

# **A JOB\_ SHOP ACCOUNTING SYSTEM**

NAME- SUJATA SAHU

COURSE NAME-DSA4513-995

OU ID- 113592001

SEMESTER AND YEAR- 3rd and 2023

INSTRUCTOR NAME- Dr. Le Gruenwald

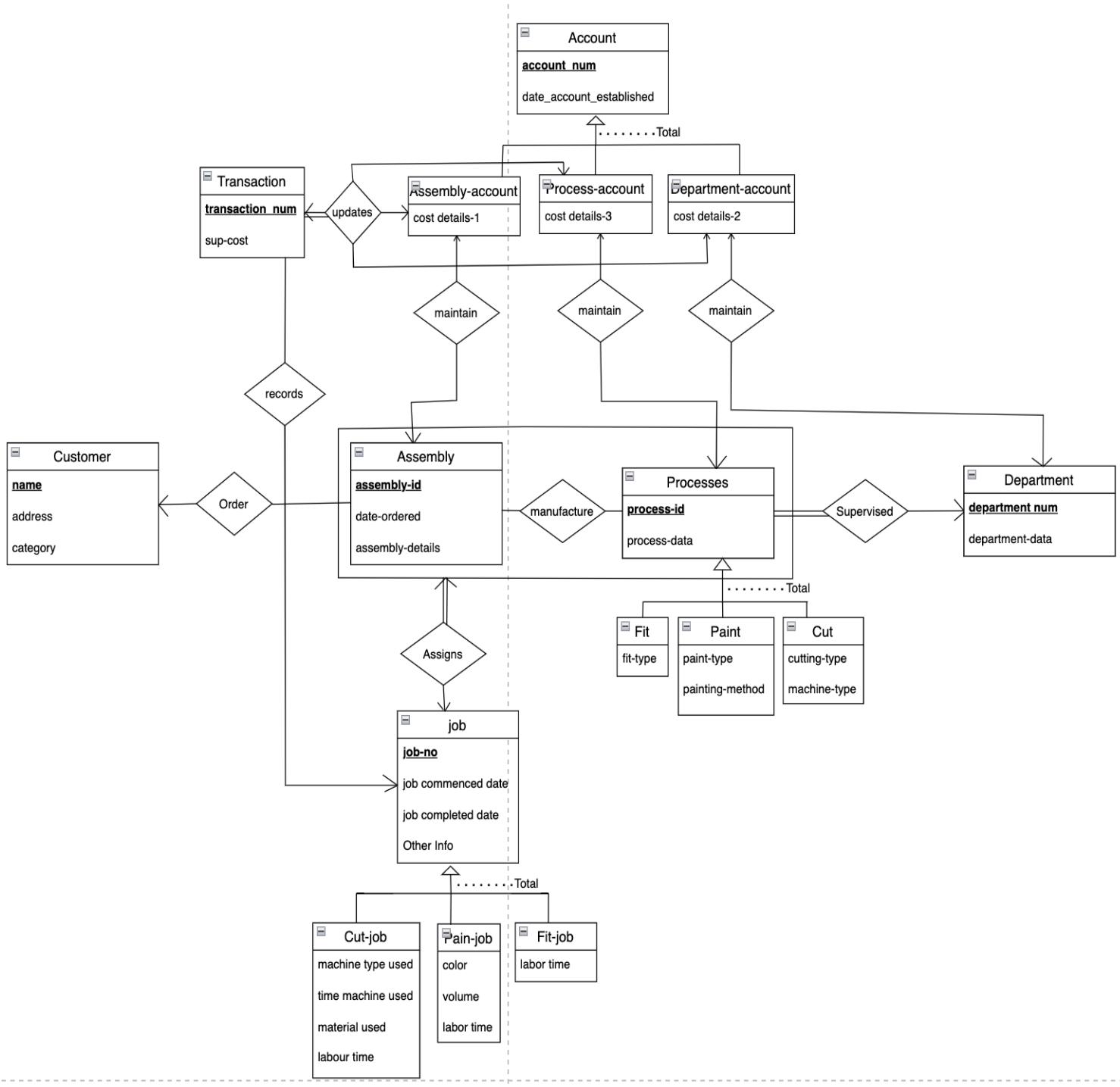
Email-id - [sujata.sahu-1@ou.edu](mailto:sujata.sahu-1@ou.edu)

## **TABLE OF CONTENTS**

<b>NO.</b>	<b>TASK</b>	<b>PAGE NO.</b>
1.	Design of Entity Relation Diagram(ERD)	3
2	Conversion of ERD to Relational Database-Relational Schema	4
3.1	Choice of Storage Structures for each relational table	5-7
3.2	Storage Structures for each relational table in Azure SQL Database	8
4	Creation of Tables in Database	9-13
5.1	SQL Statements and Associated stored procedures	14-25
5.2	Java Source Program and Associated Screenshots on successful completion	26-34
6	Java Program Execution	35-92
7	Web Database Application and its Execution	93-98

# **TASK 1**

## ENTITY RELATIONSHIP DIAGRAM



## **TASK 2**

### **RELATIONAL DATABASE SCHEMA**

Account(account\_num, date\_account\_established)  
Assembly-Account(account\_num, cost\_details\_1)  
Process-Account(account\_num, cost\_details\_3)  
Department-Account(account\_num, cost\_details\_2)  
Transactions(transaction\_num, account\_num, sup-cost)  
Customer(customer\_name, address, category)  
Assemblies(assembly\_id, date\_ordered, assembly\_details)  
Processes(process\_id, process\_data)  
Department(department\_num, department\_data)  
Supervised(process\_id, department\_num)  
Manufacture(assembly\_id, process\_id)  
Orders(assembly\_id, customer\_name)  
Assigns(assembly\_id, process\_id, job\_no)  
Records(transaction\_num, job\_no)  
Fit(fit\_type, process\_id)  
Paint(paint\_type, painting\_method, process\_id)  
Cut(process\_id, cutting\_type, machine\_type)  
Job(job\_no, job\_commenced\_date, job\_completed\_date)  
Cut\_Job(job\_no, machine type used, time machine used, material used, labour time)  
Paint\_Job(job\_no, color, volume, labor time)  
Fit\_Job(job\_no, labor time)  
Maintain\_Assembly(assembly\_id, account\_num)  
Maintain\_Process(process\_id, account\_num)  
Maintain\_Department(department\_num, account\_num)

## **TASK 3**

### **3.1**

#### **STORAGE STRUCTURE DISCUSSION FOR EACH TABLE IN SCHEMA**

Table Name	Query and Type	Search Key	Query Frequency	Selected File Organisation	justifications
Account	5(insertion)	account_num	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Assembly-Account	5(insertion)	account_num	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Process-Account	5(insertion)	account_num	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Department-Account	5(insertion)	account_num	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Transaction	9(Random Search)	transaction_num	200/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent and fast retrieval
Customer	1(insertion) 12(rangesearch)	Customer_name Category	30/day 100/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Assemblies	4(insertion) 9(Random Search)	assembly_id	40/day 200/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent and fast retrieval
Processes	3(insertion)	process_id	infrequent	B+Tree	B+Trees efficiently supports insertion

Department	2(insertion)	department_num	infrequent	B+Tree	B+Trees because its insertion and the department which is inserted might be used more often
Supervised	3(insertion) 10(random Search) 11(random Search)	process_id	Infrequent 20/day 100/day	Heap	Heap is used as initially the data is stored but it is not used frequently
Manufacture	4(insertion)		40/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Orders	4(insertion)	assembly_id	40/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Assigns	6(insertion) 10(random Search) 11(random Search)	Assembly_id process_id	50/day 20/day 100/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Fit	3(insertion)	job_no	infrequent	Heap	Heap is used as initially the data is stored but it is not used frequently
Paint	3(insertion)	job_no	infrequent	Heap	Heap is used as initially the data is stored but it is not used frequently
Cut	3(insertion)	job_no	infrequent	Heap	Heap is used as initially the data is stored but it is not used frequently

Job	6(insertion) 7(Random Search) 10(random Search) 11(random Search)	job_no	50/day 50/day 20/day 100/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Paint_Job	7(insertion) 14(insertion)	job_no	50/day 1/week	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Cut_Job	7(insertion) 13(deletion)	job_no	50/day 1/month	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Fit_Job	7(insertion) 10(random Search)	job_no	50/day 20/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Maintain_Assembly	5(insertion) 9(Random Search)	assembly_id	10/day 200/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent and fast retrieval
Maintain_Process	5(insertion)	process_id	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent
Maintain_Department	5(insertion)	department_num	10/day	B+Tree	B+Trees efficiently supports insertion where the query is more frequent

## 3.2

When implementing a relational database in Azure SQL Database, it's essential to consider the choices of storage structures for each table based on the specific requirements and usage patterns of the application.

**Assembly\_Account, Process\_Account, Department\_Account:** These tables appear to be related to financial data. A clustered index on the primary key (account\_num) and possibly a non-clustered index on the foreign key (account\_num) for each table would be appropriate.

**Transactions:** A clustered index on the primary key (transaction\_num) and a non-clustered index on the foreign key (account\_num) for efficient retrieval.

**Customer:** A clustered index on the primary key (customer\_name) for efficient retrieval. non-clustered index on the category column if it's frequently used in queries.

