Course name: Database Management System

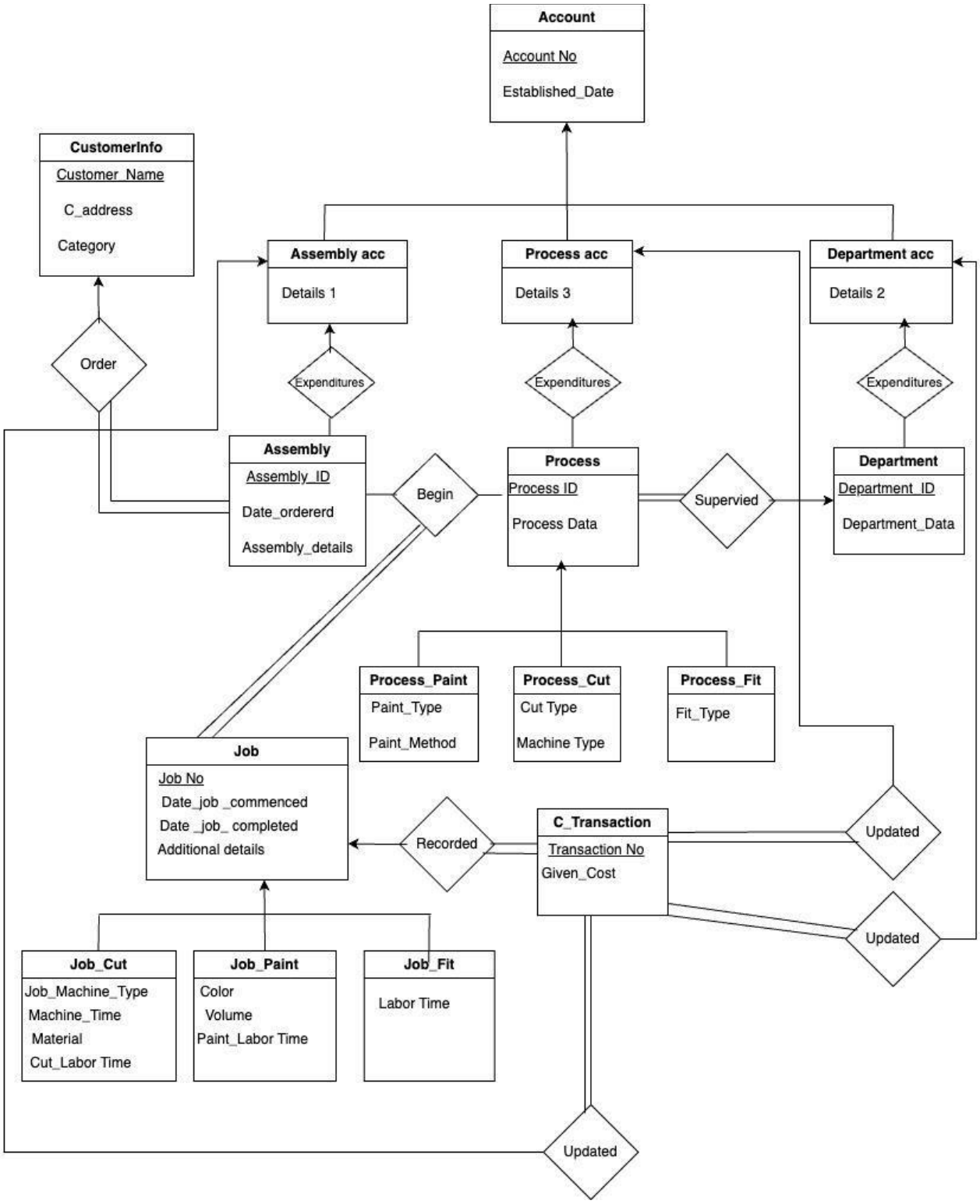
Instructor's name: Dr. Le Gruenwald

Author's name: Leama Nolvenne Tah

Title of the project: JOB-SHOP ACCOUNTING DATABASE SYSTEM

Tasks Performed **Page Number**

Task 1. : Design an ER diagram to represent the Job-Shop Accounting database defined in part I.



Task 2. Convert the ER diagram in Task 1 to a Relational Database (i.e. a set of relational schemas).

customerInfo (Customer_Name, C_address, Categrory)

Assembly (Assembly ID, Date_ordered, Assembly_details, Customer Name)

Department (Department ID, Department Data)

Process1 (Process_ID, Process_Data, Department_ID)

Process_Fit (Process_ID, Fit_Type)

Process_Paint (Process_ID, Paint_Type, Paint_Method)

Process_Cut (Process_ID, Cut_type, Machine_type)

Job (Job_No, Date_job_commenced, Date_job_completed, Assembly_ID, Process_ID)

Job_Fit (Job_No, Labor_Time)

Job_Paint (Job_No, Color, Volume, Paint_Labor_Time)

Job_Cut (Job_No, Job_Machine_Type, Machine_Time, Material, Cut_Labor_Time)

Department_acc1 (Account_No, Established_Date, Date_established, details_2, Department_ID)

Assembly1_acc (Account_No, Established_Date ,Date_established, details_1, Assembly_ID)

Process1_acc (Account_No, Established_Date, Date_established, details_3, Process_ID)

Transaction (Transaction_No, Given Cost, Job_No, Details 1, Details 2, Details 3)

Task 3.

3.1. Discuss choices of appropriate storage structures for each relational table assuming that all types of storage structures discussed in class (Lecture Topic 4) are available.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization	Justifications
CustomerInfo	#1Insertion		30 / day	B+ Tree with Index = Category	B+Tree is used to find all records with search key as Category in a specified range [lower bound, Upper Bound].
	#13 Range Search	Category	100 / day		
Assembly	#3 Insertion		40 / day	Heap File	Heap File is used to insert details without imposing any order.
Process1	#4 Insertion		Infrequent	Dynamic Extendable Hashing. Hash key (Process_ID)	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#8 Random Search	Process_ID / Department_ID	50 / day		
	#10 Random Search	Department_ID	20 / day		
	#11 Random Search	Proces_ID	100 / day		
	#12 Random Search	Proces_ID	20 / day		
Process_Fit	#4 insertion		Infrequent	Heap File	Heap File is used to insert details without imposing any order.

Process_Paint	#4 Insertion		Infrequent	Heap File	Heap File is used to insert details without imposing any order.
Process_Cut	#4 Insertion		Infrequent	Heap File	Heap File is used to insert details without imposing any order.

Department	#2 Insertion	Infrequent	Heap File		
Job	#6 Insertion		50 / day	Dynamic Extendable Hashing with hash key (Job_No).	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#7 Random Search	Job_No	50 / day		
	#8 Random Search	Job_No	50 / day		
	#10 Random search	Process_ID / Date_completed	20 /day		
	#11 Random Search	Assembly_ID	100 / day		
	#12 Random Search	Date_job_completed	20 / day		
	#14 Random Search	Job_No	1 / month		
Job_fit	#7 Insert		50 / day	Dynamic Extendable Hashing with hash key (Job_No).	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#10 Random Search	Job_No	20 /day		
	#12 Random Search	Job_No	20 /day		
Job_Paint	#7 Insertion		50 / day	Dynamic Extendable	Dynamic hashing is used because it

	#10 Random Search	Job_No	20 / day	Hashing with hash key (Job_No).	managed storage file and it is efficient for random search to retrieve data.
	#12 Random Search	Job_No	20 / day		
	#15 Random Search	Job_No	1 / week		
Job_Cut	#7 Insertion		50 / day	B+ Tree with Index = Job_No	B+Tree is used to find all records with search key as Category in a
	#10	Job_No	20 /day		
	Random Search				specified range [lower bound, Upper Bound].
	#12 Random Search	Job_No	20 /day		
	#14 Range Search	Job_No	1 / month		
Department_acc	#5 Insertion		10 / day	Dynamic Extendable Hashing with hash key (Account_No) .	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#8 Random Search	Department_ID	50 / day		
	#8 random Search (update)	Account_No	50 / day		
Assembly_accc	#5 Insertion		10 / day	Dynamic Extendable Hashing with hash key (Account_No) .	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#8 Random Search	Assembly_ID	50 / day		
	#8 Random Search (update)	Account_No	50 / day		
	#9 Random Search	Assembly_ID	200 / day		

Process_acc	#5 Insertion		10 / day	Dynamic Extendable Hashing with hash key (Account_No)	Dynamic hashing is used because it managed storage file and it is efficient for random search to retrieve data.
	#8 Random Search	Process_ID	50 / day		
	#8 Random Search (update)	Account_No	50 / day		
C_Transaction	#8 Insertion	50 / day	50 / day	Heap File	Heap File is used to insert details without imposing any order.

3.2. Discuss choices of storage structures for each relational table when implementing it in Azure SQL Database (if different from the previous choices specified in 3.1).

A relational database's implementation in Azure SQL Database to store each table is determined by several factors such as the nature of the data, access patterns, and performance considerations.
I do not change any of the search key.

