

Individual Project

A JOB-SHOP ACCOUNTING DATABASE SYSTEM

Course: Database Management Systems (CS 4513)

Semester: Fall-2019

Instructor: Dr. Le Gruenwald

Author: Venkata Subbaraju Sagi

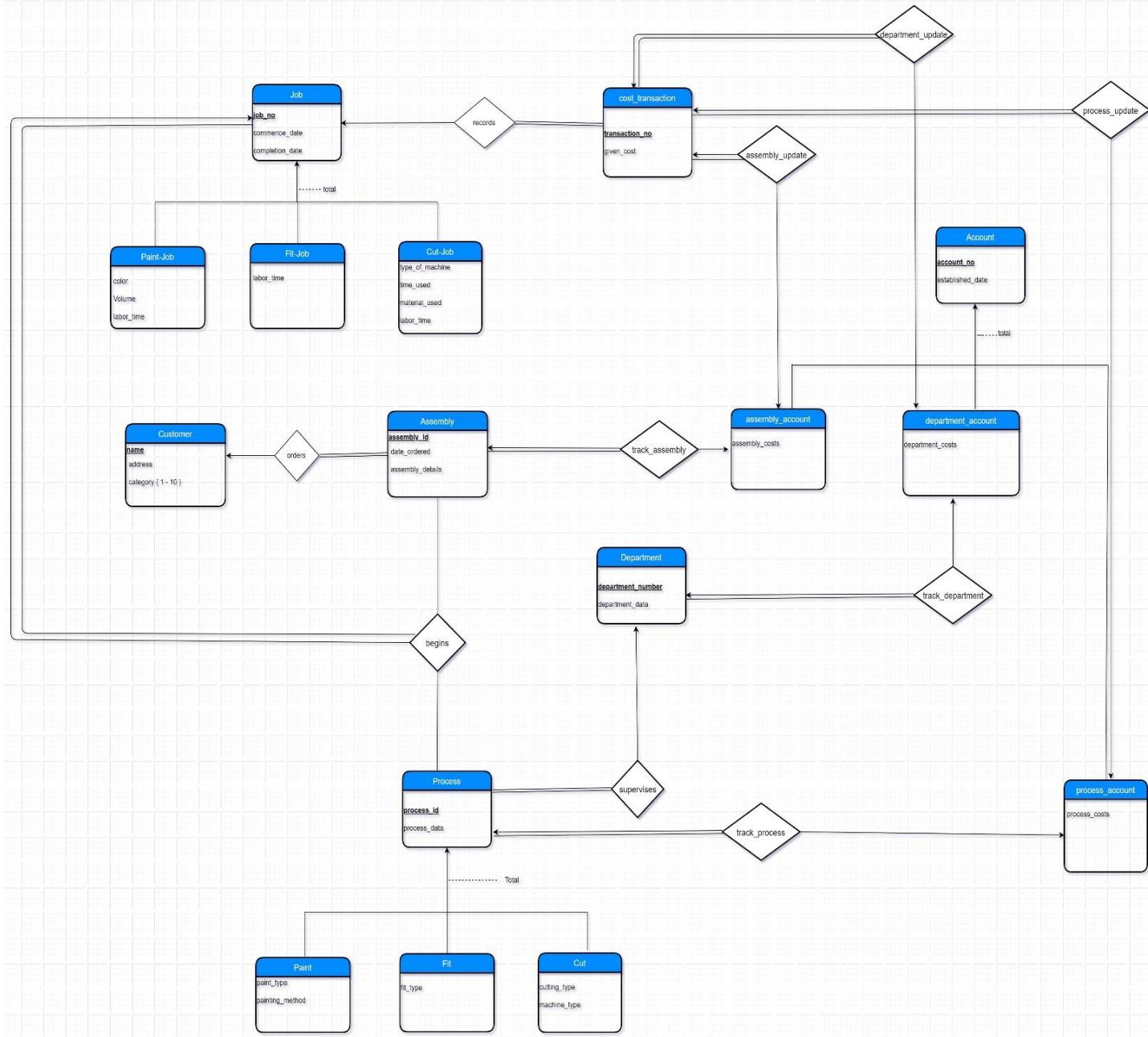
OU ID: 113472902

Email: subbaraju.v@ou.edu

TABLE OF CONTENTS

Tasks Performed	Page Numbers
Task 1	3-4
1.1. ER Diagram	3-3
1.2. Relational Database Schema	4-4
Task 2 Data Dictionary	5-8
Task 3	9-12
3.1. Discussion of storage structures for tables	9-12
3.2. Discussion of storage structures for tables (Azure SQL Database)	12-12
Task 4	13-20
SQL and text files showing the creation of tables in Azure SQL Database	
Task 5	21-43
The Java Source Program and Screenshots Showing Its successful compilation	
Task 6. Java program Execution	44-114
6.1. Screenshots showing the testing of query 1	44-47
6.2. Screenshots showing the testing of query 2	48-51
6.3. Screenshots showing the testing of query 3	52-57
6.4. Screenshots showing the testing of query 4	58-64
6.5. Screenshots showing the testing of query 5	65-71
6.6. Screenshots showing the testing of query 6	72-77
6.7. Screenshots showing the testing of query 7	78-84
6.8. Screenshots showing the testing of query 8	85-90
6.9. Screenshots showing the testing of query 9	91-92
6.10. Screenshots showing the testing of query 10	93-94
6.11. Screenshots showing the testing of query 11	95-97
6.12. Screenshots showing the testing of query 12	99-98
6.13. Screenshots showing the testing of query 13	100-101
6.14. Screenshots showing the testing of query 14	102-104
6.15. Screenshots showing the testing of query 15	105-107
6.16. Screenshots showing the testing of the import and export	108-111
6.17. Screenshots showing the testing of errors	112-113
6.18. Screenshots showing the testing of the quit option	114
Task 7. Web Database Application and its execution	115-126
7.1 web database application source program and Screenshots showing its successful compilation.	115-123
7.2 Screenshots showing the testing of Web DB Application	124-127

1.1 E-R-Diagram



1.2 Relational Database Schema

Relational database to represent the job-shop Accounting database defined

Customer(name,address,category(1-10))

Assembly(assembly_id, date_ordered, assembly_details, name,account_no)

Department(department number, department_data, account_no)

Process(process id,process_data,department_number,account_no)

paint_process(process id,paint_type,painting_method)

Fit_process(process id,fit_type)

Cut_process(process id,cutting_type,machine_type)

Job(job no, commence_date, completion_date, assembly_id, process_id)

Paint_job(paint job no,color,volume,labor_time)

Fit_job(fit job no,labor_time)

Cut_job(cut job no,type_of_machine,time_used,material_used,labor_time)

Cost_transactions(transaction no, job_no,given_cost, asse_account_no,dep_account_no, pro_account_no)

Assembly_account(account no, established_date,assembly_costs)

Department_account(account no, established_date,department_costs)

Process_account(account no, established_date, process_costs)

Assigned_job(job no, commence_date, completion_date, assembly_id, date_ordered, assembly_details, process id, process_data)

Task 2: Data Dictionary

Table Name	Attribute Name	Type	Size(max length)	Constraints
Customer	name	varchar	22 bytes	Primary Key
Customer	address	varchar	42 bytes	
Customer	category	int	4 bytes	Values will be an integer number from 1-10

Table Name	Attribute Name	Type	Size(max length)	Constraints
Assembly_account	account_no	int	4 bytes	Primary Key
Assembly_account	established_date	date	3 bytes	YYYY-MM-DD
Assembly_account	assembly_costs	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Department_account	account_no	int	4 bytes	Primary Key
Department_account	established_date	date	3 bytes	YYYY-MM-DD
Department_account	department_costs	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Process_account	account_no	int	4 bytes	Primary Key
Process_account	established_date	date	3 bytes	YYYY-MM-DD
Process_account	process_costs	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Assembly	assembly_id	int	4 bytes	Primary Key
Assembly	date_ordered	Date	3 bytes	YYYY-MM-DD
Assembly	assembly_details	varchar	100	
Assembly	c_name	varchar	20	Foreign Key
Assembly	account_no	int	4 bytes	Foreign Key DEFAULT NULL

Table Name	Attribute Name	Type	Size(max length)	Constraints
Department	department_number	int	4 bytes	Primary Key
Department	department_data	varchar	102 bytes	
Department	account_no	int	4 bytes	Foreign Key DEFAULT NULL

Table Name	Attribute Name	Type	Size(max length)	Constraints
Process	Process_id	int	4 bytes	Primary Key
Process	process_data	varchar	102 bytes	
Process	department_number	int	4 bytes	Foreign Key
Process	account_no	int	4 bytes	Foreign Key DEFAULT NULL

Table Name	Attribute Name	Type	Size(max length)	Constraints
paint_process	Process_id	int	4 bytes	Primary Key Foreign Key
paint_process	paint_type	varchar	22 bytes	
paint_process	painting_method	varchar	22 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Fit_process	Process_id	int	4 bytes	Primary Key Foreign Key
Fit_process	fit_type	varchar	22 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Cut_process	Process_id	int	4 bytes	Primary Key Foreign Key
Cut_process	cutting_type	varchar	22 bytes	
Cut_process	machine_type	varchar	22 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Job	job_no	int	4 bytes	Primary Key
Job	commence_date	Date	3 bytes	YYYY-MM-DD
Job	completion_date	Date	3 bytes	YYYY-MM-DD DEFAULT NULL
Job	assembly_id	int	4 bytes	FOREIGN KEY
Job	process_id	int	4 bytes	FOREIGN KEY

Table Name	Attribute Name	Type	Size(max length)	Constraints
Paint_job	Paint_job_no	int	4 bytes	Primary Key Foreign Key
Paint_job	color	varchar	22 bytes	
Paint_job	volume	Float	8 bytes	
Paint_job	labor_time	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Fit_job	Fit_job_no	int	4 bytes	Primary Key Foreign Key
Fit_job	labor_time	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Cut_job	Cut_job_no	int	4 bytes	Primary Key Foreign Key
Cut_job	type_of_machine	varchar	22 bytes	
Cut_job	time_used	Float	8 bytes	
Cut_job	material_used	varchar	22 bytes	
Cut_job	labor_time	Float	8 bytes	

Table Name	Attribute Name	Type	Size(max length)	Constraints
Cost_transactions	transaction_no	int	4 bytes	Primary Key
Cost_transactions	job_no	int	4 bytes	FOREIGN KEY
Cost_transactions	given_cost	Float	8 bytes	
Cost_transactions	asse_account_no	int	4 bytes	FOREIGN KEY DEFAULT NULL
Cost_transactions	dep_account_no	int	4 bytes	FOREIGN KEY DEFAULT NULL
Cost_transactions	pro_account_no	int	4 bytes	FOREIGN KEY DEFAULT NULL

Task 3

3.1: Discussion of storage structures for tables

Indexing-Storage Structure:

Table Name	Query # and Type	Search Key	Query Frequency	Selected File Organization	Justifications
customer	1.Insertion 13.Range Search	category	30/day 100/day	B+ Tree index with search key on category	B+ tree is good for range search
assembly	3. Insertion 5. Random Search 8.Random Search	assembly_id assembly_id	40/day 10/day 50/day	Extendable Hashing with hash key on assembly_id	Since we have random search on assembly_id which has almost same frequency as insertion we don't want our performance to degrade as the file grows, so I chose expendable hashing
department	2. Insertion 5.Random Search 8.Random Search	department_number department_number	Infrequent 10/day 50/day	Extendable Hashing with hashkey on department_number	Since My insertion is not Frequent, I need my Indexing only on the Random search Key, Since Extendable hashing is good for random Search , I plan to use that.

process	4. Insertion 5.Random Search 8.Random Search 8.Random Search 11.Random Search 12.Random Search	process_id department_number process_id process_id process_id	Infrequent 10/day 50/day 50/day 100/day 20/day	Extendable Hashing with hashkey on process_id	My search key process_id is being used more frequent to other search key and all my searches are random, so I choose to use exendable hashing as it is good for Random Search
paint_processes	4. Insertion		Infrequent	Heap File organization	Since I have only Insertion, which is not frequent for my table, I would like to use Heap file Organization.
Fit_process	4. Insertion		Infrequent	Heap File organization	Since I have only Insertion, which is not frequent for my table, I would like to use Heap file Organization.
Cut_process	4. Insertion		Infrequent	Heap File organization	Since I have only Insertion, which is not frequent for my table, I would like to use Heap file Organization.
Assembly_account	5. Insertion 8.Random Search (Update)	account_no	10/day 50/day	Extendable hashing with hash key on account_no	I need my Indexing only on the search Key, Since Extendable hashing is good for random Search , I plan to use that.

department_account	5. Insertion 8.Random Search (Update)	account_no	10/day 50/day	Extendable hashing with hash key on	I have a case where Both my search keys are occurring with same frequency, I choose Extendable hashing on department_number
Process_account	5. Insertion 8.Random Search (Update)	account_no	10/day 50/day	Extendable hashing with hash key on account_no	I have a case where Both my search keys are occurring with same frequency, I choose Extendable hashing on account_number.
Job	6. Insertion 7.Random Search 8.Random Search 10.Random Search 10.Random Search 11.Random Search 12.Random search 14. Range Search	Job_no Job_no Completion_date Process_id Assembly_id Completion_date Job_no	50/day 50/day 50/day 20/day 20/day 100/day 20/day 1/month	Dynamic Extendable Hashing with hashkey on assembly_id	As I have Multiple requirement for search key, I choose to have my indexing on assembly_id purely based on its frequency.
Paint_job	7.Insertion 10.Random Search 12.Random search 15.Random Search(Update)	Paint_Job_no Paint_Job_no Paint_Job_no	50/day 20/day 20/day 1/week	Dynamic Extendable Hashing with hashkey on Paint_Job_no	I need my Indexing only on the search Key, Since Extendable hashing is good for random Search , I plan to use that.

Fit_job	7.Insertion 10.Random Search 12.Random search	Fit_Job_no Fit_Job_no	50/day 20/day 20/day	Dynamic Extendable Hashing with hashkey on Job_no	I need my Indexing only on the search Key, Since Extendable hashing is good for random Search , I plan to use that.
Cut_job	7.Insertion 10.Random Search 12.Random search 14. Range Search	Cut_Job_no Cut_Job_no Cut_Job_no	50/day 20/day 20/day 1/month	B+ tree Index on Job_no	B+ tree Indexing as it works good for both random and range search
Cost_transactions	8. Insertion		50/day	Index Sequential File	

3.2. Discussion of storage structures for tables (Azure SQL Database)

Even though I plan to use the above-mentioned Storage structures, Azure SQL Database automatically creates an Index on my Primary Key which I feel is helpful for my queries. Most of my search keys are primary keys in the table, so I will use the same indexing which Azure already provides me and will create new index only on the search keys which are not primary Keys.

```
CREATE INDEX Assembly_job ON JOB(assembly_id);
```

I have created a secondary Index on assembly_id for the JOB as My choice of Extendable hashing was not possible on this table as I already a Primary key which is clustered, so I have used sequential index instead for creating the above Index.

And again, I will be using the index provided by Azure to retrieve my queries, which might not be efficient, but as far as my consideration, It works decent and I have only created a new Index to retrieve my queries faster.

Task 4

Creating Customer Table:

SQL Statement:

```
CREATE TABLE customer
(
    name VARCHAR(20) PRIMARY KEY,
    address VARCHAR(40),
    category INT
    CONSTRAINT CHK_category_value CHECK (category IN (1,2,3,4,5,6,7,8,9,10))
);
```

The screenshot shows the SQL Server Management Studio (SSMS) interface. In the center pane, there is a code editor window containing the SQL script for creating a table named 'customer'. The code is as follows:

```
CREATE TABLE customer
(
    name VARCHAR(20) PRIMARY KEY,
    address VARCHAR(40),
    category INT
    CONSTRAINT CHK_category_value CHECK (category IN (1,2,3,4,5,6,7,8,9,10))
);
```

The code editor has several tabs at the top: 'ID_project_tables.sql - sagi0000...gi0000' (selected), 'DROP_TABLES.sql - sagi0000...gi0000', 'Procedures.sql - disconnected', and 'Index.sql - disconnected'. Below the tabs, there are buttons for 'Run', 'Cancel', 'Disconnect', 'Change Connection' (set to 'cs-dsa-4513-sql-db'), 'Explain', and 'Enable SQLCMD'. The status bar at the bottom of the code editor shows the path 'C: > Users > dvsnv > OneDrive > Documents > ID_project_tables.sql'.

Below the code editor is a 'Messages' pane. It displays the following log entries:

- 6:14:03 PM Started executing query at Line 1
- Commands completed successfully.
- Total execution time: 00:00:00.040

Creating Assembly_account Table:

SQL:

```
CREATE TABLE assembly_account
(
    account_no INT PRIMARY KEY,
    established_date DATE,
    assembly_costs FLOAT,
);
```



The screenshot shows the SQL Server Management Studio interface. In the center pane, the code for creating the assembly_account table is displayed. The code is as follows:

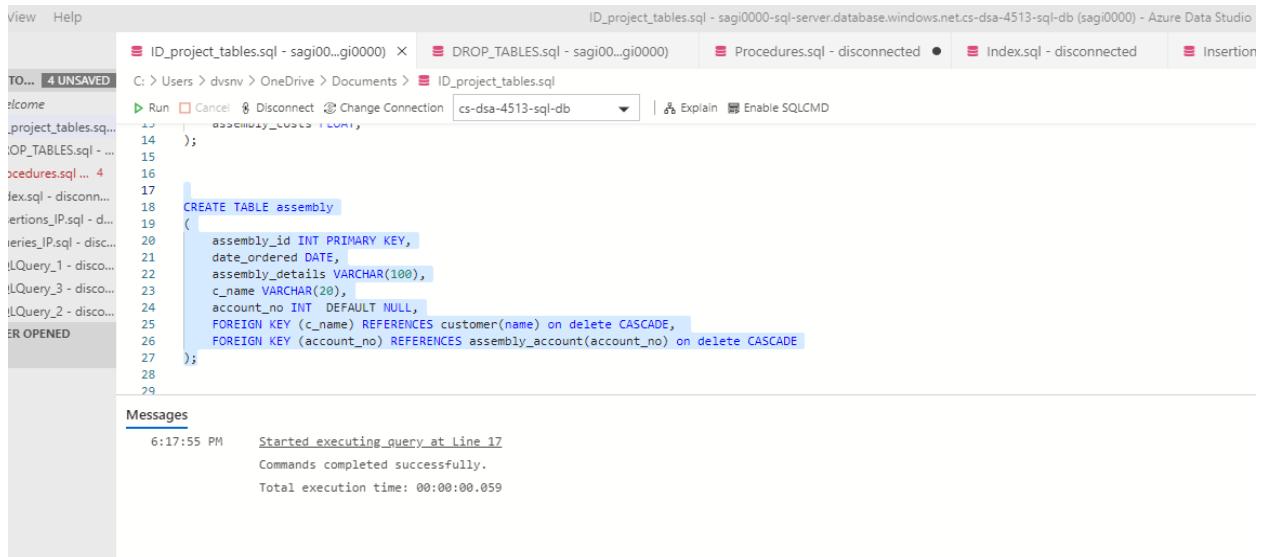
```
CREATE TABLE assembly_account
(
    account_no INT PRIMARY KEY,
    established_date DATE,
    assembly_costs FLOAT,
);
```

In the bottom pane, the 'Messages' section shows the execution results:

```
Started executing query at Line 9
Commands completed successfully.
Total execution time: 00:00:00.041
```

Creating assembly Table:

```
CREATE TABLE assembly
(
    assembly_id INT PRIMARY KEY,
    date_ordered DATE,
    assembly_details VARCHAR(100),
    c_name VARCHAR(20),
    account_no INT DEFAULT NULL,
    FOREIGN KEY (c_name) REFERENCES customer(name) on delete CASCADE,
    FOREIGN KEY (account_no) REFERENCES assembly_account(account_no) on delete CASCADE
);
```



The screenshot shows the Azure Data Studio interface. In the center pane, the code for creating the assembly table is displayed. The code is as follows:

```
CREATE TABLE assembly
(
    assembly_id INT PRIMARY KEY,
    date_ordered DATE,
    assembly_details VARCHAR(100),
    c_name VARCHAR(20),
    account_no INT DEFAULT NULL,
    FOREIGN KEY (c_name) REFERENCES customer(name) on delete CASCADE,
    FOREIGN KEY (account_no) REFERENCES assembly_account(account_no) on delete CASCADE
);
```

In the bottom pane, the 'Messages' section shows the execution results:

```
Started executing query at Line 17
Commands completed successfully.
Total execution time: 00:00:00.059
```

Creating depart_account Table:

```
CREATE TABLE department_account
(
    account_no INT PRIMARY KEY,
    established_date DATE,
    department_costs FLOAT,
);
```



Azure Data Studio interface showing the creation of the `depart_account` table. The code editor contains the `CREATE TABLE` statement. The status bar at the bottom shows the command was started at 6:24:22 PM and completed successfully.

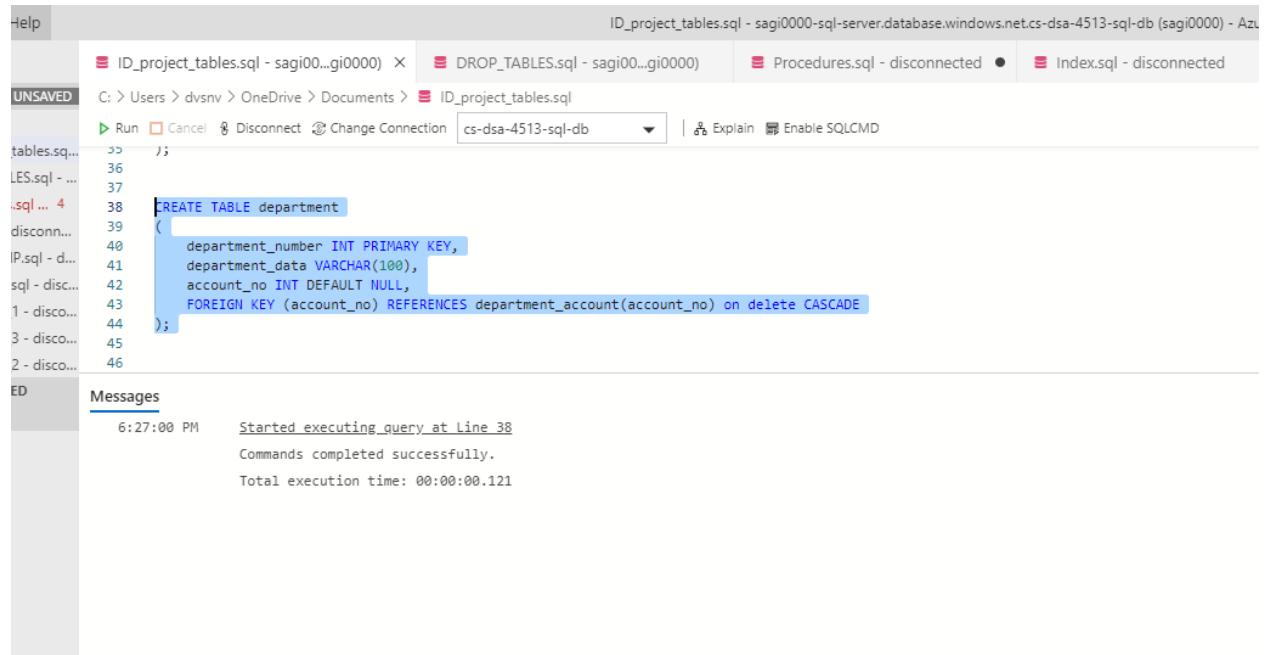
```
23     c_name VARCHAR(20),
24     account_no INT DEFAULT NULL,
25     FOREIGN KEY (c_name) REFERENCES customer(name) on delete CASCADE,
26     FOREIGN KEY (account_no) REFERENCES assembly_account(account_no) on delete CASCADE
27 );
28
29 CREATE TABLE department_account
30 (
31     account_no INT PRIMARY KEY,
32     established_date DATE,
33     department_costs FLOAT,
34 );
35
36
37
38 CREATE TABLE department
```

Messages

```
6:24:22 PM Started executing_query_at Line_38
Commands completed successfully.
Total execution time: 00:00:00.040
```

Creating department Table:

```
CREATE TABLE department
(
    department_number INT PRIMARY KEY,
    department_data VARCHAR(100),
    account_no INT DEFAULT NULL,
    FOREIGN KEY (account_no) REFERENCES department_account(account_no) on delete CASCADE
);
```



Azure Data Studio interface showing the creation of the `department` table. The code editor contains the `CREATE TABLE` statement. The status bar at the bottom shows the command was started at 6:27:00 PM and completed successfully.

```
35 );
36
37
38 CREATE TABLE department
39 (
40     department_number INT PRIMARY KEY,
41     department_data VARCHAR(100),
42     account_no INT DEFAULT NULL,
43     FOREIGN KEY (account_no) REFERENCES department_account(account_no) on delete CASCADE
44 );
45
46
```