**Assemblies:** A clustered index on the primary key (assembly\_id) for efficient retrieval of assembly-related data.

Considering the online queries and their frequencies, the indexing strategy is crucial. Ensuring that the frequently queried columns and those involved in JOIN operations should have appropriate indexes. Also, monitoring and optimising the database regularly using tools provided by Azure SQL Database to ensure optimal performance based on the actual usage patterns of your application is required

## TASK4

### SOL STATEMENTS FOR CREATION OF TABLES IN AZURE SOL DATABASE

All the tables were created and the code executed successfully. Attached the below Screenshots

```
25
26 -----Table Creation-----
27
28 CREATE TABLE Account(
29     account_num INT PRIMARY KEY,
30     date_account_established DATE
31 );
32 CREATE NONCLUSTERED INDEX idx_account_num ON Account (account_num);
33
34 CREATE TABLE Assembly_Account (
35     account_num INT PRIMARY KEY,
36     cost_details_1 DECIMAL,
37     FOREIGN KEY (account_num) REFERENCES Account(account_num)
38 );
39
40 CREATE TABLE Process_Account (
41     account_num INT PRIMARY KEY,
42     cost_details_3 DECIMAL,
43     FOREIGN KEY (account_num) REFERENCES Account(account_num)
44 );
```

#### Messages

```
02:22:16 Started executing query at Line 1
          Commands completed successfully.
          Total execution time: 00:00:00.346
```

### SQL STATEMENTS FOR CREATION OF TABLES

```
CREATE TABLE Account(
    account_num INT PRIMARY KEY,
    date_account_established DATE
);
CREATE NONCLUSTERED INDEX idx_account_num ON Account (account_num);

CREATE TABLE Assembly_Account (
    account_num INT PRIMARY KEY,
    cost_details_1 DECIMAL,
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);

CREATE TABLE Process_Account (
    account_num INT PRIMARY KEY,
    cost_details_3 DECIMAL,
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);

CREATE TABLE Department_Account (
    account_num INT PRIMARY KEY,
    cost_details_2 DECIMAL,
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);

CREATE TABLE Transactions (
    transaction_num INT PRIMARY KEY,
    sup_cost DECIMAL,
    account_num INT,
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);
CREATE NONCLUSTERED INDEX idx_transactions_transaction_num ON Transactions (transaction_num);
CREATE INDEX idx_transactions_account_num ON Transactions (account_num);

CREATE TABLE Customer (
    customer_name VARCHAR(255) PRIMARY KEY,
    customer_address VARCHAR(255),
    category INT check (category between 1 and 10)
);
CREATE INDEX idx_customer_name ON Customer (customer_name);
CREATE INDEX idx_customer_category ON Customer (category);

CREATE TABLE Assemblies (
    assembly_id INT PRIMARY KEY,
    date_ordered DATE,
    assembly_details VARCHAR(255)
);
CREATE INDEX idx_assembly_id ON Assemblies (assembly_id);

CREATE TABLE Processes(
    process_id INT PRIMARY KEY,
    process_data VARCHAR(255)
);
```

```

CREATE TABLE Department(
    department_num INT PRIMARY KEY,
    department_data VARCHAR(255)
);

CREATE TABLE Job (
    job_no INT PRIMARY KEY,
    job_commenced_date DATE,
    job_completed_date DATE,
);
CREATE INDEX idx_job_no ON Job (job_no);
CREATE INDEX idx_job_job_completed_date ON Job (job_completed_date);

CREATE TABLE Supervised(
    process_id INT PRIMARY KEY,
    department_num INT
    FOREIGN KEY (department_num) REFERENCES Department(department_num),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id)
);
CREATE INDEX idx_supervised_department_num ON Supervised (department_num)

CREATE TABLE Manufacture (
    assembly_id INT ,
    process_id INT,
    PRIMARY KEY (assembly_id, process_id),
    FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id)
);
CREATE INDEX idx_manufacture_assembly_id ON Manufacture (assembly_id);

CREATE TABLE Orders (
    assembly_id INT PRIMARY KEY,
    customer_name VARCHAR(255),
    FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id),
    FOREIGN KEY (customer_name) REFERENCES Customer(customer_name)
);
CREATE INDEX idx_orders_customer_name ON Orders (customer_name);

CREATE TABLE Assigns (
    assembly_id INT,
    process_id INT ,
    job_no INT,
    PRIMARY KEY (assembly_id, process_id),
    FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id),
    FOREIGN KEY (job_no) REFERENCES Job(job_no)
);
CREATE INDEX idx_assigns_assembly_id ON Assigns (assembly_id);
CREATE INDEX idx_assigns_process_id ON Assigns (process_id);

CREATE TABLE ...

```

```
CREATE TABLE Fit (
    process_id INT PRIMARY KEY,
    fit_type VARCHAR(255),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id)
);

CREATE TABLE Paint (
    process_id INT PRIMARY KEY,
    paint_type VARCHAR(255),
    painting_method VARCHAR(255),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id)
);

CREATE TABLE Cut (
    process_id INT PRIMARY KEY,
    cutting_type VARCHAR(255),
    machine_type VARCHAR(255),
    FOREIGN KEY (process_id) REFERENCES Processes(process_id)
);

CREATE TABLE Cut_Job (
    job_no INT PRIMARY KEY,
    machine_type_used VARCHAR(255),
    time_machine_used DECIMAL,
    material_used VARCHAR(255),
    labour_time DECIMAL,
    FOREIGN KEY (job_no) REFERENCES Job(job_no)
);

CREATE TABLE data type int (
    job_no INT PRIMARY KEY,
    color VARCHAR(255),
    volume DECIMAL,
    labour_time DECIMAL,
    FOREIGN KEY (job_no) REFERENCES Job(job_no)
);

CREATE TABLE Fit_Job (
    job_no INT PRIMARY KEY,
    labour_time DECIMAL,
    FOREIGN KEY (job_no) REFERENCES Job(job_no)
);

CREATE TABLE Records (
    transaction_num INT PRIMARY KEY,
    job_no INT,
    FOREIGN KEY (transaction_num) REFERENCES Transactions(transaction_num),
    FOREIGN KEY (job_no) REFERENCES Job(job_no)
);
```

```
CREATE TABLE Maintain_Assembly (
    assembly_id INT PRIMARY KEY,
    account_num INT,
    FOREIGN KEY (assembly_id) REFERENCES Assemblies(assembly_id),
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);
CREATE INDEX idx_maintain_assembly_assembly_id ON Maintain_Assembly (assembly_id);

CREATE TABLE Maintain_Process (
    process_id INT PRIMARY KEY,
    account_num INT,
    FOREIGN KEY (process_id) REFERENCES Processes(process_id),
    FOREIGN KEY (process_id) REFERENCES Account(account_num)
);

CREATE TABLE Maintain_Department (
    department_num INT PRIMARY KEY,
    account_num INT,
    FOREIGN KEY (department_num) REFERENCES Department(department_num),
    FOREIGN KEY (account_num) REFERENCES Account(account_num)
);
```

# TASK 5

## 5.1

### SQL IMPLEMENTING ALL QUERIES

```
19 ----- 1. Create the procedure for InsertNewCustomer-----
20 GO
21 CREATE PROCEDURE EnterNewCustomer
22     @new_customer_name NVARCHAR(255),
23     @new_customer_address NVARCHAR(255),
24     @new_category INT
25 AS
26 BEGIN
27     INSERT INTO Customer (customer_name, customer_address, category)
28     VALUES (@new_customer_name, @new_customer_address, @new_category);
29 END;
30
31 -----2.Enter a new department-----
32 GO
33 CREATE PROCEDURE EnterNewDepartment
34     @new_department_num INT,
35     @new_department_data NVARCHAR(255)
36 AS
37 BEGIN
38     -- Insert the new department without checking for duplicates
39     INSERT INTO department (department_num, department_data)
40     VALUES (@new_department_num, @new_department_data);
41 END.
```

#### Messages

```
07:35:29 Started executing query at Line 1
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 21
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 33
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 45
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 97
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 132
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 223
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 260
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 311
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 353
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 372
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 445
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 467
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 490
07:35:29 Commands completed successfully.
07:35:29 Started executing query at Line 506
07:35:29 Commands completed successfully.
Total execution time: 00:00:01.119
```

#### ----- 1. Create the procedure for InsertNewCustomer-----

```
GO
CREATE PROCEDURE EnterNewCustomer
    @new_customer_name NVARCHAR(255),
    @new_customer_address NVARCHAR(255),
    @new_category INT
AS
BEGIN
    INSERT INTO Customer (customer_name, customer_address, category)
    VALUES (@new_customer_name, @new_customer_address, @new_category);
END;
```

---

-----2.Enter a new department-----

```
GO
CREATE PROCEDURE EnterNewDepartment
    @new_department_num INT,
    @new_department_data NVARCHAR(255)
AS
BEGIN
    -- Insert the new department without checking for duplicates
    INSERT INTO department (department_num, department_data)
    VALUES (@new_department_num, @new_department_data);
END;
```

GO

CREATE PROCEDURE EnterNewProcess

```

    @process_id INT,
    @department_num INT,
    @process_type VARCHAR(255),
    @process_info VARCHAR(255),
    @machine_type VARCHAR(255) ,
    @cutting_type VARCHAR(255),
    @painting_method VARCHAR(255),
    @paint_type VARCHAR(255),
    @process_data VARCHAR(255)

```

AS

BEGIN

```
-- Check if the process and department exist
```

```
IF EXISTS (SELECT 1 FROM Processes WHERE process_id = @process_id) AND
   EXISTS (SELECT 1 FROM Department WHERE department_num = @department_num)
```

BEGIN

```

    PRINT 'Process and department already exist.';
    RETURN;

```

END

```
--Insert into the process table
```

```
INSERT INTO Processes(process_id,process_data)
VALUES(@process_id, @process_data)
```

```
-- Insert the new process information into the Supervised table
```

```
INSERT INTO Supervised (process_id, department_num)
VALUES (@process_id, @department_num);
```

```
-- Insert the process details into the respective table based on the type
```

```
IF @process_type = 'Cut'
```

BEGIN

```

    INSERT INTO Cut (process_id, cutting_type, machine_type)
    VALUES (@process_id, @cutting_type, @machine_type);

```

END

```
ELSE IF @process_type = 'Paint'
```

BEGIN

```

    INSERT INTO Paint (process_id, paint_type, painting_method)
    VALUES (@process_id, @paint_type, @painting_method);

```

END

```
ELSE IF @process_type = 'Fit'
```

BEGIN

```

    INSERT INTO Fit (process_id, fit_type)
    VALUES (@process_id, @process_info);

```

END

```
ELSE
```

BEGIN

```

    PRINT 'Invalid process type.';
    -- You can handle the case where an invalid type is provided.