Table	**Index Key**
customerInfo	Category
Process1	Department_ID
Job1	Assembly_ID
Department_acc1	Department_ID
Assembly1_acc	Assembly_ID
Process1_acc	Process_ID

Task 4. Construct SQL statements to create tables and implement them on Azure SQL Database.

```
-- Create Customer table
CREATE TABLE customerInfo (
   Customer_Name VARCHAR(20),
   C_address VARCHAR(100),
   Category INT,
   CONSTRAINT Category CHECK (Category BETWEEN 1 AND 10),
   PRIMARY KEY (customer_Name)
);
SELECT*FROM customerInfo;
```

Customer_Name	C_address	Category
No results		

```
-- Create Assembly table
CREATE TABLE Assembly1 (
   Assembly_ID VARCHAR(10),
   Date_ordered DATE,
   Assembly_details VARCHAR(100),
   Customer_name VARCHAR(20),
   PRIMARY KEY (Assembly_ID),
   FOREIGN KEY (Customer_name) REFERENCES CustomerInfo(Customer_name)
);
SELECT*FROM Assembly1;

--Create Department Table
```

```
CREATE TABLE Department (
    Department_ID VARCHAR(10),
    Department_Data VARCHAR(100),
    PRIMARY KEY (Department_ID)
);
SELECT*FROM Department;

--Create Process Table
CREATE TABLE Process1 (
    Process_ID VARCHAR(10) PRIMARY KEY,
    Process_Data VARCHAR(100),
    Department_ID VARCHAR(10) FOREIGN KEY REFERENCES Department(Department_ID)
);
SELECT*FROM Process1;
```

Process_ID	Process_Data	Department_ID
No results		

```
--Create Process_Fit Table
CREATE TABLE Process_Fit (
    Process_ID VARCHAR(10) FOREIGN KEY REFERENCES Process1(Process_ID),
    Fit_Type VARCHAR(10),
    PRIMARY KEY (Process_ID)
);
SELECT*FROM Process_Fit;
```

Process_ID	Fit_Type
No results	

```
--Create Process_Paint Table
CREATE TABLE Process_Paint (
    Process_ID VARCHAR(10) FOREIGN KEY REFERENCES Process1(Process_ID),
    Paint_Type VARCHAR(8),
    Paint_Method VARCHAR(8),
    PRIMARY KEY (Process_ID)
);
SELECT*FROM Process_Paint;
```

Process_ID	Paint_Type	Paint_Method
No results		

```
--Create Process_Cut Table
CREATE TABLE Process_Cut (
    Process_ID VARCHAR(10) FOREIGN KEY REFERENCES Process1(Process_ID),
    Cut_Type VARCHAR(8),
    Machine_type VARCHAR(10),
    PRIMARY KEY (Process_ID)
);
SELECT*FROM Process_Cut;
```

Process_ID	Cut_Type	Machine_type
No results		

```
--Create Job Table
CREATE TABLE Job1 (
   Job_No INT PRIMARY KEY,
   Date_job_commenced DATE,
   Date_job_completed DATE,
   Assembly_ID VARCHAR(10) FOREIGN KEY REFERENCES Assembly1(Assembly_ID),
Process_ID VARCHAR(10) FOREIGN key REFERENCES Process1(Process_ID)
);
SELECT*FROM Job1;
```

Job_No	Date_job_commenced	Date_job_completed	Assembly_ID	Process_ID
No results				

```
--Create Job_Fit Table
CREATE TABLE Job_Fit (
   Job_No INT FOREIGN KEY REFERENCES Job1(Job_No),
   Labor_Time TIME,
   PRIMARY KEY (Job_No)
);
SELECT*FROM Job_Fit;
```

Job_No	Labor_Time
No results	

```
--Create Job_Paint Table
CREATE TABLE Job_Paint (
```

```
    Job_No INT FOREIGN KEY REFERENCES Job1(Job_No),
    Color VARCHAR(10),
    Volume INT,
    Paint_Labor_Time TIME,
    PRIMARY KEY (Job_No)
);
SELECT*FROM Job_Paint;
```

Job_No	Color	Volume	Paint_Labor_Time
No results			

```
--Create Job_Cut Table
CREATE TABLE Job_Cut (
    Job_No INT FOREIGN KEY REFERENCES Job1(Job_No),
    Job_Machine_Type VARCHAR(10),
    Machine_Time TIME,
    Material VARCHAR(6),
    Cut_Labor_Time TIME,
    PRIMARY KEY (Job_No)
);
SELECT*FROM Job_Cut;
```

Job_No	Job_Machine_Type	Machine_Time	Material	Cut_Labor_Time
No results				

```
--Create Department Account Table
CREATE TABLE Department_acc1(
```

```
    Account_No INT,
Established_Date DATE,     details_2
FLOAT,
    Department_ID VARCHAR(10) FOREIGN KEY REFERENCES Department(Department_ID) UNIQUE, PRIMARY
KEY (Account_No)
);
SELECT*FROM Department_acc1;
```


```
--Create Assembly Account
CREATE TABLE Assembly1_acc (
    Account_No INT,
    Established_Date DATE,
    Details_1 FLOAT,
    Assembly_ID VARCHAR(10) FOREIGN KEY REFERENCES Assembly1(Assembly_ID) UNIQUE,
PRIMARY KEY (Account_No)
);
SELECT*FROM Assembly1_acc;
```


```
--Create Process Account
CREATE TABLE Process1_acc (
```

```
    Account_No INT,
    Established_Date DATE,
    Details_3 FLOAT,
    Process_ID VARCHAR(10) FOREIGN KEY REFERENCES Process1(Process_ID) UNIQUE,
PRIMARY KEY (Account_No)
);
```


```
--Create Customer Transaction Table
CREATE TABLE C_Transaction (
    Transaction_No VARCHAR(10) PRIMARY KEY,
    Given_Cost FLOAT,
    Job_No INT FOREIGN KEY REFERENCES Job1(Job_No),
    Department_acc INT FOREIGN KEY REFERENCES Department_acc(Dept_Account_No),
    Assembly1_acc INT FOREIGN KEY REFERENCES Assembly1_acc(Assembly_Account_No),
    Process1_acc INT FOREIGN KEY REFERENCES Process1_acc(Process_Account_No)
);
SELECT*FROM C_Transaction;
```


```
/*Create Index
*/
```

```
-- Create index on Phone Number - customer table

CREATE INDEX customer_Category ON CustomerInfo(Category);


-- Create index on Department_No - Process table

CREATE INDEX process_department ON Process1(Department_ID);


-- Create index on Assembly_ID - Job table

CREATE INDEX Assembly_Job ON job1(Assembly_ID);


-- Create index on Department_No - Department_acc table

CREATE INDEX Department_No_Acc ON Department_acc(Department_ID);


-- Create index on Assembly_ID - Assembly_acc table

CREATE INDEX Assembly_ID_acc ON Assembly1_acc(Assembly_ID);


-- Create index on process_ID - Process_acc table

CREATE INDEX Process_ID_acc ON Process1_acc(Process_ID);
```

Task 5a. (Task 5 and Task 6 together): Write SQL statements for all queries (1-14) defined in part I.

