second Range Number: 100
Deleted Cut Jobs in range 100 and 100

Please select the number corresponding to the query: 14

Contents of Cut Job table:
--Job IDs of Cut Jobs--
----110

First Range Number: 109

Second Range Number: 111
Deleted Cut Jobs in range 109 and 111
```

6.15: Screenshots for Query 15

```
Please select the number corresponding to the query: 15

Existing Paint_Job entries
--Paint_Job ID
----101
----102
----105
----106
----107
----111

New Paint_Job ID: 101

New Color: Red

Updated Paint_Job 101 to new color
```

	job_ID	color	volume	labor_time	assembly_ID	proc_ID
1	101	Red	6.500	3.400	100	101
2	102	red	98.900	2.300	100	103
3	105	Blue	8.700	9.000	106	109
4	106	Green	4.500	3.400	106	104
5	107	Yello	3.400	5.600	105	102
6	111	Purple	5.600	7.600	108	111

```
Please select the number corresponding to the query: 15

Existing Paint_Job entries
--Paint_Job ID
----101
----102
----105
----106
----107
----111

New Paint_Job ID: 105

New Color: Yello

Updated Paint_Job 105 to new color
```

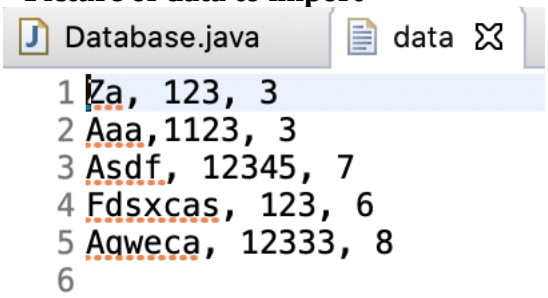
	job_ID	color	volume	labor_time	assembly_ID	proc_ID
1	101	Red	6.500	3.400	100	101
2	102	red	98.900	2.300	100	103
3	105	Yello	8.700	9.000	106	109
4	106	Green	4.500	3.400	106	104
5	107	Yello	3.400	5.600	105	102
6	111	Purple	5.600	7.600	108	111

```
Please select the number corresponding to the query: 15

Existing Paint_Job entries
--Paint_Job ID
----101
----102
----105
----106
----107
----111

New Paint_Job ID: 111

New Color: Blue

Updated Paint_Job 111 to new color
```

	job_ID	color	volume	labor_time	assembly_ID	proc_ID
1	101	Red	6.500	3.400	100	101
2	102	red	98.900	2.300	100	103
3	105	Yello	8.700	9.000	106	109
4	106	Green	4.500	3.400	106	104
5	107	Yello	3.400	5.600	105	102
6	111	Blue	5.600	7.600	108	111

6.16: Screenshots of Query 16

--Picture of data to import



--Table with customers will appear if query 1 is selected

```
Please select the number corresponding to the query: 16

Please enter the file path: data
Successfully added new customers

Please select the number corresponding to the query: 1

Existing Customer entries
--Customer Name
----Aaa
----Aqweca
----Asdf
----Fdsxcas
----Sammy Najib
----Sarah Kayali
----Trevor Bryant
----Trey Sullivent
----Za
----Zaki Refai
```

6.17: Screenshots of Query 17

```
Please select the number corresponding to the query: 17

Enter range start: 1

Enter range end: 8

Enter output file name: output

Successfully added all customers in category range to file: /Users/zakirefai/Desktop/Databases/INDIVIDUAL PROJECT/Individu
```

output

a	100	1
Aaa	1123	3
Aqweca	12333	8
Asdf	12345	7
Fdsxcas	123	6
Sammy Najib	750 Imhoff Rd	6
Sarah Kayali	1063 Greystone Crest	7
Trevor Bryant	750 W Imhoff Rd	8
Trey Sullivent	Traditions East	2
Za	123	3
Zaki Refai	1063 Greystone Crest	4

6.18: Screenshot of Query 18

```
Successful connection - Schema:dbo
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 to the 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 by adding sup-cost
(9) Retrieve the cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the departme
(12) Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given dep
(13) Retrieve the customers (in name order) whose category is in a given range
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import new customers from file
(17) Export customers to file based on category range
(18) Quit

Please select the number corresponding to the query: 18
You have exited the program
```

Task 7:

Task 7.1: Screnshots of source code

Database.java | data | DataHandler.java

```
3
4  import java.sql.Connection;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.sql.DriverManager;
8  import java.sql.PreparedStatement;
9
10 public class DataHandler {
11     private Connection conn;
12     // Azure SQL connection credentials
13     private String server = "113380402-sql-server.database.windows.net";
14     private String database = "cs-dsa-4513-sql-db";
15     private String username = "ZAKI-113380402";
16     private String password = "Sims31997^";
17     // Resulting connection string final private
18
19     String url = String.format(
20             "jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustS erverCertificate=false;hostNameInCertificate
21             server, database, username, password);
22
23     // Initialize and save the database connection
24     private void getDBConnection() throws SQLException, ClassNotFoundException {
25         if (conn != null) {
26             return;
27         }
28         Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
29         this.conn = DriverManager.getConnection(url);
30     }
31
32     public boolean addCustomer(String customerName, String customerAddress, int customerCategory)
33             throws SQLException, ClassNotFoundException {
34         getDBConnection(); // Prepare the database connection
35         // Prepare the SQL statement
36         final String sqlQuery = "INSERT INTO Customer " + "(customer_name, customer_address, customer_category)"
37                 + "VALUES " + "(?, ?, ?)";
38
39         final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
40         System.out.println();
41         System.out.println(customerName + " " + customerAddress + " " + customerCategory);
42         // Replace the '?' in the above statement with the given attribute values
43         stmt.setString(1, customerName);
44         stmt.setString(2, customerAddress);
45         stmt.setInt(3, customerCategory);
46         // Execute the query, if only one record is updated, then we indicate success by
47         // returning true
48         return stmt.executeUpdate() == 1;
49     }
50
51     // ------------------CLASS PROEJCT--------------------
52
53     // Get customers in range
54     public ResultSet getCustomersInRange(int range1, int range2) throws SQLException, ClassNotFoundException {
55         getDBConnection();
56
57         final String sqlQuery = "SELECT * FROM Customer WHERE customer_category BETWEEN ? AND ?";
58         final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
59         // Replace ? with range numbers
60         stmt.setInt(1, range1);
61         stmt.setInt(2, range2);
62
63         return stmt.executeQuery();
64     }
65 }
```

Database.java | data | DataHandler.java | add_customer_form.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add Customer</title>
</head>
<body>

    <h2>Add Customer</h2>

    <form action="add_customer.jsp">

        <table border=1>
            <tr>
                <th colspan="2">Enter Customer Data:</th>
            </tr>

            <tr>
                <td>Customer Name:</td>
                <td><div style="text-align: center;">
                        <input type=text name=customer_name>
                    </div></td>
            </tr>

            <tr>
                <td>Customer Address:</td>
                <td><div style="text-align: center;">
                        <input type=text name=customer_address>
                    </div></td>
            </tr>

            <tr>
                <td>Customer Category:</td>
                <td><div style="text-align: center;">
                        <input type=text name=customer_category>
                    </div></td>
            </tr>

            <tr>
                <td><div style="text-align: center;">
                        <input type=reset value=Clear>
                    </div></td>
                <td><div style="text-align: center;">
                        <input type=submit value=Insert>
                    </div></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

Database.java | data | DataHandler.java | add_customer_form.jsp | add_customer.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
    <%@page import="jsp_azure.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>
    <%
        // The handler is the one in charge of establishing the connection.
        DataHandler handler = new DataHandler();
        // Get the attribute values passed from the input form.
        String customer_name = request.getParameter("customer_name");
        String customer_address = request.getParameter("customer_address");
        String customer_category_string = request.getParameter("customer_category");

        System.out.println(customer_name + " " + customer_address + " " + customer_category_string);

        //Error Checking
        if (customer_name.equals("") || customer_address.equals("") || customer_category_string.equals("")) {
            response.sendRedirect("add_movie_form.jsp");
        } else {

            int customer_category = Integer.parseInt(customer_category_string);
            System.out.println(customer_category);

            boolean success = handler.addCustomer(customer_name, customer_address, customer_category);

            System.out.println(success);
            if (!success) {
    %>
    <h2>There was a problem inserting the course</h2>
    <%
        } else {
    %>
    <h2>The Customer:</h2>
    <ul>
        <li>Start Time: <%=customer_name%></li>
        <li>Movie Name: <%=customer_address%></li>
        <li>Duration: <%=customer_category_string%></li>
    </ul>
    <h2>Was successfully inserted.</h2>

    <a href="add_customer_form.jsp">Add another customer</a>
    </br>
    <a href="get_customer_range_form.jsp">See all customers in category
        range</a>
    <%
        }
        }
    %>
</body>
</html>
```