```

END

END;

-----4. Enter a new assembly with its customer-name, assembly-det

```
GO
CREATE PROCEDURE NewAssembly
    @assembly_id INT,
    @date_ordered VARCHAR(255),
    @customer_name VARCHAR(255),
    @assembly_details VARCHAR(255),
    @process_id INT
AS
BEGIN
    -- Check if the assembly ID exists
    IF EXISTS (SELECT 1 FROM Assemblies WHERE assembly_id = @assembly_id)
    BEGIN
        PRINT 'Assembly ID exists.';
    END
    IF NOT EXISTS (SELECT 1 FROM Processes WHERE process_id = @process_id)
    BEGIN
        PRINT 'Process ID does not exist.';
        RETURN;
    END

    -- Insert data into the Assemblies table
    INSERT INTO Assemblies (assembly_id, date_ordered, assembly_details)
    VALUES (@assembly_id, @date_ordered, @assembly_details);

    -- Insert data into the Orders table
    INSERT INTO Orders (assembly_id, customer_name)
    VALUES (@assembly_id, @customer_name);

    -- Insert data into the Manufacture table
    INSERT INTO Manufacture (assembly_id, process_id)
    VALUES (@assembly_id, @process_id);
END;
```

-----5.Create a new account and associate it with the process, assembly, or department to v

```
GO
CREATE PROCEDURE CreateAccount
    @account_num INT,
    @date_account_established VARCHAR(15),
    @process_id INT ,
    @cost_details_3 DECIMAL ,
    @cost_details_2 DECIMAL ,
    @cost_details_1 DECIMAL ,
    @department_data VARCHAR(255) ,
    @assembly_details VARCHAR(255),
    @account_type VARCHAR(50),
    @department_num INT,
    @assembly_id INT
AS
BEGIN
    -- Check if the account exists
    IF EXISTS (SELECT 1 FROM Account WHERE account_num = @account_num)
    BEGIN
        PRINT 'Account already exists.';
        RETURN;
    END

    -- Create a new account
    INSERT INTO Account (account_num, date_account_established)
    VALUES (@account_num, @date_account_established);

    -- Prompt the user for the type of account if not provided
    IF @account_type IS NULL
    BEGIN
        SET @account_type = (SELECT UPPER(
            CAST(
                (SELECT N'Enter account type (Process, Department, or Assembly): ') AS NVARCHAR(MAX)
            )
        ));

        WHILE @account_type NOT IN ('PROCESS', 'DEPARTMENT', 'ASSEMBLY')
        BEGIN
            PRINT 'Invalid account type. Please enter Process, Department, or Assembly.';
            SET @account_type = (SELECT UPPER(
                CAST(
                    (SELECT N'Enter account type (Process, Department, or Assembly): ') AS NVARCHAR(MAX)
                )
            ));
        END
    END
END

    -- Insert into the respective account type table
    IF @account_type = 'PROCESS'
    BEGIN
        -- Insert into Process table
        ...
    END
    ELSE IF @account_type = 'DEPARTMENT'
    BEGIN
        -- Insert into Department table
        ...
    END
    ELSE IF @account_type = 'ASSEMBLY'
    BEGIN
        -- Insert into Assembly table
        ...
    END
```

```

IF @account_type = 'PROCESS'
BEGIN
    -- Insert into Process_Account table
    INSERT INTO Process_Account (account_num, cost_details_3)
    VALUES (@account_num, @cost_details_3);

    -- Maintain the process through Maintain_Process
    IF @process_id IS NOT NULL
    BEGIN
        INSERT INTO Maintain_Process (process_id, account_num)
        VALUES (@process_id, @account_num);
    END
END
ELSE IF @account_type = 'DEPARTMENT'
BEGIN
    -- Insert into Department_Account table
    INSERT INTO Department_Account (account_num, cost_details_2)
    VALUES (@account_num, @cost_details_2);

    IF @department_num IS NOT NULL
    BEGIN
        INSERT INTO Maintain_Department (department_num, account_num)
        VALUES (@department_num, @account_num);
    END
END
ELSE IF @account_type = 'ASSEMBLY'
BEGIN
    -- Insert into Assembly_Account table
    INSERT INTO Assembly_Account (account_num, cost_details_1)
    VALUES (@account_num, @cost_details_1);

    IF @assembly_id IS NOT NULL
    BEGIN
        INSERT INTO Maintain_Assembly (assembly_id, account_num)
        VALUES (@assembly_id, @account_num);

        -- Connect to the Assembly table
        -- Add your logic here based on the requirements
    END
END;

```

-----6.Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced -----

GO

CREATE PROCEDURE EnterNewJob

```
@job_no INT,
@assembly_id INT,
@process_id INT,
@job_commenced_date VARCHAR(20),
@job_type VARCHAR(15),
@color VARCHAR(255),
@volume VARCHAR(255),
@machine_type_used VARCHAR(255),
@material_used VARCHAR

AS
BEGIN
    INSERT INTO Job(job_no, job_commenced_date, job_completed_date)
    VALUES (@job_no, @job_commenced_date, NULL)
    INSERT INTO Assigns (assembly_id, process_id, job_no)
    VALUES (@assembly_id, @process_id, @job_no);
    IF @job_type = 'paint'
        BEGIN
            INSERT INTO Paint_Job(job_no, color, volume, labour_time)
            VALUES (@job_no, @color, @volume, NULL)
        END
    ELSE IF @job_type = 'fit'
        BEGIN
            INSERT INTO Fit_Job(job_no, labour_time)
            VALUES (@job_no, NULL)
        END
    ELSE IF @job_type = 'cut'
        BEGIN
            INSERT INTO Cut_Job(job_no, machine_type_used, time_machine_used, material_used, labour_time)
            VALUES (@job_no, @machine_type_used, NULL, @material_used, NULL)
        END
END;
```

-----7. At the completion of a job, enter the date it completed and the information relevant

GO

```
CREATE PROCEDURE InsertJobData
```

```
    @job_no INT,  
    @completion_date VARCHAR(15),  
    @process_type VARCHAR(225),  
    @labour_time DECIMAL(10,2),  
    @time_machine_used DECIMAL(10,2)
```

```
AS
```

```
BEGIN
```

```
    UPDATE Job  
    SET job_completed_date = @completion_date  
    WHERE job_no = @job_no;
```

```
    SELECT @process_type =  
        CASE  
            WHEN EXISTS (SELECT 1 FROM Cut_Job WHERE job_no = @job_no) THEN 'Cut'  
            WHEN EXISTS (SELECT 1 FROM Fit_Job WHERE job_no = @job_no) THEN 'Fit'  
            WHEN EXISTS (SELECT 1 FROM Paint_Job WHERE job_no = @job_no) THEN 'Paint'  
            ELSE NULL  
        END;
```

```
    IF @process_type = 'Cut'
```

```
        BEGIN
```

```
            Update Cut_Job SET  
            time_machine_used = @time_machine_used,  
            labour_time = @labour_time  
            WHERE job_no = @job_no;
```

```
        END
```

```
    ELSE IF @process_type = 'Fit'
```

```
        BEGIN
```

```
            Update Fit_Job SET  
            labour_time = @labour_time  
            WHERE job_no = @job_no;
```

```
        END
```

```
    ELSE IF @process_type = 'Paint'
```

```
        BEGIN
```

```
            Update Paint_Job SET  
            labour_time = @labour_time  
            WHERE job_no = @job_no;
```

```
        END
```

```
    END;
```

-----8. Enter a transaction-no and its sup-cost and update all the costs (details) of the

```
GO
CREATE PROCEDURE enterTransactionNum
    @transaction_num INT,
    @sup_cost DECIMAL,
    @account_num INT
AS
BEGIN
    -- Check if the transaction number already exists
    IF EXISTS (SELECT 1 FROM Transactions WHERE transaction_num = @transaction_num)
    BEGIN
        PRINT 'Transaction number already exists.';
        RETURN;
    END

    INSERT INTO Transactions(transaction_num,sup_cost, account_num) VALUES
    (@transaction_num, @sup_cost, @account_num)

    IF EXISTS (SELECT 1 FROM Process_Account WHERE account_num = @account_num)
    BEGIN
        UPDATE Process_Account
        SET cost_details_3 = cost_details_3 + @sup_cost
        WHERE account_num = @account_num;
    END

    -- Update assembly account details
    IF EXISTS (SELECT 1 FROM Assembly_Account WHERE account_num = @account_num)
    BEGIN
        UPDATE Assembly_Account
        SET cost_details_1 = cost_details_1 + @sup_cost
        WHERE account_num = @account_num;
    END

    -- Update department account details
    IF EXISTS (SELECT 1 FROM Department_Account WHERE account_num = @account_num)
    BEGIN
        UPDATE Department_Account
        SET cost_details_2 = cost_details_2 + @sup_cost
        WHERE account_num = @account_num;
    END
END;
```

-----9. Retrieve the total cost incurred on an assembly-id -----

```
GO
CREATE PROCEDURE GetTotalCostForAssembly
    @assembly_id INT,
    @total_cost DECIMAL(10, 2) OUTPUT
AS
BEGIN
    -- Calculate total cost
    SELECT @total_cost = COALESCE(SUM(t.sup_cost), 0)
    FROM Assemblies AS a
    LEFT JOIN Maintain_Assembly AS ma ON a.assembly_id = ma.assembly_id
    LEFT JOIN Transactions AS t ON ma.account_num = t.account_num
    WHERE a.assembly_id = @assembly_id;

    -- Print the result
    -- PRINT 'Total Cost for Assembly ID ' + CAST(@assembly_id AS VARCHAR) + ': ' + CAST(@total_cost AS VARCHAR);
END;
```

-----10. Retrieve the total labor time within a department for jobs completed in the department

GO

CREATE PROCEDURE GetTotalLaborTime

    @department\_num INT,  
    @completion\_date VARCHAR(15),  
    @total\_labor\_time DECIMAL(10, 2) OUTPUT

AS

BEGIN

    -- Declare a variable to store the total labor time

    -- Calculate the total labor time

    SELECT @total\_labor\_time = ISNULL(SUM(labour\_time), 0)  
    FROM (

        SELECT labour\_time  
        FROM Fit\_Job AS fj  
        WHERE fj.job\_no IN (  
            SELECT job\_no  
            FROM Job  
            WHERE job\_completed\_date = @completion\_date  
            AND job\_no IN (  
                SELECT job\_no  
                FROM Assigns  
                WHERE process\_id IN (  
                    SELECT process\_id  
                    FROM Supervised  
                    WHERE department\_num = @department\_num  
                )

    )

    UNION ALL

        SELECT labour\_time  
        FROM Paint\_Job AS pj  
        WHERE pj.job\_no IN (  
            SELECT job\_no  
            FROM Job  
            WHERE job\_completed\_date = @completion\_date  
            AND job\_no IN (  
                SELECT job\_no  
                FROM Assigns  
                WHERE process\_id IN (  
                    SELECT process\_id  
                    FROM Supervised  
                    WHERE department\_num = @department\_num  
                )

    )