SQL code to execute all queries

```
--CREATE PROCEDURES

-- Enter a new customer
CREATE PROCEDURE InsertCustomerInfo
   @Customer_Name VARCHAR(20),
   @C_address VARCHAR(100),
   @Category INT
AS
BEGIN

INSERT INTO customerInfo
(Customer_Name, C_address, Category)
VALUES (@Customer_Name, @C_address, @Category)
END
GO
-- Enter a new Department
CREATE PROCEDURE InsertDepartmentInfo
   @Department_ID VARCHAR(10),
   @Department_Data VARCHAR(100)
AS
BEGIN
   INSERT INTO Department
   (Department_ID, Department_Data)
   VALUES (@Department_ID, @Department_Data)
```

```
END
GO
--Enter a new assembly with its customer-name, assembly-details, assembly-id,
-- and date-ordered
CREATE PROCEDURE InsertAssemblyInfo
   @Assembly_ID VARCHAR(10),
   @Date_ordered VARCHAR(10),
   @Assembly_details VARCHAR(100),
   @Customer_Name VARCHAR(20)
AS
BEGIN
   INSERT INTO Assembly1
   (
   Assembly_ID,Date_ordered, Assembly_details,Customer_Name)
   VALUES (@Assembly_ID, CAST (@Date_ordered as DATE), @Assembly_details, @Customer_Name)
END
GO

--Enter a new process-id and its department together
--with its type and information relevant to the type
CREATE PROCEDURE InsertProcessIDdept
   @Process_ID VARCHAR(10),
   @Process_Data VARCHAR(100),
   @Department_ID VARCHAR(10)
AS
BEGIN
   INSERT INTO Process1
   (Process_ID, Process_Data, Department_ID)
```

```
    VALUES (@Process_ID, @Process_Data, @Department_ID)
END
GO

CREATE PROCEDURE InsertProcessFit
    @Process_ID VARCHAR(10),
    @Fit_Type VARCHAR(10)
AS
BEGIN
    INSERT INTO Process_Fit
    (Process_ID, Fit_Type)
    VALUES (@Process_ID, @Fit_Type)
END

GO
CREATE PROCEDURE InsertProcessPaint
    @Process_ID VARCHAR(10),
    @Paint_Type VARCHAR(8),
    @Paint_Method VARCHAR(8)
AS
BEGIN
    INSERT INTO Process_Paint
    (Process_ID, Paint_Type, Paint_Method)
    VALUES (@Process_ID, @Paint_Type, @Paint_Method)
END
GO

CREATE PROCEDURE InsertProcessCut
```

```
    @Process_ID VARCHAR(10),
    @Cut_Type VARCHAR(8),
    @Machine_Type VARCHAR(10)
AS
BEGIN
    INSERT INTO Process_Cut
    (Process_ID, Cut_Type, Machine_Type)
    VALUES (@Process_ID, @Cut_Type, @Machine_Type)
END

GO
-- Create a new account and associate it with the
-- process, assembly, or department to which it is applicable
CREATE PROCEDURE InsertAccountDept
    @Account_No INT,
    @Details_2 FLOAT,
    @Department_ID VARCHAR(10)
AS
BEGIN
    INSERT INTO Department_acc
    (Account_No, Details_2, Department_ID)
    VALUES (@Account_No, @Details_2, @Department_ID)
END

GO
CREATE PROCEDURE InsertAccountAssembly
    @Account_No INT,
    @Details_1 FLOAT,
```

```
    @Assembly_ID VARCHAR(10)
AS
BEGIN
    INSERT INTO Assembly1_acc
    (Account_No,Details_1, Assembly_ID)
    VALUES (@Account_No, @Details_1, @Assembly_ID)
END

GO
CREATE PROCEDURE InsertAccountProcess
    @Account_No INT,
    @Details_3 FLOAT,
    @Process_ID VARCHAR(10)
AS
BEGIN
    INSERT INTO Process1_acc
    (Account_No, Details_3, Process_ID)
    VALUES (@Account_No, @Details_3, @Process_ID)
END

GO
-- Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
CREATE PROCEDURE InsertJobInfo
    @Job_No INT,
    @Date_job_commenced VARCHAR(10),
    @Assembly_ID VARCHAR(10),
    @Process_ID VARCHAR(7)
AS
```

```
BEGIN
    INSERT INTO Job
    (Job_No, Date_job_commenced, Assembly_ID, Process_ID)
    VALUES (@Job_No, CAST(@Date_job_commenced as DATE), @Assembly_ID, @Process_ID)
END
GO

-- At the completion of a job, enter the date it completed
--and the information relevant to the type of job
CREATE PROCEDURE UpdateJobInfo
    @Job_No INT,
    @Date_job_completed VARCHAR(10)

AS
BEGIN
    UPDATE Job SET Date_job_completed = @Date_job_completed WHERE Job_No = @Job_No
END
GO

CREATE PROCEDURE InsertJobFitInfo
    @Job_No INT,
    @Labor_Time VARCHAR(10)
AS
BEGIN
    INSERT INTO Job_Fit
    (Job_No, Labor_Time)
    VALUES (@Job_No, CAST(@Labor_Time as TIME))
END
```

```
GO

CREATE PROCEDURE InsertJobPaintInfo
   @Job_No INT,
   @Color VARCHAR(10),
   @Volume INT,
   @Paint_Labor_Time VARCHAR(10)
AS
BEGIN
   INSERT INTO Job_Paint
   (Job_No, Color, Volume, Paint_Labor_Time)
   VALUES (@Job_No, @Color, @Volume, CAST(@Paint_Labor_Time as TIME))
END
GO

CREATE PROCEDURE InsertJobCutInfo
   @Job_No INT,
   @Job_Machine_Type VARCHAR(10),
   @Machine_Time VARCHAR(10),
   @Material VARCHAR(6),
   @Cut_Labor_Time VARCHAR(10)
AS
BEGIN
   INSERT INTO Job_cut
   (Job_No, Job_Machine_Type, Machine_Time, Material, Cut_Labor_Time)
   VALUES (@Job_No, @Job_Machine_Type, CAST(@Machine_Time as TIME), @Material,
CAST(@Cut_Labor_Time as TIME))
END
```

```
GO

-- Enter a transaction-no and its Given-cost and update all the costs (details) of the affected
--accounts by adding sup-cost to their current values of details
CREATE PROCEDURE InsertTransactionInfo
   @Transaction_No VARCHAR(10),
   @Given_Cost FLOAT,
   @Job_No INT
AS
BEGIN
   DECLARE @Dept_acc_no INT,
         @Assembly_acc_no INT,
         @Process_acc_no INT;

   SET @Dept_acc_no = (SELECT Account_No
               FROM Department_acc, Process1, Job1
               WHERE Process1.Process_ID = Job1.Process_ID AND Job1.Job_No = @Job_No AND
Department_acc.Department_ID = Process1.Department_ID);

   SET @Assembly_acc_no = (SELECT Account_No
               FROM Assembly1_acc, Job1
               WHERE Assembly1_acc.Assembly_ID = Job1.Assembly_ID AND Job1.Job_No = @Job_No);

   SET @Process_acc_no = (SELECT Account_No
               FROM Process1_acc, Job1
               WHERE Process1_acc.Process_ID = Job1.Process_ID AND Job1.Job_No = @Job_No);

   INSERT INTO C_Transaction
   (Transaction_No, Given_Cost, Job_No, Department_No, Assembly_acc, Process1_acc)
```

```
    VALUES (@Transaction_No, @Given_Cost, @Job_No, @Dept_acc_no, @Assembly_acc_no, @Process_acc_no);

    UPDATE Department_acc
    SET Details_2 = Details_2 + @Given_Cost
    WHERE Account_No = @Dept_acc_no;

    UPDATE Assembly1_acc
    SET Details_1 = Details_1 + @Given_Cost
    WHERE Account_No = @Assembly_acc_no;

    UPDATE Process1_acc
    SET Details_3 = Details_3 + @Given_Cost
    WHERE Account_No = @Process_acc_no;
END
GO

--Retrieve the cost incurred on an assembly_ID
CREATE PROCEDURE RetriveCost
    @Assembly_ID VARCHAR(10)

AS
BEGIN
    SELECT Details_1 FROM Assembly1_acc WHERE Assembly1_acc.Assembly_ID = @Assembly_ID; END

GO
-- Retrieve the total labor time within a department for jobs completed
-- in the department during a given date
CREATE PROCEDURE RetrieveLaboreTime
    @Department_ID VARCHAR(10),
```

```
    @Date_job_completed VARCHAR(10)

AS
BEGIN
    DECLARE @Fit_Labor_Time FLOAT,
            @Paint_Labor_Time FLOAT,
            @Cut_Labor_Time FLOAT,
            @Total_Labor_Time FLOAT;
    SET @Fit_Labor_Time = (SELECT  SUM(( DATEPART(hh, Labor_Time) * 3600 ) + ( DATEPART(mi,Labor_Time)
* 60 ) + DATEPART(ss,Labor_time))/60 as minute
FROM Job_Fit WHERE Job_Fit.Job_No in (
SELECT distinct(Job_No) FROM Job1
WHERE Job1.Process_ID in (SELECT distinct(Process_ID) FROM Process1 WHERE Process1.Department_ID =
@Department_ID) AND Job1.Date_job_completed = CAST(@Date_job_completed as DATE)));

IF @Fit_Labor_Time IS NULL SET @Fit_Labor_Time = 0;

    SET @Paint_Labor_Time = (SELECT  SUM(( DATEPART(hh, Paint_Labor_Time) * 3600 ) + ( DATEPART(mi,
Paint_Labor_Time) * 60 ) + DATEPART(ss,  Paint_Labor_Time))/60 as minute
FROM Job_Paint WHERE Job_Paint.Job_No in (
SELECT distinct(Job_No) FROM Job1
WHERE Job1.Process_ID in (SELECT distinct(Process_ID) FROM Process1 WHERE Process1.Department_ID =
@Department_ID) AND Job1.Date_job_completed =CAST(@Date_job_completed as DATE)));

IF @Paint_Labor_Time IS NULL SET @Paint_Labor_Time = 0;

    SET @Cut_Labor_Time = (SELECT  SUM(( DATEPART(hh, Cut_Labor_Time) * 3600 ) + ( DATEPART(mi,
Cut_Labor_Time) * 60 ) + DATEPART(ss, Cut_Labor_Time))/60 as minute
FROM Job_Cut WHERE Job_Cut.Job_No in (
```

```
SELECT distinct(Job_No) FROM Job1
WHERE Job1.Process_ID in (SELECT distinct(Process_ID) FROM Process1 WHERE Process1.Department_ID =
@Department_ID) AND Job1.Date_job_completed =CAST(@Date_job_completed as DATE)));

IF @Cut_Labor_Time IS NULL SET @Cut_Labor_Time = 0;

   SET @Total_Labor_Time = @Fit_Labor_Time + @Paint_Labor_Time + @Cut_Labor_Time
   SELECT  @Total_Labor_Time
END
GO

-- Retrieve the processes through which a given assembly-id has passed so far (in date_job-commenced order) and the
department responsible for each process
CREATE PROCEDURE RetrieveProcessAssembly
   @Assembly_ID VARCHAR(10)
AS
BEGIN
   SELECT Job1.Date_job_commenced, Job1.Process_ID, Process1.Department_ID
   FROM Job1, Process1
   WHERE Job1.Assembly_ID = @Assembly_ID AND Process1.Process_ID = Job1.Process_ID
ORDER BY 1 ;

END

GO
-- Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given
department
CREATE PROCEDURE RetrieveJobAssemblyFit
```

```
    @Date_job_completed VARCHAR(10),
    @Department_ID VARCHAR(10)
AS
BEGIN
    SELECT DISTINCT(Job1.Job_No), Job1.Assembly_ID, Job_Fit.Labor_Time
    FROM Job1, Job_Fit
    WHERE Date_job_completed = @Date_job_completed and Job1.Process_ID in (SELECT Process1.Process_ID
FROM Process1 WHERE Department_ID = @Department_ID) AND Job_Fit.Job_No = Job1.Job_No;

END
GO

CREATE PROCEDURE RetrieveJobAssemblyPaint
    @Date_job_completed VARCHAR(10),
    @Department_ID VARCHAR(10)
AS
BEGIN
    SELECT DISTINCT(Job1.Job_No), Job1.Assembly_ID, Job_Paint.Color, Job_Paint.Volume,
Job_Paint.Paint_Labor_Time
    FROM Job1, Job_Paint
    WHERE Date_job_completed = @Date_job_completed and Job1.Process_ID in (SELECT process_ID FROM
Process1 WHERE Department_ID = @Department_ID) AND Job_Paint.Job_No = Job1.Job_No;

END
GO
CREATE PROCEDURE RetrieveJobAssemblyCut
    @Date_job_completed VARCHAR(10),
    @Department_ID VARCHAR(10)
```

```
AS
BEGIN
   SELECT DISTINCT(Job1.Job_No), Job1.Assembly_ID, Job_Cut.Job_Machine_Type, Job_Cut.Machine_Time,
Job_Cut.Material, Job_Cut.Cut_Labor_Time
   FROM Job1, Job_Cut
   WHERE Date_job_completed = @Date_job_completed and Job1.Process_ID in (SELECT Process1.Process_ID
FROM Process1 WHERE Department_ID = @Department_ID) AND Job_Cut.Job_No = Job1.Job_No;

END

GO
-- Retrieve the customers (in name order) whose category is in a given range
CREATE PROCEDURE RetrieveCustomerByCategory
   @Lower_b INT,
   @Upper_b INT
AS
BEGIN
   SELECT Customer_Name, Category AS Name FROM customerInfo
   WHERE Category >= @Lower_b AND Category <= @Upper_b
   ORDER BY 1 ;
END

GO
-- Delete all cut-jobs whose job-no is in a given range
CREATE PROCEDURE DeleteCutJob
   @Lower_b INT,
   @Upper_b INT
AS
```

```
BEGIN
   DELETE FROM Job_Cut WHERE Job_No >= @Lower_b AND Job_No <= @Upper_b
END

GO
-- Change the color of a given paint
CREATE PROCEDURE ChangePaintColor
   @Job_No INT,
   @Color VARCHAR(10)
AS
BEGIN
   UPDATE Job_Paint SET Color = @Color WHERE Job_No = @Job_No;
END
GO

CREATE PROCEDURE DeleteJobCutIdentified
   --@lower_b INT,
  -- @upper_b INT
AS
BEGIN
   DECLARE @JB INT

   SET @JB = (SELECT DISTINCT( Job_Cut.Job_No ) FROM Job_Cut WHERE Job_No >= 1 AND Job_No <= 10 )
   --DELETE FROM Job_cut WHERE job_no >= @lower_b AND job_no <= @upper_b
   SELECT @JB;
END
```

Task 5b. Write a Java application program that uses JDBC and Azure SQL Database to implement all SQL queries (options 1-14),

1. Enter a new customer (30/day).

```
try (// initializing Scanner object
Scanner myScan = new Scanner(System.in)) {
    //reading the input given by the user
    Choice = myScan.nextInt();

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

    switch (Choice) {

        case 1:
            // try and catch are used to not terminate loop in case of error.
            try {

            //Declaring the variables
            int Category;
            String Customer_Name, C_address;

            //Taking customer name from user
            System.out.println("Enter the Customer Name");
            Customer_Name = myScan.next();

            // Taking Customer Address from the user
            System.out.println("Enter the Customer Address");
            C_address = myScan.next();

            // Taking customer category from the user
            System.out.println("Enter the Customer Category");
            Category = myScan.nextInt();

            // Executing procedure for Query 1.
            final String Sql1 = "EXEC InsertCustomerInfo @Customer_Name = '"+Customer_Name+"', @C_address = '"+C_address+"', @Category = '"+Category+"';

            final Statement statement1 = connection.createStatement();
                statement1.executeUpdate(Sql1);
                System.out.println("Customer details inserted successfully.");
                System.out.println("=======================================");
            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

2. Enter a new department (infrequent).

