

JOB-SHOP ACCOUNTING DATABASE SYSTEM

COURSE: CS/DSA-4513

NAME: Database Management Systems

SECTION: 001

SEMESTER/YEAR: FALL 2023

INSTRUCTOR: Dr. Le Gruenwald

AUTHOR: Sofia Schnurrenberger

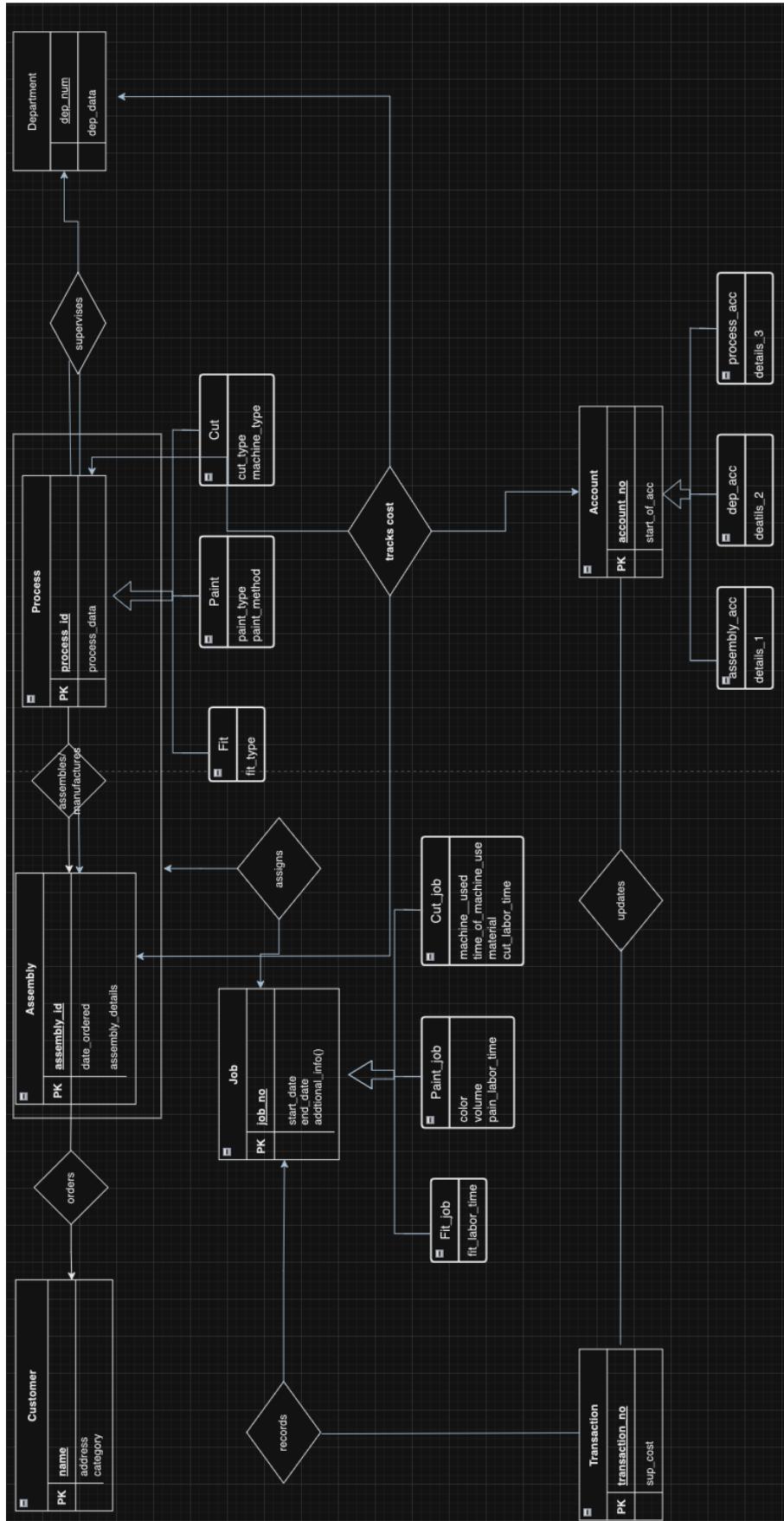
ID: 113489233

EMAIL ADDRESS: sofia.e.schnurrenberger-1@ou.edu

Table of Contents

<i>Task 1: ER Diagram</i>	3
<i>Task 2: Relational Database Schemas</i>	4
<i>Task 3: Storage Structures.....</i>	6
Task 3.1:	6
Task 3.2	11
3.3 Data Dictionary.....	16
<i>Task 4: Table Creation in Azure SQL</i>	18
<i>Task 5:</i>	26
5.1 SQL Statements and Transact SQL Stored procedures (queries 1-15) :	26
5.2 Java Source Program	42
<i>Task 6:.....</i>	85
Task 6.2 Query 1 Screenshots:.....	85
6. 2: Query 2 Screenshots.....	88
Task 6.3 Query 3 Screenshots:.....	91
Task 6. 4 Query 4 Screenshots:.....	98
Task 6.5 Query 5 Screenshots:.....	105
Task 6.6 Query 6 Screenshots:.....	113
Task 6.7 Query 7 Screenshots:.....	119
Task 6.8 Query 8 Screenshots:.....	126
Task 6.9 Query 9 Screenshots:.....	131
Task 6.10 Query 10 Screenshots:.....	132
Task 6.11 Query 11 Screenshots:.....	133
Task 6.12 Query 12 Screenshots:.....	134
Task 6.13 Query 13 Screenshots:.....	136
Task 6.14 Query 14 Screenshots:.....	138
Task 6.15 Query 15 Screenshots:.....	140
Task 6.15 Query 16 Screenshots:.....	141
Task 6.16 Error Screenshots:	142
Task 6.17 Query 17 Screenshots:.....	143
<i>Task 7: Web database application source program and its execution.....</i>	144
7.1 Web database application source program and screenshots showing successful compilation.....	144
7.2 Screenshots showing the testing of the Web database application.....	158

Task 1: ER Diagram



Task 2: Relational Database Schemas

Underline -> primary key, **Bold** → foreign key

Customer(name, address, category)

Assembly(assembly_id, date_ordered_assembly_details, **name**)

Process(process_id, process_data, **dep_num**, **assembly_id**)

Fit_process(process_id, fit_type)

Paint_process (process_id, paint_type, paint_method)

Cut_process (process_id, cut_type, machine_type,)

Department(dep_num, dep_data)

Job(job_no, start_date, end_date, additional_info, **process_id**, **assembly_id**)

Fit_Job(job_no, fit_labor_time, additional_info)

Paint_Job(job_no, color, vol, paint_labor_time)

Cut_Job(job_no, cut_labor_time, machine_used, time_of_machine_use, material)

Transaction(transaction_no, sup_cost, **job_no**)

Updates(transaction_no, account_no)

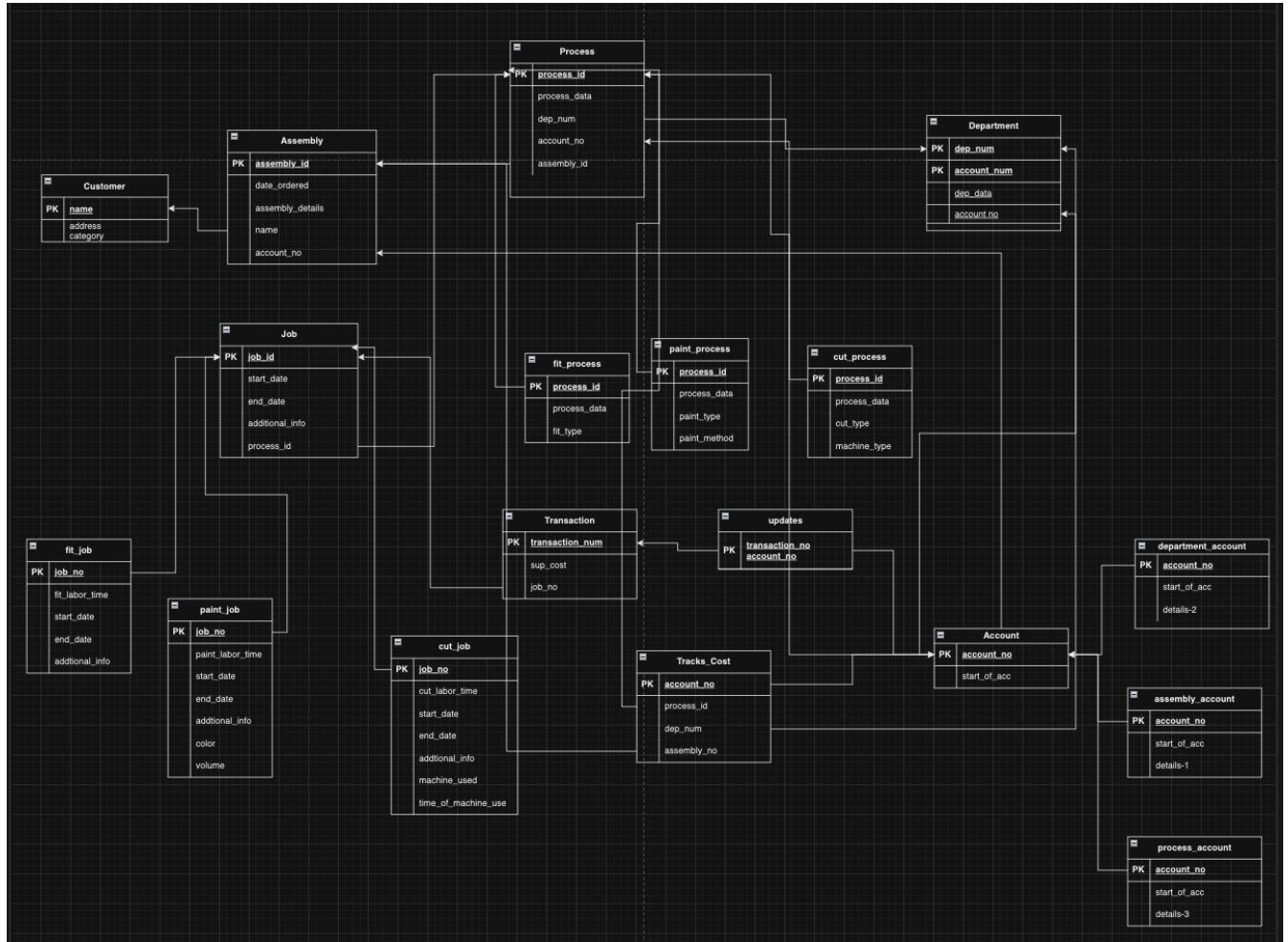
Tracks_Cost(account_no, **process_id**, **dep_num**, **assembly_no**)

Account(account_no, start_of_acc)

Assembly-account(account_no, start_of_acc ,details-1)

Department-account (account_no, start_of_acc,details-2)

Process-account(account_no,start_of_acc,details-3)



Task 3: Storage Structures

Task 3.1:

Table Name	Query# and Type	Search key	Query Frequency	Selected File Organization	Justifications
Customer	#1 Insertion, #12 Range search	#1 – N/A #12 – category number	#1- 30/day #12 – 100/day	Primary Index of the sequential file	Because there is only one insertion procedure call, there's no need to implement hashing or B-/+ Tree organization etc. It's best to use index sequential file organization because of the one, needed range search on query #12.
Department	#2 Insertion, #10 Range Search, #11 Random Search	#2 – N/A #10 – department_no #11 – department_no	#2 – infrequent, #10 – 20/day #11 – (100/day)	Multitable Clustering on Department -Process relationship	This is best file organization because queries #10 and #11 access the department table through the foreign keys of the job and process tables. And, in my jobs table, process is the foreign key. So I would have to join on process to access the department num. Therefore, creating a clustered index on that relationship expedites the retrieval of the

					department details for each process/job.
Assemblies	#4 Insertion, #9 Random Search, #11 Random Search	#4 – N/A #9 – assembly_id #11 – date_ordered	#4 – 40/day #9 – 200/day #11 – 100/day	Static Hashing on assembly_id	Using static hashing on assembly_id is beneficial because we know the amount of data being stored in advance (40/day) and due to the large number of insertions for query #4 and searching efficiently through the specified assembly_id for query #9 and #11. You could index the assemblies' date ordered column, but I think it's better to just use SQL's SORT function on date-ordered and index the process_id in process's table instead.
Process	#3 - Insertion, #11 Random Search, #10 Random Search , #4 – Random Search & Insertion #5 – Random Search	#3 – N/A, #11 – process_id, #10 – process_id/d ep_num, #4 – process_id (for search) #5 – process_id,	#3 – infrequent #11 – (100/day) #10 – 20/day #4 – (40/day) #5 – (10/day)	Dynamic Hashing on process_id	Dynamic hashing is best suitable because we don't know the amount of data for process to be stored in advance.

Paint_Process	#3 – Insertion, #11— random search	#3 – N/A, #11 – process_id,	#3 – infrequent #11 – (100/day)	Dynamic Hashing on process_id.	Because paint process is a subtype of process, it should inherit how the supertype process is being organized.
Cut_Process	#3 – Insertion, #11— random search	#3 – N/A, #11 – process_id,	#3 – infrequent #11 – (100/day)	Dynamic Hashing on process_id.	Because paint process is a subtype of process, it should inherit how the supertype process is being organized.
Fit_Process	#3 – Insertion, #11— random search	#3 – N/A, #11 – process_id,	#3 – infrequent #11 – (100/day)	Dynamic Hashing on process_id	Because paint process is a subtype of process, it should inherit how the supertype process is being organized
Job	#6 – Insertion, #7 – Insertion, #10 – Range Search, #13 – Deletion/ range search #14 Random Search	#6 – N/A, #7 N/A, #10 – job_id #13 – job_id #14 – job_id	#6- 50/day #7 – 50/day #10 – 20/day #13 – 1/month #14 – 1/week	B-Tree Index on job_id	Though there isn't a large frequency of queries using job_id, there are multiple, different procedure calls needing to access this table. Therefore, I think implementing a B-tree index is the best file organization for this table.
Paint_Job	#14 – Range search/insertion #7 -- Insertion	#14 – job_no #7 – N/A	#14 – (1/week) #7 – (50/day)		Same reason as Job.
Cut_Job	#13 – Deletion #7 -- Insertion	#13 – job_no #7 – N/A	#13 – (1/month) #7 – (50/day)		Same reason as Job.

Fit_Job	#7 -- Insertion	#7—N/A	#7—(50/day)		Same reason as Job
Account	#5 – Insertion, #8 – Random Search #9 – Random Search	#5 –N/A #8 – account_id, #9 – assembly_id and account_id	#5 – (10/day) #8 – 50/day #9 –200/day	Primary Index	The queries that use the account table don't have a high enough frequency to use a more complex storage system. Especially because you don't need to retrieve any of the accounts in a specific order.
Assembly_Acc	#5 – Insertion, #8 – Random Search #9 – Random Search	#5 –N/A #8 – account_no, #9 – assembly_id and account_id	#5 – (10/day) #8 – 50/day	Primary Index	Same reason as Account..
Process_Acc	#5 – Insertion, #8 – Random Search	#5 –N/A #8 – account_id,	#5 – (10/day) #8 – 50/day	Primary Index	Same reason as Account.
Department_Acc	#5 – Insertion, #8 – Random Search	#5 –N/A #8 – account_no,	#5 – (10/day) #8 – 50/day	Primary Index	Same reason as Account.
Transactions	#8 – Insertion	#8—N/A	#8 –50/day	Primary index on transaction_no	Transactions table only contains one query with a low frequency. Therefore, it doesn't need to have a complex organization.

Updates	#8 – Insertion	#8—N/A	#8 –50/day	Primary index on transaction_no and account_no	Transactions table only contains one query with a low frequency. Therefore, it doesn't need to have a complex organization.
Tracks_Cost	#5-- Insertion #8 – Range Search	#5 –N/A #8 – account_no	#8 –50/day	Static Hashing	We know the amount of data being entered into this table each day and though the frequency isn't high, this table contains a lot of columns that I think would benefit from static hashing when searching.

Task 3.2

Azure SQL index architecture includes clustered, nonclustered, unique, filtered, clustered columnstore, nonclustered columnstore, has, and memory optimized nonclustered storage structures. There's a normal index which is called row store index that creates a B-tree index

For a non-clustered index, the data will be sorted by a clustered index, however inside the nonclustered index data is stored in a specific order.

