

# **Databases Individual Project**

**Tho Trinh 113383811**  
[tho.q.trinh-1@ou.edu](mailto:tho.q.trinh-1@ou.edu)

**Database Management Systems**  
**CS4513-001 Fall 2021**  
**Dr. Le-Gruenwald**

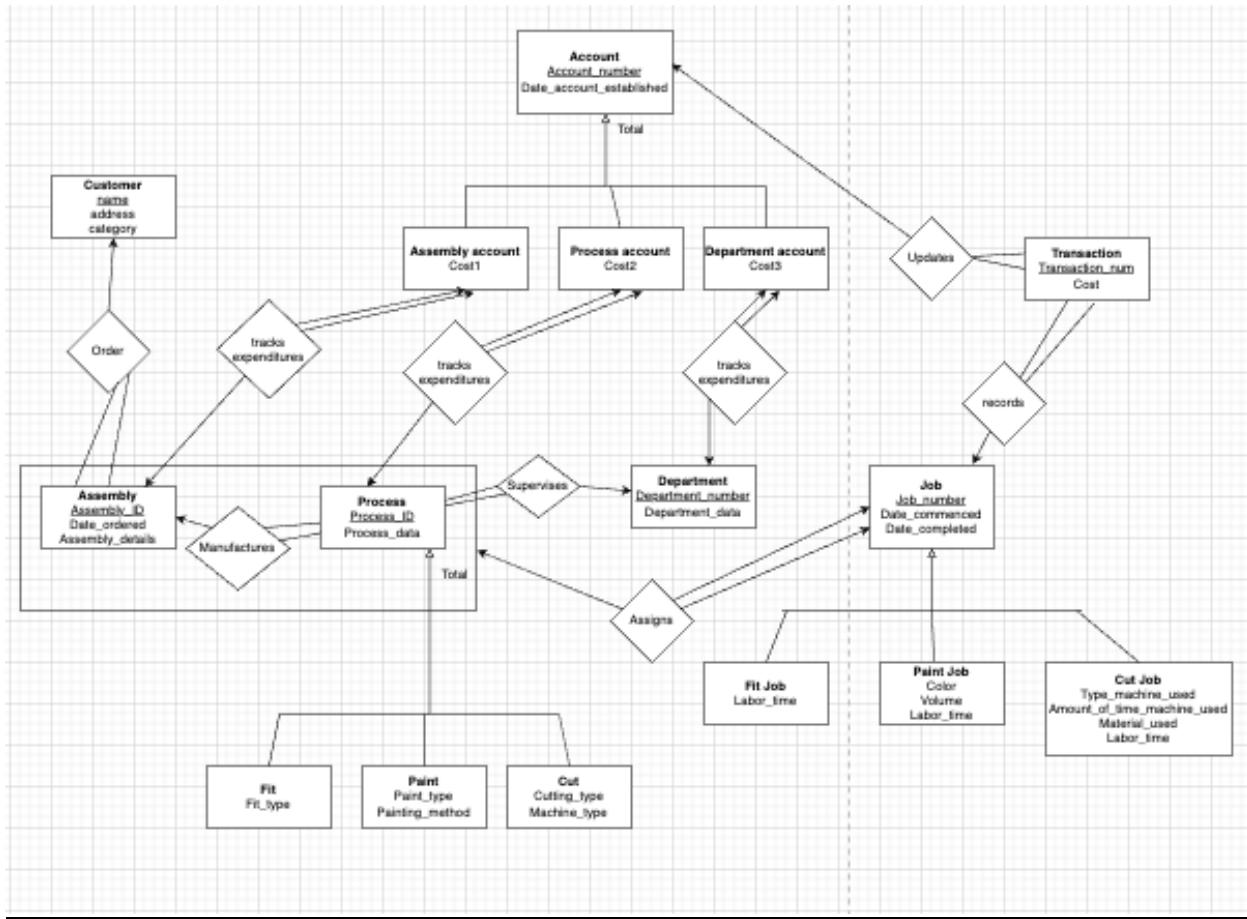
<b>Tasks Performed</b>	<b>Page Number</b>
Table of contents	2-3
Task 1.	
1.1 ER Diagram	4
1.2 Relational Database Schema	4-5
Task 2. Data Dictionary	5-9
Task 3.	
3.1 Discussion of storage structures for table	9-15
3.2 Discussion of storage structures for tables (Azure SQL Database)	15
Task 4. SQL statements and showing the creation of tables in Azure SQL Database	15-20
Task 5.	
5.1 The Java source program and screenshots showing its successful compilation	20-46
Task 6. Java program Execution	
6.1 Screenshots showing the testing of query 1	46-48
6.2 Screenshots showing the testing of query 2	48-51
6.3 Screenshots showing the testing of query 3	51-57
6.4 Screenshots showing the testing of query 4	58-65
6.5 Screenshots showing the testing of query 5	65-72
6.6 Screenshots showing the testing of query 6	73-77
6.7 Screenshots showing the testing of query 7	77-84
6.8 Screenshots showing the testing of query 8	84-90
6.9 Screenshots showing the testing of query 9	90-91
6.10 Screenshots showing the testing of query 10	91-92
6.11 Screenshots showing the testing of query 11	92-93
6.12 Screenshots showing the testing of query 12	93-95
6.13 Screenshots showing the testing of query 13	95-96
6.14 Screenshots showing the testing of query 14	96-98
6.15 Screenshots showing the testing of query 15	98-101
6.16 Screenshots showing the testing of query 16	101-102
6.17 Screenshots showing the testing of query 17	102-103
6.18 Screenshots showing the testing of query 18	103-104
6.19 Error Checking	104-106
Task 7 Web database application and its execution	
7.1 Web database application source program and screenshots showing its	107-112

successful compilation

7.2 Screenshots showing the testing of the Web database application

113-116

## ER Diagram



## Relational Database Schema

Customer(name, address, category)  
 Assembly1(assembly\_id, date\_ordered, assembly\_details, name)  
 Process(process\_id, process\_data, assembly\_id, department\_number)  
 Fit\_Process(process\_id, process\_data, assembly\_id, department\_number, fit\_type)  
 Paint\_Process(process\_id, process\_data, assembly\_id, department\_number, paint\_type, painting\_method)  
 Cut\_Process(process\_id, process\_data, assembly\_id, department\_number, cutting\_type, machine\_type)  
 Department(department\_number, department\_data)  
 Job(job\_number, date\_commenced, date\_completed, assembly\_id, process\_id)  
 Fit\_Job(job\_number, labor\_time)  
 Paint\_Job(job\_number, labor\_time, color, volume)

Cut\_Job(job\_number, labor\_time, type\_machine\_used, amount\_of\_time\_machine\_used, material\_used)  
 Transaction(transanction\_num, cost, job\_number, account\_number)  
 Account(account\_number, date\_account\_established)  
 Assembly\_Account(account\_number, date\_account\_established, cost1, assembly\_id)  
 Process\_Account(account\_number, date\_account\_established, cost2, process\_id)  
 Department\_Account(account\_number, date\_account\_established, cost3, department\_number)

## Data Dictionary

<b>Customer</b>			
<b>Variable</b>	<b>Type</b>	<b>Size (bytes)</b>	<b>Constraints</b>
Name	varchar(64)	66	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Address	varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Category	int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Check if integer from 1-10 inclusive</li> </ul>

<b>Assembly</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_ordered	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Assembly_details	Varchar(256)	258	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Name	varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES customer</li> </ul>

<b>Process</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Process_data	Varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Assembly</li> </ul>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Department</li> </ul>

<b>Fit Process</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Process_data	Varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Assembly</li> </ul>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Department</li> </ul>
Fit_type	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Paint Process</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Process_data	Varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Assembly</li> </ul>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Department</li> </ul>
Paint_type	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Painting_method	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Cut Process</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Process_data	Varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Assembly</li> </ul>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Department</li> </ul>
Cutting_type	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Machine_type	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Department</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Department_data	Varchar(128)	130	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Job</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Job_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_commenced	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Data_completed	Date	3	
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Process</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Assembly</li> </ul>

<b>Fit_Job</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Job_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Labor_time	Time	5	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Paint_Job</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Job_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Labor_time	Time	5	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Color	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Volume	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Cut_Job</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Job_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Labor_time	Time	5	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Type_machine_used	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Amount_of_time_machine_used	Time	5	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Material_used	Varchar(64)	66	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Transaction</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Transaction_num	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>

Cost	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Job_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Job</li> </ul>
Account_number	Int	4	<ul style="list-style-type: none"> <li>• Not Null</li> <li>• Foreign Key REFERENCES Account</li> </ul>

<b>Account</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Account_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_account_established	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>

<b>Assembly_Account</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Account_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_account_established	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Cost1	Int	4	<ul style="list-style-type: none"> <li>• Default = 0</li> </ul>
Assembly_id	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Assembly</li> </ul>

<b>Process Account</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Account_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_account_established	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Cost2	Int	4	<ul style="list-style-type: none"> <li>• Default = 0</li> </ul>
Process_ID	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Process</li> </ul>

<b>Department_Account</b>			
<b>Variable</b>	<b>Type</b>	<b>Size</b>	<b>Constraints</b>
Account_number	Int	4	<ul style="list-style-type: none"> <li>• Primary Key</li> </ul>
Date_account_established	Date	3	<ul style="list-style-type: none"> <li>• Not Null</li> </ul>
Cost3	Int	4	<ul style="list-style-type: none"> <li>• Default = 0</li> </ul>
Department_number	Int	4	<ul style="list-style-type: none"> <li>• Foreign Key REFERENCES Department</li> </ul>

## Discussion of Storage Structures for table

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Customer	#1 – Insertion  #13 – Random Search (result must be in name order)	#1 – no search key  #13 – given range for category	#1 – 30/day  #13 – 100/day	B+ Tree Index File where search key is name.
<b>Justification</b>				
I choose a B+ tree due to the speed it offers due to the high number of insertions and searches for this table. I chose a B+ tree because it offers less storage and better speed than the B tree. Insertion shouldn't be too hard. However, it is necessary to retrieve customers in order by their name in a given category range. I chose to have the search key be name instead of category so that it will decrease the I/O of the table to return the customers in name order. It will have to linearly search the B+ tree to see if the category is in the specified range.				

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Assembly	#4 – Insertion	#4 – no search key	#4 – 40/day	Heap
<b>Justification</b>				
I chose a heap for this table as there is no need to ever search the table with the current queries. Therefore, I choose a heap as it offers the quickest insertion with the less storage.				

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Process	#3 – Insertion  #4 – Range Search  #10 – Random Search  # 11 – Range Search	#3 – no search key  #4 – process_id  #10 – department_number  #11 – process_id  #12 – department_number	#3 – infrequent  #4 – 40/day  #10 – 20/day  #11 – 100/day  #12 – 20/day	B+ tree where search key is process_id.

	#12 – Random search			
<b>Justification</b>				
I chose a B+ tree for the speed it offers with the high number of searches and insertions for this table. I chose the search key to be process_id due to the high number of searches that will need to use process_id. This table also has searches for department_number, however, it is a less frequent query than the searches for process_id.				

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Fit Process	#3 – Insertion  #4 – Range Search	#3 – no search key  #4 – process_id	#3 – infrequent  #4 – 40/day	B+ tree where search key is process_id.

<b>Justification</b>				
I chose a B+ tree for the speed it offers with the high number of searches and insertions for this table. I chose the search key to be process_id due to the high number of searches that will need to use process_id. This table only has searches done on it for process_id. This table will only be searched when the Process table is search. The search on this table is used to find more specific details about a certain process_id.				

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization

Paint Process	#3 – Insertion  #4 – Range Search	#3 – no search key  #4 – process_id	#3 – infrequent  #4 – 40/day	B+ tree where search key is process_id.
---------------	---	--	---------------------------------------	---

**Justification**

I chose a B+ tree for the speed it offers with the high number of searches and insertions for this table. I chose the search key to be process\_id due to the high number of searches that will need to use process\_id. This table only has searches done on it for process\_id. This table will only be searched when the Process table is search. The search on this table is used to find more specific details about a certain process\_id.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Cut Process	#3 – Insertion  #4 – Range Search	#3 – no search key  #4 – process_id	#3 – infrequent  #4 – 40/day	B+ tree where search key is process_id.

**Justification**

I chose a B+ tree for the speed it offers with the high number of searches and insertions for this table. I chose the search key to be process\_id due to the high number of searches that will need to use process\_id. This table only has searches done on it for process\_id. This table will only be searched when the Process table is search. The search on this table is used to find more specific details about a certain process\_id.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Department	#2 – Insertion	#2 – no search key	#2 – infrequent	Heap

**Justification**

A heap would be best for this table since it only has insertions and no searches, updates, or deletes. Thus, a heap would offer the fastest insertion since it only associates with an insertion query.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Job	#6 – Insertion  #7 – Range Search  #10 – Random Search  #11 – Range search  #12 – Random Search	#6 – no search key  #7 – job_number  #10 – process_id  #11 – date_commenced  #12 – date_completed	#6 – 50/day  #7 – 50/day  #10 – 20/day  #11 – 100/day  #12 – 20/day	I choose an indexed sequential files with a primary index on job number, secondary index on date commenced.

#### Justification

I chose an indexed sequential file because of the different query request that this table has to go through. First of all, query #11 will only be used on this table because that query doesn't require knowledge about job type (just date commenced, process\_id, and assembly\_id). I choose a primary index on job\_number which will be necessary for query #7. And, I chose to have a secondary index on date\_commenced since query#11 is done 100/day. The other two queries are rather infrequent in comparison so another index is not needed.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Fit Job	#7 – Insertion  #10 – Range Search  #12 – Range Search	#7 – none  #10 – job_number  #12 – job_number	#7 – 50/day  #10 – 20/day  #12 – 20/day	B+ tree where search key is job_number.

#### Justification

I chose a B+ tree on job\_number as the job\_number is the most used search key out of the queries that this table goes through. I chose a B+ tree because of the speed it offers as well as the storage advantages that it offers.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Paint Job	#7 – Insertion	#7 – none	#7 – 50/day	B+ tree where search key is job_number

	#10 – Range Search  #12 – Range Search  #15 – Range Search	#10 – job_number  #12 – job_number  #15 – job_number	#10 – 20/day  #12 – 20/day  #15 – 1/week	
--	--	--	--	--

**Justification**

I chose a B+ tree on job\_number as the job\_number is the most used search key out of the queries that this table goes through. I chose a B+ tree because of the speed it offers as well as the storage advantages that it offers.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Cut Job	#7 – Insertion  #10 – Range Search  #12 – Range Search  #14 – Random Search	#7 – none  #10 – job_number  #12 – job_number  #14 – job_number	#7 – 50/day  #10 – 20/day  #12 – 20/day  #14 – 1/month	B+ tree where search key is job_number

**Justification**

I chose a B+ tree on job\_number as the job\_number is the most used search key out of the queries that this table goes through. I chose a B+ tree because of the speed it offers as well as the storage advantages that it offers.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Transaction	#8 – Insertion	#8 – no search key	#8 – 50/day	Heap file.

**Justification**

I chose a heap file for this table because there are no searches being done on this table. A heap file offers the quickest insertion.

<b>Table Name</b>	<b>Query# and Type</b>	<b>Search Key</b>	<b>Query Frequency</b>	<b>Selected File Organization</b>
Account	#5 – Insertion	#5 – no search key number	#5 – 10/day	Heap file.

#### **Justification**

Since this table is only involved in insertions, there is no need to create an index on this table. A heap would offer the best insertion time.

<b>Table Name</b>	<b>Query# and Type</b>	<b>Search Key</b>	<b>Query Frequency</b>	<b>Selected File Organization</b>
Assembly Account	#5 – Insertion  #8 – Range Search  #9 – Range Search	#5 – no search key  #8 – account_number  #9 – account_number	#5 – 10/day  #8 – 50/day  #9 – 200/day	B+ tree where search key is account number.