```
        case 2:
            // try and catch are used to not terminate loop in case of error.
            try {

            // Declaring variables
            String Department_ID, Department_Data;

            // Taking Department Number from user.
            System.out.println("Enter the Department ID\n");
            Department_ID = myScan.next();

            // Taking Department data from user.
            System.out.println("Enter the Department Data\n");
            Department_Data = myScan.next();

            // Executing Procedure for Query 2.
            final String Sql2 = "EXEC InsertDepartmentInfo @Department_ID = '"+Department_ID+"', @Department_Data = '"+Department_Data+"';";

            final Statement statement2 = connection.createStatement();
                statement2.executeUpdate(Sql2);

                System.out.println("Department details inserted successfully.");
                System.out.println("========================================");
            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

3. Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered (40/day).

```
        case 3:
            // try and catch are used to not terminate loop in case of error.
            try {
                // Declaring Variables
            String Assembly_ID, Assembly_details, Customer_Name;
            String Date_ordered;

            // Taking Assembly-id from user
            System.out.println("Enter Assembly ID");
            Assembly_ID = myScan.next();

            // Taking date ordered from user
            System.out.println("Enter Date Ordered in yyyy-mm-dd format");
            Date_ordered = myScan.next();

            // Taking Assembly details from user
            System.out.println("Enter Assembly Details");
            Assembly_details = myScan.next();

            // Taking customer name from user
            System.out.println("Enter the Customer Name");
            Customer_Name = myScan.next();

            // Executing procedure for Query 3
            final String Sql3 = "EXEC InsertAssemblyInfo @Assembly_ID = '"+Assembly_ID+"', @Date_ordered = '"+Date_ordered+"'," +
                    "@Assembly_details = '"+Assembly_details+"', @Customer_Name = '"+Customer_Name+"';";

            final Statement statement3 = connect
                statement3.executeUpdate(Sql3);

                System.out.println("Assembly details inserted successfully.");
                System.out.println("========================================");

            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

String Assembly_ID, Assembly_details, Customer_Name;
Press 'F2' for focus

4. Enter a new process-id and its department together with its type and information relevant to the type (infrequent).

```
        case 4:
            // try and catch are used to not terminate loop in case of error.
            try {

            // Declaring Variables
            String Process_ID, Process_Data, Department_ID;

            // Taking process-id from the user
            System.out.println("Enter Process ID");
            Process_ID = myScan.next();

            // Taking process data from the user
            System.out.println("Enter Process Data");
            Process_Data = myScan.next();

            // Taking Department-no from the user
            System.out.println("Enter Department ID");
            Department_ID = myScan.next();

            //Executing Procedure for Query 4
            final String Sql4 = "EXEC InsertProcessIDdept @Process_ID = '"+Process_ID+"'," +
                    " @Process_Data = '"+Process_Data+"', @Department_ID = '"+Department_ID+"';";

            final Statement statement4 = connection.createStatement();
                statement4.executeUpdate(Sql4);

                System.out.println("Process details inserted successfully.");
                System.out.println("========================================");

            // Asking user to enter the type of procedure
            System.out.println("Choose one of the following type of process:\n 1.Fit\n 2. Paint\n 3.Cut\n");
            int choice1;
            // Taking the type of process from the user
            choice1 = myScan.nextInt();

            if (choice1 ==1) {

                // Declaring Variables
                String Fit_Type;

                // Taking fit type from user
                System.out.println("Enter fit type\n");
                Fit_Type = myScan.next();

                // Executing Procedure to insert data in Process_fit table
                final String Sql4_1 = "EXEC InsertProcessFit @Process_ID = '"+Process_ID+"',  @Fit_Type = '"+Fit_Type+"';";

                final Statement statement4_1 = connection.createStatement();
                    statement4_1.executeUpdate(Sql4_1);

                    System.out.println("Process_Fit details inserted successfully.");
                    System.out.println("========================================");
            }
                    System.out.println("========================================");
            }

            if (choice1 ==2) {

                // Declaring Variables
                String Paint_Type, Paint_Method;

                //Taking paint type from the user
                System.out.println("Enter paint type");
                Paint_Type = myScan.next();

                // Taking paint method from the user
                System.out.println("Enter paint method");
                Paint_Method = myScan.next();

                // Executing procedure to insert data into Process_paint table
                final String Sql4_2 = "EXEC InsertProcessPaint @Process_ID = '"+Process_ID+"', " +
                        " @Paint_Type = '"+Paint_Type+"', @Paint_Method = '"+Paint_Method+"';";

                final Statement statement4_2 = connection.createStatement();
                    statement4_2.executeUpdate(Sql4_2);

                    System.out.println("Process_Paint details inserted successfully.");
                    System.out.println("========================================");
            }

            if (choice1 ==3) {

                // Declaring Variables
                String Cut_Type, Machine_Type;

                //Taking cut type from the user
                System.out.println("Enter cutting type");
                Cut_Type = myScan.next();

                // Take machine type from user
                System.out.println("Enter machine type");
                Machine_Type = myScan.next();

                //Executing Procedure to insert data into Process_cut table.
                final String Sql4_3 = "EXEC InsertProcessCut @Process_ID = '"+Process_ID+"', " +
                        " @Cut_Type = '"+Cut_Type+"', @Machine_Type = '"+Machine_Type+"';";

                final Statement statement4_3 = connection.createStatement();
                    statement4_3.executeUpdate(Sql4_3);

                    System.out.println("Process_Cut record inserted successfully.");
                    System.out.println("========================================");
            }
            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

5. Create a new account and associate it with the process, assembly, or department to which it is applicable (10/day).

```
274
275         case 5:
276             // try and catch are used to not terminate loop in case of error.
277             try {
278 
279                 // Declaring Variables
280             int Account_No, choice2;
281 
282             // Taking account number from user
283             System.out.println("Enter account number");
284             Account_No = myScan.nextInt();
285 
286 
287             // Asking user to provide the type of account
288             System.out.println("Choose one of the following type of account:\n 1. Department Account.\n 2. Assembly Account\n 3.Process Account\n");
289             choice2 = myScan.nextInt();
290 
291 
292             if (choice2 == 1) {
293                 // Declaring Variables
294                 float details_2;
295                 String Department_ID;
296                 String Established_Date;
297 
298                 // Taking details-2 from the user
299                 System.out.println("Enter account details");
300                 details_2 = myScan.nextFloat();
301 
302                 // Taking date account established from user
303                 System.out.println("Enter date account established");
304                 Established_Date = myScan.next();
305 
306                 // Taking department no from the user
307                 System.out.println("Enter Department Number of the account");
308                 Department_ID = myScan.next();
309 
310                 // Executing procedure to insert data into Department Account Table
311                 final String Sql5_1 = "EXEC InsertAccountDept5 @Account_No = '"+Account_No+"', @Established_Date = '"+Established_Date+"',"+
312                         " @details_2 = '"+details_2+"', @Department_ID = '"+Department_ID+"';";
313 
314                 final Statement statement5_1 = connection.createStatement();
315                     statement5_1.executeUpdate(Sql5_1);
316 
317                     System.out.println("Department_account record inserted successfully.");
318                     System.out.println("========================================");
319             }
320 
321             if (choice2 == 2) {
322 
323                 // Declaring Variables
324                 float Details_1;
325                 String Assembly_ID;
326                 String Established_Date;
327 
328                 // Taking details1 from user
329                 System.out.println("Enter account details");
330                 Details_1 = myScan.nextFloat();
```

```
331
332                 // Taking date account established from user
333                 System.out.println("Enter date account established");
334                 Established_Date = myScan.next();
335
336                 // Taking Assembly-ID from user
337                 System.out.println("Enter Assembly id of the account");
338                 Assembly_ID = myScan.next();
339
340                 // Executing procedure to insert data into Assembly account
341                 final String Sql5_2 = "EXEC InsertAccountAssembly4 @Account_No = '"+Account_No+"',@Established_Date = '"+Established_Date+"',"+
342                         " @Details_1 = '"+Details_1+"',@Assembly_ID = '"+Assembly_ID+"';";
343
344                 final Statement statement5_2 = connection.createStatement();
345                     statement5_2.executeUpdate(Sql5_2);
346
347                     System.out.println("Assembly_acc record inserted successfully.");
348                     System.out.println("========================================");
349             }
350
351             if (choice2 == 3) {
352
353                 // Declaring Variables
354                 float Details_3;
355                 String Process_ID;
356                 String Established_Date;
357
358
359                 // Taking details from the user
360                 System.out.println("Enter account details");
361                 Details_3 = myScan.nextFloat();
362
363                 // Taking date account established from user
364                 System.out.println("Enter date account established");
365                 Established_Date = myScan.next();
366
367                 // Taking process-id from the user
368                 System.out.println("Enter Process id of the account");
369                 Process_ID = myScan.next();
370
371                 // Executing the procedure to insert data into Process account
372                 final String Sql5_3 = "EXEC InsertAccountProcess2 @Account_No = '"+Account_No+"',@Established_Date = '"+Established_Date+"',"+
373                         " @Details_3 = '"+Details_3+"',  @Process_ID = '"+Process_ID+"';";
374
375                 final Statement statement5_3 = connection.createStatement();
376                     statement5_3.executeUpdate(Sql5_3);
377
378                     System.out.println("Process_acc record inserted successfully.");
379                     System.out.println("========================================");
380             }
381             } catch (Exception e) {
382                 System.out.println("Error!. Returning to main menu");
383             }
384             break;
```