Messages

```
6:27:00 PM Started executing_query_at Line_38
Commands completed successfully.
Total execution time: 00:00:00.121
```

Creating process_account Table:

```
CREATE TABLE process_account
(
    account_no INT PRIMARY KEY,
    established_date DATE,
    process_costs FLOAT,
);
```

The screenshot shows the SQL Server Management Studio interface. In the center pane, the code for creating the `process_account` table is displayed. The code is as follows:

```
CREATE TABLE process_account
(
    account_no INT PRIMARY KEY,
    established_date DATE,
    process_costs FLOAT,
);
```

In the bottom pane, the execution results are shown:

- Started executing query at Line 38
- Commands completed successfully.
- Total execution time: 00:00:00.121

Creating process Table:

```
CREATE TABLE process
(
    process_id INT PRIMARY KEY,
    department_number INT,
    process_data VARCHAR(100),
    account_no INT DEFAULT NULL,
    FOREIGN KEY (department_number) REFERENCES department(department_number) on delete CASCADE
,
    FOREIGN KEY (account_no) REFERENCES process_account(account_no) on delete CASCADE
);
;
```

The screenshot shows the SQL Server Management Studio interface. In the center pane, the code for creating the `process` table is displayed. The code is as follows:

```
CREATE TABLE process
(
    process_id INT PRIMARY KEY,
    department_number INT,
    process_data VARCHAR(100),
    account_no INT DEFAULT NULL,
    FOREIGN KEY (department_number) REFERENCES department(department_number) on delete CASCADE,
    FOREIGN KEY (account_no) REFERENCES process_account(account_no) on delete CASCADE
);
;
```

In the bottom pane, the execution results are shown:

- Started executing query at Line 56
- Commands completed successfully.
- Total execution time: 00:00:00.086

Creating paint_process, cut_process, fit_process Tables:

```
CREATE TABLE paint_process
(
    process_id INT PRIMARY KEY,
    paint_type VARCHAR(20),
    painting_method VARCHAR(20),
    FOREIGN KEY (process_id) REFERENCES process(process_id) on delete CASCADE
);

CREATE TABLE fit_process
(
    process_id INT PRIMARY KEY,
    fit_type VARCHAR(20),
    FOREIGN KEY (process_id) REFERENCES process(process_id) on delete CASCADE
);

CREATE TABLE cut_process
(
    process_id INT PRIMARY KEY,
    cutting_type VARCHAR(20),
    machine_type VARCHAR(20)
    FOREIGN KEY (process_id) REFERENCES process(process_id) on delete CASCADE
);
```

The screenshot shows the Azure Data Studio interface with the following details:

- Title Bar:** ID_project_tables.sql - sagi0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (sagi0000) - Azure Data Studio
- Toolbar:** Run, Cancel, Disconnect, Change Connection, cs-dsa-4513-sql-db, Explain, Enable SQLCMD.
- Code Editor:** Displays the three CREATE TABLE statements for paint_process, fit_process, and cut_process.
- Messages Panel:** Shows the execution log:
 - Started executing query at Line 67
 - Commands completed successfully.
 - Total execution time: 00:00:00.058

Creating job, paint_job, fit_job, cut_job Tables:

```
CREATE TABLE job
(
    job_no INT PRIMARY KEY,
    commence_date DATE,
    completion_date DATE DEFAULT NULL,
    assembly_id INT,
    process_id INT,
    FOREIGN KEY (assembly_id) REFERENCES assembly(assembly_id) on delete CASCADE,
    FOREIGN KEY (process_id) REFERENCES process (process_id) on delete CASCADE
);

CREATE TABLE paint_job
(
    paint_job_no INT PRIMARY KEY,
    color VARCHAR(20),
    volume FLOAT,
    labor_time FLOAT
    FOREIGN KEY (paint_job_no) REFERENCES job(job_no) on delete CASCADE
);

CREATE TABLE fit_job
(
    fit_job_no INT PRIMARY KEY,
    labor_time FLOAT
    FOREIGN KEY (fit_job_no) REFERENCES job(job_no) on delete CASCADE
);

CREATE TABLE cut_job
(
    cut_job_no INT PRIMARY KEY,
    type_of_machine VARCHAR(20),
    time_used FLOAT,
    material_used VARCHAR(20),
    labor_time FLOAT
    FOREIGN KEY (cut_job_no) REFERENCES job(job_no) on delete CASCADE
);
```

Help

ID_project_tables.sql - sagi000...gi0000) • DROP_TABLES.sql - sagi000...gi0000) Procedures.sql - disconnected Index.sql - disconnected Insertions

C:\> Users > dsvn > OneDrive > Documents > ID_project_tables.sql

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

ct_ta... 9+
ABLES.sql - ...
res.sql ... 4
l - disconn...
is_IP.sql - d...
JP.sql - disc...
ry_1 - disco...
ry_2 - disco...
ENED

```
90 CREATE TABLE job
91 (
92     job_no INT PRIMARY KEY,
93     commence_date DATE,
94     completion_date DATE DEFAULT NULL,
95     assembly_id INT,
96     process_id INT,
97     FOREIGN KEY (assembly_id) REFERENCES assembly(assembly_id) on delete CASCADE,
98     FOREIGN KEY (process_id) REFERENCES process (process_id) on delete CASCADE
99 );
100
101 CREATE TABLE paint_job
102 (
103     paint_job_no INT PRIMARY KEY,
104     color VARCHAR(20),
105     volume FLOAT,
106     labor_time FLOAT
107     FOREIGN KEY (paint_job_no) REFERENCES job(job_no) on delete CASCADE
108 );
109
110 CREATE TABLE fit_job
111 (
112     fit_job_no INT PRIMARY KEY,
113     labor_time FLOAT
114     FOREIGN KEY (fit_job_no) REFERENCES job(job_no) on delete CASCADE
115 );
116
117
118 CREATE TABLE cut_job
119 (
120     cut_job_no INT PRIMARY KEY,
121     type_of_machine VARCHAR(20),
122     time_used FLOAT,
123     material_used VARCHAR(20),
124     labor_time FLOAT
125     FOREIGN KEY (cut_job_no) REFERENCES job(job_no) on delete CASCADE
126 );
127
128
```

Messages

6:50:44 PM Started executing query at Line 90

Commands completed successfully.

Total execution time: 00:00:00.067

Creating Cost_transactions Table:

```
CREATE TABLE cost_transactions
(
    transaction_no INT PRIMARY KEY,
    job_no INT UNIQUE,
    sup_cost FLOAT,
    asse_account_no INT DEFAULT NULL,
    dep_account_no INT DEFAULT NULL,
    pro_account_no INT DEFAULT NULL,
    FOREIGN KEY (job_no) REFERENCES job(job_no) ,
    FOREIGN KEY (asse_account_no) REFERENCES assembly_account(account_no) ,
    FOREIGN KEY (dep_account_no) REFERENCES department_account(account_no),
    FOREIGN KEY (pro_account_no) REFERENCES process_account(account_no)
);
```

The screenshot shows the Azure Data Studio interface with the following details:

- Top Bar:** Shows the current database as "cs-dsa-4513-sql-db".
- Left Sidebar:** Lists several open files: "ct_ta... 9+", "TABLES.sql - ...", "res.sql ... 4", and "ENED".
- Main Editor Area:** Displays the SQL code for creating the "cost_transactions" table. The code is highlighted with syntax coloring.
- Toolbar:** Includes buttons for "Run", "Cancel", "Disconnect", "Change Connection", "Explain", and "Enable SQLCMD".
- Messages Panel:** Shows the execution log:
 - Started executing query at Line 129
 - Commands completed successfully.
 - Total execution time: 00:00:00.065

Task 5

The Java Source program

```
//Importing the necessary Packages
import java.util.Scanner;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.BufferedWriter;

public class individual_project{
    public static void main(String[] args) throws SQLException, IOException {
        // Connect to database
        final String hostName = "sagi0000-sql-server.database.windows.net";
        final String dbName = "cs-dsa-4513-sql-db";
        final String user = "sagi0000";
        final String password = "Avinash8&65";
        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);

        try (final Connection connection = DriverManager.getConnection(url)) {

            final String schema = connection.getSchema();
            Statement statement = connection.createStatement();
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Successful connection to Job-Shop Accounting DataBase:" + schema);
            System.out.println("WELCOME TO Job-Shop Accounting DataBase");

            // initializing Scanner object
            Scanner myScan = new Scanner(System.in);
            int Choice = 0;
            while(Choice != 18)
```

```

{
// printing out the available choices for the User to choose.
// It is assumed that user will have prior knowledge of the problem
System.out.println("=====You have the Following Options to Choose=====");
System.out.println("1. Enter a new customer");
System.out.println("2. Enter a new department ");
System.out.println("3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered");
System.out.println("4. Enter a new process-id and its department together with its type and information relevant to the type");
System.out.println("5. Create a new account and associate it with the process, assembly, or department to which it is applicable ");
System.out.println("6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced");
System.out.println("7. At the completion of a job, enter the date it completed and the information relevant to the type of job ");
System.out.println("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");
System.out.println("9. Retrieve the cost incurred on an assembly-id");
System.out.println("10. Retrieve the total labor time within a department for jobs completed in the department during a given date ");
System.out.println("11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process ");
System.out.println("12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department ");
System.out.println("13. Retrieve the customers (in name order) whose category is in a given range ");
System.out.println("14. Delete all cut-jobs whose job-no is in a given range ");
System.out.println("15. Change the color of a given paint job ");
System.out.println("16. Import: enter new customers from a data file until the file is empty");
System.out.println("17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen ");
System.out.println("18. QUIT");
System.out.println("=====");

//reading the input given by the user
Choice = myScan.nextInt();

//Depending on the choice made by the user, we are defining the operations to be done

if (Choice == 1) {
    //Enter a new customer in to the Data Base

    //asking user to enter the customer name and reading the value
}

```

```

System.out.println("Enter the customer Name ");
String name = br.readLine();

//reading the customer's address from the user
System.out.println("Enter the customer's address ");
String address = br.readLine();

//reading the customer's category value given by user
System.out.println("Enter the customer's category- values from 1 to 10");
int category = Integer.parseInt(br.readLine());

//Executing the query 1

final String query1 = "INSERT INTO customer values(" + name + "','" + address + "','" + category + ")";

statement.executeUpdate(query1);

System.out.println("New Customer record inserted successfully.");
System.out.println("=====");

}

if (Choice == 2) {

    // Enter a new department

    //asking user to enter the department number and reading the value
    System.out.println("Enter the Unique department number ");
    int department_number = Integer.parseInt(br.readLine());

    //reading the department data as given by the user
    System.out.println("Enter the department data");
    String data = br.readLine();

    //Executing the query 2

    final String query2 = "INSERT INTO Department(department_number,department_data)
values(" + department_number + "','" + data + ")";

    statement.executeUpdate(query2);
}

```

```

//showing successful running of query
System.out.println("New Department inserted successfully.");
System.out.println("=====");
}

if (Choice == 3) {
    //Enter a new assembly with its customer-name, assembly-details, assembly-id, and
dateordered

    //asking user to enter the assembly and reading the value
    System.out.println("Enter the Unique Assembly ID ");
    int assembly_id = Integer.parseInt(br.readLine());
    //reading the ordered date from the user

    System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
    String date = br.readLine();

    //reading the assembly details from the user
    System.out.println("Enter the assembly details");
    String data = br.readLine();

    //reading the customer name from the user
    System.out.println("Enter the Customer's name who ordered the assembly");
    String cname = br.readLine();

    //Executing the query 3

    final String query3 = "INSERT INTO
Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" +
date + "','" + data + "','" + cname + "')";

    statement.executeUpdate(query3);

    System.out.println("New Assembly record inserted successfully.");
    System.out.println("=====");
}

if (Choice == 4) {

    //Enter a new process-id and its department together with its type and information
relevant to the type

    //asking user to enter the process ID and reading the value
    System.out.println("Enter the Process ID ");
    int process_id = Integer.parseInt(br.readLine());

```

```

//reading the ordered date from the user
System.out.println("Enter the Department Number that supervises this process");
int department_number = Integer.parseInt(br.readLine());

//reading the customer name from the user
System.out.println("Enter Process Data");
String process_data = br.readLine();

//reading the customer name from the user
System.out.println("Enter the Type of Process - paint, fit or cut ");
String type = br.readLine();

//if the type entered by the user is paint then asking for paint process information
if(type.equals("paint")) {

    //reading the paint type
    System.out.println("Enter the paint type ");

    String paint_type = br.readLine();

    //reading the paint Method
    System.out.println("Enter the painting Method");

    String p_method = br.readLine();

    //Executing the query 4

    final String query4 = "INSERT INTO
Process(process_id,department_number,process_data) values('" + process_id + "','" +
department_number + "','" + process_data + "')";

    statement.executeUpdate(query4);

    //inserting values into Paint_process table
    final String query4_1 = "INSERT INTO paint_process values('" + process_id + +
"','" + paint_type + "','" + p_method + "')";

    statement.executeUpdate(query4_1);

    System.out.println(" New Process record and paint_process record are inserted
successfully.");
    System.out.println(" ======");

}

//if the type entered by the user is fit then asking for fit process information

```

```

else if(type.equals("fit")) {

    //reading the fit type
    System.out.println("Enter the fit type ");

    String fit_type = br.readLine();

    //Executing the query 4

    //inserting values into Process table
    final String query4 = "INSERT INTO
Process(process_id,department_number,process_data) values('" + process_id + "','" +
department_number + "','" + process_data + "')";

    statement.executeUpdate(query4);

    //inserting values into fit_process table
    final String query4_1 = "INSERT INTO fit_process values('" + process_id + "','" +
+ fit_type + "')";

    statement.executeUpdate(query4_1);

    System.out.println(" New Process record and fit_process record are inserted
successfully.");
    System.out.println(" =====");
}

///if the type entered by the user is cut then asking for cut process information
else if(type.equals("cut")) {

    //reading cutting type
    System.out.println("Enter the cutting type ");

    String cut_type = br.readLine();

    //reading machine type from user
    System.out.println("Enter the Machine type");

    String m_type = br.readLine();

    //Executing the query 4
    //inserting values into Process table
}

```

```

final String query4 = "INSERT INTO Process(process_id,department_number,process_data)
values(" + process_id + "','" + department_number + "','" + process_data + "')";

statement.executeUpdate(query4);

//inserting values into paint_process table
final String query4_1 = "INSERT INTO paint_process values(" + process_id + "','" + cut_type +
"','" + m_type + "')";

statement.executeUpdate(query4_1);

System.out.println(" New Process record and cut_process record are inserted successfully.");
System.out.println(" =====");

}

//if the type entered by the user is not valid then asking user about the type again!!
else {
System.out.println("please enter correct type - Is it paint or fit or cut ??");
}

}

if (Choice == 5) {
    //Create a new account and associate it with the process, assembly, or department to
    which it is applicable
    //Executing the query5
    //Asking for account type so that we could insert them into the respective account Table
    System.out.println("Enter Type Of Account (Department or Assembly or Process ):");
    String account_type = br.readLine();

    //Depending on the type of account, asking user for details of the account

    //considering when Type of account is Department
    if(account_type.equals("Department")) {

        //asking user to enter the department number and reading the value
        System.out.println("Enter the department number you will be tracking with this account
");
        String department_no = br.readLine();

        //asking user to enter the account number and reading the value
        System.out.println("Enter the account number ");
        String account_no = br.readLine();

        //reading the established_date from the user
}

```

```

System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
String date = br.readLine();

//reading the department_costs value given by user
System.out.println("Enter the department costs");
int costs = Integer.parseInt(br.readLine());

//Executing the query 5

//Inserting into department account
final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date +
"','" + costs + "')";

statement.executeUpdate(query5);

// Updating the department account as per the department number number given
final String query5_1= "UPDATE department SET account_no = '" + account_no + "' WHERE
department_number = '" + department_no + "'";

statement.executeUpdate(query5_1);

System.out.println("Deparment Account record inserted successfully.");
System.out.println("=====");
}

//considering when Type of account is Assembly
else if(account_type.equals("Assembly")) {

//asking user to enter the assembly id and reading the value
System.out.println("Enter the Assembly ID you will be tracking with this account ");
String assembly_id = br.readLine();

//asking user to enter the account and reading the value
System.out.println("Enter the account number ");
String account_no = br.readLine();

//reading the established_date from the user
System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
String date = br.readLine();

//reading the assembly costs value given by user
System.out.println("Enter the Assembly costs");
int costs = Integer.parseInt(br.readLine());

```

```

//Executing the query 5
//Inserting the values assembly account
final String query5 = "INSERT INTO assembly_account values(" + account_no + "','" + date +
"','" + costs + "')";

statement.executeUpdate(query5);

final String query5_1= "UPDATE assembly SET account_no ='" + account_no + "' WHERE
assembly_id ='" + assembly_id + "'";

statement.executeUpdate(query5_1);

System.out.println("Assembly Account record inserted successfully.");
System.out.println("=====");
}

//considering when Type of account is Process
else if(account_type.equals("Process")) {

//asking the process ID from User and reading the value
System.out.println("Enter the process ID you will be tracking with this account ");
String process_id = br.readLine();

//asking user to enter the account and reading the value
System.out.println("Enter the account number ");
String account_no = br.readLine();

//reading the established_date from the user
System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
String date = br.readLine();

//reading the process costs value given by user
System.out.println("Enter the Process costs");
int costs = Integer.parseInt(br.readLine());

//Executing the query 5

final String query5 = "INSERT INTO process_account values(" + account_no + "','" + date +
"','" + costs + "')";

statement.executeUpdate(query5);

```

```

final String query5_1= "UPDATE process SET account_no = '" + account_no + "' WHERE
process_id = '" + process_id + "'";
statement.executeUpdate(query5_1);
System.out.println("Process Account record inserted successfully.");
System.out.println("=====");
}
else {
    System.out.println("please enter- Department or Assembly or Process ");
}
}

if(Choice == 6) {

//Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced

//asking user to enter the job number and reading the value
System.out.println("Enter the Job Number");
int job_number = Integer.parseInt(br.readLine());

//Reading the commencement date
System.out.println("Enter the Job Commencement date - in the form of YYYY-MM-DD:");
String date = br.readLine();

//reading the assembly ID value given by user
System.out.println("Enter the Assembly ID");
int assembly_id = Integer.parseInt(br.readLine());



//reading the process ID value given by user
System.out.println("Enter the process ID");
int process_id = Integer.parseInt(br.readLine());


//Executing the query 6

//Inserting the values into Job table
final String query6 = "INSERT INTO job(job_no,commence_date,assembly_id,process_id)
values('" + job_number + "','" + date + "','" + assembly_id + "','" + process_id + "')";

```

```

statement.executeUpdate(query6);

System.out.println("New Job record inserted successfully.");

System.out.println("=====");

}

if(Choice == 7) {
//At the completion of a job, enter the date it completed and the information relevant to the type
of job

//asking the user for the job number and reading it
System.out.println("Enter the Job number which got completed");
int job_no = Integer.parseInt(br.readLine());

//Reading the completion date
System.out.println("Enter the Job Completion date - in the form of YYYY-MM-DD:");
String completion_date = br.readLine();

//asking user for the type of job
System.out.println("Enter the type of Job - paint or fit or cut");
String type = br.readLine();

//considering when type is paint
if(type.equals("paint")) {

//reading the paint color
System.out.println("Enter the color ");
String color = br.readLine();

//reading the paint volume
System.out.println("Enter the volume");
float volume = Float.valueOf(br.readLine()).floatValue();

//reading the labor time from the user for this job
System.out.println("Enter the labor time");
float labor_time = Float.valueOf(br.readLine()).floatValue();

//Executing the query 7

//updating the job table as per the completion date given
final String query7 = "UPDATE job SET completion_date = " + completion_date + " WHERE
job_no = " + job_no + """;

statement.executeUpdate(query7);

```

```

//inserting the values of Paint Job
final String query7_1 = "INSERT INTO paint_job values('" + job_no + "','" + color + "','" +
volume + "','" + labor_time + "')";

statement.executeUpdate(query7_1);

System.out.println(" Completion date has been updated and paint job record is inserted");
System.out.println(" =====");

}

//considering when job type is fit
else if(type.equals("fit")) {

//asking user for the labor time and reading it
System.out.println("Enter the labor time");
float labor_time = Float.valueOf(br.readLine()).floatValue();

//Executing the query 7

//updating the job table as per the completion date given
final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE
job_no = '" + job_no + "'";

statement.executeUpdate(query7);

//inserting the values of fit Job
final String query7_1 = "INSERT INTO fit_job values('" + job_no + "','" + labor_time + "')";

statement.executeUpdate(query7_1);

System.out.println(" Completion date has been updated and paint job record is inserted");
System.out.println(" =====");
}

else if(type.equals("cut")) {
//reading the machine Type
System.out.println("Enter the type of Machine used ");
String machine_type = br.readLine();

//reading the time the machine used
System.out.println("Enter the amount of time machine was used(in hours)");
float time = Float.valueOf(br.readLine()).floatValue();

//asking user about the material used
System.out.println("Enter the Material Used ");

```

```

String material = br.readLine();

//asking user for the labor time and reading it
System.out.println("Enter the labor time");
float labor_time = Float.valueOf(br.readLine()).floatValue();

//Executing the query 7
//updating the job table as per the completion date given
final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE
job_no = '" + job_no + "'";

statement.executeUpdate(query7);

//inserting the values of cut Job
final String query7_1 = "INSERT INTO cut_job values('" + job_no + "','" + machine_type + "','" +
+ time + "','" + material + "','" + labor_time + "')";

statement.executeUpdate(query7_1);

System.out.println(" Completion date has been updated and paint job record is inserted");
System.out.println(" =====");

}

else {
System.out.println("please enter correct type - Is it paint or fit or cut ??");
}

}

if(Choice == 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

//asking user to enter the transaction number and reading it
System.out.println("Enter the transaction Number");
int trans_no= Integer.parseInt(br.readLine());

//reading JOB Number on which this transaction is recoded
System.out.println("Enter the job Number on which this transaction is recoded");
int job_no= Integer.parseInt(br.readLine());

//reading the sup-cost from the user
System.out.println("Enter the sup-cost");
float sup_cost = Float.valueOf(br.readLine()).floatValue();

```

```

//Executing the query 8
// inserting the values into the cost transactions table

final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost)
values(" + trans_no + "','" + job_no + "','" + sup_cost + "')";

statement.executeUpdate(query8);

//updating cost_transcaions with the specific assembly account as per the given job number
final String query8_1 = "UPDATE cost_transactions SET asse_account_no = (SELECT
DISTINCT assembly.account_no FROM cost_transactions,job,assembly "
        + "WHERE job.job_no = '" + job_no + "' AND job.assembly_id =
assembly.assembly_id ) "
+ "WHERE job_no = '" + job_no + "'";

statement.executeUpdate(query8_1);

//updating cost_transcaions with the specific process account as per the given job number
final String query8_2 ="UPDATE cost_transactions SET pro_account_no = (SELECT
DISTINCT process.account_no FROM cost_transactions,job,process "
        + "WHERE job.job_no = '" + job_no + "' AND job.process_id =
process.process_id ) "
+ "WHERE job_no = '" + job_no + "'";

statement.executeUpdate(query8_2);

//updating cost_transcaions with the specific department account as per the given job number
final String query8_3 = "UPDATE cost_transactions SET dep_account_no = (SELECT
DISTINCT department.account_no FROM cost_transactions,job,process,department "
        + "WHERE job.job_no = '" + job_no + "' AND job.process_id =
process.process_id AND process.department_number = department.department_number)"
+ "WHERE job_no = '" + job_no + "'";

statement.executeUpdate(query8_3);

// Updating the assembly account by adding the sup -costs to already available assembly_costs
final String query8_4 ="UPDATE assembly_account SET assembly_costs = assembly_costs +
(select sup_cost FROM cost_transactions,assembly_account WHERE \r\n" +
        "    cost_transactions.asse_account_no = assembly_account.account_no \r\n" +
        "    AND job_no = '" + job_no + "') "
        + "WHERE account_no = (select asse_account_no FROM
cost_transactions WHERE job_no = '" + job_no + "')";

statement.executeUpdate(query8_4);

