A JOB-SHOP ACCOUNTING SYSTEM

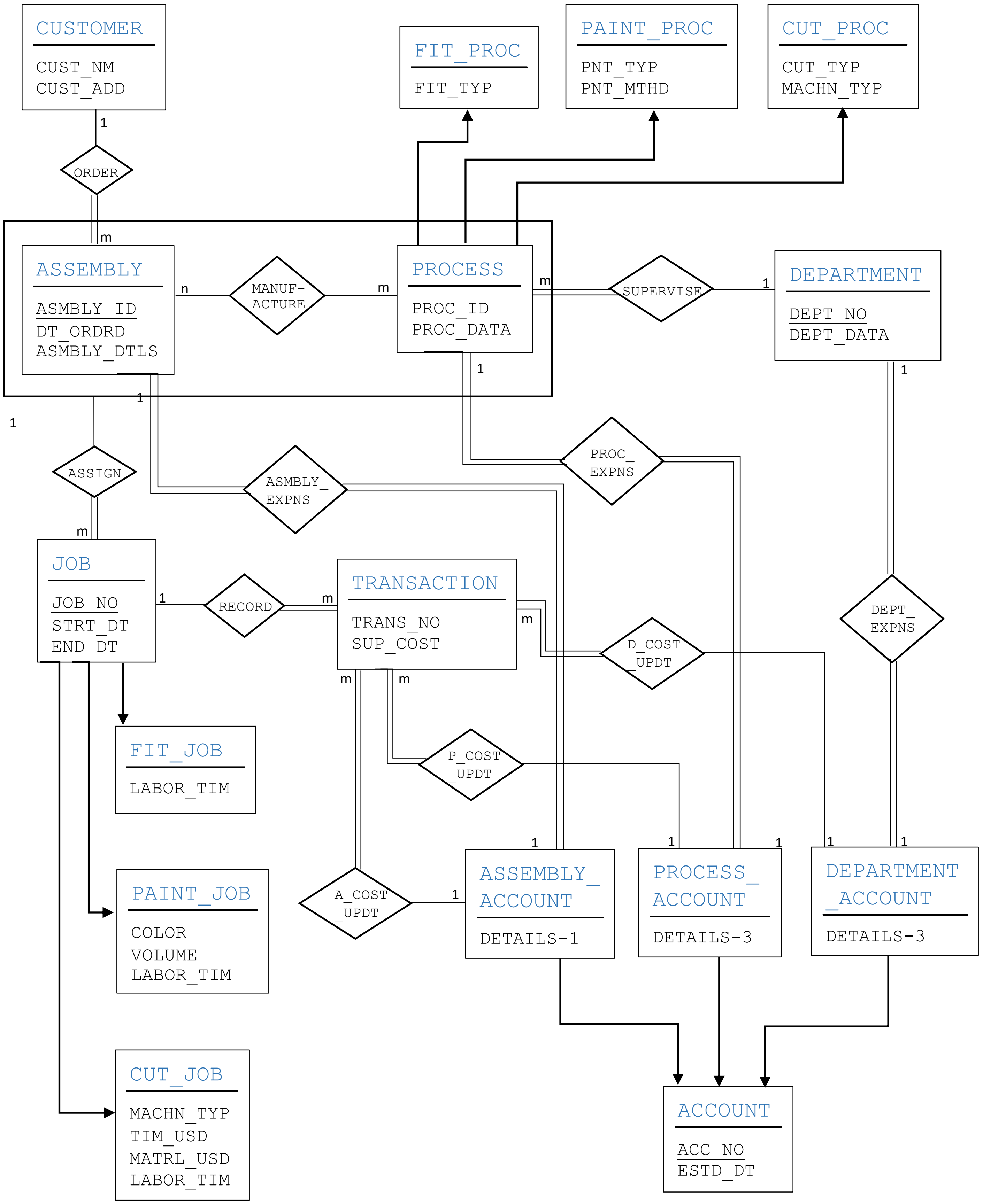
CS 4513
Database Management System
Sec 001
Fall 2015
Course Instructor: Dr. Le Gruenwald

Submitted by:
Akshay Gaur
113294004
akshaygaur@ou.edu

Contents

TASK 1

1.1 ER Diagram


CUSTOMER
CUST_NM
CUST_ADD
FIT_PROC
FIT_TYP
PAINT_PROC
PNT_TYP
PNT_MTHD
CUT_PROC
CUT_TYP
MACHN_TYP
ORDER
1
m
ASSEMBLY
ASMBLY_ID
DT_ORDRD
ASMBLY_DTLS
n
MANUF-ACTURE
m
PROCESS
PROC_ID
PROC_DATA
m
SUPERVISE
1
DEPARTMENT
DEPT_NO
DEPT_DATA
1
1
1
1
ASSIGN
ASMBLY_EXPNS
PROC_EXPNS
m
JOB
JOB_NO
STRT_DT
END_DT
1
RECORD
m
TRANSACTION
TRANS_NO
SUP_COST
m
D_COST_UPDT
DEPT_EXPNS
m
m
FIT_JOB
LABOR_TIM
P_COST_UPDT
PAINT_JOB
COLOR
VOLUME
LABOR_TIM
A_COST_UPDT
1
1
1
1
1
1
ASSEMBLY_ACCOUNT
DETAILS-1
PROCESS_ACCOUNT
DETAILS-3
DEPARTMENT_ACCOUNT
DETAILS-3
CUT_JOB
MACHN_TYP
TIM_USD
MATRL_USD
LABOR_TIM
ACCOUNT
ACC_NO
ESTD_DT

1.2 RELATION DATABASE SCHEMA

1) cust (cust_nm, cust_add)

2) asmbly (asmbly_id, ordr_dt, asmbly_dtls, cust_nm)

3) proc (proc_id, proc_data, dept_no)

4) fit_proc (proc_id, fit_typ)

5) pnt_proc (proc_id, pnt_typ, pnt_mthd)

6) cut_ proc (proc_id, cut_typ, mchn_typ)

7) dept (dept_no, dept_data)

8) job (job_no, strt_dt, end_dt, lbr_tim)

9) fit_job (job_no)

10) pnt_job (job_no, color, vol)

11) cut_job (job_no, mchn_typ, tim_usd, mtrl_usd)

12) trans (trans_id, sup_cost, job_no)

13) accnt (accnt_no, estd_dt)

14) asmbly_accnt (accnt_no, details-1, asmbly_id)

15) proc_accnt (accnt_no, details-3, proc_id)

16) dept_accnt (accnt_no, details-2, dept_no)

17) job_assgn (job_no, asmbly_id, proc_id)

18) accnt_trans (trans_id , accnt_no)

19) *NOTES: A table named "order" (to map customer to assembly ordered) is not required here because the customer name assoiated with any assembly order has been incorporated in the "asmbly" table itself. Similarly, "supervise" (to map process to supervising department) and "transaction record" (to map each transaction to a job) are not required as these relations have been incorporated in "proc" and "trans" tables itself. Since, asmbly_id, proc_id and dept_no are incorporated in asmbly_acnt, proc_accnt and dept_accnt respectively, separate tables to store the mapping of assembly, process and department with their accounts is not required.*
20) *Fit_job table is not created separately as the only job information is labor time which has been incorporated in the job table.*
21) *asmbly_mnfctr (to map assemblies and processes it used) is not created because the same information is incorporated in job_assgn table.*

TASK 2 DATA DICTIONARY

Using Oracle SQL Developer, the following was the Data Element Dictionary for my schema:

Task2_Data_Elemen
t_Dictonary.pdf

TASK 3

3.1 Discussion of storage structures for tables

Below is my analysis of the queries and the tables that are affected by the tables.

Query	Frequency	Type	Table Affected
Enter a new customer (30/day).	30/day	Insert	cust
Enter a new department (infrequent).		Insert	dept
Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered (40/day).	40/day	Insert	asmbly
Enter a new process-id and its department together with its type and information relevant to the type (infrequent).		Insert	proc, fit_proc, pnt_proc, cut_proc
Create a new account and associate it with the process, assembly, or department to which it is applicable (10/day).	10/day	Insert	accnt, asmbly_accnt, proc_accnt, dept_accnt
Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced (50/day).	50/day	Insert	job, job_assgn, fit_proc, pnt_proc, cut_proc, fit_job, pnt_job, cut_job
At the completion of a job, enter the date it completed and the information relevant to the type of job (50/day).	50/day	Update	job, fit_job, pnt_job, cut_job
Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details (50/day).	50/day	Insert	trans, job_assgn, proc, asmbly_accnt, proc_accnt, dept_accnt, accnt_trans
Retrieve the cost incurred on an assembly-id (200/day).	200/day	Select	asmbly_accnt
Retrieve the labor time recorded on an assembly-id (100/day).	100/day	Select	job_assgn, job
Retrieve the total labor time within a department for jobs completed in the department during a given date (20/day).	20/day	Select	proc, job_assgn, job
Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process (100/day).	100/day	Select	job_assgn, proc, job
Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department (20/day).	20/day	Select	proc, job_assgn, fit_job, pnt_job, cut_job
Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method (50/day).	50/day	Select	pnt_job, pnt_proc, job_assgn, asmbly
Delete all cut-jobs whose job-no is in some range (1/month).	1/month	Delete	cut_job, job
Change the color of a given paint job (1/week).	1/week	Update	pnt_job
Retrieve the average cost of all accounts (5/day).	5/day	Select	asmbly_accnt, proc_accnt, dept_accnt
NOTE:			
Heap-Organized table in oracle has no specific organization to follow while insertion and is thus good for frequent insertion.			
Index-Organized table is variant of B-Tree index in storage organization and is thus good for retrieval.			

Drilling down on the tables affected, below gives more details on what attribute is affected by which type of query.

Table	Type	Frequency	Column	Type	Frequency	Column
accnt	Insert	10/Day	accnt_no			
asmbly	Insert	40/Day	asmbly_id	Select	50/Day	asmbly_id
asmbly_accnt	Insert	4/Day	accnt_no	Update	50/Day	asmbly_id
	Select	200/Day	asmbly_id	Select	5/Day	NA (Select ALL)
cust	Insert	30/Day	cust_id			
cut_job	Insert	17/Day	job_no	Update	17/Day	job_no
	Select	20/Day	job_no	Delete	1/Month	job_no
cut_proc	Insert	Infrequent	proc_id	Select	50/Day	proc_id
dept	Insert	Infrequent	dept_no			
dept_accnt	Insert	4/Day	accnt_no	Update	50/Day	dept_no
	Select	5/Day	NA (Select ALL)			
fit_job	Insert	17/Day	job_no	Select	20/Day	job_no
fit_proc	Insert	Infrequent	proc_id	Select	50/Day	proc_id
job	Insert	50/Day	job_no	Update	50/Day	job_no
	Select	100/Day	job_no	Select	20/Day	job_no, end_dt
	Select	100/Day	job_no	Delete	1/Month	job_no
job_assgn	Insert	50/Day	job_no	Select	50/Day	job_no
	Select	100/Day	asmbly_id	Select	20/Day	proc_id
	Select	100/Day	asmbly_id	Select	20/Day	proc_id
	Select	50/Day	job_no, proc_id			
pnt_job	Insert	17/Day	job_no	Update	17/Day	job_no
	Select	20/Day	job_no	Select	50/Day	color
	Update	1/Week	job_no			
pnt_proc	Insert	Infrequent	proc_id	Select	50/Day	proc_id
	Select	50/Day	pnt_mthd			
proc	Insert	Infrequent	proc_id	Select	50/Day	proc_id
	Select	20/Day	dept_no	Select	100/Day	proc_id
	Select	20/Day	dept_no			
proc_accnt	Insert	4/Day	accnt_no	Update	50/Day	proc_id
	Select	5/Day	NA (Select ALL)			
trans	Insert	50/Day	trans_id			
accnt_trans	Insert	50/Day	accnt_no			

3.2 Discussion of storage structures for tables (Oracle 12c)

Based on the queries that were being implemented on the tables, if there was large number of inserts on less retrievals, HEAP ORGANIZATION was chosen to aid in faster insertion.