Table Name	Query# and Type	Search key	Query Frequency	Selected File Organization	Justifications
Customer	#1 Insertion, #12 Range search	#1 – N/A #12 – category number	#1- 30/day #12 – 100/day	Unique clustered index (primary key) on cname	The database is not accessing nor updating much information from the customer table, so just a unique clustered index clustered on cname is fine. Azure SQL's website states that creating a primary key automatically creates a corresponding unique clustered index.
Department	#2 Insertion, #10 Range Search, #11 Random Search	#2 – N/A #10 – department_no #11 – department_no	#2 – infrequent, #10 – 20/day #11 – (100/day)	Unique clustered index (primary key) on dep_no	Though this table has insertion and search operations, I think a clustered index on the dep_no attribute is fine as the frequency of those queries aren't too high. And the highest frequency query is more focused on the process table instead of directly the department table.
Assemblies	#4 Insertion,	#4 – N/A	#4 – 40/day	Nonclustered unique	Implementing a nonclustered index

	#9 Random Search, #11 Random Search	#9 – assembly_id #11 – date_ordered	#9 – 200/day #11 – 100/day	index on assembly_id	on the primary key is beneficial because of the query types that need access to this column. Query 4,9,11 are all specifically tied to assembly_id.
Process	#3 -Insertion, #11 Random Search, #10 Random Search , #4 –Insertion #5 –Insertion Add querires where you have to go back to process table and update acco_no etc	#3 – N/A, #11 – process_id, #10 – process_id/ dep_num, #4 – process_id #5 – process_id,	#3 – infrequent #11 – (100/day) #10 – 20/day #4 – (40/day) #5 – (10/day)	Nonclustere d index on primary key process_id and nonclustere d index on account_no	Values in this table are constantly have values updated, values that are used in JOIN and WHERE clauses. A nonclustered unique index on process_id will save disk space. And nonclustered index on account_no will help the database be more efficient when needing to access data from not just the process table, but the assemblies and department table too as you have to join on process table to get the assembly and department associated with that process.
Paint_Process	#3 – Insertion,	#3 – N/A,	#3 – infrequent	Primary key constraint	This process has an infrequent frequency and is not accessed at all, therefore just a primary key of process_id
Cut_Process	#3 – Insertion,	#3 – N/A,	#3 – infrequent	Primary key constraint	This process has an infrequent frequency and is not accessed at all, therefore just a primary key of process_id

Fit_Process	#3 – Insertion,	#3 – N/A,	#3 – infrequent	Unique clustered index (primary key) on process_id	This process has an infrequent frequency and is not accessed at all, therefore just a primary key of process_id
Job	#6 – Insertion, #7 – Insertion, #10 – Range Search, #13 – Deletion/range search #14 Random Search	#6 – N/A, #7 N/A, #10 – job_id #13 – job_id #14 – job_id	#6 – 50/day #7 – 50/day #10 – 20/day #13 – 1/month #14 – 1/week	Nonclustered index on primary key and a unique constraint (nonclustered index) on process_id	Values in this table are frequently accessed directly or by using join and where statements, hence using a nonclustered index is best suited. Then a unique constraint on the process id because of its 1-1 relationship with process. For each unique process there's one unique job.
Paint_Job	#14 – Range search/insertion #7 -- Insertion	#14 – job_no #7 – N/A	#14 – (1/week) #7 – (50/day)	Unique primary key constraint	Insertion frequency is low and so is the range search, no need for a complex storage structure. But placing a unique constraint to make sure all jobs are classified uniquely
Cut_Job	#13 – Deletion #7 -- Insertion	#13 – job_no #7 – N/A	#13 – (1/month) #7 – (50/day)	Unique primary key constraint	Insertion frequency is low and so is the range search, no need for a complex storage structure. But placing a unique constraint to make sure all jobs are classified uniquely
Fit_Job	#7 -- Insertion	#7 – N/A	#7 – (50/day)	Unique primary key constraint	Insertion frequency is low and so is the range search, no need for a complex storage structure. But placing a unique

					constraint to make sure all jobs are classified uniquely
Account	#5 – Insertion, #8 – Insertion #9 – Random Search	#5 –N/A #8 – account_id, #9 – assembly_id and account_id	#5 – (10/day) #8 – 50/day #9 – 200/day	Clustered index on primary key account_no	Just a primary key constraint because the queries accessing this table aren't at a high enough frequency to need a clustered index.
Assembly_Acc	#5 – Insertion, #8 – Insertion #9 – Random Search	#5 –N/A #8 – account_id, #9 – assembly_id and account_id	#5 – (10/day) #8 – 50/day #9 – 200/day	Primary key constraint	Though this account needs to be searched a lot, there aren't a lot of accounts needed to be searched through
Process_Acc	#5 – Insertion, #8 – Insertion Random Search	#5 –N/A #8 – account_id,	#5 – (10/day) #8 – 50/day	Primary key constraint	Though this account needs to be searched a lot, there aren't a lot of accounts needed to be searched through
Department_Acc	#5 – Insertion, #8 – Insertion	#5 –N/A #8 – account_id,	#5 –N/A #8 – account_id,	Primary key constraint	Though this account needs to be searched a lot, there aren't a lot of accounts needed to be searched through
Transactions	#8 – Insertion	#8 –N/A	#8 –50/day	Primary index on transaction_no	Transactions table only contains one query with a low frequency. Therefore, it doesn't need to have a complex organization.
Updates	#8 – Insertion	#8 –N/A	#8 –50/day	Primary index on transaction_no and account_no	Transactions table only contains one query with a low frequency. Therefore, it doesn't need to have a complex organization.

Tracks_Cost	#5— Insertion #8— Random Search	#5 –N/A #8 – account_id,	#5 –N/A #8 – account_id,	Nonclustere d Index on primary key account_no	We know the amount of data being entered into this table each day and though the frequency isn't high, this table contains a lot of columns that I think would benefit from a nonclustered index on primary key
-------------	---	--------------------------------	--------------------------------	--	--

3.3 Data Dictionary

Attribute	Type	Constraints
cname	VARCHAR(40)	PK, FK_cname
address	VARCHAR(40)	
category	INT	CHK_category
dep_num	INT	PK, FK_dep_num, FK_dep_tracks
dep_data	VARCHAR(50)	
assembly_no	INT	PK, FK_assembly_no, FK_assembly_tracks
process_id	INT	PK, FK_process_id_paint, FK_process_id_cut FK_process_id_fit FK_process_id_job FK_assembly_no_job FK_process_tracks CHK_Fit_process CHK_Paint_process CHK_Cut_process
process_data	VARCHAR(50)	
paint_type	VARCHAR(20)	
paint_method	VARCHAR(30)	
fit_type	VARCHAR(20)	
cut_type	VARCHAR(20)	
machine_type	VARCHAR(30)	
job_no	INT	PK, FK_job_no_fit, FK_job_no_paint FK_job_no_cut CHK_Fit_Job CHK_Paint_Job CHK_Cut_Job, FK_job_no_transactions
start_date	DATE	
end_date	DATE	CHK_end_date (end_date >= start_date)
additional_info	VARCHAR(100)	
fit_labor_time	INT	
color	VARCHAR(20)	
volume	INT	
paint_labor_time	INT	
cut_labor_time	INT	
machine_used	VARCHAR(20)	
time_of_machine_use	INT	

account_no	INT	PK, FK_account_3, FK_account_2, FK_account_1, FK_account_tracks, FK_account_no_updates
start_of_acc	DATE	
details_3	INT	
details_2	INT	
details_1	INT	
transaction_no	INT	PK, FK_transaction_no_updates
sup_cost	INT	

Task 4: Table Creation in Azure SQL

```
--updates create procedures

DROP TABLE IF EXISTS Customer;
DROP TABLE IF EXISTS Assemblies;
DROP TABLE IF EXISTS Process;
DROP TABLE IF EXISTS Fit_process;
DROP TABLE IF EXISTS Paint_process;
DROP TABLE IF EXISTS Cut_process;
DROP TABLE IF EXISTS Department;
DROP TABLE IF EXISTS Job;
DROP TABLE IF EXISTS Fit_Job;
DROP TABLE IF EXISTS Cut_Job;
DROP TABLE IF EXISTS Paint_Job;
DROP TABLE IF EXISTS Transactions;
DROP TABLE IF EXISTS Updates;
DROP TABLE IF EXISTS Account;
DROP TABLE IF EXISTS Assembly_acc;
DROP TABLE IF EXISTS Department_acc;
DROP TABLE IF EXISTS Process_acc;
DROP TABLE IF EXISTS Tracks_cost;

CREATE TABLE Customer(
    cname VARCHAR(40) PRIMARY KEY,
    address VARCHAR(40),
    category INT,
    CONSTRAINT CHK_category CHECK (category BETWEEN 1 AND 10),
);

CREATE TABLE Department(
    dep_num INT PRIMARY KEY,
    dep_data VARCHAR (50),
);

CREATE TABLE Assemblies(
    assembly_no INT PRIMARY KEY NONCLUSTERED,
```

```

date_ordered DATE,
assembly_details VARCHAR (50),
cname VARCHAR (40),
CONSTRAINT FK_cname FOREIGN KEY (cname) REFERENCES Customer,
);

--new process table
CREATE TABLE Process(
process_id INT PRIMARY KEY NONCLUSTERED,
process_data VARCHAR(50),
dep_num INT NOT NULL,
assembly_no INT,
CONSTRAINT FK_assembly_no FOREIGN KEY (assembly_no) REFERENCES Assemblies,
CONSTRAINT FK_dep_num FOREIGN KEY (dep_num) REFERENCES Department,
);

--specialized process tables:
CREATE TABLE Paint_process (
process_id INT PRIMARY KEY,
paint_type VARCHAR(20),
paint_method VARCHAR(30),
CONSTRAINT FK_process_id_paint FOREIGN KEY (process_id) REFERENCES Process ON DELETE NO
ACTION,
CONSTRAINT CHK_Paint_Process_ProcessID UNIQUE (process_id),
);

CREATE TABLE Fit_process (
process_id INT PRIMARY KEY,
fit_type VARCHAR (20),
CONSTRAINT FK_process_id_fit FOREIGN KEY (process_id) REFERENCES Process ON DELETE NO ACTION,
CONSTRAINT CHK_Fit_Process_ProcessID UNIQUE (process_id),
);

CREATE TABLE Cut_process (
process_id INT PRIMARY KEY,
cut_type VARCHAR(20),
machine_type VARCHAR(30),
);

```

```

CONSTRAINT FK_process_id_cut FOREIGN KEY (process_id) REFERENCES Process ON DELETE NO
ACTION,
CONSTRAINT CHK_Cut_Process_ProcessID UNIQUE (process_id),
);

CREATE TABLE Job(
job_no INT,
start_date DATE,
end_date DATE,
additional_info VARCHAR (100),
process_id INT NOT NULL UNIQUE,
assembly_no INT NOT NULL,
CONSTRAINT PK_job PRIMARY KEY NONCLUSTERED (job_no),
CONSTRAINT FK_process_id_job FOREIGN KEY (process_id) REFERENCES Process,
CONSTRAINT FK_assembly_no_job FOREIGN KEY (assembly_no) REFERENCES Assemblies,
CONSTRAINT CHK_end_date CHECK (end_date >= start_date),
);

CREATE TABLE Fit_Job(
job_no INT PRIMARY KEY,
fit_labor_time INT,
CONSTRAINT FK_job_no_fit FOREIGN KEY (job_no) REFERENCES Job,
CONSTRAINT CHK_Fit_Job UNIQUE (job_no),
);

CREATE TABLE Paint_Job(
job_no INT PRIMARY KEY,
color VARCHAR (20),
volume INT,
paint_labor_time INT,
CONSTRAINT FK_job_no_paint FOREIGN KEY (job_no) REFERENCES Job,
CONSTRAINT CHK_Paint_Job UNIQUE (job_no),
);

CREATE TABLE Cut_Job(
job_no INT PRIMARY KEY,

```

```

cut_labor_time INT,
machine_used VARCHAR(20),
time_of_machine_use INT,
material VARCHAR (20),
CONSTRAINT FK_job_no_cut FOREIGN KEY (job_no) REFERENCES Job,
CONSTRAINT CHK_Cut_Job UNIQUE (job_no),
);

CREATE TABLE Account(
account_no INT IDENTITY (1,1) PRIMARY KEY, --IDENTITY (1,1)
start_of_acc DATE,
);

CREATE TABLE Process_acc(
account_no INT PRIMARY KEY,
start_of_acc DATE,
details_3 INT,
CONSTRAINT FK_account_no3 FOREIGN KEY (account_no) REFERENCES Account,
);

CREATE TABLE Assembly_acc(
account_no INT PRIMARY KEY,
start_of_acc DATE,
details_1 INT,
CONSTRAINT FK_account_no1 FOREIGN KEY (account_no) REFERENCES Account,
);

CREATE TABLE Department_acc(
account_no INT PRIMARY KEY,
start_of_acc DATE,
details_2 INT,
CONSTRAINT FK_account_no2 FOREIGN KEY (account_no) REFERENCES Account,
);

CREATE TABLE Tracks_Cost(
account_no INT PRIMARY KEY NONCLUSTERED,
process_id INT,
dep_num INT,
assembly_no INT,

```

```
CONSTRAINT FK_account_tracks FOREIGN KEY (account_no) REFERENCES Account,  
CONSTRAINT FK_assembly_tracks FOREIGN KEY (assembly_no) REFERENCES Assemblies,  
CONSTRAINT FK_process_tracks FOREIGN KEY (process_id) REFERENCES Process,  
CONSTRAINT FK_dep_tracks FOREIGN KEY (dep_num) REFERENCES Department,  
);
```

```
CREATE TABLE Transactions(  
    transaction_no INT PRIMARY KEY,  
    sup_cost INT,  
    job_no INT,  
    CONSTRAINT FK_job_no_transactions FOREIGN KEY (job_no) REFERENCES Job,  
);
```

```
CREATE TABLE Updates (  
    transaction_no INT,  
    account_no INT,  
    CONSTRAINT PK_updates PRIMARY KEY (transaction_no, account_no),  
    CONSTRAINT FK_transaction_no_updates FOREIGN KEY (transaction_no) REFERENCES Transactions,  
    CONSTRAINT FK_account_no_updates FOREIGN KEY(account_no) REFERENCES Account,  
);
```

```

1  SELECT * FROM Customer;
2  SELECT * FROM Department;
3  SELECT * FROM Process;
4  SELECT * FROM Fit_Process;
5  SELECT * FROM Cut_Process;
6  SELECT * FROM Paint_Process;
7  SELECT * FROM Assemblies;
8  SELECT * FROM Job;
9  SELECT * FROM Fit_Job;
10 SELECT * FROM Cut_Job;
11 SELECT * FROM Paint_Job;
12 SELECT * FROM Transactions;
13 SELECT * FROM Updates;
14 SELECT * FROM Tracks_Cost;
15 SELECT * FROM Account;
16 SELECT * FROM Process_acc;
17 SELECT * FROM Department_acc;
18 SELECT * FROM Assembly_acc;

```

Results Messages

	cname	address	category
--	-------	---------	----------

	dep_num	dep_data
--	---------	----------

	process_id	process_data	dep_num	assembly_no
--	------------	--------------	---------	-------------

	process_id	fit_type
--	------------	----------

	process_id	cut_type	machine_type
--	------------	----------	--------------

	process_id	paint_type	paint_method
--	------------	------------	--------------

assembly_no	date_ordered	assembly_detail	cname
-------------	--------------	-----------------	-------

job_no	start_date	end_date	additional_info	process_id	assembly_no
--------	------------	----------	-----------------	------------	-------------

job_no	fit_labor_time
--------	----------------

job_no	cut_labor_time	machine_used	time_of_machin...	material
--------	----------------	--------------	-------------------	----------

job_no	color	volume	paint_labor_time
--------	-------	--------	------------------

transaction_no	sup_cost	job_no
----------------	----------	--------

transaction_no	account_no
----------------	------------

account_no	process_id	dep_num	assembly_no
------------	------------	---------	-------------

account_no	process_id	dep_num	assembly_no
account_no	start_of_acc		
account_no	start_of_acc	details_3	
account_no	start_of_acc	details_2	
account_no	start_of_acc	details_1	

Task 5:

5.1 SQL Statements and Transact SQL Stored procedures (queries 1-15) :

For Query 1 (SQL Statements):

```
INSERT INTO Customer (cname, address, category)
VALUES
(?, ?, ?);
```

For Query 2 (SQL Statements):

```
INSERT INTO Department(dep_num, dep_data)
VALUES
(?, ?);
```

For Query 3 (SQL Statements):

```
INSERT INTO Process (process_id, process_data, dep_num)
```

```
VALUES
```

```
(?, ?, ?);
```

```
INSERT INTO Paint_process (process_id, paint_type, paint_method)
```

```
VALUES
```

```
(?, ?, ?);
```

```
INSERT INTO Fit_process (process_id, fit_type)
```

```
VALUES
```

```
(?, ?);
```

```
INSERT INTO Cut_process (process_id, cut_type, machine_type)
```

```
VALUES
```

```
(?, ?, ?);
```

For Query 4 (SQL Statements):

```
INSERT INTO Assemblies (assembly_no, date_ordered, assembly_details, cname)
VALUES
(?, ?, ?, ?);
```

```
UPDATE Process
SET assembly_no = ?
WHERE process_id = ?
```