```

```

// Updating the process account by adding the sup -costs to already available process_costs
final String query8_5 = "UPDATE process_account SET process_costs = process_costs +
(select sup_cost FROM cost_transactions,process_account WHERE \r\n" +
"    cost_transactions.pro_account_no =
process_account.account_no \r\n" +
"        AND job_no = "+ job_no + ") where account_no = (select
pro_account_no FROM cost_transactions WHERE job_no = "+ job_no + ");

statement.executeUpdate(query8_5);

// Updating the department account by adding the sup -costs to already available
department_costs
final String query8_6 = "UPDATE department_account SET department_costs =
department_costs + (select sup_cost FROM cost_transactions,department_account WHERE
\r\n" +
"    cost_transactions.dep_account_no = department_account.account_no \r\n" +
"        AND job_no = "+ job_no + ") where account_no = (select dep_account_no
FROM cost_transactions WHERE job_no = "+ job_no + ");

statement.executeUpdate(query8_6);

System.out.println("Transaction number is Inserted and All the affected Accounts costs are
updated");
System.out.println("=====");

}

if (Choice == 9) {
//Retrieve the cost incurred on an assembly-id

//asking user to enter the assembly ID and reading the value
System.out.println("Enter the assembly ID on which you want to retrive the costs ");

int assembly_id = Integer.parseInt(br.readLine());

//Executing the query 9

//Retrieving the assembly_costs
final String query9 = "SELECT assembly_costs FROM assembly_account WHERE account_no
IN (SELECT account_no FROM assembly WHERE assembly_id = "+ assembly_id + ")";


```

```

ResultSet rs = statement.executeQuery(query9);

//since it is a result set, printing all the values
while (rs.next()) {
    System.out.println(rs.getString(1)); }

System.out.println("Costs Incurred on the given Assembly ID are retrieved");

System.out.println("=====");

}

if(Choice == 10) {

    //asking department number from user and reading it
    System.out.println("Enter the Department Number");
    int department_no= Integer.parseInt(br.readLine());
    //Asking user to enter the completion date
    System.out.println("Enter the completion Date - in the form of YYYY-MM-DD:");
    String date = br.readLine();
    final String query10 ="EXEC query10 @department_number = '"+department_no+"',
@completion_date = '"+date+"';";
    ResultSet rs = statement.executeQuery(query10);
    System.out.println("Total Labor Hours");

    //since it is a result set, printing all the values
    while (rs.next()) {
        System.out.println(rs.getInt(1)); }

}

if(Choice == 11) {

    //Retrieve the processes through which a given assembly-id has passed so far (in
    datecommenced order) and the department responsible for each process

    //Asking user to enter the assembly ID
    System.out.println("Enter the Assembly ID");
    int assembly_id= Integer.parseInt(br.readLine());

    final String query11 = "SELECT process.process_id,process.department_number FROM
process, department,job "
                        + "WHERE job.assembly_id = " + assembly_id + " and
department.department_number = process.department_number "
                        + "and job.process_id = process.process_id ORDER BY
job.commence_date";
}

```

```

//since it is a result set, printing all the values
ResultSet rs = statement.executeQuery(query11);

System.out.println("Process ID | Department No");

while (rs.next()) {
    System.out.println(String.format("%s | %s ",
        rs.getInt(1),
        rs.getInt(2)));
}

System.out.println("=====");
}

if(Choice == 12) {

//Retrieve the jobs (together with their type information and assembly-id) completed during a
given date in a given department

//Asking user to enter the department Number
System.out.println("Enter the Department Number");
int department_no= Integer.parseInt(br.readLine());
//Asking user to enter the completion date
System.out.println("Enter the completion Date");
String date = br.readLine();
final String query12_1 = "SELECT job.job_no, job.commence_date, completion_date,
cut_job.type_of_machine, cut_job.time_used, cut_job.material_used, labor_time,
assembly_id\r\n" +
"
"
FROM job, process, cut_job \r\n" +
"
WHERE job.completion_date=" + date + " AND "
+ "job.process_id=process.process_id AND
job.job_no=cut_job.job_no AND department_number=" + department_no + " ";

ResultSet rs = statement.executeQuery(query12_1);
//since it is a result set, printing all the values
while (rs.next()) {
    System.out.println(String.format(" %s | %s",
        rs.getInt(1),
        rs.getString(2),
        rs.getString(3),
        rs.getString(4),
        rs.getFloat(5),

```

```

        rs.getString(6),
        rs.getInt(7),
        rs.getString(8)));
    }

//since it is a result set, printing all the values

final String query12_2 ="SELECT job.job_no, job.commence_date, completion_date,
labor_time, assembly_id\r\n" +
    "  FROM job, process, fit_job \r\n" +
    " WHERE job.completion_date=" + date + "" AND "
    + "job.process_id=process.process_id AND job.job_no=fit_job.job_no AND
department_number=" + department_no + "";

ResultSet rs1 = statement.executeQuery(query12_2);

//System.out.println("job_no | commence_date | completion_date | labor_time | assembly_id ");
//since it is a result set, printing all the values
while (rs1.next()) {
    System.out.println(String.format(" %s | %s | %s | %s | %s ",
        rs1.getInt(1),
        rs1.getString(2),
        rs1.getString(3),
        rs1.getInt(4),
        rs1.getString(5)));
}

final String query12_3 = "SELECT job.job_no, job.commence_date, completion_date,
paint_job.color, paint_job.volume,paint_job.labor_time, assembly_id\r\n" +
    "  FROM job, process, paint_job \r\n" +
    " WHERE job.completion_date=" + date + "" AND "
    + "job.process_id=process.process_id AND job.job_no=paint_job.job_no AND
department_number= " + department_no + "";
;

ResultSet rs2 = statement.executeQuery(query12_3);
//since it is a result set, printing all the values
while (rs2.next()) {
    System.out.println(String.format(" %s | %s | %s | %s | %s | %s | %s ",
        rs.getInt(1),
        rs.getString(2),
        rs.getString(3),
        rs.getString(4),
        rs.getFloat(5),
        rs.getInt(7),

```

```

        rs.getString(8)));

    System.out.println("=====");
    }

}

if(Choice == 13) {

// Retrieve the customers (in name order) whose category is in a given range

//asking user to enter the range
System.out.println("Enter the category range");
//asking user for the minimum value in the range
System.out.println("Minimum category value(category has values from 1 to 10)");
int min_cat= Integer.parseInt(br.readLine());

//asking user to enter the maximum value in the range in the range
System.out.println("Maximum category value(category has values from 1 to 10)");
int max_cat = Integer.parseInt(br.readLine());

//Executing the query 13

final String query13 = "SELECT * FROM customer WHERE category BETWEEN " + min_cat
+ " and " + max_cat + "";

ResultSet rs1 = statement.executeQuery(query13);
System.out.println("Name | Address | Category");
//since it is a result set, printing all the values

while (rs1.next()) {
    System.out.println(String.format("%s | %s | %s",
        rs1.getString(1),
        rs1.getString(2),
        rs1.getInt(3)));
}

System.out.println("Retrieved the customers (in name order) whose category is in a given
range");
System.out.println("=====");

}

if(Choice == 14) {
    // Delete all cut-jobs whose job-no is in a given range
}

```

```

//asking user to enter the range
System.out.println("Enter the Job Number range");
//asking the starting value on the range

System.out.println("Enter the First job number in the range ");
int first_job= Integer.parseInt(br.readLine());
//asking the last value in the range

System.out.println("Enter the Last job number in the range");
int last_job = Integer.parseInt(br.readLine());

//Executing the query 14

final String query14 = "DELETE FROM cut_job WHERE job_no BETWEEN '" +
first_job + "' and '" + last_job + "'";

statement.executeUpdate(query14);

System.out.println("All the cut_jobs in the given range are Deleted");

System.out.println("=====");

}

if(Choice == 15) {

// Change the color of a given paint job

//asking user to enter the Job Number
System.out.println("Enter the Job Number of the Paint Job you want to change the color for");
int first_job= Integer.parseInt(br.readLine());

//reading the color that user want to change the existing cut color to
System.out.println("which color do you want to change it to?");
String color = br.readLine();

//Executing the query 15

final String query15 = "UPDATE paint_job SET color = '" + color + "' WHERE job_no = '" +
first_job + "'";

statement.executeUpdate(query15);

System.out.println("Color of a given paint job is changed");

```

```

System.out.println("=====");
}

if(Choice == 16) {

String file_name;
//Import: Enter new customers from a data file until the file is empty
//asking user to enter the name of the file that is to be imported

System.out.println("Enter the file name: ");

file_name = myScan.next();

//path where the filename is searched

FileInputStream fstream = new FileInputStream("/Users/dvsnv/Desktop/" + file_name);

DataInputStream i = new DataInputStream(fstream);
BufferedReader br1 = new BufferedReader(new InputStreamReader( i ));

String Line;
//Importing each row until the file is empty

while ((Line = br1.readLine()) != null) {
String x[] = Line.split(",");
String name = x[0];
String address = x[1];
String category=x[2];

statement.executeUpdate("INSERT INTO customer values('" + name + "','" + address + "','" +
category + "')");

System.out.println("row inserted succesfully");

}
System.out.println("File imported succesfully!!!!");

}

if(Choice == 17) {

String out_file;
//Export: Retrieve the customers (in name order) whose category is in a given range and output
them to a data file instead of screen

```

```
//asking user to enter the range
System.out.println("Enter the category range");

//reading minimum value in the range
System.out.println("Minimum category value(category has values from 1 to 10)");
int min_cat= Integer.parseInt(br.readLine());

//reading the maximum value in the range
System.out.println("Maximum category value(category has values from 1 to 10)");
int max_cat = Integer.parseInt(br.readLine());

final String query13 = "SELECT * FROM customer WHERE category BETWEEN " + min_cat
+ " and " + max_cat + "";

ResultSet rs = statement.executeQuery(query13);
String string = null;

System.out.println("Enter output file name: ");
out_file = br.readLine();

BufferedWriter export = new BufferedWriter(new FileWriter("/Users/dvsnv/Desktop/" +
out_file));

//reading each row and exporting it to the file
while (rs.next()) {
    string = rs.getString("name") + " " + rs.getString("address") + " " +
rs.getString("category");
        export.write("\n");
        export.write(string);
        export.write("\n");

    System.out.println(" Row Exported succesfully!"); }

    System.out.println(" File exported succesfully!!!");

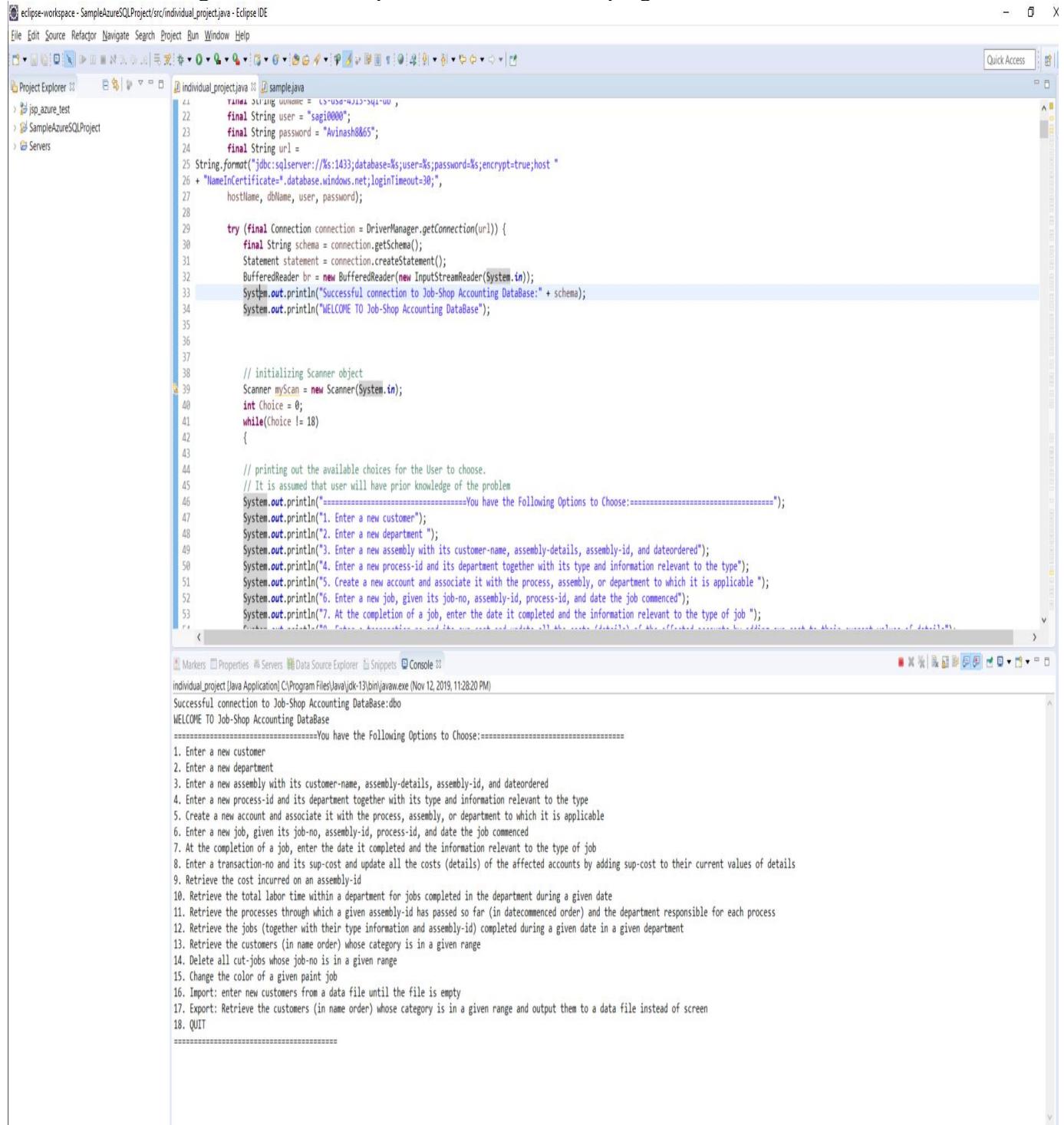
    export.close();

}

if (Choice == 18)
//Choice 18 will exit the program
System.out.println("Thanks... Bye...");
}}}

}
```

Screen shot showing Successful Compilation of Java Source program



Task 6

Java program Execution

6.1. Screenshots showing the testing of query 1

The screenshot shows the Eclipse IDE interface with two tabs open: 'individual_project.java' and 'sample.java'. The code in 'sample.java' is a Java program that inserts a new customer record into a database. It prompts the user for customer name, address, and category, and then prints a success message. The 'Console' view at the bottom shows the execution of the program, including user input and system output.

```
1 if (Choice == 1) {
2     //Enter a new customer in to the Data Base
3
4     //asking user to enter the customer name and reading the value
5     System.out.println("Enter the customer Name ");
6     String name = br.readLine();
7
8     //reading the customer's address from the user
9     System.out.println("Enter the customer's address ");
10    String address = br.readLine();
11
12
13    //reading the customer's category value given by user
14    System.out.println("Enter the customer's category- values from 1 to 10");
15    int category = Integer.parseInt(br.readLine());
16
17    //Executing the query 1
18
19    final String query1 = "INSERT INTO customer values('" + name + "','" + address + "','" + category + "')";
20
21    statement.executeUpdate(query1);
22
23    System.out.println("New Customer record inserted successfully.");
24    System.out.println("=====");
25
26}
27
28 if (Choice == 2) {
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101 }
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:09:17 AM)

15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====

1
Enter the customer Name
Subbu
Enter the customer's address
1080 East Brooks Street Norman
Enter the customer's category- values from 1 to 10
1
New Customer record inserted successfully.
=====You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced order) and the department responsible for each process

```

18. QUIT
=====
Enter the customer Name
Ghani
Enter the customer's address
C106,Creson Park,Norwan
Enter the customer's category- values from 1 to 10
10
New Customer record inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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

```

```

18. QUIT
=====
Enter the customer Name
Abhinash
Enter the customer's address
250 spring fields, Norman
Enter the customer's category- values from 1 to 10
10
New Customer record inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
11. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
12. Retrieve the customers (in name order) whose category is in a given range
13. Delete all cut-jobs whose job-no is in a given range
14. Change the color of a given paint job

```

The screenshot shows an IDE interface with two tabs open: `individual_project.java` and `sample.java`. The `individual_project.java` tab displays the following Java code:

```
71 if (Choice == 1) {
72     //Enter a new customer in to the Data Base
73
74     //asking user to enter the customer name and reading the value
75     System.out.println("Enter the customer Name ");
76     String name = br.readLine();
77
78
79     //reading the customer's address from the user
80     System.out.println("Enter the customer's address ");
81     String address = br.readLine();
82
83
84
85     //reading the customer's category value given by user
86     System.out.println("Enter the customer's category- values from 1 to 10");
87     int category = Integer.parseInt(br.readLine());
88
89
90     //Executing the query 1
91
92     final String query1 = "INSERT INTO customer values('" + name + "','" + address + "','" + category + "')";
93
94     statement.executeUpdate(query1);
95
96     System.out.println("New Customer record inserted successfully.");
97     System.out.println("=====");
98
99 }
100 if ((Choice == 2) &&
```

The `sample.java` tab shows the command-line interface options:

```
15: Enter the color of a given paint job
16: Import: enter new customers from a data file until the file is empty
17: Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18: QUIT
=====
1
Enter the customer Name
Goutham
Enter the customer's address
brooks st oklahoma
Enter the customer's category- values from 1 to 10
5
New Customer record inserted successfully.
=====
You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
4. Enter a new job with its job-id and date completed 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 no. of records through which a given assembly-id has moved on for (in datocommanded order) and the department responsible for each process
```

The screenshot shows an IDE interface with a code editor and a terminal window.

Code Editor:

```
1  if (Choice == 1) {
2      //Enter a new customer in to the Data Base
3
4      //asking user to enter the customer name and reading the value
5      System.out.println("Enter the customer Name ");
6      String name = br.readLine();
7
8
9      //reading the customer's address from the user
10     System.out.println("Enter the customer's address ");
11     String address = br.readLine();
12
13
14
15     //reading the customer's category value given by user
16     System.out.println("Enter the customer's category- values from 1 to 10");
17     int category = Integer.parseInt(br.readLine());
18
19     //Executing the query 1
20
21     final String query1 = "INSERT INTO customer values(" + name + "," + address + "," + category + ")";
22
23     statement.executeUpdate(query1);
24
25     System.out.println("New Customer record inserted successfully.");
26     System.out.println("=====");
27 }
28
29 if (Choice == 2) {
30
31 }
```

Terminal Window:

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:09:17 AM)
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
1
Enter the customer Name
Maharshi
Enter the customer's address
227 clesen, Norman
Enter the customer's category- values from 1 to 10
6
New Customer record inserted successfully.
=====
You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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, gives 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 the total labor time in hours 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 datecommenced 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
```

Checking the customer Table after the Insertions:

The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane: Shows various database objects like Welcome, Create_Table.sql, ID_project_tables.sql, DROP_TABLES.sql, Insertions_1P.sql, Queries_1P.sql (selected), Queires.sql, and SQLQuery_1.
- OPEN EDITOR**: 5 UNSAVED
- File Path**: C:\Users\dvsnv\OneDrive\Documents\Queries_1P.sql
- Toolbar Buttons**: Run, Cancel, Disconnect, Change Connection (set to cs-dsa-4513-sql-db), Explain, Enable SQLCMD.
- Text Editor Content**:

```
1  /* SQL query for option 1 */
2  --Enter a new customer
3
4
5
6  select * from customer;
7
8  /* SQL query for option 2 */
```
- Results Tab**: Displays the output of the query "select * from customer;".
- Table Data**:

	name	address	category
1	Abhinash	250 spring fields, Norman	10
2	Ghani	C106,Crimson Park,Norman	3
3	Gowtham	brooks st oklahoma	9
4	Maharshi	227 clasen, Norman	6
5	Subbu	1003D East Brooks Street Nor...	1
- Outline Tab**: Shows NO FOLDER OPENED and OUTLINE.

6.2. Screenshots showing the testing of query 2

Department Table before Insertion:

```

-- Enter a new department
select * FROM department;

```

department_number	department_data	account_no

Entering new Department values:

```

if (Choice == 2) {
    // Enter a new department
    //asking user to enter the department number and reading the value
    System.out.println("Enter the Unique department number ");
    int department_number = Integer.parseInt(br.readLine());

    //reading the department data as given by the user
    System.out.println("Enter the department data");
    String data = br.readLine();

    //Executing the query 2
    final String query2 = "INSERT INTO Department(department_number,department_data) values('" + department_number + "','" + data + "')";
    statement.executeUpdate(query2);

    //showing successful running of query
    System.out.println("New Deparment inserted successfully.");
    System.out.println("*****");
}

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 10:9:17 AM)

16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

2
Enter the Unique department number
2000
Enter the department data
This is a new department
New Deparment inserted successfully.
=====

You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

98
99
100
101    if (Choice == 2) {
102
103        // Enter a new department
104
105        //asking user to enter the department number and reading the value
106        System.out.println("Enter the Unique department number ");
107        int department_number = Integer.parseInt(br.readLine());
108
109
110        //reading the department data as given by the user
111        System.out.println("Enter the department data");
112        String data = br.readLine();
113
114        //Executing the query 2
115
116        final String query2 = "INSERT INTO Department(department_number,department_data) values('"+ department_number + "','" + data + "')";
117
118        statement.executeUpdate(query2);
119
120        //showing successful running of query
121        System.out.println("New Department inserted successfully.");
122        System.out.println("=====");
123    }
124
125    if (Choice == 3) {
126        //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project[Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:27:10 AM)

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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

2
Enter the Unique department number
2001
Enter the department data
Department 2
New Department inserted successfully.
=====

=====You have the Following Options to Choose=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 amnlirable

```

98
99
100
101    if (Choice == 2) {
102
103        // Enter a new department
104
105        //asking user to enter the department number and reading the value
106        System.out.println("Enter the Unique department number ");
107        int department_number = Integer.parseInt(br.readLine());
108
109
110        //reading the department data as given by the user
111        System.out.println("Enter the department data");
112        String data = br.readLine();
113
114        //Executing the query 2
115
116        final String query2 = "INSERT INTO Department(department_number,department_data) values('"+ department_number + "','" + data + "')";
117
118        statement.executeUpdate(query2);
119
120        //showing successful running of query
121        System.out.println("New Department inserted successfully.");
122        System.out.println("=====");
123    }
124
125    if (Choice == 3) {
126        //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project[Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:27:10 AM)

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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

Enter the Unique department number
2002
Enter the department data
Department related to others
New Department inserted successfully.
=====

=====You have the Following Options to Choose=====

1. Enter a new customer

```

98
99
100    if (Choice == 2) {
101
102        // Enter a new department
103
104        //asking user to enter the department number and reading the value
105        System.out.println("Enter the Unique department number ");
106        int department_number = Integer.parseInt(br.readLine());
107
108
109        //reading the department data as given by the user
110        System.out.println("Enter the department data");
111        String data = br.readLine();
112
113        //Executing the query 2
114
115        final String query2 = "INSERT INTO Department(department_number,department_data) values('" + department_number + "','" + data + "')";
116
117        statement.executeUpdate(query2);
118
119        //showing successful running of query
120        System.out.println("New Department inserted successfully.");
121        System.out.println("=====");
122    }
123
124    if (Choice == 3) {
125        //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:27:10 AM)

16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

Enter the Unique department number
 2003
 Enter the department data
 Department
 New Department inserted successfully.
 =====

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

99
100
101    if (Choice == 2) {
102
103        // Enter a new department
104
105        //asking user to enter the department number and reading the value
106        System.out.println("Enter the Unique department number ");
107        int department_number = Integer.parseInt(br.readLine());
108
109
110        //reading the department data as given by the user
111        System.out.println("Enter the department data");
112        String data = br.readLine();
113
114        //Executing the query 2
115
116        final String query2 = "INSERT INTO Department(department_number,department_data) values('" + department_number + "','" + data + "')";
117
118        statement.executeUpdate(query2);
119
120        //showing successful running of query
121        System.out.println("New Department inserted successfully.");
122        System.out.println("=====");
123    }
124
125    if (Choice == 3) {
126        //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:27:10 AM)

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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

2
 Enter the Unique department number
 2004
 Enter the department data
 Department
 New Department inserted successfully.
 =====

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced order) and the department responsible for each process

Department Table After Insertions:

The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane on the left lists several files: Welcome, Create_Table.sql, ID_project_tables.sql, DROP_TABLES.sql, Insertions_IP.sql, and Queries_IP.sql.
- OBJECT Explorer** pane shows a folder structure under C:\Users\dvsnv\OneDrive\Documents\Queries_IP.sql.
- SQL Editor** pane at the top has tabs for Run, Cancel, Disconnect, Change Connection, Explain, and Enable SQLCMD. The connection is set to cs-dsa-4513-sql-db.
- Text Editor** pane contains the following SQL query:

```
--Enter a new department
select * FROM department;
```
- Results** pane at the bottom displays the data from the department table:

	department_number	department_data	account_no
1	2000	This is a new department	NULL
2	2001	Department 2	NULL
3	2002	Department related to others	NULL
4	2003	Department	NULL
5	2004	New Department	NULL

6.3. Screenshots showing the testing of query 3:

Assembly table Before Insertion

The screenshot shows the SSMS interface with the following details:

- File Explorer:** Shows several .sql files including ID_project_tables.sql, DROP_TABLES.sql, and Insertions_IP.sql.
- Object Explorer:** Shows database objects like Customer, Order, and Assembly.
- Toolbox:** Standard SSMS icons.
- Toolbar:** Run, Cancel, Disconnect, Change Connection, Explain, Enable SSQLCMD.
- Connection:** cs-dsa-4513-sql-db
- Text Editor:** Contains the following SQL code:


```

19  /* SQL query for option 3 */
20  --Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
21
22  select * from assembly;
23
24
      
```
- Results Grid:** Labeled "Results" and "Messages". The results grid has columns: assembly_id, date_ordered, assembly_details, c_name, account_no. It is currently empty.

Insertion for Assembly Table

The screenshot shows an IDE window with the following details:

- Project Explorer:** Shows individual_project.java and sample.java.
- Code Editor:** Contains Java code for inserting data into the Assembly table:


```

125 if (Choice == 3) {
126     //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128     //asking user to enter the assembly and reading the value
129     System.out.println("Enter the Unique Assembly ID");
130     int assembly_id = Integer.parseInt(br.readLine());
131
132     //reading the ordered date from the user
133     System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134     String date = br.readLine();
135
136     //reading the assembly details from the user
137     System.out.println("Enter the assembly details");
138     String data = br.readLine();
139
140     //reading the customer name from the user
141     System.out.println("Enter the Customer's name who ordered the assembly");
142     String cname = br.readLine();
143
144     //Executing the query
145
146     final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148     statement.executeUpdate(query3);
149
      
```
- Console:** Displays the output of the Java application, which includes prompts for assembly_id, date, assembly_details, and cname, followed by a success message: "New Assembly record inserted successfully."
- Bottom Status Bar:** Shows the current date and time: Nov 13, 2019, 1:51:01 AM.