    UNION ALL

        SELECT labour\_time  
        FROM Cut\_Job AS cj  
        WHERE cj.job\_no IN (  
            SELECT job\_no  
            FROM Job  
            WHERE job\_completed\_date = @completion\_date  
            AND job\_no IN (  
                SELECT job\_no  
                FROM Assigns  
                WHERE process\_id IN (  
                    SELECT process\_id  
                    FROM Supervised  
                    WHERE department\_num = @department\_num  
                )

    )

) AS LaborTimes;

    -- Print the total labor time

    PRINT 'Total Labor Time in Department ' + CAST(@department\_num AS VARCHAR) + ' for Jobs Completed on ' + CAST(@completion\_date AS VARCHAR) + ':' + CAST(@total\_labor\_time AS VARCHAR);

END;

-----11. Retrieve the processes through which a

```
GO
CREATE PROCEDURE GetProcessDepartmentDetails
    @assembly_id INT
AS
BEGIN
    SELECT
        S.process_id,
        S.department_num
    FROM
        Supervised AS S
    INNER JOIN
        Assigns AS A ON S.process_id = A.process_id
    INNER JOIN
        Job AS J ON J.job_no = A.job_no
    WHERE
        A.assembly_id = @assembly_id
    ORDER BY
        J.job_commenced_date;
END;
```

-----12. Retrieve the customers (in name order) whose category is in a given range-----

```
GO
CREATE PROCEDURE GetCustomersByCategoryRange
    @start_category INT,
    @end_category INT
AS
BEGIN
    -- Print the header
    PRINT 'Customer Names in Category Range ' + CAST(@start_category AS VARCHAR) + ' to ' + CAST(@end_category AS VARCHAR) + ':';
    PRINT '-----';

    -- Select and print customer names
    SELECT
        customer_name
    FROM
        Customer
    WHERE
        category BETWEEN @start_category AND @end_category
    ORDER BY
        customer_name;
END;
```

GO

```
CREATE PROCEDURE DeleteCutJobsByJobNoRange
    @start_job_no INT,
    @end_job_no INT
AS
BEGIN
    -- Delete cut-jobs
    DELETE FROM Cut_Job
    WHERE
        job_no BETWEEN @start_job_no AND @end_job_no;

    -- Print a message indicating the deletion
    PRINT 'Cut-Jobs with Job No in the range ' + CAST(@start_job_no AS VARCHAR) + ' to ' + CAST(@end_job_no AS VARCHAR) + ' deleted.';
END;
```

-----14. Change the color of a given paint job-----

```
GO
CREATE PROCEDURE ChangePaintJobColor
    @job_no INT,
    @new_color VARCHAR(255)
AS
BEGIN
    -- Update the color of the paint job
    UPDATE Paint_Job
    SET color = @new_color
    WHERE job_no = @job_no;

    -- Print a message indicating the update
    PRINT 'Color of Paint Job with Job No ' + CAST(@job_no AS VARCHAR) + ' changed to ' + @new_color + '.';
END;
```

-----16. Export Customers-----

```
GO
CREATE PROCEDURE ExportCustomer
    @min INT,
    @max INT
AS
BEGIN
    SELECT * FROM Customer
    WHERE category BETWEEN @min and @max
END;
```

## 5.2

# JAVA SOURCE PROGRAM AND ITS EXECUTION

```
import java.sql.Connection;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Types;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.time.format.DateTimeFormatter;
import java.io.*;
import java.math.BigDecimal;
import java.sql.CallableStatement;

public class indProject {
    // Database credentials
    final static String HOSTNAME = "sahu0001-sql-server.database.windows.net";
    final static String DBNAME = "cs-dsa-4513-sql-db";
    final static String USERNAME = "sahu0001";
    final static String PASSWORD = "Jaijagannath2023@";
    // Database connection string
    final static String URL = String.format("jdbc:sqlserver://:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=FALSE;loginTimeout=30;socketTimeout=30;connectionTimeout=30;authenticationScheme=Windows;%s", HOSTNAME, DBNAME, USERNAME, PASSWORD);
    // Query templates
    final static String QUERY_TEMPLATE_1 = "EXEC EnterNewCustomer @new_customer_name = ?, @new_customer_address = ?, @new_category = ?;";
    final static String QUERY_TEMPLATE_2 = "EXEC EnterNewDepartment @new_department_num = ?, @new_department_data = ?;";
    final static String QUERY_TEMPLATE_3 = "EXEC EnterNewProcess @process_id = ?, @department_num = ?, @process_type = ?, @process_info=?, @machine_type=?, @cutting_time=?";
    final static String QUERY_TEMPLATE_4 = "EXEC NewAssembly @assembly_id = ?, @date_ordered = ?, @customer_name = ?, @assembly_details = ?, @process_id = ?;";
    final static String QUERY_TEMPLATE_5 = "EXEC CreateAccount @account_num = ?, @date_account_established = ?, @process_id=?, @cost_details_3=?, @cost_details_2=?, @volume=?";
    final static String QUERY_TEMPLATE_6 = "EXEC EnterNewJob @job_no = ?, @assembly_id = ?, @process_id=?, @job_commenced_date=?, @job_type=?, @color=?, @volume=?";
    final static String QUERY_TEMPLATE_7 = "EXEC InsertJobData @job_no = ?, @completion_date=?, @process_type=?, @labour_time=?, @time_machine_used=?;";
    final static String QUERY_TEMPLATE_8 = "EXEC enterTransactionNum @transaction_num = ?, @sup_cost = ?, @account_num=?;";
    final static String QUERY_TEMPLATE_9 = "EXEC GetTotalCostForAssembly @assembly_id=?, @total_cost=?;";
    final static String QUERY_TEMPLATE_10 = "EXEC GetTotalLaborTime @department_num=?, @completion_date=?, @total_labor_time=? ;";
    final static String QUERY_TEMPLATE_11 = "EXEC GetProcessDepartmentDetails @assembly_id=?";
    final static String QUERY_TEMPLATE_12 = "EXEC GetCustomersByCategoryRange @start_category=?, @end_category=?;";
    final static String QUERY_TEMPLATE_13 = "EXEC DeleteCutJobsByJobNoRange @start_job_no=?, @end_job_no=?;";
    final static String QUERY_TEMPLATE_14 = "EXEC ChangePaintJobColor @job_no=?, @new_color=?;";

    final static String QUERY_TEMPLATE_15 = "SELECT * FROM Performer;";
    final static String QUERY_TEMPLATE_16 = "SELECT * FROM Performer;";
    final static String QUERY_TEMPLATE_17 = "SELECT * FROM Performer;";

    final static DateTimeFormatter dtf = DateTimeFormatter.ofPattern("MM-dd-yyyy");

    // User input prompt
    final static String PROMPT = "\nPlease select one of the options below: \n" +
        "(1) Enter a new customer \n" +
        "(2) Enter a new department \n" +
        "(3) Enter a new process-id and its department together with its type and information \n" +
        "(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, \n" +
        "(5) Create a new account and associate it with the process, assembly, or department \n" +
        "(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced\n" +
        "(7) At the completion of a job, enter the date it completed and the information \n" +
        "(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the \n" +
        "(9) Retrieve the total cost incurred on an assembly-id \n" +
        "(10) Retrieve the total labor time within a department for jobs completed in the \n" +
        "(11) Retrieve the processes through which a given assembly-id has passed so far \n" +
        "(12) Retrieve the department during a given date\n" +
        "(13) Retrieve the processes through which a given assembly-id has passed so far \n" +
        "(14) Change the paint color of a job by its job number and new color\n" + ..
```

```

// User input prompt
final static String PROMPT =
"\nPlease select one of the options below: \n" +
"(1) Enter a new customer \n" +
"(2) Enter a new department \n" +
"(3) Enter a new process-id and its department together with its type and information \n" +
"\trelevant to the type\n" +
"(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, \n" +
"\tand dateordered and associate it with one or more processes\n" +
"(5) Create a new account and associate it with the process, assembly, or department \n" +
"\tto which it is applicable\n" +
"(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced\n" +
"(7) At the completion of a job, enter the date it completed and the information \n" +
"\trelevant to the type of job \n" +
"(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the \n" +
"\taffected accounts by adding sup-cost to their current values of details \n" +
"(9) Retrieve the total cost incurred on an assembly-id \n" +
"(10) Retrieve the total labor time within a department for jobs completed in the \n" +
"\tdepartment during a given date\n" +
"(11) Retrieve the processes through which a given assembly-id has passed so far \n" +
"\t(in datecommenced order) and the department responsible for each process\n" +
"(12) Retrieve the customers (in name order) whose category is in a given range\n" +
"(13) Delete all cut-jobs whose job-no is in a given range\n" +
"(14) Change the color of a given paint job\n" +
"(15) Import: enter new customers from a data file until the file is empty \n" +
"(16) Export: Retrieve the customers (in name order) whose category is in a given range \n" +
"\tand output them to a data file instead of screen (the user must
"(17) Quit\n";

public static void main(String[] args) throws SQLException, IOException {
    System.out.println("WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM !");
    final Scanner sc = new Scanner(System.in);

    String option = "";
    while (!option.equals("17")) {
        System.out.println(PROMPT);
        System.out.println("Enter your option : ");
        option = sc.nextLine();

        switch (option) {
            case "1":
                System.out.println("name:");
                final String new_customer_name = sc.nextLine();

                System.out.println("address:");
                final String new_customer_address = sc.nextLine();

                System.out.println("category:");
                final int new_category = sc.nextInt();
                System.out.println("Connecting to the database...");

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_1)) {
                        statement.setString(1, new_customer_name);
                        statement.setString(2, new_customer_address);
                        statement.setInt(3, new_category);
                        statement.execute();
                    }
                }
                break;
            case "2":
                System.out.println("department number:");
                final int new_department_num = sc.nextInt();

                System.out.println("department data:");
                final String new_department_data = sc.nextLine();

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

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_2)) {
                        statement.setInt(1, new_department_num);
                        statement.setString(2, new_department_data);
                        statement.execute();
                    }
                }
                break;
        }
    }
}