If the retrieval queries were in large amount, then Index Organization was chosen to aid in faster access.

Table	Appropriate Table Organization	Applicable Table Organization
cust	Heap Organization	Heap Organization
asmbly	B^+-Tree Organization	Index Organized Organization
dept	Heap Organization	Heap Organization
proc	B^+-Tree Organization	Index Organized Organization
fit_proc	B^+-Tree Organization	Index Organized Organization
cut_proc	B^+-Tree Organization	Index Organized Organization
pnt_proc	B^+-Tree Organization	Index Organized Organization
jobs	B^+-Tree Organization	Index Organized Organization
fit_job	B^+-Tree Organization	Index Organized Organization
cut_job	B^+-Tree Organization	Index Organized Organization
pnt_job	B^+-Tree Organization	Index Organized Organization
trans	Heap Organization	Heap Organization
accnt	Heap Organization	Heap Organization
asmbly_accnt	B^+-Tree Organization	Index Organized Organization
dept_accnt	B^+-Tree Organization	Index Organized Organization
proc_accnt	B^+-Tree Organization	Index Organized Organization
job_assgn	B^+-Tree Organization	Index Organized Organization
accnt_trans	Heap Organization	Heap Organization

Based on what attributes are affected by what type of query, indexes were built on those attributes.

Table	Primary Key	Constraint	ColumnName	Constraint	ColumnName	Constraint	ColumnName
cust	cust_nm	NA	NA				
asmbly	asmbly_id	Index	asmbly_id				
dept	dept_no	NA	NA				
proc	proc_id	Index	proc_id	Index	dept_no		
fit_proc	proc_id	Index	proc_id				
cut_proc	proc_id	Index	proc_id				
pnt_proc	proc_id	Index	proc_id	Index	pnt_mthd		
jobs	job_no	Index	job_no	Index	end_dt		
fit_job	job_no	Index	job_no				
cut_job	job_no	Index	job_no				
pnt_job	job_no	Index	job_no	Index	color		
trans	trans_id	NA	NA				
accnt	accnt_no	NA	NA				
asmbly_accnt	accnt_no, asmbly_id	Index	asmbly_id				

dept_accnt	accnt_no, dept_no	Index	dept_no				
proc_accnt	accnt_no, proc_id	Index	proc_id				
job_assgn	job_no	Index	job_no	Index	proc_id	Index	asmbly_id
accnt_trans	trans_id	NA	NA				