For Query 5 (SQL Statements):

```
INSERT INTO Account (start_of_acc)
```

```
VALUES
```

```
(?);
```

```
INSERT INTO Process_acc(account_no, start_of_acc, details_3)
```

```
VALUES
```

```
(?, ?, ?);
```

```
INSERT INTO Assembly_acc(account_no, start_of_acc, details_1)
```

```
VALUES
```

```
(?, ?, ?);
```

```
INSERT INTO Department_acc(account_no, start_of_acc, details_2)
```

```
VALUES
```

```
(?, ?, ?);
```

```
INSERT INTO Tracks_Cost(account_no, process_id)
```

```
VALUES
```

```
(?,?);
```

```
INSERT INTO Tracks_Cost(account_no, dep_num)
```

```
VALUES (?, ?);
```

```
INSERT INTO Tracks_Cost(account_no, assembly_no)
```

```
VALUES (?, ?);
```

For Query 6 (TSP):

```
DROP PROCEDURE IF EXISTS query_6
```

```

GO
CREATE PROCEDURE query_6
    @job_no INT,
    @start_date DATE,
    @details VARCHAR (100),
    @process_id INT,
    @assembly_no INT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Process AP
        WHERE AP.assembly_no = @assembly_no AND AP.process_id = @process_id
    )
    BEGIN
        INSERT INTO Job (job_no, start_date, additional_info, process_id, assembly_no)
        VALUES
        (@job_no, @start_date, @details, @process_id, @assembly_no);
    END
    ELSE
    BEGIN
        THROW 50000, Error: The assembly number entered is not associated with the process_id.', 1;
    RETURN
    END
END;
---Executing
EXEC @job_no =?, @start_date=? , @details = ?, @process_id =?, @assembly_no =?;
```

For Query 7 (TSP):

(TSP for inserting fit job & updating end date)

```

DROP PROCEDURE IF EXISTS query_7A;
GO
CREATE PROCEDURE query_7A
    @job_no INT,
    @end_date DATE,
```

```

@fit_labor_time INT
AS
BEGIN
IF EXISTS (
    SELECT 1
    FROM Job J
    LEFT JOIN Process P ON J.process_id = P.process_id
    LEFT JOIN Fit_Process FP ON P.process_id = FP.process_id
    WHERE J.job_no = @job_no AND FP.process_id IS NOT NULL
)
BEGIN
    UPDATE Job
    SET end_date = @end_date
    WHERE job_no = @job_no;

    INSERT INTO Fit_Job(job_no, fit_labor_time)
    VALUES(@job_no, @fit_labor_time);
END
ELSE
BEGIN
    THROW 50000, Error: The specified process_id is not associated with a Fit process. ', 1;
    RETURN
END
END;

--Executing
GO
EXEC query_7A @job_no = ?, @end_date = ?, @fit_labor_time = ?;
```

(TSP for inserting paint job & updating end date)

```
DROP PROCEDURE IF EXISTS query_7B
```

```
GO
```

```
CREATE PROCEDURE query_7B
```

```

    @job_no INT,
    @end_date DATE,
    @paint_labor_time INT,
```

```

@color VARCHAR (20),
@volume INT
AS
BEGIN
IF EXISTS (
SELECT 1
FROM Job J
LEFT JOIN Process P ON J.process_id = P.process_id
LEFT JOIN Paint_Process PP ON P.process_id = PP.process_id
WHERE J.job_no = @job_no AND PP.process_id IS NOT NULL
)
BEGIN
UPDATE Job
SET end_date = @end_date
WHERE job_no = @job_no;

INSERT INTO Paint_Job(job_no, color, volume, paint_labor_time)
VALUES(@job_no, @color, @volume, @paint_labor_time)
END
ELSE
BEGIN
THROW 50000, Error: The specified process_id is not associated with a Paint process. ', 1;
RETURN
END
END;
--Executing
GO
EXEC query_7B @job_no = ?, @end_date = ?, @paint_labor_time = ?, @color = ?, @volume = ?;

```

(TSP for inserting cut job & updating end date)

```

DROP PROCEDURE IF EXISTS query_7C
GO
CREATE PROCEDURE query_7C

```

```

@job_no INT,
@end_date DATE,
@cut_labor_time INT,
@machine_used VARCHAR (20),
@time_of_machine_use INT,
@material VARCHAR (40)

AS
BEGIN
IF EXISTS (
SELECT 1
FROM Job J
LEFT JOIN Process P ON J.process_id = P.process_id
LEFT JOIN Cut_Process CP ON P.process_id = CP.process_id
WHERE J.job_no = @job_no AND CP.process_id IS NOT NULL
)
BEGIN
UPDATE Job
SET end_date = @end_date
WHERE job_no = @job_no;

INSERT INTO Cut_Job(job_no, cut_labor_time, machine_used, time_of_machine_use, material)
VALUES(@job_no, @cut_labor_time, @machine_used, @time_of_machine_use, @material)
END
ELSE
BEGIN
THROW 50000, Error: The specified process_id is not associated with a Cut process. ', 1;
RETURN
END
END;
--Executing
GO
EXEC query_7C @job_no = ?, @cut_labor_time = ?, @machine_used = ?, @time_of_machine_use = ?, @material
=?;
```

For Query 8 (TSP):

(TSP for inserting into transaction table and updates table)

```

DROP PROCEDURE IF EXISTS query8_insert
GO
CREATE PROCEDURE query8_insert
    @process_acc_no INT,
    @assembly_acc_no INT,
    @dep_acc_no INT,
    @transaction_no INT,
    @sup_cost INT
AS
BEGIN
    INSERT INTO Transactions(transaction_no, sup_cost)
    VALUES
        (@transaction_no, @sup_cost);

    INSERT INTO Updates(transaction_no, account_no)
    VALUES
        (@transaction_no,@process_acc_no),
        (@transaction_no,@assembly_acc_no),
        (@transaction_no,@dep_acc_no);
END;
-- 
GO
EXEC QUERY8_INSERT @process_acc_no = ?, @assembly_acc_no = ?, @dep_acc_num = ?, @transaction_no =
?, @sup_cost = ?;
```

(TSP for updating Process Account Table)

```

DROP PROCEDURE IF EXISTS query_8A
GO
CREATE PROCEDURE query_8A
    @process_acc_no INT,
    @transaction_no INT,
    @sup_cost INT
```

```

AS
BEGIN
    UPDATE Process_acc
    SET details_3 = details_3 + @sup_cost
    FROM Process_acc pa
    JOIN Updates u ON pa.account_no= u.account_no
    WHERE pa.account_no = @process_acc_no AND u.transaction_no = @transaction_no
END;
---

GO
EXEC query_8A @process_acc_no = ?, @transaction_no = ?, @sup_cost = ?;
(TSP for updating Assembly Account Table)
Checks if the assembly_acc is associated with the process id
DROP PROCEDURE IF EXISTS query_8B
GO
CREATE PROCEDURE query_8B
    @assembly_acc_no INT,
    @transaction_no INT,
    @sup_cost INT,
    @process_acc_no INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Tracks_Cost WHERE account_no = @assembly_acc_no)
        BEGIN
            -- Assembly not found, raise an error
            THROW 50000, 'Assembly Account not found.', 1;
        RETURN
    END

    DECLARE @assembly_no INT;
    DECLARE @process_id INT;
    SET @process_id= (SELECT T.process_id FROM Tracks_Cost T WHERE T.account_no= @process_acc_no);
    SET @assembly_no = (SELECT assembly_no FROM Process WHERE process_id = @process_id);

    UPDATE Assembly_acc
    SET details_1 = details_1 + @sup_cost
    FROM Assembly_acc A

```

```

JOIN Tracks_Cost T ON A.account_no = T.account_no
WHERE A.account_no = @assembly_acc_no
AND T.assembly_no = @assembly_no;
END;
--GO
EXEC query_8B @assembly_acc_no = ?, @transaction_no = ?, @sup_cost = ?, @process_acc_no = ?;

```

(TSP for updating Department Account Table)
Checks if the department_acc is associated with the process id

```

DROP PROCEDURE IF EXISTS query_8C
GO
CREATE PROCEDURE query_8C
    @dep_acc_no INT,
    @transaction_no INT,
    @sup_cost INT,
    @process_acc_no INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Tracks_Cost WHERE account_no = @dep_acc_no)
    BEGIN
        -- Assembly not found, raise an error
        THROW 50000, 'Department Account not found.', 1;
    END
    RETURN
END
DECLARE @dep_num INT;
DECLARE @process_id INT;
SET @process_id = (SELECT T.process_id FROM Tracks_Cost T WHERE T.account_no = @process_acc_no);
SET @dep_num = (SELECT dep_num FROM Process WHERE process_id = @process_id);

```

```

UPDATE Department_acc
SET details_2 = details_2 + @sup_cost
FROM Department_acc A
JOIN Tracks_Cost T ON T.account_no = A.account_no
WHERE T.account_no = @dep_acc_no
AND T.dep_num = @dep_num;
END;
--
GO
EXEC query_8C @dep_acc_no = ?, @transaction_no = ?, @sup_cost = ?, @process_acc_no = ?;

```

For Query 9 (TSP):

```

DROP PROCEDURE IF EXISTS query_9
GO
CREATE PROCEDURE query_9
    @assembly_no INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Assemblies WHERE assembly_no = @assembly_no)
    BEGIN
        -- Assembly not found, raise an error
        THROW 50000, 'No assembly with that account number.', 1;
    RETURN
    END

    IF NOT EXISTS (SELECT 1 FROM Tracks_Cost WHERE assembly_no = @assembly_no)
    BEGIN

```

```

--assembly doesn't have the account
THROW 50000, No account with that assembly id.', 1;

END

DECLARE @total_cost INT;

SELECT SUM(details_1) AS total_cost
FROM Assembly_Acc S
JOIN Tracks_Cost T ON S.account_no = T.account_no
WHERE T.assembly_no = @assembly_no;
END;
-- 
GO
EXEC query_9 @assembly_no = ?;
```

For Query 10 (TSP):

```

DROP PROCEDURE IF EXISTS query_10
GO
CREATE PROCEDURE query_10
    @dep_num INT,
    @job_start DATE,
    @job_end DATE
AS
BEGIN
    DECLARE @total_labor_time INT;
    SELECT
        d.dep_num,
        SUM(ISNULL(fj.fit_labor_time, 0) + ISNULL(pj.paint_labor_time, 0) + ISNULL(cj.cut_labor_time, 0))AS
total_labor_time
    FROM
        Department d
    JOIN
        Process p ON d.dep_num = p.dep_num
    LEFT JOIN
        Job j ON p.process_id = j.process_id
    LEFT JOIN
```

```

Fit_Job fj ON j.job_no = fj.job_no
LEFT JOIN
    Paint_Job pj ON j.job_no = pj.job_no
LEFT JOIN
    Cut_Job cj ON j.job_no = cj.job_no
WHERE
    d.dep_num = @dep_num AND j.end_date BETWEEN @job_start AND @job_end
GROUP BY
    d.dep_num;
END;
--EXECUTE
GO
EXEC query_10 @dep_num = ?, @job_start = ?, @job_end = ?;

```

For Query 11(TSP):

```

DROP PROCEDURE IF EXISTS query_11
GO
CREATE PROCEDURE query_11
    @assembly_no INT
AS
BEGIN
    -- Check if assembly_no exists
    IF NOT EXISTS (SELECT 1 FROM Assemblies WHERE assembly_no = @assembly_no)
    BEGIN
        -- Assembly not found, raise an error
        THROW 50000, 'Assembly not found.', 1;
    END
    RETURN
END
SELECT
    p.process_id,
    p.process_data,
    a.date_ordered,
    d.dep_num,
    d.dep_data,
    CASE
        WHEN fp.process_id IS NOT NULL THEN 'Fit Process'

```

```

WHEN pp.process_id IS NOT NULL THEN 'Paint Process'
WHEN cp.process_id IS NOT NULL THEN 'Cut Process'
END AS process_type
FROM
Assemblies a
JOIN
Process p ON a.assembly_no = p.assembly_no
JOIN
Department d ON p.dep_num = d.dep_num
LEFT JOIN
Fit_process fp ON p.process_id = fp.process_id
LEFT JOIN
Paint_process pp ON p.process_id = pp.process_id
LEFT JOIN
Cut_process cp ON p.process_id = cp.process_id
WHERE a.assembly_no = @assembly_no
ORDER BY a.date_ordered;
END;
--Executing
GO
EXEC query_11 @assembly_no = @assembly_no;

```

For Query 12(TSP):

```

--Query 12
DROP PROCEDURE IF EXISTS query_12
GO
CREATE PROCEDURE query_12
    @lower_limit INT,
    @upper_limit INT
AS
BEGIN
    SELECT * FROM Customer
    WHERE
        category BETWEEN @lower_limit AND @upper_limit
    ORDER BY
        cname;
END;

```

```
--Executing
GO
EXEC query_12 @lower_limit = ?, @upper_limit = ?;
```

For Query 13 (TSP):

```
DROP PROCEDURE IF EXISTS query_13
GO
CREATE PROCEDURE query_13
    @lower_limit INT,
    @upper_limit INT
AS
BEGIN
    BEGIN TRANSACTION;
    -- Delete entries from Cut_Job subtype
    BEGIN TRY
        DECLARE @deletedJobNos TABLE (job_no INT);
        DELETE Cut_Job
        OUTPUT deleted.job_no INTO @deletedJobNos
        --FROM Cut_Job cj
        --LEFT JOIN Job j ON cj.job_no = j.job_no
        WHERE job_no BETWEEN @lower_limit AND @upper_limit;

        -- Delete entries from Job table
        DELETE FROM Job
        WHERE job_no IN (SELECT job_no FROM @deletedJobNos);

        COMMIT;
    END TRY
    BEGIN CATCH
        -- An error occurred, rollback the transaction
    END CATCH

```

```

ROLLBACK TRANSACTION;
-- Optionally, raise or log an error
THROW;
END CATCH
END;
-- 
GO
EXEC query_13 @lower_limit = ?, @upper_limit = ?;

```

```

--query 13 show remaining cut jobs
DROP PROCEDURE IF EXISTS query_13_DJ
GO
CREATE PROCEDURE query_13_DJ
AS
BEGIN
    SELECT * FROM Cut_Job;
END;
-- 
GO
EXEC query_13_DJ;

```

For Query 14 (TSP):

```

DROP PROCEDURE IF EXISTS query_14;
GO
CREATE PROCEDURE query_14
    @color VARCHAR(20),
    @job_id INT
AS
BEGIN
    UPDATE pj
    SET pj.color = @color
    FROM Paint_Job pj
    JOIN Job J ON pj.job_no = J.job_no
    WHERE J.job_no = @job_id;
END;
-- 
GO

```

```
EXEC query_14 @color = ?, @job_id = ?;
```

For Query 15:

```
CREATE PROCEDURE ImportCustomers
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION;
        DECLARE @Name NVARCHAR(255), @Address NVARCHAR(255), @category INT;

        -- Cursor to iterate over records
        DECLARE importCursor CURSOR FOR
        SELECT name, address, category FROM #TempCustomer;

        OPEN importCursor;
        FETCH NEXT FROM importCursor INTO @Name, @Address, @category;

        WHILE @@FETCH_STATUS = 0
        BEGIN
            INSERT INTO Customer (cname, address, category)
            VALUES (@Name, @Address, @category);
            FETCH NEXT FROM importCursor INTO @Name, @Address, @category;
        END

        CLOSE importCursor;
        DEALLOCATE importCursor;

        COMMIT;
    END TRY
    BEGIN CATCH
        ROLLBACK;
        THROW;
    END CATCH
END;
```

For Query 16:

```
SELECT cname, address, category
FROM Customer
WHERE category BETWEEN ? AND ?
ORDER BY cname
```

5.2 Java Source Program

```
import java.sql.CallableStatement;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
```

```
import java.io.FileWriter;
import java.io.IOException;
import java.sql.*;
import java.sql.Connection;
import java.sql.Statement;
import java.time.LocalDate;
import java.util.Scanner;
//import java.util.UUID;
import java.sql.ResultSet;
import java.sql.SQLWarning;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class sample {

    // Database credentials
    final static String HOSTNAME = "<schn0050>-sql-server.database.windows.net";
    final static String DBNAME = "schn0050";
    final static String USERNAME = "<schn0050>";
    final static String PASSWORD = "<Sofiase10>";

    // Database connection string
    final static String URL =
String.format("jdbc:sqlserver://schn0050.database.windows.net:1433;database=schn0050;user=" +
    "schn0050@schn0050;password={Sofiase10};encrypt=true;trustServerCertificate=false;hostNameInCertificate=" +
    "*.database.windows.net;loginTimeout=30;");

    // Query templates
    final static String QUERY_TEMPLATE_1 = "INSERT INTO Customer (cname, address, category) " +
```

```

    "VALUES (?, ?, ?);";

final static String QUERY_TEMPLATE_2 = "INSERT INTO Department(dep_num, dep_data) "
+
    "VALUES (?, ?);";

private static String QUERY_TEMPLATE_3 = "INSERT INTO Process(process_id,
process_data, dep_num)" +
    "VALUES (?,?,?);";

final static String QUERY_TEMPLATE_3_A = "INSERT INTO
Fit_process(process_id,fit_type)" +
    "VALUES
(?,?);";

final static String QUERY_TEMPLATE_3_B = "INSERT INTO Paint_process(process_id,
paint_type, paint_method)" +
    "VALUES (?,?,?);";

final static String QUERY_TEMPLATE_3_C = "INSERT INTO Cut_process(process_id,
cut_type, machine_type)" +
    "VALUES (?,?,?);";

final static String QUERY_6 = "INSERT INTO Job(job_no, start_date, process_id,
assembly_no) VALUES (?,?,?,?,?);";