```

```

case "3":
    System.out.println("process id:");
    final int new_process_id = sc.nextInt();

    System.out.println("department number:");
    final int new_department_num1 = sc.nextInt();

    System.out.println("process_type (Cut, Paint, Fit) case-sensitive:");
    final String new_type = sc.next();

    System.out.println("new info:");
    final String new_information = sc.next();

    System.out.println("machine type:");
    final String machine_type = sc.next();

    System.out.println("cutting type:");
    final String cutting_type = sc.next();

    System.out.println("painting method:");
    final String painting_method = sc.next();

    System.out.println("paint type:");
    final String paint_type = sc.next();

    System.out.println("process data:");
    final String process_data = sc.next();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        System.out.println("Dispatching the query...");
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_3)) {
            statement.setInt(1, new_process_id);
            statement.setInt(2, new_department_num1);
            statement.setString(3, new_type);
            statement.setString(4, new_information);
            statement.setString(5, machine_type);
            statement.setString(6, cutting_type);
            statement.setString(7, painting_method);
            statement.setString(8, paint_type);
            statement.setString(9, process_data);
            statement.execute();
        }
    }
    break;
}

}

break;
case "4":
    System.out.println("assembly id param:");
    final int assembly_id_param = sc.nextInt();

    System.out.println("date ordered param (MM-dd-yyyy):");
    final String date_ordered_param = sc.next();

    System.out.println("customer name param:");
    final String customer_name_param = sc.next();

    System.out.println("assembly ordered param:");
    final String assembly_details_param = sc.next();

    System.out.println("process id param:");
    final String process_ids_param = sc.next();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_4)) {
            statement.setInt(1, assembly_id_param);
            statement.setString(2, date_ordered_param);
            statement.setString(3, customer_name_param);
            statement.setString(4, assembly_details_param);
            statement.setString(5, process_ids_param);
            statement.execute();
        }
    }
    break;
}

```

```

case "5":
    System.out.println("account number:");
    final int account_num = sc.nextInt();

    System.out.println("date established:");
    final String date_established = sc.next();

    System.out.println("process id:");
    final int process_id = sc.nextInt();

    System.out.println("cost detail 3:");
    final float cost_details_3 = sc.nextFloat();

    System.out.println("cost detail 2:");
    final float cost_details_2 = sc.nextFloat();

    System.out.println("cost detail 1:");
    final float cost_details_1 = sc.nextFloat();

    System.out.println("dept data:");
    final String department_data = sc.next();

    System.out.println("assembly details:");
    final String assembly_details = sc.next();

    System.out.println("account type (PROCESS, DEPARTMENT, ASSEMBLY) case-sensitive:");
    final String account_type = sc.next();

    System.out.println("dept number:");
    final int department_num = sc.nextInt();

    System.out.println("assembly id:");
    final int assembly_id = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_5)) {
            statement.setInt(1, account_num);
            statement.setString(2, date_established);
            statement.setInt(3, process_id);
            statement.setFloat(4, cost_details_3);
            statement.setFloat(5, cost_details_2);
            statement.setFloat(6, cost_details_1);
            statement.setString(7, department_data);
            statement.setString(8, assembly_details);
            statement.setString(9, account_type);
            statement.setInt(10, department_num);
            statement.setInt(11, assembly_id);
            statement.execute();
        }
    }
    break;
}

```

```
case "6":
    System.out.println("job no:");
    final int job_no = sc.nextInt();

    System.out.println("assembly id:");
    final int assembly_id1 = sc.nextInt();

    System.out.println("process id:");
    final int process_id1 = sc.nextInt();

    System.out.println("job commenced date:");
    final String job_commenced_date = sc.next();

    System.out.println("job type (paint, fit, cut) case-sensitive:");
    final String job_type = sc.next();

    System.out.println("color:");
    final String color = sc.next();

    System.out.println("volume:");
    final String volume = sc.next();

    System.out.println("machine type used:");
    final String machine_type_used = sc.next();

    System.out.println("material used:");
    final String material_used = sc.next();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_6)) {
            statement.setInt(1, job_no);
            statement.setInt(2, assembly_id1);
            statement.setInt(3, process_id1);
            statement.setString(4, job_commenced_date);
            statement.setString(5, job_type);
            statement.setString(6, color);
            statement.setString(7, volume);
            statement.setString(8, machine_type_used);
            statement.setString(9, material_used);
            statement.execute();
        }
    }
    break;
case "7":
```

```

break;
case "7":
    System.out.println("job no:");
    final int job_no1 = sc.nextInt();

    System.out.println("job competition date:");
    final String completion_date = sc.next();

    System.out.println("process type:");
    final String process_type = sc.next();

    System.out.println("labour time:");
    final float labour_time = sc.nextFloat();

    System.out.println("time machine used:");
    final float time_machine_used = sc.nextFloat();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_7)) {
            cs.setInt(1, job_no1);
            cs.setString(2, completion_date);
            cs.setString(3, process_type);
            cs.setFloat(4, labour_time);
            cs.setFloat(5, time_machine_used);
            // Execute the stored procedure
            cs.execute();
        }
    }
    break;
}

break;
case "8":
    System.out.println("transaction number:");
    final int transaction_num = sc.nextInt();

    System.out.println("sup cost:");
    final float sup_cost = sc.nextFloat();

    System.out.println("account number:");
    final int acc_no = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_8)) {
            statement.setInt(1, transaction_num);
            statement.setFloat(2, sup_cost);
            statement.setInt(3, acc_no);
            statement.execute();
        }
    }
    break;
}

```

```

break;
case "9":

    System.out.println("assembly id:");
    final int assembly_id3 = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_9)) {

            cs.setInt(1, assembly_id3);
            cs.registerOutParameter(2, Types.DECIMAL);
            // Execute the stored procedure
            cs.execute();

            // Retrieve the output parameter value
            BigDecimal totalCost = cs.getBigDecimal(2);
            System.out.println("Total cost: " + totalCost);
        }
    }
    break;
--- "10"
case "10":

    System.out.println("department number:");
    final int department_number = sc.nextInt();

    System.out.println("job completed date:");
    final String job_completed_date2 = sc.next();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_10);
        // Set the assigned value(s) to the procedures input
        cs.setInt(1, department_number);
        cs.setString(2, job_completed_date2);
        // Run the stored procedure and store values in resultSet
        cs.registerOutParameter(3, Types.DECIMAL);
        // Execute the stored procedure
        cs.execute();

        // Retrieve the output parameter value
        BigDecimal totalCost = cs.getBigDecimal(3);
        System.out.println("Total cost: " + totalCost);
    }
    break;
--- "11"
case "11":

    System.out.println("assembly id:");
    final int assembly_id4 = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Prepare a call to the stored procedure
        CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_11);
        // Set the assigned value(s) to the procedures input
        cs.setInt(1, assembly_id4);
        // Run the stored procedure and store values in resultSet
        System.out.println("Dispatching the query...");
        ResultSet resultSet = cs.executeQuery();
        System.out.println("Done.");
        System.out.println("\nProcess for assembly-id: " + assembly_id4 +
                           ", and its department number; Sorted by date commenced.");
        System.out.println("processID | deptNo");
        // Unpack the tuples returned by the database and print them out to the user
        while (resultSet.next()) {
            System.out.println(String.format("%s | %s ",
                                           resultSet.getString(1),
                                           resultSet.getString(2)));
        }
    }
    break;
--- "12"

```

```

break;
case "12":

    System.out.println("start category:");
    final int range_start = sc.nextInt();

    System.out.println("end category:");
    final int range_end = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_12)) {
            // Set the assigned value(s) to the procedures input
            cs.setInt(1, range_start);
            cs.setInt(2, range_end);
            // Run the stored procedure and store values in resultSet
            System.out.println("Dispatching the query...");
            ResultSet resultSet = cs.executeQuery();
            System.out.println("Done.");
            System.out.println("\nJobs from start date " + range_start +
                " completed on: " + range_end);
            System.out.println("customer name");
            // Unpack the tuples returned by the database and print them out to the user
            while (resultSet.next()) {
                System.out.println(String.format("%s",
                    resultSet.getString(1)));
            }
        }
    }
break;
.....

break;
case "13":

    System.out.println("job number start:");
    final int job_number_start = sc.nextInt();

    System.out.println("job number end:");
    final int job_number_end = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_13)) {
            statement.setInt(1, job_number_start);
            statement.setInt(2, job_number_end);
            statement.executeUpdate();
        }
    }
break;
.....

case "14":

    System.out.println("job number:");
    final int job_number = sc.nextInt();

    System.out.println("color:");
    final String color1 = sc.next();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_8)) {
            statement.setInt(1, job_number);
            statement.setString(2, color1);
            int rows = statement.executeUpdate();
            System.out.println(rows);
        }
    }
break;
.....

```

```

case "15":
    System.out.println("Please enter the location and name of a CSV file with customer data:" +
        "\n> ONLY INSERT CSV FILE <>");

    String filename = sc.nextLine();
    // create insert statement with values from csv file
    String query = readCSV(filename);
    // Get a database connection and prepare a query statement

    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Prepare a call to the stored procedure
        PreparedStatement ps = connection.prepareCall(query);
        System.out.println("Dispatching the query...");
        // Actually execute the populated query
        final int rows_inserted = ps.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
    }
    break;
}

case "16":
    // Set the query
    System.out.println("Please enter MIN category number (integer from 1 - 10, inclusive):");
    int min = sc.nextInt();
    System.out.println("Please enter MAX category number (integer from 1 - 10, inclusive):");
    int max = sc.nextInt();
    System.out.println("Please enter the file output name:");
    sc.nextLine();
    filename = sc.nextLine();
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Prepare a call to the stored procedure
        CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_16);
        // Set the assigned value(s) to the procedures input
        cs.setInt("min", min);
        cs.setInt("max", max);
        // Run the stored procedure and store values in resultSet

        System.out.println("Dispatching the query...");
        ResultSet resultSet = cs.executeQuery();
        try {
            FileWriter myWriter = new FileWriter(filename + ".csv");
            myWriter.write("name,address,category\n");
            // Unpack the tuples returned by the database and print them out to the user
            while (resultSet.next()) {
                myWriter.write(String.format("%s,%s,%s\n",
                    resultSet.getString(1),
                    resultSet.getString(2),
                    resultSet.getString(3)));
            }
            // close the writer
            myWriter.close();
        } catch (IOException e) {
            System.out.println("Error with file name.");
            e.printStackTrace();
        }
    }
    System.out.println("Done. File Location here:");
    System.out.println(filename + ".csv");
    break;
case "17":
    System.out.println("Finished! Your work here is done.");
    break;
default:
    System.out.println("In the default part. Something went wrong. Perhaps, wrong option !!");
    break;
}
}

sc.close();
}