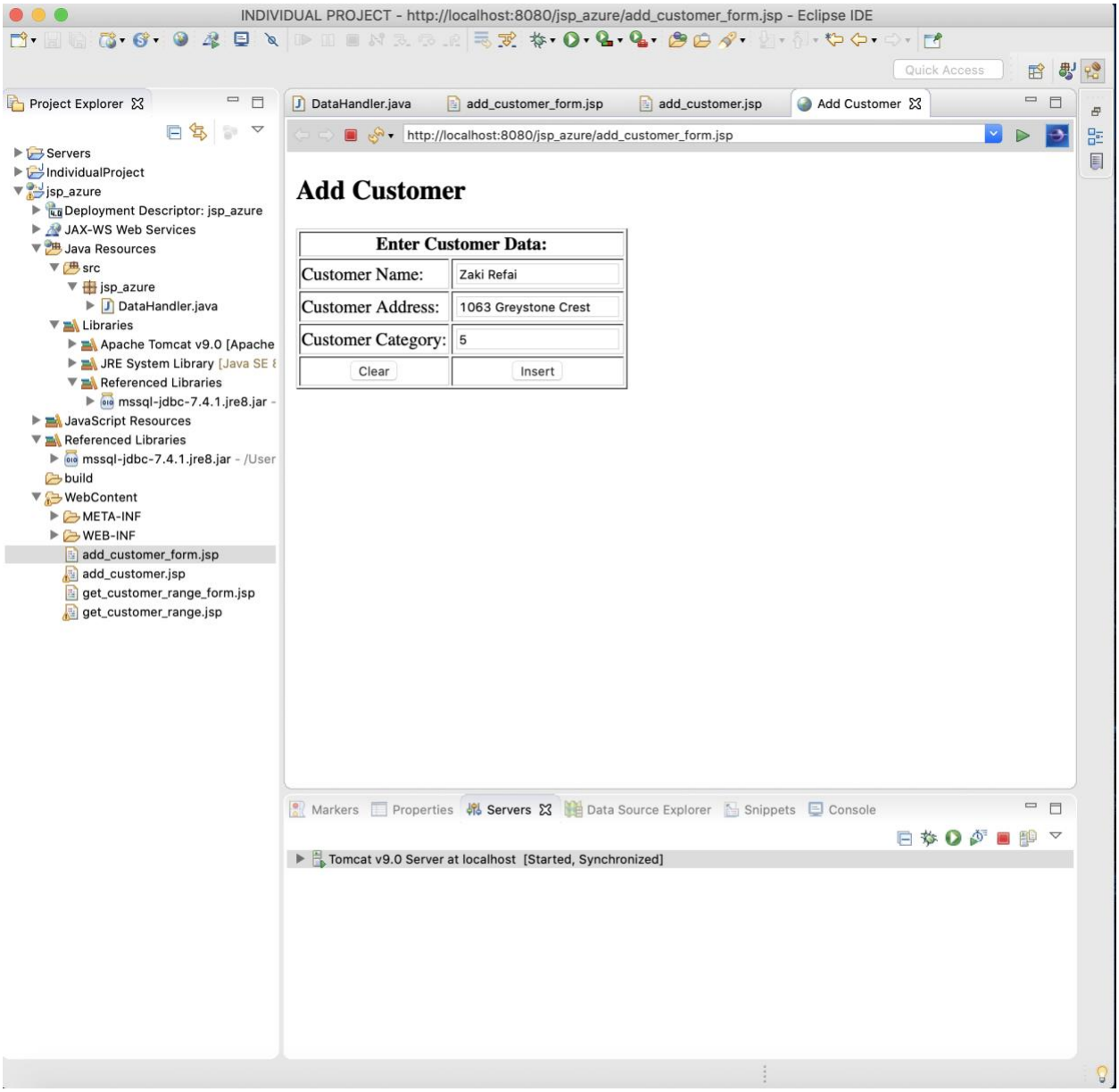
add_customer.jsp | get_customer_range_form.jsp