```

```

final static String insertAccount= "INSERT INTO Account (account_no, start_of_acc) " +
                                " VALUES (?,?);" ;

// User input prompt//
final static String PROMPT=

    "\nPlease select one of the options below: \n" +
    "1) Enter a new customer. \n" +
    "2) Enter a new department. \n" +
    "3) Enter new process_id and its department together with its type and info relevant to
the type. \n" +
    "4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and
dateordered and associate it with one or more processes. \n" +
    "5) Create a new account and associate it with the process, assembly, or department to
which it is applicable. \n" +
    "6) Enter a new job, given its job-no, assembly-id, process-id, and date the job
commenced. \n" +
    "7) At the completion of a job, enter the date it completed and the information relevant to
the type of job. \n" +
    "8) Enter a transaction-no and its sup-cost and update all the costs (details) of the
affected accounts by adding sup-cost to their current values of details. \n" +
    "9) Retrieve the total cost incurred on an assembly-id. \n" +
    "10) Retrieve the total labor time within a department for jobs completed in the
department during a given date. \n" +
    "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. \n" +
    "12) Retrieve the customers (in name order) whose category is in a given range. \n" +
    "13) Delete all cut-jobs whose job-no is in a given range. \n" +
    "14) Change the color of a given paint job. \n" +
    "15) Import: enter new customers from a data file until the file is empty. \n" +

```

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

"17) Exit!";

```
public static void main(String[] args) throws Exception {
```

```
    System.out.println("Welcome to the job-shop accounting system!");
```

```
    final Scanner sc = new Scanner(System.in); // Scanner is used to collect the user input
```

```
    String option = ""; // Initialize user option selection as nothing
```

```
    while (!option.equals("17")) { // As user for options until option 3 is selected
```

```
        System.out.println(PROMPT); // Print the available options
```

```
        option = sc.next(); // Read in the user option selection
```

```
        switch (option) { // Switch between different options
```

```
            case "1":
```

```
                // Collect the new customer data from the user...
```

```
                System.out.println("Please enter the customer name: ");
```

```
                sc.nextLine();
```

```
                final String cname = sc.nextLine(); // Read in the user input of customer name
```

```
                System.out.println("Please enter the customer's address: ");
```

```
                final String address = sc.nextLine(); // Read in the user input of customer address
```

```
                System.out.println("Please enter the customer's category, a #1-10:");
```

```
                final int category = sc.nextInt(); // Read in user input of customer category
```

```

System.out.println("Connecting to the database...");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL)) {
    try (
        final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_1) { //calling option_1 query
        statement.setString(1, cname); //inserting data
        statement.setString(2, address);
        statement.setInt(3, category);

    System.out.println("Dispatching the query...");
    //executeUpdate is for INSERT, DELETE, and UPDATE statements
    statement.executeUpdate(); //want to insert values into query first

    //close statements
    statement.close();
    connection.close();

    }
}

break;
case "2":
    // Collect the new department data from the user...
    System.out.println("Please enter the Department number:");
    final int dep_num = sc.nextInt();

    System.out.println("Please enter Department data, if none type NULL:");
}

```

```
sc.nextLine();

final String dep_data = sc.nextLine();

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

// Get a database connection and prepare a query statement

try (final Connection connection = DriverManager.getConnection(URL))

{

    try (

        final PreparedStatement statement =

connection.prepareStatement(QUERY_TEMPLATE_2)

    {

        statement.setInt(1, dep_num);

        statement.setString(2, dep_data);

        System.out.println("Dispatching the query...");

        //execute the query

        statement.executeUpdate();

        //close statements

        statement.close();

        connection.close();

    }

}

break;

case "3":



    // Collect the new process data from the user...
```

```

System.out.println("Please enter the Process ID:");
final int process_id = sc.nextInt(); // Read in the user input of process id

System.out.println("Please enter Process data, if none type NULL:");
sc.nextLine();
final String process_data = sc.nextLine();

System.out.println("Please enter the Department ID you want to associate with
this new process:");
final int dep_num_for_process = sc.nextInt();

System.out.println("Connecting to the database...");
// Get a database connection and prepare a query statement
try (Connection connection = DriverManager.getConnection(URL))
{
    try (
        final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_3))
    {
        //set index with specified information
        statement.setInt(1, process_id); //inserting data
        statement.setString(2, process_data);
        statement.setInt(3, dep_num_for_process);

        System.out.println("Dispatching the query...");
    }
}

```

```

        //execute the query
        statement.executeUpdate();
        //close the connection
        statement.close();
    }

//for specific process type
System.out.println("Please enter the process type (fit, paint, cut)");
sc.nextLine();
final String process_type = sc.nextLine();

//if condition to check user input of type & inserting type in their tables
if(process_type.equalsIgnoreCase("fit"))
{
    System.out.println("Please enter information on which fit type:");
    final String fit_type = sc.nextLine();

    //call function
    insertFitData(connection, process_id, fit_type);

} else if (process_type.equalsIgnoreCase("paint"))
{
    System.out.println("Please enter information on which paint
type:");
    final String paint_type = sc.nextLine();

    System.out.println("Please enter information on painting
method:");
}

```

```

final String paint_method = sc.nextLine();

//call function
insertPaintData(connection, process_id, paint_type,
paint_method);

} else if (process_type.equalsIgnoreCase("cut"))
{
    System.out.println("Please enter information on which cut
type:");

final String cut_type = sc.nextLine();

System.out.println("Please enter information on machine
type:");

final String machine_type = sc.nextLine();

//call function
insertCutData(connection, process_id, cut_type, machine_type);

} else {
    System.out.println("Failed to retrieve process type, please re-enter");
}

connection.close();
}

break;
case "4":
    // Collect the new assembly data from user...

System.out.println("Please enter the Assembly number:");
final int assembly_no = sc.nextInt();

```

```

System.out.println("Please enter the date this assembly was ordered (YEAR-MONTH-DATE): ");
sc.nextLine();
final String date_ordered = sc.nextLine();

System.out.println("Please enter the assembly details: ");
final String assembly_details = sc.nextLine();

System.out.println("Please enter the name of the customer who ordered: ");
final String cname_assembly = sc.nextLine();

System.out.println("Connecting to the database...");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{
    String stopKeyword = "stop";
    String assemblyQuery = "INSERT INTO Assemblies (assembly_no, date_ordered, assembly_details, cname) VALUES (?,?,?,?)";
    try (
        final PreparedStatement statement =
connection.prepareStatement(assemblyQuery)) //calling option_2 query
    {
        //set index with specified information
        statement.setInt(1, assembly_no);
        statement.setDate(2, java.sql.Date.valueOf(date_ordered));
        statement.setString(3, assembly_details);
    }
}

```

```

statement.setString(4, cname_assembly);

statement.executeUpdate();

while(true) {
    System.out.println("Please enter the process id you want to
associate this assembly with, type 'stop' to stop entering a new process: ");
    final String process_id_for_assembly = sc.nextLine();

    // Check if the user wants to stop
    if (process_id_for_assembly.equalsIgnoreCase(stopKeyword)) {
        System.out.println("Exiting the query.");
        break; // Exit the loop
    }

    //converting back to int
    final int ass_process_id = Integer.parseInt(process_id_for_assembly);

    String updateQueryP = "UPDATE Process SET assembly_no = ?
WHERE process_id = ?";

    try (PreparedStatement updatepreparedStatement =
connection.prepareStatement(updateQueryP)) {
        updatepreparedStatement.setInt(1, assembly_no);
        updatepreparedStatement.setInt(2, ass_process_id);
        updatepreparedStatement.executeUpdate();
    }

}

```

```
System.out.println("Dispatching the query...");

//close statements
statement.close();
connection.close();
}

}

break;
case "5":
// Collect the new account data from the user...
System.out.println("The database will generate a unique account number.
\n");

//System.out.println("Please enter the new account_no: ");
//final int account_no = sc.nextInt();

System.out.println("Do you wish to create a Process, Department, or
Assembly account? ");
sc.nextLine();
final String type_of_account = sc.nextLine();

System.out.println("Connecting to the database...");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{

String insertQuery = "INSERT INTO Account (start_of_acc) VALUES (?);"
```

```

try (PreparedStatement preparedStatement =
connection.prepareStatement(insertQuery, Statement.RETURN_GENERATED_KEYS)) {

    LocalDate currentDate = LocalDate.now();

    // Convert LocalDate to java.sql.Date
    java.sql.Date date_established = java.sql.Date.valueOf(currentDate);

    preparedStatement.setDate(1, date_established);
    preparedStatement.executeUpdate();

    try (ResultSet generatedKeys =
preparedStatement.getGeneratedKeys()) {
        int account_no;
        if (generatedKeys.next()) {
            account_no = generatedKeys.getInt(1);

            if(type_of_account.equalsIgnoreCase("process"))
            {

                //calling query to insert into specific
                account

                //asking user for department number they
                wish to associate the new account with
                System.out.println("Please enter the
process id you wish to associate this account with: ");
                final int account_process_id =
sc.nextInt();
            }
        }
    }
}

```

```

//calling query to update account_no in
tracks table

String updateQueryP = "INSERT INTO
Tracks_Cost(account_no, process_id) VALUES ( ?,? )";
try (PreparedStatement
updatepreparedStatement = connection.prepareStatement(updateQueryP)) {
    updatepreparedStatement.setInt(1,
account_no);
    updatepreparedStatement.setInt(2,
account_process_id);

updatepreparedStatement.executeUpdate();

    updatepreparedStatement.close();
}

String insertProcessAcc = "INSERT INTO
Process_acc(account_no, details_3) VALUES( ?,? )";
try (PreparedStatement
updatepreparedStatement = connection.prepareStatement(insertProcessAcc)) {
    updatepreparedStatement.setInt(1,
account_no);
    updatepreparedStatement.setInt(2, 0);

updatepreparedStatement.executeUpdate();

    System.out.println("Completed. ");
    updatepreparedStatement.close();
}

```

```

} else

if(type_of_account.equalsIgnoreCase("department")) {

    //asking user for department number they wish to
associate the new account with

    System.out.println("Please enter the Department
number you wish to associate this account with: ");

    final int account_dep_id = sc.nextInt();

    //calling query to update account_no in
department table

    String updateQueryD = "INSERT INTO
Tracks_Cost(account_no, dep_num) VALUES (?,?)";

    try (PreparedStatement
updatepreparedStatement = connection.prepareStatement(updateQueryD)) {

        updatepreparedStatement.setInt(1,
account_no);

        updatepreparedStatement.setInt(2,
account_dep_id);

        updatepreparedStatement.executeUpdate();

        updatepreparedStatement.close();

    }

    //calling query to insert into specific
account

    String insertDepAcc = "INSERT INTO
Department_acc(account_no, details_2) VALUES(?,?)";

```

```
try (PreparedStatement updatepreparedStatement = connection.prepareStatement(insertDepAcc)) {
    updatepreparedStatement.setInt(1, account_no);
    updatepreparedStatement.setInt(2, 0);

    updatepreparedStatement.executeUpdate();
    System.out.println("Completed. ");
    updatepreparedStatement.close();
}

} else
if(type_of_account.equalsIgnoreCase("assembly")) {
    //asking user for department number they
    //wish to associate the new account with
    System.out.println("Please enter the Assembly ID
you wish to associate this account with: ");
    final int account_ass_id = sc.nextInt();

    //calling query to update account_no in
    //assembly table
    String updateQueryA = "INSERT INTO
Tracks_Cost(account_no, assembly_no) VALUES (?,?)";
    try (PreparedStatement updatepreparedStatement = connection.prepareStatement(updateQueryA)) {
        updatepreparedStatement.setInt(1, account_no);
```

```

        updatepreparedStatement.setInt(2,
account_ass_id);

updatepreparedStatement.executeUpdate();

        updatepreparedStatement.close();
    }

//calling query to insert into specific
account

String insertAssAcc = "INSERT INTO
Assembly_acc(account_no, details_1) VALUES(?,?)";
try (PreparedStatement
updatepreparedStatement = connection.prepareStatement(insertAssAcc)) {
    updatepreparedStatement.setInt(1,
account_no);
    updatepreparedStatement.setInt(2, 0);

updatepreparedStatement.executeUpdate();

System.out.println("Completed. ");
updatepreparedStatement.close();
}

} else {
//user typed in wrong type
System.out.println("Failed to retrieve process type,
please re-enter");
}

}//end of if statements

}//end of result set generated keys

```

```

        connection.close();
    } //end of prepared statement
} //end of connection
break;
case "6":
    // Collect the new department data from the user...
    System.out.println("Please enter the job_no");
    final int job_no = sc.nextInt();

    System.out.println("Please enter the date the job commenced:");
    sc.nextLine();
    final String job_start = sc.nextLine();

    System.out.println("Please enter job information:");
    final String job_details = sc.nextLine();

    System.out.println("Please enter the assembly id");
    final int job_ass_id= sc.nextInt();

    System.out.println("Please enter the process id");
    final int job_process_id = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL))
    {
        //calling query and setting values
    }
}

```

```
try (final CallableStatement statement = connection.prepareCall("EXEC
query_6 @job_no =?, @start_date=?, @details = ?, @process_id =?, @assembly_no =?"))
{
    statement.setInt(1, job_no);
    statement.setDate(2, java.sql.Date.valueOf(job_start));
    statement.setString(3, job_details);
    statement.setInt(4, job_process_id);
    statement.setInt(5, job_ass_id);

    System.out.println("Dispatching the query...");

    //execute the query
    statement.executeUpdate();

    //close statements
    statement.close();
    connection.close();
}

}

break;
case "7":
    // Collect the new department data from the user...
    System.out.println("Please enter the job_no that is completed");
    final int job_no_finished = sc.nextInt();

    System.out.println("Please enter the date the job completed:");
    sc.nextLine();
    final String job_end = sc.nextLine();
```

```

System.out.println("Please enter which type of Job completed (fit/paint/cut)");
final String job_type = sc.nextLine();

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

// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{
    System.out.println("Dispatching the query...");

    //if condition to check user input of type & inserting type in their
    tables
    if(job_type.equalsIgnoreCase("fit"))
    {
        //ask user input
        System.out.println("Please enter the fit labor time: ");
        final int fit_time = sc.nextInt();

        //calling query and setting values
        try (
            final CallableStatement statement =
connection.prepareCall("EXEC query_7A @job_no = ?, @end_date = ?, @fit_labor_time = ?;"))
        {
            statement.setInt(1, job_no_finished);
            statement.setDate(2,
java.sql.Date.valueOf(job_end));
            statement.setInt(3, fit_time);

            //execute the update
        }
    }
}

```

```

        statement.executeUpdate();

        //close the statement
        statement.close();

        //print out successfully inserted
        System.out.println("End date and Fit job
inserted successfully.");
    }

    //if its a paint job...
} else if (job_type.equalsIgnoreCase("paint"))
{
    //ask user input
    System.out.println("Please enter the paint labor time: ");
    final int paint_time = sc.nextInt();

    System.out.println("Please enter the paint color:");
    sc.nextLine();
    final String color = sc.nextLine();

    System.out.println("Please enter the paint volume:");
    final int paint_vol = sc.nextInt();

    //calling query and setting values
    try (
        //callable statement for stored transaction
procedure calls
        final CallableStatement statement =
connection.prepareCall("EXEC query_7B @job_no = ?, @end_date = ?, @paint_labor_time =
?, @color = ?, @volume = ?;"))
    {
        statement.setString(1, job_no);
        statement.setDate(2, end_date);
        statement.setInt(3, paint_time);
        statement.setString(4, color);
        statement.setInt(5, paint_vol);
        statement.executeUpdate();
    }
}

```

```

    {
        statement.setInt(1, job_no_finished);
        statement.setDate(2,
java.sql.Date.valueOf(job_end));
        statement.setInt(3, paint_time);
        statement.setString(4, color);
        statement.setInt(5, paint_vol);

        //execute the update
        statement.executeUpdate();

        //close the statement
        statement.close();

        //print out if successfully inserted
        System.out.println("End date and Paint
job inserted successfully.");
    }

    //if its a cut job...
}else if(job_type.equalsIgnoreCase("cut"))
{
    //ask user input
    System.out.println("Please enter the cut labor time: ");
    final int cut_time = sc.nextInt();

    System.out.println("Please enter the cut machine
used:");
    sc.nextLine();
    final String machine = sc.nextLine();
}

```

```

System.out.println("Please enter the material");
final String matieral = sc.nextLine();

System.out.println("Please enter the time of machine
use");
final int machine_time = sc.nextInt();

//calling query and setting values
try (
    //callable statement for stored transaction
procedure calls
    final CallableStatement statement =
connection.prepareCall("EXEC query_7C @job_no = ?, @end_date =?, @cut_labor_time = ?,
@machine_used = ?, @time_of_machine_use = ?, @material =?;"))

{
    statement.setInt(1, job_no_finished);
    statement.setDate(2,
java.sql.Date.valueOf(job_end));
    statement.setInt(3, cut_time);
    statement.setString(4, machine);
    statement.setInt(5, machine_time);
    statement.setString(6, matieral);

    //execute the update
    statement.executeUpdate();

    //close the statement
    statement.close();
}