```

# TASK 6

## JAVA PROGRAM EXECUTION(TESTING)

### OPTION1:

```
Enter your option :
```

```
1  
name:  
SUJATA  
address:  
OKLAHOMA  
category:  
2  
Connecting to the database...
```

```
Please select one of the options below:
```

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

```
Enter your option :
```

Enter your option :

1

name:

PANKAJ

address:

CALIFORHIA

category:|

4

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty ( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

1

name:

ISHITA

address:

CALIFORNIA

category:|

5

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty ( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

1

name:

MANAV

address:

NEWYORK

category:

8

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

1

name:

PRIYA

address:

HYDERABAD

category:

10

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

	customer_name	customer_address	category
1	ISHITA	CALIFORNIA	5
2	MANAV	NEWYORK	8
3	NARAYAN	DALLAS	7
4	PANKAJ	CALIFORHIA	4
5	PRIYA	HYDERABAD	10
6	RAHUL	ATLANTA	4
7	RON	NORMAN	5
8	SAHU	HYDERABAD	3
9	SAM	7	8
10	SUJATA	OKLAHOMA	2

## OPTION 2:

Enter your option :

2

department number:

001

department data:

Data

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

2  
department number:

002

department data:

Science

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

2

department number:

003

department data:

Chemistry

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

2 department number:

004

department data:

Physics

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Updated Table:

t	department_num	department_data
1	1	Data
2	2	Science
3	3	Chemistry
4	4	Physics
5	5	Chemistry
6	6	Bio
7	7	dance
8	8	civil
9	9	mechanics
10	10	meta

## **OPTION 3:**

```
Enter your option :  
3  
process id:  
1001  
department number:  
1  
process_type (Cut, Paint, Fit) case-sensitive:  
Cut  
new info:  
cutting  
machine type:  
mechanical  
cutting type:  
straight  
painting method:  
manual  
paint type:  
wall paint  
process data:  
Connecting to the database...  
Dispatching the query...  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information  
    relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,  
    and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
    (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range  
    and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```

```
3
process id:
1002
department number:
2
process_type (Cut, Paint, Fit) case-sensitive:
Paint
new info:
painter
machine type:
civil
cutting type:
edge
painting method:
Automatic
paint type:
floor paint
process data:
Connecting to the database...
Dispatching the query...

Please select one of the options below:
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and information
     relevant to the type
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,
     and dateordered and associate it with one or more processes
(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 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 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
(15) Import: enter new customers from a data file until the file is empty
      (the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
      and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit
```

Enter your option :

```
3
process id:
1003
department number:
3
process_type (Cut, Paint, Fit) case-sensitive:
fit
new info:
fitting
machine type:
mechanical
cutting type:
edge
painting method:
automatic
paint type:
thickwallmounted
process data:
wall
Connecting to the database...
Dispatching the query...

Please select one of the options below:
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and information
     relevant to the type
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,
     and dateordered and associate it with one or more processes
(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 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 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
(15) Import: enter new customers from a data file until the file is empty
      (the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
      and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit
```

Enter your option :

```
Enter your option :
3
process id:
1004
department number:
4
process_type (Cut, Paint, Fit) case-sensitive:
cut
new info:
cutting
machine type:
civil
cutting type:
edge
painting method:
automatic
paint type:
semithick
process data:
floor
Connecting to the database...
Dispatching the query...

Please select one of the options below:
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and information
     relevant to the type
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,
     and dateordered and associate it with one or more processes
(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 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 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
(15) Import: enter new customers from a data file until the file is empty
      (the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
      and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit
```

```
process id:  
1005  
department number:  
5  
process_type (Cut, Paint, Fit) case-sensitive:  
paint  
new info:  
paiting  
machine type:  
mechanical  
cutting type:  
edge  
painting method:  
manual  
paint type:  
ticken  
process data:  
wall  
Connecting to the database...  
Dispatching the query...  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information  
relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,  
and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range  
and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```

Enter your option :

Updated Table:

	process_id	process_data
1	1001	paint
2	1002	paint
3	1003	wall
4	1004	floor
5	1005	wall
6	1007	thickened
7	1008	processed
8	1009	processed
9	1010	processing
1...	10006	data

	process_id	department_num
1	1001	1
2	1002	2
3	1003	3
4	1004	4
5	1005	5
6	10006	6
7	1007	7
8	1008	8
9	1009	9
1...	1010	10

	process_id	cutting_type	machine_type
1	1001	straight	mechanical
2	1004	edge	civil
3	1007	sharp	mechanical
4	1009	edge	mechanical
5	1010	edge	mechanical

	process_id	fit_type
1	1003	fitting
2	10006	fitting

	process_id	paint_type	painting_method
1	1002	floor	Automatic
2	1005	thickened	manual
3	1008	thin	automatic

## OPTION 4:

Enter your option :

4

assembly id param:

101

date ordered param (MM-dd-yyyy):

01-01-2001

customer name param:

ISHITA

assembly ordered param:

yes

process id param:

1001

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

4 assembly id param:

102

date ordered param (MM-dd-yyyy):

02-02-2002

customer name param:

MANAV

assembly ordered param:

no

process id param:

1002

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

4 assembly id param:

1003

date ordered param (MM-dd-yyyy):

03-03-2003

customer name param:

NARAYAN

assembly ordered param:

yes

process id param:

1003

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

4  
assembly id param:  
104  
date ordered param (MM-dd-yyyy):  
04-04-2004  
customer name param:  
PANKAJ  
assembly ordered param:  
yes  
process id param:  
1004  
Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

	assembly_id	date_ordered	assembly_details
1	101	2001-01-01	yes
2	102	2002-02-02	no
3	104	2004-04-04	yes
4	105	2005-05-05	yes
5	106	2006-06-06	no
6	107	2007-07-07	yes
7	108	2008-08-08	no
8	109	2008-08-08	yes
9	110	2010-10-10	yes
10	1003	2003-03-03	yes

	assembly_id	customer_name
1	101	ISHITA
2	102	MANAV
3	1003	NARAYAN
4	104	PANKAJ
5	105	PRIYA
6	106	RAHUL
7	107	RON
8	108	SAHU
9	109	SAM
10	110	SUJATA

	assembly_id	process_id
1	101	1001
2	102	1002
3	104	1004
4	105	1005
5	106	10006
6	107	1007
7	108	1007
8	109	1009
9	110	1010
10	1003	1003

## OPTION 5:

Enter your option :

```
5
account number:
10000001
date established:
01-20-2001
process id:
1001
cost detail 3:
34
cost detail 2:
234
cost detail 1:
762
dept data:
data
assembly details:
NA
account type (PROCESS, DEPARTMENT, ASSEMBLY) case-sensitive:
PROCESS
dept number:
1
assembly id:
1001
Connecting to the database...
```

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :  
5  
account number:  
**10000002**  
date established:  
**12-03-2004**  
process id:  
**1002**  
cost detail 3:  
**54**  
cost detail 2:  
**34**  
cost detail 1:  
**789**  
dept data:  
**Science**  
assembly details:  
**Good**  
account type (PROCESS, DEPARTMENT, ASSEMBLY) case-sensitive:  
**DEPARTMENT**  
dept number:  
**2**  
assembly id:  
**102**  
Connecting to the database...  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit

Enter your option :

Enter your option :  
5  
account number:  
**10000003**  
date established:  
**12-09-2009**  
process id:  
**1003**  
cost detail 3:  
**456**  
cost detail 2:  
**7654**  
cost detail 1:  
**876**  
dept data:  
**Chemistry**  
assembly details:  
**1003**  
account type (PROCESS, DEPARTMENT, ASSEMBLY) case-sensitive:  
**Department**  
dept number:  
**3**  
assembly id:  
**1003**  
Connecting to the database...  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit

Enter your option :

Enter your option :  
5  
account number:  
**10000004**  
date established:  
**11-08-2025**  
process id:  
**1004**  
cost detail 3:  
**8976**  
cost detail 2:  
**4532**  
cost detail 1:  
**6543**  
dept data:  
**Physics**  
assembly details:  
**YES**  
account type (PROCESS, DEPARTMENT, ASSEMBLY) case-sensitive:  
**ASSEMBLY**  
dept number:  
**4**  
assembly id:  
**104**  
Connecting to the database...  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit

Enter your option :

Results    Messages

	account_num ▼	date_account_established ▼
1	10000001	2001-01-20
2	10000002	2004-12-03
3	10000003	2009-12-09
4	10000004	2025-11-08
..	10000005	2021-12-09
6	10000006	2004-11-02
7	10000007	2008-09-08
8	10000008	2009-05-12
9	10000009	2009-02-13
1...	10000010	2009-07-09

Re

	account_num ▼	cost_details_3 ▼
1	10000001	34
2	10000005	543
3	10000006	567
4	10000008	432
5	10000009	452
6	10000010	654

	account_num	cost_details_2
1	10000002	34
2	10000003	7654

	department_num	account_num
1	2	10000002
2	3	10000003

	account_num	cost_details_1
1	10000004	6543
2	10000007	234

	assembly_id	account_num
1	104	10000004
2	107	10000007

## OPTION 6:

```
Enter your option :  
6  
job no:  
0001  
assembly id:  
101  
process id:  
1001  
job commenced date:  
2001-01-20  
job type (paint, fit, cut) case-sensitive:  
paint  
color:  
red  
volume:  
3 gallons  
machine type used:  
material used:  
wood  
Connecting to the database...
```

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

```
Enter your option :  
|
```

Enter your option :  
6  
job no:  
0002  
assembly id:  
102  
process id:  
1002  
job commenced date:  
2004-12-03  
job type (paint, fit, cut) case-sensitive:  
fit  
color:  
green  
volume:  
6  
machine type used:  
civil  
material used:  
metal  
Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

```
Enter your option :  
6  
job no:  
0003  
assembly id:  
1003  
process id:  
1003  
job commenced date:  
2009-12-09  
job type (paint, fit, cut) case-sensitive:  
cut  
color:  
brown  
volume:  
9  
machine type used:  
mechanical  
material used:  
wood  
Connecting to the database...
```

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :  
6  
job no:  
004  
assembly id:  
104  
process id:  
1004  
job commenced date:  
2025-11-08  
job type (paint, fit, cut) case-sensitive:  
cur  
color:  
pink  
volume:  
5  
machine type used:  
mechanical  
material used:  
wood  
Connecting to the database...