TASK 4 SQL STATEMENTS AND SCREENSHOTS SHOWING CREATION OF TABLES

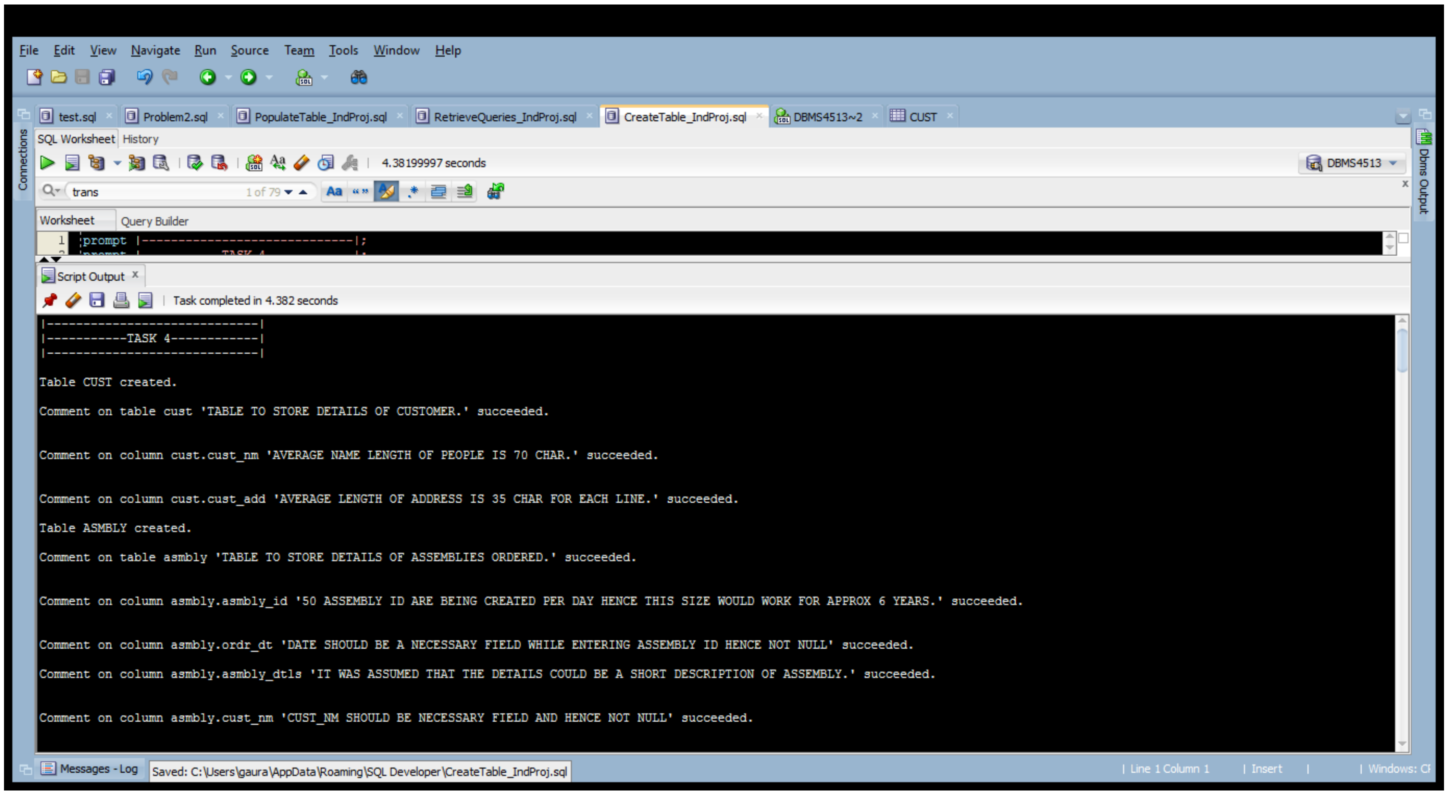
SQL Statements

```
prompt |-----------------------------|;
prompt |-----------TASK 4------------|;
prompt |-----------------------------|;

--Creating table "cust" as Heap Organized.
CREATE TABLE cust
  ( cust_nm VARCHAR2(80 CHAR)
  , cust_add VARCHAR2(128 CHAR)
  , CONSTRAINT pk_cust PRIMARY KEY (cust_nm)
  )
  ORGANIZATION HEAP;

  --COMMENTS
  COMMENT ON TABLE cust IS 'Table to store details of customer.';
  COMMENT ON COLUMN cust.cust_nm IS 'Average name length of people is 70 CHAR.';
  COMMENT ON COLUMN cust.cust_add IS 'Average length of address is 35 CHAR for each
line.';

--Creating table "asmbly" as Index Organized.
CREATE TABLE asmbly
  ( asmbly_id NUMBER(6)
  , ordr_dt DATE NOT NULL
  , asmbly_dtls VARCHAR2(128 CHAR)
  , cust_nm VARCHAR2(80 CHAR) NOT NULL
  , CONSTRAINT pk_asmbly
      PRIMARY KEY (asmbly_id)
  , CONSTRAINT fk_asmbly
      FOREIGN KEY (cust_nm)
      REFERENCES cust(cust_nm)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE asmbly IS 'Table to store details of assemblies ordered.';
  COMMENT ON COLUMN asmbly.asmbly_id IS '50 Assembly ID are being created per day hence
this size would work for approx 6 years.';
  COMMENT ON COLUMN asmbly.ordr_dt IS 'Date should be a necessary field while entering
assembly id hence NOT NULL';
  COMMENT ON COLUMN asmbly.asmbly_dtls IS 'It was assumed that the details could be a
short description of assembly.';
  COMMENT ON COLUMN asmbly.cust_nm IS 'cust_nm should be necessary field and hence NOT
NULL';

--Creating table "dept" as Heap Organized.
CREATE TABLE dept
  ( dept_no NUMBER(4)
  , dept_data VARCHAR2(128 CHAR)
  , CONSTRAINT pk_dept PRIMARY KEY (dept_no)
  )
  ORGANIZATION HEAP;

  --COMMENTS
  COMMENT ON TABLE dept IS 'Table is to store details of department.';
  COMMENT ON COLUMN dept.dept_no IS 'Departments are added infrequently, hence the size
has been kept to 4 digits.';
  COMMENT ON COLUMN dept.dept_data IS 'Assumed that it would be a short description of
the deparment.';

--Creating table "proc" as Index Organized.
--index on dept_no is created separately.
CREATE TABLE proc
  ( proc_id NUMBER(4)
  , proc_data VARCHAR2(128 CHAR)
  , dept_no NUMBER(4) NOT NULL
  , CONSTRAINT pk_proc
      PRIMARY KEY (proc_id)
  , CONSTRAINT fk_proc
      FOREIGN KEY (dept_no)
      REFERENCES dept(dept_no)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE proc IS 'Table to store process details.';
  COMMENT ON COLUMN proc.proc_id IS 'Since number of process ids entered is infrequent,
size has been kept to 4 digits';
  COMMENT ON COLUMN proc.proc_data IS 'Assumed brief description of process.';
  COMMENT ON COLUMN proc.dept_no IS 'The supervising department number is stored in this
column. Since each process has supervising department, hence not null.';

  --INDEXES
  CREATE INDEX ix_proc_dept_no ON proc (dept_no);

--Creating table "fit_proc" as Index Organized.
CREATE TABLE fit_proc
  ( proc_id NUMBER(4)
  , fit_typ VARCHAR2(16 CHAR)
  , CONSTRAINT pk_fit_proc
      PRIMARY KEY (proc_id)
  , CONSTRAINT fk_fit_proc
      FOREIGN KEY (proc_id)
      REFERENCES proc(proc_id)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE fit_proc IS 'Table to store the process of type fit.';
  COMMENT ON COLUMN fit_proc.proc_id IS 'Process id which references the proc_id from
proc table.';
  COMMENT ON COLUMN fit_proc.fit_typ IS 'Assumed type would be stored so would take less
characters than description.';

--Creating table "pnt_proc" as Index Organized.
CREATE TABLE pnt_proc
  ( proc_id NUMBER(4)
  , pnt_typ VARCHAR2(16 CHAR)
  , pnt_mthd VARCHAR2(16 CHAR)
  , CONSTRAINT pk_pnt_proc PRIMARY KEY (proc_id)
  , CONSTRAINT fk_pnt_proc FOREIGN KEY (proc_id) REFERENCES proc(proc_id)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE pnt_proc IS 'Table to store paint process.';
  COMMENT ON COLUMN pnt_proc.proc_id IS 'Process id which references the proc_id from
proc table.';
  COMMENT ON COLUMN pnt_proc.pnt_typ IS 'To store paint type.';
  COMMENT ON COLUMN pnt_proc.pnt_mthd IS 'Column to store paint method.';

  --INDEXES
  CREATE INDEX ix_pnt_proc_pnt_mthd ON pnt_proc (pnt_mthd);

--Creating table "cut_proc" as Index Organized.
CREATE TABLE cut_proc
  ( proc_id NUMBER(4)
  , cut_typ VARCHAR2(16 CHAR)
  , mchn_typ VARCHAR2(16 CHAR)
  , CONSTRAINT pk_cut_proc
      PRIMARY KEY (proc_id)
  , CONSTRAINT fk_cut_proc
      FOREIGN KEY (proc_id)
      REFERENCES proc(proc_id)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE CUT_proc IS 'Table to store cut process.';
  COMMENT ON COLUMN cut_proc.proc_id IS 'Process id which references the proc_id from
proc table.';
  COMMENT ON COLUMN cut_proc.cut_typ IS 'To store cut type.';
  COMMENT ON COLUMN cut_proc.mchn_typ IS 'Column to store machine type.';

--Creating table "jobs" as Index Organized
CREATE TABLE jobs
  ( job_no NUMBER(7)
  , strt_dt DATE NOT NULL
  , end_dt DATE
  , lbr_tim INTERVAL DAY TO SECOND
  , CONSTRAINT pk_jobs PRIMARY KEY (job_no)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE jobs IS 'Table to store job no and other details.';
  COMMENT ON COLUMN jobs.job_no IS 'Jobs are entered at the rate of 50/day. Keeping it 6
digits would be good enough for less than 5 years. Hence 7 digits.';
  COMMENT ON COLUMN jobs.strt_dt IS 'Start date of the job.';
  COMMENT ON COLUMN jobs.end_dt IS 'End date of the job.';
  COMMENT ON COLUMN jobs.lbr_tim IS 'Time that took to complete the job. Assumbed to be
in hours.';

  --INDEXES
  CREATE INDEX ix_jobs_end_dt ON jobs (end_dt);

--Creating table 'fit_job' as Index Organized.
CREATE TABLE fit_job
  ( job_no NUMBER(7)
  , CONSTRAINT pk_fit_job
      PRIMARY KEY (job_no)
  , CONSTRAINT fk_fit_job
      FOREIGN KEY (job_no)
```

```
       REFERENCES jobs(job_no)
   )
   ORGANIZATION INDEX;

   --COMMENTS
   COMMENT ON TABLE fit_job IS 'Table to store fit jobs.';
   COMMENT ON COLUMN fit_job.job_no IS 'This column references job_no in the jobs table.';

--Creating table 'cut_job' as Index Organized.
CREATE TABLE cut_job
   ( job_no NUMBER(7)
   , mchn_typ VARCHAR2(16 CHAR)
   , tim_usd INTERVAL DAY TO SECOND
   , mtrl_usd VARCHAR2(16 CHAR)
   , CONSTRAINT pk_cut_job
       PRIMARY KEY (job_no)
   , CONSTRAINT fk_cut_job
       FOREIGN KEY (job_no)
       REFERENCES jobs(job_no)
   )
   ORGANIZATION INDEX;

   --COMMENTS
   COMMENT ON TABLE cut_job IS 'Table to store cut jobs.';
   COMMENT ON COLUMN cut_job.job_no IS 'This column references job_no in the jobs table.';
   COMMENT ON COLUMN cut_job.mchn_typ IS 'This column stores the machine type used in the
job. Assumed to be abbreviation.';
   COMMENT ON COLUMN cut_job.tim_usd IS 'Duration of time. Assumed to be in hours.';
   COMMENT ON COLUMN cut_job.mtrl_usd IS 'Stores the material used in the cut job.';

--Creating table 'pnt_job' as Index Organized.
CREATE TABLE pnt_job
   ( job_no NUMBER(7)
   , color VARCHAR2(16 CHAR)
   , vol NUMBER(3,2)
   , CONSTRAINT pk_pnt_job
       PRIMARY KEY (job_no)
   , CONSTRAINT fk_pnt_job
       FOREIGN KEY (job_no)
       REFERENCES jobs(job_no)
   )
   ORGANIZATION INDEX;

   --COMMENTS
   COMMENT ON TABLE pnt_job IS 'Table to store paint jobs';
   COMMENT ON COLUMN pnt_job.job_no IS 'The job number stored in this column references
the job number in the jobs table.';
   COMMENT ON COLUMN pnt_job.color IS 'To store the color of the paint job.';
   COMMENT ON COLUMN pnt_job.vol IS 'To store the volume of the paint used.';

   --INDEX
   CREATE INDEX ix_pnt_job_color ON pnt_job (color);

--Creating table "trans" as Heap Organized
CREATE TABLE trans
   ( trans_id NUMBER(7)
   , sup_cost NUMBER(7,2) NOT NULL
   , job_no NUMBER(7) NOT NULL
   , CONSTRAINT pk_trans
       PRIMARY KEY (trans_id)
   , CONSTRAINT fk_trans
       FOREIGN KEY (job_no)
```

```
      REFERENCES jobs(job_no)
  )
  ORGANIZATION HEAP;
  --COMMENTS
  COMMENT ON TABLE trans IS 'Table to record the transactions against a job.';
  COMMENT ON COLUMN trans.trans_id IS 'This column stores the transaction. Since 50
transactions are created each day, 7 digits seem appropriate.';
  COMMENT ON COLUMN trans.sup_cost IS 'Stores cost of the transaction. Assumed that the
cost of any transaction would not be more than 5 figures.';
  COMMENT ON COLUMN trans.job_no IS 'The job against which the transaction is being
recorded. This references job_no in jobs table.';

--Creating table "accnt" as Heap Organized.
CREATE TABLE accnt
  ( accnt_no NUMBER(6)
  , estd_dt DATE NOT NULL
  , CONSTRAINT pk_accnt
      PRIMARY KEY (accnt_no)
  )
  ORGANIZATION HEAP;

  --COMMENTS
  COMMENT ON TABLE accnt IS 'This table stores account numbers and their established
date.';
  COMMENT ON COLUMN accnt.accnt_no IS 'Column to store account number. 10 accounts are
created per day 5 digits would not even last 3 years.';
  COMMENT ON COLUMN accnt.estd_dt IS 'Established date of the account. Should be not null
as this is an important attibute.';

--Creating table "asmbly_accnt" as Index Organized
CREATE TABLE asmbly_accnt
  ( accnt_no NUMBER(6)
  , details1 NUMBER(8,2)
  , asmbly_id NUMBER(6) NOT NULL
  , CONSTRAINT pk_asmbly_accnt
      PRIMARY KEY (asmbly_id, accnt_no)
  , CONSTRAINT fk_asmbly_accnt_asmbly_id
      FOREIGN KEY (asmbly_id)
      REFERENCES asmbly(asmbly_id)
  , CONSTRAINT fk_asmbly_accnt_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE asmbly_accnt IS 'Table to store the account, expenditure on the
account and the assembly it is associated to.';
  COMMENT ON COLUMN asmbly_accnt.accnt_no IS 'Stores account number which references
accnt table.';
  COMMENT ON COLUMN asmbly_accnt.details1 IS 'This column stores the sum of cost of all
the transactions against the relevant assembly id.';
  COMMENT ON COLUMN asmbly_accnt.asmbly_id IS 'This column stores the asmbly_id that the
account is associated with.';

  --INDEX
  CREATE INDEX ix_asmbly_accnt_asmbly_id ON asmbly_accnt (asmbly_id);

--Creating table "proc_accnt" as Index Organized
CREATE TABLE proc_accnt
  ( accnt_no NUMBER(6)
  , details3 NUMBER(8,2)
  , proc_id NUMBER(4) NOT NULL
  , CONSTRAINT pk_proc_accnt
      PRIMARY KEY (proc_id, accnt_no)
  , CONSTRAINT fk_proc_accnt_proc_id
      FOREIGN KEY (proc_id)
      REFERENCES proc(proc_id)
  , CONSTRAINT fk_proc_accnt_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE proc_accnt IS 'Table to store the account, expenditure on the account
and the process it is associated to.';
  COMMENT ON COLUMN proc_accnt.accnt_no IS 'Stores account number which references accnt
table.';
  COMMENT ON COLUMN proc_accnt.details3 IS 'This column stores the sum of cost of all the
transactions against the relevant process id.';
  COMMENT ON COLUMN proc_accnt.proc_id IS 'This column stores the proc_id that the
account is associated with.';

  --INDEX
  CREATE INDEX ix_proc_accnt_proc_id ON proc_accnt (proc_id);

--Creating table "dept_accnt" as Index Organized
CREATE TABLE dept_accnt
  ( accnt_no NUMBER(6)
  , details2 NUMBER(8, 2)
  , dept_no NUMBER(4) NOT NULL
  , CONSTRAINT pk_dept_accnt
      PRIMARY KEY (dept_no, accnt_no)
  , CONSTRAINT fk_dept_accnt_dept_no
      FOREIGN KEY (dept_no)
      REFERENCES dept(dept_no)
  , CONSTRAINT fk_dept_accnt_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE dept_accnt IS 'Table to store the account, expenditure on the account
and the department it is associated to.';
  COMMENT ON COLUMN dept_accnt.accnt_no IS 'Stores account number which references accnt
table.';
  COMMENT ON COLUMN dept_accnt.details2 IS 'This column stores the sum of cost of all the
transactions against the relevant department number.';
  COMMENT ON COLUMN dept_accnt.dept_no IS 'This column stores the dept_no that the
account is associated with.';

  --INDEX
  CREATE INDEX ix_dept_accnt_dept_no ON dept_accnt (dept_no);

--Creating table "a_accnt_trans" as Heap Organized.
CREATE TABLE a_accnt_trans
  ( trans_id NUMBER(7)
  , accnt_no NUMBER(6) NOT NULL
  , CONSTRAINT pk_a_accnt_trans
      PRIMARY KEY (trans_id)
  , CONSTRAINT fk_a_accnt_trans_trans_id
      FOREIGN KEY (trans_id)
      REFERENCES trans(trans_id)
  , CONSTRAINT fk_a_accnt_trans_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION HEAP;
  --COMMENTS
  COMMENT ON TABLE a_accnt_trans IS 'This table stores all the transactions against
assembly accounts.';
  COMMENT ON COLUMN a_accnt_trans.trans_id IS 'This is the transaction id and references
table trans.';
  COMMENT ON COLUMN a_accnt_trans.accnt_no IS 'This column stores the account against
which the transaction is applicable. accnt_no references asbmly_accnt table.';

--Creating table "p_accnt_trans" as Heap Organized.
CREATE TABLE p_accnt_trans
  ( trans_id NUMBER(7)
  , accnt_no NUMBER(6) NOT NULL
  , CONSTRAINT pk_p_accnt_trans
      PRIMARY KEY (trans_id)
  , CONSTRAINT fk_p_accnt_trans_trans_id
      FOREIGN KEY (trans_id)
      REFERENCES trans(trans_id)
  , CONSTRAINT fk_p_accnt_trans_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION HEAP;
  --COMMENTS
  COMMENT ON TABLE p_accnt_trans IS 'This table stores all the transactions against
process accounts.';
  COMMENT ON COLUMN p_accnt_trans.trans_id IS 'This is the transaction id and references
table trans.';
  COMMENT ON COLUMN p_accnt_trans.accnt_no IS 'This column stores the account against
which the transaction is applicable. accnt_no references asbmly_accnt table.';

--Creating table "d_accnt_trans" as Heap Organized.
CREATE TABLE d_accnt_trans
  ( trans_id NUMBER(7)
  , accnt_no NUMBER(6) NOT NULL
  , CONSTRAINT pk_d_accnt_trans
      PRIMARY KEY (trans_id)
  , CONSTRAINT fk_d_accnt_trans_trans_id
      FOREIGN KEY (trans_id)
      REFERENCES trans(trans_id)
  , CONSTRAINT fk_d_accnt_trans_accnt_no
      FOREIGN KEY (accnt_no)
      REFERENCES accnt(accnt_no)
  )
  ORGANIZATION HEAP;
  --COMMENTS
  COMMENT ON TABLE d_accnt_trans IS 'This table stores all the transactions against
department accounts.';
  COMMENT ON COLUMN d_accnt_trans.trans_id IS 'This is the transaction id and references
table trans.';
  COMMENT ON COLUMN d_accnt_trans.accnt_no IS 'This column stores the account against
which the transaction is applicable. accnt_no references asbmly_accnt table.';

--Creating table "job_assgn" as Index Organized
CREATE TABLE job_assgn
  ( job_no NUMBER(7)
  , asmbly_id NUMBER(6) NOT NULL
  , proc_id NUMBER(4) NOT NULL
  , CONSTRAINT pk_job_assgn
      PRIMARY KEY (job_no)
  , CONSTRAINT fk_job_assgn_job_no
      FOREIGN KEY (job_no)
      REFERENCES jobs(job_no)
  , CONSTRAINT fk_job_assgn_asmbly_id
      FOREIGN KEY (asmbly_id)
      REFERENCES asmbly(asmbly_id)
  , CONSTRAINT fk_job_assgn_proc_id
      FOREIGN KEY (proc_id)
      REFERENCES proc(proc_id)
  )
  ORGANIZATION INDEX;

  --COMMENTS
  COMMENT ON TABLE job_assgn IS 'This tables stores the mapping of the job number,
assembly id and the process id.';
  COMMENT ON COLUMN job_assgn.job_no IS 'This is the job number assigned when process
assigned to any assembly starts.';
  COMMENT ON COLUMN job_assgn.asmbly_id IS 'The assembly on which the job is being
executed.';
  COMMENT ON COLUMN job_assgn.proc_id IS 'This is the process id that is assigned to the
assembly.';

  --INDEXES
  CREATE INDEX ix_job_assgn_asmbly_id ON job_assgn (asmbly_id);
  CREATE INDEX ix_job_assgn_proc_id ON job_assgn (proc_id);
```