```

```

        //print out if successfully inserted
        System.out.println("End date and Cut job
inserted successfully");

    }

}else{
    //user typed in wrong type
    System.out.println("Failed to retrieve job type, please re-enter");
}

//end connection
connection.close();
}

break;
case "8":
    // Collect the new department data from the user...
    System.out.println("Please enter the transaction number");
    final int transaction_no = sc.nextInt();

    System.out.println("Please enter the sup-cost");
    final int supcost = sc.nextInt();

    System.out.println("Please enter the Process account number you want to
update:");
    final int update_process = sc.nextInt();

    System.out.println("Please enter the Assembly account number you want to
update:");
    final int update_assembly = sc.nextInt();
}

```

```
System.out.println("Please enter the Department account number you want to
update:");
final int update_dep = sc.nextInt();

System.out.println("Connecting to the database...");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{
    try (
        final CallableStatement statement =
connection.prepareCall("{CALL query8_insert(?, ?, ?, ?, ?, ?)}"))
    {
        statement.setInt(1, update_process);
        statement.setInt(2, update_assembly);
        statement.setInt(3, update_dep);
        statement.setInt(4, transaction_no);
        statement.setInt(5, supcost);

System.out.println("Dispatching the query...");
//execute the query
statement.executeUpdate();

//close statements
statement.close();
    }
}
```

```
try (final CallableStatement statement = connection.prepareCall("{CALL query_8A(?, ?, ?)}")) {  
    statement.setInt(1, update_process);  
    statement.setInt(2, transaction_no);  
    statement.setInt(3, supcost);  
  
    System.out.println("Dispatching the query...");  
  
    //execute the query  
    statement.executeUpdate();  
  
    //close statements  
    statement.close();  
  
}  
try (final CallableStatement statement = connection.prepareCall("{CALL query_8B(?, ?, ?, ?)}")) {  
    statement.setInt(1, update_assembly);  
    statement.setInt(2, transaction_no);  
    statement.setInt(3, supcost);  
    statement.setInt(4, update_process);  
  
    System.out.println("Dispatching the query...");  
  
    //execute the query  
    statement.executeUpdate();  
}
```

```

        //close statements
        statement.close();

    }

    try (
        final CallableStatement statement = connection.prepareCall("{CALL
query_8C(?, ?, ?, ?, ?)}"))
    {

        statement.setInt(1, update_dep);
        statement.setInt(2, transaction_no);
        statement.setInt(3, supcost);
        statement.setInt(4, update_process);

        System.out.println("Dispatching the query...");

        //execute the query
        statement.executeUpdate();

        //close statements
        statement.close();

    }

    connection.close();
}

break;
case "9":
    //Collecting data from user
    System.out.println("Please enter the Assembly Number:");
    final int assembly_search = sc.nextInt();
}

```

```
System.out.println("Connecting to the database... \n");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{
    //executing transaction procedure call
    try (
        final CallableStatement statement = connection.prepareCall("{CALL
query_9(?)}"))
    {
        statement.setInt(1,assembly_search);

        System.out.println("Dispatching the query... \n");

        boolean hasResults = statement.execute();
        if (hasResults) {
            ResultSet rs = statement.getResultSet();
            // Process and print the result set
            while (rs.next()) {
                int total_cost= rs.getInt("total_cost");

                System.out.println("Total Cost: " + total_cost);
            }
        }
    }

    //close statements
    statement.close();
    connection.close();
}
```

```

        }
    }

    break;

    case "10":
        //Collecting data from user
        System.out.println("Please enter the Department number:");
        final int dep_num_search = sc.nextInt();

        System.out.println("Please enter start date");
        sc.nextLine();
        final String start_date_search = sc.nextLine();

        System.out.println("Please enter end date");
        final String end_date_search = sc.nextLine();

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL))
        {
            //executing transaction procedure call
            try (
                final CallableStatement statement = connection.prepareCall("{CALL
query_10(?, ?, ?)}"))
            {
                statement.setInt(1, dep_num_search);
                statement.setDate(2, java.sql.Date.valueOf(start_date_search));
                statement.setDate(3, java.sql.Date.valueOf(end_date_search));
            }
        }
    }
}

```

```

System.out.println("Dispatching the query...");

boolean hasResults = statement.execute();
if (hasResults) {
    ResultSet rs = statement.getResultSet();
    // Process and print the result set
    while (rs.next()) {
        int dep_no_job_search= rs.getInt("dep_num");
        int total_time = rs.getInt("total_labor_time");

        System.out.println("Total labor timne: " + total_time);
    }
}

//close statements
statement.close();
connection.close();

}

}

break;
case "11":
    // Collect the new department data from the user...
    System.out.println("Please enter the Assembly number:");
    final int assembly_pro = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL))

```

```

{
    try (
        final CallableStatement statement =
connection.prepareCall("{CALL query_11(?)}"))
    {
        statement.setInt(1, assembly_pro);

        System.out.println("Dispatching the query...");

        boolean hasResults = statement.execute();
        if (hasResults) {
            ResultSet rs = statement.getResultSet();
            // Process and print the result set
            while (rs.next()) {
                int processId_ass = rs.getInt("process_id");
                String processType = rs.getString("process_type");
                int depNum = rs.getInt("dep_num");

                System.out.println("Process ID: " + processId_ass + ", Dep
Num: " + depNum + ", Process Type: " + processType);
            }
        }
        //close statements
        statement.close();
        connection.close();
    }
}

```

```

break;

case "12":
    // Collect the new department data from the user...
    System.out.println("Please enter the lower limit of the category range:");
    final int lower_limit = sc.nextInt();

    System.out.println("Please enter the upper limit of the category range:");
    final int upper_limit = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL))
    {
        //executing stored transaction procedure call
        try (
            final CallableStatement statement =
            connection.prepareCall("{CALL query_12(?, ?)}"))
        {
            statement.setInt(1, lower_limit);
            statement.setInt(2, upper_limit);

            System.out.println("Dispatching the query...");
            boolean hasResults = statement.execute();
            if (hasResults) {
                ResultSet rs = statement.getResultSet();
                // print the result set
                while (rs.next()) {
                    String customer_name = rs.getString("cname");
                    String customer_addy= rs.getString("address");

```

```

        int cust_category = rs.getInt("category");

        System.out.println("Customer Name: " + customer_name + ", "
Address: " + customer_addy + ", Category: " + cust_category);

    }

}

//close statements
statement.close();
connection.close();

}

}

break;
case "13":
    // Collect the new department data from the user...
    System.out.println("Please enter the lower limit of the Job number range:");
    final int job_lower_limit = sc.nextInt();

    System.out.println("Please enter the upper limit of the Job number range:");
    final int job_upper_limit = sc.nextInt();

    System.out.println("Connecting to the database... \n");
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL))
    {
        //executing stored transaction procedure call
        try (

```



```

        String cut_material = rs.getString("material");

        System.out.println("Cut Job Number: " +
cut_job_no + ", Cut Labor Time: " + cut_labor + ", Cut Machine: " + cut_machine + ", Cut
Machine Time: " + time_machine + ", Cut Material: " + cut_material);

    }

}

// Close the statement after executing
cs.close();
}

connection.close();
}

}

break;
case "14":
// Collect the new department data from the user...
System.out.println("Please enter the Paint Job Number:");
final int paint_no = sc.nextInt();
sc.nextLine();

System.out.println("Please enter the color you wish to change it to:");
final String change_color = sc.nextLine();

System.out.println("Connecting to the database... \n");
// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL))
{
    //executing stored transaction procedure call
    try (

```

```
final CallableStatement statement =
connection.prepareCall("{CALL query_14(?, ?)}"))

{
    statement.setString(1, change_color);
    statement.setInt(2, paint_no);

    statement.execute();
    System.out.println("Dispatching the query... \n");

    System.out.println("Paint color successfully changed \n");

    //close statements
    statement.close();
    connection.close();

}

}

break;
case "15":
    //import function
    System.out.println("Do you want to import? Press 1: ");
    final int random = sc.nextInt();
    sc.nextLine();

    System.out.println("Please enter the path of the file import");
    final String fileName = sc.nextLine();
```

```

try (final Connection connection = DriverManager.getConnection(URL))
{
    insertDataFromFile(connection, fileName);

} catch (SQLException e) {
    e.printStackTrace();
}

break;
case "16":
    System.out.println("Enter the lower limit of the category range:");

    int lowerLimit = sc.nextInt();

    System.out.println("Enter the upper limit of the category range:");
    int upperLimit = sc.nextInt();

    System.out.println("Enter the output file name:");
    String outputFileName = sc.next();

    try (Connection connection = DriverManager.getConnection(URL)) {

        exportCustomersToFile(connection, lowerLimit, upperLimit,
outputFileName);

    } catch (SQLException | IOException e) {
        e.printStackTrace();
    }

break;
case "17": // Do nothing, the while loop will terminate upon the next iteration
    System.out.println("Exiting! Good-bye!");
}

```

```

        break;

    default: // Unrecognized option, re-prompt the user for the correct one
        System.out.println(String.format(
            "Unrecognized option: %s\n" +
            "Please try again!",
            option));
        break;
    }
}

sc.close(); // Close the scanner before exiting the application
}

private static void insertFitData(Connection connection, int process_id, String fit_type) throws
Exception {
    try (PreparedStatement preparedStatement =
connection.prepareStatement(QUERY_TEMPLATE_3_A)) {
        preparedStatement.setInt(1, process_id);
        preparedStatement.setString(2, fit_type);
        preparedStatement.executeUpdate();
        System.out.println("Fit data inserted successfully.");
    }
}

private static void insertPaintData(Connection connection, int process_id, String paint_type,
String paint_method) throws Exception {
    try (PreparedStatement preparedStatement =
connection.prepareStatement(QUERY_TEMPLATE_3_B)) {
        preparedStatement.setInt(1, process_id);
        preparedStatement.setString(2, paint_type);
    }
}

```

```
        preparedStatement.setString(3, paint_method);
        preparedStatement.executeUpdate();
        System.out.println("Paint data inserted successfully.");
    }
}

private static void insertCutData(Connection connection, int process_id, String cut_type,
String machine_type) throws Exception {
    try (PreparedStatement preparedStatement =
connection.prepareStatement(QUERY_TEMPLATE_3_C)) {
        preparedStatement.setInt(1, process_id);
        preparedStatement.setString(2, cut_type);
        preparedStatement.setString(3, machine_type);
        preparedStatement.executeUpdate();
        System.out.println("Cut data inserted successfully.");
    }
}

private static void insertProcessAcc(Connection connection, int account_no, String
start_of_acc, int process_id) throws Exception {
    try (PreparedStatement preparedStatement = connection.prepareStatement("EXEC
query_5_process @account_no = ?, @start_of_acc = ?, @process_id = ?;")) {
        preparedStatement.setInt(1, account_no);
        preparedStatement.setString(2, start_of_acc);
        preparedStatement.setInt(3, process_id);
        preparedStatement.executeUpdate();
        System.out.println("Process Account inserted successfully.");
    }
}
```

```

private static void insertDataFromFile(Connection connection, String fileName) throws
IOException, SQLException {
    try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
        String headerLine = reader.readLine();
        String[] headers = headerLine.split(" ");

        String line;
        while ((line = reader.readLine()) != null) {
            // Assuming CSV format: id,firstName,lastName,email
            String[] data = line.split(",");
            String firstName = data[0];
            String lastName = data[1];
            int customerID = Integer.parseInt(data[2]);

            // Step 4: Insert data into the Customer table
            insertCustomer(connection, firstName, lastName, customerID);
        }
    }
}

private static void insertCustomer(Connection connection, String firstName, String lastName,
int customerID) throws SQLException {
    String sql = "INSERT INTO Customer (cname, address, category) VALUES (?, ?, ?)";
    try (PreparedStatement preparedStatement = connection.prepareStatement(sql)) {
        preparedStatement.setString(1, firstName);
        preparedStatement.setString(2, lastName);
        preparedStatement.setInt(3, customerID);
        preparedStatement.executeUpdate();
    }
}

```

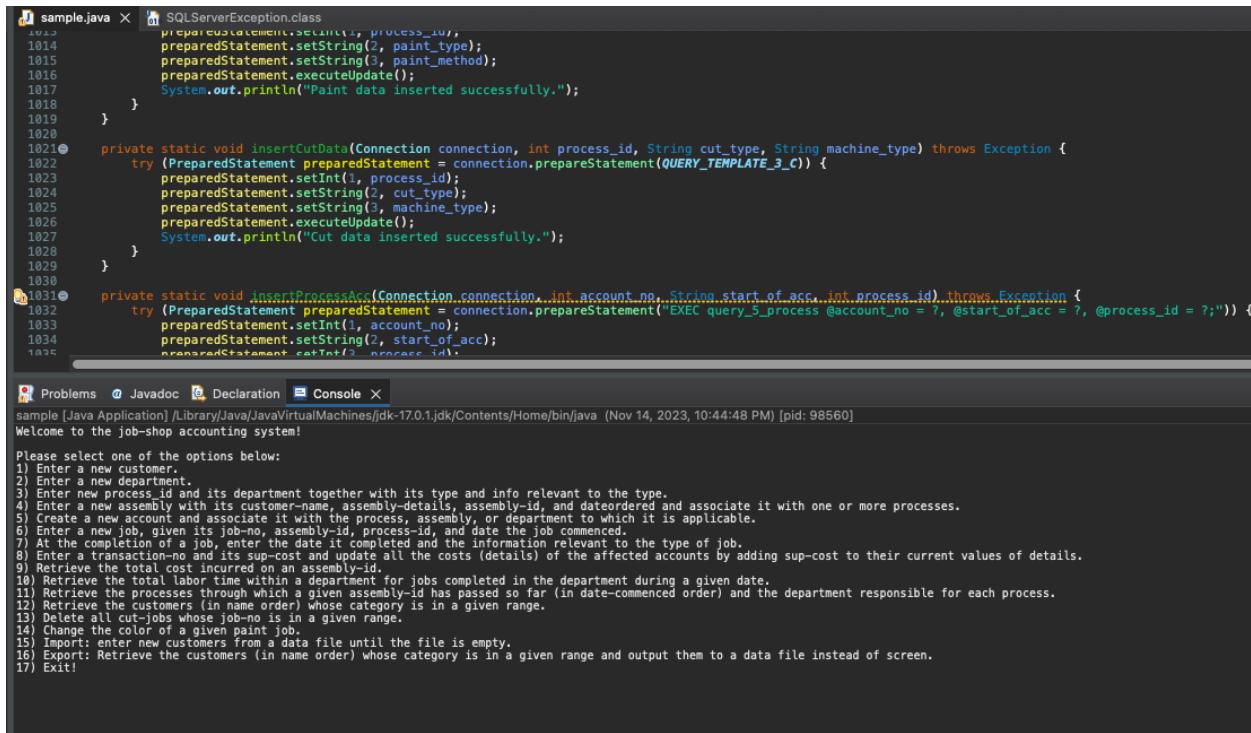
```
private static void exportCustomersToFile(Connection connection, int lowerLimit, int
upperLimit, String outputFileName)
throws SQLException, IOException {
String sql = "SELECT cname, address, category FROM Customer WHERE category
BETWEEN ? AND ? ORDER BY cname";
try (PreparedStatement statement = connection.prepareStatement(sql)) {
statement.setInt(1, lowerLimit);
statement.setInt(2, upperLimit);

try (ResultSet resultSet = statement.executeQuery()) {
BufferedWriter writer = new BufferedWriter(new FileWriter(outputFileName)) {
while (resultSet.next()) {
String name = resultSet.getString("cname");
String address = resultSet.getString("address");
int category = resultSet.getInt("category");

// Write to file
writer.write("Name: " + name + ", Address: " + address + ", Category: " + category);
writer.newLine();
}

System.out.println("Export completed. Data saved to: " + outputFileName);
}
}
}
}
}
```

Screenshot of Compilation:



```

1 sample.java x  SQLServerException.class
1013     preparedStatement.setString(1, process_id);
1014     preparedStatement.setString(2, paint_type);
1015     preparedStatement.setString(3, paint_method);
1016     preparedStatement.executeUpdate();
1017     System.out.println("Paint data inserted successfully.");
1018 }
1019
1020 private static void insertCutData(Connection connection, int process_id, String cut_type, String machine_type) throws Exception {
1021     try (PreparedStatement preparedStatement = connection.prepareStatement(QUERY_TEMPLATE_3_C)) {
1022         preparedStatement.setInt(1, process_id);
1023         preparedStatement.setString(2, cut_type);
1024         preparedStatement.setString(3, machine_type);
1025         preparedStatement.executeUpdate();
1026     }
1027     System.out.println("Cut data inserted successfully.");
1028 }
1029
1030 private static void insertProcessData(Connection connection, int account_no, String start_of_acc, int process_id) throws Exception {
1031     try (PreparedStatement preparedStatement = connection.prepareStatement("EXEC query_5_process @account_no = ?, @start_of_acc = ?, @process_id = ?")) {
1032         preparedStatement.setInt(1, account_no);
1033         preparedStatement.setString(2, start_of_acc);
1034         preparedStatement.setInt(3, process_id);
1035     }
1036 }

```

Problems Javadoc Declaration Console X

sample [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java (Nov 14, 2023, 10:44:48 PM) [pid: 98560]

Welcome to the job-shop accounting system!