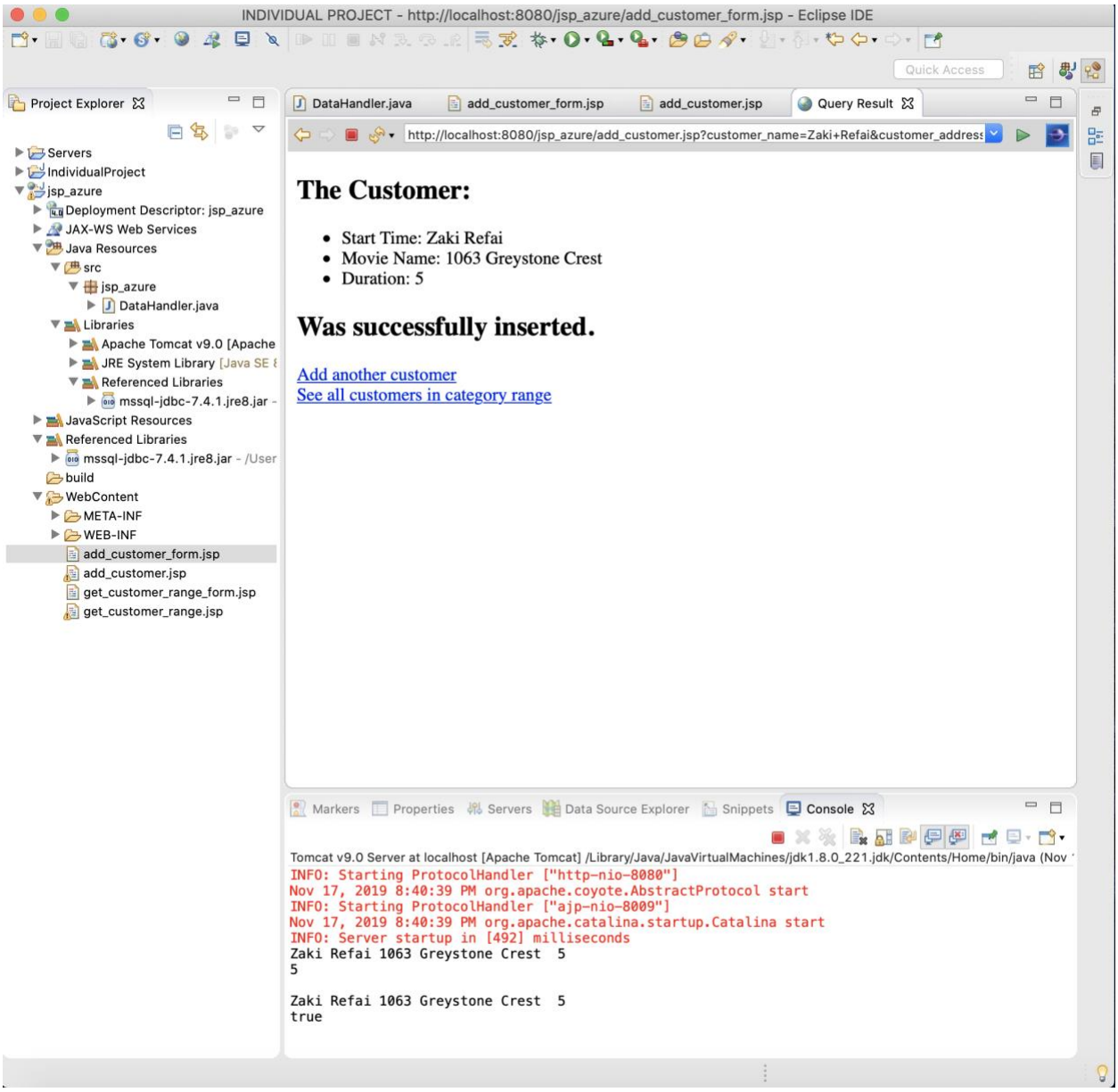
```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Get Customer Category Range</title>
</head>
<body>

    <h2>Get Customer Category Range</h2>

    <form action="get_customer_range.jsp">
        <table border=1>
            <tr>
                <th colspan="2">Enter Customer Category Ranges:</th>
            </tr>

            <tr>
                <td>Customer Range Start:</td>
                <td><div style="text-align: center;">
                        <input type=text name=range_1>
                    </div></td>

            </tr>

            <tr>
                <td>Customer Range End:</td>
                <td><div style="text-align: center;">
                        <input type=text name=range_2>
                    </div></td>

            </tr>

            <tr>
                <td><div style="text-align: center;">
                        <input type=reset value=Clear>
                    </div></td>
                <td><div style="text-align: center;">
                        <input type=submit value=Insert>
                    </div></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

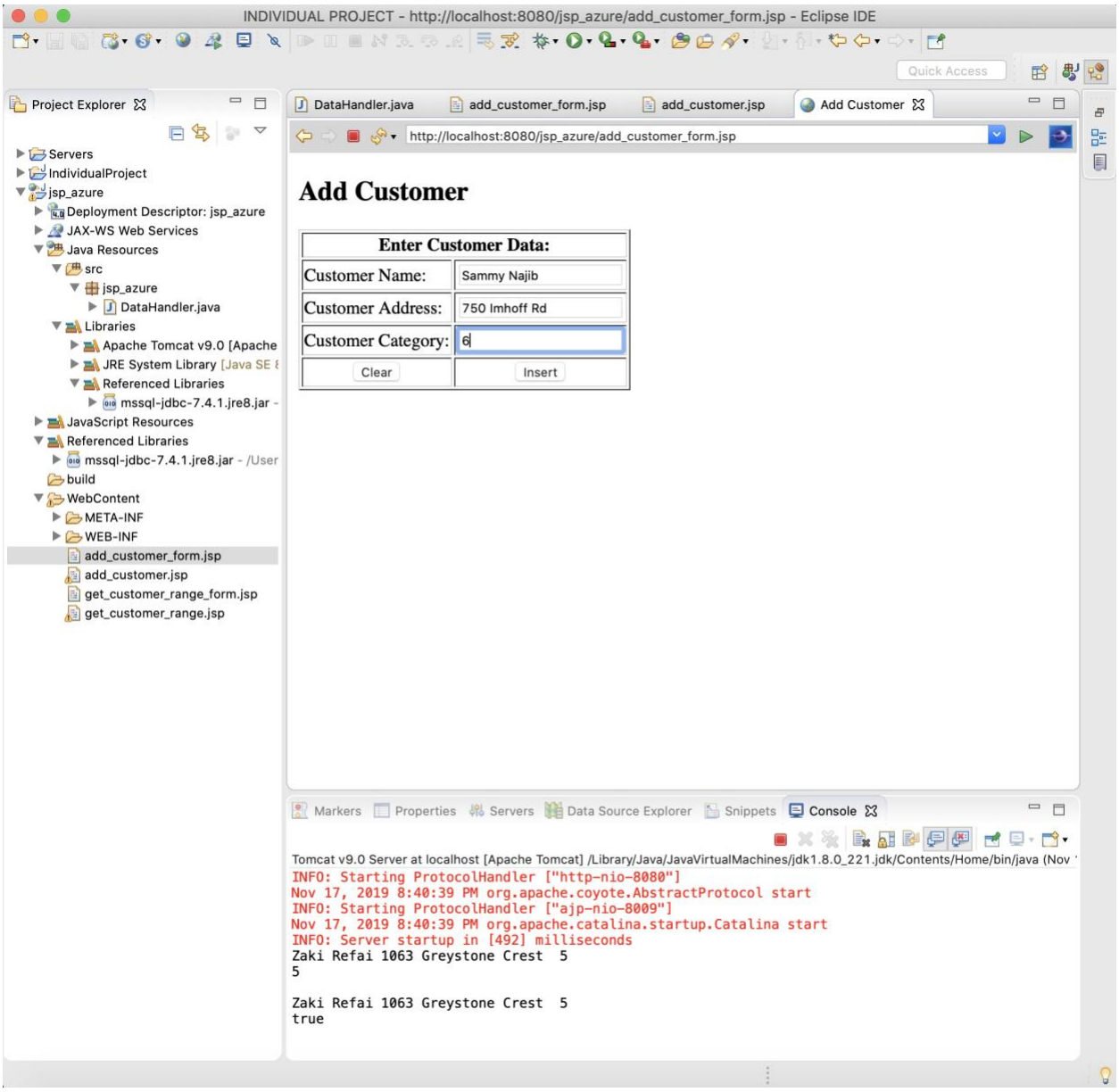
add_customer.jsp | get_customer_range_form.jsp | get_customer_range.jsp

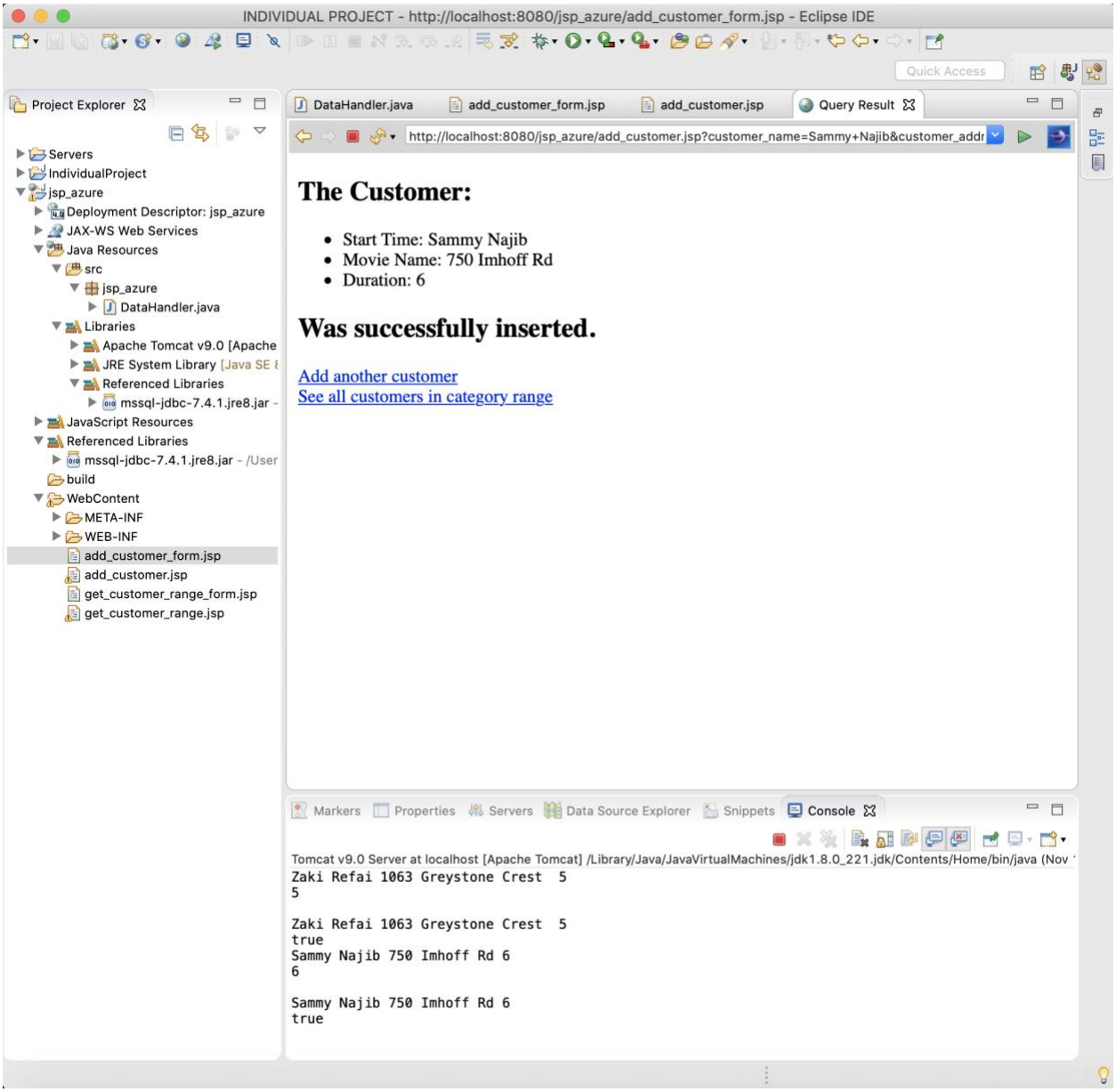
```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Movie Nights</title>
</head>
<body>
    <%@page import="jsp_azure.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
        final DataHandler handler = new DataHandler();