Screenshot in Oracle SQL Developer of successful creation of tables.



```
prompt |----------------------------|;
```

```
|----------------------------|
|-----------TASK 4-----------|
|----------------------------|

Table CUST created.

Comment on table cust 'TABLE TO STORE DETAILS OF CUSTOMER.' succeeded.


Comment on column cust.cust_nm 'AVERAGE NAME LENGTH OF PEOPLE IS 70 CHAR.' succeeded.


Comment on column cust.cust_add 'AVERAGE LENGTH OF ADDRESS IS 35 CHAR FOR EACH LINE.' succeeded.

Table ASMBLY created.

Comment on table asmbly 'TABLE TO STORE DETAILS OF ASSEMBLIES ORDERED.' succeeded.


Comment on column asmbly.asmbly_id '50 ASSEMBLY ID ARE BEING CREATED PER DAY HENCE THIS SIZE WOULD WORK FOR APPROX 6 YEARS.' succeeded.


Comment on column asmbly.ordr_dt 'DATE SHOULD BE A NECESSARY FIELD WHILE ENTERING ASSEMBLY ID HENCE NOT NULL' succeeded.

Comment on column asmbly.asmbly_dtls 'IT WAS ASSUMED THAT THE DETAILS COULD BE A SHORT DESCRIPTION OF ASSEMBLY.' succeeded.


Comment on column asmbly.cust_nm 'CUST_NM SHOULD BE NECESSARY FIELD AND HENCE NOT NULL' succeeded.
```