```

arcn project run window help
individual_project.java sample.java
125     if (Choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query 3
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148         statement.executeUpdate(query3);
149

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:51:01 AM)

11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

```

=====
3
Enter the Unique Assembly ID
1003
Enter the ordered date - in the form of YYYY-MM-DD:
2019-09-29
Enter the assembly details
New Assembly 2
Enter the Customer's name who ordered the assembly
Ghani
New Assembly record inserted successfully.

=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

```

arcn Project Run Window Help
individual_project.java sample.java
125     if (Choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query 3
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148         statement.executeUpdate(query3);
149

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:51:01 AM)

14. Delete all cut-jobs whose job-no is in a given range
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

```

=====
3
Enter the Unique Assembly ID
1003
Enter the ordered date - in the form of YYYY-MM-DD:
2019-10-01
Enter the assembly details
New Assembly 3
Enter the Customer's name who ordered the assembly
Goorthum
New Assembly record inserted successfully.

=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

```

125     if (Choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148         statement.executeUpdate(query3);
149

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)

- Create a new account and associate it with the process, assembly, or department to which it is applicable
- Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
- At the completion of a job, enter the date it completed and the information relevant to the type of job
- 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
- Retrieve the cost incurred on an assembly-id
- Retrieve the total labor time within a department for jobs completed in the department during a given date
- Retrieve the processes through which a given assembly-id has passed so far (in datocommenced order) and the department responsible for each process
- Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
- Retrieve the customers (in name order) whose category is in a given range
- Delete all cut-jobs whose job-no is in a given range
- Change the color of a given paint job
- Import: enter new customers from a data file until the file is empty
- Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
- QUIT

=====

3
Enter the Unique Assembly ID
1003
Enter the ordered date - in the form of YYYY-MM-DD:
2019-01-15
Enter the assembly details
New assembly 4
Enter the Customer's name who ordered the assembly
Abhinash
New Assembly record inserted successfully.
=====

You have the Following Options to Choose:

- Enter a new customer
- Enter a new department
- Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
- Enter a new process-id and its department together with its type and information relevant to the type

```

125     if (Choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148         statement.executeUpdate(query3);
149

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)

- Retrieve the customers (in name order) whose category is in a given range
- Delete all cut-jobs whose job-no is in a given range
- Change the color of a given paint job
- Import: enter new customers from a data file until the file is empty
- Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
- QUIT

=====

3
Enter the Unique Assembly ID
1004
Enter the ordered date - in the form of YYYY-MM-DD:
2019-02-18
Enter the assembly details
New assembly 5
Enter the Customer's name who ordered the assembly
Maharshi
New Assembly record inserted successfully.
=====

You have the Following Options to Choose:

- Enter a new customer
- Enter a new department
- Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
- Enter a new process-id and its department together with its type and information relevant to the type
- Create a new account and associate it with the process, assembly, or department to which it is applicable
- Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced

The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "individual_projectJava" and contains a file named "sample.java".
- Code Content:** The code in sample.java is as follows:

```
129 if (choice == 3) {
130     //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
131
132     //asking user to enter the assembly and reading the value
133     System.out.println("Enter the Unique Assembly ID");
134     int assembly_id = Integer.parseInt(br.readLine());
135
136     //reading the ordered date from the user
137     System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
138     String date = br.readLine();
139
140     //reading the assembly details from the user
141     System.out.println("Enter the assembly details");
142     String data = br.readLine();
143
144     //reading the customer name from the user
145     System.out.println("Enter the Customer's name who ordered the assembly");
146     String cname = br.readLine();
147
148     //Executing the query 3
149
150     final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values(" + assembly_id + "," + date + "," + data + "," + cname + ")";
151
152     statement.executeUpdate(query3);
153 }
```

- Console Output:** The console shows the execution of the Java application and the resulting database insertions.

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)
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 datecommenced order) and the department responsible for each process
12. List all cut-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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
3
Enter the Unique Assembly ID
1007
Enter the ordered date - in the form of YYYY-MM-DD:
2019-08-20
Enter the assembly details
New assembly 6
Enter the Customer's name who ordered the assembly
Goutham
New Assembly record inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 the job-commence
7. At the completion of a job, enter the date it completed and the information relevant to the type of job
8. Exit the program
```

The screenshot shows a Java application running in an IDE. The code in the editor is as follows:

```
125     if (Choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD ");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values(" + assembly_id + "," + date + "," + data + "," + cname + ")";
147
148         statement.executeUpdate(query3);
149     }
```

The terminal window below shows the execution of the application:

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Nov 13, 2019 15:55:55 AM)
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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
3
Enter the Unique Assembly ID
1008
Enter the ordered date - in the form of YYYY-MM-DD:
2019-07-19
Enter the assembly details
New assembly 9
Enter the Customer's name who ordered the assembly
Subbu
New Assembly record inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
```

```

125     if (choice == 3) {
126         //Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
127
128         //asking user to enter the assembly and reading the value
129         System.out.println("Enter the Unique Assembly ID ");
130         int assembly_id = Integer.parseInt(br.readLine());
131
132         //reading the ordered date from the user
133         System.out.println("Enter the ordered date - in the form of YYYY-MM-DD:");
134         String date = br.readLine();
135
136         //reading the assembly details from the user
137         System.out.println("Enter the assembly details");
138         String data = br.readLine();
139
140         //reading the customer name from the user
141         System.out.println("Enter the Customer's name who ordered the assembly");
142         String cname = br.readLine();
143
144         //Executing the query 3
145
146         final String query3 = "INSERT INTO Assembly(assembly_id,date_ordered,assembly_details,c_name) values('" + assembly_id + "','" + date + "','" + data + "','" + cname + "')";
147
148         statement.executeUpdate(query3);
149

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)

14. Delete all cut-jobs whose job-no is in a given range
 15. Change the color of a given paint job
 16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

3
 Enter the Unique Assembly ID
 1009
 Enter the ordered date - in the form of YYYY-MM-DD:
 2019-04-11
 Enter the assembly details
 New Assembly 10
 Enter the Customer's name who ordered the assembly
 Subbu
 New Assembly record inserted successfully.

=====

====You have the Following Options to Choose=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

Assembly Table after Insertions:

VED

ID_project_tables.sql - sagi00...gi0000) ● DROP_TABLES.sql - sagi00...gi0000) ● Insertions_IP.sql - sagi00...gi0000) ●

C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

19  /* SQL query for option 3 */
20  --Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
21
22  select * from assembly;
23
24

```

Results Messages

	assembly_id	date_ordered	assembly_details	c_name	account_no
1	1000	2019-05-12	New assembly 1	Subbu	NULL
2	1001	2019-09-22	New Assembly 2	Ghani	NULL
3	1002	2019-01-15	New assembly 4	Abhinash	NULL
4	1003	2019-10-01	New Assembly 3	Gowtham	NULL
5	1004	2019-02-18	New assembly 5	Maharshi	NULL
6	1005	2019-03-06	New assembly 6	Maharshi	NULL
7	1006	2019-04-30	New assembly 7	Abhinash	NULL
8	1007	2019-06-30	New assembly 8	Gowtham	NULL
9	1008	2019-07-19	New assembly 9	Subbu	NULL
10	1009	2019-04-11	New Assembly 10	Subbu	NULL

6.4. Screenshots showing the testing of query 4: Process Table Before Insertion:

MySQL Workbench Screenshot:

- Connections: ID_project_tables.sql - sagi00...gi0000 (selected), DROP_TABLES.sql - sagi00...gi0000, Insertions_IP.sql - sagi00...gi0000
- Session: cs-dsa-4513-sql-db
- Query Editor:


```

31
32  /* SQL query for option 4 */
33  --Enter a new process-id and its department together with its type and information relevant to the t
34
35  select * FROM process;
36
      
```
- Results Tab: Shows the 'process' table structure with four columns: process_id, department_number, process_data, and account_no. No data is present.

Insertions:

Eclipse IDE Screenshot:

Java Code Snippet (individual_project.java):

```

154
155
156     if (Choice == 4) {
157
158         //Enter a new process-id and its department together with its type and information relevant to the type
159
160         //asking user to enter the process ID and reading the value
161         System.out.println("Enter the Process ID ");
162         int process_id = Integer.parseInt(br.readLine());
163
164         //reading the ordered date from the user
165         System.out.println("Enter the Department Number that supervises this process");
166         int department_number = Integer.parseInt(br.readLine());
167
168         //reading the customer name from the user
169         System.out.println("Enter Process Data");
170         String process_data = br.readLine();
171
172         //reading the customer name from the user
173         System.out.println("Enter the Type of Process - paint, fit or cut ");
174         String type = br.readLine();
175
176         //if the type entered by the user is paint then asking for paint process information
177         if(type.equals("paint")) {
178
179             //reading the paint type
180             System.out.println("Enter the paint type ");
181
182             String paint_type = br.readLine();
183
184             //reading the paint Method
185             System.out.println("Enter the paint Method ");
      
```

Console Output:

```

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
4
Enter the Process ID
3001
Enter the Department Number that supervises this process
2000
Enter Process Data
New Process 1
Enter the Type of Process - paint, fit or cut
paint
Enter the paint type
Gloss Paint
Enter the painting Method
Underpainting
New Process record and paint_process record are inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
      
```

```

154
155
156     if (Choice == 4) {
157
158         //Enter a new process-id and its department together with its type and information relevant to the type
159
160         //asking user to enter the process ID and reading the value
161         System.out.println("Enter the Process ID ");
162         int process_id = Integer.parseInt(br.readLine());
163
164         //reading the ordered date from the user
165         System.out.println("Enter the Department Number that supervises this process");
166         int department_number = Integer.parseInt(br.readLine());
167
168         //reading the customer name from the user
169         System.out.println("Enter Process Data");
170         String process_data = br.readLine();
171
172         //reading the customer name from the user
173         System.out.println("Enter the Type of Process - paint, fit or cut ");
174         String type = br.readLine();
175
176         //if the type entered by the user is paint then asking for paint process information
177         if(type.equals("paint")) {
178
179             //reading the paint type
180             System.out.println("Enter the paint type ");
181
182             String paint_type = br.readLine();
183
184             //reading the paint Method
185
186             //reading the paint Method
187
188             //reading the paint Method
189
190             //reading the paint Method
191
192             //reading the paint Method
193
194             //reading the paint Method
195
196             //reading the paint Method
197
198             //reading the paint Method
199
200             //reading the paint Method
201
202             //reading the paint Method
203
204             //reading the paint Method
205
206             //reading the paint Method
207
208             //reading the paint Method
209
210             //reading the paint Method
211
212             //reading the paint Method
213
214             //reading the paint Method
215
216             //reading the paint Method
217
218             //reading the paint Method
219
220             //reading the paint Method
221
222             //reading the paint Method
223
224             //reading the paint Method
225
226             //reading the paint Method
227
228             //reading the paint Method
229
230             //reading the paint Method
231
232             //reading the paint Method
233
234             //reading the paint Method
235
236             //reading the paint Method
237
238             //reading the paint Method
239
240             //reading the paint Method
241
242             //reading the paint Method
243
244             //reading the paint Method
245
246             //reading the paint Method
247
248             //reading the paint Method
249
250             //reading the paint Method
251
252             //reading the paint Method
253
254             //reading the paint Method
255
256             //reading the paint Method
257
258             //reading the paint Method
259
260             //reading the paint Method
261
262             //reading the paint Method
263
264             //reading the paint Method
265
266             //reading the paint Method
267
268             //reading the paint Method
269
270             //reading the paint Method
271
272             //reading the paint Method
273
274             //reading the paint Method
275
276             //reading the paint Method
277
278             //reading the paint Method
279
280             //reading the paint Method
281
282             //reading the paint Method
283
284             //reading the paint Method
285
286             //reading the paint Method
287
288             //reading the paint Method
289
290             //reading the paint Method
291
292             //reading the paint Method
293
294             //reading the paint Method
295
296             //reading the paint Method
297
298             //reading the paint Method
299
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2798
2799
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2898
2899
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2998
2999
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3098
3099
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3198
3199
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3298
3299
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3398
3399
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3498
3499
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3598
3599
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3698
3699
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3798
3799
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3869
3870
3871
3872
3873
3874
3875
3876
3877
```

```

154
155
156     if (Choice == 4) {
157
158         //Enter a new process-id and its department together with its type and information relevant to the type
159
160         //asking user to enter the process ID and reading the value
161         System.out.println("Enter the Process ID ");
162         int process_id = Integer.parseInt(br.readLine());
163
164         //reading the ordered date from the user
165         System.out.println("Enter the Department Number that supervises this process");
166         int department_number = Integer.parseInt(br.readLine());
167
168         //reading the customer name from the user
169         System.out.println("Enter Process Data");
170         String process_data = br.readLine();
171
172         //reading the customer name from the user
173         System.out.println("Enter the Type of Process - paint, fit or cut ");
174         String type = br.readLine();
175
176         //if the type entered by the user is paint then asking for paint process information
177         if(type.equals("paint")) {
178
179             //reading the paint type
180             System.out.println("Enter the paint type ");
181
182             String paint_type = br.readLine();
183
184             //reading the paint Method
185             System.out.println("Enter the paint Method ");

```

individual_project [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)
 1/. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT
 ======
 4
 Enter the Process ID
 3004
 Enter the Department Number that supervises this process
 2000
 Enter Process Data
 New Process 4
 Enter the Type of Process - paint, fit or cut
 cut
 Enter the cutting type
 Sawing
 Enter the Machine type
 Underpainting
 New Process record and cut_process record are inserted successfully.
 ======
 ======You have the Following Options to Choose:=====
 1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
 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

```

154
155
156     if (Choice == 4) {
157
158         //Enter a new process-id and its department together with its type and information relevant to the type
159
160         //asking user to enter the process ID and reading the value
161         System.out.println("Enter the Process ID ");
162         int process_id = Integer.parseInt(br.readLine());
163
164         //reading the ordered date from the user
165         System.out.println("Enter the Department Number that supervises this process");
166         int department_number = Integer.parseInt(br.readLine());
167
168         //reading the customer name from the user
169         System.out.println("Enter Process Data");
170         String process_data = br.readLine();
171
172         //reading the customer name from the user
173         System.out.println("Enter the Type of Process - paint, fit or cut ");
174         String type = br.readLine();
175
176         //if the type entered by the user is paint then asking for paint process information
177         if(type.equals("paint")) {
178
179             //reading the paint type
180             System.out.println("Enter the paint type ");
181
182             String paint_type = br.readLine();
183
184             //reading the paint Method
185             System.out.println("Enter the paint Method ");

```

individual_project [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (Nov 13, 2019, 1:56:55 AM)
 15. Change the color of a given paint job
 16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT
 ======

4
 Enter the Process ID
 3005
 Enter the Department Number that supervises this process
 2000
 Enter Process Data
 New Process 5
 Enter the Type of Process - paint, fit or cut
 fit
 Enter the fit type
 Fitting flat
 New Process record and fit_process record are inserted successfully.
 ======
 ======You have the Following Options to Choose:=====
 1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
 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

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project "individual_project" with files "individual_project.java" and "sample.java".
- Code Editor:** Displays the content of "individual_project.java". The code handles various user inputs for process management, including department numbers, process types (paint or cut), and cutting methods.
- Terminal:** Shows the output of the Java application running in the terminal window. The application performs the following steps:
 - Changes the color of a given paint job.
 - Enters new customers from a data file until the file is empty.
 - Exports the customers (in name order) whose category is in a given range and outputs them to a data file instead of screen.
 - Quits.

After quitting, it prompts for a choice (4). The user enters 4, which triggers the following interaction:

```
Enter the Process ID  
3006  
Enter the Department Number that supervises this process  
2006  
Enter Process Data  
New Process 6  
Enter the Type of Process - paint, fit or cut  
cut  
Enter the cutting type  
painting  
Enter the Machine type  
Blocking in  
New Process record and cut_process record are inserted successfully.
```

Finally, it displays the following options:

```
=====You have the Following Options to Choose=====  
1. Enter a new customer  
2. Enter a new department  
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered  
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
```

The screenshot shows an IDE interface with two main windows. The top window is titled 'individual_project.java' and contains Java code for a process management application. The bottom window is titled 'Console' and displays a command-line interface for interacting with the application.

Java Code (individual_project.java):

```
145
146     if (Choice == 4) {
147
148         //Enter a new process-id and its department together with its type and information relevant to the type
149
150         //asking user to enter the process ID and reading the value
151         System.out.println("Enter the Process ID ");
152         int process_id = Integer.parseInt(br.readLine());
153
154         //reading the ordered date from the user
155         System.out.println("Enter the Department Number that supervises this process");
156         int department_number = Integer.parseInt(br.readLine());
157
158         //reading the customer name from the user
159         System.out.println("Enter Process Data");
160         String process_data = br.readLine();
161
162         //reading the customer name from the user
163         System.out.println("Enter the Type of Process - paint, fit or cut ");
164         String type = br.readLine();
165
166         //if the type entered by the user is paint then asking for paint process information
167         if(type.equals("paint")) {
168
169             //reading the paint type
170             System.out.println("Enter the paint type ");
171
172             String paint_type = br.readLine();
173
174             //reading the paint Method
175             System.out.println("Enter the painting Method ");
176
177             String paint_Method = br.readLine();
178
179             //inserting the record into the database
180             insert_record(process_id, department_number, process_data, type, paint_type, paint_Method);
181
182             System.out.println("New Process record and paint_process record are inserted successfully.");
183
184             System.out.println("-----");
185
186             System.out.println("1. Enter a new customer");
187             System.out.println("2. Enter a new department");
188             System.out.println("3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered");
189             System.out.println("4. Enter a new process record and its department together with its type and information relevant to the type");
190             System.out.println("5. Exit the application");
191
192         }
193
194     }
195
196 }
```

Console Output:

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\java.exe (Nov 13, 2019, 1:56:55 AM)
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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
4
Enter the Process ID
3007
Enter the Department Number that supervises this process
2000
Enter Process Data
New Process ?
Enter the Type of Process - paint, fit or cut
paint
Enter the paint type
Flat_Paints
Enter the painting Method
Dry_brushing
New Process record and paint_process record are inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
4. Enter a new process record and its department together with its type and information relevant to the type
5. Exit the application
```

The screenshot shows the Eclipse IDE interface with two main panes. The top pane displays a Java code editor for a file named `individual_project.java`. The code implements a menu-driven application for managing processes. The bottom pane shows a terminal window with the output of running the application, which includes prompts for entering process details like ID, department number, and type, followed by a success message indicating record insertion.

```
1. Enter a new customer  
2. Enter a new department  
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered  
4. Enter a new process id and its department together with its type and information relevant to the type  
5. Enter a new process record and fit_process record are inserted successfully.  
=====You have the Following Options to Choose:=====
```

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** The current file is `sample.java`. The code reads user input for a process ID, department number, and process data.
- Terminal Window:** The terminal shows the execution of the Java program. It prompts for a choice (4), enters a process ID (3009), and a department number (2003). It then asks for process data, which is entered as "New Process 9". The output indicates that a new process record and a cut_process record were inserted successfully.
- Output:** The terminal also displays a list of 18 options for the user to choose from, ranging from creating a new account to completing a job.

```

144
145
146     if (Choice == 4) {
147
148         //Enter a new process-id and its department together with its type and information relevant to the type
149
150
151         //asking user to enter the process ID and reading the value
152         System.out.println("Enter the Process ID ");
153         int process_id = Integer.parseInt(br.readLine());
154
155         //reading the ordered date from the user
156         System.out.println("Enter the Department Number that supervises this process");
157         int department_number = Integer.parseInt(br.readLine());
158
159         //reading the customer name from the user
160         System.out.println("Enter Process Data");
161         String process_data = br.readLine();
162
163
164
165
166
167
168
169
170
171

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 1:5655 AM)

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_id is in a given range
 15. Change the color of a given paint job
 16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

4
 Enter the Process ID
 3010
 Enter the Department Number that supervises this process
 2003
 Enter Process Data
 New Process 10
 Enter the Type of Process - paint, fit or cut
 Cut
 Enter the cutting type
 Drill
 Enter the Machine type
 Dry
 New Process record and cut_process record are inserted successfully.
 =====

===== You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new assembly
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

Process Table After Insertion:

```

31  /* SQL query for option 4 */
32  --Enter a new process-id and its department together with its type and information relevant to the type
33  select * FROM process;
34  Select * from paint_process;
35  select * from fit_process;
36  select * from cut_process;
37
38  /* SQL query for option 5 */

```

Results Messages

	process_id	department_number	process_data	account_no
1	3001	2000	New Process 1	NULL
2	3002	2000	New Process 2	NULL
3	3003	2002	New Process 3	NULL
4	3004	2000	New Process 4	NULL
5	3005	2004	New Process 5	NULL
6	3006	2002	New Process 6	NULL
7	3007	2000	New Process 7	NULL
8	3008	2004	New Process 8	NULL
9	3009	2003	New Process 9	NULL
10	3010	2003	New Process 10	NULL

	process_id	paint_type	painting_method
1	3001	Gloss Paint	Underpainting
2	3002	Satin Finish	Blocking in
3	3007	Flat Paints	Dry brushing

	process_id	fit_type
1	3003	fitting1
2	3005	Fitting flat
3	3008	fit over the top

UNSAVED

C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db ▾

```
35  select * from tit_process;
36  select * from cut_process;
37
```

Results Messages

	process_id	cutting_type	machine_type
1	3004	Sawing	Underpainting
2	3006	Broaching	Blocking in
3	3009	Drillings	Dry brushing
4	3010	Drill	Dry

6.5. Screenshots showing the testing of query 5:

Tables Before Account Creation:

l-server... C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

▶ Run □ Cancel ⚙ Disconnect ⚙ Change Connection cs-dsa-4513-sql-db ▾ | ⚙ Explain ⚙ Enable SQLCMD

```
36 select * from cut_process;
37
38 /* SQL query for option 5 */
39 --Create a new account and associate it with the process, assembly, or department to which it is applicable
40 select * from assembly_account;
41 select * from department_account;
42 select * from process_account;
43
```

Results Messages

account_no	established_date	assembly_costs
------------	------------------	----------------

account_no	established_date	department_costs
------------	------------------	------------------

account_no	established_date	process_costs
------------	------------------	---------------

Insertions:

```

273 //Depending on the type of account, asking user for details of the account
274
275 //considering when Type of account is Department
276 if(account_type.equals("Department")) {
277
278
279     //asking user to enter the department number and reading the value
280     System.out.println("Enter the department number you will be tracking with this account ");
281     String department_no = br.readLine();
282
283     //asking user to enter the account number and reading the value
284     System.out.println("Enter the account number ");
285     String account_no = br.readLine();
286
287     //reading the established_date from the user
288     System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
289     String date = br.readLine();
290
291     //reading the department_costs value given by user
292     System.out.println("Enter the department costs");
293     int costs = Integer.parseInt(br.readLine());
294
295     //Executing the query 5
296
297     //Inserting into department_account
298     final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date + "','" + costs + "')";
299
300     statement.executeUpdate(query5);
301
302     // Updating the department account as per the department number number given
303     final String query5_1 = "UPDATE department SET account_no = '" + account_no + "' WHERE department_number = '" + department_no + "'";
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900

```

File Search Project Run Window Help

individual_projectjava sample.java

```

1 //Depending on the type of account, asking user for details of the account
2 //considering when Type of account is Department
3 if(account_type.equals("Department")) {
4
5     //asking user to enter the department number and reading the value
6     System.out.println("Enter the department number you will be tracking with this account ");
7     String department_no = br.readLine();
8
9     //asking user to enter the account number and reading the value
10    System.out.println("Enter the account number ");
11    String account_no = br.readLine();
12
13    //reading the established_date from the user
14    System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
15    String date = br.readLine();
16
17    //reading the department_costs value given by user
18    System.out.println("Enter the department costs");
19    int costs = Integer.parseInt(br.readLine());
20
21    //Executing the query 5
22
23    //Inserting into department account
24    final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date + "','" + costs + "')";
25
26    statement.executeUpdate(query5);
27
28    // Updating the department account as per the department number number given
29    final String query6 = "UPDATE department SET account_no = '" + account_no + "' WHERE department_number = '" + department_no + "'";
30
31    statement.executeUpdate(query6);
32
33}

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

10. Enter the Assembly ID you will be tracking with this account
11. Retrieve the processes through which a given assembly-id has passed so far (in dateascended 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

Enter Type Of Account (Department or Assembly or Process):
Assembly
Enter the Assembly ID you will be tracking with this account
1000
Enter the account number
100
Enter the established_date - in the form of YYYY-MM-DD:
2019-06-12
Enter the Assembly costs
400
Assembly Account record inserted successfully.
=====You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
4. Enter a new process-id and its department together with its type and information relevant to the type ..