        String range1 = request.getParameter("range_1");
        String range2 = request.getParameter("range_2");

        if (range1.equals("") || range2.equals("")) {
            response.sendRedirect("get_customer_range_form.jsp");
        } else {
            int range_1 = Integer.parseInt(range1);
            int range_2 = Integer.parseInt(range2);

            final ResultSet customers = handler.getCustomersInRange(range_1, range_2);
    %>
    <table cellspacing="2" cellpadding="2" border="1">
        <tr>
            <!-- The table headers row -->
            <td align="center">
                <h4>Time</h4>
            </td>
            <td align="center">
                <h4>Movie Name</h4>
            </td>
            <td align="center">
                <h4>Duration</h4>
            </td>
        </tr>
        <%
            while (customers.next()) { // For each movie_night record returned...
                    // Extract the attribute values for every row returned
                    final String customer_name = customers.getString("customer_name");
                    final String customer_address = customers.getString("customer_address");
                    final String customer_category = customers.getString("customer_category");

                    out.println("<tr>");
                    out.println("<td align=\"center\">" + customer_name + "</td><td align=\"center\"> "
                            + customer_address + "</td><td align=\"center\"> " + customer_category + "</td>");
                    out.println("</tr>");
                }

            }
            //final ResultSet movies = handler.getAllMovies();
        %>
    </table>
</body>
</html>
```

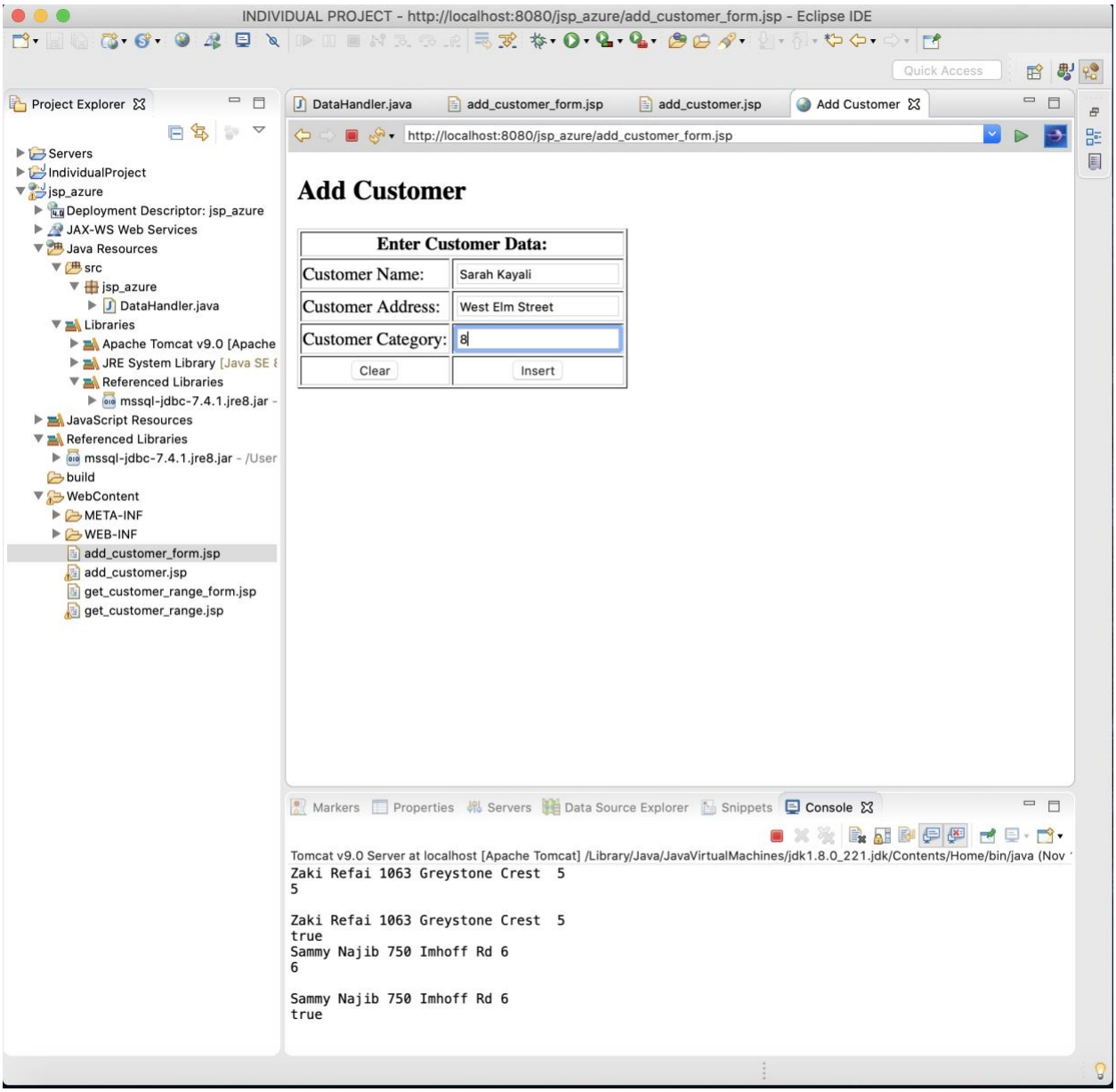
Task 7.2: Screnshots of Web Application working

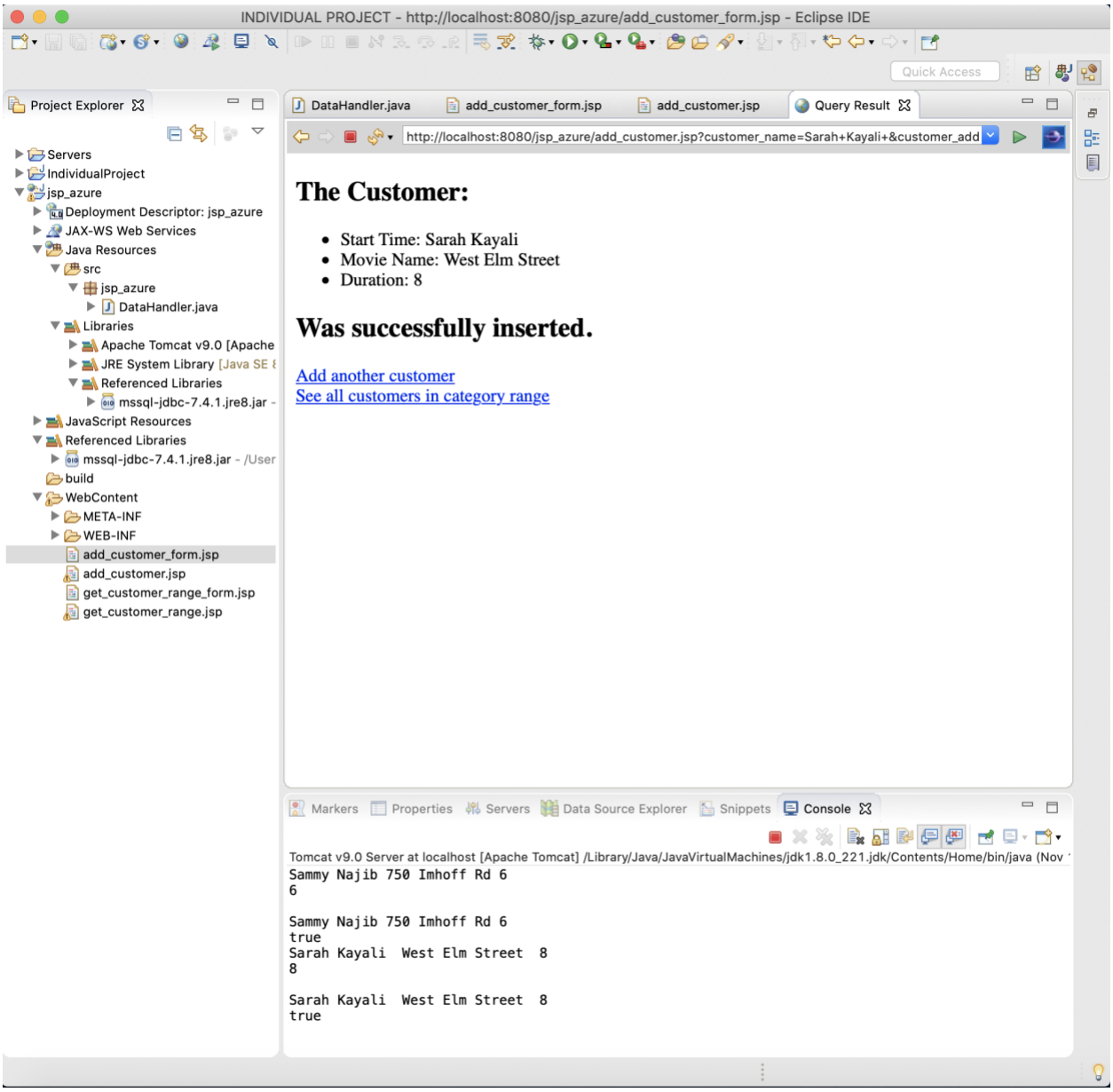


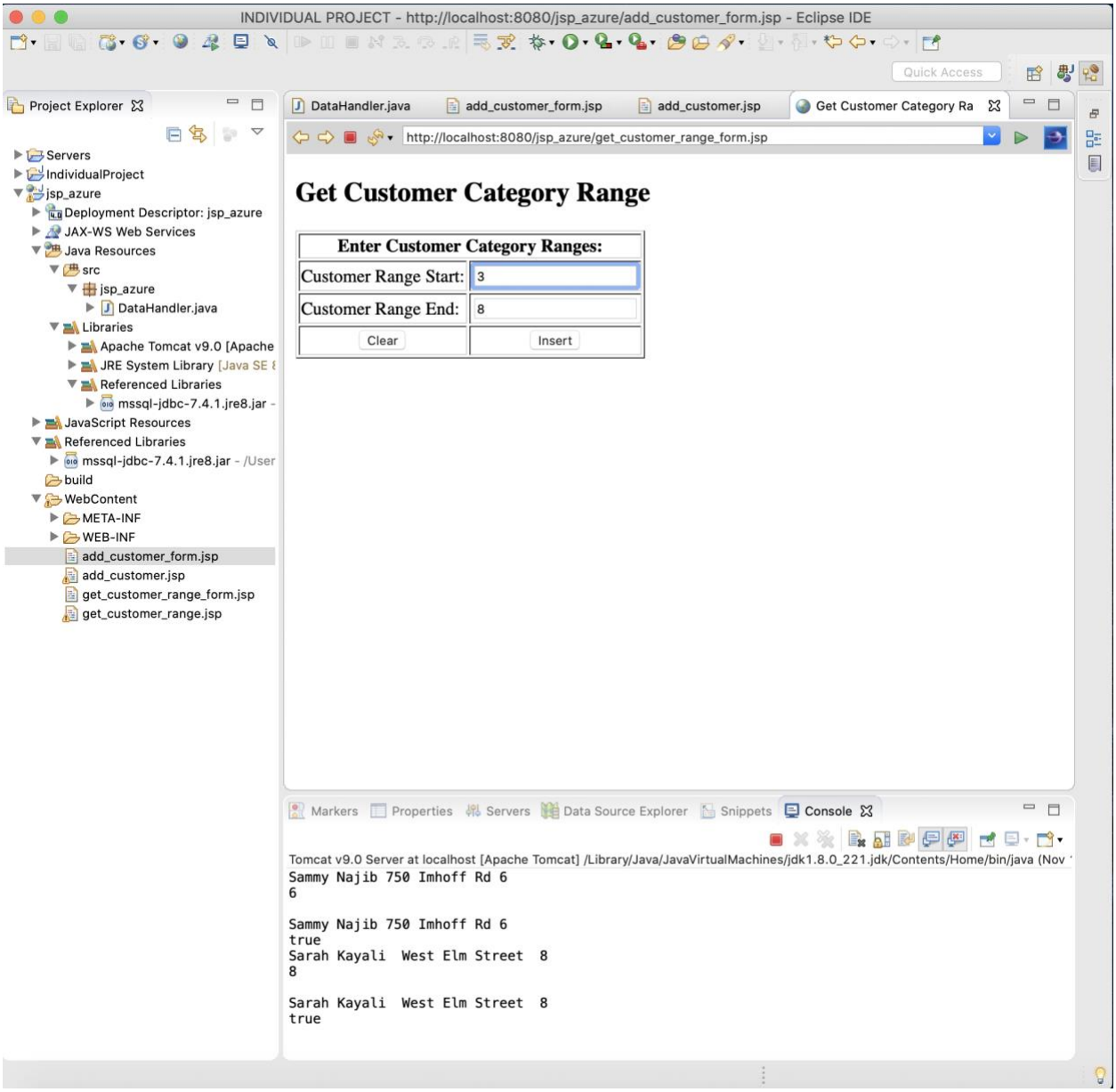

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Query Result
http://localhost:8080/jsp_azure/add_customer.jsp?customer_name=Zaki+Refai&customer_address
The Customer:
Start Time: Zaki Refai
Movie Name: 1063 Greystone Crest
Duration: 5
Was successfully inserted.
Add another customer
See all customers in category range
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 17, 2019 8:40:39 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Nov 17, 2019 8:40:39 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [492] milliseconds
Zaki Refai 1063 Greystone Crest 5
5
Zaki Refai 1063 Greystone Crest 5
true

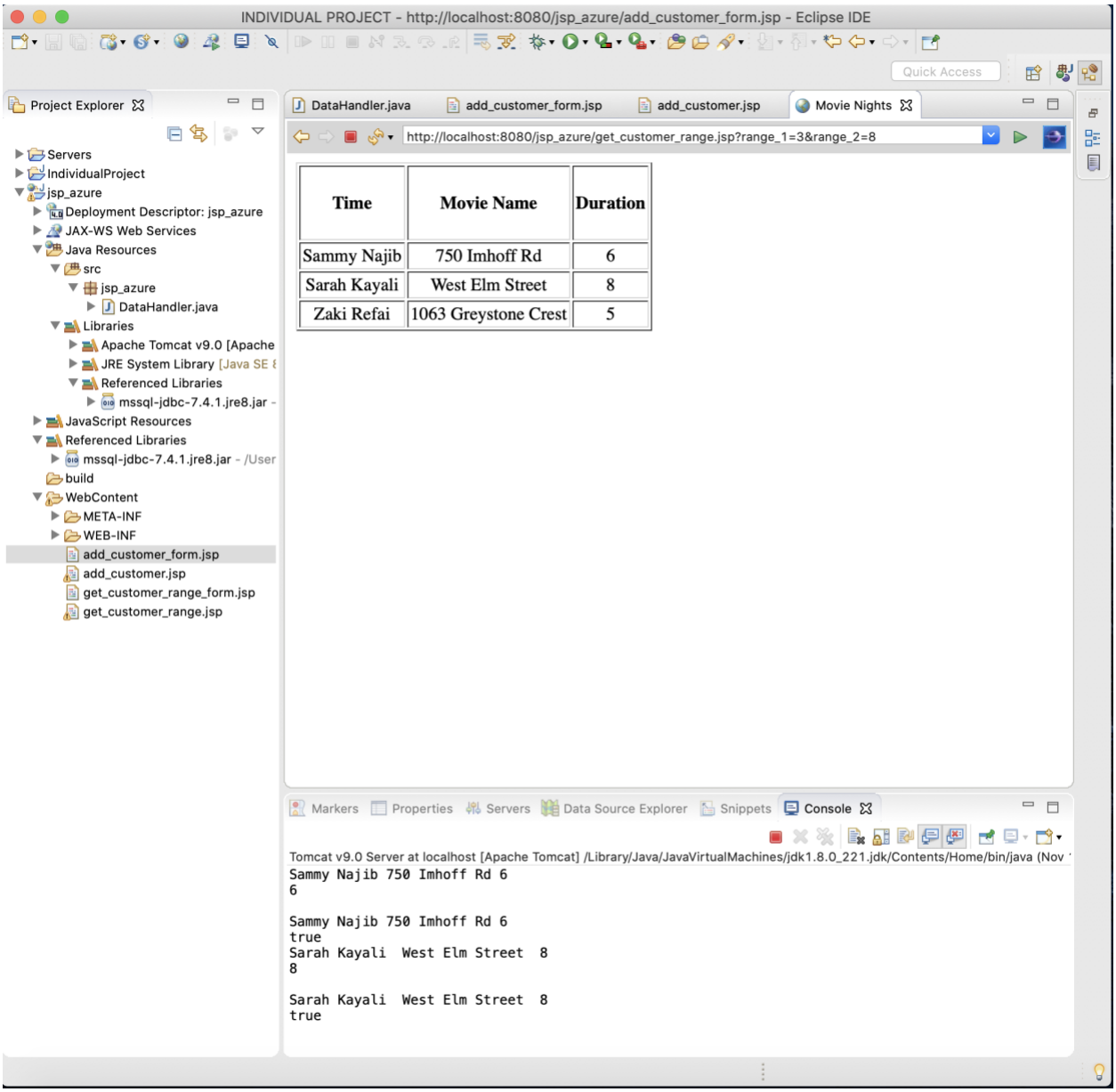

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Add Customer
http://localhost:8080/jsp_azure/add_customer_form.jsp
Add Customer
Enter Customer Data:
Customer Name:
Sammy Najib
Customer Address:
750 Imhoff Rd
Customer Category:
6
Clear
Insert
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 17, 2019 8:40:39 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Nov 17, 2019 8:40:39 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [492] milliseconds
Zaki Refai 1063 Greystone Crest 5
5