Saved: C:\Users\gaura\AppData\Roaming\SQL Developer\CreateTable_IndProj.sql

Complete Output from the table creation queries.

```
|-----------------------------|

|-----------TASK 4------------|

|-----------------------------|


Table CUST created.


Comment on table cust 'TABLE TO STORE DETAILS OF CUSTOMER.' succeeded.



Comment on column cust.cust_nm 'AVERAGE NAME LENGTH OF PEOPLE IS 70 CHAR.' succeeded.



Comment on column cust.cust_add 'AVERAGE LENGTH OF ADDRESS IS 35 CHAR FOR EACH LINE.' succeeded.


Table ASMBLY created.


Comment on table asmbly 'TABLE TO STORE DETAILS OF ASSEMBLIES ORDERED.' succeeded.



Comment on column asmbly.asmbly_id '50 ASSEMBLY ID ARE BEING CREATED PER DAY HENCE THIS SIZE WOULD WORK FOR
APPROX 6 YEARS.' succeeded.



Comment on column asmbly.ordr_dt 'DATE SHOULD BE A NECESSARY FIELD WHILE ENTERING ASSEMBLY ID HENCE NOT NULL'
succeeded.


Comment on column asmbly.asmbly_dtls 'IT WAS ASSUMED THAT THE DETAILS COULD BE A SHORT DESCRIPTION OF
ASSEMBLY.' succeeded.
```

```
Comment on column asmbly.cust_nm 'CUST_NM SHOULD BE NECESSARY FIELD AND HENCE NOT NULL' succeeded.


Table DEPT created.


Comment on table dept 'TABLE IS TO STORE DETAILS OF DEPARTMENT.' succeeded.


Comment on column dept.dept_no 'DEPARTMENTS ARE ADDED INFREQUENTLY, HENCE THE SIZE HAS BEEN KEPT TO 4 DIGITS.'
succeeded.


Comment on column dept.dept_data 'ASSUMED THAT IT WOULD BE A SHORT DESCRIPTION OF THE DEPARMENT.' succeeded.


Table PROC created.


Comment on table proc 'TABLE TO STORE PROCESS DETAILS.' succeeded.


Comment on column proc.proc_id 'SINCE NUMBER OF PROCESS IDS ENTERED IS INFREQUENT, SIZE HAS BEEN KEPT TO 4
DIGITS' succeeded.


Comment on column proc.proc_data 'ASSUMED BRIEF DESCRIPTION OF PROCESS.' succeeded.


Comment on column proc.dept_no 'THE SUPERVISING DEPARTMENT NUMBER IS STORED IN THIS COLUMN. SINCE EACH
PROCESS HAS SUPERVISING DEPARTMENT, HENCE NOT NULL.' succeeded.
```

```
Index IX_PROC_DEPT_NO created.



Table FIT_PROC created.


Comment on table fit_proc 'TABLE TO STORE THE PROCESS OF TYPE FIT.' succeeded.



Comment on column fit_proc.proc_id 'PROCESS ID WHICH REFERENCES THE PROC_ID FROM PROC TABLE.' succeeded.



Comment on column fit_proc.fit_typ 'ASSUMED TYPE WOULD BE STORED SO WOULD TAKE LESS CHARACTERS THAN
DESCRIPTION.' succeeded.


Table PNT_PROC created.



Comment on table pnt_proc 'TABLE TO STORE PAINT PROCESS.' succeeded.



Comment on column pnt_proc.proc_id 'PROCESS ID WHICH REFERENCES THE PROC_ID FROM PROC TABLE.' succeeded.



Comment on column pnt_proc.pnt_typ 'TO STORE PAINT TYPE.' succeeded.


Comment on column pnt_proc.pnt_mthd 'COLUMN TO STORE PAINT METHOD.' succeeded.



Index IX_PNT_PROC_PNT_MTHD created.
```

```
Table CUT_PROC created.


Comment on table cut_proc 'TABLE TO STORE CUT PROCESS.' succeeded.


Comment on column cut_proc.proc_id 'PROCESS ID WHICH REFERENCES THE PROC_ID FROM PROC TABLE.' succeeded.


Comment on column cut_proc.cut_typ 'TO STORE CUT TYPE.' succeeded.


Comment on column cut_proc.mchn_typ 'COLUMN TO STORE MACHINE TYPE.' succeeded.


Table JOBS created.


Comment on table jobs 'TABLE TO STORE JOB NO AND OTHER DETAILS.' succeeded.


Comment on column jobs.job_no 'JOBS ARE ENTERED AT THE RATE OF 50/DAY. KEEPING IT 6 DIGITS WOULD BE GOOD
ENOUGH FOR LESS THAN 5 YEARS. HENCE 7 DIGITS.' succeeded.


Comment on column jobs.strt_dt 'START DATE OF THE JOB.' succeeded.


Comment on column jobs.end_dt 'END DATE OF THE JOB.' succeeded.


Comment on column jobs.lbr_tim 'TIME THAT TOOK TO COMPLETE THE JOB. ASSUMBED TO BE IN HOURS.' succeeded.
```

```
Index IX_JOBS_END_DT created.



Table FIT_JOB created.



Comment on table fit_job 'TABLE TO STORE FIT JOBS.' succeeded.


Comment on column fit_job.job_no 'THIS COLUMN REFERENCES JOB_NO IN THE JOBS TABLE.' succeeded.



Table CUT_JOB created.



Comment on table cut_job 'TABLE TO STORE CUT JOBS.' succeeded.


Comment on column cut_job.job_no 'THIS COLUMN REFERENCES JOB_NO IN THE JOBS TABLE.' succeeded.



Comment on column cut_job.mchn_typ 'THIS COLUMN STORES THE MACHINE TYPE USED IN THE JOB. ASSUMED TO BE
ABBREVIATION.' succeeded.



Comment on column cut_job.tim_usd 'DURATION OF TIME. ASSUMED TO BE IN HOURS.' succeeded.


Comment on column cut_job.mtrl_usd 'STORES THE MATERIAL USED IN THE CUT JOB.' succeeded.



Table PNT_JOB created.
```

```
Comment on table pnt_job 'TABLE TO STORE PAINT JOBS' succeeded.


Comment on column pnt_job.job_no 'THE JOB NUMBER STORED IN THIS COLUMN REFERENCES THE JOB NUMBER IN THE
JOBS TABLE.' succeeded.


Comment on column pnt_job.color 'TO STORE THE COLOR OF THE PAINT JOB.' succeeded.



Comment on column pnt_job.vol 'TO STORE THE VOLUME OF THE PAINT USED.' succeeded.



Index IX_PNT_JOB_COLOR created.


Table TRANS created.



Comment on table trans 'TABLE TO RECORD THE TRANSACTIONS AGAINST A JOB.' succeeded.



Comment on column trans.trans_id 'THIS COLUMN STORES THE TRANSACTION. SINCE 50 TRANSACTIONS ARE CREATED EACH
DAY, 7 DIGITS SEEM APPROPRIATE.' succeeded.



Comment on column trans.sup_cost 'STORES COST OF THE TRANSACTION. ASSUMED THAT THE COST OF ANY TRANSACTION
WOULD NOT BE MORE THAN 5 FIGURES.' succeeded.


Comment on column trans.job_no 'THE JOB AGAINST WHICH THE TRANSACTION IS BEING RECORDED. THIS REFERENCES
JOB_NO IN JOBS TABLE.' succeeded.
```

```
Table ACCNT created.


Comment on table accnt 'THIS TABLE STORES ACCOUNT NUMBERS AND THEIR ESTABLISHED DATE.' succeeded.

Comment on column accnt.accnt_no 'COLUMN TO STORE ACCOUNT NUMBER. 10 ACCOUNTS ARE CREATED PER DAY 5 DIGITS
WOULD NOT EVEN LAST 3 YEARS.' succeeded.


Comment on column accnt.estd_dt 'ESTABLISHED DATE OF THE ACCOUNT. SHOULD BE NOT NULL AS THIS IS AN IMPORTANT
ATTIBUTE.' succeeded.


Table ASMBLY_ACCNT created.

Comment on table asmbly_accnt 'TABLE TO STORE THE ACCOUNT, EXPENDITURE ON THE ACCOUNT AND THE ASSEMBLY IT IS
ASSOCIATED TO.' succeeded.


Comment on column asmbly_accnt.accnt_no 'STORES ACCOUNT NUMBER WHICH REFERENCES ACCNT TABLE.' succeeded.


Comment on column asmbly_accnt.details1 'THIS COLUMN STORES THE SUM OF COST OF ALL THE TRANSACTIONS AGAINST
THE RELEVANT ASSEMBLY ID.' succeeded.


Comment on column asmbly_accnt.asmbly_id 'THIS COLUMN STORES THE ASMBLY_ID THAT THE ACCOUNT IS ASSOCIATED
WITH.' succeeded.

Index IX_ASMBLY_ACCNT_ASMBLY_ID created.
```

```
Table PROC_ACCNT created.


Comment on table proc_accnt 'TABLE TO STORE THE ACCOUNT, EXPENDITURE ON THE ACCOUNT AND THE PROCESS IT IS
ASSOCIATED TO.' succeeded.



Comment on column proc_accnt.accnt_no 'STORES ACCOUNT NUMBER WHICH REFERENCES ACCNT TABLE.' succeeded.



Comment on column proc_accnt.details3 'THIS COLUMN STORES THE SUM OF COST OF ALL THE TRANSACTIONS AGAINST THE
RELEVANT PROCESS ID.' succeeded.


Comment on column proc_accnt.proc_id 'THIS COLUMN STORES THE PROC_ID THAT THE ACCOUNT IS ASSOCIATED WITH.'
succeeded.



Index IX_PROC_ACCNT_PROC_ID created.



Table DEPT_ACCNT created.


Comment on table dept_accnt 'TABLE TO STORE THE ACCOUNT, EXPENDITURE ON THE ACCOUNT AND THE DEPARTMENT IT IS
ASSOCIATED TO.' succeeded.



Comment on column dept_accnt.accnt_no 'STORES ACCOUNT NUMBER WHICH REFERENCES ACCNT TABLE.' succeeded.
Comment on column dept_accnt.details2 'THIS COLUMN STORES THE SUM OF COST OF ALL THE TRANSACTIONS AGAINST THE
RELEVANT DEPARTMENT NUMBER.' succeeded.


Comment on column dept_accnt.dept_no 'THIS COLUMN STORES THE DEPT_NO THAT THE ACCOUNT IS ASSOCIATED WITH.'
succeeded.



Index IX_DEPT_ACCNT_DEPT_NO created.



Table A_ACCNT_TRANS created.


Comment on table a_accnt_trans 'THIS TABLE STORES ALL THE TRANSACTIONS AGAINST ASSEMBLY ACCOUNTS.' succeeded.



Comment on column a_accnt_trans.trans_id 'THIS IS THE TRANSACTION ID AND REFERENCES TABLE TRANS.' succeeded.



Comment on column a_accnt_trans.accnt_no 'THIS COLUMN STORES THE ACCOUNT AGAINST WHICH THE TRANSACTION IS
APPLICABLE. ACCNT_NO REFERENCES ASBMLY_ACCNT TABLE.' succeeded.


Table P_ACCNT_TRANS created.



Comment on table p_accnt_trans 'THIS TABLE STORES ALL THE TRANSACTIONS AGAINST PROCESS ACCOUNTS.' succeeded.



Comment on column p_accnt_trans.trans_id 'THIS IS THE TRANSACTION ID AND REFERENCES TABLE TRANS.' succeeded.


Comment on column p_accnt_trans.accnt_no 'THIS COLUMN STORES THE ACCOUNT AGAINST WHICH THE TRANSACTION IS
APPLICABLE. ACCNT_NO REFERENCES ASBMLY_ACCNT TABLE.' succeeded.
```

```
Table D_ACCNT_TRANS created.


Comment on table d_accnt_trans 'THIS TABLE STORES ALL THE TRANSACTIONS AGAINST DEPARTMENT ACCOUNTS.'
succeeded.


Comment on column d_accnt_trans.trans_id 'THIS IS THE TRANSACTION ID AND REFERENCES TABLE TRANS.' succeeded.


Comment on column d_accnt_trans.accnt_no 'THIS COLUMN STORES THE ACCOUNT AGAINST WHICH THE TRANSACTION IS
APPLICABLE. ACCNT_NO REFERENCES ASBMLY_ACCNT TABLE.' succeeded.


Table JOB_ASSGN created.


Comment on table job_assgn 'THIS TABLES STORES THE MAPPING OF THE JOB NUMBER, ASSEMBLY ID AND THE PROCESS ID.'
succeeded.


Comment on column job_assgn.job_no 'THIS IS THE JOB NUMBER ASSIGNED WHEN PROCESS ASSIGNED TO ANY ASSEMBLY
STARTS.' succeeded.


Comment on column job_assgn.asmbly_id 'THE ASSEMBLY ON WHICH THE JOB IS BEING EXECUTED.' succeeded.


Comment on column job_assgn.proc_id 'THIS IS THE PROCESS ID THAT IS ASSIGNED TO THE ASSEMBLY.' succeeded.


Index IX_JOB_ASSGN_ASMBLY_ID created.
Index IX_JOB_ASSGN_PROC_ID created.
```