#### **Justification**

This table will be searched in a similar manner to the Department table. Since this table utilizes both insertion and search, it is beneficial to use a B+ tree as that will give us the fastest search time with minimal storage. The search key will be on account\_number since that is the only value that we are searching on.

<b>Table Name</b>	<b>Query# and Type</b>	<b>Search Key</b>	<b>Query Frequency</b>	<b>Selected File Organization</b>
-------------------	------------------------	-------------------	------------------------	-----------------------------------

Process Account	#5 – Insertion  #8 – Range Search	#5 – no search key  #8 – account_number	#5 – 10/day  #8 – 50/day	B+ tree where search key is account number.
-----------------	---	---	--------------------------------	---

#### Justification

This table will be searched in a similar manner to the Department table. Since this table utilizes both insertion and search, it is beneficial to use a B+ tree as that will give us the fastest search time with minimal storage. The search key will be on account\_number since that is the only value that we are searching on.

Table Name	Query# and Type	Search Key	Query Frequency	Selected File Organization
Department Account	#5 – Insertion  #8 – Range Search	#5 – no search key  #8 – account_number	#5 – 10/day  #8 – 50/day	B+ tree where search key is account number.

#### Justification

This table will be searched in a similar manner to the Department table. Since this table utilizes both insertion and search, it is beneficial to use a B+ tree as that will give us the fastest search time with minimal storage. The search key will be on account\_number since that is the only value that we are searching on.

## Discussion of storage structures for tables (Azure SQL Database)

Azure SQL does not allow the option to create many storage structures. Every time that Azure makes an index, it creates a B+ tree to index the table. Therefore, Azure can only do 3 file organizations: B+ tree, clustered B+ tree, and a heap. Azure allows the option to create a clustered index. Also, if you don't create an index on the table, it will be inserted as a heap. Therefore, I couldn't create any hash-indexed files or anything other than the 3 choices that Azure offers me.

## SQL Program for creation of tables and indexes

-- While working on the database design, it's useful to start from scratch every time

```

-- Hence, we drop tables in reverse order they are created (so the foreign key
constraints are not violated)
DROP TABLE IF EXISTS Transaction1;
DROP TABLE IF EXISTS Department_Account;
DROP TABLE IF EXISTS Process_Account;
DROP TABLE IF EXISTS Assembly_Account;
DROP TABLE IF EXISTS Account;
DROP TABLE IF EXISTS Cut_Job;
DROP TABLE IF EXISTS Paint_Job;
DROP TABLE IF EXISTS Fit_Job;
DROP TABLE IF EXISTS Job;
DROP TABLE IF EXISTS Cut_Process;
DROP TABLE IF EXISTS Paint_Process;
DROP TABLE IF EXISTS Fit_Process;
DROP TABLE IF EXISTS Process;
DROP TABLE IF EXISTS Assembly1;
DROP TABLE IF EXISTS Department;
DROP TABLE IF EXISTS Customer;

-- Create tables
CREATE TABLE Customer (
    name VARCHAR(64) PRIMARY KEY,
    address VARCHAR(128) NOT NULL,
    category INT NOT NULL,
    CONSTRAINT category_range CHECK(category < 11 AND category > 0)
)

CREATE TABLE Department (
    department_number INT PRIMARY KEY,
    department_data VARCHAR(128) NOT NULL
)

CREATE TABLE Assembly1 ( -- Renamed to Assembly1 because Assembly isn't allowed
    assembly_id INT PRIMARY KEY,
    date_ordered DATE NOT NULL,
    assembly_details VARCHAR(256) NOT NULL,
    name VARCHAR(64) NOT NULL,
    CONSTRAINT FK_customer_name FOREIGN KEY (name) REFERENCES Customer
)

CREATE TABLE Process (
    process_id INT PRIMARY KEY,
    process_data VARCHAR(128) NOT NULL,
    assembly_id INT,
    department_number INT NOT NULL,

```

```

CONSTRAINT FK_assembly_id FOREIGN KEY (assembly_id) REFERENCES Assembly1,
CONSTRAINT FK_department_number FOREIGN KEY (department_number) REFERENCES
Department
)

CREATE TABLE Fit_Process (
process_id INT PRIMARY KEY,
process_data VARCHAR(128) NOT NULL,
fit_type VARCHAR(64) NOT NULL,
assembly_id INT,
department_number INT NOT NULL,

CONSTRAINT FK_assembly_id2 FOREIGN KEY (assembly_id) REFERENCES Assembly1,
CONSTRAINT FK_department_number2 FOREIGN KEY (department_number) REFERENCES
Department
)

CREATE TABLE Paint_Process (
process_id INT PRIMARY KEY,
process_data VARCHAR(128) NOT NULL,
paint_type VARCHAR(64) NOT NULL,
painting_method VARCHAR(64) NOT NULL,
assembly_id INT,
department_number INT NOT NULL,

CONSTRAINT FK_assembly_id3 FOREIGN KEY (assembly_id) REFERENCES Assembly1,
CONSTRAINT FK_department_number3 FOREIGN KEY (department_number) REFERENCES
Department
)

CREATE TABLE Cut_Process (
process_id INT PRIMARY KEY,
process_data VARCHAR(128) NOT NULL,
cutting_type VARCHAR(64) NOT NULL,
machine_type VARCHAR(64) NOT NULL,
assembly_id INT,
department_number INT NOT NULL,

CONSTRAINT FK_assembly_id4 FOREIGN KEY (assembly_id) REFERENCES Assembly1,
CONSTRAINT FK_department_number4 FOREIGN KEY (department_number) REFERENCES
Department
)

CREATE TABLE Job (
job_number INT PRIMARY KEY,
date_commenced DATE NOT NULL,
date_completed DATE,

```

```

process_id INT NOT NULL,
assembly_id INT NOT NULL,

CONSTRAINT FK_process_id FOREIGN KEY (process_id) REFERENCES Process,
CONSTRAINT FK_assembly_id5 FOREIGN KEY (assembly_id) REFERENCES Assembly1
)

CREATE TABLE Fit_Job (
    job_number INT PRIMARY KEY,
    labor_time TIME NOT NULL
)

CREATE TABLE Paint_Job (
    job_number INT PRIMARY KEY,
    labor_time TIME NOT NULL,
    color VARCHAR(64) NOT NULL,
    volume INT NOT NULL
)

CREATE TABLE Cut_Job (
    job_number INT PRIMARY KEY,
    labor_time TIME NOT NULL,
    type_machine_used VARCHAR(64) NOT NULL,
    amount_of_time_machine_used TIME NOT NULL,
    material_used VARCHAR(64) NOT NULL
)

CREATE TABLE Account (
    account_number INT PRIMARY KEY,
    date_account_established DATE NOT NULL
)

CREATE TABLE Assembly_Account (
    account_number INT PRIMARY KEY,
    date_account_established DATE NOT NULL,
    cost1 INT DEFAULT 0,
    assembly_id INT NOT NULL,

CONSTRAINT FK_assembly_id9 FOREIGN KEY (assembly_id) REFERENCES Assembly1
)

CREATE TABLE Process_Account (
    account_number INT PRIMARY KEY,
    date_account_established DATE NOT NULL,
    cost2 INT DEFAULT 0,
    process_id INT NOT NULL,

CONSTRAINT FK_process_id5 FOREIGN KEY (process_id) REFERENCES Process
)

```

```

)

CREATE TABLE Department_Account (
    account_number INT PRIMARY KEY,
    date_account_established DATE NOT NULL,
    cost3 INT DEFAULT 0,
    department_number INT NOT NULL

    CONSTRAINT FK_department_number5 FOREIGN KEY (department_number) REFERENCES
Department
)

CREATE TABLE Transaction1 ( -- Renamed Transaction1 because Transaction isn't allowed
    transaction_num INT PRIMARY KEY,
    cost INT NOT NULL,
    job_number INT NOT NULL,
    account_number INT NOT NULL,

    CONSTRAINT FK_job_number FOREIGN KEY (job_number) REFERENCES Job,
    CONSTRAINT FK_account_number FOREIGN KEY (account_number) REFERENCES Account
)
-- Create Indexes on tables -----
-- Customer Table Index on name
CREATE INDEX idx_customer
ON Customer (name);

-- Process Table Index on process_id
CREATE INDEX idx_process
ON Process (process_id);

-- Fit Process Table Index on process_id
CREATE INDEX idx_fit_process
ON Fit_Process (process_id);

-- Paint Process Table Index on process_id
CREATE INDEX idx_paint_process
ON Paint_Process (process_id);

-- Cut Process Table Index on process_id
CREATE INDEX idx_cut_process
ON Cut_Process (process_id);

-- Job Table with 2 Indexes
-- Primary Index on job_number; Secondary Index on date_commenced
CREATE INDEX idx_job
ON Job (job_number, date_commenced);

```

```

-- Fit Job Table Index on job_number
CREATE INDEX idx_fit_job
ON Fit_Job (job_number);

-- Paint Job Table Index on job_number
CREATE INDEX idx_paint_job
ON Paint_Job (job_number);

-- Cut Job Table Index on job_number
CREATE INDEX idx_cut_job
ON Cut_Job (job_number);

-- Assembly Account Table Index on account_number
CREATE INDEX idx_assembly_account
ON Assembly_Account (account_number);

-- Process Account Table Index on account_number
CREATE INDEX idx_process_account
ON Process_Account (account_number);

-- Department Account Table Index on account_number
CREATE INDEX idx_department_account
ON Department_Account (account_number);

```

## **Java source program for implementation of queries**

```

import java.io.File;
import java.io.FileWriter;
import java.sql.Connection;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DatabaseIP {

    // Database connection string
    final static String URL = "jdbc:sqlserver://trin0003-sql-
server.database.windows.net:1433;database=cs-dsa-4513-sql-db;user=trin0003@trin0003-sql-
server;password=ClearMap2013$;encrypt=true;trustServerCertificate=false;hostNameInCertifica-
te=*.database.windows.net;loginTimeout=30;";

    final static String QUERY_TEMPLATE_1 = "INSERT INTO Customer " +
        "VALUES (?, ?, ?);";

```

```

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

final static String QUERY_TEMPLATE_3A = "INSERT INTO Process " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_3FIT = "INSERT INTO Fit_Process " +
    "VALUES (?, ?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_3PAINT = "INSERT INTO Paint_Process " +
    "VALUES (?, ?, ?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_3CUT = "INSERT INTO Cut_Process " +
    "VALUES (?, ?, ?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_4 = "INSERT INTO Assembly1 " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_4B = "UPDATE Process\n" + "SET assembly_id = ?\n" + "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_4BFIT = "UPDATE Fit_Process\n" + "SET assembly_id = ?\n" + "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_4BPAINT = "UPDATE Paint_Process\n" + "SET assembly_id = ?\n" + "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_4BCUT = "UPDATE Cut_Process\n" + "SET assembly_id = ?\n" + "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_5A = "INSERT INTO Account " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_5B = "INSERT INTO Assembly_Account " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_5C = "INSERT INTO Process_Account " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_5D = "INSERT INTO Department_Account " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_6 = "INSERT INTO Job " +
    "VALUES (?, ?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_7 = "UPDATE Job\n" + "SET date_completed = ?\n" + "WHERE job_number = ?;";

final static String QUERY_TEMPLATE_7FIT = "INSERT INTO Fit_Job " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_7PAINT = "INSERT INTO Paint_Job " +
    "VALUES (?, ?, ?, ?, ?);";

```

```

final static String QUERY_TEMPLATE_7CUT = "INSERT INTO Cut_Job " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_8 = "INSERT INTO Transaction1 " +
    "VALUES (?, ?, ?, ?, ?);";

final static String QUERY_TEMPLATE_8ASSEMBLY = "UPDATE Assembly_Account\n" +
    "SET cost1 = cost1 + ?\n" + "WHERE account_number = ?;";

final static String QUERY_TEMPLATE_8PROCESS = "UPDATE Process_Account\n" +
    "SET cost2 = cost2 + ?\n" + "WHERE account_number = ?;";

final static String QUERY_TEMPLATE_8DEPARTMENT = "UPDATE
Department_Account\n" + "SET cost3 = cost3 + ?\n" + "WHERE account_number = ?;";

final static String QUERY_TEMPLATE_9 = "SELECT cost1 FROM Assembly_Account
WHERE assembly_id = ?;";

final static String QUERY_TEMPLATE_10FIT = "SELECT SUM(( DATEPART(hh,
labor_time) * 3600 ) + ( DATEPART(mi, labor_time) * 60 ) + DATEPART(ss, labor_time))/60
as minute\n" +
    "+ \"FROM Fit_Job WHERE Fit_Job.job_number in (\n\" +
    "+ "SELECT distinct(job_number) FROM Job\n" +
    "+ \"WHERE Job.process_id in (SELECT distinct(process_id) FROM Process WHERE
Process.department_number = ?) AND Job.date_completed = ? );\";

final static String QUERY_TEMPLATE_10PAINT = "SELECT SUM(( DATEPART(hh,
labor_time) * 3600 ) + ( DATEPART(mi, labor_time) * 60 ) + DATEPART(ss, labor_time))/60
as minute\n" +
    "+ \"FROM Paint_Job WHERE Paint_Job.job_number in (\n\" +
    "+ "SELECT distinct(job_number) FROM Job\n" +
    "+ \"WHERE Job.process_id in (SELECT distinct(process_id) FROM Process WHERE
Process.department_number = ?) AND Job.date_completed = ? );\";

final static String QUERY_TEMPLATE_10CUT = "SELECT SUM(( DATEPART(hh,
labor_time) * 3600 ) + ( DATEPART(mi, labor_time) * 60 ) + DATEPART(ss, labor_time))/60
as minute\n" +
    "+ \"FROM Cut_Job WHERE Cut_Job.job_number in (\n\" +
    "+ "SELECT distinct(job_number) FROM Job\n" +
    "+ \"WHERE Job.process_id in (SELECT distinct(process_id) FROM Process WHERE
Process.department_number = ?) AND Job.date_completed = ? );\";

final static String QUERY_TEMPLATE_11 = "SELECT Job.process_id,
Process.department_number, Job.date_commenced\n" +
    "+ \" FROM Job, Process\n" +
    "+ \" WHERE Job.assembly_id = ? AND Process.process_id = Job.process_id \n\" +
    "+ \" ORDER BY 1 ;\";
```

```

final static String QUERY_TEMPLATE_12A = "SELECT DISTINCT(Job.job_number),
Job.assembly_id, Fit_Job.labor_time\n"
+ "    FROM Job, Fit_Job\n"
+ "    WHERE date_completed = ? and Job.process_id in (SELECT process.process_id
FROM Process WHERE department_number = ?) AND Fit_Job.job_number =
Job.job_number;";

final static String QUERY_TEMPLATE_12B = "SELECT DISTINCT(Job.job_number),
Job.assembly_id, Paint_Job.color, Paint_Job.volume, Paint_Job.labor_time\n"
+ "    FROM Job, Paint_Job\n"
+ "    WHERE date_completed = ? and Job.process_id in (SELECT process.process_id
FROM Process WHERE department_number = ?) AND Paint_Job.job_number =
Job.job_number;";

final static String QUERY_TEMPLATE_12C = "SELECT DISTINCT(Job.job_number),
Job.assembly_id, Cut_Job.type_machine_used, Cut_Job.amount_of_time_machine_used,
Cut_Job.material_used, Cut_Job.labor_time\n"
+ "    FROM Job, Cut_Job\n"
+ "    WHERE date_completed = ? and Job.process_id in (SELECT process.process_id
FROM Process WHERE department_number = ?) AND Cut_Job.job_number =
Job.job_number;";

final static String QUERY_TEMPLATE_13 = "SELECT name, category AS name FROM
Customer\n"
+ "    WHERE category >= ? AND category <= ?\n"
+ "    ORDER BY 1 ;";

final static String QUERY_TEMPLATE_14 = "DELETE FROM Cut_Job WHERE
job_number >= ? AND job_number <= ?;";

final static String QUERY_TEMPLATE_15 = "UPDATE Paint_Job SET color = ? WHERE
job_number= ?;";

// User input prompt //
final static String PROMPT =
"\nWELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM \n" +
"(1) Enter a new customer \n" +
"(2) Enter a new department \n" +
"(3) Enter a new process and department together \n" +
"(4) Enter a new assembly and associate it with one or more processes \n" +
"(5) Create a new account and associate it with a process, assembly, or department \n" +
"(6) Enter a new job and date the job began\n" +
"(7) At the completion of a job, enter the date completed and relevant info \n" +
"(8) Enter a transaction and update all the costs of affected accounts with the transaction
cost \n" +
"(9) Retrieve the total cost incurred on a specific assembly \n" +