6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced (50/day).

```
386         case 6:
387             // try and catch are used to not terminate loop in case of error.
388             try {
389
390                 // Declaring Variables
391             String Date_job_commenced, Assembly_ID, Process_ID;
392             int Job_No;
393
394             // Taking Job-no from the user
395             System.out.println("Enter Job-No for a new job");
396             Job_No = myScan.nextInt();
397
398             // Taking assembly-id from the user
399             System.out.println("Enter assembly-ID for the job");
400             Assembly_ID = myScan.next();
401
402             // Taking process-id from user
403             System.out.println("Enter process-ID for the job");
404             Process_ID = myScan.next();
405
406             // Taking job commenced from user
407             System.out.println("Enter the commenced date for the job");
408             Date_job_commenced = myScan.next();
409
410             // Executing Procedure for Query 6
411             final String Sql6 = "EXEC InsertJobInfo1 @Job_No = '"+Job_No+"', @Date_job_commenced = '"+Date_job_commenced+"',"+
412                     " @Assembly_ID = '"+Assembly_ID+"', @Process_ID = '"+Process_ID+"';";
413
414             final Statement statement6 = connection.createStatement();
415                 statement6.executeUpdate(Sql6);
416
417                 System.out.println("Job record inserted successfully.");
418                 System.out.println("========================================");
419
420             } catch (Exception e) {
421                 System.out.println("Error!. Returning to main menu");
422             }
423             break;
424
425         case 7:
```

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

type of job (50/day).

```
        case 7:
            // try and catch are used to not terminate loop in case of error.
            try {
                // Declaring Variables
            String Date_job_completed;
            int Job_No;

            // Taking Job-no from user
            System.out.println("Enter Job-No for the completed job");
            Job_No = myScan.nextInt();

            // Taking date completed from user
            System.out.println("Enter job completed date in yyyy-mm-dd format");
            Date_job_completed = myScan.next();

            // Executing the Procedure for Query 7
            final String Sql7 = "EXEC UpdateJobInfo1 @Job_No = '"+Job_No+"', @Date_job_completed = '"+Date_job_completed+"';";

            final Statement statement7 = connection.createStatement();
                statement7.executeUpdate(Sql7);

                System.out.println("Job record Updated successfully.");
                System.out.println("========================================");

                // Declaring Variables
            int choice3;

            // Asking the type of job from the user
            System.out.println("Choose one of the following type of job:\n 1. Fit Job.\n 2. Paint Job\n 3.Cut Job\n");
            choice3 = myScan.nextInt();


            if (choice3 == 1) {

                // Declaring Variables
                String Labor_Time;

                // Taking fit labor time from user
                System.out.println("Enter the fit labor time fo Job in HH:MM:SS format.");
                Labor_Time = myScan.next();

                // Executing procedure to insert data into Job_fit table
                final String Sql7_1 = "EXEC InsertJobFitInfo @Job_No = '"+Job_No+"', @Labor_Time = '"+Labor_Time+"';";

                final Statement statement7_1 = connection.createStatement();
                    statement7_1.executeUpdate(Sql7_1);

                    System.out.println("Fit Job record inserted successfully.");
                    System.out.println("========================================");
            }

            if (choice3 == 2) {

                // Declaring Variables
                String Color, Paint_Labor_Time;
                int Volume;
```

```
                // Taking color from user
                System.out.println("Enter the paint color.");
                Color = myScan.next();

                // Taking labor time from user
                System.out.println("Enter the paint job labor time in HH:MM:SS format.");
                Paint_Labor_Time = myScan.next();

                // Taking volume of paint from user
                System.out.println("Enter the volume of paint.");
                Volume = myScan.nextInt();

                // Executing the procedure to insert data into Job paint table
                final String Sql7_2 = "EXEC InsertJobPaintInfo @Job_No = '"+Job_No+"', @Color = '"+Color+"', "+
                        " @Volume = '"+Volume+"', @Paint_Labor_Time = '"+Paint_Labor_Time+"';";

                final Statement statement7_2 = connection.createStatement();
                    statement7_2.executeUpdate(Sql7_2);

                    System.out.println("Paint Job record inserted successfully.");
                    System.out.println("========================================");
            }

            if (choice3 == 3) {

                // Declaring Variables
                String Job_Machine_Type, Machine_Time, Material, Cut_Labor_Time;

                // Taking machine type from user
                System.out.println("Enter the Job Machine Type.");
                Job_Machine_Type = myScan.next();

                // Taking machine time from user
                System.out.println("Enter the cut job machine time in HH:MM:SS format.");
                Machine_Time = myScan.next();

                // Taking material used from the user
                System.out.println("Enter the material used.");
                Material = myScan.next();

                // Taking labor time from the user
                System.out.println("Enter the cut job labor time in HH:MM:SS format.");
                Cut_Labor_Time = myScan.next();

                // Executing Procedure to insert data into Job_cut table
                final String Sql7_3 = "EXEC InsertJobCutInfo @Job_No = '"+Job_No+"', @Job_Machine_Type= '"+Job_Machine_Type+"', "+
                " @Machine_Time = '"+Machine_Time+"', @Material = '"+Material+"', @Cut_Labor_Time = '"+Cut_Labor_Time+"';";

                final Statement statement7_3 = connection.createStatement();
                    statement7_3.executeUpdate(Sql7_3);

                    System.out.println("Cut Job record inserted successfully.");
                    System.out.println("========================================");
            }
            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

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 (50/day).

```
            break;
        case 8:
            // try and catch are used to not terminate loop in case of error.
            try {

                // Declaring Variables
            String Transaction_No;
            int Job_No;
            float Given_Cost;

            // Taking transaction number from the user
            System.out.println("Enter the Transaction Number.");
            Transaction_No = myScan.next();

            // Taking sup-cost from the user
            System.out.println("Enter the cost of the transaction.");
            Given_Cost= myScan.nextFloat();

            // Taking job-No from the user
            System.out.println("Enter the Job number related to transaction.");
            Job_No = myScan.nextInt();

            // Executing procedure for Query 8.
            final String Sql8 = "EXEC InsertTransactionInfo5 @Transaction_No = '"+Transaction_No+"', @Given_Cost = '"+Given_Cost+"', @Job_No = '"+Job_No+"';

            final Statement statement8 = connection.createStatement();
                statement8.executeUpdate(Sql8);

                System.out.println("Transaction record inserted and related accounts updated successfully.");
                System.out.println("========================================");

            } catch (Exception e) {
                System.out.println("Error!. Returning to main menu");
            }
            break;
```

9. Retrieve the cost incurred on an assembly-id (200/day).

```
575     case 9:
576         // try and catch are used to not terminate loop in case of error.
577         try {
578             // Declaring Variables
579         String Assembly_ID3;
580 
581         // Taking assembly-id from the user
582         System.out.println("Enter Assembly Id to retrieve the cost.");
583         Assembly_ID3 = myScan.next();
584 
585         try {
586 
587         // Executing procedure for Query 9
588         final String Sql9 = "EXEC RetriveCost @Assembly_ID = '"+Assembly_ID3+"';";
589 
590         try (final Statement statement9 = connection.createStatement();
591                 final ResultSet resultSet1 = statement9.executeQuery(Sql9)) {
592 
593                 System.out.println(String.format("Cost incurred on Assembly-ID %s:", Assembly_ID3));
594                 while (resultSet1.next()) {
595                     System.out.println(String.format("%f",
596                         resultSet1.getFloat(1)));
597 
598                     }
599                 }
600         } catch (Exception e) {
601             System.out.println("error!. Please try again.");
602         }
603 
604         } catch (Exception e) {
605             System.out.println("Error!. Returning to main menu");
606         }
607         break;
608 
```

10. Retrieve the total labor time within a department for jobs completed in the department during a given date (20/day).

```
609     case 10:
610         // try and catch are used to not terminate loop in case of error.
611         try {
612 
613             // Declaring Variables
614         String Department_No2,Date_job_completed1;
615 
616         // Taking department-no from user
617         System.out.println("Enter Department Number to get the total labor time.");
618         Department_No2 = myScan.next();
619 
620         // Taking job-completed date from user
621         System.out.println("Enter Job completed date to get the total labor time.");
622         Date_job_completed1 = myScan.next();
623 
624         // Executing the procedure for Query 10
625         final String Sql10 = "EXEC RetrieveLaboreTime @Department_ID = '"+Department_No2+"', @Date_job_completed = '"+Date_job_completed1+"';";
626 
627         try (final Statement statement10 = connection.createStatement();
628                 final ResultSet resultSet2 = statement10.executeQuery(Sql10)) {
629 
630                 System.out.println(String.format("Total labor-time in minutes for department number %s and date" +
631                         " job completed %s:", Department_No2, Date_job_completed1));
632                 while (resultSet2.next()) {
633                     System.out.println(String.format("%s",
634                         resultSet2.getInt(1)));
635 
636                     }
637                 }
638         } catch (Exception e) {
639             System.out.println("Error!. Returning to main menu");
640         }
641         break;
642 
```

11. Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process (100/day).

```
        break;

    case 11:
        // try and catch are used to not terminate loop in case of error.
        try {

            // Declaring Variables
        String Assembly_ID4;

        // Taking assembly-id from user
        System.out.println("Enter the Assembly Id to retrieve the processes.");
        Assembly_ID4 = myScan.next();

        // Executing procedure for Query 11d
        final String Sql11 = "EXEC RetrieveProcessAssembly @Assembly_ID = '"+Assembly_ID4+"';";

        try (final Statement statement11 = connection.createStatement();
                final ResultSet resultSet3 = statement11.executeQuery(Sql11)) {

                System.out.println(String.format("The processes through which Assembly ID %s passed so far:", Assembly_ID4));
                while (resultSet3.next()) {
                    System.out.println(String.format("%s | %s | %s",
                        resultSet3.getString(1),
                        resultSet3.getString(2),
                        resultSet3.getString(3)));

                    }
                }
        } catch (Exception e) {
            System.out.println("Error!. Returning to main menu");
        }
        break;
```

12. Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department (20/day).