File Search Project Run Window Help

individual_projectjava sample.java

```

1 //Depending on the type of account, asking user for details of the account
2 //considering when Type of account is Department
3 if(account_type.equals("Department")) {
4
5     //asking user to enter the department number and reading the value
6     System.out.println("Enter the department number you will be tracking with this account ");
7     String department_no = br.readLine();
8
9     //asking user to enter the account number and reading the value
10    System.out.println("Enter the account number ");
11    String account_no = br.readLine();
12
13    //reading the established_date from the user
14    System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
15    String date = br.readLine();
16
17    //reading the department_costs value given by user
18    System.out.println("Enter the department costs");
19    int costs = Integer.parseInt(br.readLine());
20
21    //Executing the query 5
22
23    //Inserting into department account
24    final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date + "','" + costs + "')";
25
26    statement.executeUpdate(query5);
27
28    // Updating the department account as per the department number number given
29    final String query6 = "UPDATE department SET account_no = '" + account_no + "' WHERE department_number = '" + department_no + "'";
30
31    statement.executeUpdate(query6);
32
33}

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

Enter Type Of Account (Department or Assembly or Process):
Assembly
Enter the Assembly ID you will be tracking with this account
1000
Enter the account number
100
Enter the established_date - in the form of YYYY-MM-DD:
2019-10-01
Enter the Assembly costs
400
Assembly Account record inserted successfully.
=====You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

Search Project Run Window Help

individual_project.java samplejava

```

271     //Depending on the type of account, asking user for details of the account
272
273     //considering when Type of account is Department
274     if(account_type.equals("Department")) {
275
276         //asking user to enter the department number and reading the value
277         System.out.println("Enter the department number you will be tracking with this account ");
278         String department_no = br.readLine();
279
280         //asking user to enter the account number and reading the value
281         System.out.println("Enter the account number ");
282         String account_no = br.readLine();
283
284         //reading the established_date from the user
285         System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
286         String date = br.readLine();
287
288         //reading the department_costs value given by user
289         System.out.println("Enter the department costs");
290         int costs = Integer.parseInt(br.readLine());
291
292         //Executing the query 5
293
294         //Inserting into department account
295         final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date + "','" + costs + "')";
296
297         statement.executeUpdate(query5);
298
299         // Updating the department account as per the department number number given
300         final String query6 = "UPDATE department SET account_no = '" + account_no + "' WHERE department_number = '" + department_no + "'";
301
302     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

5 Enter Type Of Account (Department or Assembly or Process):
Assembly
 Enter the Assembly ID you will be tracking with this account
 1002
 Enter the account number
 10
 Enter the established_date - in the form of YYYY-MM-DD:
 2019-01-15
 Enter the Assembly costs
 900
 Assembly Account record inserted successfully.
 ======You have the Following Options to Choose:=====

1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
 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

Search Project Run Window Help

individual_project.java samplejava

```

271     //Depending on the type of account, asking user for details of the account
272
273     //considering when Type of account is Department
274     if(account_type.equals("Department")) {
275
276         //asking user to enter the department number and reading the value
277         System.out.println("Enter the department number you will be tracking with this account ");
278         String department_no = br.readLine();
279
280         //asking user to enter the account number and reading the value
281         System.out.println("Enter the account number ");
282         String account_no = br.readLine();
283
284         //reading the established_date from the user
285         System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
286         String date = br.readLine();
287
288         //reading the department_costs value given by user
289         System.out.println("Enter the department costs");
290         int costs = Integer.parseInt(br.readLine());
291
292         //Executing the query 5
293
294         //Inserting into department account
295         final String query5 = "INSERT INTO department_account values('" + account_no + "','" + date + "','" + costs + "')";
296
297         statement.executeUpdate(query5);
298
299         // Updating the department account as per the department number number given
300         final String query6 = "UPDATE department SET account_no = '" + account_no + "' WHERE department_number = '" + department_no + "'";
301
302     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

=====

5 Enter Type Of Account (Department or Assembly or Process):
Process
 Enter the process ID you will be tracking with this account
 3001
 Enter the account number
 301
 Enter the established_date - in the form of YYYY-MM-DD:
 2019-04-12
 Enter the Process costs
 120
 Process Account record inserted successfully.
 ======You have the Following Options to Choose:=====

1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
 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

File Search Project Run Window Help

Quick Access

individual_project.java 3 sample.java

```

348     //considering when Type of account is Process
349     else if(account_type.equals("Process")) {
350
351         //asking the process ID from User and reading the value
352         System.out.println("Enter the process ID you will be tracking with this account ");
353         String process_id = br.readLine();
354
355
356         //asking user to enter the account and reading the value
357         System.out.println("Enter the account number ");
358         String account_no = br.readLine();
359
360
361         //reading the established_date from the user
362         System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
363         String date = br.readLine();
364
365
366         //reading the process costs value given by user
367         System.out.println("Enter the Process costs");
368         int costs = Integer.parseInt(br.readLine());
369
370         //Executing the query 5
371
372         final String query5 = "INSERT INTO process_account values('" + account_no + "','" + date + "','" + costs + "')";
373
374         statement.executeUpdate(query5);
375
376         final String query5_1= "UPDATE process SET account_no = '" + account_no + "' WHERE process_id = '" + process_id + "'";
377
378
379

```

Markers Properties Servers Data Source Explorer Snippets Console

individual.project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

 5 Enter Type Of Account (Department or Assembly or Process):
 Process
 Enter the process ID you will be tracking with this account
 3002
 Enter the account number
 302
 Enter the established_date - in the form of YYYY-MM-DD:
 2019-03-02
 Enter the Process costs
 600
 Process Account record inserted successfully.
 *****You have the Following Options to Choose:
 1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
 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

Search Project Run Window Help

Quick Access

individual_project.java 3 sample.java

```

348     //considering when Type of account is Process
349     else if(account_type.equals("Process")) {
350
351         //asking the process ID from User and reading the value
352         System.out.println("Enter the process ID you will be tracking with this account ");
353         String process_id = br.readLine();
354
355
356         //asking user to enter the account and reading the value
357         System.out.println("Enter the account number ");
358         String account_no = br.readLine();
359
360
361         //reading the established_date from the user
362         System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
363         String date = br.readLine();
364
365
366         //reading the process costs value given by user
367         System.out.println("Enter the Process costs");
368         int costs = Integer.parseInt(br.readLine());
369
370         //Executing the query 5
371
372         final String query5 = "INSERT INTO process_account values('" + account_no + "','" + date + "','" + costs + "')";
373
374         statement.executeUpdate(query5);
375
376         final String query5_1= "UPDATE process SET account_no = '" + account_no + "' WHERE process_id = '" + process_id + "'";
377
378
379

```

Markers Properties Servers Data Source Explorer Snippets Console

individual.project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 12:38:46 PM)

16. Import: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

 5 Enter Type Of Account (Department or Assembly or Process):
 Process
 Enter the process ID you will be tracking with this account
 3003
 Enter the account number
 303
 Enter the established_date - in the form of YYYY-MM-DD:
 2019-09-13
 Enter the Process costs
 700
 Process Account record inserted successfully.
 *****You have the Following Options to Choose:
 1. Enter a new customer
 2. Enter a new department
 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered

Navigate Search Project Run Window Help

individual_project.java sample.java

```

348     //considering when Type of account is Process
349     else if(account_type.equals("Process")) {
350
351         //asking the process ID from User and reading the value
352         System.out.println("Enter the process ID you will be tracking with this account ");
353         String process_id = br.readLine();
354
355
356         //asking user to enter the account and reading the value
357         System.out.println("Enter the account number ");
358         String account_no = br.readLine();
359
360
361         //reading the established_date from the user
362         System.out.println("Enter the established_date - in the form of YYYY-MM-DD:");
363         String date = br.readLine();
364
365
366         //reading the process costs value given by user
367         System.out.println("Enter the Process costs");
368         int costs = Integer.parseInt(br.readLine());
369
370         //Executing the query
371
372         final String query5 = "INSERT INTO process_account values('" + account_no + "','" + date + "','" + costs + "')";
373         statement.executeUpdate(query5);
374
375
376         final String query5_1= "UPDATE process SET account_no = '" + account_no + "' WHERE process_id = '" + process_id + "'";
377
378
379

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019 12:38:46 PM)

16. Import: enter new customers from a data file until the file is empty

17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen

18. QUIT

=====

5

Enter Type Of Account (Department or Assembly or Process):

Process

Enter the process ID you will be tracking with this account

3004

Enter the account number

3004

Enter the established_date - in the form of YYYY-MM-DD:

2019-07-13

Enter the Process costs

530

Process Account record inserted successfully.

=====

=====You have the Following Options to Choose:=====

1. Enter a new account

2. Enter a new department

3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered

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

Account Tables After the Insertions:

JS ID_project_tables.sql - sagi00...gi0000 DROP_TABLES.sql - sagi00...gi0000 Insertions_IP.sql - sagi00...gi0000

C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```

40 --Create a new account and associate it with the process, assembly, or department to which it is associated
41 select * from assembly_account;
42 select * from department_account;
43 select * from process_account;
44

```

Results

	account_no	established_date	assembly_costs
1	100	2019-06-12	120
2	101	2019-10-01	400
3	102	2019-01-15	900

	account_no	established_date	department_costs
1	200	2019-06-12	120
2	201	2019-11-02	300
3	202	2019-04-13	400

	account_no	established_date	process_costs
1	301	2019-04-12	120
2	302	2019-03-02	600
3	303	2019-09-13	700
4	304	2019-07-13	530

Since we are required to have account for each assembly, each process and each department. After the Insertion of 10 accounts, I have created 15 more accounts and associated with the type of the account. The below Screen shot shows the all the accounts available after this insertion.

The screenshot shows the Azure Data Studio interface with three tables displayed in the Results tab:

- Assembly_Account** (Top Table):

	account_no	established_date	assembly_costs
1	100	2019-09-22	200
2	101	2019-10-01	400
3	102	2019-01-15	900
4	103	2019-02-18	800
5	104	2019-03-06	100
6	105	2019-04-30	300
7	106	2019-06-30	500
8	107	2019-07-19	230
9	108	2019-08-03	250
10	109	2019-06-03	250

- Department_Account** (Middle Table):

	account_no	established_date	department_costs
1	200	2019-06-12	120
2	201	2019-11-02	300
3	202	2019-04-13	400
4	203	2019-03-13	500
5	204	2019-05-21	400

- Process_Account** (Bottom Table):

	account_no	established_date	process_costs
1	301	2019-04-12	120
2	302	2019-03-02	600
3	303	2019-09-13	700
4	304	2019-07-13	530
5	305	2019-08-04	660
6	306	2019-05-21	470
7	307	2019-07-12	625
8	308	2019-04-11	755
9	309	2019-06-17	160
10	310	2019-11-13	512

6.6. Screenshots showing the testing of query 6:

Table Before Job Entry

```

53    select * from job;
54
55

```

Insertions:

```

if(Choice == 6) {
    //Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
    //asking user to enter the job number and reading the value
    System.out.println("Enter the Job Number");
    int job_number = Integer.parseInt(br.readLine());

    //Reading the commencement date
    System.out.println("Enter the Job Commencement date - in the form of YYYY-MM-DD:");
    String date = br.readLine();

    //reading the assembly ID value given by user
    System.out.println("Enter the Assembly ID");
    int assembly_id = Integer.parseInt(br.readLine());

    //reading the process ID value given by user
    System.out.println("Enter the process ID");
    int process_id = Integer.parseInt(br.readLine());

    //Executing the query 6
    //Inserting the values into Job table
    final String query6 = "INSERT INTO job(job_no,commence_date,assembly_id,process_id) values('" + job_number + "','" + date + "','" + assembly_id + "','" + process_id + "')";
    statement.executeUpdate(query6);
}

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 4:49:06 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

6
Enter the Job Number
5001
Enter the Job Commencement date - in the form of YYYY-MM-DD:
2019-05-11
Enter the Assembly ID
1000
Enter the process ID
3001
New Job record inserted successfully.
=====You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

 1. Navigate Search Project Run Window Help
 2. File Edit View Insert Tools Window Help
 3. individual_project.java sample.java
 4. 391     if(Choice == 6) {
 5.         //Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
 6.         //asking user to enter the job number and reading the value
 7.         System.out.println("Enter the Job Number");
 8.         int job_number = Integer.parseInt(br.readLine());
 9. 
10.        //Reading the commencement date
11.        System.out.println("Enter the Job Commencement date - in the form of YYYY-MM-DD:");
12.        String date = br.readLine();
13. 
14.        //reading the assembly ID value given by user
15.        System.out.println("Enter the Assembly ID");
16.        int assembly_id = Integer.parseInt(br.readLine());
17. 
18.        //reading the process ID value given by user
19.        System.out.println("Enter the process ID");
20.        int process_id = Integer.parseInt(br.readLine());
21. 
22.        //Executing the query 6
23. 
24.        //Inserting the values into Job table
25.        final String query6 = "INSERT INTO job(job_no,commence_date,assembly_id,process_id) values('" + job_number + "','" + date + "','" + assembly_id + "','" + process_id + "')";
26. 
27.        statement.executeUpdate(query6);
28. 
29.        System.out.println("New Job record inserted successfully ");
30. 
31. =====
32. 
33. Enter the Job Number
34. 5002
35. Enter the Job Commencement date - in the form of YYYY-MM-DD:
36. 2019-05-11
37. Enter the Assembly ID
38. 1000
39. Enter the process ID
40. 3000
41. New Job record inserted successfully.
42. =====
43. =====You have the Following Options to Choose:=====
44. 1. Enter a new customer
45. 2. Enter a new department
46. 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
47. 4. Enter a new process-id and its department together with its type and information relevant to the type
48. 5. Create a new account and associate it with the process, assembly, or department to which it is applicable
49. 6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
50. 7. At the completion of a job, enter the date it completed and the information relevant to the type of job
51. 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
52. 9. Retrieve the cost incurred on an assembly-id
53. 10. Retrieve the total labor time within a department for jobs completed in the department during a given date
54. 11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
55. 12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
56. 13. Retrieve the customers (in name order) whose category is in a given range
57. 14. Delete all cut jobs whose job-no's in a given range
58. 15. Change the color of a given saint id
59. 
60. =====
61. 
62. individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 4:49:06 PM)
63. 18. QUIT
64. 
65. =====
66. 
67. Enter the Job Number
68. 5003
69. Enter the Job Commencement date - in the form of YYYY-MM-DD:
70. 2019-05-10
71. Enter the Assembly ID
72. 1000
73. Enter the process ID
74. 3000
75. New Job record inserted successfully.
76. =====
77. =====You have the Following Options to Choose:=====
78. 1. Enter a new customer
79. 2. Enter a new department
80. 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
81. 4. Enter a new process-id and its department together with its type and information relevant to the type
82. 5. Create a new account and associate it with the process, assembly, or department to which it is applicable
83. 6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
84. 7. At the completion of a job, enter the date it completed and the information relevant to the type of job
85. 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
86. 9. Retrieve the cost incurred on an assembly-id
87. 10. Retrieve the total labor time within a department for jobs completed in the department during a given date
88. 11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
89. 12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
90. 13. Retrieve the customers (in name order) whose category is in a given range
91. 
92. =====
93. 
94. individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 4:49:06 PM)
95. 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
96. 18. QUIT
97. 
98. =====
99. 
100. Enter the Job Number
101. 5003
102. Enter the Job Commencement date - in the form of YYYY-MM-DD:
103. 2019-05-10
104. Enter the Assembly ID
105. 1000
106. Enter the process ID
107. 3000
108. New Job record inserted successfully.
109. =====
110. =====You have the Following Options to Choose:=====
111. 1. Enter a new customer
112. 2. Enter a new department
113. 3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
114. 4. Enter a new process-id and its department together with its type and information relevant to the type
115. 5. Create a new account and associate it with the process, assembly, or department to which it is applicable
116. 6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
117. 7. At the completion of a job, enter the date it completed and the information relevant to the type of job
118. 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
119. 9. Retrieve the cost incurred on an assembly-id
120. 10. Retrieve the total labor time within a department for jobs completed in the department during a given date
121. 11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
122. 12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
123. 13. Retrieve the customers (in name order) whose category is in a given range

```


The screenshot shows a Java IDE interface with a code editor and a terminal window.

Code Editor:

```
if(Choice == 6) {
    //Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
    //asking user to enter the job number and reading the value
    System.out.println("Enter the Job Number");
    int job_number = Integer.parseInt(br.readLine());

    //Reading the commencement date
    System.out.println("Enter the Job Commencement date - in the form of YYYY-MM-DD:");
    String date = br.readLine();

    //reading the assembly ID value given by user
    System.out.println("Enter the Assembly ID");
    int assembly_id = Integer.parseInt(br.readLine());
}
```

Terminal Window:

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 4:49:06 PM)
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 cu
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 datecommenced order) and the department responsibl
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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scre
18. QUIT
=====
6
Enter the Job Number
5006
Enter the Job Commencement date - in the form of YYYY-MM-DD:
2019-05-13
Enter the Assembly ID
1002
Enter the process ID
3006
New Job record inserted successfully.
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 new department
9. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
10. Enter a new process-id and its department together with its type and information relevant to the type
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer (left):** Shows a project named "individual_project.java" with a file "samplejava.java". The file contains Java code for a menu-driven application.
- Code Editor (top center):** Displays the Java code for "samplejava.java". The code includes logic for creating a new job, entering job number, assembly ID, and commencement date, and printing them to the console.
- Terminal Window (bottom center):** Shows the execution of the Java program. The user enters job number 6, assembly ID 1003, and commencement date 2019-05-09. The program then lists various menu options and ends with a quit message.
- Status Bar (bottom):** Shows the path "individual_project.java Application C:\Program Files\Java\jdk-13\bin\javaw.exe Nov 14, 2019, 4:49:06 PM".

```
individual_project.java 33 sample.java
379     }
380     if(Choice == 6) {
381         //Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
382
383         //asking user to enter the job number and reading the value
384         System.out.println("Enter the Job Number");
385         int job_number = Integer.parseInt(br.readLine());
386
387         //Reading the commencement date
388         System.out.println("Enter the Job Commencement date - in the form of YYYY-MM-DD:");
389         String date = br.readLine();
390
391         //reading the assembly ID value given by user
392         System.out.println("Enter the Assembly ID");
393         int assembly_id = Integer.parseInt(br.readLine());
394
395
396
397
398
399
400
401
402
403
404
405
<
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 4:49:06 PM)
11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
6
Enter the Job Number
5009
Enter the Job Commencement date - in the form of YYYY-MM-DD:
2019-05-04
Enter the Assembly ID
1001
Enter the process ID
3005
New Job record inserted successfully.
=====
===== You have the Following Options to Choose: =====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer (left):** Shows the project structure with files like `individual_project.java`, `sample.java`, `object.java`, `project.java`, and `assembly_id.java`.
- Code Editor (top):** Displays the `individual_project.java` file containing Java code for a job management system.
- Console (bottom):** Shows the execution of the program and its output. The output includes:
 - Instructions for entering job details (job number, commencement date, assembly ID).
 - A message indicating a new job record was inserted successfully.
 - A prompt for choosing from 12 options related to customer, assembly, and department management.

Job Table After the Insertions :

5 UNSAVED

ID_project_tables.sql - sagi00...gi0000) DROP_TABLES.sql - sagi00...gi0000) Insertions_IP.sql - sagi00...gi0000) Quer...