Zaki Refai 1063 Greystone Crest 5
true

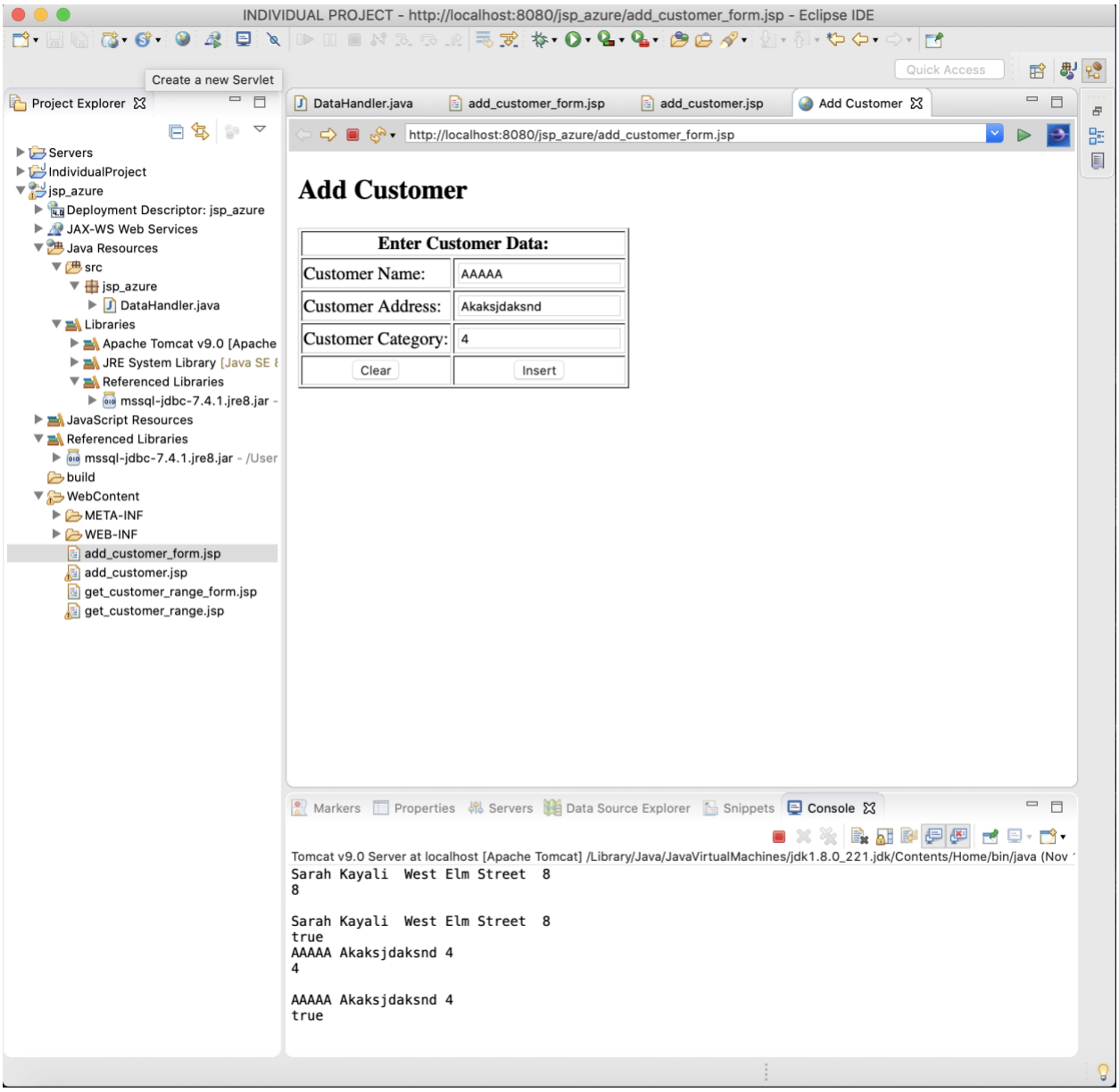

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Query Result
http://localhost:8080/jsp_azure/add_customer.jsp?customer_name=Sammy+Najib&customer_addr
The Customer:
Start Time: Sammy Najib
Movie Name: 750 Imhoff Rd
Duration: 6
Was successfully inserted.
Add another customer
See all customers in category range
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Zaki Refai 1063 Greystone Crest 5
5
Zaki Refai 1063 Greystone Crest 5
true
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true

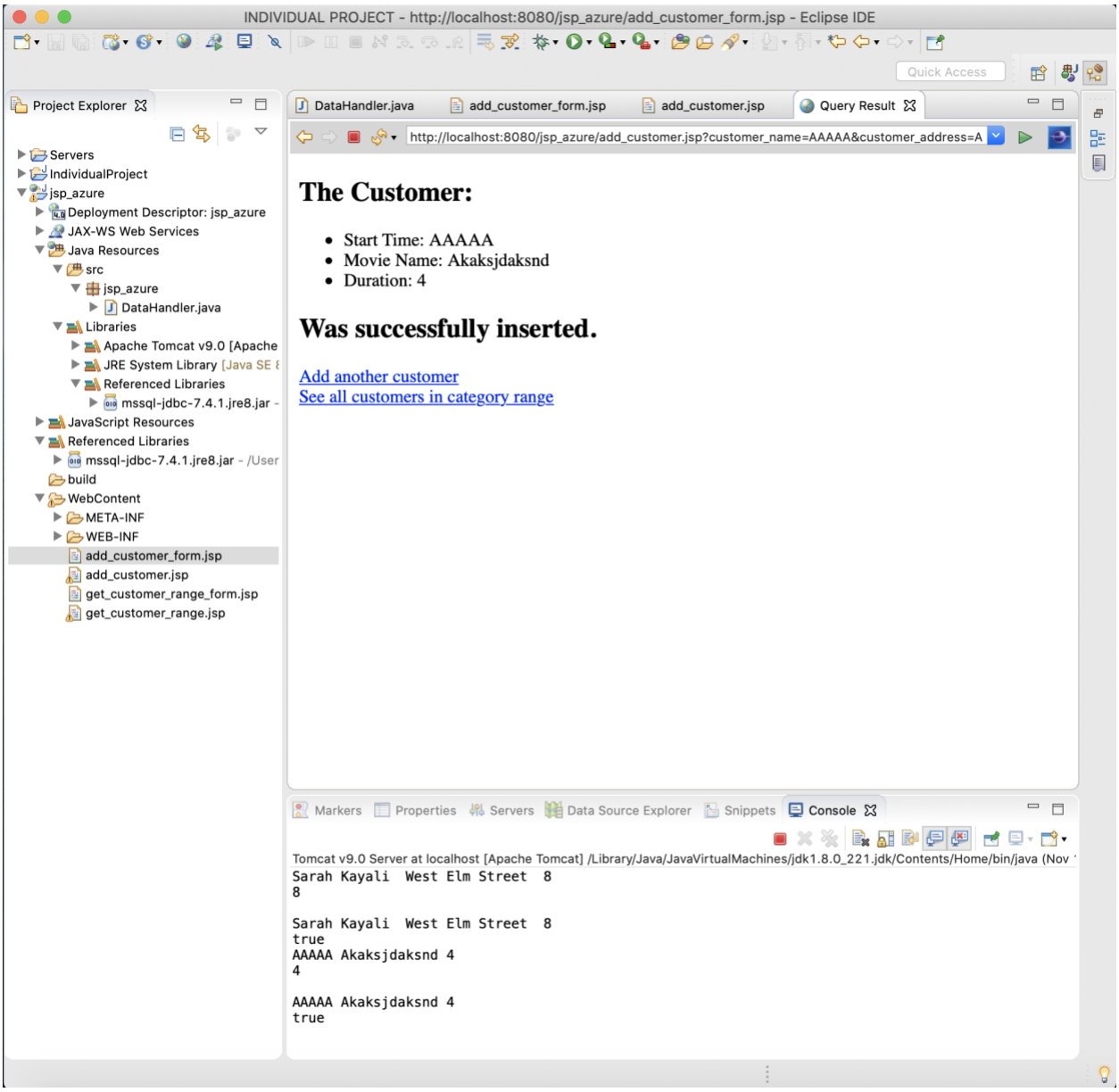

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Add Customer
http://localhost:8080/jsp_azure/add_customer_form.jsp
Add Customer
Enter Customer Data:
Customer Name:
Sarah Kayali
Customer Address:
West Elm Street
Customer Category:
8
Clear
Insert
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Zaki Refai 1063 Greystone Crest 5
5
Zaki Refai 1063 Greystone Crest 5
true
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true

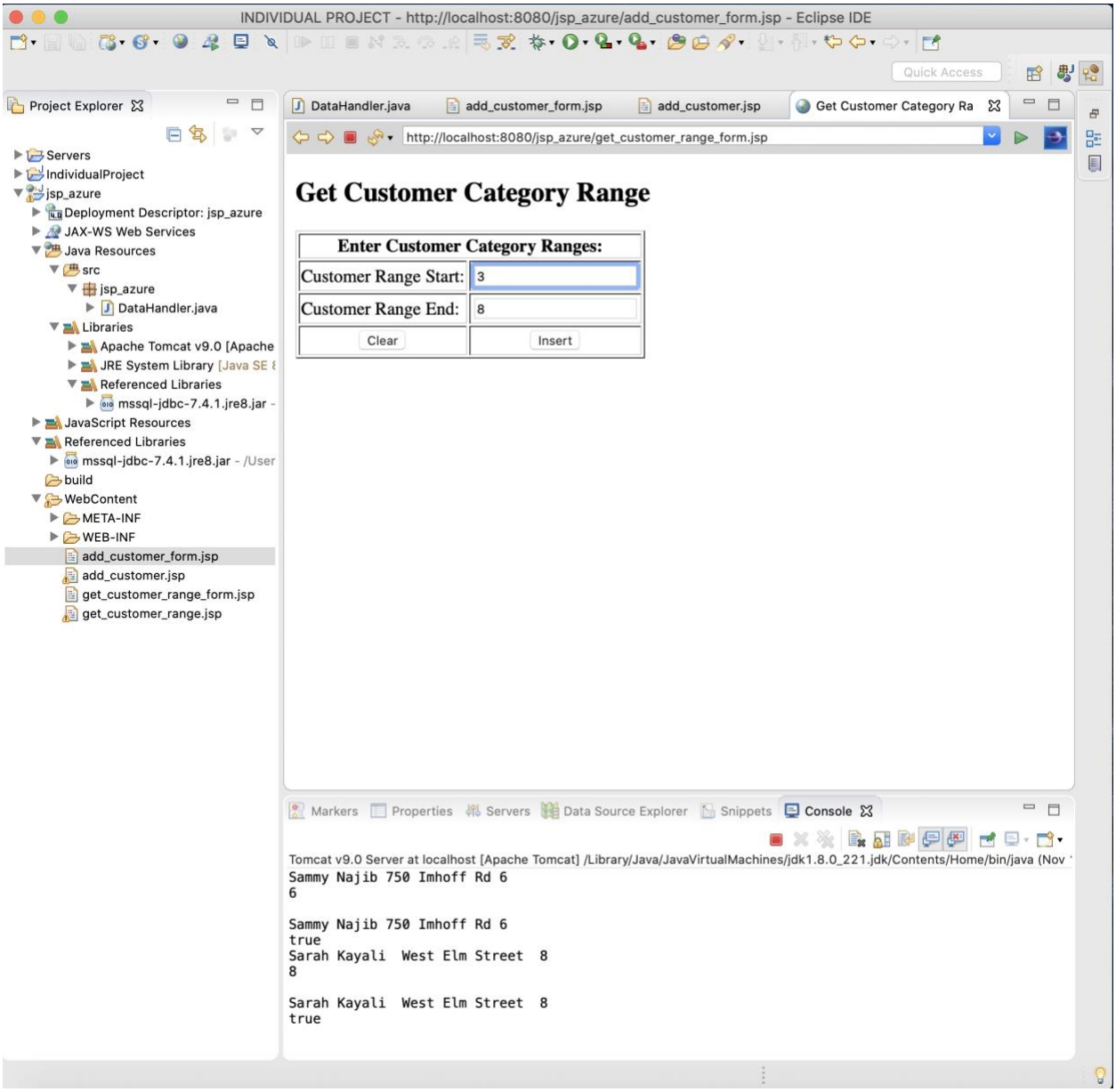

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Query Result
http://localhost:8080/jsp_azure/add_customer.jsp?customer_name=Sarah+Kayali+&customer_add
The Customer:
Start Time: Sarah Kayali
Movie Name: West Elm Street
Duration: 8
Was successfully inserted.
Add another customer
See all customers in category range
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true

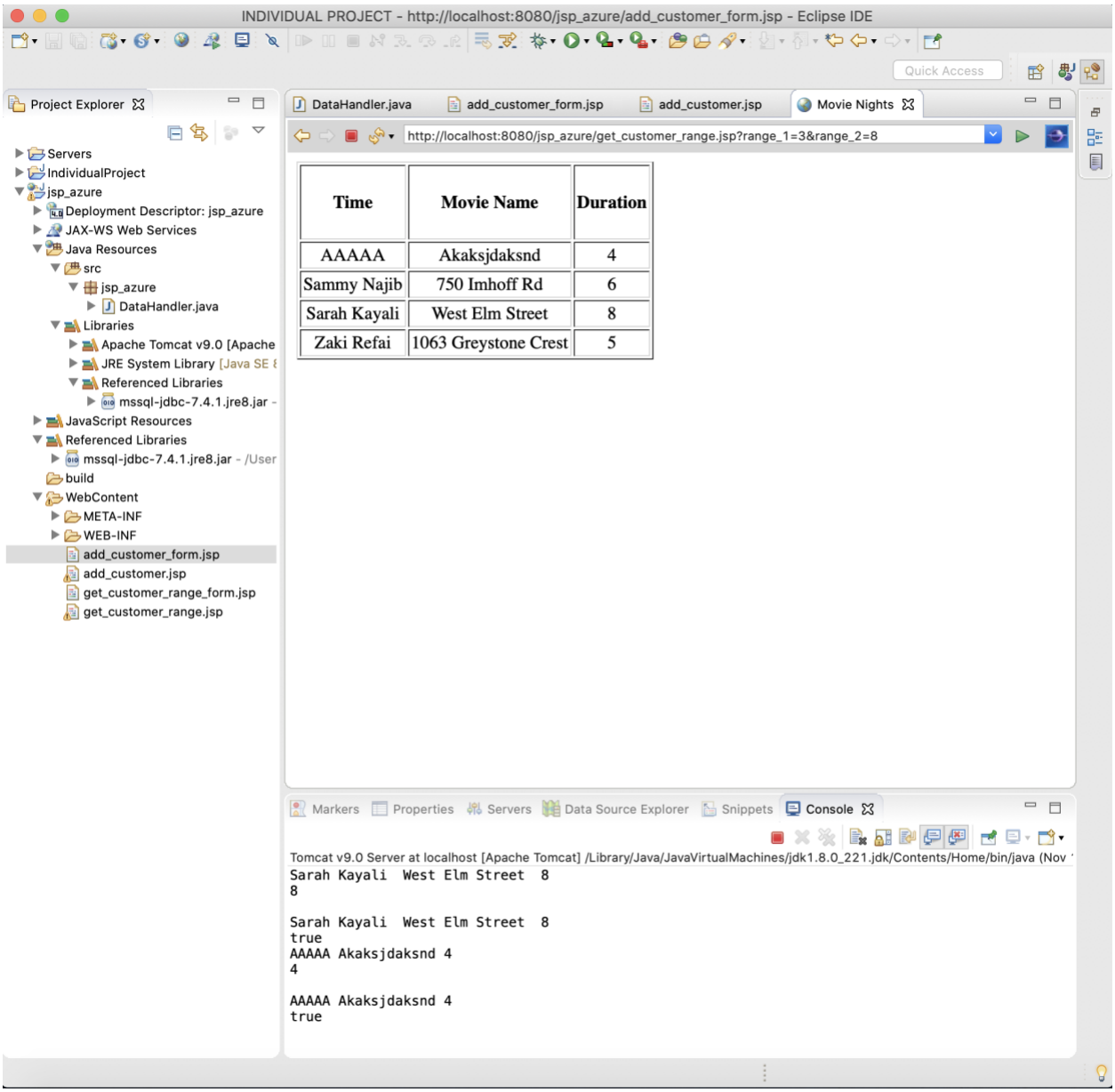

INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Get Customer Category Ra
http://localhost:8080/jsp_azure/get_customer_range_form.jsp
Get Customer Category Range
Enter Customer Category Ranges:
Customer Range Start:
3
Customer Range End:
8
Clear
Insert
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true


INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Movie Nights
http://localhost:8080/jsp_azure/get_customer_range.jsp?range_1=3&range_2=8
Time
Movie Name
Duration
Sammy Najib
750 Imhoff Rd
6
Sarah Kayali
West Elm Street
8
Zaki Refai
1063 Greystone Crest
5
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true


INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Create a new Servlet
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Add Customer
http://localhost:8080/jsp_azure/add_customer_form.jsp
Add Customer
Enter Customer Data:
Customer Name:
AAAAA
Customer Address:
Akaksjdaksnd
Customer Category:
4
Clear
Insert
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true
AAAAA Akaksjdaksnd 4
4
AAAAA Akaksjdaksnd 4
true


INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Query Result
http://localhost:8080/jsp_azure/add_customer.jsp?customer_name=AAAAA&customer_address=A
The Customer:
Start Time: AAAAA
Movie Name: Akaksjdaksnd
Duration: 4
Was successfully inserted.
Add another customer
See all customers in category range
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true
AAAAA Akaksjdaksnd 4
4
AAAAA Akaksjdaksnd 4
true


INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Get Customer Category Ra