```

"(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date \n" +  
 "(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process \n" +  
 "(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department \n" +  
 "(13) Retrieve the customers(in name order) whose category is in a specific range \n" +  
 "(14) Delete all cut-jobs whose job-no is in a specific range \n" +  
 "(15) Change the color of a specific paint job \n" +  
 "(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name) \n" +  
 "(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen \n" +  
 "(18) Quit \n";

```

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

    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("18")) { // Ask user for options until option 3 is selected
        System.out.println(PROMPT); // Print the available options
        option = sc.nextLine(); // Read in the user option selection

        switch (option) { // Switch between different options
            case "1": // Enter a new Customer
                try { // In case of an error, this returns it to the main menu instead of terminating
                    program
                        // Collect data to input for the Customer
                        System.out.println("Please enter name of the new customer:");
                        sc.nextLine();
                        final String name = sc.nextLine();

                        System.out.println("Please enter address of the new customer:");

                        final String address = sc.nextLine();

                        System.out.println("Please enter Customer category (a number from 1-10): ");
                        final int category = sc.nextInt();

                        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_1)){
            // Make a new customer and populate the database with it.
            statement.setString(1, name);
            statement.setString(2, address);
            statement.setInt(3, category);

            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done. %d rows inserted.",
rows_inserted));
        }
    }
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

break;
case "2":
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for Department information
    System.out.println("Please enter new department number:");
    final int department_number = sc.nextInt();

    System.out.println("Please enter department data:");
    sc.nextLine();

    final String department_data = sc.nextLine();

try (final Connection connection = DriverManager.getConnection(URL)) {
    // Add department first
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_2)) {

        statement.setInt(1, department_number);
        statement.setString(2, department_data);

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.",
rows_inserted));
    }
}
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

```

```

    }

break;

case "3": // Enter a Process
try { // In case of an error, this returns it to the main menu instead of terminating
program
    System.out.println("Please enter a new process ID: ");
    final int process_id = sc.nextInt();

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

    System.out.println("Please enter new department number associated with
process:");
    final int department_number3 = sc.nextInt();

    System.out.println("Please enter type of process (Fit, Paint, Cut)? ");
    sc.nextLine();
    final String type = sc.nextLine();

    String fit_type = "";
    String paint_type = "";
    String paint_method = "";
    String cutting_type = "";
    String machine_type = "";

    if (type.equals("Fit")) {

        System.out.println("Please enter the corresponding fit_type: ");
        fit_type = sc.nextLine();

    } else if (type.equals("Paint")) {

        System.out.println("Please enter the corresponding paint_type: ");
        paint_type = sc.nextLine();
        System.out.println("Please enter the corresponding paint_method: ");
        paint_method = sc.nextLine();

    } else if (type.equals("Cut")) {

        System.out.println("Please enter the corresponding cutting type: ");
        cutting_type = sc.nextLine();
    }
}

```

```

        System.out.println("Please enter the corresponding machine type: ");
        machine_type = sc.nextLine();

    }

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Add department first
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_3A)) {

            statement.setInt(1, process_id);
            statement.setString(2, process_data);
            statement.setNull(3,java.sql.Types.INTEGER);
            statement.setInt(4, department_number3);

            final int rows_inserted2 = statement.executeUpdate();
            System.out.println(String.format("Done. %d rows inserted.",

rows_inserted2));

            if (type.equals("Fit")) {

                //Add fit Process next
                final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_3FIT);
                statement3.setInt(1, process_id);
                statement3.setString(2, process_data);
                statement3.setString(3,fit_type);
                statement3.setNull(4,java.sql.Types.INTEGER);
                statement3.setInt(5, department_number3);

                final int rows_inserted3 = statement3.executeUpdate();
                System.out.println(String.format("Done. %d rows inserted.",

rows_inserted3));

            } else if (type.equals("Paint")) {

                //Add paint Process next
                final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_3PAINT);
                statement3.setInt(1, process_id);
                statement3.setString(2, process_data);
                statement3.setString(3,paint_type);

```

```

        statement3.setString(4, paint_method);
        statement3.setNull(5,java.sql.Types.INTEGER);
        statement3.setInt(6, department_number3);

        final int rows_inserted3 = statement3.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.",

rows_inserted3));

    } else if (type.equals("Cut")) {

        //Add cut Process next
        final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_3CUT);
        statement3.setInt(1, process_id);
        statement3.setString(2, process_data);
        statement3.setString(3,cutting_type);
        statement3.setString(4, machine_type);
        statement3.setNull(5,java.sql.Types.INTEGER);
        statement3.setInt(6, department_number3);

        final int rows_inserted3 = statement3.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.",

rows_inserted3));

    }

}

} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

break;
case "4": // Enter a new assembly and associate it with one or more processes
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for assembly information
    System.out.println("Please enter new assembly id");
    final int assembly_id = sc.nextInt();

    System.out.println("Please enter date ordered of assembly in yyyy-mm-dd
format:");
    sc.nextLine();

    final String date_ordered= sc.nextLine();

    System.out.println("Please enter assembly_details: ");

```

```

final String assembly_details = sc.nextLine();

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

System.out.println("How many processes is this assembly associated with: ");
final int num_of_processes = sc.nextInt();

int process_id_array[] = new int[num_of_processes];

System.out.println("List all the process ID's associated. Please click enter after
each ID: ");

for(int i = 0; i < num_of_processes; ++i)
{
    process_id_array[i] = sc.nextInt();
}

try (final Connection connection = DriverManager.getConnection(URL)) {
    // Add assembly first
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_4)) {

        statement.setInt(1, assembly_id);
        statement.setDate(2, java.sql.Date.valueOf(date_ordered));
        statement.setString(3, assembly_details);
        statement.setString(4, customer_name);

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Inserted
Assembly.", rows_inserted));

        for(int i = 0; i < num_of_processes; ++i)
        {
            // Now associate the processes with the new assembly id
            final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_4B);
            statement2.setInt(1, assembly_id);
            statement2.setInt(2, process_id_array[i]);
            final int rows_inserted2 = statement2.executeUpdate();
            System.out.println(String.format("Done. %d rows inserted. Process
associated.", rows_inserted2));

            // Associate every Fit Process
        }
    }
}

```

```

        final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_4BFIT);
        statement3.setInt(1, assembly_id);
        statement3.setInt(2, process_id_array[i]);
        final int rows_inserted3 = statement3.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Fit Process
associated.", rows_inserted3));

        // Associate every Cut Process
        final PreparedStatement statement4 =
connection.prepareStatement(QUERY_TEMPLATE_4BCUT);
        statement4.setInt(1, assembly_id);
        statement4.setInt(2, process_id_array[i]);
        final int rows_inserted4 = statement4.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Cut Process
associated.", rows_inserted4));

        // Associated every Paint Process
        final PreparedStatement statement5 =
connection.prepareStatement(QUERY_TEMPLATE_4BPAINT);
        statement5.setInt(1, assembly_id);
        statement5.setInt(2, process_id_array[i]);
        final int rows_inserted5 = statement5.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Paint Process
associated.", rows_inserted5));

    }

}

}

} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

break;

case "5": //Enter a new account and associate it with the process, assembly, or
department
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for Account information
    System.out.println("Please enter new account number:");
    final int account_number = sc.nextInt();

    System.out.println("Please enter date account established in yyyy-mm-dd
format:");
    sc.nextLine();
}

```

```

final String date_account_established = sc.nextLine();

System.out.println("Please enter associated process_id for the account:");
final int account_process_id = sc.nextInt();

System.out.println("Please enter associated assembly_id for the account:");
final int account_assembly_id = sc.nextInt();

System.out.println("Please enter associated department_number for the
account:");
final int account_department_number = sc.nextInt();

try (final Connection connection = DriverManager.getConnection(URL)) {
    // Add department first
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_5A)) {

        statement.setInt(1, account_number);
        statement.setDate(2, java.sql.Date.valueOf(date_account_established));

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Account
created.", rows_inserted));

        // Insert Assembly Account
        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_5B);
        statement2.setInt(1, account_number);
        statement2.setDate(2, java.sql.Date.valueOf(date_account_established));
        statement2.setInt(3, 0);
        statement2.setInt(4, account_assembly_id);
        final int rows_inserted2 = statement2.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Assembly
Account created.", rows_inserted2));

        // Insert Process Account
        final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_5C);
        statement3.setInt(1, account_number);
        statement3.setDate(2, java.sql.Date.valueOf(date_account_established));
        statement3.setInt(3, 0);
        statement3.setInt(4, account_process_id);
        final int rows_inserted3 = statement3.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Process Account
created.", rows_inserted3));
    }
}

```

```

        // Insert Department Account
        final PreparedStatement statement4 =
connection.prepareStatement(QUERY_TEMPLATE_5D);
        statement4.setInt(1, account_number);
        statement4.setDate(2, java.sql.Date.valueOf(date_account_established));
        statement4.setInt(3, 0);
        statement4.setInt(4, account_department_number);
        final int rows_inserted4 = statement4.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Department
Account created.", rows_inserted4));
    }
}
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

break;

case "6":
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for Job information
    System.out.println("Please enter new job number:");
    final int job_number = sc.nextInt();

    System.out.println("Please enter date commenced for the job in yyyy-mm-dd
format:");
    sc.nextLine();

    final String date_commenced = sc.nextLine();

    System.out.println("Please enter the associated assembly id for the job: ");
    final int job_assembly_id = sc.nextInt();

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

    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Add department first
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_6)) {

            statement.setInt(1, job_number);
            statement.setDate(2, java.sql.Date.valueOf(date_commenced));
            statement.setNull(3, java.sql.Types.DATE);

```

```

        statement.setInt(4, job_process_id);
        statement.setInt(5, job_assembly_id);

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", 
rows_inserted));
    }
}

} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;

case "7": // Enter date completed and information related to type of job

try { // In case of an error, this returns it to the main menu instead of terminating
program
    System.out.println("Please enter the job number completed :");
    final int job_number_completed = sc.nextInt();

    System.out.println("Please enter date completed for job in yyyy-mm-dd format:");
    sc.nextLine();
    final String date_completed = sc.nextLine();

    System.out.println("Please enter type of job (Fit, Paint, Cut)? ");
    final String job_type = sc.nextLine();

    // Have to initialize all variables
    String color, type_machine_used, material_used, labor_time,
amount_of_time_machine_used;
    color = "";
    type_machine_used = "";
    material_used = "";
    labor_time = "";
    amount_of_time_machine_used = "";

    int volume = 0;

    if(job_type.equals("Fit")) {

        System.out.println("Please enter labor time in hh:mm:ss format");
        labor_time = sc.nextLine();

    } else if (job_type.equals("Paint")) {

```

```

System.out.println("Please enter labor time in hh:mm:ss format");
labor_time = sc.nextLine();

System.out.println("Please enter color for Paint job: ");
color = sc.nextLine();

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

} else if (job_type.equals("Cut")) {

    System.out.println("Please enter labor time in hh:mm:ss format");
    labor_time = sc.nextLine();

    System.out.println("Please enter the type of machine used: ");
    type_machine_used = sc.nextLine();

    System.out.println("Please enter amount of time machine used in hh:mm:ss
format");
    amount_of_time_machine_used = sc.nextLine();

    System.out.println("Please enter material used");
    material_used = sc.nextLine();
}

try (final Connection connection = DriverManager.getConnection(URL)) {
    // Add department first
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_7)) {

        statement.setDate(1, java.sql.Date.valueOf(date_completed));
        statement.setInt(2, job_number_completed);

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.",

rows_inserted));

        if(job_type.equals("Fit")) {

            final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_7FIT);
            statement2.setInt(1, job_number_completed);
            statement2.setTime(2, java.sql.Time.valueOf(labor_time));

            final int rows_inserted2= statement2.executeUpdate();
        }
    }
}

```

```

        System.out.println(String.format("Done. %d rows inserted. Fit Job
Added.", rows_inserted2));

    } else if (job_type.equals("Paint")) {

        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_7PAINT);
        statement2.setInt(1, job_number_completed);
        statement2.setTime(2, java.sql.Time.valueOf(labor_time));
        statement2.setString(3, color);
        statement2.setInt(4, volume);

        final int rows_inserted2= statement2.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Paint Job
Added.", rows_inserted2));

    } else if (job_type.equals("Cut")) {

        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_7CUT);
        statement2.setInt(1, job_number_completed);
        statement2.setTime(2, java.sql.Time.valueOf(labor_time));
        statement2.setString(3, type_machine_used);
        statement2.setTime(4,
java.sql.Time.valueOf(amount_of_time_machine_used));
        statement2.setString(5, material_used);

        final int rows_inserted2= statement2.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Cut Job
Added.", rows_inserted2));

    }

}

} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;
case "8": // Enter transaction and update costs of affected accounts
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for transaction info
    System.out.println("Please enter new transaction number");
    final int transaction_number = sc.nextInt();

```

```

System.out.println("Please enter cost:");
final int transaction_cost = sc.nextInt();

System.out.println("Enter the associated account number: ");
final int transaction_account_number = sc.nextInt();

System.out.println("Please enter the associated job number: ");
final int transaction_job_number = sc.nextInt();

try (final Connection connection = DriverManager.getConnection(URL)) {
    // Add department first
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_8)) {

        statement.setInt(1, transaction_number);
        statement.setInt(2, transaction_cost);
        statement.setInt(3, transaction_job_number);
        statement.setInt(4, transaction_account_number);

        final int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));

        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_8ASSEMBLY);
        statement2.setInt(1, transaction_cost);
        statement2.setInt(2, transaction_account_number);

        final int rows_inserted2 = statement2.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Assembly Account updated", rows_inserted2));

        final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_8PROCESS);
        statement3.setInt(1, transaction_cost);
        statement3.setInt(2, transaction_account_number);

        final int rows_inserted3 = statement3.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Process Account updated", rows_inserted3));

        final PreparedStatement statement4 =
connection.prepareStatement(QUERY_TEMPLATE_8DEPARTMENT);
        statement4.setInt(1, transaction_cost);
        statement4.setInt(2, transaction_account_number);
    }
}

```

```

        final int rows_inserted4 = statement4.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted. Department
Account updated", rows_inserted4));
    }
}
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}

break;

case "9":
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for assembly id
    System.out.println("Please enter assembly ID you want to find cost of:");
    final int cost_assembly_id = sc.nextInt();

    try (final Connection connection = DriverManager.getConnection(URL)) {
        // Add department first
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_9)) {

            statement.setInt(1, cost_assembly_id);
            final ResultSet resultSet = statement.executeQuery();

            resultSet.next();
            System.out.println(String.format("The cost associated with the assembly ID
is %s", resultSet.getInt(1)));
        }
    }
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;

case "10": // Retrieve total labor time
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for department and completion date
    System.out.println("Please enter the department number:");
    final int time_department_number = sc.nextInt();

    System.out.println("Please enter the completion date in yyyy-mm-dd format:");
    sc.nextLine();
}