C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel ⚡ Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

```
/* SQL query for option 6 */
--Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
select * from job;
```

Results Messages

	job_no	commence_date	completion_date	assembly_id	process_id
1	5001	2019-05-11	NULL	1000	3001
2	5002	2019-05-11	NULL	1000	3004
3	5003	2019-05-10	NULL	1000	3007
4	5004	2019-05-10	NULL	1002	3002
5	5005	2019-05-14	NULL	1002	3003
6	5006	2019-05-13	NULL	1002	3006
7	5007	2019-05-09	NULL	1003	3009
8	5008	2019-05-03	NULL	1003	3010
9	5009	2019-05-04	NULL	1001	3005
10	5010	2019-05-11	NULL	1001	3008

6.7. Screenshots showing the testing of query 7

Job Type Tables before Insertion:

```

62 select * from job;
63 select * from paint_job;
64 select * from fit_job;
65 select * from cut_job;
66
67
68
69 DELETE FROM [dbo].[paint_job]
70 WHERE paint_job_no BETWEEN 5001 AND 5003 /* add search conditions here */
71 GO
72
    
```

Results	Messages										
<table border="1"> <thead> <tr> <th>paint_job_no</th> <th>color</th> <th>volume</th> <th>labor_time</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td></tr> </tbody> </table>	paint_job_no	color	volume	labor_time							
paint_job_no	color	volume	labor_time								
<table border="1"> <thead> <tr> <th>fit_job_no</th> <th>labor_time</th> </tr> </thead> <tbody> <tr><td></td><td></td></tr> </tbody> </table>	fit_job_no	labor_time									
fit_job_no	labor_time										
<table border="1"> <thead> <tr> <th>cut_job_no</th> <th>type_of_machine</th> <th>time_used</th> <th>material_used</th> <th>labor_time</th> </tr> </thead> <tbody> <tr><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	cut_job_no	type_of_machine	time_used	material_used	labor_time						
cut_job_no	type_of_machine	time_used	material_used	labor_time							

Insertions:

```

433
434
435 //asking user for the type of job
436 System.out.println("Enter the type of Job - paint or fit or cut");
437 String type = br.readLine();
438
439 //considering when type is paint
440 if(type.equals("paint")) {
441
442     //reading the paint color
443     System.out.println("Enter the color ");
444     String color = br.readLine();
445
446     //reading the paint volume
447     System.out.println("Enter the volume");
448     float volume = Float.valueOf(br.readLine()).floatValue();
449
450     //reading the labor time from the user for this job
451     System.out.println("Enter the labor time");
452     float labor_time = Float.valueOf(br.readLine()).floatValue();
453
454     //Executing the query
455
456     //updating the job table as per the completion date given
457     final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
458     statement.executeUpdate(query7);
459
460     //inserting the values of Paint Job
461     final String query7_1 = "INSERT INTO paint_job values('" + job_no + "','" + color + "','" + volume + "','" + labor_time + "')";
462     statement.executeUpdate(query7_1);
        
```

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:38:42 PM)

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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

7
Enter the Job number which got completed
5001
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-17
Enter the type of Job - paint or fit or cut
paint
Enter the color
blue
Enter the volume
3
Enter the labor time
12.5
Completion date has been updated and paint job record is inserted

Search Project Run Window Help

individual_project.java sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = " + completion_date + " WHERE job_no = " + job_no + "";
457
458         statement.executeUpdate(query7);
459

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

```

=====
7 Enter the Job number which got completed
5002
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-17
Enter the type of Job - paint or fit or cut
paint
Enter the color
green
Enter the volume
3
Enter the labor time
12
Completion date has been updated and paint job record is inserted
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

Search Project Run Window Help

individual_project.java sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = " + completion_date + " WHERE job_no = " + job_no + "";
457
458         statement.executeUpdate(query7);
459

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

```

=====
14. Delete all cut-jobs whose job-no is in a given range
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
7 Enter the Job number which got completed
5003
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-17
Enter the type of Job - paint or fit or cut
paint
Enter the color
red
Enter the volume
3
Enter the labor time
12
Completion date has been updated and paint job record is inserted
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like `individual_project.java`, `project.java`, and `sample.java`.
- Code Editor:** Displays the `individual_project.java` file containing Java code for a paint job application.
- Console Output:** Shows the execution of the application, including user input and system output. The user inputs completion date (2019-05-17), job type (fit), and labor time (12.5). The application then lists options for the user to choose from.

```
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
1. Enter the Job number which got completed
5004
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-17
Enter the type of Job - paint or fit or cut
fit
Enter the labor time
12.5
Completion date has been updated and paint job record is inserted
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datem 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
```

gate Search Project Run Window Help

individual_project.java sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query ?
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
457
458         statement.executeUpdate(query7);

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

7
Enter the Job number which got completed
5006
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-16
Enter the type of Job - paint or fit or cut
fit
Enter the labor time
12
Completion date has been updated and paint job record is inserted
=====You have the Following Options to Choose=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

Search Project Run Window Help

individual_project.java sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query ?
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
457
458         statement.executeUpdate(query7);

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

7
Enter the Job number which got completed
5007
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-11
Enter the type of Job - paint or fit or cut
fit
Enter the labor time
15
Completion date has been updated and paint job record is inserted
=====You have the Following Options to Choose=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced order) and the department responsible for each process

sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query ?
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
457
458         statement.executeUpdate(query7);
459     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

14. Delete all cut-jobs whose job-no is in a given range
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

7
Enter the Job number which got completed
5000
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-11
Enter the type of Job - paint or fit or cut
cut
Enter the type of Machine used
Sawing
Enter the amount of time machine was used(in hours)
12
Enter the Material Used
Steel
Enter the labor time
10
Completion date has been updated and paint job record is inserted
=====You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 compilation of a job enter the data it generated and the information relevant to the type of job

sample.java

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query ?
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
457
458         statement.executeUpdate(query7);
459     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)

13. Retrieve all 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

7
Enter the Job number which got completed
5000
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-16
Enter the type of Job - paint or fit or cut
cut
Enter the type of Machine used
Broaching
Enter the amount of time machine was used(in hours)
12
Enter the Material Used
cast iron
Enter the labor time
12
Completion date has been updated and paint job record is inserted
=====You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

434     //asking user for the type of job
435     System.out.println("Enter the type of Job - paint or fit or cut");
436     String type = br.readLine();
437
438     //considering when type is paint
439     if(type.equals("paint")) {
440
441         //reading the paint color
442         System.out.println("Enter the color ");
443         String color = br.readLine();
444
445         //reading the paint volume
446         System.out.println("Enter the volume");
447         float volume = Float.valueOf(br.readLine()).floatValue();
448
449         //reading the labor time from the user for this job
450         System.out.println("Enter the labor time");
451         float labor_time = Float.valueOf(br.readLine()).floatValue();
452
453         //Executing the query
454
455         //updating the job table as per the completion date given
456         final String query7 = "UPDATE job SET completion_date = '" + completion_date + "' WHERE job_no = '" + job_no + "'";
457
458         statement.executeUpdate(query7);
        }
    }

Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 5:41:02 PM)
18. QUIT
=====
7
Enter the Job number which got completed
5010
Enter the Job Completion date - in the form of YYYY-MM-DD:
2019-05-15
Enter the type of Job - paint or fit or cut
cut
Enter the type of Machine used
Drilling
Enter the amount of time machine was used(in hours)
12
Enter the Material Used
metal
Enter the labor time
12
Completion date has been updated and paint job record is inserted
=====
You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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, gives its job-no, assembly-id, process-id, and date the job commenced
7. At the completion of a job, gives 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 datecommenced order) and the department responsible for each process

```

Tables after Insertion:

• Queries_IP.sql - sagi0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (sagi0000) - Azure Data Studio

UNSAVED

C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel ⚡ Disconnect ⚡ Change Connection cs-dsa-4513-sql-db | Explain Enable SQLCMD

```

63     select * from paint_job;
64     select * from fit_job;
65     select * from cut_job;
66

```

Results

	fit_job_no	labor_time
1	5004	12.5
2	5005	13
3	5006	12
4	5007	15

	cut_job_no	type_of_machine	time_used	material_used	labor_time
1	5008	Sawing	12	Steel	10
2	5009	Broaching	12	cast iron	12
3	5010	Drilling	12	metal	12

ID_project_tables.sql - sagi00...gi0000)
DROP_TABLES.sql - sagi00...gi0000)
Insertions

UNSAVED
C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run
Cancel
Disconnect
Change Connection
cs-dsa-4513-sql-db
Explain
Enable SQL

```

62 select * from job;
63 select * from paint_job;
64 select * from fit_job;
65 select * from cut_job;

```

Results
Messages

	job_no	commence_date	completion_date	assembly_id	process_id
1 - disco...	5001	2019-05-11	2019-05-17	1000	3001
3 - disco...	5002	2019-05-11	2019-05-17	1000	3004
2 - disco...	5003	2019-05-10	2019-05-17	1000	3007
	5004	2019-05-10	2019-05-17	1002	3002
	5005	2019-05-14	2019-05-16	1002	3003
	5006	2019-05-13	2019-05-16	1002	3006
	5007	2019-05-09	2019-05-11	1003	3009
	5008	2019-05-03	2019-05-11	1003	3010
	5009	2019-05-04	2019-05-16	1001	3005
	5010	2019-05-11	2019-05-15	1001	3008

	paint_job_no	color	volume	labor_time
1	5001	blue	3	12.5
2	5002	green	3	12
3	5003	red	3	12

PROBLEMS
4
OUTPUT
TASKS
TERMINAL

6.8. Screenshots showing the testing of query 8

Cost_transaction Table Before Insertions:

The screenshot shows the SSMS interface with the following details:

- Tab bar: ID_project_tables.sql - sagi00...gi0000, DROP_TABLES.sql - sagi00...gi0000, Insertions_IP.sql - sagi00...gi0000, Queries_IP.sql.
- Toolbar: Run, Cancel, Disconnect, Change Connection (set to cs-dsa-4513-sql-db), Explain, Enable SQLCMD.
- Query window:

```
71 GO
72
73 /* SQL query for option 8 */
74 --Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to th
75
76
77 select * from cost_transactions;
78
79
```
- Results tab: Shows a table structure with columns: transaction_no, job_no, sup_cost, asse_account_no, dep_account_no, pro_account_no.

Insertions:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer: individual_project.java, sample.java.
- Code Editor:

```
531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('" + trans_no + "','" + job_no + "','" + sup_cost + "')";
553
554         statement.executeUpdate(query8);
555
556         //update cost transcaions with the specific assembly account as per the given job number
557     }
```
- Console tab: Displays the execution of the Java application and the resulting output. The output includes:

```
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
8
Enter the transaction Number
9001
Enter the job Number on which this transaction is recoded
5001
Enter the sup-cost
50
Transaction number is Inserted and All the affected Accounts costs are updated
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
```

```

531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('"+ trans_no +"','"+ job_no +"','"+ sup_cost +"')";
553
554         statement.executeUpdate(query8);
555
556         //updating cost transactions with the specific assembly account as per the given job number

```

```

Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
8
Enter the transaction Number
9002
Enter the job Number on which this transaction is recoded
5002
Enter the sup-cost
70
Transaction number is Inserted and All the affected Accounts costs are updated
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

```

e Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
8
Enter the transaction Number
9003
Enter the job Number on which this transaction is recoded
5003
Enter the sup-cost
70
Transaction number is Inserted and All the affected Accounts costs are updated
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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

```

```
File Search Project Run Window Help
individual_project.java sample.java
531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the Job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('" + trans_no + "','" + job_no + "','" + sup_cost + "')";
553
554         statement.executeUpdate(query8);
555
556         //Updating cost transactions with the specific assembly account as per the given job number
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)

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: Enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

```
=====
8
Enter the transaction Number
9004
Enter the job Number on which this transaction is recoded
5004
Enter the sup-cost
120
Transaction number is Inserted and All the affected Accounts costs are updated
=====
======You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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
```



The screenshot shows the Eclipse IDE interface with the following details:

- Menu Bar:** File, Search, Project, Run, Window, Help.
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Select All, and others.
- Project Explorer:** Shows "individual_project.java" as the selected project.
- Editor:** Displays the Java code for "sample.java".
- Output View:** Shows some log output.
- Console View:** Shows the command line interface.

The code in the editor is as follows:

```
531
532
533     if((Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('"+ trans_no +"','"+ job_no +"','"+ sup_cost +"')";
553
554         statement.executeUpdate(query8);
555
556         //Updating cost traansactions with the specific assembly account as per the given job number
```

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)
17. Export: enter new customers from a data file until the file is empty
```

```
18. QUIT
=====
0
Enter the transaction Number
9005
Enter the job Number on which this transaction is recorded
5005
Enter the sup-cost
50
[transaction number is Inserted and All the affected Accounts costs are updated
=====
=====You have the Following Options to Choose:
=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 assembly-id
10. Retrieve the total labor time within a department
11. Retrieved the processes through which a given assembly-id has passed so far (in datecommencing order) and the department responsible for each process
12. Retrive the department name, department-id, and department-name for a given department
```

```

531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('" + trans_no + "','" + job_no + "','" + sup_cost + "')";
553
554         statement.executeUpdate(query8);
555
556         //updating cost transactions with the sncrific assembly account as per the given inh number

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)

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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

8
 Enter the transaction Number
 9006
 Enter the job Number on which this transaction is recoded
 5006
 Enter the sup-cost
 30
 Transaction number is Inserted and All the affected Accounts costs are upated

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

```

531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading Job Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('" + trans_no + "','" + job_no + "','" + sup_cost + "')";
553
554         statement.executeUpdate(query8);
555
556         //updating cost transactions with the sncrific assembly account as per the given inh number

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)

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 datecommenced 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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

9
 Enter the transaction Number
 0007
 Enter the job Number on which this transaction is recoded
 5007
 Enter the sup-cost
 50
 Transaction number is Inserted and All the affected Accounts costs are upated

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

The screenshot shows a Java application running in an IDE. The code in the editor is for a program that handles transactions and cost calculations. The terminal window below shows the execution of the application, starting with a menu of 18 options. The user selects option 8, which prompts for a transaction number, job number, and sup-cost. The application then inserts the data into a database table and provides feedback.

```
531
532
533     if(Choice == 8) {
534
535         // 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
536
537         //asking user to enter the transaction number and reading it
538         System.out.println("Enter the transaction Number");
539         int trans_no= Integer.parseInt(br.readLine());
540
541         //reading JOB Number on which this transaction is recoded
542         System.out.println("Enter the job Number on which this transaction is recoded");
543         int job_no= Integer.parseInt(br.readLine());
544
545         //reading the sup-cost from the user
546         System.out.println("Enter the sup-cost");
547         float sup_cost = Float.valueOf(br.readLine()).floatValue();
548
549         //Executing the query 8
550
551         // inserting the values into the cost transactions table
552         final String query8 = "INSERT INTO cost_transactions(transaction_no,job_no,sup_cost) values('"+trans_no+"','"+job_no+"','"+sup_cost+"')";
553
554         statement.executeUpdate(query8);
555
556         //Updating cost transactions with the snarific assembly account as per the given job number
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
599
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\java.exe (NoV14, 2019, 7:47:58 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

8
Enter the transaction Number
9010
Enter the job Number on which this transaction is recoded
5010
Enter the sup-cost
100
Transaction number is Inserted and All the affected Accounts costs are updated

===== You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 dateordered order) and the department responsible for each process

ID_project_tables.sql - sagi00...gi0000
DROP_TABLES.sql - sagi00...gi0000
Insertions_IP.sql - sagi00...gi0000
Queries_IP.sql - sagi00...g

5 UNSAVED C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run
Cancel
Disconnect
Change Connection
cs-dsa-4513-sql-db
▼
Explain
Enable SQLCMD

```

t_tables.sql...
BLES.sql - ...
IP.sql - s...
P.sql - ... 4
pl - discon...
l1 - disco...
l3 - disco...
l2 - disco...
NED
Results Messages

```

	transaction_no	job_no	sup_cost	asse_account_no	dep_account_no	pro_account_no
1	9001	5001	50	100	200	301
2	9002	5002	60	100	200	304
3	9003	5003	70	100	200	307
4	9004	5004	120	102	200	302
5	9005	5005	50	102	202	303
6	9006	5006	30	102	202	306
7	9007	5007	50	103	203	309
8	9008	5008	40	103	203	310
9	9009	5009	60	101	204	305
10	9010	5010	100	101	204	308

PROBLEMS
4
OUTPUT
TASKS
TERMINAL

Showing Database access is Possible:

6.9. Screenshots showing the testing of query 9

The screenshot shows the Eclipse IDE interface. At the top, there are two tabs: "individual_project.java" and "sample.java". Below the tabs, the code for "individual_project.java" is displayed, specifically the section for executing query 9. The code uses a JDBC connection to retrieve assembly costs based on an assembly ID entered by the user.

```
605     }
606
607     if (Choice == 9) {
608         //Retrieve the cost incurred on an assembly-id
609
610         //asking user to enter the assembly ID and reading the value
611         System.out.println("Enter the assembly ID on which you want to retrieve the costs ");
612
613         int assembly_id = Integer.parseInt(br.readLine());
614
615
616         //Executing the query 9
617
618         //Retrieving the assembly_costs
619         final String query9 = "SELECT assembly_costs FROM assembly_account WHERE account_no IN (SELECT account_no FROM assembly WHERE assembly_id = '" + assembly_id + "')";
620
621         ResultSet rs = statement.executeQuery(query9);
622
623         //since it is a result set, printing all the values
624         while (rs.next()) {
625             System.out.println(rs.getString(1));
626
627             System.out.println("Costs Incurred on the given Assembly ID are retrieved");
628             System.out.println("=====");
629     }
```

Below the code editor is a terminal window titled "Console". It shows the command-line output of the Java application. The application prints a list of 18 numbered options, then prompts the user for an assembly ID. The user enters "1000", and the application responds with "380.0" and a message indicating the costs are retrieved. Finally, it displays a menu of six options for the user to choose from.

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
9
Enter the assembly ID on which you want to retrieve the costs
1000
380.0
Costs Incurred on the given Assembly ID are retrieved
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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
```

```

605
606
607     if (Choice == 9) {
608         //Retrieve the cost incurred on an assembly-id
609
610         //asking user to enter the assembly ID and reading the value
611         System.out.println("Enter the assembly ID on which you want to retrieve the costs ");
612
613         int assembly_id = Integer.parseInt(br.readLine());
614
615
616         //Executing the query 9
617
618         //Retrieving the assembly_costs
619         final String query9 = "SELECT assembly_costs FROM assembly_account WHERE account_no IN (SELECT account_no FROM assembly WHERE assembly_id = '" + assembly_id + "')";
620
621         ResultSet rs = statement.executeQuery(query9);
622
623         //since it is a result set, printing all the values
624         while (rs.next()) {
625             System.out.println(rs.getString(1));
626
627         System.out.println("Costs Incurred on the given Assembly ID are retrieved");
628         System.out.println("=====");
629     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

9
Enter the assembly ID on which you want to retrieve the costs
1009
1009.0
Costs Incurred on the given Assembly ID are retrieved
=====

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 total labor time within a department
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 datecommenced 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

```

605
606
607     if (Choice == 9) {
608         //Retrieve the cost incurred on an assembly-id
609
610         //asking user to enter the assembly ID and reading the value
611         System.out.println("Enter the assembly ID on which you want to retrieve the costs ");
612
613         int assembly_id = Integer.parseInt(br.readLine());
614
615
616         //Executing the query 9
617
618         //Retrieving the assembly_costs
619         final String query9 = "SELECT assembly_costs FROM assembly_account WHERE account_no IN (SELECT account_no FROM assembly WHERE assembly_id = '" + assembly_id + "')";
620
621         ResultSet rs = statement.executeQuery(query9);
622
623         //since it is a result set, printing all the values
624         while (rs.next()) {
625             System.out.println(rs.getString(1));
626
627         System.out.println("Costs Incurred on the given Assembly ID are retrieved");
628         System.out.println("=====");
629     }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 7:47:58 PM)

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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

9
Enter the assembly ID on which you want to retrieve the costs
1009
1009.0
Costs Incurred on the given Assembly ID are retrieved
=====

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

6.10. Screenshots showing the testing of query 10

```

629
630     }
631
632     if(Choice == 10) {
633
634         //asking department number from user and reading it
635         System.out.println("Enter the Department Number");
636         int department_no= Integer.parseInt(br.readLine());
637         //Asking user to enter the completion date
638         System.out.println("Enter the completion Date - in the form of YYYY-MM-DD:");
639         String date = br.readLine();
640         final String query10 ="EXEC query10 @department_number = '"+department_no+"', @completion_date = '"+date+"'";
641         ResultSet rs = statement.executeQuery(query10);
642         System.out.println("Total Labor Hours");
643
644         //since it is a result set, printing all the values
645         while (rs.next()) {
646             System.out.println(rs.getInt(1));
647         }
648
649     if(Choice == 11) {
650
651         //Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
652
653         //Asking user to enter the assembly ID
654         System.out.println("Enter the Assembly ID");
655         int assembly_id= Integer.parseInt(br.readLine());
656
657         final String query11 = "SELECT process.process_id,process.department_number FROM process, department,job "
658                         + "WHERE job.assembly_id = " + assembly_id + " and department.department_number = process.department_number "
659                         + "and job.process_id = process.process_id ORDER BY job.commence_date";
660
661
662         //since it is a result set, printing all the values
663         ResultSet rs = statement.executeQuery(query11);
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900

```

6.11. Screenshots showing the testing of query 11:

The screenshot shows the Eclipse IDE interface with two tabs open: `individual_project.java` and `sample.java`. The `individual_project.java` tab contains the following Java code:

```
647
648     }
649
650     if(Choice == 11) {
651
652         //Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
653
654         //Asking user to enter the assembly ID
655         System.out.println("Enter the Assembly ID");
656         int assembly_id= Integer.parseInt(br.readLine());
657
658         final String query11 = "SELECT process.process_id,process.department_number FROM process, department,job "
659         + "WHERE job.assembly_id = '" + assembly_id + "' and department.department_number = process.department_number "
660         + "and job.process_id = process.process_id ORDER BY job.commence_date";
661
662
663         //since it is a result set, printing all the values
664         ResultSet rs = statement.executeQuery(query11);
665
666         System.out.println("Process ID | Department No");
667
668         while (rs.next()) {
669             System.out.println(String.format("%s | %s",
670                         rs.getInt(1),
671                         rs.getInt(2)));
672
673     }
674
675 }
```

The `Console` tab shows the execution output:

```
Markers Properties Servers Data Source Explorer Snippets Console
individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:12:51 PM)
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
11
Enter the Assembly ID
1000
Process ID | Department No
3007 | 2000
3001 | 2000
3004 | 2000
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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
```

File Search Project Run Window Help

individual_project.java sample.java

```

647
648    }
649
650    if(Choice == 11) {
651
652        //Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
653
654        //Asking user to enter the assembly ID
655        System.out.println("Enter the Assembly ID");
656        int assembly_id= Integer.parseInt(br.readLine());
657
658        final String query11 = "SELECT process.process_id,process.department_number FROM process, department,job "
659                    + "WHERE job.assembly_id = " + assembly_id + " and department.department_number = process.department_number "
660                    + "and job.process_id = process.process_id ORDER BY job.commence_date";
661
662
663        //since it is a result set, printing all the values
664        ResultSet rs = statement.executeQuery(query11);
665
666        System.out.println("Process ID | Department No");
667
668        while (rs.next()) {
669            System.out.println(String.format("%s      | %s",
670                                rs.getInt(1),
671                                rs.getInt(2)));
672
673    }
674
675    Markers Properties Servers Data Source Explorer Snippets Console

```

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:12:51 PM)

11. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

11
Enter the Assembly ID
1002
Process ID | Department No
3002 | 2000
3006 | 2002
3003 | 2002

=====
=====
=====You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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

Project Run Window Help

individual_project.java sample.java

```

647
648    }
649
650    if(Choice == 11) {
651
652        //Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
653
654        //Asking user to enter the assembly ID
655        System.out.println("Enter the Assembly ID");
656        int assembly_id= Integer.parseInt(br.readLine());
657
658        final String query11 = "SELECT process.process_id,process.department_number FROM process, department,job "
659        + "WHERE job.assembly_id = " + assembly_id + " and department.department_number = process.department_number "
660        + "and job.process_id = process.process_id ORDER BY job.commence_date";
661
662
663        //since it is a result set, printing all the values
664        ResultSet rs = statement.executeQuery(query11);
665
666        System.out.println("Process ID | Department No");
667
668        while (rs.next()) {
669            System.out.println(String.format("%s | %s",
670                rs.getInt(1),
671                rs.getInt(2)));
672
673    }
674
675    }
676
677    else
678    {
679        System.out.println("Process ID | Department No");
680
681        while (rs.next()) {
682            System.out.println(String.format("%s | %s",
683                rs.getInt(1),
684                rs.getInt(2)));
685
686        }
687    }
688
689    rs.close();
690    statement.close();
691    connection.close();
692
693    System.out.println("Process completed successfully");
694
695}
696
697}

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:12:51 PM)

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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

11
Enter the Assembly ID
1003
Process ID | Department No
3010 | 2003
3009 | 2003

=====You have the Following Options to Choose=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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

6.12. Screenshots showing the testing of query 12:

```
individual_project.java sample.java
734     + "job_process_id=process.process_id AND job.job_no=paint_job.paint_job_no AND department_number= '" + department_no + "'";
735     ;
736
737     ResultSet rs2 = statement.executeQuery(query12_3);
738     //since it is a result set, printing all the values
739     while (rs2.next()) {
740         System.out.println(String.format(" %s | %s ", 
741             rs2.getInt(1),
742             rs2.getString(2),
743             rs2.getString(3),
744             rs2.getString(4),
745             rs2.getFloat(5),
746             rs2.getFloat(6),
747             rs2.getString(7)));
748     }
749
750     System.out.println("=====");

Markers Properties Servers Data Source Explorer Snippets Console
individual.project[Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:45:25 PM)
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
12
Enter the Department Number
2002
Enter the completion Date
2019-05-16
 5005 | 2019-05-14 | 2019-05-16 | 13 | 1002
 5006 | 2019-05-13 | 2019-05-16 | 12 | 1002
=====
=====You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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
```

Project Run Window Help

individual_project.java sample.java

```

676
677
678 }
679
680 if(choice == 12) {
681
682     //Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
683
684     //Asking user to enter the department Number
685     System.out.println("Enter the Department Number");
686     int department_no= Integer.parseInt(br.readLine());
687
688     //Asking user to enter the completion date
689     System.out.println("Enter the completion Date");
690     String date = br.readLine();
691
692     final String query12_1 = "SELECT job.job_no, job.commence_date, completion_date, cut_job.type_of_machine, cut_job.time_used, cut_job.material_used, labor_time, assembly_id\r\n" +
693         " FROM job, process, cut job \r\n" +

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:45:25 PM)

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 datecommenced 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: enter new customers from a data file until the file is empty
 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 18. QUIT

12

Enter the Department Number

2003

Enter the completion Date

2019-05-11

5008		2019-05-03		2019-05-11		Sawing		12.0		Steel		10		1003
5007		2019-05-09		2019-05-11		15								1003

You have the Following Options to Choose:

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered

A. Enter a new process-id and its department, transection with its type and information relevant to the type

6.13. Screenshots showing the testing of query 13:

The screenshot shows an IDE interface with a code editor and a console window.