TASK 5 The Java source program and screenshots showing its successful compilation

```
/*
CS 4513
Database Management System
Sec 001
Fall 2015
Course Instructor: Dr. Le Gruenwald
Submitted by:
               Akshay Gaur
               113294004
               akshaygaur@ou.edu


The following is java source code for Task 5 of Individual Project 1.

Code presents the user with 4 choices and takes action as per choice selected.
Choice 1:      Enter a new customer.
Choice 2:      Enter a new department.
Choice 3:      Enter a new assembly with its customer-name, assembly-details, assembly-
id, and date-ordered.
Choice 4:      Enter a new process-id and its department together with its type and
information relevant to the type.
Choice 5:      Create a new account and associate it with the process, assembly, or
department to which it is applicable.
Choice 6:      Enter a new job, given its job-no, assembly-id, process-id, and date the
job commenced.
Choice 7:      At the completion of a job, enter the date it completed and the
information relevant to the type of job.
Choice 8:      Enter a transaction-no, and its sup-cost and update all the costs
(details) of the affected accounts by adding sup-cost to their current values of details.
Choice 9:      Retrieve the cost incurred on an assembly-id.
Choice 10:     Retrieve the labor time recorded on an assembly-id.
Choice 11:     Retrieve the total labor time within a department for jobs completed in
the department during a given date.
Choice 12:     Retrieve the processes through which a given assembly-id has passed so far
(in date-commenced order) and the department responsible for each process.
Choice 13:     Retrieve the jobs (together with their type information and assembly-id)
completed during a given date in a given department.
Choice 14:     Retrieve the customers (in name order) whose assemblies are painted RED
using a given painting method.
Choice 15:     Delete all cut-jobs whose job-no is in some range.
Choice 16:     Change the color of a given paint job.
Choice 17:     Retrieve the average cost of all accounts.
Choice 18:     Import: enter new customers from a data file.
Choice 19:     Export: Retrieve the customers (in name order) whose assemblies are
painted RED using a given painting method and output them to a data file instead of
screen.
Choice 20:     Ends the program.
The program will terminate only when user selects option 20.
*/
```

```
import java.sql.*;
import java.util.*;
import java.io.Console;
import java.io.FileWriter;
import java.io.FileReader;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.File;
import java.util.Scanner;
import java.io.FileInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;


public class IP_JAVA_Gaur_Akshay {
    public static void main(String[] args) throws FileNotFoundException {
                // Declare variables.
                int loop=0;
                int choice=0;

                // Console reader.
                Console cnsl = null;

                // Declare connection set to null. We will create an object that will
                // exist until the program terminates so that oonnectionsto the DB
                // don't need to be made again and again.
                Connection conn = null;
                try {
                        // Load and register Oracle driver
                Class.forName("oracle.jdbc.driver.OracleDriver");
                // Establish connection that will persist for the duration of the program
until user exits.
                        // This way, the connection will not have to be made and closed
again and again.
                // connection string:
"jdbc:oracle:thin@hostname:port/servername","username","password"
                conn = DriverManager.getConnection(
                "jdbc:oracle:thin:@oracle.cs.ou.edu:1521/pdborcl.cs.ou.edu",
                        "gaur4004","IVgm3Um6");

                        //Statement stmt = conn.createStatement();

                        cnsl = System.console();

                        // Start while loop to present the user with options, user will
stay in while loop
                        // unless user chooses option 20.
                        while (loop==0) {
                                System.out.println();
                                System.out.println("                        WELCOME TO THE
JOB-SHOP ACCOUNTING DATABASE SYSTEM");
                                System.out.println();
                                System.out.println("    1.   Enter a new customer.");
                                System.out.println("    2.   Enter a new department.");
                                System.out.println("    3.   Enter a new assembly with its
customer-name, assembly-details, assembly-id, and date-ordered.");
                                System.out.println("    4.   Enter a new process-id and its
department together with its type and information relevant to the type.");
                                System.out.println("    5.   Create a new account and
associate it with the process, assembly, or department to which it is applicable.");
```

```
                              System.out.println("    6.   Enter a new job, given its
job-no, assembly-id, process-id, and date the job commenced.");
                              System.out.println("    7.   At the completion of a job,
enter the date it completed and the information relevant to the type of job.");
                              System.out.println("    8.   Enter a transaction-no, and
its sup-cost and update all the costs (details) of the affected accounts.");
                              System.out.println("    9.   Retrieve the cost incurred on
an assembly-id.");
                              System.out.println("    10.  Retrieve the labor time
recorded on an assembly-id.");
                              System.out.println("    11.  Retrieve the total labor time
within a department for jobs completed in the department during a given date.");
                              System.out.println("    12.  Retrieve the processes through
which a given assembly-id has passed so far (in date-commenced order) and");
                              System.out.println("         the department responsible for
each process.");
                              System.out.println("    13.  Retrieve the jobs (together
with their type information and assembly-id) completed during a given date in a given
department.");
                              System.out.println("    14.  Retrieve the customers (in
name order) whose assemblies are painted RED using a given painting method.");
                              System.out.println("    15.  Delete all cut-jobs whose job-
no is in some range.");
                              System.out.println("    16.  Change the color of a given
paint job.");
                              System.out.println("    17.  Retrieve the average cost of
all accounts.");
                              System.out.println("    18.  Import: enter new customers
from a data file.");
                              System.out.println("    19.  Export: Retrieve the customers
(in name order) whose assemblies are painted RED using a given painting method.");
                              System.out.println("    20.  Quit.");
                              // Read user input.
                              choice = Integer.parseInt(cnsl.readLine());
                              // Select action based on user input.
                              switch(choice) {
                                      case 1:
                                              // Declare variables.
                                              String custName1 = null;
                                              String custAdd1 = null;
                                              String custInsertQuery1 = null;

                                              // Present user with prompt for the data to
be entered
                                              // accept the same using Console Reader.
                                              custName1 = cnsl.readLine("Please enter the
customer name: ");
                                              custAdd1 = cnsl.readLine("Please enter the
customer address: ");

                                              // Create the query to be sent to Oracle.
                                              custInsertQuery1 = "INSERT INTO cust VALUES
('" + custName1
                                                      + "', '" + custAdd1 +"')";

                                              // Execute the query.

       conn.createStatement().executeQuery(custInsertQuery1);

                                              // Print Success message.
                                              System.out.println();
```

```
                                                System.out.println("Customer Successfully
Entered!!");

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 2:
                                                // Declare variables.
                                                int deptNo2 = 0;
                                                String deptData2 = null;
                                                String deptInsertQuery2 = null;

                                                // Present user prompt for the data to be
entered
                                                // accept the same using Console Reader.
                                                deptNo2 =
Integer.parseInt(cnsl.readLine("Please enter the department number: "));
                                                deptData2 = cnsl.readLine("Please enter the
department data: ");

                                                // Create the query to be sent to Oracle.
                                                deptInsertQuery2 = "INSERT INTO dept VALUES
(" + deptNo2 + ", "
                                                        + "'" + deptData2 + "')";

                                                // Execute the query.

        conn.createStatement().executeQuery(deptInsertQuery2);

                                                // Print the success message.
                                                System.out.println();
                                                System.out.println("Department Successfully
Entered!!");

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 3:
                                                // Declare variables.
                                                int asmblyID3 = 0;
                                                String ordrDt3 = null;
                                                String asmblyDet3 = null;
                                                String custName3 = null;
                                                String asmblyInsertQuery3 = null;

                                                // Present user prompt for the data to be
entered
                                                // accept the same using Console Reader.
                                                asmblyID3 =
Integer.parseInt(cnsl.readLine("Please enter the assembly number: "));
                                                ordrDt3 = cnsl.readLine("Please enter the
order date in MM/DD/YYYY format: ");
                                                asmblyDet3 = cnsl.readLine("Please enter the
assembly details: ");
                                                custName3 = cnsl.readLine("Please enter the
customer who entered this assembly: ");

                                                // Create the query to be sent to Oracle.
                                                asmblyInsertQuery3 = "INSERT INTO asmbly
VALUES (" + asmblyID3 + ", "
                                                        + "TO_DATE('" + ordrDt3 + "', 
'MM/DD/YYYY'), "
                                                        + "'" + asmblyDet3 + "', "
                                                        + "'" + custName3 + "')";

                                                // Execute the statement.
        
        conn.createStatement().executeQuery(asmblyInsertQuery3);

                                                // Print the success message.
                                                System.out.println();
                                                System.out.println("Assembly Successfully 
Entered!!");


                                                cnsl.readLine("Hit enter to 
continue......");

                                                break;
                                        case 4:
                                                // Declare variables.
                                                int procID4 = 0;
                                                String procData4 = null;
                                                String deptNo4 = null;
                                                String procType4 = null;
                                                String fitType4 = null;
                                                String paintType4 = null;
                                                String paintMthd4 = null;
                                                String cutType4 = null;
                                                String mchnType4 = 
null;
                                                String procInsertQuery4 = null;
                                                String procTypeInsertQuery4 = null;

                                                // Present user prompt for the data to be 
entered
                                                // accept the same using Console Reader.
                                                procID4 = 
Integer.parseInt(cnsl.readLine("Please enter the process number: "));
                                                procData4 = cnsl.readLine("Please enter the 
process data: ");
                                                deptNo4 = cnsl.readLine("Please enter the 
supervising department: ");


                                                // Create the query to be sent to Oracle.
                                                procInsertQuery4 = "INSERT INTO proc VALUES 
(" + procID4 + ", "
                                                        + "'" + procData4 + "', "
                                                        + deptNo4 + ")";

                                                // Execute the statement.
        
        conn.createStatement().executeQuery(procInsertQuery4);

                                                procType4 = cnsl.readLine("Please enter 
f/c/p if the process is fit proc/cut proc/paint proc respectively: ");

                                                if (procType4.equals("f")) {
                                                        fitType4 = cnsl.readLine("Please 
enter the fit type for the fit process: ");
                                                        procTypeInsertQuery4 = "INSERT INTO 
fit_proc VALUES (" + procID4 +", '" + fitType4 + "')";
```

```
        conn.createStatement().executeQuery(procTypeInsertQuery4);
                                        } else if (procType4.equals("c")) {
                                                cutType4 = cnsl.readLine("Please
enter the cut type for the cut process: ");
                                                mchnType4 = cnsl.readLine("Please
enter the machine type for the cut process: ");
                                                procTypeInsertQuery4 = "INSERT INTO
cut_proc VALUES (" + procID4 +", '" + cutType4 + "', '" + mchnType4 + "')";

        conn.createStatement().executeQuery(procTypeInsertQuery4);
                                        } else if (procType4.equals("p")) {
                                                paintType4 = cnsl.readLine("Please
enter the paint type for the paint process: ");
                                                paintMthd4 = cnsl.readLine("Please
enter the paint method for the paint process: ");
                                                procTypeInsertQuery4 = "INSERT INTO
pnt_proc VALUES (" + procID4 +", '" + paintType4 + "', '" + paintMthd4 + "')";

        conn.createStatement().executeQuery(procTypeInsertQuery4);
                                        }

                                        // Print the success message.
                                        System.out.println();
                                        System.out.println("Assembly Successfully
Entered!!");

                                        cnsl.readLine("Hit enter to
continue......");

                                        break;
                                case 5:
                                        // Declare variables.
                                        int accntNo5 = 0;
                                        String estdDt5 = null;
                                        String accntType5 = null;
                                        int details5 = 0;
                                        int id5 = 0;
                                        String accntInsertQuery5 = null;
                                        String accntTypeInsertQuery5 = null;

                                        // Present user prompt for the data to be
entered
                                        // accept the same using Console Reader.
                                        accntNo5 =
Integer.parseInt(cnsl.readLine("Please enter the accnt number: "));
                                        estdDt5 = cnsl.readLine("Please enter the
account established date in MM/DD/YYYY format: ");


                                        // Create the query to be sent to Oracle.
                                        accntInsertQuery5 = "INSERT INTO accnt
VALUES (" + accntNo5 + ", "
                                                + "TO_DATE('" + estdDt5 + "',
'MM/DD/YYYY'))";

                                        // Execute the statement.

        conn.createStatement().executeQuery(accntInsertQuery5);

                                        accntType5 = cnsl.readLine("Please enter
a/p/d if the account is assembly/process/department account respectively: ");
```

```
                                                    if (accntType5.equals("a")) {
                                                            id5 =
Integer.parseInt(cnsl.readLine("Please enter the assembly id for the assembly account:
"));
                                                            accntTypeInsertQuery5 = "INSERT INTO
asmbly_accnt VALUES (" + accntNo5 +", " + details5 + ", " + id5 + ")";

        conn.createStatement().executeQuery(accntTypeInsertQuery5);
                                                    } else if (accntType5.equals("p")) {
                                                            id5 =
Integer.parseInt(cnsl.readLine("Please enter the process id for the process account: "));
                                                            accntTypeInsertQuery5 = "INSERT INTO
proc_accnt VALUES (" + accntNo5 +", " + details5 + ", " + id5 + ")";

        conn.createStatement().executeQuery(accntTypeInsertQuery5);
                                                    } else if (accntType5.equals("d")) {
                                                            id5 =
Integer.parseInt(cnsl.readLine("Please enter the department number for the department
account: "));
                                                            accntTypeInsertQuery5 = "INSERT INTO
dept_accnt VALUES (" + accntNo5 +", " + details5 + ", " + id5 + ")";

        conn.createStatement().executeQuery(accntTypeInsertQuery5);
                                                    }

                                                    // Print the success message.
                                                    System.out.println();
                                                    System.out.println("Account Successfully
Entered!!");


                                                    cnsl.readLine("Hit enter to
continue......");

                                                    break;
                                            case 6:
                                                    // Declare variables.
                                                    int jobNo6 = 0;
                                                    String strtDt6 = null;
                                                    int asmblyId6 = 0;
                                                    int procId6 = 0;
                                                    String jobType6 = null;
                                                    String jobsInsertQuery6 = null;
                                                    String jobassgnInsertQuery6 = null;
                                                    String jobTypeInserQuery6 = null;

                                                    // Present user prompt for the data to be
entered
                                                    // accept the same using Console Reader.
                                                    jobNo6 =
Integer.parseInt(cnsl.readLine("Please enter the job number: "));
                                                    strtDt6 = cnsl.readLine("Please enter the
job start date in MM/DD/YYYY format: ");


                                                    // Create the query to be sent to Oracle.
                                                    jobsInsertQuery6 = "INSERT INTO jobs
(job_no, strt_dt) VALUES (" + jobNo6 + ", "
                                                            + "TO_DATE('" + strtDt6 + "',
'MM/DD/YYYY'))";

                                                    // Execute the statement.
        conn.createStatement().executeQuery(jobsInsertQuery6);

                                                jobType6 = cnsl.readLine("Please enter f/c/p
if the job is fit/cut/paint job respectively: ");

                                                if (jobType6.equals("f")) {
                                                        jobTypeInserQuery6 = "INSERT INTO
fit_job VALUES (" + jobNo6 + ")";

        conn.createStatement().executeQuery(jobTypeInserQuery6);
                                                } else if (jobType6.equals("c")) {
                                                        jobTypeInserQuery6 = "INSERT INTO
cut_job (job_no) VALUES (" + jobNo6 + ")";

        conn.createStatement().executeQuery(jobTypeInserQuery6);
                                                } else if (jobType6.equals("p")) {
                                                        jobTypeInserQuery6 = "INSERT INTO
pnt_job (job_no) VALUES (" + jobNo6 + ")";

        conn.createStatement().executeQuery(jobTypeInserQuery6);
                                                }