```
    case 12:
        // try and catch are used to not terminate loop in case of error.
        try {

            // Declaring Variables
        String Date_job_completed2, Department_No3;

        // Taking date completed from user
        System.out.println("Enter the date job is completed.");
        Date_job_completed2 = myScan.next();

        // Taking department number from user
        System.out.println("Enter the department to retrieve the jobs.");
        Department_No3 = myScan.next();

        // Executing Procedure to retrieve fit jobs
        final String Sql12_1 = "EXEC RetrieveJobAssemblyFit @Date_job_completed = '"+Date_job_completed2+"', @Department_ID = '"+Department_No3+"';";

        try (final Statement statement12_1 = connection.createStatement();
                final ResultSet resultSet5 = statement12_1.executeQuery(Sql12_1)) {

                System.out.println(String.format("The fit jobs completed on date %s in department %s:", Date_job_completed2, Department_No3));
                while (resultSet5.next()) {
                    System.out.println(String.format("%s | %s | %s",
                        resultSet5.getString(1),
                        resultSet5.getString(2),
                        resultSet5.getString(3)));

                    }
                }
        // Executing procedure to retrieve paint jobs
        final String Sql12_2 = "EXEC RetrieveJobAssemblyPaint @Date_job_completed = '"+Date_job_completed2+"', @Department_ID = '"+Department_No3+"';";

        try (final Statement statement12_2 = connection.createStatement();
                final ResultSet resultSet6 = statement12_2.executeQuery(Sql12_2)) {

                System.out.println(String.format("The Paint jobs completed on date %s in department %s:", Date_job_completed2, Department_No3));
                while (resultSet6.next()) {
                    System.out.println(String.format("%s | %s | %s | %s | %s",
                        resultSet6.getString(1),
                        resultSet6.getString(2),
                        resultSet6.getString(3),
                        resultSet6.getString(4),
                        resultSet6.getString(5)));

                    }
                }
```

```
21
22
23          // Executing procedure to retrieve cut jobs
24          final String Sql12_3 = "EXEC RetrieveJobAssemblyCut @Date_job_completed = '"+Date_job_completed2+"', @Department_ID = '"+Department_No3+"';";
25
26          try (final Statement statement12_3 = connection.createStatement();
27                  final ResultSet resultSet7 = statement12_3.executeQuery(Sql12_3)) {
28
29                  System.out.println(String.format("The Cut jobs completed on date %s in department %s:", Date_job_completed2, Department_No3));
30                  while (resultSet7.next()) {
31                      System.out.println(String.format("%s | %s | %s | %s | %s | %s",
32                          resultSet7.getString(1),
33                          resultSet7.getString(2),
34                          resultSet7.getString(3),
35                          resultSet7.getString(4),
36                          resultSet7.getString(5),
37                          resultSet7.getString(6)));
38
39                      }
40                  }
41          } catch (Exception e) {
42              System.out.println("Error!. Returning to main menu");
43          }
44
45          break;
```

13. Retrieve the customers (in name order) whose category is in a given range (100/day).

```
747     case 13:
748         // try and catch are used to not terminate loop in case of error.
749         try {
750
751             // Declaring Variables
752         int Lower_b, Upper_b;
753
754         // Taking lower bound of category from user
755         System.out.println("Enter the lower bound of the category.");
756         Lower_b = myScan.nextInt();
757
758         // Taking upper bound of category from user
759         System.out.println("Enter the upper bound of the category.");
760         Upper_b = myScan.nextInt();
761
762         // Executing procedure for Query 13
763         final String Sql13 = "EXEC RetrieveCustomerByCategory @Lower_b = '"+Lower_b+"', @Upper_b = '"+Upper_b+"';";
764
765         try (final Statement statement13 = connection.createStatement();
766                 final ResultSet resultSet4 = statement13.executeQuery(Sql13)) {
767
768                 System.out.println("The customers in the given range of category are");
769                 while (resultSet4.next()) {
770                     System.out.println(String.format("%s | %s",
771                         resultSet4.getString(1),
772                         resultSet4.getString(2)));
773
774                     }
775                 }
776         } catch (Exception e) {
777             System.out.println("Error!. Returning to main menu");
778         }
779         break;
780
```

14. Delete all cut-jobs whose job-no is in a given range (1/month).

```
781      case 14:
782          // try and catch are used to not terminate loop in case of error.
783          try {
784
785              // Declaring Variables
786          int lower_b1, upper_b1;
787
788          // Taking lower bound job-no from user
789          System.out.println("Enter the lower bound of the cut job-no.");
790          lower_b1 = myScan.nextInt();
791
792          // Taking upper bound job-no from user
793          System.out.println("Enter the upper bound of the cut job-no.");
794          upper_b1 = myScan.nextInt();
795
796          // Executing procedure for Query 14
797          final String Sql14 = "EXEC DeleteCutJob @Lower_b = '"+lower_b1+"', @Upper_b = '"+upper_b1+"';";
798
799          final Statement statement14 = connection.createStatement();
800              statement14.executeUpdate(Sql14);
801
802              System.out.println("Deleted all cut jobs in the given range of Job-No.");
803              System.out.println("========================================");
804
805          } catch (Exception e) {
806              System.out.println("Error!. Returning to main menu");
807          }
808          break;
809
```

15. Change the color of a given paint job (1/week).

```
810      case 15:
811          // try and catch are used to not terminate loop in case of error.
812          try {
813
814              // Declaring Variables
815          String Color;
816          int Job_No3;
817
818          // Taking job-no from user
819          System.out.println("Enter the paint Job-No.");
820          Job_No3 = myScan.nextInt();
821
822          // Taking color from user
823          System.out.println("Enter the new color for the job-no.");
824          Color = myScan.next();
825
826          // Executing procedure for Query 15
827          final String Sql15 = "EXEC ChangePaintColor @Job_No = '"+Job_No3+"', @Color = '"+Color+"';";
828
829          final Statement statement15 = connection.createStatement();
830              statement15.executeUpdate(Sql15);
831
832              System.out.println("Changed the color of a given paint job.");
833              System.out.println("========================================");
834
835          } catch (Exception e) {
836              System.out.println("Error!. Returning to main menu");
837          }
838          break;
839
```

16. Import: enter new customers from a data file until the file is empty (the user must be asked to enter the input file name).

```
840     case 16:
841         // try and catch are used to not terminate loop in case of error.
842         try {
843             // Declaring Variables
844         String file_name, line;
845
846         // Taking Input file name from user
847         System.out.println("Enter the file-name to Import data.");
848         file_name = myScan.next();
849
850         // Creating new File object
851         File file = new File(file_name);
852
853         // Creating new Scanner Object
854         Scanner sc = new Scanner(file);
855
856         // While loop to read all lines in the input file
857         while(sc.hasNextLine()) {
858             line = sc.nextLine();
859
860             // Dividing line to parts separated by Delimiter (",")
861             String[] parts = line.split(",");
862
863
864                 String Customer_Name= parts[0];
865                 String C_address = parts[1];
866                 int Category = Integer.parseInt(parts[2]);
867
868                 // Execute procedure for Query 16
869                 final String Sql16 = "EXEC InsertCustomerInfo @Customer_Name = '"+Customer_Name+"', @C_address = '"+C_address+"', @Category = '"+Categor
870
871                 final Statement statement16 = connection.createStatement();
872                     statement16.executeUpdate(Sql16);
873             }
874         sc.close();
875         }
876
877
878     catch (Exception e) {
879             System.out.print("Error!. Returning to main menu: ");
880         }
881
```

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

```
384     case 17:
385         // try and catch are used to not terminate loop in case of error.
386         try {
387
388             // Declaring Variables
389             String File_name1, Lower_b2, Upper_b2;
390
391             // Enter the filename to output result
392             System.out.println("Enter the file-name to Export data.");
393             File_name1 = myScan.next();
394
395             // Taking lower bound of category from user
396             System.out.println("Enter the lower bound of category.");
397             Lower_b2 = myScan.next();
398
399             // Taking upper bound of category from user
400             System.out.println("Enter the upper bound of category.");
401             Upper_b2 = myScan.next();
402
403             // Creating new file writer Object
404             FileWriter fw = new FileWriter(File_name1);
405
406             // Executing Procedure(for Query 13) to get output
407             final String Sql17 = "EXEC RetrieveCustomerByCategory1- @Lower_b2 = '"+Lower_b2+"', @Upper_b2 = '"+Upper_b2+"';";
408
409
410             try (final Statement statement17 = connection.createStatement();
411                     final ResultSet resultSet4 = statement17.executeQuery(Sql17)) {
412
413                     //System.out.println("The customers in the given range of category are");
414                     while (resultSet4.next()) {
415
416                         //fw.write(String.format("%s,%s", resultSet4.getString(1), resultSet4.getString(2)));
417                         fw.write(resultSet4.getString(1) + "," + resultSet4.getString(2) + "\n");
418                         }
419                     fw.close();
420                     } catch (SQLException e) {
421                         e.getCause().getMessage();
422                     }
423
424
425             } catch (Exception e) {
426                System.out.println("Error!. Returning to main menu");
427             }
428         break;
429
```

18. Quit

```
        case 18:
            System.out.println("You choose to Quit!");

        }
}
```

Task 6: execution of Queries

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
1
Enter the Customer Name
Lisa
Enter the Customer Address
Boston
Enter the Customer Category
1
Customer details inserted successfully.
========================================
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
1
Enter the Customer Name
John
Enter the Customer Address
Chicago
Enter the Customer Category
2
Customer details inserted successfully.
----------------------------------------
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
1
Enter the Customer Name
Rita
Enter the Customer Address
Oklahoma
Enter the Customer Category
4
Customer details inserted successfully.
========================================
```

```
----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
1
Enter the Customer Name
Ben
Enter the Customer Address
Michigan
Enter the Customer Category
6
Customer details inserted successfully.
```

customerInfo Table: -

Results Messages

Search to filter items...