```

```

final String time_completion_date = sc.nextLine();

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

int fit_time = 0;
int paint_time = 0;
int cut_time = 0;
int total_time_in_minutes = 0;

// Get a database connection and prepare a query statement
try (final Connection connection = DriverManager.getConnection(URL)) {
    try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_10FIT)){ // get total fit labor time

        statement.setInt(1, time_department_number);
        statement.setDate(2, java.sql.Date.valueOf(time_completion_date));

        final ResultSet resultSet = statement.executeQuery();

        resultSet.next();
        fit_time += resultSet.getInt(1);

        // get total paint labor time
        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_10PAINT);
        statement2.setInt(1, time_department_number);
        statement2.setDate(2, java.sql.Date.valueOf(time_completion_date));

        final ResultSet resultSet2 = statement2.executeQuery();

        resultSet2.next();
        paint_time += resultSet2.getInt(1);

        // get total cut labor time
        final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_10CUT);
        statement3.setInt(1, time_department_number);
        statement3.setDate(2, java.sql.Date.valueOf(time_completion_date));

        final ResultSet resultSet3 = statement3.executeQuery();

        resultSet3.next();
        cut_time += resultSet3.getInt(1);

        total_time_in_minutes = fit_time + paint_time + cut_time;
    }
}

```

```

        System.out.println(String.format("The total labor time for jobs completed in
the department during a given date is %s"
                                         , total_time_in_minutes));
    }
}
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;

case "11":
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Read in the user input for assembly id
    System.out.println("Please enter the assembly id:");
    final int process_assembly_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)) {
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_11)){ // get processes

            statement.setInt(1, process_assembly_id);
            final ResultSet resultSet = statement.executeQuery();

            System.out.println("Processes through which a given assembly ID has
passed so far in date commenced order: ");
            System.out.println(" Process ID | Department Number | Date Commenced ");

            while (resultSet.next()) {
                System.out.println(String.format("%s | %s | %s",
resultSet.getInt(1),
resultSet.getInt(2),
resultSet.getDate(3)));
            }
        }
    } catch (Exception e) {
        System.out.println("You got an error! Returning to the main menu.");
    }
break;

case "12": // Retrieve jobs completed during given date and given department
try { // In case of an error, this returns it to the main menu instead of terminating
program

```

```

// Read in the user input for job information
System.out.println("Please enter the completion date in yyyy-mm-dd format: ");
sc.nextLine();
final String job_completion_date = sc.nextLine();

System.out.println("Please enter the department number: ");
final int job_department_number = 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 PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_12A)){ // search fit jobs

        statement.setDate(1, java.sql.Date.valueOf(job_completion_date));
        statement.setInt(2, job_department_number);
        final ResultSet resultSet = statement.executeQuery();

        // Print out Fit Jobs
        System.out.println("Fit Jobs completed during a given date in a given
department: ");
        System.out.println("Job Number | Assembly ID | Labor Time
(hh:mm:ss)");
        while (resultSet.next()) {
            System.out.println(String.format("%s | %s | %s",
resultSet.getInt(1),
resultSet.getInt(2),
resultSet.getTime(3)));
        }
        System.out.println("");

        // Print out Paint Jobs
        final PreparedStatement statement2 =
connection.prepareStatement(QUERY_TEMPLATE_12B);
        statement2.setDate(1, java.sql.Date.valueOf(job_completion_date));
        statement2.setInt(2, job_department_number);
        final ResultSet resultSet2 = statement2.executeQuery();

        System.out.println("Paint Jobs completed during a given date in a given
department: ");
        System.out.println("Job Number | Assembly ID | color | volume | Labor
Time (hh:mm:ss)");
        while (resultSet2.next()) {
            System.out.println(String.format("%s | %s | %s | %s | %s",
resultSet2.getInt(1),
resultSet2.getInt(2),

```

```

        resultSet2.getString(3),
        resultSet2.getInt(4),
        resultSet2.getTime(5)));
    }
    System.out.println("");

    // Print out Cut Jobs
    final PreparedStatement statement3 =
connection.prepareStatement(QUERY_TEMPLATE_12C);
    statement3.setDate(1, java.sql.Date.valueOf(job_completion_date));
    statement3.setInt(2, job_department_number);
    final ResultSet resultSet3 = statement3.executeQuery();

    System.out.println("Cut Jobs completed during a given date in a given
department: ");
    System.out.println("Job Number | Assembly ID | Type Machine Used |
Amount of Time Machine Used (hh:mm:ss) | Material Used | Labor Time (hh:mm:ss)");
    while (resultSet3.next()) {
        System.out.println(String.format("%s | %s | %s | %s | %s | %s",
            resultSet3.getInt(1),
            resultSet3.getInt(2),
            resultSet3.getString(3),
            resultSet3.getTime(4),
            resultSet3.getString(5),
            resultSet3.getTime(6)));
    }

}
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;
case "13": // Retrieve customers in name order whose category is in a given range
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Collect data to input for the Customer retrieval
    System.out.println("Please enter desired lower bound of category range: ");
    final int lower_bound = sc.nextInt();

    System.out.println("Please enter desired upper bound of category range:");
    final int upper_bound = 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 PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_13)){
            // Make a new customer and populate the database with it.
            statement.setInt(1,lower_bound);
            statement.setInt(2, upper_bound);

            final ResultSet resultSet = statement.executeQuery();

            // Print out Customers
            System.out.println("Customers (in name order)");
            while (resultSet.next()) {
                System.out.println(String.format("%s",
                    resultSet.getString(1)));
            }
        }
    } catch (Exception e) {
        System.out.println("You got an error! Returning to the main menu.");
    }

    break;
case "14": // Delete all cut jobs in a given range
try { // In case of an error, this returns it to the main menu instead of terminating
program
        // Collect data to input for the job deletion
        System.out.println("Please enter desired lower bound of job number range: ");
        final int job_lower_bound = sc.nextInt();

        System.out.println("Please enter desired upper bound of job number range:");
        final int job_upper_bound = 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 PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_14)){
                // Make a new customer and populate the database with it.
                statement.setInt(1,job_lower_bound);
                statement.setInt(2, job_upper_bound);

                statement.executeUpdate();

                // Print out Customers
                System.out.println(String.format("Job numbers from %s to %s are deleted.", 
                    job_lower_bound, job_upper_bound));
            }
        }
    }
}

```

```

        }
    } catch (Exception e) {
        System.out.println("You got an error! Returning to the main menu.");
    }
    break;
case "15": // Change color of a given paint job
try { // In case of an error, this returns it to the main menu instead of terminating
program
    // Collect data to input for the color change
    System.out.println("What is the paint job number you wish to update: ");
    final int color_job_number = sc.nextInt();

    System.out.println("Please enter the color you wish to update to:");
    sc.nextLine();
    final String new_color = 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_15)){
            // Make a new customer and populate the database with it.
            statement.setString(1, new_color);
            statement.setInt(2, color_job_number);

            int rows_updated = statement.executeUpdate();
            System.out.println(String.format("Done. %d rows updated.",

rows_updated));
        }
    }
} catch (Exception e) {
    System.out.println("You got an error! Returning to the main menu.");
}
break;

case "16":
// try and catch are used to not terminate loop in case of error.
try {
String line;

// Retrieve input file name from user
System.out.println("Enter the file-name to import data.");
sc.nextLine();
String file_name = sc.nextLine();

File file = new File(file_name);

```

```

// Creating new Scanner Object to read in file
Scanner filescanner = new Scanner(file);

// Read in input file
while(filescanner.hasNextLine()) {
    line = filescanner.nextLine();

    // Dividing line to parts separated by Delimiter ","
    String[] parts = line.split(",");

    String file_customer_name = parts[0];
    String file_address = parts[1];
    int file_category= Integer.parseInt(parts[2]);

    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)){
            // Make a new customer and populate the database with it.
            statement.setString(1, file_customer_name);
            statement.setString(2, file_address);
            statement.setInt(3, file_category);

            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done. %d rows inserted.",
rows_inserted));
        }
    }
}

} catch (Exception e) {
    System.out.print("You got an error!. Returning to main menu");
}
break;
case "17":
    // try and catch are used to not terminate loop in case of error.
    try {

        // Enter the filename to output result
        System.out.println("Enter the file-name to Export data.");
        sc.nextLine();
        final String file_name1 = sc.nextLine();
    }

```

```

// Taking lower bound of category from user
System.out.println("Enter the lower bound of category.");
final int lower_bound2 = sc.nextInt();

// Taking upper bound of category from user
System.out.println("Enter the upper bound of category.");
final int upper_bound2 = sc.nextInt();

// Creating new file writer Object
FileWriter fw = new FileWriter(file_name1);

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_13)){
        // Make a new customer and populate the database with it.
        statement.setInt(1,lower_bound2);
        statement.setInt(2, upper_bound2);
        final ResultSet resultSet = statement.executeQuery();

        // Output customer name to data file
        while (resultSet.next()) {
            fw.write("Customer name is " + resultSet.getString(1) + "\n");
        }
        fw.close();
    } catch (SQLException e) {
        e.getCause().getMessage();
    }
}

} catch (Exception e) {
    System.out.println("You got an error!. Returning to main menu");
}
break;
case "18": // Do nothing, the while loop will terminate upon the next iteration
    System.out.println("Quitting.");
    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
}
}

```

## **Screen shot showing successful compilation of Java source program**

```

1033 }
1034
Console X
DatabaseIP [Java Application] /Library/Java/JavaVirtualMachines/jdk-11.0.11.jdk/Contents/Home/bin/java (Nov 21, 2021, 9:05:29 PM)
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
(18) Quit

```

## **5 queries of Query#1**

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

1  
Please enter name of the new customer:  
Bob Hope  
Please enter address of the new customer:  
1098 Washington Avenue, Oklahoma City, Oklahoma, 70941  
Please enter Customer category (a number from 1-10):  
3  
Connecting to the database...  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer

- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

1

Please enter name of the new customer:

Mike Tyson

Please enter address of the new customer:

9998 Oklahoma St., Morgantown, West Virginia, 12424

Please enter Customer category (a number from 1-10):

4

Connecting to the database...

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

1

Please enter name of the new customer:

Buzz Lightyear

Please enter address of the new customer:

10922 Main St., Garden Grove, California, 90210

Please enter Customer category (a number from 1-10):

7

Connecting to the database...

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process

- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

1

Please enter name of the new customer:

Pope Stevens

Please enter address of the new customer:

90412 Norman Avenue, Norman, Oklahoma, 70891

Please enter Customer category (a number from 1-10):

10

Connecting to the database...

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

1

Please enter name of the new customer:

Koi Fish

Please enter address of the new customer:

8958 Geico Trail, Nowhere, Oklahoma, 90423

Please enter Customer category (a number from 1-10):

6

Connecting to the database...

Done. 1 rows inserted.

Content of Customer Table after 5 queries of Query#1

	Results	Messages	
	name	address	category
1	Bob Hope	1098 Washington Avenue, 0...	3
2	Buzz Lightyear	10922 Main St., Garden Gr...	7
3	Koi Fish	8958 Geico Trail, Nowhere...	6
4	Mike Tyson	9998 Oklahoma St., Morgan...	4
5	Pope Stevens	90412 Norman Avenue, Norm...	10

- ⏪

S

## 5 queries of Query#2

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

2

Please enter new department number:

1

Please enter department data:

This department is a pretty good department

Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)

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

2

Please enter new department number:

2

Please enter department data:

The underwater department

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

2

Please enter new department number:

3

Please enter department data:

The department in heaven

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

2

Please enter new department number:

4

Please enter department data:

The department in wisconsin

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department

- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

2

Please enter new department number:

5

Please enter department data:

The department in new zealand

Done. 1 rows inserted.

### Content of Department Table after 5 queries of Query#2

	Results	Messages
	department_number	department_data
1	1	This department is a pret...
2	2	The underwater department
3	3	The department in heaven
4	4	The department in wiscons...
5	5	The department in new zea...

### 10 Queries of Query#3

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

3

Please enter a new process ID:

420

Please enter the corresponding process\_data:  
This process is meant to be done quickly  
Please enter new department number associated with process:  
1  
Please enter type of process (Fit, Paint, Cut)?  
Fit  
Please enter the corresponding fit\_type:  
The fit must be tight  
Connecting to the database...  
Done. 1 rows inserted.  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

421  
Unrecognized option: 421  
Please try again!

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

3  
Please enter a new process ID:  
421  
Please enter the corresponding process\_data:  
This process is going to manufacture desks  
Please enter new department number associated with process:  
1  
Please enter type of process (Fit, Paint, Cut)?  
Paint  
Please enter the corresponding paint\_type:  
Glossy  
Please enter the corresponding paint\_method:  
By Hand  
Connecting to the database...

Done. 1 rows inserted.

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

3

Please enter a new process ID:

422

Please enter the corresponding process\_data:

This process is going to make desks

Please enter new department number associated with process:

1

Please enter type of process (Fit, Paint, Cut)?

Cut

Please enter the corresponding cutting type:

Square

Please enter the corresponding machine type:

Csar Saw

Connecting to the database...

Done. 1 rows inserted.

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

3

Please enter a new process ID:

423

Please enter the corresponding process\_data:

This is going to make notebooks

Please enter new department number associated with process:

2

Please enter type of process (Fit, Paint, Cut)?

Fit

Please enter the corresponding fit\_type:

Not too tight  
Connecting to the database...  
Done. 1 rows inserted.  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

3  
Please enter a new process ID:  
424  
Please enter the corresponding process\_data:  
THis is going to make the color of the notebook cover  
Please enter new department number associated with process:  
2  
Please enter type of process (Fit, Paint, Cut)?  
Paint  
Please enter the corresponding paint\_type:  
Matte  
Please enter the corresponding paint\_method:  
Laser Painted  
Connecting to the database...  
Done. 1 rows inserted.  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

3  
Please enter a new process ID:  
425  
Please enter the corresponding process\_data:  
This will cut the notebook into rectangles  
Please enter new department number associated with process:  
2  
Please enter type of process (Fit, Paint, Cut)?

Cut  
Please enter the corresponding cutting type:  
Rectangular  
Please enter the corresponding machine type:  
A Laser Cutter  
Connecting to the database...  
Done. 1 rows inserted.  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

3  
Please enter a new process ID:  
426  
Please enter the corresponding process\_data:  
This will make monitors  
Please enter new department number associated with process:  
3  
Please enter type of process (Fit, Paint, Cut)?  
Fit  
Please enter the corresponding fit\_type:  
Has to fit on my desk  
Connecting to the database...  
Done. 1 rows inserted.  
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

3  
Please enter a new process ID:  
427  
Please enter the corresponding process\_data:  
This will paint the back of the monitor black  
Please enter new department number associated with process:

3  
Please enter type of process (Fit, Paint, Cut)?  
Paint

Please enter the corresponding paint\_type:

Shiny Paint

Please enter the corresponding paint\_method:

With your fingers

Connecting to the database...

Done. 1 rows inserted.

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

3  
Please enter a new process ID:  
428

Please enter the corresponding process\_data:

This will make macbooks

Please enter new department number associated with process:

4

Please enter type of process (Fit, Paint, Cut)?  
Cut

Please enter the corresponding cutting type:

Cut Meta

Please enter the corresponding machine type:

Metal Cutter

Connecting to the database...

Done. 1 rows inserted.

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

3  
Please enter a new process ID:

429  
 Please enter the corresponding process\_data:  
 This will make apple headphones  
 Please enter new department number associated with process:  
 5  
 Please enter type of process (Fit, Paint, Cut)?  
 Paint  
 Please enter the corresponding paint\_type:  
 Paint White  
 Please enter the corresponding paint\_method:  
 With a brush  
 Connecting to the database...  
 Done. 1 rows inserted.  
 Done. 1 rows inserted.