Please select one of the options below: |  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit

Enter your option :

	job_no	job_commenced_date	job_completed_date
1	1	2001-01-20	NULL
2	2	2004-12-03	NULL
3	3	2009-12-09	NULL
4	4	2025-11-08	NULL
5	5	2021-12-09	NULL
6	6	2004-11-02	NULL
7	7	2008-09-08	NULL
8	8	2009-05-12	NULL
9	9	2009-02-13	NULL
10	10	2009-07-09	NULL

	assembly_id	process_id	job_no
1	101	1001	1
2	102	1002	2
3	104	1004	4
4	105	1005	5
5	106	10006	6
6	107	1007	7
7	108	1008	8
8	109	1009	9
9	110	1010	10
10	1003	1003	3

Results grid

	job_no	machine_type_used	time_machine_used	material_used	labour_time
1	3	mechanical	NULL	w	NULL
2	5	civil	NULL	m	NULL
3	7	mechanical	NULL	w	NULL
4	8	mechanical	NULL	w	NULL
5	9	mechanical	NULL	m	NULL
6	10	metal	NULL	w	NULL

	job_no	color	volume	labour_time
1	1	red	3	NULL

	job_no	labour_time
1	2	NULL
2	6	NULL

## **OPTION 7:**

Enter your option :

7

job no:

1

job competition date:

2004-02-20

process type:

Machine

labour time:

2

time machine used:

3.32

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

7

job no:

2

job competition date:

2006-12-02

process type:

manual

labour time:

2.32

time machine used:

2

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

7

job no:

3

job competition date:

2010-02-09|

process type:

machine

labour time:

2.39

time machine used:

4.32

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

7

job no:

4

job competition date:

2027-11-14

process type:

manual

labour time:

3.32

time machine used:

1.23

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

	job_no	job_commenced_date	job_completed_date
1	1	2001-01-20	2004-02-20
2	2	2004-12-03	2006-12-02
3	3	2009-12-09	2010-02-09
4	4	2025-11-08	2027-11-14
5	5	2021-12-09	2022-12-08
6	6	2004-11-02	2009-11-02
7	7	2008-09-08	2012-09-08
8	8	2009-05-12	2010-02-13
9	9	2009-02-13	2010-02-12
10	10	2009-07-09	2022-08-09

	job_no	machine_type_used	time_machine_used	material_used	labour_time
1	3	mechanical	4	w	2
2	5	civil	3	m	1
3	7	mechanical	2	w	1
4	8	mechanical	5	w	3
5	9	mechanical	7	w	2
6	10	metal	2	w	2

	job_no	color	volume	labour_time
1	1	red	3	2

	job_no	labour_time
1	2	2
2	6	2

## OPTION 8:

```
Enter your option :
```

```
8
```

```
transaction number:
```

```
123
```

```
sup cost:
```

```
3245
```

```
account number:
```

```
10000001
```

```
Connecting to the database...
```

```
Please select one of the options below:
```

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

```
Enter your option :
```

Enter your option :  
8  
transaction number:  
345  
sup cost:  
2345  
account number:  
10000002  
Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :  
8  
transaction number:  
567  
sup cost:  
234  
account number:  
10000003  
Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

8  
transaction number:  
987  
sup cost:  
345  
account number:  
10000004

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

8 transaction number:

657

sup cost:

654

account number:

10000005

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

	transaction_num	sup_cost	account_num
1	123	3245	10000001
2	325	444	10000009
3	345	2345	10000002
4	432	654	10000006
5	456	876	10000010
6	543	234	10000007
7	567	234	10000003
8	657	654	10000005
9	987	345	10000004

	account_num	cost_details_3
1	10000001	3279
2	10000005	1197
3	10000006	1221
4	10000008	432
5	10000009	896
6	10000010	1530

	account_num	cost_details_2
1	10000002	2379
2	10000003	7888

Result

	account_num ▼	cost_details_1 ▼
1	10000004	6888
2	10000007	468

## OPTION 9:

Enter your option :

9

assembly id:

103

Connecting to the database...

Total cost: 0.0000

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

9

assembly id:

104

Connecting to the database...

Total cost: 345.0000

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

ENTER YOUR OPTION :

9 assembly id:

107

Connecting to the database...

Total cost: 234.0000

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

## OPTION 10:

```
Enter your option :  
10  
department number:  
2  
job completed date:  
2006-12-02  
Connecting to the database...  
Total cost: 2.0000  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information  
    relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,  
    and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
    ( the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range  
    and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```

```
Enter your option :  
10  
department number:  
5  
job completed date:  
2022-08-09  
Connecting to the database...  
Total cost: 0.0000  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information  
    relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,  
    and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
    ( the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range  
    and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```

```
Enter your option :  
10  
department number:  
7  
job completed date:  
2012-09-08  
Connecting to the database...  
Total cost: 1.0000  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information  
    relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,  
    and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
    (the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range  
    and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```

## **OPTION 11:**

Enter your option :

11

assembly id:

101

Connecting to the database...

Dispatching the query...

Done.

Process for assembly-id: 101, and its department number; Sorted by date commenced.

processID | deptNo

1001 | 1

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

11  
assembly id:

108

Connecting to the database...

Dispatching the query...

Done.

Process for assembly-id: 108, and its department number; Sorted by date commenced.

processID | deptNo

108 | 8

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Enter your option :

11

assembly id:

105

Connecting to the database...

Dispatching the query...

Done.

Process for assembly-id: 105, and its department number; Sorted by date commenced.

processID | deptNo

105 | 5

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

## **OPTION 12:**

```
Enter your option :
```

```
12
```

```
start category:
```

```
5
```

```
end category:
```

```
10
```

```
Connecting to the database...
```

```
Dispatching the query...|
```

```
Done.
```

```
Jobs from start date 5 completed on: 10
```

```
customer name
```

```
ISHITA
```

```
MANAV
```

```
NARAYAN
```

```
PRIYA
```

```
RON
```

```
SAM
```

```
Please select one of the options below:
```

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty

Enter your option :

12

start category:

8

end category:

10

Connecting to the database...

Dispatching the query...

Done.

Jobs from start date 8 completed on: 10

customer name

MANAV

PRIYA

SAM

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(      the user must be asked to enter the input file name).

Enter your option :

12  
start category:

2

end category:

4

Connecting to the database...

Dispatching the query...

Done.

Jobs from start date 2 completed on: 4

customer name

PANKAJ

RAHUL

SAHU

SUJATA

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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

## **OPTION 13:**

Enter your option :

13 job number start:

3 job number end:

6

Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

Results Messages

	job_no	machine_type_used	time_machine_used	material_used	labour_time
1	7	mechanical	2	w	1
2	8	mechanical	5	w	3
3	9	mechanical	7	m	2
4	10	metal	2	w	2

Enter your option :

13 job number start:

7 job number end:

8 Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

	job_no	machine_type_used	time_machine_used	material_used	labour_time
1	9	mechanical	7	m	2
2	10	metal	2	w	2

Enter your option :

13 job number start:

9 job number end:

10 Connecting to the database...

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty  
(the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

job_no	machine_type_u...	time_machine_u...	material_used	labour_time
--------	-------------------	-------------------	---------------	-------------

Results grid

## OPTION 14:

Before:

	job_no	color	volume	labour_time
1	1	red	3	2

Enter your option :

14  
job number:

1  
color:  
white

Connecting to the database...

1

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).
- (16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
- (17) Quit

Enter your option :

After:

	job_no	color	volume	labour_time
1	1	white	3	2

**OPTION 15:**

Enter your option :

15

Please enter the location and name of a CSV file with customer data:

>> ONLY INSERT CSV FILE <<

/Users/sujatasahu/Downloads/customer.csv

Dispatching the query...

Done. 4 rows inserted.

Please select one of the options below:

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department together with its type and information relevant to the type
- (4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes
- (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 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 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
- (15) Import: enter new customers from a data file until the file is empty

ll\_movies.jsp



add\_movie.jsp



add\_movie\_form.jsp



customer.csv

Users > sujatasahu > Downloads > customer.csv

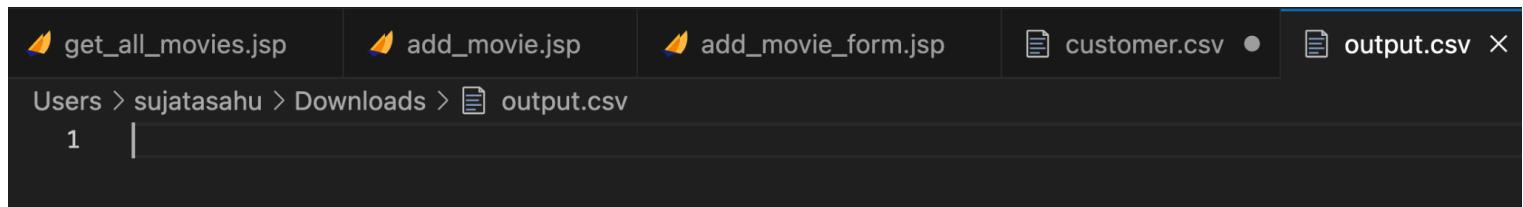
1	KRUSHNA, london, 1
2	JAGANNATH, California, 4
3	RAM, California, 3
4	VASUDEV, Atlanta, 6

Updated Table:

	customer_name	customer_address	category
1	Debashish	California	4
2	gulam	london	8
3	ISHITA	CALIFORNIA	Results grid
4	JAGANNATH	California	4
5	KRUSHNA	london	1
6	MANAV	NEWYORK	8
7	NARAYAN	DALLAS	7
8	PANKAJ	CALIFORHIA	4
9	Parul	Atlanta	6
10	PRIYA	HYDERABAD	10
11	Priyanjali	California	3
12	RAHUL	ATLANTA	4
13	RAM	California	3
14	RON	NORMAN	5
15	SAHU	HYDERABAD	3
16	SAM	7	8
17	SUJATA	OKLAHOMA	2
18	Suju	london	1
19	VASUDEV	Atlanta	6

## **OPTION 16:**

```
Enter your option :  
16  
Please enter MIN category number (integer from 1 - 10, inclusive):  
1  
Please enter MAX category number (integer from 1 - 10, inclusive):  
10  
Please enter the file output name:  
/Users/sujatasahu/Downloads/output.csv  
Dispatching the query...  
Done. File Location here:  
/Users/sujatasahu/Downloads/output.csv.csv  
  
Please select one of the options below:  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process-id and its department together with its type and information relevant to the type  
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes  
(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 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 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  
(15) Import: enter new customers from a data file until the file is empty  
( the user must be asked to enter the input file name).  
(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).  
(17) Quit
```



⚡ get\_all\_movies.jsp

⚡ add\_movie.jsp

⚡ add\_movie\_form.jsp

📄 customer.csv

📄 output.csv

Users > sujatasahu > Downloads > 📄 output.csv.csv

```
1 name,address,category
2 Debashish,California,4
3 gulam,london,8
4 ISHTA,CALIFORNIA,5
5 JAGANNATH,California,4
6 KRUSHNA,london,1
7 MANAV,NEWYORK,8
8 NARAYAN,DALLAS,7
9 PANKAJ,CALIFORHIA,4
10 Parul,Atlanta,6
11 PRIYA,HYDERABAD,10
12 Priyanjali,California,3
13 RAHUL,ATLANTA,4
14 RAM,California,3
15 RON,NORMAN,5
16 SAHU,HYDERABAD,3
17 SAM,7,8
18 SUJATA,OKLAHOMA,2
19 Suju,london,1
20 VASUDEV,Atlanta,6
21
```

## Option 17:

Enter your option :

17

Finished! Your work here is done.

## Screenshots showing 3 types of error

Enter your option :

```
1
name:
SUJATA
address:
chicago
category:
```

4

Connecting to the database...