Customer_Name	C_address	Category
Ben	Michigan	6
John	Chicago	2
Lisa	Boston	1
Rita	Oklahoma	4

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
2
Enter the Department ID

depth5
Enter the Department Data

ghte2
Department details inserted successfully.
========================================
```

Department Table: -

Department_ID	Department_Data
Dept1	CS1
Dept2	TB2
Dept3	Phy4
depth5	ghte2

```
----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
3
Enter Assembly ID
Assbl1
Enter Date Ordered in yyyy-mm-dd format
1999-12-23
Enter Assembly Details
Assembly23
Enter the Customer Name
Lisa
Assembly details inserted successfully.
========================================
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
3
Enter Assembly ID
Assbl2
Enter Date Ordered in yyyy-mm-dd format
2000-03-12
Enter Assembly Details
Assembly2
Enter the Customer Name
John
Assembly details inserted successfully.
========================================
Choose one Option:
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
3
Enter Assembly ID
Assbl3
Enter Date Ordered in yyyy-mm-dd format
1898-04-12
Enter Assembly Details
Assembl67
Enter the Customer Name
Ben
Assembly details inserted successfully.
========================================
Choose one Option:
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
3
Enter Assembly ID
Assbl4
Enter Date Ordered in yyyy-mm-dd format
2023-11-12
Enter Assembly Details
Assembl45
Enter the Customer Name
Rita
Assembly details inserted successfully.
========================================
```

```
----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
4
Enter Process ID
Proc2
Enter Process Data
ProcessT2
Enter Department ID
Dept2
Process details inserted successfully.
========================================
Choose one of the following type of process:
 1.Fit
 2. Paint
 3.Cut

2
Enter paint type
Paint2
Enter paint method
PiantMachine
Process_Paint details inserted successfully.
```

```
<terminated> Performed_Queries [Java Application] /Library/Java/JavaVirtualMachines/jdk-11.jdk/Contents/Home/bin/java  (Nov 16, 202
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
4
Enter Process ID
Proc1
Enter Process Data
ProcessT1
Enter Department ID
Dept1
Process details inserted successfully.
========================================
Choose one of the following type of process:
 1.Fit
 2. Paint
 3.Cut

1
Enter fit type

Fit1
Process_Fit details inserted successfully.
========================================
```

```
-----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
=========================================
4
Enter Process ID
Proc3
Enter Process Data
ProcessT3
Enter Department ID
Dept3
Process details inserted successfully.
=========================================
Choose one of the following type of process:
 1.Fit
 2. Paint
 3.Cut

3
Enter cutting type
CutMachine
Enter machine type
Machine1
Process_Cut record inserted successfully.
=========================================
```

Pocess1 Table

Results Messages

Search to filter items...

Process_ID	Process_Data	Department_ID
Proc1	ProcessT1	Dept1
Proc2	ProcessT2	Dept2
Proc3	ProcessT3	Dept3

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
1
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

1
Enter account details
12
Enter date account established
2010-12-23
Enter Department Number of the account
Dept1
Department_account record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
45
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

1
Enter account details
456
Enter date account established
2001-01-27
Enter Department Number of the account
Dept2
Department_account record inserted successfully.
----------------------------------------
```

Department_acc Table

Results Messages

Search to filter items...

Account_No	Established_Date	details_2	Department_ID
1	2010-12-23T00:00:00.0000000	12	Dept1
45	2001-01-27T00:00:00.0000000	456	Dept2

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
56
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

2
Enter account details
657
Enter date account established
2009-07-10
Enter Assembly id of the account
Assbl1
Assembly_acc record inserted successfully.
========================================
```

```
----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
79
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

2
Enter account details
98
Enter date account established
2000-10-23
Enter Assembly id of the account
Assbl2
Assembly_acc record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
6
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

3
Enter account details
78
Enter date account established
2009-05-12
Enter Process id of the account
Proc1
Process_acc record inserted successfully.
========================================
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
5
Enter account number
8
Choose one of the following type of account:
 1. Department Account.
 2. Assembly Account
 3.Process Account

3
Enter account details
987
Enter date account established
1999-02-12
Enter Process id of the account
Proc2
Process_acc record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
6
Enter Job-No for a new job
2
Enter assembly-ID for the job
Assbl2
Enter process-ID for the job
Proc2
Enter the commenced date for the job
2022-12-23
Job record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
6
Enter Job-No for a new job
7
Enter assembly-ID for the job
Assbl1
Enter process-ID for the job
Proc1
Enter the commenced date for the job
2023-12-24
Job record inserted successfully.
========================================
Choose one Option:
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
6
Enter Job-No for a new job
4
Enter assembly-ID for the job
Assbl3
Enter process-ID for the job
Proc1
Enter the commenced date for the job
2021-06-23
Job record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
7
Enter Job-No for the completed job
4
Enter job completed date in yyyy-mm-dd format
2023-06-21
Job record Updated successfully.
========================================
Choose one of the following type of job:
 1. Fit Job.
 2. Paint Job
 3.Cut Job

2
Enter the paint color.
Blue
Enter the paint job labor time in HH:MM:SS format.
06:20:00
Enter the volume of paint.
1
Paint Job record inserted successfully.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
7
Enter Job-No for the completed job
7
Enter job completed date in yyyy-mm-dd format
2023-12-01
Job record Updated successfully.
========================================
Choose one of the following type of job:
 1. Fit Job.
 2. Paint Job
 3.Cut Job

1
Enter the fit labor time fo Job in HH:MM:SS format.
03:30:00
Fit Job record inserted successfully.
----------------------------------------
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
7
Enter Job-No for the completed job
2
Enter job completed date in yyyy-mm-dd format
2023-12-12
Job record Updated successfully.
========================================
Choose one of the following type of job:
 1. Fit Job.
 2. Paint Job
 3.Cut Job

3
Enter the Job Machine Type.
machine1
Enter the cut job machine time in HH:MM:SS format.
02:00:00
Enter the material used.
M1
Enter the cut job labor time in HH:MM:SS format.
03:00:00
Cut Job record inserted successfully.
========================================
```

Job1 Table

Results Messages

Search to filter items...

Job_No	Date_job_commenced	Date_job_completed	Assembly_ID	Process_ID
2	2022-12-23T00:00:00.0000000	2023-12-12T00:00:00.0000000	Assbl2	Proc2
4	2021-06-23T00:00:00.0000000	2023-06-21T00:00:00.0000000	Assbl3	Proc1
7	2023-12-24T00:00:00.0000000	2023-12-01T00:00:00.0000000	Assbl1	Proc1

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
8
Enter the Transaction Number.
1
Enter the cost of the transaction.
300
Enter the Job number related to transaction.
2
Transaction record inserted and related accounts updated successfully.
========================================
```

```
----------------------------------------
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
8
Enter the Transaction Number.
2
Enter the cost of the transaction.
200
Enter the Job number related to transaction.
4
Transaction record inserted and related accounts updated successfully.
========================================
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
9
Enter Assembly Id to retrieve the cost.
Assbl1
Cost incurred on Assembly-ID Assbl1:
657.000000
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
10
Enter Department Number to get the total labor time.
Dept1
Enter Job completed date to get the total labor time.
2023-12-12
Total labor-time in minutes for department number Dept1 and date job completed 2023-12-12:
0
Choose one Option:
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
11
Enter the Assembly Id to retrieve the processes.
Assbl1
The processes through which Assembly ID Assbl1 passed so far:
2023-12-24 | Proc1 | Dept1
```

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
12
Enter the date job is completed.
2023-12-12
Enter the department to retrieve the jobs.
Dept2
The fit jobs completed on date 2023-12-12 in department Dept2:
The Paint jobs completed on date 2023-12-12 in department Dept2:
The Cut jobs completed on date 2023-12-12 in department Dept2:
2 | Assbl2 | machine1 | 02:00:00.0000000 | M1 | 03:00:00.0000000
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
13
Enter the lower bound of the category.
2
Enter the upper bound of the category.
4
The customers in the given range of category are
John | 2
Rita | 4
```

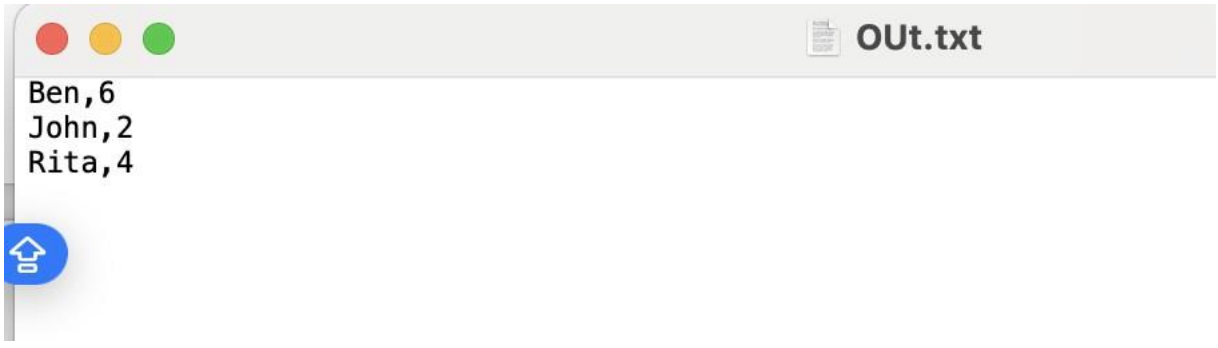
```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
14
Enter the lower bound of the cut job-no.
3
Enter the upper bound of the cut job-no.
7
Deleted all cut jobs in the given range of Job-No.
========================================
```

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
15
Enter the paint Job-No.
4
Enter the new color for the job-no.
Red
Changed the color of a given paint job.
```

Input.txt

```
Customer4,Paris,8
Customer5,Syracus,6
Customer6,NewYork,7
Customer7,SanJose,9
Customer8,Venis,3
```

```
==========
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
==========
16
Enter the file-name to Import data.
Input.txt
==========
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
==========
17
Enter the file-name to Export data.
OUt.txt
Enter the lower bound of category.
2
Enter the upper bound of category.
6
```



Results Messages

Search to filter items...