Please select one of the options below:

- 1) Enter a new customer.
- 2) Enter a new department.
- 3) Enter new process_id and its department together with its type and info relevant to the type.
- 4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
- 5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
- 6) Enter a new job given its job-no, assembly-id, process-id, and date the job commenced.
- 7) At the completion of a job enter the date it completed and the information relevant to the type of job.
- 8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
- 9) Retrieve the total cost incurred on an assembly-id.
- 10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
- 11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
- 12) Retrieve the customers (in name order) whose category is in a given range.
- 13) Delete all cut-jobs whose job-no is in a given range.
- 14) Change the color of a given paint job.
- 15) Import: enter new customers from a data file until the file is empty.
- 16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
- 17) Exit!

Task 6:

Task 6.2 Query 1 Screenshots:

First Input:

```
Welcome to the job-shop accounting system!
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scr
17) Exit!
1
Please enter the customer name:
Sofia
Please enter the customer's address:
6210 Del Norte Ln
Please enter the customer's category, a #1-10:
1
Connecting to the database...
Dispatching the query...
```

Second input

```
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more pr
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department respons
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scr
17) Exit!
1
Please enter the customer name:
Matthew
Please enter the customer's address:
3497 Hanover Ave
Please enter the customer's category, a #1-10:
5
Connecting to the database...
Dispatching the query...
```

Third input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more p
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department respons
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scr
17) Exit!
1
Please enter the customer name:
Michelle
Please enter the customer's address:
2094 Michelle Ln
Please enter the customer's category, a #1-10:
2
Connecting to the database...
Dispatching the query...

```

Fourth input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more p
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department respons
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scr
17) Exit!
1
Please enter the customer name:
Jacob
Please enter the customer's address:
90210 Bev Hills
Please enter the customer's category, a #1-10:
2
Connecting to the database...
Dispatching the query...

```

Fifth Input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current value.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
1
Please enter the customer name:
John
Please enter the customer's address:
6808 Topsfield Dr
Please enter the customer's category, a #1-10:
1
Connecting to the database...
Dispatching the query...

```

Output from 5 queries:

```
1   SELECT * FROM Customer;
```

Results Messages

	cname	address	category
1	Jacob	90210 Bev Hills	2
2	John	6808 Topsfield Dr	1
3	Matthew	3497 Hanover Ave	5
4	Michelle	2094 Michelle Ln	2
5	Sofia	6210 Del Norte Ln	1

6. 2: Query 2 Screenshots

First Input:

```
Sample [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java (Nov 14, 2023)
Welcome to the job-shop accounting system!

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with the process, assembly, or department to which it is applicable.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding to it.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order).
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file.
17) Exit!
2
Please enter the Department number:
1
Please enter Department data, if none type NULL:
Data
Connecting to the database...
Dispatching the query...
```

Second input:

```
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with the process, assembly, or department to which it is applicable.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding to it.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and associate it with the process, assembly, or department to which it is applicable.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file.
17) Exit!
2
Please enter the Department number:
2
Please enter Department data, if none type NULL:
Data
Connecting to the database...
Dispatching the query...
```

Third Input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected acco
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output t
17) Exit!
2
Please enter the Department number:
3
Please enter Department data, if none type NULL:
Data
Connecting to the database...
Dispatching the query...

```

Fourth input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department r
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead
17) Exit!
2
Please enter the Department number:
4
Please enter Department data, if none type NULL:
Data
Connecting to the database...
Dispatching the query...

```

Fifth Input:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead
17) Exit!
2
Please enter the Department number:
5
Please enter Department data, if none type NULL:
Data
Connecting to the database...
Dispatching the query...

```

Output:

1 **SELECT * FROM Department;**

Results **Messages**

	dep_num	dep_data
1	1	Data
2	2	Data
3	3	Data
4	4	Data
5	5	Data

Task 6.3 Query 3 Screenshots:

Input 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and
5) Create a new account and associate it with the process, assembly, or department
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job com
7) At the completion of a job, enter the date it completed and the information rele
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the dep
11) Retrieve the processes through which a given assembly-id has passed so far (in
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given ran
17) Exit!
3
Please enter the Process ID:
1
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
1
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
fit
Please enter information on which fit type:
fit type
Fit data inserted successfully.

```

Input 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and
5) Create a new account and associate it with the process, assembly, or department
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job com
7) At the completion of a job, enter the date it completed and the information rele
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the dep
11) Retrieve the processes through which a given assembly-id has passed so far (in
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given ran
17) Exit!
3
Please enter the Process ID:
2
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
1
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
paint
Please enter information on which paint type:
paint type
Please enter information on painting method:
paint method
Paint data inserted successfully.

```

Input 3:

```
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to t
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateo
5) Create a new account and associate it with the process, assembly, or department to wh
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commence
7) At the completion of a job, enter the date it completed and the information relevant
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the aff
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the departme
11) Retrieve the processes through which a given assembly-id has passed so far (in date-
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and
17) Exit!
3
Please enter the Process ID:
3
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
1
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
cut
Please enter information on which cut type:
cut type
Please enter information on machine type:
cut machine
Cut data inserted successfully.
```

Input 4:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected account
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output
17) Exit!
3
Please enter the Process ID:
4
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
2
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
fit
Please enter information on which fit type:
fit type
Fit data inserted successfully.

```

Input 5:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the account
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range
17) Exit!
3
Please enter the Process ID:
5
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
2
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
paint
Please enter information on which paint type:
paint type
Please enter information on painting method:
paint method
Paint data inserted successfully.

```

Input 6:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associat
5) Create a new account and associate it with the process, assembly, or department to which it is applicabl
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by a
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given d
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) an
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a
17) Exit!
3
Please enter the Process ID:
6
Please enter Process data, if none type NULL:
data
Please enter the Department ID you want to associate with this new process:
2
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
cut
Please enter information on which cut type:
cut type
Please enter information on machine type:
cut machine
Cut data inserted successfully.

```

Input 7:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associat
5) Create a new account and associate it with the process, assembly, or department to which it is applicabl
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by a
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given d
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) an
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a
17) Exit!
3
Please enter the Process ID:
7
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
3
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
fit
Please enter information on which fit type:
fit type
Fit data inserted successfully.

```

Input 8:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output
17) Exit!
3
Please enter the Process ID:
8
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
3
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
paint
Please enter information on which paint type:
pepaint ty
Please enter information on painting method:
paint method
Paint data inserted successfully.

```

Input 9:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected account.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output
17) Exit!
3
Please enter the Process ID:
9
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
3
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
cut
Please enter information on which cut type:
cut type
Please enter information on machine type:
cut machine
Cut data inserted successfully.

```

Input 10:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data
17) Exit!
3
Please enter the Process ID:
10
Please enter Process data, if none type NULL:
process data
Please enter the Department ID you want to associate with this new process:
4
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)|fit
Please enter information on which fit type:
fit type
Fit data inserted successfully.

```

Out from Process Table:

1 **SELECT * FROM Process;**

Results **Messages**

	process_id	process_data	dep_num	assembly_no
1	1	process data	1	NULL
2	2	process data	1	NULL
3	3	process data	1	NULL
4	4	process data	2	NULL
5	6	data	2	NULL
6	7	process data	3	NULL
7	8	process data	3	NULL
8	9	process data	3	NULL
9	10	process data	4	NULL
10	5	process data	2	NULL

R

Output from Fit Table:

```
1  SELECT * FROM Fit_process;
2
```

Results Messages

	process_id	fit_type
1	1	fit type
2	4	fit type
3	7	fit type
4	10	fit type

Output from Paint table:

```
1  SELECT * FROM Paint_process;
2
```

Results Messages

	process_id	paint_type	paint_method
1	2	paint type	paint method
2	5	paint type	paint method
3	8	pepaint ty	paint method

Output from Cut table:

```
1  SELECT * FROM Cut_process;
2
```

Results Messages

	process_id	cut_type	machine_type
1	3	cut type	cut machine
2	6	cut type	cut machine
3	9	cut type	cut machine

Task 6. 4 Query 4 Screenshots:

Input 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file in
17) Exit!
4
Please enter the Assembly number:
1
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-12
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Sofia
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
1
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
2
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file in
17) Exit!
4
Please enter the Assembly number:
2
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-12
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Sofia
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
3
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 3:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to them.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
3
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-12
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Michelle
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
4
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 4:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to them.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
4
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-13
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Matthew
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
5
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 5:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to the
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department to which it is
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of the screen.
17) Exit!
4
Please enter the Assembly number:
5
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-13
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Matthew
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
6
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 6:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to the
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department to which it is
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of the screen.
17) Exit!
4
Please enter the Assembly number:
6
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-14
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
John
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
7
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.

```

Input 7:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
7
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-14
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
John
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
8
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 8:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current va
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for e
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
8
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-14
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Jacob
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
9
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 9:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of det
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
9
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-14
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Jacob
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
10
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Input 10:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of det
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
4
Please enter the Assembly number:
10
Please enter the date this assembly was ordered (YEAR-MONTH-DATE):
2023-11-14
Please enter the assembly details:
details
Please enter the name of the customer who ordered:
Jacob
Connecting to the database...
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
11
Please enter the process id you want to associate this assembly with, type 'stop' to stop entering a new process:
stop
Exiting the query.
Dispatching the query...

```

Created an extra process for input 10 of assembly table

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with on
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-co
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the depart
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file in
17) Exit!
3
Please enter the Process ID:
11
Please enter Process data, if none type NULL:
data
Please enter the Department ID you want to associate with this new process:
3
Connecting to the database...
Dispatching the query...
Please enter the process type (fit, paint, cut)
fit
Please enter information on which fit type:
fit type
Fit data inserted successfully.

```

Output of updated Process table:

1 **SELECT * FROM Process;**

SQLQuery_1 1 of 1 problem

Results Messages

	process_id	process_data	dep_num	assembly_no
1	1	process data	1	1
2	2	process data	1	1
3	3	process data	1	2
4	4	process data	2	3
5	6	data	2	5
6	7	process data	3	6
7	8	process data	3	7
8	9	process data	3	8
9	10	process data	4	9
10	5	process data	2	4
11	11	data	3	10

Output of Assemblies Table:

```
1  SELECT * FROM Assemblies;
2
```

Results Messages

	assembly_no	date_ordered	assembly_details	cname
1	1	2023-11-12	details	Sofia
2	2	2023-11-12	details	Sofia
3	3	2023-11-12	details	Michelle
4	4	2023-11-13	details	Matthew
5	6	2023-11-14	details	John
6	7	2023-11-14	details	John
7	8	2023-11-14	details	Jacob
8	9	2023-11-14	details	Jacob
9	10	2023-11-14	details	Jacob
10	5	2023-11-13	details	Matthew

Task 6.5 Query 5 Screenshots:

Input 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and a
5) Create a new account and associate it with the process, assembly, or department to which it is ap
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accoun
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced or
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output the
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Process
Connecting to the database...
Please enter the process id you wish to associate this account with:
1

```

Input 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and a
5) Create a new account and associate it with the process, assembly, or department to which it is ap
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accoun
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced or
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output the
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Process
Connecting to the database...
Please enter the process id you wish to associate this account with:
2

```

Input 3:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by addin
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and th
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Process
Connecting to the database...
Please enter the process id you wish to associate this account with:
3

```

Input 4:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more proc
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their cu
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsib
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Department
Connecting to the database...
Please enter the Department number you wish to associate this account with:
1

```

Input 5:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Department
Connecting to the database...
Please enter the Department number you wish to associate this account with:
2

```

Input 6:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Department
Connecting to the database...
Please enter the Department number you wish to associate this account with:
3

```

Input 7:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Assembly
Connecting to the database...
Please enter the Assembly ID you wish to associate this account with:
1

```

Input 8:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Assembly
Connecting to the database...
Please enter the Assembly ID you wish to associate this account with:
2

```

Input 9:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Assembly
Connecting to the database...
Please enter the Assembly ID you wish to associate this account with:
3
|

```

Input 10:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
5
The database will generate a unique account number.

Do you wish to create a Process, Department, or Assembly account?
Assembly
Connecting to the database...
Please enter the Assembly ID you wish to associate this account with:
4
|

```

Output of Account Table:

```

1  SELECT * FROM Account;
2
3

```

Results Messages

	account_no	start_of_acc
1	1	2023-11-14
2	2	2023-11-14
3	3	2023-11-14
4	4	2023-11-14
5	5	2023-11-14
6	6	2023-11-14
7	7	2023-11-14
8	8	2023-11-14
9	9	2023-11-14
10	10	2023-11-14

Output of Assembly Account Table:

```

1  SELECT * FROM Assembly_acc;
2
3

```

Results Messages

	account_no	start_of_acc	details_1
1	7	NULL	0
2	8	NULL	0
3	9	NULL	0
4	10	NULL	0

Output of Department Account Table:

```
1  SELECT * FROM Department_acc;  
2  |  
3  |  
4  |
```

Results Messages

	account_no ▾	start_of_acc ▾	details_2 ▾
	4	NULL	0
	5	NULL	0
	6	NULL	0

Output of Process Account Table:

```
1  SELECT * FROM Process_acc;  
2  |  
3  |  
4  |
```

Results Messages

	account_no ▾	start_of_acc ▾	details_3 ▾
1	1	NULL	0
2	2	NULL	0
3	3	NULL	0

Output of Tracks Cost Table (relationship entity between account- process, dep, assembly)

```
1  SELECT * FROM Tracks_Cost;
2
3
4
```

results Messages

account_no	process_id	dep_num	assembly_no
1	1	NULL	NULL
2	2	NULL	NULL
3	3	NULL	NULL
4	NULL	1	NULL
5	NULL	2	NULL
6	NULL	3	NULL
7	NULL	NULL	1
8	NULL	NULL	2
9	NULL	NULL	3
10	NULL	NULL	4

Task 6.6 Query 6 Screenshots:

Input 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date the job commenced
5) Create a new account and associate it with the process, assembly, or department
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
7) At the completion of a job, enter the date it completed and the information relevant
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the transaction
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department
11) Retrieve the processes through which a given assembly-id has passed so far (in order)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range.
17) Exit!
6
Please enter the job_no
1
Please enter the date the job commenced:
2023-11-12
Please enter the assembly id
1
Please enter the prcoess id
1
Connecting to the database...
Dispatching the query...

```

Input 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date the job commenced
5) Create a new account and associate it with the process, assembly, or department
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
7) At the completion of a job, enter the date it completed and the information relevant
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the transaction
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department
11) Retrieve the processes through which a given assembly-id has passed so far (in order)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range.
17) Exit!
6
Please enter the job_no
2
Please enter the date the job commenced:
2023-11-12
Please enter the assembly id
1
Please enter the prcoess id
2
Connecting to the database...
Dispatching the query...

```

Input3:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateof
5) Create a new account and associate it with the process, assembly, or department to whi
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
7) At the completion of a job, enter the date it completed and the information relevant to
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affe
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department
11) Retrieve the processes through which a given assembly-id has passed so far (in date-ord
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and
17) Exit!
6
Please enter the job_no
3
Please enter the date the job commenced:
2023-11-12
Please enter the assembly id
2
Please enter the process id
3
Connecting to the database...
Dispatching the query...

```

Input 4:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateof
5) Create a new account and associate it with the process, assembly, or department to whi
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
7) At the completion of a job, enter the date it completed and the information relevant to
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affe
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department
11) Retrieve the processes through which a given assembly-id has passed so far (in date-ord
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and
17) Exit!
6
Please enter the job_no
4
Please enter the date the job commenced:
2023-11-12
Please enter the assembly id
3
Please enter the process id
4
Connecting to the database...
Dispatching the query...

```

Input 5:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
5) Create a new account and associate it with the process, assembly, or department to which it
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the job
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected assembly
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given time period.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-ordered)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output to a file.
17) Exit!
6
Please enter the job_no
5
Please enter the date the job commenced:
2023-11-13
Please enter the assembly id
4
Please enter the process id
5
Connecting to the database...
Dispatching the query...

```

Input 6:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered
5) Create a new account and associate it with the process, assembly, or department to which it
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the job
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected assembly
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given time period.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-ordered)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output to a file.
17) Exit!
6
Please enter the job_no
6
Please enter the date the job commenced:
2023-11-13
Please enter the assembly id
5
Please enter the process id
6
Connecting to the database...
Dispatching the query...

```

Input 7:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the typ
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected acco
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output t
17) Exit!
6
Please enter the job_no
7
Please enter the date the job commenced:
2023-11-14
Please enter the assembly id
6
Please enter the process id
7
Connecting to the database...
Dispatching the query...

```

Input 8:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the typ
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected a
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commence
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output
17) Exit!
6
Please enter the job_no
8
Please enter the date the job commenced:
2023-11-14
Please enter the assembly id
7
Please enter the process id
8
Connecting to the database...
Dispatching the query...