```
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Customer_5B894
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServer
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatem
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecu
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(TDSTableOutputBuffer.java:7675)
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnec
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeUpdate(SQLServerStateme
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerState
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPrepar
        at indProject.main(indProject.java:101)
```

Enter your option :

```
14
job number:
red
Exception in thread "main" java.util.InputMismatchException
  at java.base/java.util.Scanner.throwFor(Scanner.java:939)
  at java.base/java.util.Scanner.next(Scanner.java:1594)
  at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
  at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
  at indProject.main(indProject.java:478)
```

Connecting to the database...

```
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Conversion failed when converting date and/or time from
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException)
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement)
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecu
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(TDByteBuffer.java:7675)
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConne
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatem
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeUpdate(SQLServerStat
  at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPrepa
  at indProject.main(indProject.java:296)
```

18:40:42

Started executing query at Line 1

Msg 2627, Level 14, State 1, Line 2

Violation of PRIMARY KEY constraint 'PK\_\_Customer\_\_5B894ACA81E7B247'. Cannot insert duplicate key in object 'dbo.Customer'. The duplicate key value is (SUJATA).

The statement has been terminated.

Total execution time: 00:00:00.085

18:41:28

Started executing query at Line 1

Msg 547, Level 16, State 0, Line 2

The INSERT statement conflicted with the CHECK constraint "CK\_\_Customer\_\_catego\_\_1E112859". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.Customer", column 'category'.

The statement has been terminated.

Total execution time: 00:00:00.06 Refresh

18:45:24

Started executing query at Line 1

Msg 547, Level 16, State 0, Line 2

The INSERT statement conflicted with the CHECK constraint "CK\_\_Customer\_\_catego\_\_1E112859". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.Customer", column 'category'.

The statement has been terminated.

Design

Total execution time: 00:00:00.092

# TASK 7

## Web based data Execution and its application

```
package jspazuretest;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DataHandle {
    private Connection conn;
    // Azure SQL connection credentials
    private String server = "sahu0001-sql-server.database.windows.net";
    private String database = "cs-dsa-4513-sql-db";
    private String username = "sahu0001";
    private String password = "Jaijagannath2023@";
    // Resulting connection string
    final private String url = String.format(
        "jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=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 movie_night table
    public ResultSet getAllCustomer() throws SQLException {
        getDBConnection();
        final String sqlQuery = "SELECT * FROM Customer";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }

    // Inserts a record into the movie_night table with the given attribute values
    public boolean addCustomer(String customer_name, String address, int category) throws SQLException {
        getDBConnection(); // Prepare the database connection
        // Prepare the SQL statement
        final String sqlQuery =
            "INSERT INTO Customer " +
            "(customer_name, customer_address, category) " +
            "VALUES " +
            "(?, ?, ?)";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        // Replace the '?' in the above statement with the given attribute values
        stmt.setString(1, customer_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;
    }
}
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Customer Data</title>
    </head>
    <body>
        <%@page import="jspazuretest.DataHandle"%>
        <%@page import="java.sql.ResultSet"%>
        <%
            // We instantiate the data handler here, and get all the movies from the database
            final DataHandle handler = new DataHandle();
            final ResultSet customers = handler.getAllCustomer();
        %>
        <!-- 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>address</h4>
                </td>
                <td align="center">
                    <h4>category</h4>
                </td>
            </tr>
        <%
            while(customers.next()) { // For each movie_night record returned...
                // Extract the attribute values for every row returned
                final String customer_name = customers.getString("customer_name");
                final String address = customers.getString("customer_address");
                final int category = customers.getInt("category");

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

```
1<!DOCTYPE html>
2<html>
3    <head>
4        <meta charset="UTF-8">
5        <title>Add Customers</title>
6    </head>
7    <body>
8        <h2>Add Details</h2>
9        <!--
10           Form for collecting user input for the new movie_night record
11           Upon form submission, add_movie.jsp file will be invoked
12       -->
13        <form action="add_movie.jsp">
14            <!-- The form organized in an HTML table for better clarity
15            <table border=1>
16                <tr>
17                    <th colspan="2">Enter the customer name:</th>
18                </tr>
19                <tr>
20                    <td>Customer Name:</td>
21                    <td><div style="text-align: center;">
22                        <input type="text" name="customer_name">
23                    </div></td>
24                </tr>
25                <tr>
26                    <td>Customer Address:</td>
27                    <td><div style="text-align: center;">
28                        <input type="text" name="address">
29                    </div></td>
30                </tr>
31                <tr>
32                    <td>Customer Category:</td>
33                    <td><div style="text-align: center;">
34                        <input type="text" name="category">
35                    </div></td>
36                </tr>
37                <tr>
38                    <td><div style="text-align: center;">
39                        <input type="reset" value="Clear">
40                    </div></td>
41                    <td><div style="text-align: center;">
42                        <input type="submit" value="Insert">
43                    </div></td>
44                </tr>
45            </table>
46        </form>
47    </body>
48 </html>
```

```

<%@ 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="jspazuretest.DataHandle"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandle handler = new DataHandle();

// Get the attribute values passed from the input form.
String customer_name = request.getParameter("customer_name");
String address = request.getParameter("address");
String category = request.getParameter("category");

/*
 * If the user hasn't filled out all the time, movie name and duration. This is very simple checking.
 */
if (customer_name.equals("") || address.equals("") || category.equals ("")) {
    response.sendRedirect("add_movie_form.jsp");
} else {
    int categoryno = Integer.parseInt(category);

    // Now perform the query with the data from the form.
    boolean success = handler.addCustomer(customer_name, address, categoryno);
    if (!success) { // Something went wrong
        %>
            <h2>There was a problem inserting the course</h2>
    } else { // Confirm success to the user
        %>
            <h2>Customer Database:</h2>

            <ul>
                <li>Start Time: <%=customer_name%></li>
                <li>Movie Name: <%=address%></li>
                <li>Duration: <%=category%></li>
            </ul>

            <h2>Was successfully inserted.</h2>
            <a href="get_all_movies.jsp">See all movie nights.</a>
    }
}
%>
</body>
</html>

```

## Add Details

Enter the customer name:	
Customer Name:	<input type="text"/>
Customer Address:	<input type="text"/>
Customer Category:	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

## Add Details

Enter the customer name:	
Customer Name:	Amour
Customer Address:	TEXAS
Customer Category:	7
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

## Customer Database:

- Start Time: Amour
- Movie Name: TEXAS
- Duration: 7

**Was successfully inserted.**

[Get all customers.](#)

customer_name	address	category
amma	oklahoma	3
Amour	TEXAS	7
Debashish	California	4
gulam	london	8
ISHITA	CALIFORNIA	5
JAGANNATH	California	4
KRUSHNA	london	1
MANAV	NEWYORK	8
NARAYAN	DALLAS	7
PANKAJ	CALIFORHIA	4
Parul	Atlanta	6
PRIYA	HYDERABAD	10
Priyanjali	California	3
RAHUL	ATLANTA	4
RAM	California	3
RON	NORMAN	5
SAHU	HYDERABAD	3
SAM	7	8
Shivangi	West B	7
SUJATA	OKLAHOMA	2
Suju	london	1
VASUDEV	Atlanta	6

# Retrieve customers

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

/*
* If the user hasn't filled out all the category numbers. This is very simple
checking.
*/
if (from.equals("") || to.equals("")) {
    response.sendRedirect("Rcustomers.jsp");
} else {
    int min_category = Integer.parseInt(from);
    int max_category = Integer.parseInt(to);
    // Now perform the query with the data from the form.
    final ResultSet Customer = handler.fetchCustomers(min_category, max_category);
%>
<!-- The table for displaying all the movie records -->
<table cellspacing="2" cellpadding="2" border="1">
<tr>
    <!-- The table headers row -->
    <td align="center">
        <h4>Name</h4>
    </td>
    <td align="center">
        <h4>Address</h4>
    </td>
    <td align="center">
        <h4>Category</h4>
    </td>
</tr>
<%
while (Customer.next()) { // For each customer record returned...
    // Extract the attribute values for every row returned
    String name = Customer.getString("customer_name");
    String address = Customer.getString("customer_address");
    String category = Customer.getString("category");
    out.println("<tr>"); // Start printing out the new table row
    out.println( // Print each attribute value
        "<td align=\"center\">" + name + "<td align=\"center\">" + address + "<td align=\"center\">" + category
        + "</td>");
    out.println("</tr>");
}
}
```

```

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

```

## Input Range to get the customers

Enter Range to fetch Customers	
Category From:	<input type="text"/>
Category To:	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="search"/>

## Input Range to get the customers

Enter Range to fetch Customers	
Category From:	<input type="text" value="4"/>
Category To:	<input type="text" value="7"/>
<input type="button" value="Clear"/>	<input type="button" value="search"/>

Name	Address	Category
Amour	TEXAS	7
Debashish	California	4
ISHITA	CALIFORNIA	5
JAGANNATH	California	4
NARAYAN	DALLAS	7
PANKAJ	CALIFORHIA	4
Parul	Atlanta	6
RAHUL	ATLANTA	4
RON	NORMAN	5
Shivangi	West B	7
VASUDEV	Atlanta	6