                                                asmblyId6 =
Integer.parseInt(cnsl.readLine("Please enter the assembly id for the job: "));
                                                procId6 =
Integer.parseInt(cnsl.readLine("Please enter the proess id for the job: "));

                                                jobassgnInsertQuery6 = "INSERT INTO
job_assgn VALUES (" + jobNo6 + "," + asmblyId6 + "," + procId6 + ")";


        conn.createStatement().executeQuery(jobassgnInsertQuery6);

                                                // Print the success message.
                                                System.out.println();
                                                System.out.println("Job Successfully
Entered!!");


                                                cnsl.readLine("Hit enter to
continue......");


                                                break;
                                        case 7:
                                                // Declare variables.
                                                int jobNo7 = 0;
                                                String endDt6 = null;
                                                int hours7 = 0;
                                                String mchn7 = null;
                                                int usd7 = 0;
                                                String mtrl7 = null;
                                                String color7 = null;
                                                int vol7 = 0;
                                                String jobsUpdateQuery7 = null;
                                                String jobTypeUpdQuery7 = null;

                                                // Present user prompt for the data to be
entered
                                                // accept the same using Console Reader.
                                                jobNo7 =
Integer.parseInt(cnsl.readLine("Please enter the job number to update: "));
```

```
                                              endDt6 = cnsl.readLine("Please enter the job 
end date in MM/DD/YYYY format: ");
                                              hours7 = 
Integer.parseInt(cnsl.readLine("Please enter the hours taken by the job: "));


                                              // Create the query to be sent to Oracle.
                                              jobsUpdateQuery7 = "UPDATE jobs SET 
end_dt=TO_DATE('" + endDt6 + "', 'MM/DD/YYYY'), lbr_tim=NUMTODSINTErVAL(" + hours7 + ", 
'HOUR') WHERE job_no = +" + jobNo7;

                                              // Execute the statement.
                                              
        conn.createStatement().executeQuery(jobsUpdateQuery7);

                                              //System.out.println("SELECT * FROM fit_job 
WHERE job_no = " + jobNo7);
                                              //System.out.println("SELECT * FROM cut_job 
WHERE job_no = " + jobNo7);

                                              ResultSet rs1 = 
conn.createStatement().executeQuery("SELECT * FROM fit_job WHERE job_no = " + jobNo7);
                                              ResultSet rs2 = 
conn.createStatement().executeQuery("SELECT * FROM cut_job WHERE job_no = " + jobNo7);

                                              if(rs1.next()){
                                                     break;
                                              } else if (rs2.next()){
                                                     mchn7 = cnsl.readLine("Please enter 
the machine type used for the cut job: ");
                                                     usd7 = 
Integer.parseInt(cnsl.readLine("Please enter the amount of time machine was used (in 
hrs): "));
                                                     mtrl7 = cnsl.readLine("Please enter 
the material used for the cut job: ");
                                                     jobTypeUpdQuery7 = "UPDATE cut_job 
SET mchn_typ='" + mchn7 + "', tim_usd=NUMTODSINTERVAL(" 
                                                             + usd7 + ", 'HOUR'), 
mtrl_usd='" + mtrl7 + "' WHERE job_no = " + jobNo7;
                                                     
        conn.createStatement().executeQuery(jobTypeUpdQuery7);
                                              } else {
                                                     color7 = cnsl.readLine("Please enter 
the colorfor the paint job: ");
                                                     vol7 = 
Integer.parseInt(cnsl.readLine("Please enter the volume of paint: "));
                                                     jobTypeUpdQuery7 = "UPDATE pnt_job 
SET color='"+color7+"', vol = "+vol7+" WHERE job_no = "+jobNo7;
                                                     
        conn.createStatement().executeQuery(jobTypeUpdQuery7);
                                              };


                                              // Print the success message.
                                              System.out.println();
                                              System.out.println("Job Successfully 
Updated!!");


                                              cnsl.readLine("Hit enter to 
continue......");

                                              break;
```

```
                                        case 8:
                                                // Declare variables.
                                                int trans8 = 0;
                                                double sup8 = 0.0;
                                                int job8 = 0;
                                                String transInsert8 = null;
                                                String aAct8 = null;
                                                String pAct8 = null;
                                                String dAct8 = null;
                                                String aAcc8 = null;
                                                String pAcc8 = null;
                                                String dAcc8 = null;

                                                // Present user prompt for the data to be
entered
                                                // accept the same using Console Reader.
                                                trans8 =
Integer.parseInt(cnsl.readLine("Please enter the transaction number: "));
                                                sup8 =
Double.parseDouble(cnsl.readLine("Please enter the transaction cost: "));
                                                job8 =
Integer.parseInt(cnsl.readLine("Please enter the job no for which transaction is being
recorded: "));

                                                // Create the query to be sent to Oracle.
                                                transInsert8 = "INSERT INTO trans VALUES ("
+ trans8 + ", " + sup8 + ", " + job8 + ")";
                                                aAct8 = "INSERT INTO a_accnt_trans"
                                                        + " SELECT " + trans8 + ", accnt_no
FROM (SELECT accnt_no FROM asmbly_accnt"
                                                        + " WHERE asmbly_id=(SELECT
asmbly_id FROM job_assgn WHERE job_no="
                                                        + "(SELECT job_no FROM trans WHERE
trans_id=" + trans8 + ")))";
                                                pAct8 = "INSERT INTO p_accnt_trans"
                                                        + " SELECT " + trans8 + ", accnt_no
FROM (SELECT accnt_no FROM proc_accnt"
                                                        + " WHERE proc_id=(SELECT proc_id
FROM job_assgn WHERE job_no="
                                                        + "(SELECT job_no FROM trans WHERE
trans_id=" + trans8 + ")))";
                                                dAct8 = "INSERT INTO d_accnt_trans"
                                                        + " SELECT " + trans8 + ", accnt_no
FROM (SELECT accnt_no FROM dept_accnt "
                                                        + "WHERE dept_no=(SELECT dept_no
FROM proc WHERE proc_id = (SELECT "
                                                        + "proc_id FROM job_assgn WHERE
job_no=(SELECT job_no FROM trans WHERE trans_id=" + trans8 + "))))";
                                                aAcc8 = "UPDATE asmbly_accnt SET
details1=details1+(SELECT sup_cost FROM trans WHERE trans_id = "+trans8+")"
                                                        + " WHERE asmbly_id=(SELECT
asmbly_id FROM job_assgn WHERE job_no=(SELECT job_no FROM trans WHERE
trans_id="+trans8+"))";
                                                pAcc8 = "UPDATE proc_accnt SET
details3=details3+(SELECT sup_cost FROM trans WHERE trans_id = "+trans8+")"
                                                        + " WHERE proc_id=(SELECT proc_id
FROM job_assgn WHERE job_no=(SELECT job_no FROM trans WHERE trans_id="+trans8+"))";
                                                dAcc8 = "UPDATE dept_accnt SET
details2=details2+(SELECT sup_cost FROM trans WHERE trans_id = "+trans8+")"
                                                        + " WHERE dept_no=(SELECT dept_no
FROM proc WHERE proc_id=(SELECT proc_id FROM job_assgn WHERE job_no=(SELECT job_no FROM
trans WHERE trans_id="+trans8+")))";
```

```
                                                    // Execute the statement.

        conn.createStatement().executeQuery(transInsert8);
                                                    conn.createStatement().executeQuery(aAct8);
                                                    conn.createStatement().executeQuery(pAct8);
                                                    conn.createStatement().executeQuery(dAct8);
                                                    conn.createStatement().executeQuery(aAcc8);
                                                    conn.createStatement().executeQuery(pAcc8);
                                                    conn.createStatement().executeQuery(dAcc8);

                                                    // Print the success message.
                                                    System.out.println();
                                                    System.out.println("Trans Successfully
Entered!!");

                                                    cnsl.readLine("Hit enter to
continue......");

                                                    break;
                                            case 9:
                                                    // Declare variables.
                                                    int asmbly9 = 0;
                                                    String asmCost9 = null;

                                                    // Present user with prompt for the data to
be entered
                                                    // accept the same using Console Reader.
                                                    asmbly9 =
Integer.parseInt(cnsl.readLine("Please enter the assembly id to retreive cost incurred:
"));

                                                    // Create the query to be sent to Oracle.
                                                    asmCost9 = "SELECT details1 FROM
asmbly_accnt WHERE asmbly_id = " + asmbly9;

                                                    // Fetch the results.
                                                    ResultSet rs3 =
conn.createStatement().executeQuery(asmCost9);

                                                    // Print output.
                                                    System.out.println();
                                                    while(rs3.next()){
                                                            System.out.println("Cost incurred on
the entered assembly id is : " + rs3.getDouble(1));
                                                    }

                                                    cnsl.readLine("Hit enter to
continue......");

                                                    break;
                                            case 10:
                                                    // Declare variables.
                                                    int asmbly10 = 0;
                                                    String asmLbr10 = null;

                                                    // Present user with prompt for the data to
be entered
                                                    // accept the same using Console Reader.
                                                    asmbly10 =
Integer.parseInt(cnsl.readLine("Please enter the assembly id to retreive labour time
recorded: "));
```

```
                                                // Create the query to be sent to Oracle.
                                                asmLbr10 = "SELECT SUM(EXTRACT(HOUR FROM
lbr_tim)) FROM jobs WHERE job_no IN ( SELECT job_no FROM job_assgn WHERE asmbly_id =
"+asmbly10+")";

                                                // Fetch the results.
                                                ResultSet rs4 =
conn.createStatement().executeQuery(asmLbr10);

                                                // Print output.
                                                System.out.println();
                                                while(rs4.next()){
                                                        System.out.println("Labor time
recorded on the entered assembly id is : " + rs4.getInt(1) + " Hrs");
                                                }

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 11:
                                                // Declare variables.
                                                int dept11 = 0;
                                                String dt11 = null;
                                                String depLbr11 = null;

                                                // Present user with prompt for the data to
be entered
                                                // accept the same using Console Reader.
                                                dept11 =
Integer.parseInt(cnsl.readLine("Please enter the dept number to retreive labour time
recorded: "));
                                                dt11 = cnsl.readLine("Please enter the date
to calculate the labor time for in MM/DD/YYYY: ");

                                                // Create the query to be sent to Oracle.
                                                depLbr11 = "SELECT SUM(EXTRACT(HOUR FROM
lbr_tim)) FROM jobs WHERE job_no IN "
                                                + "(SELECT job_no FROM job_assgn WHERE
proc_id IN (SELECT proc_id FROM proc WHERE dept_no = "+dept11+"))"
                                                + " AND strt_dt >=
TO_DATE('"+dt11+"','MM/DD/YYYY') AND end_dt <= TO_DATE('"+dt11+"','MM/DD/YYYY')";

                                                // Fetch the results.
                                                ResultSet rs5 =
conn.createStatement().executeQuery(depLbr11);

                                                // Print output.
                                                System.out.println();
                                                while(rs5.next()){
                                                        System.out.println("Labor time
recorded on the entered department id for the given date is : " + rs5.getInt(1) + "
Hrs");
                                                }

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 12:
                                                // Declare variables.
```

```
                                                  int asmbly12 = 0;
                                                  String asmProc12 = null;

                                                  // Present user with prompt for the data to
be entered
                                                  // accept the same using Console Reader.
                                                  asmbly12 =
Integer.parseInt(cnsl.readLine("Please enter the assembly id to retreive the processes
passed: "));

                                                  // Create the query to be sent to Oracle.
                                                  asmProc12 = "SELECT p.proc_id, p.dept_no
FROM (SELECT job_no, asmbly_id, proc_id FROM job_assgn "
                                                          + "WHERE asmbly_id = "+asmbly12+")
ja INNER JOIN jobs jo ON ja.job_no = jo.job_no INNER JOIN proc p "
                                                          + "ON ja.proc_id = p.proc_id ORDER
BY jo.strt_dt, p.proc_id";

                                                  // Fetch the results.
                                                  ResultSet rs6 =
conn.createStatement().executeQuery(asmProc12);

                                                  // Print output.
                                                  System.out.println();
                                                  System.out.println("Processes and the
Supervising departments that the provided assembly has passed through are:");
                                                  System.out.println("proc_id\tdept_no");
                                                  while(rs6.next()){
                                                          System.out.println(rs6.getInt(1) +
"\t" + rs6.getInt(2));
                                                  }

                                                  cnsl.readLine("Hit enter to
continue......");

                                                  break;
                                          case 13:
                                                  // Declare variables.
                                                  int dept13 = 0;
                                                  String dt13 = null;
                                                  String depJob13 = null;

                                                  // Present user with prompt for the data to
be entered
                                                  // accept the same using Console Reader.
                                                  dept13 =
Integer.parseInt(cnsl.readLine("Please enter the department number to retreive the jobs:
"));
                                                  dt13 = cnsl.readLine("Please enter the date
to retreive the jobs: ");

                                                  // Create the query to be sent to Oracle.
                                                  depJob13 = "SELECT j.job_no AS \"JOB
NUMBER\", 'FIT' AS \"JOB TYPE\", ja.asmbly_id AS \"ASSEMBLY ID\" FROM jobs j "
                                                          + "INNER JOIN job_assgn ja ON
j.job_no = ja.job_no INNER JOIN fit_job fj ON j.job_no = fj.job_no "
                                                          + "WHERE strt_dt >=
TO_DATE('"+dt13+"', 'MM/DD/YYYY') AND end_dt <= TO_DATE('"+dt13+"', 'MM/DD/YYYY') "
                                                          + " AND ja.proc_id IN  ( SELECT
proc_id FROM proc WHERE dept_no="+dept13+" ) "
                                                          + "UNION "
```

```
                                                     + " SELECT j.job_no AS \"JOB
NUMBER\", 'PAINT' AS \"JOB TYPE\", ja.asmbly_id AS \"ASSEMBLY ID\" FROM jobs j "
                                                     + " INNER JOIN job_assgn ja ON
j.job_no = ja.job_no INNER JOIN pnt_job pj ON j.job_no = pj.job_no"
                                                     + " WHERE strt_dt >=
TO_DATE('"+dt13+"', 'MM/DD/YYYY') AND end_dt <= TO_DATE('"+dt13+"', 'MM/DD/YYYY')"
                                                     + " AND ja.proc_id IN ( SELECT
proc_id FROM proc WHERE dept_no="+dept13+" )"
                                                     + " UNION"
                                                     + " SELECT j.job_no AS \"JOB
NUMBER\", 'CUT' AS \"JOB TYPE\", ja.asmbly_id AS \"ASSEMBLY ID\" FROM jobs j "
                                                     + " INNER JOIN job_assgn ja ON
j.job_no = ja.job_no INNER JOIN cut_job cj ON j.job_no = cj.job_no "
                                                     + " WHERE strt_dt >=
TO_DATE('"+dt13+"', 'MM/DD/YYYY') AND end_dt <= TO_DATE('"+dt13+"', 'MM/DD/YYYY') "
                                                     + " AND ja.proc_id IN ( SELECT
proc_id FROM proc WHERE dept_no="+dept13+" )";

                                             // Fetch the results.
                                             ResultSet rs7 =
conn.createStatement().executeQuery(depJob13);

                                             // Print output.
                                             System.out.println();
                                             System.out.println("Jobs completed in the
department for the provided date are:");

        System.out.println("job_no\tjob_type\tassembly_id");
                                             while(rs7.next()){
                                                     System.out.println(rs7.getInt(1) +
"\t" + rs7.getString(2) + "\t\t" + rs7.getInt(3));
                                             }

                                             cnsl.readLine("Hit enter to
continue......");

                                             break;
                                     case 14:
                                             // Declare variables.
                                             String pmthd14 = null;
                                             String redAss = null;

                                             // Present user with prompt for the data to
be entered
                                             // accept the same using Console Reader.
                                             pmthd14 = cnsl.readLine("Please enter the
paint method to retreive the customers with Red assemblies: ");

                                             // Create the query to be sent to Oracle.
                                             redAss = "SELECT cust_nm FROM asmbly WHERE
asmbly_id IN ( SELECT asmbly_id"
                                                     + " FROM job_assgn WHERE job_no IN (
SELECT job_no FROM pnt_job WHERE color='RED' )"
                                                     + " AND proc_id IN ( SELECT proc_id
FROM pnt_proc WHERE pnt_mthd='"+pmthd14+"' ) ) ORDER BY cust_nm";

                                             // Fetch the results.
                                             ResultSet rs8 =
conn.createStatement().executeQuery(redAss);

                                             // Print output.
                                             System.out.println();
```