Content of Process Table after 10 queries of Query#3

Results		Messages		
	process_id	process_data	assembly_id	department_number
1	420	This process is meant to ...	NULL	1
2	421	This process is going to ...	NULL	1
3	422	This process is going to ...	NULL	1
4	423	This is going to make not...	NULL	2
5	424	THis is going to make the...	NULL	2
6	425	This will cut the notebook...	NULL	2
7	426	This will make monitors	NULL	3
...	427	This will paint the back ...	NULL	3
9	428	This will make macbooks	NULL	4
10	429	This will make apple head...	NULL	5

Content of Fit Process Table after 10 queries of Query#3

Results		Messages			
	process_id	process_data	fit_type	assembly_id	department_number
1	420	This process is meant to ...	The fit must be tight	NULL	1
2	423	This is going to make not...	Not too tight	NULL	2
3	426	This will make monitors	Has to fit on my desk	NULL	3

Content of Paint Process Table after 10 queries of Query#3

Results Messages						
	process_id	process_data	paint_type	painting_method	assembly_id	department_number
1	421	This process is going to ...	Glossy	By Hand	NULL	1
2	424	THis is going to make the...	Matte	Laser Painted	NULL	2
3	427	This will paint the back ...	Shiny Paint	With your fingers	NULL	3
4	429	This will make apple head...	Paint White	With a brush	NULL	5

Content of Cut Process Table after 10 queries of Query#3

Results Messages						
	process_id	process_data	cutting_type	machine_type	assembly_id	department_number
1	422	This process is going to ...	Square	Csar Saw	NULL	1
2	425	This will cut the noteboo...	Rectangular	A Laser Cutter	NULL	2
3	428	This will make macbooks	Cut Meta	Metal Cutter	NULL	4

10 Queries of Query#4

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

69

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-01

Please enter assembly\_details:

This assembles manufactures notebooks

Please enter the customer name who ordered the assembly:

Bob Hope

How many processes is this assembly associated with:

1

List all the process ID's associated. Please click enter after each ID:

420

Done. 1 rows inserted. Inserted Assembly.

Done. 1 rows inserted. Process associated.

Done. 1 rows inserted. Fit Process associated.

Done. 0 rows inserted. Cut Process associated.

Done. 0 rows inserted. Paint Process associated.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together

- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

70

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-01

Please enter assembly\_details:

This assembles manufactures the best

Please enter the customer name who ordered the assembly:

Bob Hope

How many processes is this assembly associated with:

1

List all the process ID's associated. Please click enter after each ID:

421

Done. 1 rows inserted. Inserted Assembly.

Done. 1 rows inserted. Process associated.

Done. 0 rows inserted. Fit Process associated.

Done. 0 rows inserted. Cut Process associated.

Done. 1 rows inserted. Paint Process associated.

## WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

72

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-02

Please enter assembly\_details:

This assembles Teslas

Please enter the customer name who ordered the assembly:

Bob Hope

How many processes is this assembly associated with:

1

List all the process ID's associated. Please click enter after each ID:

422

Done. 1 rows inserted. Inserted Assembly.

Done. 1 rows inserted. Process associated.  
Done. 0 rows inserted. Fit Process associated.  
Done. 1 rows inserted. Cut Process associated.  
Done. 0 rows inserted. Paint Process associated.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

73

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-03

Please enter assembly\_details:

This assemblies notebookos

Please enter the customer name who ordered the assembly:

Buzz Lightyear

How many processes is this assembly associated with:

1

List all the process ID's associated. Please click enter after each ID:

423

Done. 1 rows inserted. Inserted Assembly.

Done. 1 rows inserted. Process associated.

Done. 1 rows inserted. Fit Process associated.

Done. 0 rows inserted. Cut Process associated.

Done. 0 rows inserted. Paint Process associated.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

74

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-05

Please enter assembly\_details:  
This assembles laptops  
Please enter the customer name who ordered the assembly:  
Buzz Lightyear  
How many processes is this assembly associated with:  
0  
List all the process ID's associated. Please click enter after each ID:  
Done. 1 rows inserted. Inserted Assembly.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

4  
Please enter new assembly id  
75  
Please enter date ordered of assembly in yyyy-mm-dd format:  
2020-01-01  
Please enter assembly\_details:  
This assembles the Mach7 Landing Pad  
Please enter the customer name who ordered the assembly:  
Koi Fish  
How many processes is this assembly associated with:  
1  
List all the process ID's associated. Please click enter after each ID:  
424  
Done. 1 rows inserted. Inserted Assembly.  
Done. 1 rows inserted. Process associated.  
Done. 0 rows inserted. Fit Process associated.  
Done. 0 rows inserted. Cut Process associated.  
Done. 1 rows inserted. Paint Process associated.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

Please enter new assembly id  
76  
Please enter date ordered of assembly in yyyy-mm-dd format:  
2020-01-09  
Please enter assembly\_details:  
This assembles fountain pains  
Please enter the customer name who ordered the assembly:  
Koi Fish  
How many processes is this assembly associated with:  
1  
List all the process ID's associated. Please click enter after each ID:  
425  
Done. 1 rows inserted. Inserted Assembly.  
Done. 1 rows inserted. Process associated.  
Done. 0 rows inserted. Fit Process associated.  
Done. 1 rows inserted. Cut Process associated.  
Done. 0 rows inserted. Paint Process associated.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department  
(13) Retrieve the customers(in name order) whose category is in a specific range  
(14) Delete all cut-jobs whose job-no is in a specific range  
(15) Change the color of a specific paint job  
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)  
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen  
(18) Quit

4  
Please enter new assembly id  
77  
Please enter date ordered of assembly in yyyy-mm-dd format:  
2020-01-10  
Please enter assembly\_details:  
This assembles mice for computers  
Please enter the customer name who ordered the assembly:  
Mike Tyson  
How many processes is this assembly associated with:  
1  
List all the process ID's associated. Please click enter after each ID:  
426  
Done. 1 rows inserted. Inserted Assembly.  
Done. 1 rows inserted. Process associated.  
Done. 1 rows inserted. Fit Process associated.  
Done. 0 rows inserted. Cut Process associated.  
Done. 0 rows inserted. Paint Process associated.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM  
(1) Enter a new customer  
(2) Enter a new department  
(3) Enter a new process and department together  
(4) Enter a new assembly and associate it with one or more processes  
(5) Create a new account and associate it with a process, assembly, or department  
(6) Enter a new job and date the job began  
(7) At the completion of a job, enter the date completed and relevant info  
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost  
(9) Retrieve the total cost incurred on a specific assembly  
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date  
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process  
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department

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

4

Please enter new assembly id

78

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-03

Please enter assembly\_details:

This assembles calculators

Please enter the customer name who ordered the assembly:

Mike Tyson

How many processes is this assembly associated with:

0

List all the process ID's associated. Please click enter after each ID:

Done. 1 rows inserted. Inserted Assembly.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

4

Please enter new assembly id

79

Please enter date ordered of assembly in yyyy-mm-dd format:

2020-01-24

Please enter assembly\_details:

This assembles java programs

Please enter the customer name who ordered the assembly:

Pope Stevens

How many processes is this assembly associated with:

3

List all the process ID's associated. Please click enter after each ID:

427

428

429

Done. 1 rows inserted. Inserted Assembly.

Done. 1 rows inserted. Process associated.

Done. 0 rows inserted. Fit Process associated.

Done. 0 rows inserted. Cut Process associated.

Done. 1 rows inserted. Paint Process associated.

Done. 1 rows inserted. Process associated.

Done. 0 rows inserted. Fit Process associated.

Done. 1 rows inserted. Cut Process associated.

Done. 0 rows inserted. Paint Process associated.

Done. 1 rows inserted. Process associated.

Done. 0 rows inserted. Fit Process associated.

Done. 0 rows inserted. Cut Process associated.

Done. 1 rows inserted. Paint Process associated.

Content of Assembly Table after 10 queries of Query#4

	assembly_id	date_ordered	assembly_details	name
1	69	2020-01-01	This assembly manufacture...	Bob Hope
2	70	2020-01-01	THis assembly manufacture...	Bob Hope
3	72	2020-01-02	This assembles Teslas	Bob Hope
4	73	2020-01-03	This assembles notebooks	Buzz Lightyear
5	74	2020-01-05	This assembles laptops	Buzz Lightyear
6	75	2020-01-01	This assembles the Mach7 ...	Koi Fish
7	76	2020-01-09	This assembles fountain p...	Koi Fish
8	77	2020-01-10	This assembles mice for c...	Mike Tyson
9	78	2020-01-03	This assembles calculators	Mike Tyson
10	79	2020-01-24	This assembles java progra...	Pope Stevens

Content of Process Table after 10 queries of Query#4

	process_id	process_data	assembly_id	department_number
1	420	This process is meant to ...	69	1
2	421	This process is going to ...	70	1
3	422	This process is going to ...	72	1
4	423	This is going to make not...	73	2
5	424	THis is going to make the...	75	2
6	425	This will cut the notebook...	76	2
7	426	This will make monitors	77	3
8	427	This will paint the back ...	79	3
9	428	This will make macbooks	79	4
10	429	This will make apple head...	79	5

Content of Fit Process Table after 10 queries of Query#4

Results Messages					
	process_id	process_data	fit_type	assembly_id	department_number
1	420	This process is meant to ...	The fit must be tight	69	1
2	423	This is going to make not...	Not too tight	73	2
3	426	This will make monitors	Has to fit on my desk	77	3

Content of Paint Process Table after 10 queries of Query#4

Results Messages						
	process_id	process_data	paint_type	painting_method	assembly_id	department_num...
1	421	This process is going to ...	Glossy	By Hand	70	1
2	424	THis is going to make the...	Matte	Laser Painted	75	2
3	427	This will paint the back ...	Shiny Pa...	With your fing...	79	3
4	429	This will make apple head...	Paint Wh...	With a brush	79	5

Content of Cut Process Table after 10 queries of Query#4

Results Messages						
	process_id	process_data	cutting_type	machine_type	assembly_id	department_number
1	422	This process is going to ...	Square	Csar Saw	72	1
2	425	This will cut the notebook...	Rectangular	A Laser Cutter	76	2
3	428	This will make macbooks	Cut Meta	Metal Cutter	79	4

## 10 Queries of Query#5

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

Please enter new account number:  
2000  
Please enter date account established in yyyy-mm-dd format:  
2020-01-02  
Please enter associated process\_id for the account:  
420  
Please enter associated assembly\_id for the account:  
69  
Please enter associated department\_number for the account:  
1  
Done. 1 rows inserted. Account created.  
Done. 1 rows inserted. Assembly Account created.  
Done. 1 rows inserted. Process Account created.  
Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5  
Please enter new account number:  
20001  
Please enter date account established in yyyy-mm-dd format:  
2020-01-04  
Please enter associated process\_id for the account:  
421  
Please enter associated assembly\_id for the account:  
70  
Please enter associated department\_number for the account:  
1  
Done. 1 rows inserted. Account created.  
Done. 1 rows inserted. Assembly Account created.  
Done. 1 rows inserted. Process Account created.  
Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

20002

Please enter date account established in yyyy-mm-dd format:

2020-04-01

Please enter associated process\_id for the account:

422

Please enter associated assembly\_id for the account:

72

Please enter associated department\_number for the account:

1

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

2003

Please enter date account established in yyyy-mm-dd format:

2020-01-03

Please enter associated process\_id for the account:

423

Please enter associated assembly\_id for the account:

73

Please enter associated department\_number for the account:

2

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)

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

5  
Please enter new account number:  
2004  
Please enter date account established in yyyy-mm-dd format:  
2020-01-02  
Please enter associated process\_id for the account:  
424  
Please enter associated assembly\_id for the account:  
75  
Please enter associated department\_number for the account:  
2  
Done. 1 rows inserted. Account created.  
Done. 1 rows inserted. Assembly Account created.  
Done. 1 rows inserted. Process Account created.  
Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5  
Please enter new account number:  
2005  
Please enter date account established in yyyy-mm-dd format:  
2020-01-04  
Please enter associated process\_id for the account:  
425  
Please enter associated assembly\_id for the account:  
76  
Please enter associated department\_number for the account:  
2  
Done. 1 rows inserted. Account created.  
Done. 1 rows inserted. Assembly Account created.  
Done. 1 rows inserted. Process Account created.  
Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range

- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

2006

Please enter date account established in yyyy-mm-dd format:

2020-01-01

Please enter associated process\_id for the account:

426

Please enter associated assembly\_id for the account:

77

Please enter associated department\_number for the account:

3

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

Please try again!

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

2007

Please enter date account established in yyyy-mm-dd format:

2020-01-01

Please enter associated process\_id for the account:

427

Please enter associated assembly\_id for the account:

79

Please enter associated department\_number for the account:

3

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process

- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

2008

Please enter date account established in yyyy-mm-dd format:

2020-01-02

Please enter associated process\_id for the account:

428

Please enter associated assembly\_id for the account:

79

Please enter associated department\_number for the account:

4

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

5

Please enter new account number:

2009

Please enter date account established in yyyy-mm-dd format:

2020-01-02

Please enter associated process\_id for the account:

429

Please enter associated assembly\_id for the account:

79

Please enter associated department\_number for the account:

5

Done. 1 rows inserted. Account created.

Done. 1 rows inserted. Assembly Account created.

Done. 1 rows inserted. Process Account created.

Done. 1 rows inserted. Department Account created.

#### Content of Account Table after 10 queries of Query#5

**Results**   **Messages**

---

	account_n...	date_account_established
1	2000	2020-01-02
2	2003	2020-01-03
3	2004	2020-01-02
4	2005	2020-01-04
5	2006	2020-01-01
6	2007	2020-01-01
7	2008	2020-01-02
8	2009	2020-01-02
9	20001	2020-01-04
1...	20002	2020-04-01

Content of Assembly Account Table after 10 queries of Query#5

**Results**   **Messages**

---

	account_number	date_account_established	cost1	assembly_id
1	2000	2020-01-02	0	69
2	2003	2020-01-03	0	73
3	2004	2020-01-02	0	75
4	2005	2020-01-04	0	76
5	2006	2020-01-01	0	77
6	2007	2020-01-01	0	79
7	2008	2020-01-02	0	79
8	2009	2020-01-02	0	79
9	20001	2020-01-04	0	70
10	20002	2020-04-01	0	72

Content of Process Account Table after 10 queries of Query#5

**Results**   **Messages**

	account_number	date_account_established	cost2	process_id
1	2000	2020-01-02	0	420
2	2003	2020-01-03	0	423
3	2004	2020-01-02	0	424
4	2005	2020-01-04	0	425
5	2006	2020-01-01	0	426
6	2007	2020-01-01	0	427
7	2008	2020-01-02	0	428
8	2009	2020-01-02	0	429
9	20001	2020-01-04	0	421
10	20002	2020-04-01	0	422

Content of Department Account Table after 10 queries of Query#5

**Results**   **Messages**

	account_number	date_account_established	cost3	department_number
1	2000	2020-01-02	0	1
2	2003	2020-01-03	0	2
3	2004	2020-01-02	0	2
4	2005	2020-01-04	0	2
5	2006	2020-01-01	0	3
6	2007	2020-01-01	0	3
7	2008	2020-01-02	0	4
8	2009	2020-01-02	0	5
9	20001	2020-01-04	0	1
10	20002	2020-04-01	0	1

## 10 Queries of Query#6

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

22

Please enter date commenced for the job in yyyy-mm-dd format:

2020-04-02

Please enter the associated assembly id for the job:

69

Please enter the associated process id for the job:

420

Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

23

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-05

Please enter the associated assembly id for the job:

70

Please enter the associated process id for the job:

421

Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes

- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

24

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-04

Please enter the associated assembly id for the job:

72

Please enter the associated process id for the job:

422

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

25

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

73

Please enter the associated process id for the job:

423

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department

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

6

Please enter new job number:

26

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

75

Please enter the associated process id for the job:

424

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

27

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

76

Please enter the associated process id for the job:

424

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

28

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

77

Please enter the associated process id for the job:

426

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

29

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

79

Please enter the associated process id for the job:

427

Done. 1 rows inserted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

30

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

79

Please enter the associated process id for the job:

428

Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

6

Please enter new job number:

31

Please enter date commenced for the job in yyyy-mm-dd format:

2020-01-02

Please enter the associated assembly id for the job:

79

Please enter the associated process id for the job:

429

Done. 1 rows inserted.