Code Editor (individual_project.java):

```
rs.getString(8));
        System.out.println("=====");
    }
}

if(Choice == 13) {
    // Retrieve the customers (in name order) whose category is in a given range
    //asking user to enter the range
    System.out.println("Enter the category range");
    //asking user for the minimum value in the range
    System.out.println("Minimum category value(category has values from 1 to 10)");
    int min_cat= Integer.parseInt(br.readLine());

    //asking user to enter the maximum value in the range
    System.out.println("Maximum category value(category has values from 1 to 10)");
    int max_cat = Integer.parseInt(br.readLine());
}

//Executing the query 13

final String query13 = "SELECT * FROM customer WHERE category BETWEEN '" + min_cat + "' and '" + max_cat + "'";
ResultSet rs1 = statement.executeQuery(query13);
System.out.println("Name | Address | Category");
//since it is a result set, printing all the values

while (rs1.next()) {
    System.out.println(String.format("%s | %s | %s",
        rs1.getString(1),
        rs1.getString(2),
        rs1.getInt(3)));
}
}

Markers Properties Servers Data Source Explorer Snippets Console
```

Console Output:

```
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
13
Enter the category range
Minimum category value(category has values from 1 to 10)
1
Maximum category value(category has values from 1 to 10)
3
Name | Address | Category
Ghani | C106,Crimson Park,Norman | 3
Subbu | 1003D East Brooks Street Norman | 1
Retrieved the customers (in name order) whose category is in a given range
=====
```

Search Project Run Window Help

individual_project.java sample.java

```

747     rs.getString(8));
748     System.out.println("*****");
749   }
750 }
751
752 if(Choice == 13) {
753
754   // Retrieve the customers (in name order) whose category is in a given range
755
756   //asking user to enter the range
757   System.out.println("Enter the category range");
758   //asking user for the minimum value in the range
759   System.out.println("Minimum category value(category has values from 1 to 10)");
760   int min_cat= Integer.parseInt(br.readLine());
761
762   //asking user to enter the maximum value in the range
763   System.out.println("Maximum category value(category has values from 1 to 10)");
764   int max_cat = Integer.parseInt(br.readLine());
765
766   //Executing the query 13
767
768
769   final String query13 = "SELECT * FROM customer WHERE category BETWEEN '" + min_cat + "' and '" + max_cat + "'";
770
771   ResultSet rs1 = statement.executeQuery(query13);
772   System.out.println("Name | Address | Category");
773   //since it is a result set, printing all the values
774
775   while (rs1.next()) {
776     System.out.println(String.format("%s | %s | %s",
777       rs1.getString(1),
778       rs1.getString(2),
779       rs1.getInt(3)));
780   }
781 }
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:21:52 PM)

17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

13
Enter the category range
Minimum category value(category has values from 1 to 10)
1
Maximum category value(category has values from 1 to 10)
5
Name | Address | Category
Ghani | C106,Crimson Park,Norman | 3
Subbu | 1089 East Brooks Street,Norman | 1
Retrieved the customers (in name order) whose category is in a given range
=====
=====You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
4. Enter a new process-id with 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 ...

Search Project Run Window Help

individual_project.java sample.java

```

756   //asking user to enter the range
757   System.out.println("Enter the category range");
758   //asking user for the minimum value in the range
759   System.out.println("Minimum category value(category has values from 1 to 10)");
760   int min_cat= Integer.parseInt(br.readLine());
761
762   //asking user to enter the maximum value in the range
763   System.out.println("Maximum category value(category has values from 1 to 10)");
764   int max_cat = Integer.parseInt(br.readLine());
765
766   //Executing the query 13
767
768
769   final String query13 = "SELECT * FROM customer WHERE category BETWEEN '" + min_cat + "' and '" + max_cat + "'";
770
771   ResultSet rs1 = statement.executeQuery(query13);
772   System.out.println("Name | Address | Category");
773   //since it is a result set, printing all the values
774
775   while (rs1.next()) {
776     System.out.println(String.format("%s | %s | %s",
777       rs1.getString(1),
778       rs1.getString(2),
779       rs1.getInt(3)));
780   }
781
782   System.out.println("Retrieved the customers (in name order) whose category is in a given range");
783   System.out.println("=====");
784
785
786 }
787
788 if(Choice == 14) {
789   // Delete all cut-jobs whose job-no is in a given range
790 }
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 8:21:52 PM)

13
Enter the category range
Minimum category value(category has values from 1 to 10)
6
Maximum category value(category has values from 1 to 10)
10
Name | Address | Category
Abhinash | 250 spring fields, Norman | 10
Goutham | brooks st oklahoma | 9
Maharshi | 227 clisen, Norman | 6
Retrieved the customers (in name order) whose category is in a given range
=====
=====You have the Following Options to Choose:=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered

6.14. Screenshots showing the testing of query 14:

Cut Job table Before Deletion:

The screenshot shows the SSMS interface with the following details:

- File Explorer:** Shows several open files including `DROP_TABLES.sql`, `Procedures.sql`, `Insertions_IP.sql`, `Queries_IP.sql`, and `queries.sql`.
- Toolbar:** Includes buttons for Run, Cancel, Disconnect, Change Connection, Explain, and Enable SQLCMD.
- Text Editor:** Displays the following SQL code:


```

18
19
20  select * from cut_job;
21
22  /* SQL query for option 15 */
23  --Change the color of a given paint job
24
25
26
27
28
29
      
```
- Results Grid:** Shows the data from the `cut_job` table:

	cut_job_no	type_of_machine	time_used	material_used	labor_time
1	5008	Sawing	12	Steel	10
2	5009	Broaching	12	cast iron	12
3	5010	Drilling	12	metal	12

Deleting the Cut jobs by running the Query 14:

The screenshot shows a Java application window with the following details:

- Code View:** Displays the Java code for executing a delete query on the `cut_job` table based on a user-specified range of job numbers.
- Console Output:** Shows the execution of the program and the resulting output:


```

3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT
=====
14
Enter the Job Number range
Enter the First job number in the range
5005
Enter the Last job number in the range
5008
All the cut_jobs in the given range are Deleted
=====
=====You have the Following Options to Choose=====
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
      
```

SQLProject/src/individual_project.java - Eclipse IDE

File Edit View Insert Run Project Help

individual_projectjava sample.java

```
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
```

After Deletions:

The screenshot shows the SSMS interface with the following details:

- File Explorer:** Shows multiple open files including `DROP_TABLES.sql`, `Procedures.sql`, `Insertions_IP.sql`, `Queries_IP.sql`, and `SQLQuery`. The `Queries_IP.sql` file is currently selected.
- Status Bar:** Displays "5 UNSAVED" changes.
- Toolbar:** Includes buttons for Run, Cancel, Disconnect, Change Connection (set to `cs-dsa-4513-sql-db`), Explain, and Enable SQLCMD.
- Text Editor:** Contains the following SQL code:

```
218
219     select * from cut_job;
220
221 /* SQL query for option 15 */
222 --Change the color of a given paint job
223
224
225
226
227
228
229
```
- Results Tab:** Selected tab, showing a results grid with the following columns:

cut_job_no	type_of_machine	time_used	material_used	labor_time
------------	-----------------	-----------	---------------	------------

Testing Query 15:

Table Before changing the colors:

```

221  /* SQL query for option 15 */
222  --Change the color of a given paint job
223
224
225
226  select * from Paint_job;
227
228
229

```

	paint_job_no	color	volume	labor_time
1	5001	blue	3	12.5
2	5002	green	3	12
3	5003	red	3	12

Executing the Query 15

```

8099 statement.executeUpdate(query14);
810
811     System.out.println("All the cut_jobs in the given range are Deleted");
812     System.out.println("=====");
813 }
814
815 if(Choice == 15) {
816
817     // Change the color of a given paint job
818
819     //asking user to enter the Job Number
820     System.out.println("Enter the Job Number of the Paint Job you want to change the color for");
821     int first_job= Integer.parseInt(br.readLine());
822
823
824     //reading the color that user want to change the existing cut color to
825     System.out.println("which color do you want to change it to?");
826     String color = br.readLine();
827
828
829     //Executing the query 15
830
831     final String query15 = "UPDATE paint_job SET color = '" + color + "' WHERE paint_job_no = '" + first_job + "'";
832
833     statement.executeUpdate(query15);
834
835     System.out.println("Color of a given paint job is changed");
836     System.out.println("=====");
837
838 }

```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 9:41:30 PM)

3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 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 datecommenced 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: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

15
Enter the Job Number of the Paint Job you want to change the color for
5001
which color do you want to change it to?
violet
Color of a given paint job is changed
=====

You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered

```
809     statement.executeUpdate(query14);
810
811     System.out.println("All the cut_jobs in the given range are Deleted");
812     System.out.println("=====");
813
814 }
815
816 if(Choice == 15) {
817
818     // Change the color of a given paint job
819
820     //asking user to enter the Job Number
821     System.out.println("Enter the Job Number of the Paint Job you want to change the color for");
822     int first_job= Integer.parseInt(br.readLine());
823
824     //reading the color that user want to change the existing cut color to
825     System.out.println("which color do you want to change it to?");
826     String color = br.readLine();
827
828     //Executing the query 15
829
830
831     final String query15 = "UPDATE paint_job SET color = '" + color + "' WHERE paint_job_no = '" + first_job + "'";
832
833     statement.executeUpdate(query15);
834
835     System.out.println("Color of a given paint job is changed");
836     System.out.println("=====");
837
838 }
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 9:41:30 PM)

1. Retrieve the cost incurred on an assembly-id
2. Retrieve the total labor time within a department for jobs completed in the department during a given date
3. Retrieve the processes through which a given assembly-id has passed so far (in datecommenced order) and the department responsible for each process
4. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department
5. Retrieve the customers (in name order) whose category is in a given range
6. Enter all cut-jobs whose job-no is in a given range
7. Change the color of a given paint job
8. Import: enter new customers from a data file until the file is empty
9. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
10. QUIT

=====

15
Enter the Job Number of the Paint Job you want to change the color for
5002
which color do you want to change it to?
Brown
Color of a given paint job is changed
=====You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

ect
809 statement.executeUpdate(query14);
810
811 System.out.println("All the cut_jobs in the given range are Deleted");
812 System.out.println("=====");
813
814 }
815
816 if(Choice == 15) {
817
818 // Change the color of a given paint job
819
820 //asking user to enter the Job Number
821 System.out.println("Enter the Job Number of the Paint Job you want to change the color for");
822 int first_job= Integer.parseInt(br.readLine());
823
824 //reading the color that user want to change the existing cut color to
825 System.out.println("which color do you want to change it to?");
826 String color = br.readLine();
827
828 //Executing the query 15
829
830
831 final String query15 = "UPDATE paint_job SET color = '" + color + "' WHERE paint_job_no = '" + first_job + "'";
832
833 statement.executeUpdate(query15);
834
835 System.out.println("Color of a given paint job is changed");
836 System.out.println("=====");
837
838 }

Markers Properties Servers Data Source Explorer Snippets Console

individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 9:41:30 PM)

1. Change the color of a given paint job
2. Import: enter new customers from a data file until the file is empty
3. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
4. QUIT

=====

15
Enter the Job Number of the Paint Job you want to change the color for
5003
which color do you want to change it to?
Yellow
Color of a given paint job is changed
=====You have the Following Options to Choose:
1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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

Paint_job Table after the colors are changed.

The screenshot shows the Azure Data Studio interface with the following details:

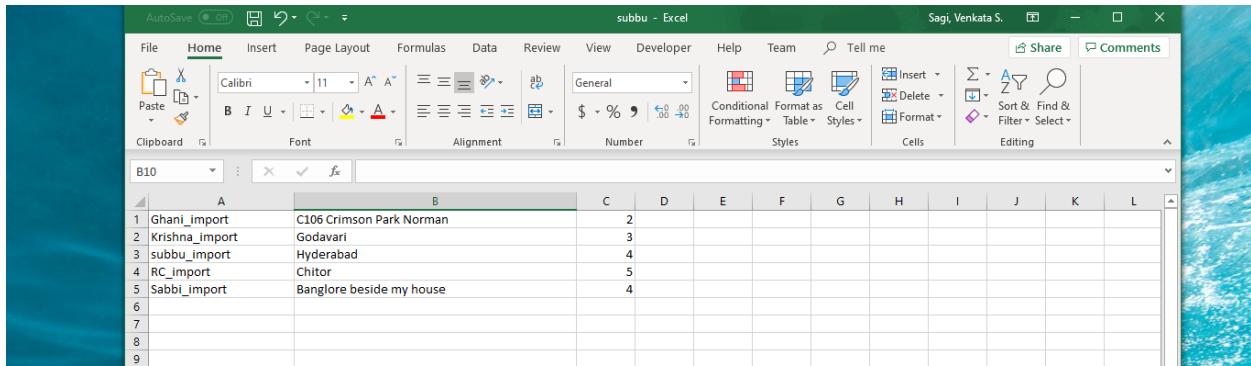
- Top Bar:** Help, Queries_IP.sql - sagi0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (sagi0000) - Azure Data Studio
- Toolbar:** DROP_TABLES.sql, Procedures.sql, Insertions_IP.sql, Queries_IP.sql
- File Explorer:** C: > Users > dvsnv > OneDrive > Documents > Queries_IP.sql
- Toolbar Buttons:** Run, Cancel, Disconnect, Change Connection (set to cs-dsa-4513-sql-db), Explain, Enable SQLCMD
- Text Editor:** SQL code for changing paint job colors, starting with line 221.
- Results Tab:** Shows the results of the query "select * from Paint_job;" with three rows of data.

	paint_job_no	color	volume	labor_time
1	5001	violet	3	12.5
2	5002	Brown	3	12
3	5003	Yellow	3	12

6.16 Screenshots showing the testing of the import and export :

Import:

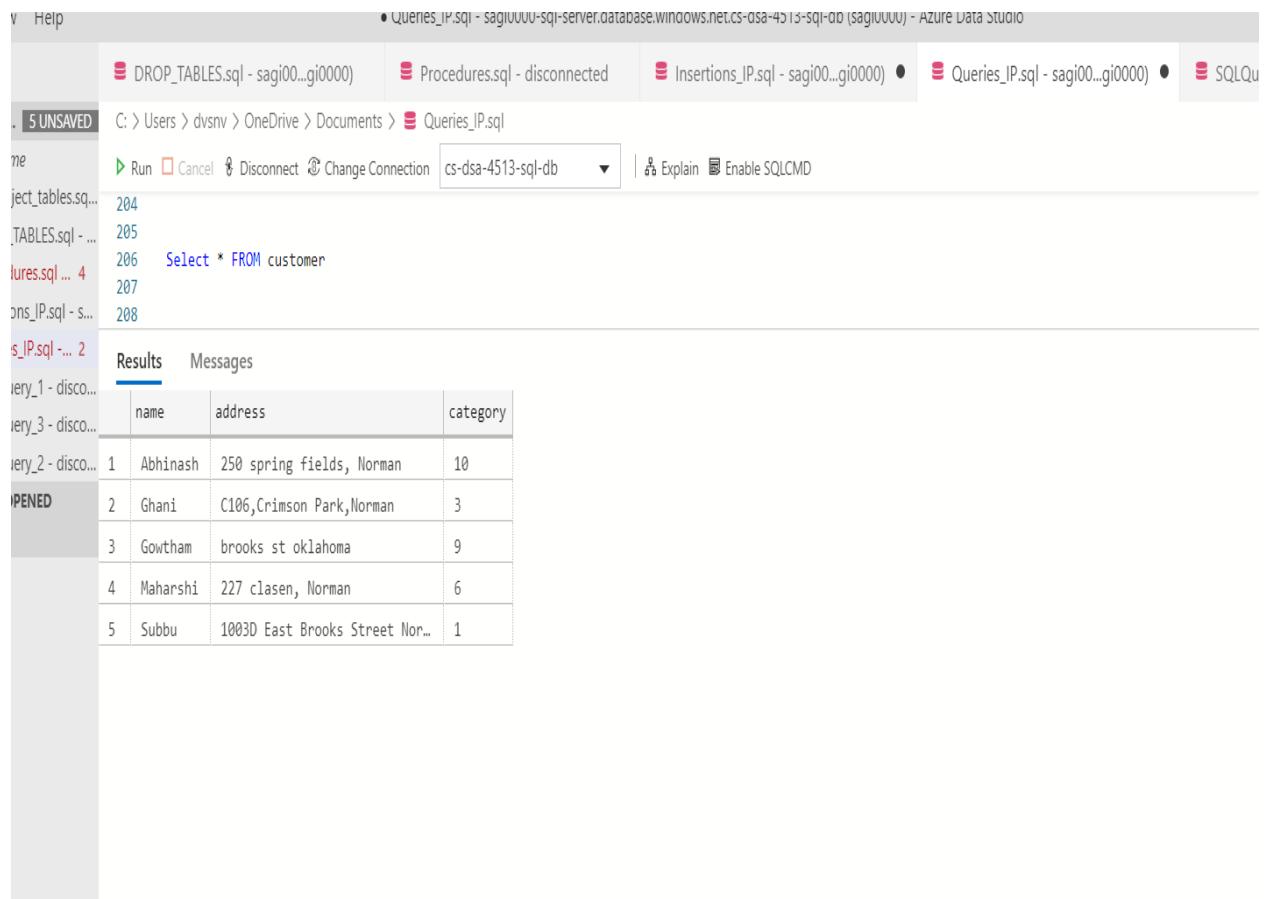
Input file I have with the data:



A screenshot of Microsoft Excel showing a table with data to be imported. The table has columns A and B. Column A contains names and column B contains addresses. The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Ghani_import	C106 Crimson Park Norman	2									
2	Krishna_import	Godavari	3									
3	subbu_import	Hyderabad	4									
4	RC_import	Chitor	5									
5	Sabbi_import	Banglore beside my house	4									
6												
7												
8												
9												

Contents of the table Before Importing.



A screenshot of Azure Data Studio showing the results of a SELECT query before importing. The results are displayed in a table with columns name, address, and category. The data is as follows:

	name	address	category
1	Abhinash	250 spring fields, Norman	10
2	Ghani	C106,Crimson Park,Norman	3
3	Gowtham	brooks st oklahoma	9
4	Maharshi	227 clasen, Norman	6
5	Subbu	1003D East Brooks Street Nor...	1

Importing using Eclipse:

```

    839
    840
    841     if(Choice == 16) {
    842
    843         String file_name;
    844         //Import: Enter new customers from a data file until the file is empty
    845         //asking user to enter the name of the file that is to be imported
    846
    847         System.out.println("Enter the file name: ");
    848
    849         file_name = myScan.nextLine();
    850
    851         //path where the filename is searched
    852
    853         FileInputStream fstream = new FileInputStream("/Users/dvsnv/Desktop/" + file_name);
    854
    855         DataInputStream i = new DataInputStream(fstream);
    856         BufferedReader br = new BufferedReader(new InputStreamReader(i));
    857
    858         String Line;
    859         //Importing each row until the file is empty
    
```

Markers Properties Servers Data Source Explorer Snippets Console

individual_project.java [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 10:31:19 PM)

1. Create a new assembly
2. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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 datecommenced 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. Select all customers in a given range
15. Change the color of a given paint job
16. Import: enter new customers from a data file until the file is empty
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
18. QUIT

=====

16
Enter the file name:
subbu.csv
row inserted successfully
File Imported successfully!!!
===== You have the Following Options to Choose:=====

1. Enter a new customer
2. Enter a new department
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
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

Contents of the Table After Importing:

File Help • Queries_IP.sql - sagi0000-sql-server.database.windows.net.cs-dsa-4513-sql-db (sagi0000) - Azure Data Studio

C:\> Users > dvsnv > OneDrive > Documents > Queries_IP.sql

Run Cancel Disconnect Change Connection cs-dsa-4513-sql-db Explain Enable SQLCMD

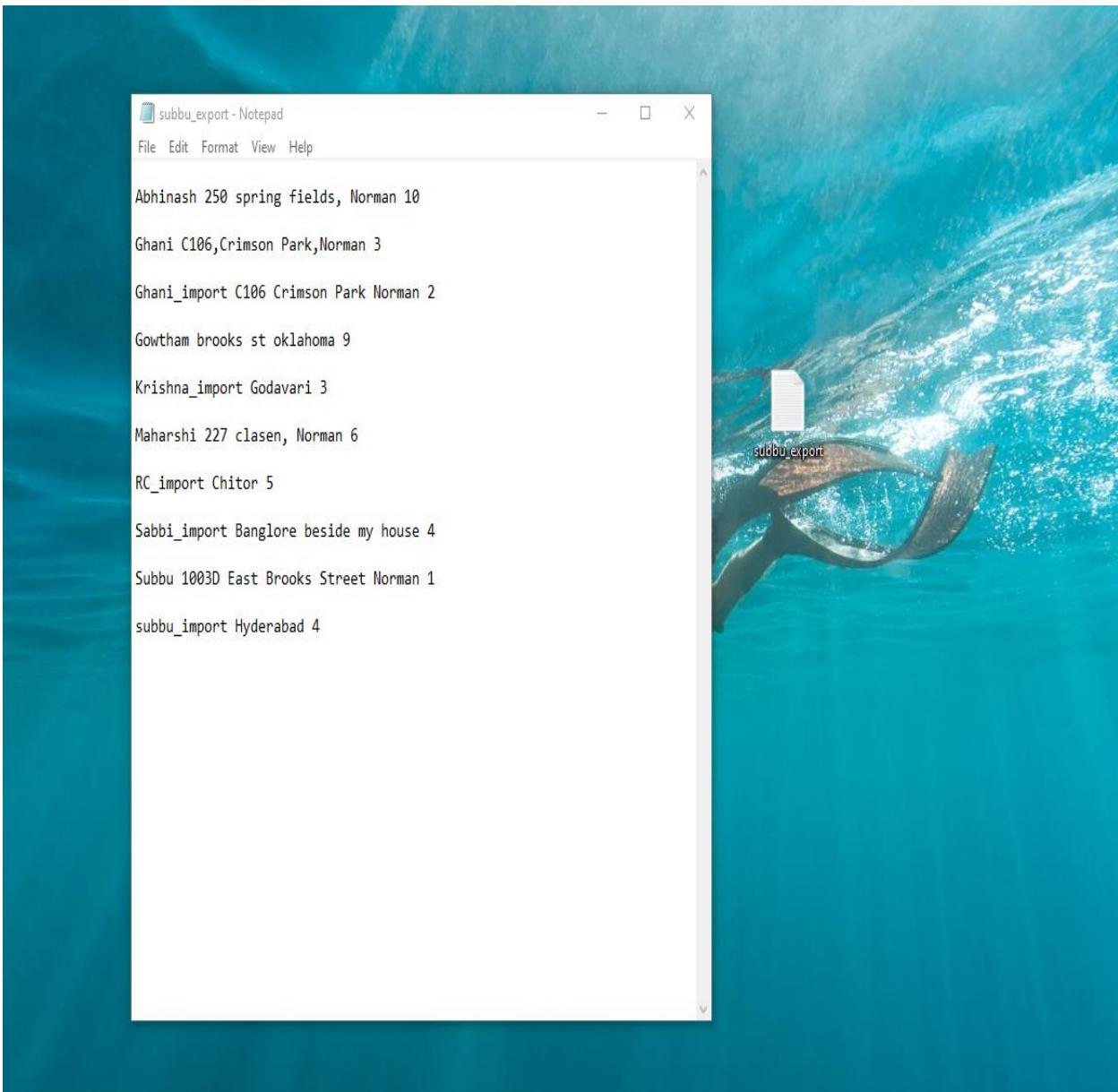
	name	address	category
1	Abhinash	250 spring fields, Norman	10
2	Ghani	C106,Crimson Park, Norman	3
3	Ghani_import	C106 Crimson Park Norman	2
4	Gowtham	brooks st oklahoma	9
5	Krishna_import	Godavari	3
6	Maharshi	227 clasen, Norman	6
7	RC_import	Chitor	5
8	Sabbi_import	Banglore beside my house	4
9	Subbu	1003D East Brooks Street Nor...	1
10	subbu_import	Hyderabad	4

Export Option:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows two files: `individual_project.java` and `sample.java`.
- Code Editor:** Displays the `individual_project.java` file containing Java code for exporting customer data based on category range.
- Console Tab:** Shows the output of running the code. It includes:
 - Project information: `individual project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 10:31:19 PM)`
 - Options listed:
 - 12. Change the color of a given point you
 - 16. Import: enter new customers from a data file until the file is empty
 - 17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
 - 18. QUIT
 - Execution steps:
 - 17
 - Enter the category range
 - Minimum category value(category has values from 1 to 10)
 - 1
 - Maximum category value(category has values from 1 to 10)
 - 10
 - Enter output file name:
 - `subbu_export.txt`
 - Row Exported successfully!
 - File exported successfully!!!
 - A message at the bottom: `=====You have the Following Options to Choose=====`

Exported File:



6.17. Screenshots showing the testing of Errors:

1) Constraint Error as the value of Category is not between 1-10

2) Primary Key Duplication Error as Primary Key name entered already exists in the database

3) Foreign Key Referencing Error as the value given for the name is not present in the referenced table

```

1. Enter the Unique Assembly ID
2. Enter the ordered date - in the form of YYYY-MM-DD:
2019-09-09
3. Enter the assembly details
4. Enter the Customer's name who ordered the assembly
xyz
Exception in thread "main" com.microsoft.sqlserver.idbc.SQLServerException: The INSERT statement conflicted with the FOREIGN KEY constraint "FK_assembly_c_name_1708600F". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.assembly", column 'c_name'. The statement failed.
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:262)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:283)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:129)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:1620)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:868)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:268)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:223)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:248)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:2979)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:248)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:223)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:711)
        at individual_project.main(individual_project.java:149)

```

4) Conversion failed error, as given date cannot exist.