```

Input 9:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is a
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected acco
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced o
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output t
17) Exit!
6
Please enter the job_no
9
Please enter the date the job commenced:
2023-11-14
Please enter the assembly id
8
Please enter the process id
9
Connecting to the database...
Dispatching the query...

```

Input 10:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and
5) Create a new account and associate it with the process, assembly, or department to which it is a
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected acco
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced o
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output t
17) Exit!
6
Please enter the job_no
10
Please enter the date the job commenced:
2023-11-14
Please enter the assembly id
9
Please enter the process id
10
Connecting to the database...
Dispatching the query...

```

Output of Job Table:

```
1  SELECT * FROM Job;
2
3
4
5
```

Results Messages

	job_no	start_date	end_date	additional_info	process_id	assembly_no
1	1	2023-11-12	NULL	NULL	1	1
2	2	2023-11-12	NULL	NULL	2	1
3	3	2023-11-12	NULL	NULL	3	2
4	4	2023-11-12	NULL	NULL	4	3
5	5	2023-11-13	NULL	NULL	5	4
6	6	2023-11-13	NULL	NULL	6	5
7	10	2023-11-14	NULL	NULL	10	9
8	7	2023-11-14	NULL	NULL	7	6
9	8	2023-11-14	NULL	NULL	8	7
10	9	2023-11-14	NULL	NULL	9	8

Task 6.7 Query 7 Screenshots:

Input 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to it.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date the job commenced.
5) Create a new account and associate it with the process, assembly, or department.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to it.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected assembly.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department.
11) Retrieve the processes through which a given assembly-id has passed so far.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range.
17) Exit!
7
Please enter the job_no that is completed
1
Please enter the date the job completed:
2023-11-13
Please enter which type of Job completed (fit/paint/cut)
fit
Connecting to the database...
Dispatching the query...
Please enter the fit labor time:
1
End date and Fit job inserted successfully.

```

Input 2:

```

5) Create a new account and associate it with the process, assembly, or department to which the job belongs.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to it.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected assembly.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-order).
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and order them.
17) Exit!
7
Please enter the job_no that is completed
2
Please enter the date the job completed:
2023-11-13
Please enter which type of Job completed (fit/paint/cut)
paint
Connecting to the database...
Dispatching the query...
Please enter the paint labor time:
2
Please enter the paint color:
Red
Please enter the paint volume:
2
End date and Paint job inserted successfully.

```

Input 3:

```

7) At the completion of a job, enter the date it completed and the information relevant to it.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the transaction.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department.
11) Retrieve the processes through which a given assembly-id has passed so far (in date order).
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range.
17) Exit!
7
Please enter the job_no that is completed
3
Please enter the date the job completed:
2023-11-13
Please enter which type of Job completed (fit/paint/cut)
cut
Connecting to the database...
Dispatching the query...
Please enter the cut labor time:
2
Please enter the cut machine used:
cut machine
Please enter the material
cut material
Please enter the time of machine use
1
End date and Cut job inserted successfully.

```

Input 4:

```

1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to it.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and department.
5) Create a new account and associate it with the process, assembly, or department to it.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job completed.
7) At the completion of a job, enter the date it completed and the information relevant to it.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the transaction.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department.
11) Retrieve the processes through which a given assembly-id has passed so far (in date order).
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range.
17) Exit!
7
Please enter the job_no that is completed
4
Please enter the date the job completed:
2023-11-13
Please enter which type of Job completed (fit/paint/cut)
fit
Connecting to the database...
Dispatching the query...
Please enter the fit labor time:
1
End date and Fit job inserted successfully.

```

Input 5:

```

5) Create a new account and associate it with the process, assembly, or department
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job com
7) At the completion of a job, enter the date it completed and the information rele
8) Enter a transaction-no and its sup-cost and update all the costs (details) of th
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the dep
11) Retrieve the processes through which a given assembly-id has passed so far (in
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given ran
17) Exit!
7
Please enter the job_no that is completed
5
Please enter the date the job completed:
2023-11-14
Please enter which type of Job completed (fit/paint/cut)
paint
Connecting to the database...
Dispatching the query...
Please enter the paint labor time:
2
Please enter the paint color:
Purple
Please enter the paint volume:
1
End date and Paint job inserted successfully.

```

Input 6:

```

7) At the completion of a job, enter the date it completed and the information
8) Enter a transaction-no and its sup-cost and update all the costs (details) o
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the
11) Retrieve the processes through which a given assembly-id has passed so far
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given
17) Exit!
7
Please enter the job_no that is completed
6
Please enter the date the job completed:
2023-11-14
Please enter which type of Job completed (fit/paint/cut)
cut
Connecting to the database...
Dispatching the query...
Please enter the cut labor time:
1
Please enter the cut machine used:
cut machine
Please enter the material
cut material
Please enter the time of machine use
1
End date and Cut job inserted successfully.

```

Input 7:

```

7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding su
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the de
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data fil
17) Exit!
7
Please enter the job_no that is completed
7
Please enter the date the job completed:
2023-11-15
Please enter which type of Job completed (fit/paint/cut)
fit
Connecting to the database...
Dispatching the query...
Please enter the fit labor time:
3
End date and Fit job inserted successfully.

```

Input 8:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associ
5) Create a new account and associate it with the process, assembly, or department to which it is applica
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of jo
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to
17) Exit!
7
Please enter the job_no that is completed
8
Please enter the date the job completed:
2023-11-15
Please enter which type of Job completed (fit/paint/cut)
paint
Connecting to the database...
Dispatching the query...
Please enter the paint labor time:
2
Please enter the paint color:
Purple
Please enter the paint volume:
1
End date and Paint job inserted successfully.

```

Input 9:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current value.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
7
Please enter the job_no that is completed
9
Please enter the date the job completed:
2023-11-15
Please enter which type of Job completed (fit/paint/cut)
cut
Connecting to the database...
Dispatching the query...
Please enter the cut labor time:
2
Please enter the cut machine used:
cut machine
Please enter the material
cut material
Please enter the time of machine use
1
End date and Cut job inserted successfully.

```

Input 10:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current value.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
7
Please enter the job_no that is completed
10
Please enter the date the job completed:
2023-11-15
Please enter which type of Job completed (fit/paint/cut)
fit
Connecting to the database...
Dispatching the query...
Please enter the fit labor time:
1
End date and Fit job inserted successfully.

```

Output of updated Job table:

(I went back and added user input for additional info column which is why its not null here)

```

1
2  SELECT * FROM Job;
3

```

Results Messages

job_no	start_date	end_date	additional_info	process_id	assembly_no
3	2023-11-12	2023-11-13		3	2
4	2023-11-12	2023-11-13		4	3
5	2023-11-13	2023-11-14		5	4
6	2023-11-13	2023-11-14		6	5
7	2023-11-14	2023-11-15		7	6
8	2023-11-14	2023-11-15		8	7
9	2023-11-14	2023-11-15		9	8
10	2023-11-14	2023-11-15		10	9
1	2023-11-12	2023-11-13	information	1	1
2	2023-11-12	2023-11-13		2	1

Output of Fit Job Table:

```

1  SELECT * FROM Fit_Job;
2

```

Results Messages

job_no	fit_labor_time
1	1
2	4
3	7
4	10

Output of Cut Job Table:

```

1  SELECT* FROM Cut_Job;
2

```

Results Messages

	job_no	cut_labor_time	machine_used	time_of_machine_use	material
1	3	2	cut machine	1	cut material
2	6	1	cut machine	1	cut material
3	9	2	cut machine	1	cut material

Output of Paint Job Table

```

1  SELECT * FROM Paint_Job;

```

Results Messages

	job_no	color	volume	paint_labor_time
1	2	Red	2	2
2	5	Purple	1	2
3	8	Purple	1	2

Task 6.8 Query 8 Screenshots:

An account can only be updated correctly if the Department and Assembly Account number are associated with the process_id from the Process Account which is why I could only update certain accounts and not all.

Input 1:

```

12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
1
Please enter the sup-cost
10
Please enter the Process account number you want to update:
1
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 2:

```

sample [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java [Nov 14, 2023, 7:44:14]
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a
17) Exit!
8
Please enter the transaction number
2
Please enter the sup-cost
20
Please enter the Process account number you want to update:
1
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 3:

```

12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file ins
17) Exit!
8
Please enter the transaction number
3
Please enter the sup-cost
100
Please enter the Process account number you want to update:
1
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 4:

Notice for this input, I inputted the incorrect assembly and department number so the transaction does not update the assembly nor department accs, only the process

```

14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output
17) Exit!
8
Please enter the transaction number
4
Please enter the sup-cost
130
Please enter the Process account number you want to update:
2
Please enter the Assembly account number you want to update:
4
Please enter the Department account number you want to update:
7
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Please select one of the options below:

```

Input 5:

```

12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
5
Please enter the sup-cost
200
Please enter the Process account number you want to update:
2
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 6:

```

12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
6
Please enter the sup-cost
14
Please enter the Process account number you want to update:
2
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 7:

```

11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
7
Please enter the sup-cost
39
Please enter the Process account number you want to update:
2
Please enter the Assembly account number you want to update:
7
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 8:

```

12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
8
Please enter the sup-cost
90
Please enter the Process account number you want to update:
3
Please enter the Assembly account number you want to update:
8
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 9:

```

sample [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java (Nov 14, 2023, 7:51:22 PM) [pid: 14219]
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
9
Please enter the sup-cost
100
Please enter the Process account number you want to update:
3
Please enter the Assembly account number you want to update:
8
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Input 10:

```

sample [Java Application] /Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java (Nov 14, 2023, 7:51:22 PM) [pid: 14219]
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
8
Please enter the transaction number
10
Please enter the sup-cost
23
Please enter the Process account number you want to update:
3
Please enter the Assembly account number you want to update:
8
Please enter the Department account number you want to update:
4
Connecting to the database...
Dispatching the query...
Dispatching the query...
Dispatching the query...
Dispatching the query...

```

Output for Transaction and Updates Tables:

```

1  SELECT * FROM Transactions;
2  SELECT * FROM Updates;
3

```

Results Messages

	transaction_no	sup_cost	job_no
1	1	10	NULL
2	2	20	NULL
3	3	100	NULL
4	4	130	NULL
5	5	200	NULL
6	6	14	NULL
7	7	35	NULL
8	8	90	NULL
9	9	100	NULL
10	10	23	NULL

	transaction_no	account_no
1	1	1
2	1	4
3	1	7
4	2	1
5	2	4
6	2	7
7	3	1
8	3	4
9	3	7
10	4	2
11	4	4
12	4	7
13	5	2
14	5	4
15	5	7
16	6	2
17	6	4
18	6	7
19	7	2
20	7	4
21	7	7
22	8	3
23	8	4
24	8	8
25	9	3
26	9	4
27	9	8
28	10	3
29	10	4
30	10	8

Output for Assembly, Department, and Process accounts

```
1  SELECT * FROM Assembly_acc;
2  SELECT * FROM Department_acc;
3  SELECT * FROM Process_acc;
4
```

Results Messages

	account_no	start_of_acc	details_1
1	7	NULL	379
2	8	NULL	213
3	9	NULL	0
4	10	NULL	0

	account_no	start_of_acc	details_2
1	4	NULL	592
2	5	NULL	0
3	6	NULL	0

	account_no	start_of_acc	details_3
1	1	NULL	130
2	2	NULL	379
3	3	NULL	213

Task 6.9 Query 9 Screenshots:

Because an Assembly Account has a one to one relationship with Assembly, it just returns the cost the in the assembly_acc that has the associated assembly_no.

Input 1:

```

9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order).
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
9
Please enter the Assembly Number:
1
Connecting to the database...
Dispatching the query...
Total Cost: 379

```

Input 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current value.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
9
Please enter the Assembly Number:
2
Connecting to the database...
Dispatching the query...
Total Cost: 213

```

Input 3:

```

13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
9
Please enter the Assembly Number:
3
Connecting to the database...
Dispatching the query...
Total Cost: 0

```

Task 6.10 Query 10 Screenshots:

Input 1:

```

11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
18
Please enter the Department number:
1
Please enter start date
2023-11-12
Please enter end date
2023-11-13
Connecting to the database...
Dispatching the query...
Total labor timne: 5

```

Input 2:

```

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 o
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each pr
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
10
Please enter the Department number:
2
Please enter start date
2023-11-14
Please enter end date
2023-11-15
Connecting to the database...
Dispatching the query...
Total labor timne: 3

```

Input 3:

```

7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current val
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for ea
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
10
Please enter the Department number:
3
Please enter start date
2023-11-12
Please enter end date
2023-11-15
Connecting to the database...
Dispatching the query...
Total labor timne: 7

```

Task 6.11 Query 11 Screenshots:

Screenshot depicts all imports:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
11
Please enter the Assembly number:
1
Connecting to the database...
Dispatching the query...
Process ID: 1, Dep Num: 1, Process Type: Fit Process
Process ID: 2, Dep Num: 1, Process Type: Paint Process

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
11
Please enter the Assembly number:
3
Connecting to the database...
Dispatching the query...
Process ID: 4, Dep Num: 2, Process Type: Fit Process

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
11
Please enter the Assembly number:
4
Connecting to the database...
Dispatching the query...
Process ID: 5, Dep Num: 2, Process Type: Paint Process

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!

```

Task 6.12 Query 12 Screenshots:

Import 1 and 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and assoc
5) Create a new account and associate it with the process, assembly, or department to which it is applic
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of j
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts b
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a give
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to
17) Exit!
12
Please enter the lower limit of the category range:
1
Please enter the upper limit of the category range:
3
Connecting to the database...
Dispatching the query...
Customer Name: Jacob, Address: 90210 Bev Hills, Category: 2
Customer Name: John, Address: 6808 Topsfield Dr, Category: 1
Customer Name: Michelle, Address: 2094 Michelle Ln, Category: 2
Customer Name: Sofia, Address: 6210 Del Norte Ln, Category: 1

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and assoc
5) Create a new account and associate it with the process, assembly, or department to which it is applic
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of j
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts b
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a give
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order)
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to
17) Exit!
12
Please enter the lower limit of the category range:
1
Please enter the upper limit of the category range:
5
Connecting to the database...
Dispatching the query...
Customer Name: Jacob, Address: 90210 Bev Hills, Category: 2
Customer Name: John, Address: 6808 Topsfield Dr, Category: 1
Customer Name: Matthew, Address: 3497 Hanover Ave, Category: 5
Customer Name: Michelle, Address: 2094 Michelle Ln, Category: 2
Customer Name: Sofia, Address: 6210 Del Norte Ln, Category: 1

```

Import 3:

```
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
12
Please enter the lower limit of the category range:
3
Please enter the upper limit of the category range:
5
Connecting to the database...
Dispatching the query...
Customer Name: Matthew, Address: 3497 Hanover Ave, Category: 5
```

Task 6.13 Query 13 Screenshots:

Import 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
13
Please enter the lower limit of the Job number range:
1
Please enter the upper limit of the Job number range:
3
Connecting to the database...
Cut Jobs in range are deleted.

1 rows affected.
Below are the remaining cut jobs from the table:
Cut Job Number: 6, Cut Labor Time: 1, Cut Machine: cut machine, Cut Machine Time: 1, Cut Material: cut material
Cut Job Number: 9, Cut Labor Time: 2, Cut Machine: cut machine, Cut Machine Time: 1, Cut Material: cut material

```

Import 2 and 3:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
13
Please enter the lower limit of the Job number range:
5
Please enter the upper limit of the Job number range:
6
Connecting to the database...
Cut Jobs in range are deleted.

1 rows affected.
Below are the remaining cut jobs from the table:
Cut Job Number: 9, Cut Labor Time: 2, Cut Machine: cut machine, Cut Machine Time: 1, Cut Material: cut material

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
13
Please enter the lower limit of the Job number range:
6
Please enter the upper limit of the Job number range:
10
Connecting to the database...
Cut Jobs in range are deleted.