Content of Job Table after 10 queries of Query#6

Results Messages					
	job_number	date_commenced	date_compl...	process_id	assembly_id
1	22	2020-04-02	NULL	420	69
2	23	2020-01-05	NULL	421	70
3	24	2020-01-04	NULL	422	72
4	25	2020-01-02	NULL	423	73
5	26	2020-01-02	NULL	424	75
6	27	2020-01-02	NULL	424	76
7	28	2020-01-02	NULL	426	77
8	29	2020-01-02	NULL	427	79
9	30	2020-01-02	NULL	428	79
1...	31	2020-01-02	NULL	429	79

10 Queries of Query#7

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together

- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

22

Please enter date completed for job in yyyy-mm-dd format:

2020-07-04

Please enter type of job (Fit, Paint, Cut)?

Fit

Please enter labor time in hh:mm:ss format

07:45:09

Done. 1 rows inserted.

Done. 1 rows inserted. Fit Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

23

Please enter date completed for job in yyyy-mm-dd format:

2020-07-27

Please enter type of job (Fit, Paint, Cut)?

Paint

Please enter labor time in hh:mm:ss format

09:42:09

Please enter color for Paint job:

Red

Please enter the volume of paint:

25

Done. 1 rows inserted.

Done. 1 rows inserted. Paint Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department

- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

24

Please enter date completed for job in yyyy-mm-dd format:

2020-07-09

Please enter type of job (Fit, Paint, Cut)?

Cut

Please enter labor time in hh:mm:ss format

09:04:04

Please enter the type of machine used:

Laser Cutter

Please enter amount of time machine used in hh:mm:ss format

05:02:05

Please enter material used

Metal

Done. 1 rows inserted.

Done. 1 rows inserted. Cut Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

25

Please enter date completed for job in yyyy-mm-dd format:

2020-03-20

Please enter type of job (Fit, Paint, Cut)?

Fit

Please enter labor time in hh:mm:ss format

09:24:09

Done. 1 rows inserted.

Done. 1 rows inserted. Fit Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department

- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

26

Please enter date completed for job in yyyy-mm-dd format:

2020-04-01

Please enter type of job (Fit, Paint, Cut)?

Paint

Please enter labor time in hh:mm:ss format

09:09:05

Please enter color for Paint job:

Red

Please enter the volume of paint:

30

Done. 1 rows inserted.

Done. 1 rows inserted. Paint Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

27

Please enter date completed for job in yyyy-mm-dd format:

2020-03-27

Please enter type of job (Fit, Paint, Cut)?

Cut

Please enter labor time in hh:mm:ss format

09:04:24

Please enter the type of machine used:

Laser

Please enter amount of time machine used in hh:mm:ss format

02:02:02

Please enter material used

Wood

Done. 1 rows inserted.

Done. 1 rows inserted. Cut Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer

- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

28

Please enter date completed for job in yyyy-mm-dd format:

2020-09-10

Please enter type of job (Fit, Paint, Cut)?

Fit

Please enter labor time in hh:mm:ss format

04:04:04

Done. 1 rows inserted.

Done. 1 rows inserted. Fit Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

29

Please enter date completed for job in yyyy-mm-dd format:

2020-09-10

Please enter type of job (Fit, Paint, Cut)?

Paint

Please enter labor time in hh:mm:ss format

04:52:42

Please enter color for Paint job:

Pink

Please enter the volume of paint:

40

Done. 1 rows inserted.

Done. 1 rows inserted. Paint Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together

- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

30

Please enter date completed for job in yyyy-mm-dd format:

2020-09-10

Please enter type of job (Fit, Paint, Cut)?

Cut

Please enter labor time in hh:mm:ss format

04:09:09

Please enter the type of machine used:

Laser

Please enter amount of time machine used in hh:mm:ss format

02:02:04

Please enter material used

Plastic

Done. 1 rows inserted.

Done. 1 rows inserted. Cut Job Added.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

7

Please enter the job number completed :

31

Please enter date completed for job in yyyy-mm-dd format:

2020-09-10

Please enter type of job (Fit, Paint, Cut)?

Fit

Please enter labor time in hh:mm:ss format

09:09:53

Done. 1 rows inserted.

Done. 1 rows inserted. Fit Job Added.

Content of Job Table after 10 queries of Query#7

	Results	Messages			
	job_number	date_commenced	date_completed	process_id	assembly_id
1	22	2020-04-02	2020-07-04	420	69
2	23	2020-01-05	2020-07-27	421	70
3	24	2020-01-04	2020-07-09	422	72
4	25	2020-01-02	2020-03-20	423	73
5	26	2020-01-02	2020-04-01	424	75
6	27	2020-01-02	2020-03-27	424	76
7	28	2020-01-02	2020-09-10	426	77
8	29	2020-01-02	2020-09-10	427	79
9	30	2020-01-02	2020-09-10	428	79
10	31	2020-01-02	2020-09-10	429	79

Content of Fit Job Table after 10 queries of Query#7

	Results	Messages
	job_number	labor_time
1	22	07:45:09
2	25	09:24:09
3	28	04:04:04
4	31	09:09:53

Content of Paint Job Table after 10 queries of Query#7

Results		Messages		
	job_number	labor_time	color	volume
1	23	09:42:09	Red	25
2	26	09:09:05	Red	30
3	29	04:52:42	Pink	40

Content of Cut Job Table after 10 queries of Query#7

Results		Messages		
	job_number	labor_time	type_machi...	amount_of_time_m...
1	24	09:04:04	Laser Cu...	05:02:05
2	27	09:04:24	Laser	02:02:02
3	30	04:09:09	Laser	02:02:04

## 10 Queries of Query#8

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

35

Please enter cost:

100

Enter the associated account number:

2000

Please enter the associated job number:

22

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

36

Please enter cost:

200

Enter the associated account number:

20001

Please enter the associated job number:

23

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

37

Please enter cost:

100

Enter the associated account number:

20002

Please enter the associated job number:

24

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

38

Please enter cost:

130

Enter the associated account number:

2003

Please enter the associated job number:

25

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

39

Please enter cost:

102

Enter the associated account number:

2004

Please enter the associated job number:

26

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer

- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

40

Please enter cost:

105

Enter the associated account number:

2005

Please enter the associated job number:

27

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

41

Please enter cost:

105

Enter the associated account number:

2006

Please enter the associated job number:

28

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together

- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

42

Please enter cost:

900

Enter the associated account number:

2007

Please enter the associated job number:

29

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

43

Please enter cost:

800

Enter the associated account number:

2008

Please enter the associated job number:

30

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department

- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

8

Please enter new transaction number

44

Please enter cost:

700

Enter the associated account number:

2009

Please enter the associated job number:

31

Done. 1 rows inserted.

Done. 1 rows inserted. Assembly Account updated

Done. 1 rows inserted. Process Account updated

Done. 1 rows inserted. Department Account updated

#### Content of Assembly Account Table after 10 queries of Query#8

	Results	Messages		
	account_number	date_account_established	cost1	assembly_id
1	2000	2020-01-02	100	69
2	2003	2020-01-03	130	73
3	2004	2020-01-02	102	75
4	2005	2020-01-04	105	76
5	2006	2020-01-01	105	77
6	2007	2020-01-01	900	79
7	2008	2020-01-02	800	79
8	2009	2020-01-02	700	79
9	20001	2020-01-04	200	70
10	20002	2020-04-01	100	72

#### Content of Process Account Table after 10 queries of Query#8

**Results    Messages**

	account_number	date_account_established	cost2	process_id
1	2000	2020-01-02	100	420
2	2003	2020-01-03	130	423
3	2004	2020-01-02	102	424
4	2005	2020-01-04	105	425
5	2006	2020-01-01	105	426
6	2007	2020-01-01	900	427
7	2008	2020-01-02	800	428
8	2009	2020-01-02	700	429
9	20001	2020-01-04	200	421
10	20002	2020-04-01	100	422

Content of Department Account Table after 10 queries of Query#8

**Results    Messages**

	account_number	date_account_established	cost3	department_number
1	2000	2020-01-02	100	1
2	2003	2020-01-03	130	2
3	2004	2020-01-02	102	2
4	2005	2020-01-04	105	2
5	2006	2020-01-01	105	3
6	2007	2020-01-01	900	3
7	2008	2020-01-02	800	4
8	2009	2020-01-02	700	5
9	20001	2020-01-04	200	1
10	20002	2020-04-01	100	1

3 Queries of Query#9

```

totstr WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
te.cl (1) Enter a new customer
harac (2) Enter a new department
harac (3) Enter a new process and department together
harac (4) Enter a new assembly and associate it with one or more processes
harac (5) Create a new account and associate it with a process, assembly, or department
harac (6) Enter a new job and date the job began
harac (7) At the completion of a job, enter the date completed and relevant info
harac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
harac (9) Retrieve the total cost incurred on a specific assembly
harac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
harac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
harac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
harac (13) Retrieve the customers(in name order) whose category is in a specific range
harSe (14) Delete all cut-jobs whose job-no is in a specific range
ass.c (15) Change the color of a specific paint job
assC (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
assC (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
assC (18) Quit
assF
assLc 9
assLc Please enter assembly ID you want to find cost of:
assN 75
assV The cost associated with the assembly ID is 102
assV
ytc.cl WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
harac (1) Enter a new customer
harac (2) Enter a new department
harac (3) Enter a new process and department together
harac (4) Enter a new assembly and associate it with one or more processes
harac (5) Create a new account and associate it with a process, assembly, or department
harac (6) Enter a new job and date the job began
harac (7) At the completion of a job, enter the date completed and relevant info
harac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
harac (9) Retrieve the total cost incurred on a specific assembly
harac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
harac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
harac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
harSe (13) Retrieve the customers(in name order) whose category is in a specific range
lass.c (14) Delete all cut-jobs whose job-no is in a specific range
lassC (15) Change the color of a specific paint job
lassC (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
lassC (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
lassC (18) Quit
lassLc
lassLc 9
lassLc Please enter assembly ID you want to find cost of:
lassN 70
lassV The cost associated with the assembly ID is 200
lassV
cl WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
ac (1) Enter a new customer
ac (2) Enter a new department
ac (3) Enter a new process and department together
ac (4) Enter a new assembly and associate it with one or more processes
ac (5) Create a new account and associate it with a process, assembly, or department
ac (6) Enter a new job and date the job began
ac (7) At the completion of a job, enter the date completed and relevant info
ac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
ac (9) Retrieve the total cost incurred on a specific assembly
ac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
ac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
ac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
Se (13) Retrieve the customers(in name order) whose category is in a specific range
sc (14) Delete all cut-jobs whose job-no is in a specific range
sc (15) Change the color of a specific paint job
sc (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
sc (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
sc (18) Quit
scLc
scLc 9
scLc Please enter assembly ID you want to find cost of:
scN 69
scV The cost associated with the assembly ID is 100
scV

```

### 3 Queries of Query#10

```

rac WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
rac (1) Enter a new customer
rac (2) Enter a new department
rac (3) Enter a new process and department together
rac (4) Enter a new assembly and associate it with one or more processes
rac (5) Create a new account and associate it with a process, assembly, or department
rac (6) Enter a new job and date the job began
rac (7) At the completion of a job, enter the date completed and relevant info
rac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
rac (9) Retrieve the total cost incurred on a specific assembly
rac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
rSe (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
rSe (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
s.c (13) Retrieve the customers(in name order) whose category is in a specific range
scC (14) Delete all cut-jobs whose job-no is in a specific range
scC (15) Change the color of a specific paint job
scF (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
scF (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
scL (18) Quit
scL

scN 10
scV1 Please enter the department number:
scV1 1
scE1 Please enter the completion date in yyyy-mm-dd format:
scN1 2020-07-27
scP1 Connecting to the database...
scP1 The total labor time for jobs completed in the department during a given date is 582

harac WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
harac (1) Enter a new customer
harac (2) Enter a new department
harac (3) Enter a new process and department together
harac (4) Enter a new assembly and associate it with one or more processes
harac (5) Create a new account and associate it with a process, assembly, or department
harac (6) Enter a new job and date the job began
harac (7) At the completion of a job, enter the date completed and relevant info
harac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
harac (9) Retrieve the total cost incurred on a specific assembly
harac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
harSe (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
harSe (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
ass.c (13) Retrieve the customers(in name order) whose category is in a specific range
assC (14) Delete all cut-jobs whose job-no is in a specific range
assC (15) Change the color of a specific paint job
assF (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
assF (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
assL (18) Quit
assL

assN 10
assV1 Please enter the department number:
assV1 2
onea1 Please enter the completion date in yyyy-mm-dd format:
oneN1 2020-03-20
oneP1 Connecting to the database...
oneP1 The total labor time for jobs completed in the department during a given date is 564

c WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
c (1) Enter a new customer
c (2) Enter a new department
c (3) Enter a new process and department together
c (4) Enter a new assembly and associate it with one or more processes
c (5) Create a new account and associate it with a process, assembly, or department
c (6) Enter a new job and date the job began
c (7) At the completion of a job, enter the date completed and relevant info
c (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
c (9) Retrieve the total cost incurred on a specific assembly
c (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
c (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
c (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
c (13) Retrieve the customers(in name order) whose category is in a specific range
c (14) Delete all cut-jobs whose job-no is in a specific range
c (15) Change the color of a specific paint job
c (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
c (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
c (18) Quit
c

N 10
Please enter the department number:
5
D Please enter the completion date in yyyy-mm-dd format:
N 2020-09-10
a Connecting to the database...
ii The total labor time for jobs completed in the department during a given date is 549

```

### 3 Queries of Query#11

```

dotst
yte.cl WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
harac (1) Enter a new customer
harac (2) Enter a new department
harac (3) Enter a new process and department together
harac (4) Enter a new assembly and associate it with one or more processes
harac (5) Create a new account and associate it with a process, assembly, or department
harac (6) Enter a new job and date the job began
harac (7) At the completion of a job, enter the date completed and relevant info
harac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
harac (9) Retrieve the total cost incurred on a specific assembly
harac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
harac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
harac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
harSe (13) Retrieve the customers(in name order) whose category is in a specific range
lass.c (14) Delete all cut-jobs whose job-no is in a specific range
lassC (15) Change the color of a specific paint job
lassC (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
lassC (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
lassF (18) Quit
lassL 11
lassL Please enter the assembly id:
lassN 79
lassV Connecting to the database...
loneal Processes through which a given assembly ID has passed so far in date commenced order: |
loneN Process ID | Department Number | Date Commenced
ompas 427 | 3 | 2020-01-02
ompas 428 | 4 | 2020-01-02
ompas 429 | 5 | 2020-01-02
otstr WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
e.cl (1) Enter a new customer
arac (2) Enter a new department
arac (3) Enter a new process and department together
arac (4) Enter a new assembly and associate it with one or more processes
arac (5) Create a new account and associate it with a process, assembly, or department
arac (6) Enter a new job and date the job began
arac (7) At the completion of a job, enter the date completed and relevant info
arac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
arac (9) Retrieve the total cost incurred on a specific assembly
arac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
arac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
arac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
arac (13) Retrieve the customers(in name order) whose category is in a specific range
arSe (14) Delete all cut-jobs whose job-no is in a specific range
ss.c (15) Change the color of a specific paint job
ssC (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
ssC (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
ssC (18) Quit
ssF 11
ssL Please enter the assembly id:
ssL 73
ssN Connecting to the database...
ssV Processes through which a given assembly ID has passed so far in date commenced order:
nea Process ID | Department Number | Date Commenced
neN 423 | 2 | 2020-01-02
ear WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
str (1) Enter a new customer
(str (2) Enter a new department
.cl (3) Enter a new process and department together
.ac (4) Enter a new assembly and associate it with one or more processes
.ac (5) Create a new account and associate it with a process, assembly, or department
.ac (6) Enter a new job and date the job began
.ac (7) At the completion of a job, enter the date completed and relevant info
.ac (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
.ac (9) Retrieve the total cost incurred on a specific assembly
.ac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
.ac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
.ac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
.ac (13) Retrieve the customers(in name order) whose category is in a specific range
.ac (14) Delete all cut-jobs whose job-no is in a specific range
.ac (15) Change the color of a specific paint job
Se (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
.s.c (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
.e.c (18) Quit
sc 11
sc Please enter the assembly id:
sc 69
sc Connecting to the database...
sn Processes through which a given assembly ID has passed so far in date commenced order:
sn Process ID | Department Number | Date Commenced
snV 420 | 1 | 2020-04-02