```

1. Enter the Unique Assembly ID
2. Enter the ordered date - in the form of YYYY-MM-DD:
2019-15-09
3. Enter the assembly details
4. Enter the Customer's name who ordered the assembly
xyz
Exception in thread "main" com.microsoft.sqlserver.idbc.SQLServerException: Conversion failed when converting date and/or time from character string.
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:262)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:1624)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:868)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:268)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:223)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:248)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:2979)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:248)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:223)
        at com.microsoft.sqlserver.jdbc.IDBCStatement.executeUpdate(IDBCStatement.java:711)
        at individual_project.main(individual_project.java:149)

```

6.18. Screenshots showing the testing of the quit option

Project/src/individual_project/java - Eclipse IDE

Search Project Run Window Help

individual_project.java sample.java Quick Access

```
final String query13 = "SELECT * FROM customer WHERE category BETWEEN '" + min_cat + "' and '" + max_cat + "'";  
ResultSet rs = statement.executeQuery(query13);  
String string = null;  
  
System.out.println("Enter output file name: ");  
out_file = br.readLine();  
  
BufferedWriter export = new BufferedWriter(new FileWriter("/Users/dvsnv/Desktop/" + out_file));  
  
//reading each row and exporting it to the file  
while (rs.next()) {  
    string = rs.getString("name") + " " + rs.getString("address") + " " + rs.getString("category");  
    export.write("\n");  
    export.write(string);  
    export.write("\n");  
  
    System.out.println(" Row Exported successfully!");  
}  
  
System.out.println(" File exported successfully!!!");  
  
export.close();  
}  
  
if (Choice == 18)  
    //Choice 18 will exit the program  
    System.out.println("Thanks... Bye...");  
}  
  
}  
}  
}  
}  
  
=====  
Markers Properties Servers Data Source Explorer Snippets Console  
<terminated> individual_project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 14, 2019, 10:31:19 PM)  
File exported sucessfully!!!  
===== You have the Following Options to Choose: ======  
1. Enter a new customer  
2. Enter a new department  
3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered  
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 datecommenced 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: enter new customers from a data file until the file is empty  
17. Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen  
18. QUIT  
=====  
18  
Thanks... Bye...
```

TASK 7

Web Database Application and its execution

7.1 web database application source program and Screenshots showing its successful compilation.

DataHandler1.java

Below java file contains code used to connect to the Azure SQL Database and execute the both the queries 1 and 13.

```
package jsp_azure_test;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DataHandler1 {
    private Connection conn;
    private String server = "sagi0000-sql-server.database.windows.net";
    private String database = "cs-dsa-4513-sql-db";
    private String username = "sagi0000";
    private String password = "Avinash8&65";
    // Resulting connection string
    final private String url =
        String.format("jdbc:sqlserver://%" + host + ";database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;loginTimeout=30",
                      server, database, username, password);
    // Initialize and save the database connection
    private void getDBConnection() throws SQLException {
        if (conn != null) {
            return;
        }
        this.conn = DriverManager.getConnection(url);
    }

    // Return the result of selecting everything from the customer table
    public ResultSet getAllCustomers() throws SQLException {
        getDBConnection();
        final String sqlQuery = "SELECT * FROM customer;";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }
}
```

```

// Inserts a record into the customer table with the given attribute values
public boolean addCustomer(
                           String name, String address, int category) throws
SQLException {
    getDBConnection(); // Prepare the database connection
    // Prepare the SQL statement
    final String sqlQuery =
        "INSERT INTO customer " +
        "(name, address, category) " +
        "VALUES " +
        "(?, ?, ?)";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
    // Replace the '?' in the above statement with the given attribute values
    stmt.setString(1, name);
    stmt.setString(2, address);
    stmt.setInt(3, category);
    // Execute the query, if only one record is updated, then we indicate success
    by returning true
    return stmt.executeUpdate() == 1;
}

// Return the result of selecting everything from the customer table in the given range
public ResultSet retrieveCustomers(int min_cat, int max_cat) throws
SQLException {
    getDBConnection();
    final String sqlQuery = "SELECT * FROM customer WHERE
category BETWEEN '" + min_cat + "' and '" + max_cat + "'";
    final PreparedStatement stmt =
conn.prepareStatement(sqlQuery);
    return stmt.executeQuery();
}
}

```

get_all_customers.jsp

Executing below JSP file generates an HTML file with a table containing all the records from the **customer** SQL database table.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Customers</title>
</head>
<body>
<%@page import="jsp_azure_test.DataHandler1"%>
<%@page import="java.sql.ResultSet"%>
<%
// We instantiate the data handler1 here, and get all the customers from the database
final DataHandler1 handler = new DataHandler1();
final ResultSet customers = handler.getAllCustomers();
%>
<!-- The table for displaying all the movie records -->
<table cellspacing="2" cellpadding="2" border="1">
<tr> <!-- The table headers row -->
<td align="center">
<h4>Customer Name</h4>
</td>
<td align="center">
<h4>Customer Address</h4>
</td>
<td align="center">
<h4>Category</h4>
</td>

</tr>
<%
while(customers.next()) { // For each customer record returned...
// Extract the attribute values for every row returned
final String name = customers.getString("name");
final String address = customers.getString("address");
final String category = customers.getString("category");

out.println("<tr>"); // Start printing out the new table row
out.println( // Print each attribute value
"<td align=\"center\">" + name +
"</td><td align=\"center\"> " + address +
"</td><td align=\"center\"> " + category +
"</td>");
out.println("</tr>");
}
%>
</table>
</body>
</html>
```

add_customer_form.jsp

Below JSP file (strictly speaking it's just a static HTML file) generates an HTML for collection of user input to insert a new record into a customer table. Upon form submission, add_customer.jsp file will be invoked to process the user input.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add New Customer</title>
</head>
<body>
<h2>Add New Customer</h2>
<!--
Form for collecting user input for the new customer record.
Upon form submission, add_customer.jsp file will be invoked.
-->
<form action="add_customer.jsp">
<!-- The form organized in an HTML table for better clarity. -->
<table border=1>
<tr>
<th colspan="2">Enter the Customer Data:</th>
</tr>
<tr>
<td>Customer Name:</td>
<td><div style="text-align: center;">
<input type=text name=name>
</div></td>
</tr>
<tr>
<td>Customer Address:</td>
<td><div style="text-align: center;">
<input type=text name=address>
</div></td>
</tr>
<tr>
<td>Category:</td>
<td><div style="text-align: center;">
<input type=text name=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>
```

add_customer.jsp

Below JSP file processes the user request to insert a new record into customer table initiated by the add_customer_form.jsp file and generates the HTML file response confirming the insertion or notifying of the problem.

```
<%@ 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_test.DataHandler1"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandler1 handler = new DataHandler1();
// Get the attribute values passed from the input form.
String name = request.getParameter("name");
String address = request.getParameter("address");
String categoryString = request.getParameter("category");
/*
* If the user hasn't filled out all the time, movie name and duration. This is very
simple
checking.
*/
if (name.equals("")) || address.equals("") || categoryString.equals("")) {
response.sendRedirect("add_customer_form.jsp");
} else {
int category = Integer.parseInt(categoryString);
// Now perform the query with the data from the form.
boolean success = handler.addCustomer(name, address, category);
if (!success) { // Something went wrong
%>
<h2>There was a problem inserting the course</h2>
<%
} else { // Confirm success to the user
%>
<h2>The New Customer:</h2>
<ul>
<li>Customer Name: <%=name%></li>
<li>Customer Address: <%=address%></li>
<li>Category: <%=categoryString%></li>
</ul>
<h2>Was successfully inserted.</h2>
<a href="get_all_customers.jsp">See all Customers.</a>
<%
}
}
%>
</body></html>
```

retrieve_customers_form.jsp

Below JSP file (strictly speaking it's just a static HTML file) generates an HTML for collection of user input to retrieve records in the range of category given from customer table. Upon form submission, retrieve_customers.jsp file will be invoked to process the user input.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Retrieve Customers</title>
</head>
<body>
<h2>Retrieve Customers in the Given Category Range</h2>
<!--
Form for collecting user input for the new movie_night record.
Upon form submission, add_customer.jsp file will be invoked.
-->
<form action="retrieve_customers.jsp">
<!-- The form organized in an HTML table for better clarity. -->
<table border=1>
<tr>
<th colspan="2">Enter the Category Range Values:</th>
</tr>
<tr>
<td>Ranges From:</td>
<td><div style="text-align: center;">
<input type=text name=min_cat>
</div></td>
</tr>
<tr>
<td>Ranges To:</td>
<td><div style="text-align: center;">
<input type=text name=max_cat>
</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=search>
</div></td>
</tr>
</table>
</form>
</body>
</html>
```

retrieve_customers.jsp

Below JSP file processes the user request to retrieve the records given the category range from customer table initiated by the retrieve_customers_form.jsp file and generates the HTML file response retrieving the information.

```
<%@ 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_test.DataHandler1"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>
    <%
    // The handler is the one in charge of establishing the connection.
    DataHandler1 handler = new DataHandler1();
    // Get the attribute values passed from the input form.
    String minCat = request.getParameter("min_cat");
    String maxCat = request.getParameter("max_cat");

    /*
     * If the user hasn't filled out all the details, minimum category range and maximum
     * category range. This is very simple
     * checking.
     */
    if (minCat.equals("") || maxCat.equals("")) {
      response.sendRedirect("retrieve_customers_form.jsp");
    }
    else
    {
      int duration = Integer.parseInt(minCat);
      int duration1 = Integer.parseInt(maxCat);
    }
  </body>
</html>
```

```

// Now perform the query with the data from the form.
final ResultSet customers = handler.retrieveCustomers(duration,duration1);
%>

<!-- The table for displaying all the movie records -->






```

Compilation Screen shots:

Running retrieve_customers_form.jsp

The screenshot shows the Eclipse IDE interface with the title bar "localhost:8080/jsp_azure_test/retrieve_customers_form.jsp - Eclipse IDE". The menu bar includes File, Edit, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The left sidebar shows a project structure with files like "java", "ler", "t.java", "ler1", "jsp", "form.jsp", and "sp". The main workspace displays a web browser window titled "Retrieve Customers" with the URL "http://localhost:8080/jsp_azure_test/retrieve_customers_form.jsp". The page content is "Retrieve Customers in the Given Category Range" with a form titled "Enter the Category Range Values:" containing fields for "Ranges From:" and "Ranges To:", and buttons for "Clear" and "search". Below the browser window, the Eclipse status bar shows "Markers Properties Servers Data Source Explorer Snippets Console" and "Tomcat v9.0 Server at localhost [Started, Synchronized]".

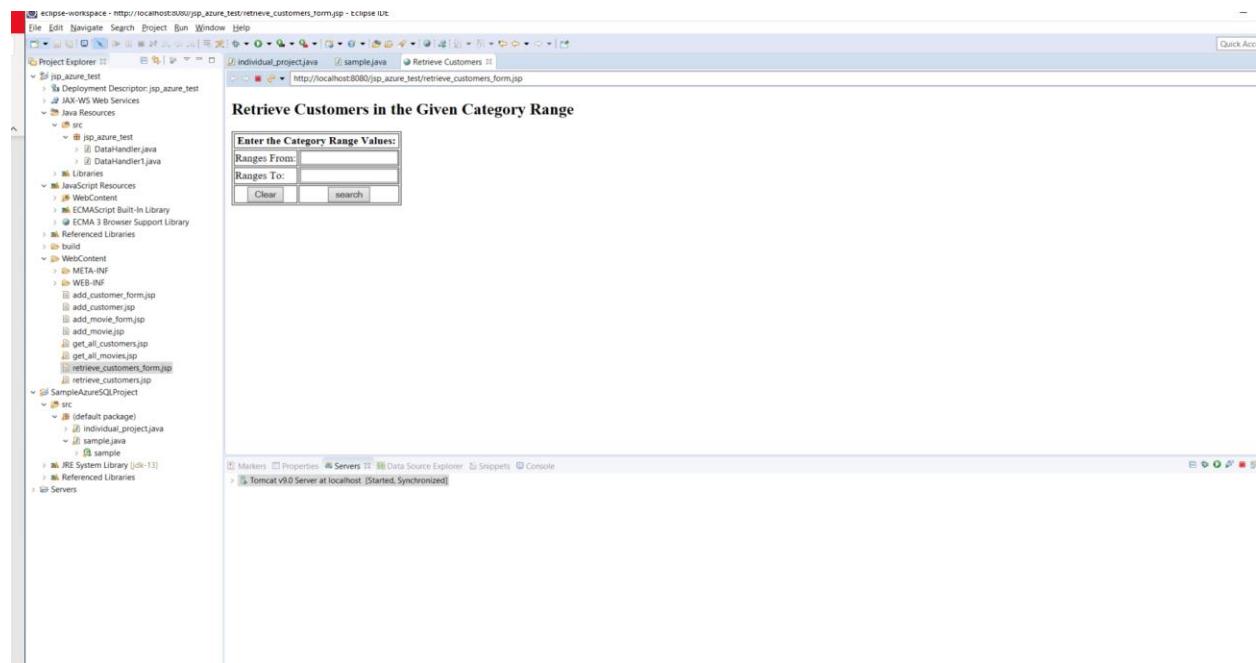
Running add_customers.jsp

The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - http://localhost:8080/jsp_azure_test/add_customer_form.jsp - Eclipse IDE". The menu bar includes File, Edit, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The left sidebar shows a project structure with files like "Deployment Descriptor", "jax-ws", "Java Resources", "src" (containing "jsp_azure_test" with "DataHandler.java" and "DataHandler1.java"), "JavaScript Resources", "WebContent" (with "ECMAScript Built-In Library" and "ECMA 3 Browser Support Library"), "Referenced Libraries", "build", "WebContent" (with "META-INF" and "WEB-INF" folders containing "add_customer_form.jsp", "add_customer.jsp", "add_movie_form.jsp", "add_movie.jsp", "get_all_customers.jsp", "get_all_movies.jsp", "retrieve_customers_form.jsp", "retrieve_customers.jsp"), "SampleAzureSQLProject" (with "src" folder containing "(default package)" with "individual_project.java" and "sample.java"), "IRE System Library [jdk-13]", "Referenced Libraries", and "Servers". The main workspace displays a web browser window titled "Add New Customer" with the URL "http://localhost:8080/jsp_azure_test/add_customer_form.jsp". The page content is "Add New Customer" with a form titled "Enter the Customer Data:" containing fields for "Customer Name:", "Customer Address:", and "Category:", and buttons for "Clear" and "Insert". Below the browser window, the Eclipse status bar shows "Markers Properties Servers Data Source Explorer Snippets Console" and "Tomcat v9.0 Server at localhost [Started, Synchronized]".

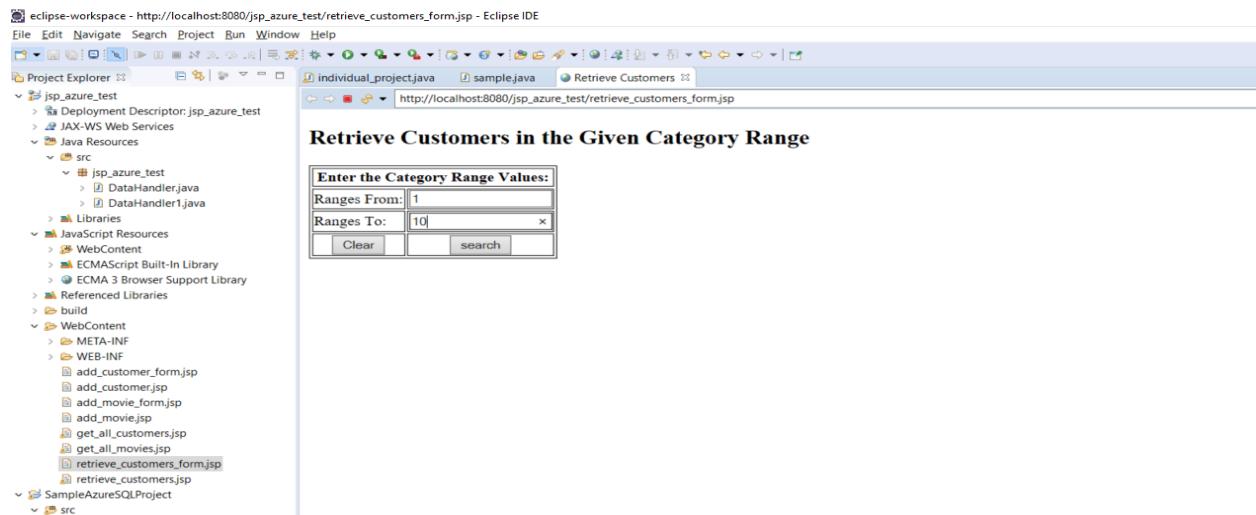
7.2 Screenshots showing the testing of Web DB Application

Running Query 13:

Web page that allow the user to enter the input data:



Given user Input and clicking on Insert



Web page that display the retrieved results

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "jsp.azure_test".
- Editor:** Displays a JSP page titled "retrieve_customers.jsp" with the following code:


```
<table border="1">
| Customer Name | Customer Address | Category |
| --- | --- | --- |
| Abhinash | 250 spring fields, Norman | 10 |
| Ghani | C106,Crimson Park,Norman | 3 |
| Ghani_import | C106 Crimson Park Norman | 2 |
| Gowtham | brooks st oklahoma | 9 |
| Krishna_import | Godavari | 3 |
| Maharshi | 227 clasen, Norman | 6 |
| RC_import | Chitor | 5 |
| Sabbi_import | Banglore beside my house | 4 |
| Subbu | 1003D East Brooks Street Norman | 1 |
| subbu_import | Hyderabad | 4 |

```
- Servers View:** Shows "Tomcat v9.0 Server at localhost [Started, Synchronized]".

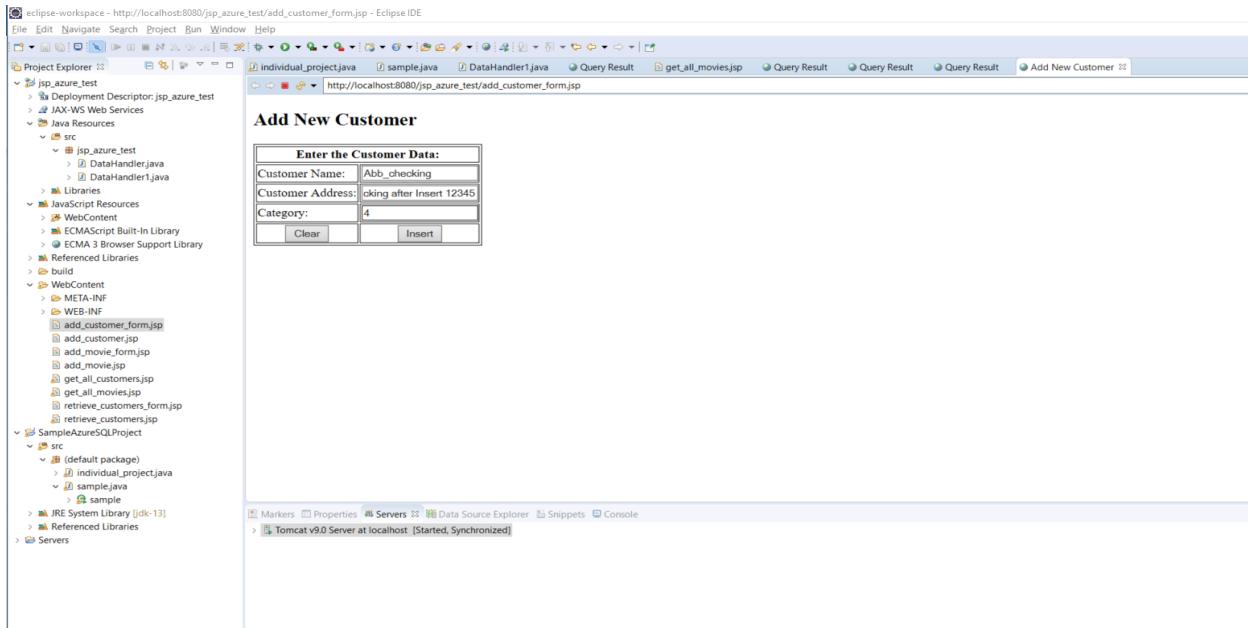
Running Query 1: Asking User for the User Information

The screenshot shows the Eclipse IDE interface with the following details:

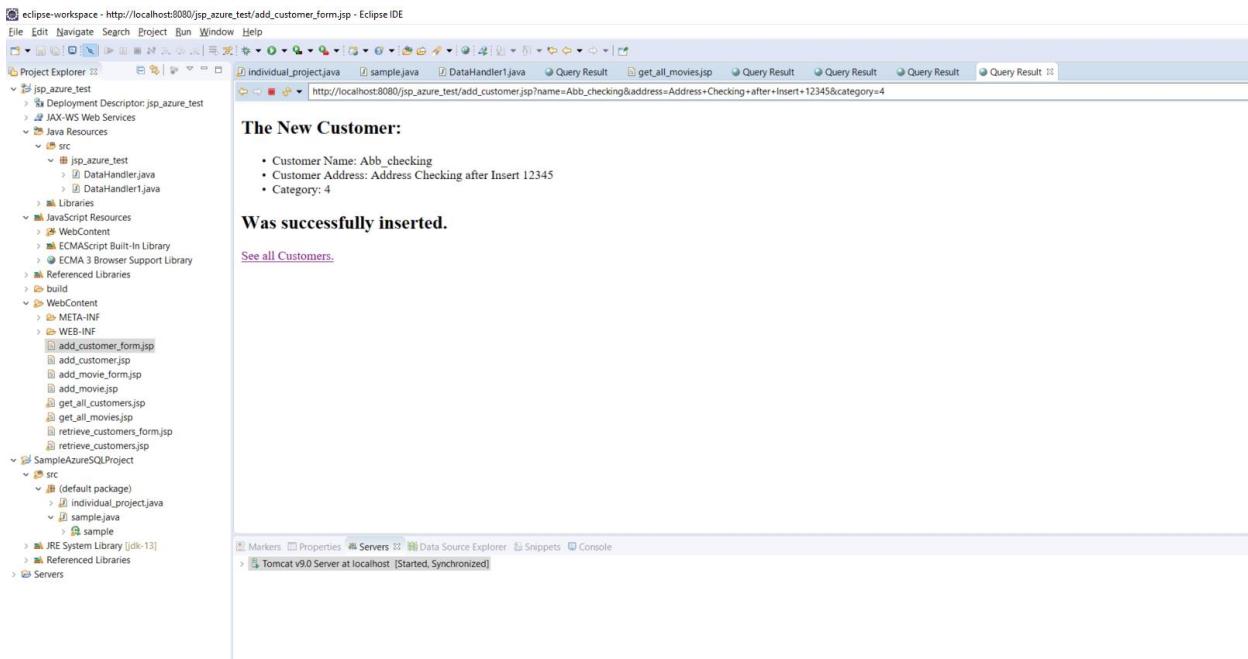
- Project Explorer:** Shows the project structure for "jsp.azure_test".
- Editor:** Displays a JSP page titled "add_customer_form.jsp" with the following code:


```
<form action="addCustomer">
    <table border="1">
        <tr>
            <td colspan="2">Enter the Customer Data:
        </tr>
        <tr>
            <td>Customer Name:
            <td>
        </tr>
        <tr>
            <td>Customer Address:
            <td>
        </tr>
        <tr>
            <td>Category:
            <td>
        </tr>
        <tr>
            <td>
            <td>
        </tr>
    </table>
</form>
```
- Servers View:** Shows "Tomcat v9.0 Server at localhost [Started, Synchronized]".

Web page that allow the user to enter the input data



Web page displaying the message confirming the successful execution of the insertion:



Running Query 13 again:

Changes are visible from the retrieved data table.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "jsp_azure_test". It includes Java Resources (src folder containing jsp_azure.java, DataHandler.java, DataHandler1.java, and Libraries), JavaScript Resources (WebContent folder containing META-INF, WEB-INF, add_customer_form.jsp, add_customer.jsp, add_movie.jsp, add_movie_form.jsp, get_all_customers.jsp, get_all_movies.jsp, retrieve_customers_form.jsp, and retrieve_customers.jsp), and Referenced Libraries (SampleAzureSQLProject).
- Browsers:** Displays the URL http://localhost:8080/jsp_azure_test/get_all_customers.jsp. The page shows a table with three columns: Customer Name, Customer Address, and Category. The data is as follows:

Customer Name	Customer Address	Category
Abb_checking	Address Checking after Insert 12345	4
Abhinash	250 spring fields, Norman	10
Ghani	C106,Crimson Park,Norman	3
Ghani_import	C106 Crimson Park Norman	2
Gowtham	brooks st oklahoma	9
Krishna_import	Godavari	3
Maharshi	227 clasen, Norman	6
RC_import	Chitor	5
Sabbi_import	Banglore beside my house	4
Subbu	1003D East Brooks Street Norman	1
subbu_import	Hyderabad	4