Customer_Name	C_address	Category
Ben	Michigan	6
Customer4	Paris	8
Customer5	Syracus	6
Customer6	NewYork	7
Customer7	SanJose	9
Customer8	Venis	3
John	Chicago	2
Lisa	Boston	1
Rita	Oklahoma	4

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
18
You choose to Quit!
```

Check error on Primary key

```
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
1
Enter the Customer Name
Lisa
Enter the Customer Address
Boston
Enter the Customer Category
5
Error!. Returning to main menu
```

Error on data format

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
7
Enter Job-No for the completed job
90-78-12
Error!. Returning to main menu
```

Error on foreign key

```
========================================
Choose one Option:
1. Enter a new Customer
2. Enter a new Department
3. Enter a new assembly
4. Enter a new process and information related to type
5. Create a new account and associate it with the one of the 3 type
6. Enter a new Job
7. Enter the date job is completed and the type of job
8. Enter a transaction details and update all accounts
9. Retrieve the cost for an assembly-id
10. Retrieve the total labor time,a department ID for jobs completed during a given date
11. Retrieve the processes through which a given assembly-id has passed with the department ID
12. Retrieve all jobs completed during the given date by a given department
13. Retrieve the customers whitin a range of category
14. Delete all cut-jobs whitin a range Job-No.
15. Change the color of paint job.
16. Import new customers detail from a data file.
17. Retrieve and export customers record whitin a range of category.
18. QUIT
========================================
4
Enter Process ID
Proc5
Enter Process Data
Process6
Enter Department ID
90
Error!. Returning to main menu
```

Task 7. (23 points): Write a Web database application using Azure SQL Database and JSP which provides the Web pages for query 1 and query 12.

Handler Code:

```
package Final_Project;


import java.sql.Connection; import

java.sql.ResultSet; import

java.sql.SQLException; import

java.sql.DriverManager; import

java.sql.PreparedStatement;


public class DataHandlerP {

        private Connection conn;
```

```
        // Azure SQL connection credentials            private
String server = "tah0001.database.windows.net";      private
String database = "cs_dsa_4513_F23";        private String
username = "tah0001";     private String password = "*****";
        // Resulting connection string
final private String url =

String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertif
icate=false;host NameInCertificate=*.database.windows.net;loginTimeout=30;",   server, database, username,
password);        // Initialize and save the database connection            private void getDBConnection() throws
SQLException {   if (conn != null) {   return; }
        this.conn = DriverManager.getConnection(url); }

// Return the result of selecting everything from the Customer table
public ResultSet getAllCustomers() throws SQLException {
getDBConnection(); // Prepare the database connection
// Prepare the SQL statement
                final String sqlQuery = "SELECT * FROM CustomerInfo;";
final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
// Execute the query return
stmt.executeQuery();
        }

// Return the result of selecting Customers with their Category in the given range from Customer table
public   ResultSet   retrieveCustomers(int   Lower_b,   int   Upper_b)   throws   SQLException   {
getDBConnection(); // Prepare the database connection
```

```
// Prepare the SQL statement
                        final String sqlQuery = "SELECT Customer_Name, Category FROM customerInfo" +"
WHERE Category >= ? AND Category <= ? ORDER BY 1 ;";                    final PreparedStatement stmt
= conn.prepareStatement(sqlQuery);
// Replace the '?' in the above statement with the given attribute values
                        stmt.setInt(1, Lower_b);
stmt.setInt(2, Upper_b);
// Execute the query
            return stmt.executeQuery();
                        }
// Inserts a record into the CustomerInfo table with the given attribute values        public
boolean addCustomer(String Customer_Name, String C_address, int Category)   throws
SQLException {getDBConnection(); // Prepare the database connection
                        // Prepare the SQL statement
                        final String sqlQuery ="INSERT INTO customerInfo " +"(Customer_Name, C_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, C_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;
            }
}
```

Add new customer Code

```
<%@ 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="Final_Project.DataHandlerP"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandlerP handler = new DataHandlerP();
// Get the attribute values passed from the input form.
String Name = request.getParameter("Customer_Name");
String Address = request.getParameter("C_address");
String category = request.getParameter("Category");
/*
```

```
* If the user hasn't filled out all the Customer_name, address and category. This is very simple
checking.
*/
if (Name.equals("") || Address.equals("") || category.equals("")) {
response.sendRedirect("Add_new_customer_form.jsp");
} else { int Category1=
Integer.parseInt(category);
// Now perform the query with the data from the form.
boolean success = handler.addCustomer(Name,Address, Category1); if
(!success) { // Something went wrong
%>
<h2>There was a problem inserting the course</h2>
<%
} else { // Confirm success to the user
%>
<h2>The Customer Details:</h2>
```



```
<ul>
<li>Customer Name: <%=Name%></li>
<li>Address: <%=Address%></li>
<li>Category: <%=category%></li>
</ul>
<h2>Was successfully inserted.</h2>
```

```
<a href="get_all_customers.jsp">See all Customers.</a>
<%
}
}
%>
</body>
</html>
```

Add new customer Form

```
DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Add Customer</title>
</head>
<body>
<h2>Add Customer</h2>
<!--
Form for collecting user input for the new customer record.
Upon form submission, add_new_customer.jsp file will be invoked.
-->
<form method= "POST" action="Add_New_Customer.jsp">
<!-- The form organized in an HTML table for better clarity. -->
<table border=1>
```

```
<tr>
<th colspan="2">Enter the Customer Data:</th>
</tr>
<tr>
<td>Customer Name:</td>
<td><div style="text-align: center;">
<input type=text name="Customer_Name">
</div></td>
</tr>
<tr>
<td>Customer Address:</td>
<td><div style="text-align: center;">
<input type=text name="C_address">
</div></td>
</tr>
<tr>
<td>Category:</td>
<td><div style="text-align: center;">
<input type=text name="Category">
</div></td>
</tr>
<tr>
<td><div style="text-align: center;">
<input type=reset value=Clear>
```

```
</div></td>
<td><div style="text-align: center;">
<input type=submit value=Insert>
</div></td>
</tr>
</table>
</form>
</body>
</html>
```

Get all Customer Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
	<head>
	<meta charset="UTF-8">
		<title>Customers</title>
	</head>
	<body>
	<%@page import="Final_Project.DataHandlerP"%>
	<%@page import="java.sql.ResultSet"%>
	<%
```

```
                // We instantiate the data handler here, and get all the customers from the
database                final DataHandlerP handler1 = new DataHandlerP();              final
ResultSet customers = handler1.getAllCustomers();
        %>
        <!-- The table for displaying all the Customer records -->
        <table  border="1">
                <tr> <!-- The table headers row -->
                        <td align="center">
                                <h4>Customer</h4>
                        </td>
                        <td align="center">
                                <h4>Address</h4>
                        </td>
                        <td align="center">
                                <h4>Category</h4>
                </tr>
                <%
                        while(customers.next()) { // For each Customer record returned...
   // Extract the attribute values for every row returned     final String Customer_Name =
customers.getString("Customer_Name");
                                final String C_address = customers.getString("C_address");
                final String Category = customers.getString("Category");
```

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

Retrieve Customer Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
        <head>
        <meta charset="UTF-8">
                <title>Customers</title>
        </head>
```

```
<body>
<%@page import="Final_Project.DataHandlerP"%>
<%@page import="java.sql.ResultSet"%>
<%
 // We instantiate the data handler here, and get all the customers from the database    final
DataHandlerP handler = new DataHandlerP();  // Get the attribute values passed from the input form.
                String lower_b = request.getParameter("Lower_b");
                String upper_b = request.getParameter("Upper_b");

                if (lower_b.equals("")||upper_b.equals("")) {
                        response.sendRedirect("retrieve_customer_form.jsp");
                } else {                int Lower_b1 =
Integer.parseInt(lower_b);              int Upper_b1 =
Integer.parseInt(upper_b);
                // Now perform the query with the data from the form.
                final ResultSet customers = handler.retrieveCustomers(Lower_b1, Upper_b1);
%>
<!-- The table for displaying all the Customer records -->
<table  border="1">
                <tr> <!-- The table headers row -->
                        <td align="center">
                                <h4>Customer</h4>
                        </td>
                        <td align="center">
```

```
                    <h4>category</h4>
                </td>
            <%
                while(customers.next()) { // For each Customer record returned...
  // Extract the attribute values for every row returned     final String Customer_Name =
customers.getString("Customer_Name");
                    final String Category = customers.getString("Category");
            out.println("<tr>"); // Start printing out the new table row
out.println( // Print each attribute value
                        "<td align=\"center\">" + Customer_Name +
                    "</td><td align=\"center\"> " + Category + "</td>");
                out.println("</tr>");
                }
            }
                %>
            </table>
        </body>
</html>
```

Retrieve Customer Form Code

```
DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<title>Retrieve customers given category range</title>
</head>
<body>
<h2>Give Customers Range</h2>
<!--
Form for collecting user input for the new customer record.
Upon form submission, retrieve_customers.jsp file will be invoked.
-->
<form method = "POST" action="retrieve_customer.jsp">
<!-- The form organized in an HTML table for better clarity. -->
<table border=1>
<tr>
<th colspan="2">Enter the Category Range:</th>
</tr>
<tr>
<td>Lower Bound:</td>
<td><div style="text-align: center;">
<input type=text name="Lower_b">
</div></td>
</tr>
<tr>
<td>Upper Bound:</td>
<td><div style="text-align: center;">
```

```
<input type=text name="Upper_b">
</div></td>
</tr>
<tr>
<td><div style="text-align: center;">
<input type=reset value=Clear>
</div></td>
<td><div style="text-align: center;">
<input type=submit value=Insert>
</div></td>
</tr>
</table>
</form>
</body>
</html>
```

Screenshot

Add Customer

Enter the Customer Data:	
Customer Name:	Remi
Customer Address:	Tunisia
Category:	8
Clear	Insert

The Customer Details:

Page 23 of 23

- Customer Name: Remi
- Address: Tunisia
- Category: 8

Was successfully inserted.

See all Customers.

Customer	Address	Category
Ben	Michigan	6
Customer4	Paris	8
Customer5	Syracus	6
Customer6	NewYork	7
Customer7	SanJose	9
Customer8	Venis	3
John	Chicago	2
Lisa	Boston	1
Remi	Tunisia	8
Rita	Oklahoma	4

Retrieve Customer details

Give Customers Range

Enter the Category Range:	
Lower Bound:	2
Upper Bound:	8
Clear	Insert

Customer	category
Ben	6
Customer4	8
Customer5	6
Customer6	7
Customer8	3
John	2
Remi	8
Rita	4