```

### 3 Queries of Query#12

```
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
(18) Quit

12
Please enter the completion date in yyyy-mm-dd format:
2020-03-27
Please enter the department number:
2
Connecting to the database...
Fit Jobs completed during a given date in a given department:
Job Number | Assembly ID | Labor Time (hh:mm:ss)

Paint Jobs completed during a given date in a given department:
Job Number | Assembly ID | color | volume | Labor Time (hh:mm:ss)

Cut Jobs completed during a given date in a given department:
Job Number | Assembly ID | Type Machine Used | Amount of Time Machine Used (hh:mm:ss) | Material Used | Labor Time (hh:mm:ss)
27 | 76 | Laser | 02:02:02 | Wood | 09:04:24

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
(18) Quit

12
Please enter the completion date in yyyy-mm-dd format:
2020-07-04
Please enter the department number:
1
Connecting to the database...
Fit Jobs completed during a given date in a given department:
Job Number | Assembly ID | Labor Time (hh:mm:ss)
22 | 69 | 07:45:09

Paint Jobs completed during a given date in a given department:
Job Number | Assembly ID | color | volume | Labor Time (hh:mm:ss)

Cut Jobs completed during a given date in a given department:
Job Number | Assembly ID | Type Machine Used | Amount of Time Machine Used (hh:mm:ss) | Material Used | Labor Time (hh:mm:ss)
```

```

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
(18) Quit

12
Please enter the completion date in yyyy-mm-dd format:
2020-09-10
Please enter the department number:
3
Connecting to the database...
Fit Jobs completed during a given date in a given department:
Job Number | Assembly ID | Labor Time (hh:mm:ss)
28 | 77 | 04:04:04

Paint Jobs completed during a given date in a given department:
Job Number | Assembly ID | color | volume | Labor Time (hh:mm:ss)
29 | 79 | Pink | 40 | 04:52:42

Cut Jobs completed during a given date in a given department:
Job Number | Assembly ID | Type Machine Used | Amount of Time Machine Used (hh:mm:ss) | Material Used | Labor Time (hh:mm:ss)

```

### 3 Queries of Query#13

```

bl WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
bl (1) Enter a new customer
bl (2) Enter a new department
bl (3) Enter a new process and department together
bl (4) Enter a new assembly and associate it with one or more processes
bl (5) Create a new account and associate it with a process, assembly, or department
bl (6) Enter a new job and date the job began
bl (7) At the completion of a job, enter the date completed and relevant info
bl (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
bl (9) Retrieve the total cost incurred on a specific assembly
bl (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
bl (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
bl (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
bl (13) Retrieve the customers(in name order) whose category is in a specific range
bl (14) Delete all cut-jobs whose job-no is in a specific range
bl (15) Change the color of a specific paint job
bl (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
bl (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
bl (18) Quit
bl

bl 13
bl Please enter desired lower bound of category range:
bl 7
bl Please enter desired upper bound of category range:
bl 10
bl Connecting to the database...
bl Customers (in name order)
bl .c Buzz Lightyear
bl .c Pope Stevens
bl

ayln WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
ayS (1) Enter a new customer
ayS (2) Enter a new department
ayti (3) Enter a new process and department together
ayti (4) Enter a new assembly and associate it with one or more processes
ayC (5) Create a new account and associate it with a process, assembly, or department
ayl (6) Enter a new job and date the job began
ayl (7) At the completion of a job, enter the date completed and relevant info
aytr (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
aycl (9) Retrieve the total cost incurred on a specific assembly
ayrac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
ayrac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
ayrac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
ayrac (13) Retrieve the customers(in name order) whose category is in a specific range
ayrac (14) Delete all cut-jobs whose job-no is in a specific range
ayrac (15) Change the color of a specific paint job
ayrac (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
ayrac (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
ayrac (18) Quit
ayrac

ayrac 13
ayrac Please enter desired lower bound of category range:
aySe 3
aysc Please enter desired upper bound of category range:
aysc 6
aysc Connecting to the database...
aysc Customers (in name order)
aysc .c Bob Hope
aysc .c Koi Fish
aysc .c Mike Tyson
aysc

aySt WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
ayti (1) Enter a new customer
ayti (2) Enter a new department
ayC (3) Enter a new process and department together
ayl (4) Enter a new assembly and associate it with one or more processes
aytr (5) Create a new account and associate it with a process, assembly, or department
ayl (6) Enter a new job and date the job began
ayl (7) At the completion of a job, enter the date completed and relevant info
aytr (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
ayrac (9) Retrieve the total cost incurred on a specific assembly
ayrac (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
ayrac (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
ayrac (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
ayrac (13) Retrieve the customers(in name order) whose category is in a specific range
ayrac (14) Delete all cut-jobs whose job-no is in a specific range
ayrac (15) Change the color of a specific paint job
ayrac (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
ayrac (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
ayrac (18) Quit
ayrac

aySe 13
aysc Please enter desired lower bound of category range:
aysc 1
aysc Please enter desired upper bound of category range:
aysc 4
aysc Connecting to the database...
aysc Customers (in name order)
aysc .c Bob Hope
aysc .c Mike Tyson
aysc

```

### 3 Queries of Query#14

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department

- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

14

Please enter desired lower bound of job number range:

20

Please enter desired upper bound of job number range:

24

Connecting to the database...

Job numbers from 20 to 24 are deleted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

14

Please enter desired lower bound of job number range:

29

Please enter desired upper bound of job number range:

31

Connecting to the database...

Job numbers from 29 to 31 are deleted.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)

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

14  
 Please enter desired lower bound of job number range:

1  
 Please enter desired upper bound of job number range:

10  
 Connecting to the database...

Job numbers from 1 to 10 are deleted.

#### Original Cut Job Table of Query#14

Results		Messages		
	job_number	labor_time	type_machine_...	amount_of...
1	24	09:04:04	Laser Cutter	05:02:05
2	27	09:04:24	Laser	02:02:02
3	30	04:09:09	Laser	02:02:04

#### Cut Job Table After first Query of Query#14

Results		Messages		
	job_number	labor_time	type_machine_used	amount_of_time_machine_used
1	27	09:04:24	Laser	02:02:02
2	30	04:09:09	Laser	02:02:04

#### Cut Job Table After second Query of Query#14

Results		Messages		
	job_number	labor_time	type_machine_used	amount_of_time_machine_used
1	27	09:04:24	Laser	02:02:02

#### Cut Job Table After third Query of Query#14

Results		Messages		
	job_number	labor_time	type_machine_used	amount_of_time_machine_used
1	27	09:04:24	Laser	02:02:02

#### 3 Queries of Query#15

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer  
 (2) Enter a new department

- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

15

What is the paint job number you wish to update:

23

Please enter the color you wish to update to:

Lightning Yellow

Connecting to the database...

Done. 1 rows updated.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

15

What is the paint job number you wish to update:

26

Please enter the color you wish to update to:

Purple

Connecting to the database...

Done. 1 rows updated.

#### WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)

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

15

What is the paint job number you wish to update:

29

Please enter the color you wish to update to:

Sooner Crimson

Connecting to the database...

Done. 1 rows updated.

### Original Paint Job Table of Query#15

Results Messages				
	job_number	labor_time	color	volume
1	23	09:42:09	Red	25
2	26	09:09:05	Red	30
3	29	04:52:42	Pink	40

### Original Paint Job Table of Query#15

Results Messages				
	job_number	labor_time	color	volume
1	23	09:42:09	Lightning Yellow	25
2	26	09:09:05	Red	30
3	29	04:52:42	Pink	40

### Original Paint Job Table of Query#15

Results grid Messages				
	job_number	labor_time	color	volume
1	23	09:42:09	Lightning Yellow	25
2	26	09:09:05	Purple	30
3	29	04:52:42	Pink	40

### Original Paint Job Table of Query#15

Purple Results Messages

	job_number	labor_time	color	volume
1	23	09:42:09	Lightning Yellow	25
2	26	09:09:05	Purple	30
3	29	04:52:42	Sooner Crimson	40

## 1 Query of Query#16

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

16

```
Enter the file-name to import data.
CustomerInput.txt
Connecting to the database...
Done. 1 rows inserted.
Connecting to the database...
Done. 1 rows inserted.]
```

Input text

The screenshot shows a Java IDE interface with three tabs open: sample.java, DatabaseP.java, and CustomerInput.txt. The CustomerInput.txt tab contains the following text:

```

1 Troy Aikman, 1504 Oakhurst Street Norman Oklahoma 73170,2
2 Mike Shephard, 1505 Oakhurst Street Norman Oklahoma 73170,3
3 Kat Kit, 1506 Oakhurst Street Norman Oklahoma 73170,4
4 Daniel Kit, 1507 Oakhurst Street Norman Oklahoma 73170,5
5 Michael Jackson, 1508 Oakhurst Street Norman Oklahoma 73170,7
6 Michael Stefner, 1509 Oakhurst Street Norman Oklahoma 73170,6
7 Lamar Jackson, 1510 Oakhurst Street Norman Oklahoma 73170,8

```

Customer Table after 1 Query of Query#16

The screenshot shows a database query results window titled "Results". The table has columns: name, address, and category. The data is as follows:

	name	address	category
1	Bob Hope	1098 Wa...	3
2	Buzz Li...	10922 M...	7
3	Daniel ...	1507 O...	5
4	Kat Kit	1506 O...	4
5	Koi Fish	8958 Ge...	6
6	Lamar J...	1510 O...	8
7	Michael...	1508 O...	7
8	Michael...	1509 O...	6
9	Mike Sh...	1505 O...	3
1...	Mike Ty...	9998 Ok...	4
1...	Pope St...	90412 N...	10
1...	Troy Ai...	1504 O...	2

## 1 Query of Query#17

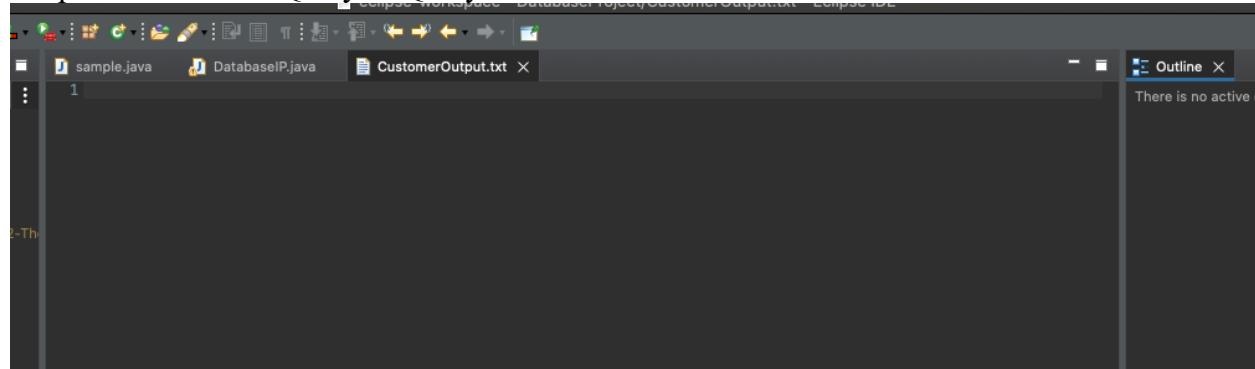
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date

- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
- (13) Retrieve the customers(in name order) whose category is in a specific range
- (14) Delete all cut-jobs whose job-no is in a specific range
- (15) Change the color of a specific paint job
- (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
- (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
- (18) Quit

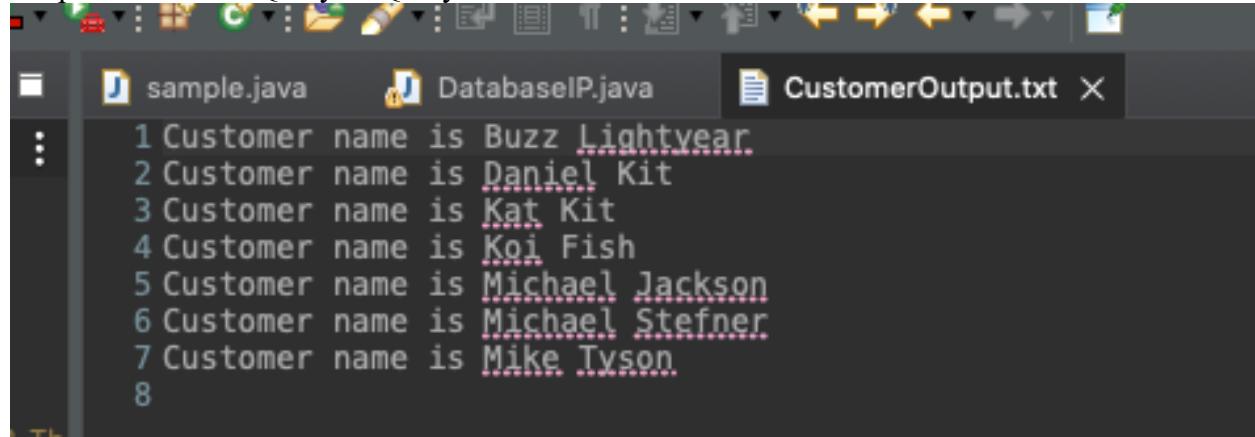
```
17
Enter the file-name to Export data.
CustomerOutput.txt
Enter the lower bound of category.
4
Enter the upper bound of category.
7
Connecting to the database...
```

### Output File before 1 Query of Query#17



```
1
```

### Output File after 1 Query of Query#17



```
1 Customer name is Buzz Lightyear
2 Customer name is Daniel Kit
3 Customer name is Kat Kit
4 Customer name is Koi Fish
5 Customer name is Michael Jackson
6 Customer name is Michael Stefner
7 Customer name is Mike Tyson
8
```