http://localhost:8080/jsp_azure/get_customer_range_form.jsp
Get Customer Category Range
Enter Customer Category Ranges:
Customer Range Start: 3
Customer Range End: 8
Clear
Insert
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sammy Najib 750 Imhoff Rd 6
6
Sammy Najib 750 Imhoff Rd 6
true
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true


INDIVIDUAL PROJECT - http://localhost:8080/jsp_azure/add_customer_form.jsp - Eclipse IDE
Quick Access
Project Explorer
Servers
IndividualProject
jsp_azure
Deployment Descriptor: jsp_azure
JAX-WS Web Services
Java Resources
src
jsp_azure
DataHandler.java
Libraries
Apache Tomcat v9.0 [Apache
JRE System Library [Java SE
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar -
JavaScript Resources
Referenced Libraries
mssql-jdbc-7.4.1.jre8.jar - /User
build
WebContent
META-INF
WEB-INF
add_customer_form.jsp
add_customer.jsp
get_customer_range_form.jsp
get_customer_range.jsp
DataHandler.java
add_customer_form.jsp
add_customer.jsp
Movie Nights
http://localhost:8080/jsp_azure/get_customer_range.jsp?range_1=3&range_2=8
Time
Movie Name
Duration
AAAAA
Akaksjdaksnd
4
Sammy Najib
750 Imhoff Rd
6
Sarah Kayali
West Elm Street
8
Zaki Refai
1063 Greystone Crest
5
Markers
Properties
Servers
Data Source Explorer
Snippets
Console
Tomcat v9.0 Server at localhost [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java (Nov
Sarah Kayali West Elm Street 8
8
Sarah Kayali West Elm Street 8
true
AAAAA Akaksjdaksnd 4
4
AAAAA Akaksjdaksnd 4
true