1 rows affected.
Below are the remaining cut jobs from the table:

```

Output of affect tables:

```
3  SELECT * FROM Job;
4  SELECT * FROM Cut_Job;
5
6
```

Results Messages

	job_no	start_date	end_date	additional_info	process_id	assembly_no
1	4	2023-11-12	2023-11-13		4	3
2	5	2023-11-13	2023-11-14		5	4
3	7	2023-11-14	2023-11-15		7	6
4	8	2023-11-14	2023-11-15		8	7
5	10	2023-11-14	2023-11-15		10	9
6	1	2023-11-12	2023-11-13	information	1	1
7	2	2023-11-12	2023-11-13		2	1

	job_no	cut_labor_time	machine_used	time_of_machin...	material

Task 6.14 Query 14 Screenshots:

Import 1:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
14
Please enter the Paint Job Number:
2
Please enter the color you wish to change it to:
Purple
Connecting to the database...
Dispatching the query...
Paint color successfully changed

```

Import 2:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dat
5) Create a new account and associate it with the process, assembly, or department to
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commen
7) At the completion of a job, enter the date it completed and the information relevan
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the a
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the depart
11) Retrieve the processes through which a given assembly-id has passed so far (in dat
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range
17) Exit!
14
Please enter the Paint Job Number:
5
Please enter the color you wish to change it to:
Blue
Connecting to the database...
Dispatching the query...
Paint color successfully changed

```

Import 3:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current value.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
14
Please enter the Paint Job Number:
8
Please enter the color you wish to change it to:
Pink
Connecting to the database...
Dispatching the query...
Paint color successfully changed

```

Output of Affected Paint table:

```

1  SELECT * FROM Paint_Job;
2
3
4
5

```

Results Messages

	job_no	color	volume	paint_labor_time
.	2	Purple	2	2
.	5	Blue	1	2
.	8	Pink	1	2

Task 6.15 Query 15 Screenshots:

```

Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
15
Do you want to import? Press 1:
1
Please enter the path of the file import
/Users/sofiaschnurrenberger/Desktop/customer_table.csv

```

Output from importing:

1 **SELECT * FROM Customer;**

2

3

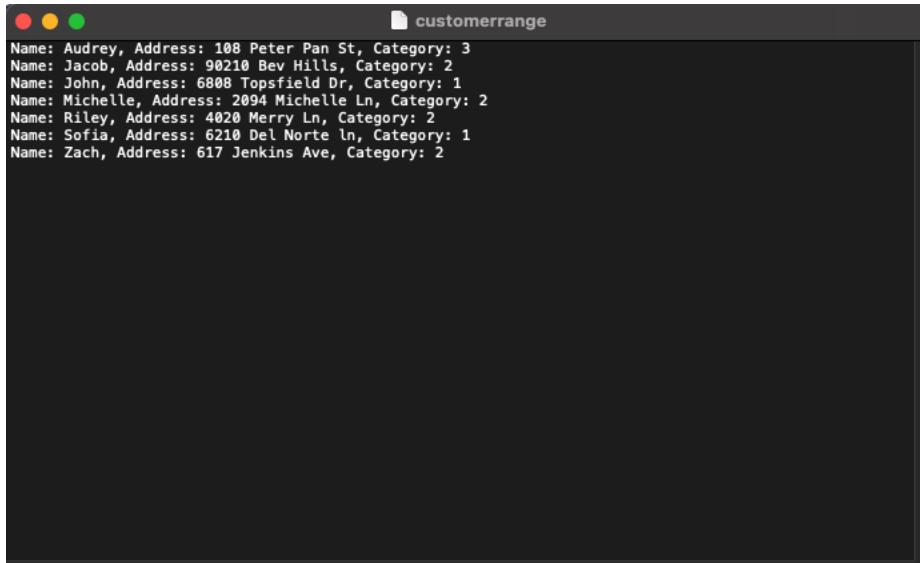
Results Messages

	cname	address	category
1	Audrey	108 Peter Pan St	3
2	Jacob	90210 Bev Hills	2
3	John	6808 Topsfield Dr	1
4	Matthew	3497 Hanover Ave	5
5	Michelle	2094 Michelle Ln	2
6	Riley	4020 Merry Ln	2
7	Sofia	6210 Del Norte ln	1
8	Zach	617 Jenkins Ave	2
9	Zoe	6808 Topsfield dr	4

Task 6.15 Query 16 Screenshots:

```
Please select one of the options below:
1) Enter a new customer.
2) Enter a new department.
3) Enter new process_id and its department together with its type and info relevant to the type.
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost.
9) Retrieve the total cost incurred on an assembly-id.
10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department to which it is applicable.
12) Retrieve the customers (in name order) whose category is in a given range.
13) Delete all cut-jobs whose job-no is in a given range.
14) Change the color of a given paint job.
15) Import: enter new customers from a data file until the file is empty.
16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead.
17) Exit!
16
Enter the lower limit of the category range:
1
Enter the upper limit of the category range:
3
Enter the output file name:
customerrange
Export completed. Data saved to: customerrange
```

File:



```
Name: Audrey, Address: 108 Peter Pan St, Category: 3
Name: Jacob, Address: 90210 Bev Hills, Category: 2
Name: John, Address: 6808 Topsfield Dr, Category: 1
Name: Michelle, Address: 2094 Michelle Ln, Category: 2
Name: Riley, Address: 4020 Merry Ln, Category: 2
Name: Sofia, Address: 6210 Del Norte Ln, Category: 1
Name: Zach, Address: 617 Jenkins Ave, Category: 2
```

Task 6.16 Error Screenshots:

(Query 3) If user tries to type in an already existing process id....

```

16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen
17) Exit!
3
Please enter the Process ID:
3
Please enter Process data, if none type NULL:
data
Please enter the Department ID you want to associate with this new process:
2
Connecting to the database...
Dispatching the query...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Process_9446C3F01827904D'. Cannot insert duplicate key in object 'dbo.Process'.
        at com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1673)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:620)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PreparedStmtExecCmd.doExecute(SQLServerPreparedStatement.java:540)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCmd(SQLServerConnection.java:627)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:268)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:242)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:486)
        at sample.main(sample.java:188)

```

(Query 6) If user enters an assembly_no not associated with the process id...

```

16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
17) Exit!
6
Please enter the job_no
12
Please enter the date the job commenced:
2023-11-14
Please enter job information:
info
Please enter the assembly id
4
Please enter the process id
12
Connecting to the database...
Dispatching the query...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Error: The assembly number entered is not associated with the process_id.
        at com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1673)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:620)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PreparedStmtExecCmd.doExecute(SQLServerPreparedStatement.java:540)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCmd(SQLServerConnection.java:7627)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3912)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:540)

```

(Query 7) If user wants to create a type of job that doesn't fit the type of process...

```

17) Exit!
7
Please enter the job_no that is completed:
12
Please enter the date the job completed:
2023-11-14
Please enter which type of Job completed (fit/paint/cut)
paint
Connecting to the database...
Dispatching the query...
Please enter the paint labor time:
3
Please enter the paint color:
red
Please enter the paint volume:
1
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Error: The specified process_id is not associated with a Paint process.
        at com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1673)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:620)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PreparedStmtExecCmd.doExecute(SQLServerPreparedStatement.java:540)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCmd(SQLServerConnection.java:7627)
        at com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3912)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:268)
        at com.microsoft.sqlserver.jdbc.SQLServerStatement.executeUpdate(SQLServerStatement.java:242)
        at com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:486)
        at sample.main(sample.java:547)

```

Task 6.17 Query 17 Screenshots:

Please select one of the options below:

- 1) Enter a new customer.
- 2) Enter a new department.
- 3) Enter new process_id and its department together with its type and info relevant to the type.
- 4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and dateordered and associate it with one or more processes.
- 5) Create a new account and associate it with the process, assembly, or department to which it is applicable.
- 6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
- 7) At the completion of a job, enter the date it completed and the information relevant to the type of job.
- 8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details.
- 9) Retrieve the total cost incurred on an assembly-id.
- 10) Retrieve the total labor time within a department for jobs completed in the department during a given date.
- 11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process.
- 12) Retrieve the customers (in name order) whose category is in a given range.
- 13) Delete all cut-jobs whose job-no is in a given range.
- 14) Change the color of a given paint job.
- 15) Import: enter new customers from a data file until the file is empty.
- 16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen.
- 17) Exit!

17
Exiting! Good-bye!

Task 7: Web database application source program and its execution

7.1 Web database application source program and screenshots showing successful compilation

Datahandler.java

```

package jsp_azure_test;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class DataHandler {
    private Connection conn;
    // Azure SQL connection credentials

    // Database credentials
    final static String HOSTNAME = "<schn0050>-sql-server.database.windows.net";
    final static String DBNAME = "schn0050";
    final static String USERNAME = "<schn0050>";
    final static String PASSWORD = "<Sofiase10>";
    final private String url =
String.format("jdbc:sqlserver://schn0050.database.windows.net:1433;database=schn0050;user=s
chn0050@schn0050;password={ Sofiase10};encrypt=true;trustServerCertificate=false;hostName
InCertificate=*.database.windows.net;loginTimeout=30;");
    // Initialize and save the database connection
    private void getDBConnection() throws SQLException {
        if (conn != null) {
            return;
        }
        this.conn = DriverManager.getConnection(url);
    }
    // Return the result of selecting everything from the movie_night table
    public ResultSet getAllCustomers() throws SQLException {
        getDBConnection();

        final String sqlQuery = "SELECT * FROM Customer";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }
    // Inserts a record into the movie_night table with the given attribute values
    public boolean addCustomer(
        String Name, String Address, int Category) throws
        SQLException {
        getDBConnection(); // Prepare the database connection
    }
}

```

```

        // Prepare the SQL statement
        final String sqlQuery =
        "INSERT INTO Customer " +
        "(cname, address, category) " +
        "VALUES " +
        "(?, ?, ?)";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        // Replace the '?' in the above statement with the given attribute
values
        stmt.setString(1, Name);
        stmt.setString(2, Address);
        stmt.setInt(3, Category);
        // Execute the query, if only one record is updated, then we
indicate success by returning true
        return stmt.executeUpdate() == 1;
    }

public ResultSet retrieveCustomer(
        int lower_number, int upper_number) throws
SQLException {
    getDBConnection(); // Prepare the database connection
    // Prepare the SQL statement
    final String sqlQuery = "SELECT * FROM Customer WHERE
category BETWEEN ? AND ? ORDER BY cname; ";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
    // Replace the '?' in the above statement with the given attribute
values
    stmt.setInt(1, lower_number);
    stmt.setInt(2, upper_number);

    // Execute the query, if only one record is updated, then we
indicate success by returning true
    return stmt.executeQuery();
}
}

```

Query1.jsp

```

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

```

```

<h4>Address</h4>
</td>
<td align="center">
<h4>Category</h4>
</td>
</tr>
<%
while(customers.next()) { // For each movie_night record
returned...
// Extract the attribute values for every row returned
final String Name = customers.getString("cname");
final String Address =
customers.getString("address");
final int Category = customers.getInt("category");

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

```

```
</html>
```

Add_query1.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```

<title>Query Result</title>
</head>
<body>
<%@page import="jsp_azure_test.DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
DataHandler handler = new DataHandler();

// Get the attribute values passed from the input form.

String name = request.getParameter("cname");
String address = request.getParameter("address");
String Category = request.getParameter("category");

/*
 * If the user hasn't filled out all the time, movie name and duration. This is very
simple checking.
*/
if (name == null || address == null || Category == null || name.equals("") ||
address.equals("") || Category.equals("")) {
    response.sendRedirect("add_query1_form.jsp");
} else {
    int categoryInt = Integer.parseInt(Category);
    boolean success = handler.addCustomer(name, address,
categoryInt);
    if (!success) { // Something went wrong
%
}

```

```
<h2>There was a problem inserting the course</h2>
<%
} else { // Confirm success to the user
%
<h2>Customers:</h2>
<ul>
<li>Name: <%=name%></li>
<li>Address: <%=address%></li>
<li>Category: <%=Category%></li>
</ul>
<h2>Was successfully inserted.</h2>

<a href= "Query1.jsp">See all Customers.</a>
<%
}
}
%
</body>
</html>
```

Add_query1_form.jsp

```
<!DOCTYPE html>

<html>
<head>
<meta charset="UTF-8">
<title>Welcome!</title>
</head>
<body>
<h2>Add Customer</h2>
<form action="add_query1.jsp" method="post">
<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="cname">
</div>

</td>
</tr>

<tr>

    <td> Address: </td>
    <td>
        <div style="text-align: center;">
            <input type="text" name="address">
        </div>
    </td>
</tr>

<tr>

    <td> Category: </td>
    <td>
        <div style="text-align: center;">
            <input type="text" name="category">
        </div>
    </td>
</tr>

<tr>
    <td>
        <div style="text-align: center;">
            <input type="reset" value="Clear">
        </div>
    </td>
</tr>
```

```
</td>

<td>

<div style="text-align: center;">
    <input type= "submit" value= "Insert">
</div>

</td>

</tr>

</table>

</form>

</body>

</html>
```

Query12.jsp

```

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

<%
// Retrieve values from request attributes
String lowerCategoryStr = request.getParameter("lcategory");
String upperCategoryStr = request.getParameter("ucategory");

// We instantiate the data handler here, and get all the movies from the database
final DataHandler handler = new DataHandler();
if (lowerCategoryStr == null || upperCategoryStr == null || lowerCategoryStr.equals("") ||
|| upperCategoryStr.equals("")) {

```

```

response.sendRedirect("add_query12_form.jsp");
} else {
    int upperCategory= Integer.parseInt(upperCategoryStr);
    int lowerCategory = Integer.parseInt(lowerCategoryStr);

    ResultSet customer_range =
handler.retrieveCustomer(lowerCategory,upperCategory);
%>

<!-- The table for displaying all the movie records -->
<table cellspacing="2" cellpadding="2" border="1">
<tr> <!-- The table headers row -->
<td align="center">
    <h4>Customer Name</h4>
</td>
<td align="center">
    <h4>Address</h4>
</td>
<td align="center">
    <h4>Category</h4>
</td>
</tr>
<%
    while (customer_range.next()) { // For each movie_night record returned...
        // Extract the attribute values for every row returned
        final String Name = customer_range.getString("cname");
        final String Address = customer_range.getString("address");

```

```

final int Category = customer_range.getInt("category");

out.println("<tr>"); // Start printing out the new table row
out.println( // Print each attribute value
    "<td align=\"center\">" + Name +
    "</td><td align=\"center\">" + Address +
    "</td><td align=\"center\">" + Category + "</td>");
out.println("</tr>");

}
}

%>

</table>
</body>
</html>

```

Add_query_12_form.jsp

```

<!DOCTYPE html>

<html>
<head>
    <meta charset="UTF-8">
    <title>Welcome!</title>
</head>
<body>
    <h2>Add Customer</h2>
    <form action="Query12.jsp" method="post">
        <table border="1">

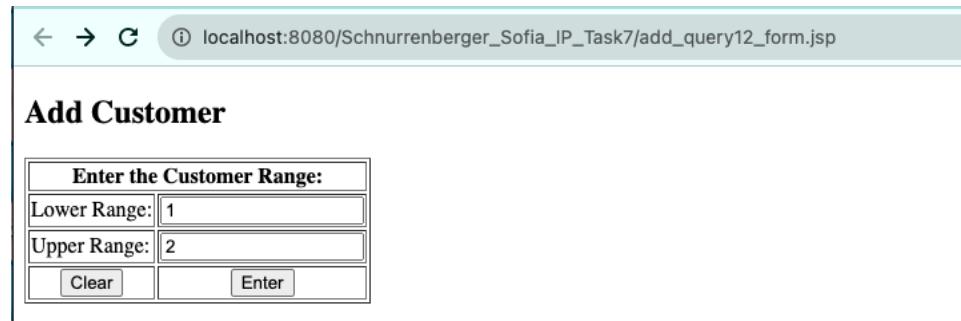
```

```
<tr>
  <th colspan="2">Enter the Customer Range:</th>
</tr>
<tr>
  <td> Lower Range: </td>
  <td>
    <div style="text-align: center;">
      <input type="text" name="lcategory">
    </div>
  </td>
</tr>
<tr>
  <td> Upper Range: </td>
  <td>
    <div style="text-align: center;">
      <input type="text" name="ucategory">
    </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="Enter">  
</div>  
</td>  
</tr>  
</table>  
</form>  
</body>  
</html>
```

7.2 Screenshots showing the testing of the Web database application

Query 12:



← → ⌂ ⓘ localhost:8080/Schnurrenberger_Sofia_IP_Task7/add_query12_form.jsp

Add Customer

Enter the Customer Range:

Lower Range:	1
Upper Range:	2
<input type="button" value="Clear"/>	<input type="button" value="Enter"/>

Customer Name	Address	Category
Jacob	90210 Bev Hills	2
John	6808 Topsfield Dr	1
Michelle	2094 Michelle Ln	2
Riley	4020 Merry Ln	2
sdfg	sdfg	1
Sofia	6210 Del Norte Ln	1
Zach	617 Jenkins Ave	2

Query 1:

Customers:

- Name: Mia
- Address: 1234 Windsor Way
- Category: 5

Was successfully inserted.

[See all Customers.](#)

Customer Name	Address	Category
Audrey	108 Peter Pan St	3
Isabella	Loverslane	5
Jacob	90210 Bev Hills	2
John	6808 Topsfield Dr	1
Matthew	3497 Hanover Ave	5
Mia	1234 Windsor Way	5
Michelle	2094 Michelle Ln	2
Riley	4020 Merry Ln	2
sdfg	sdfg	1
Sofia	6210 Del Norte ln	1
Zach	617 Jenkins Ave	2
Zoe	6808 Topsfield dr	4

Query 12 again:

Add Customer

Enter the Customer Range:	
Lower Range:	3
Upper Range:	5
<input type="button" value="Clear"/>	<input type="button" value="Enter"/>

Customer Name	Address	Category
Audrey	108 Peter Pan St	3
Isabella	Loverslane	5
Matthew	3497 Hanover Ave	5
Mia	1234 Windsor Way	5
Zoe	6808 Topsfield dr	4