### 1 Query of Query#18

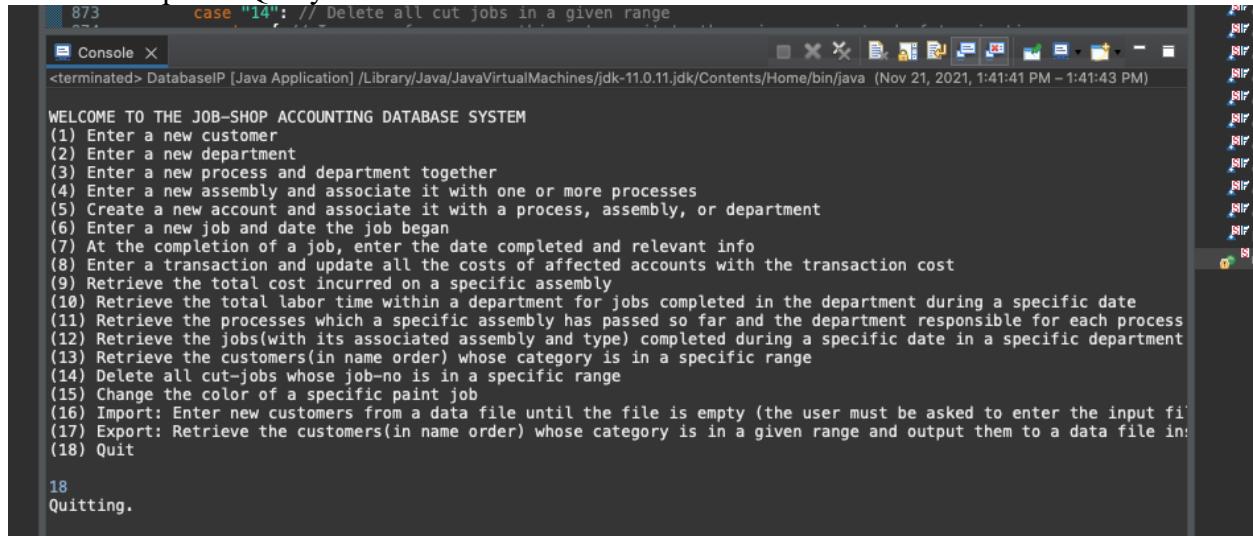
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process and department together
- (4) Enter a new assembly and associate it with one or more processes
- (5) Create a new account and associate it with a process, assembly, or department
- (6) Enter a new job and date the job began
- (7) At the completion of a job, enter the date completed and relevant info
- (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
- (9) Retrieve the total cost incurred on a specific assembly
- (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
- (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
- (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department

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

18

## Console output of Query#18



```

873 case "14": // Delete all cut jobs in a given range
874
875 Console X
876 <terminated> DatabaseIP [Java Application] /Library/Java/JavaVirtualMachines/jdk-11.0.11.jdk/Contents/Home/bin/java (Nov 21, 2021, 1:41:41 PM - 1:41:43 PM)
877
878 WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
879 (1) Enter a new customer
880 (2) Enter a new department
881 (3) Enter a new process and department together
882 (4) Enter a new assembly and associate it with one or more processes
883 (5) Create a new account and associate it with a process, assembly, or department
884 (6) Enter a new job and date the job began
885 (7) At the completion of a job, enter the date completed and relevant info
886 (8) Enter a transaction and update all the costs of affected accounts with the transaction cost
887 (9) Retrieve the total cost incurred on a specific assembly
888 (10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
889 (11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
890 (12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
891 (13) Retrieve the customers(in name order) whose category is in a specific range
892 (14) Delete all cut-jobs whose job-no is in a specific range
893 (15) Change the color of a specific paint job
894 (16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file name)
895 (17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file instead of screen
896 (18) Quit
897
898 18
899 Quitting.

```

Error #1 (Entering a string for department number)

```

-12-Th Database - (Java Application) / Library/Java/java/lib/tools/machine/jdk-11.0.1/jdk/Contents/Home/bin/java (Nov 21, 2021, 14:51:26 PM)

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file in:
(18) Quit

2
Please enter new department number:
Bob
You got an error! Returning to the main menu.

-1.8] WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file in:
(18) Quit

Unrecognized option: Bob
Please try again!

```

Error #2 (Associating a Process to a department that doesn't exist)

Department Table

Account	Results		Messages
	department	department	
	1	1	This de...
	2	2	The und...
+ ⚡	3	3	The dep...
Trinh-1...	4	4	The dep...
	5	5	The dep...

Error #2 Console Output

```

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file in
(18) Quit

3
Please enter a new process ID:
99
Please enter the corresponding process_data:
Nothing, this is gonna be an error test
Please enter new department number associated with process:
10
Please enter type of process (Fit, Paint, Cut)?
Fit
Please enter the corresponding fit_type:
labor
Connecting to the database...
You got an error! Returning to the main menu.

```

## Error #2 (Adding a duplicate primary key)

```

(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file in
(18) Quit

1
Please enter name of the new customer:
Darth Vader
Please enter address of the new customer:
Nothing
Please enter Customer category (a number from 1-10):
3
Connecting to the database...
Done. 1 rows inserted.

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process and department together
(4) Enter a new assembly and associate it with one or more processes
(5) Create a new account and associate it with a process, assembly, or department
(6) Enter a new job and date the job began
(7) At the completion of a job, enter the date completed and relevant info
(8) Enter a transaction and update all the costs of affected accounts with the transaction cost
(9) Retrieve the total cost incurred on a specific assembly
(10) Retrieve the total labor time within a department for jobs completed in the department during a specific date
(11) Retrieve the processes which a specific assembly has passed so far and the department responsible for each process
(12) Retrieve the jobs(with its associated assembly and type) completed during a specific date in a specific department
(13) Retrieve the customers(in name order) whose category is in a specific range
(14) Delete all cut-jobs whose job-no is in a specific range
(15) Change the color of a specific paint job
(16) Import: Enter new customers from a data file until the file is empty (the user must be asked to enter the input file)
(17) Export: Retrieve the customers(in name order) whose category is in a given range and output them to a data file in
(18) Quit

1
Please enter name of the new customer:
Darth Vader
Please enter address of the new customer:
Nothing
Please enter Customer category (a number from 1-10):
3
Connecting to the database...
You got an error! Returning to the main menu.

```

# Web App Source Program

## 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;

    // Initialize and save the database connection

    final static String url = "jdbc:sqlserver://trin0003-sql-server.database.windows.net:1433;database=cs-dsa-4513-sql-
db;user=trin0003@trin0003-sql-
server;password=ClearMap2013$;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;";

    private void getDBConnection() throws SQLException {
        if (conn != null) {
            return;
        }
        this.conn = DriverManager.getConnection(url);
    }

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

    // Return the result of selecting Customers with their category in the given range from Customer table
    public ResultSet retrieveCustomers(int lower_b, int upper_b) throws SQLException {
        getDBConnection(); // Prepare the database connection
        // Prepare the SQL statement
        final String sqlQuery = "SELECT name, category AS name FROM Customer" +
                               " WHERE category >= ? AND category <= ? ORDER BY 1 ;";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        // Replace the '?' in the above statement with the given attribute values
        stmt.setInt(1, lower_b);
        stmt.setInt(2, upper_b);
        // Execute the query
        return stmt.executeQuery();
    }

    // Inserts a record into the Customer table with the given attribute values
    public boolean addCustomer( String name, String address, int category) throws SQLException {
        getDBConnection(); // Prepare the database connection
        // Prepare the SQL statement
        final String sqlQuery =
                "INSERT INTO Customer " +
                "(name, address, category) " +
                "VALUES " +
                "(?, ?, ?)";

        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        // Replace the '?' in the above statement with the given attribute values
        stmt.setString(1, name);
        stmt.setString(2, address);
        stmt.setInt(3, category);
        // Execute the query, if only one record is updated, then we indicate success by returning true
        return stmt.executeUpdate() == 1;
    }
}
```

## Add new customer form.java

```

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

```

## Add new customer.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"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandler handler = new DataHandler();
// Get the attribute values passed from the input form.
String name = request.getParameter("name");
String address = request.getParameter("address");
String category = request.getParameter("category");
/*
* If the user hasn't filled out all the cname, address and category. This is very
simple
checking.
*/
if (name.equals("") || address.equals("") || category.equals("")) {
response.sendRedirect("add_new_customer_form.jsp");
} else {

int category1 = Integer.parseInt(category);
// Now perform the query with the data from the form.
boolean success = handler.addCustomer(name, address, category1);
if (!success) { // Something went wrong
%>
<h2>There was a problem inserting the course</h2>
<%
} else { // Confirm success to the user
%>
<h2>New Customer Details:</h2>
Page 23 of 23
<ul>
<li>Customer Name: <%=name%></li>
<li>Address: <%=address%></li>
<li>Category: <%=category%></li>
</ul>
<h2>Was successfully inserted.</h2>
<a href="get_all_customers.jsp">See all Customers.</a>
<%
}
}
}
%>
</body>
</html>

```

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

```

## Retrieve customer form.jsp

```

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

```

```

<input type="text" name="lower_b">
</div></td>
</tr>
<tr>
<td>Upper Bound (Inclusive):</td>
<td><div style="text-align: center;">
<input type="text" name="upper_b">
</div></td>
</tr>
<tr>
<td><div style="text-align: center;">
<input type="reset" value="Clear">
</div></td>
<td><div style="text-align: center;">
<input type="submit" value="Insert">
</div></td>
</tr>
</table>
</form>
</body>
</html>

```

## Retrieve customers.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 customers from
the database
            final DataHandler handler = new DataHandler();
            // Get the attribute values passed from the input form.
            String lower_b = request.getParameter("lower_b");
            String upper_b = request.getParameter("upper_b");

            if (lower_b.equals("") || upper_b.equals("")) {
                response.sendRedirect("retrieve_customer_form.jsp");
            } else {
                int lower_b1 = Integer.parseInt(lower_b);
                int upper_b1 = Integer.parseInt(upper_b);
                // Now perform the query with the data from the form.
                final ResultSet customers = handler.retrieveCustomers(lower_b1,
upper_b1);
            }
        %>
        <!-- The table for displaying all the Customer records -->
        <table cellspacing="2" cellpadding="2" border="1">
            <tr> <!-- The table headers row -->
                <td align="center">
                    <h4>Customer</h4>
                </td>
                <td align="center">
                    <h4>category</h4>
                </td>

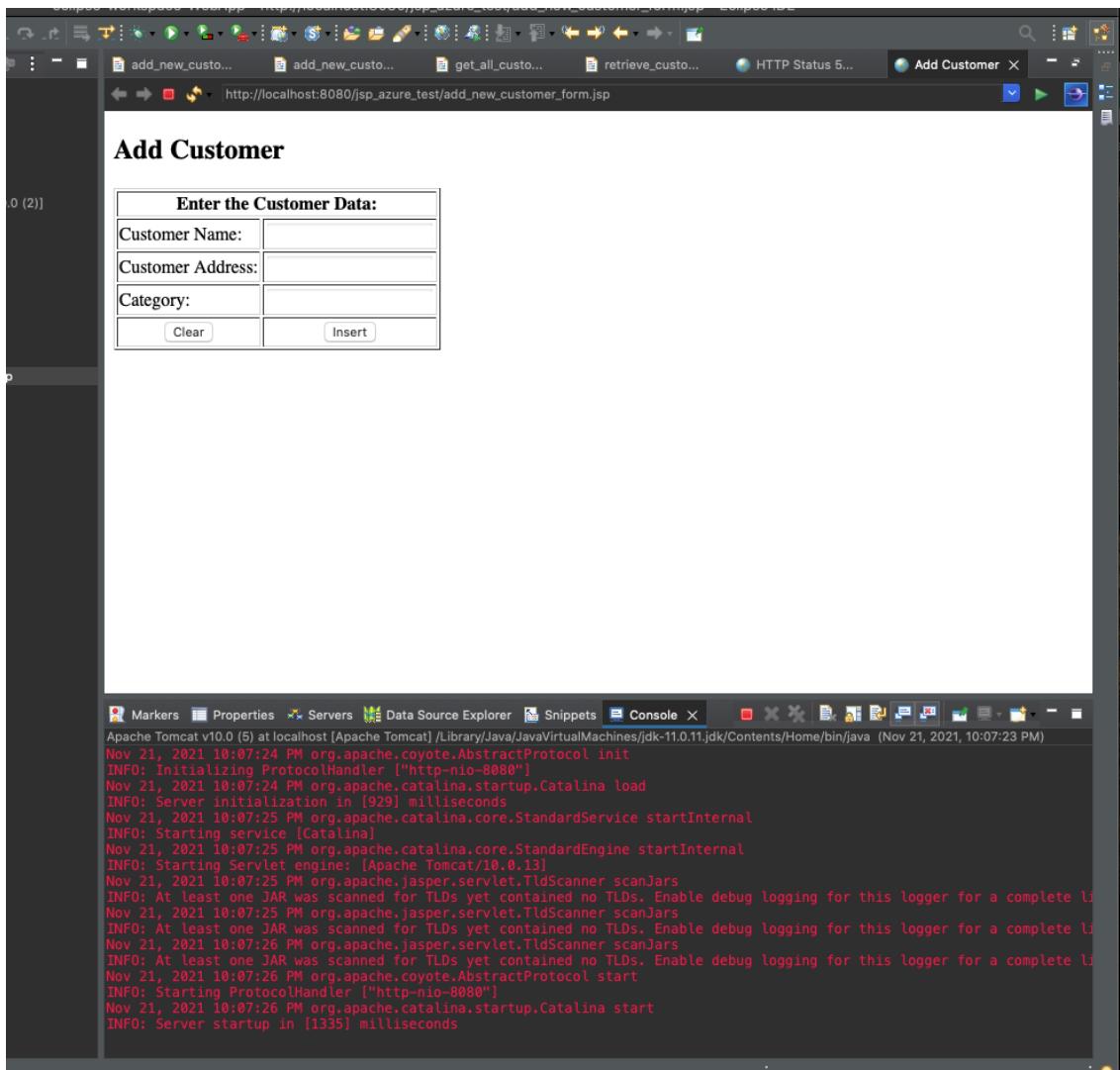
```

```

<%
    while(customers.next()) { // Read out all customer records
        // Get values from the ResultSet to display customers
        final String name = customers.getString(1);
        final String category = customers.getString(2);
        out.println("<tr>"); // Start printing out the new table
        row++;
        out.println( // Print each attribute value
            "<td align=\"center\">" + name +
            "</td><td align=\"center\"> " + category + "</td>");
        out.println("</tr>");
    }
%>
</table>
</body>
</html>

```

## Screenshot for Successful Compilation



## Screenshot for Retrieve All Customers

Customer	Address	Category
Bob Hope	1098 Washington Avenue, Oklahoma City, Oklahoma, 70941	3
Buzz Lightyear	10922 Main St., Garden Grove, California, 90210	7
Daniel Kit	1507 Oakhurst Street Norman Oklahoma 73170	5
Darth Vader	Nothing	3
Kat Kit	1506 Oakhurst Street Norman Oklahoma 73170	4
Koi Fish	8958 Geico Trail, Nowhere, Oklahoma, 90423	6
Lamar Jackson	1510 Oakhurst Street Norman Oklahoma 73170	8
Michael Jackson	1508 Oakhurst Street Norman Oklahoma 73170	7
Michael Stefner	1509 Oakhurst Street Norman Oklahoma 73170	6
Mike Shephard	1505 Oakhurst Street Norman Oklahoma 73170	3
Mike Tyson	9998 Oklahoma St., Morgantown, West Virginia, 12424	4
Pope Stevens	90412 Norman Avenue, Norman, Oklahoma, 70891	10
Troy Aikman	1504 Oakhurst Street Norman Oklahoma 73170	2

## Screenshot for Query 13

### Give Customers Range

Enter the Category Range:	
Lower Bound:	2
Upper Bound:	5
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

Customer	category
Bob Hope	3
Daniel Kit	5
Darth Vader	3
Kat Kit	4
Mike Shephard	3
Mike Tyson	4
Troy Aikman	2

### Screenshot for Query 1

### Add Customer

Enter the Customer Data:	
Customer Name:	Bob
Customer Address:	Bob's Place
Category:	3
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

## The Customer Details:

Page 23 of 23

- Customer Name: Bob
- Address: Bob's Place
- Category: 3

**Was successfully inserted.**

[See all Customers.](#)

Screenshot for Query 13

Customer	category
Bob	3
Bob Hope	3
Daniel Kit	5
Darth Vader	3
Kat Kit	4
Mike Shephard	3
Mike Tyson	4
Troy Aikman	2