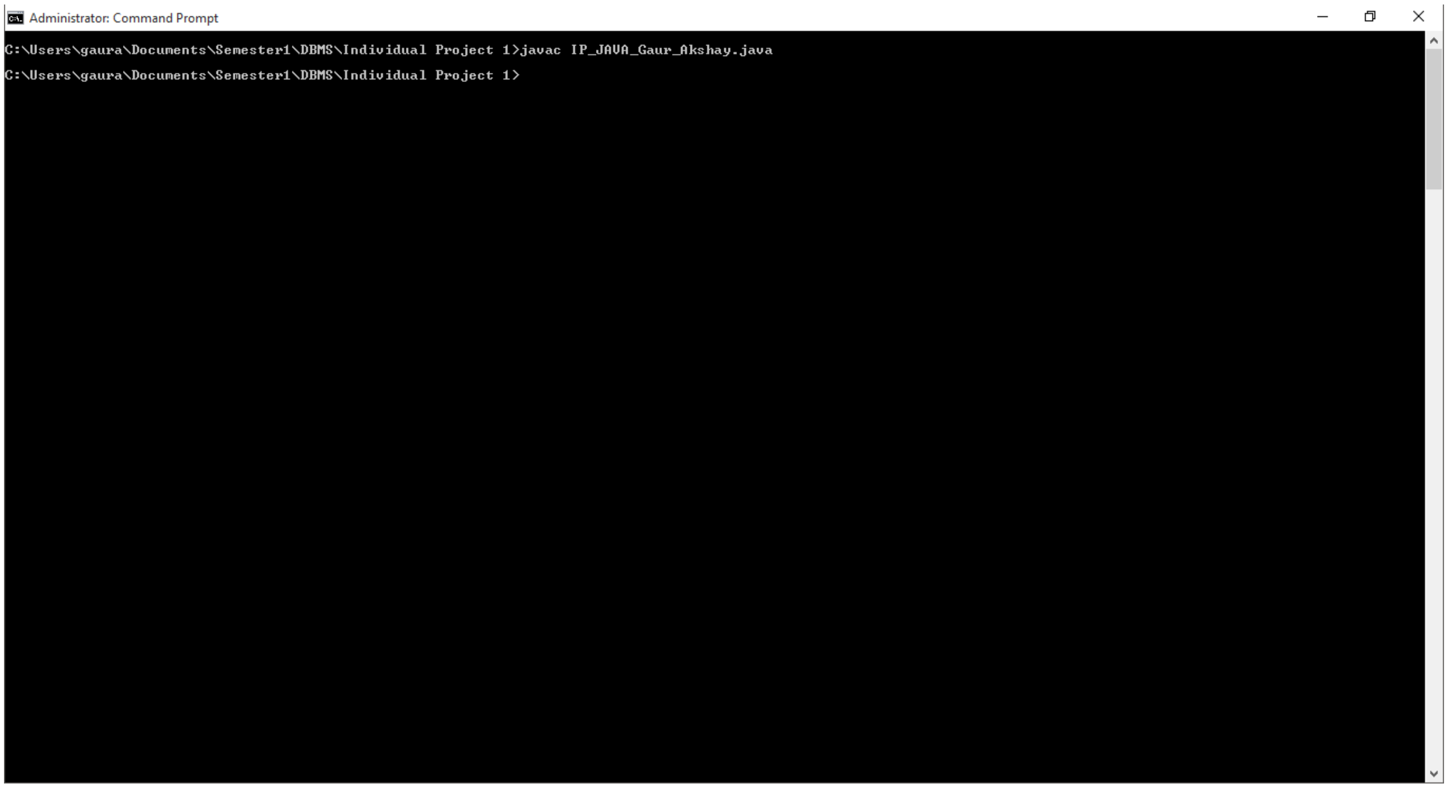
```
                                                System.out.println("Customers that fulfill
the criteria are:");
                                                while(rs8.next()){

        System.out.println(rs8.getString(1));
                                                }

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 15:
                                                // Declare variables.
                                                int jobLow15 = 0;
                                                int jobHigh15 = 0;
                                                String del15 = null;

                                                // Present user with prompt for the data to
be entered
                                                // accept the same using Console Reader.
                                                jobLow15 =
Integer.parseInt(cnsl.readLine("Please enter the lower range of the job: "));
                                                jobHigh15 =
Integer.parseInt(cnsl.readLine("Please enter the upper range of the job: "));

                                                // Create the query to be sent to Oracle.
                                                del15 = "DELETE FROM cut_job WHERE job_no
BETWEEN "+jobLow15+" AND "+jobHigh15;

                                                // Fetch the results.
                                                conn.createStatement().executeQuery(del15);

                                                // Print success message.
                                                System.out.println();
                                                System.out.println("Jobs successfully
deleted!");

                                                cnsl.readLine("Hit enter to
continue......");

                                                break;
                                        case 16:
                                                // Declare variables.
                                                int job16 = 0;
                                                String setRed15 = null;

                                                // Present user with prompt for the data to
be entered
                                                // accept the same using Console Reader.
                                                job16 =
Integer.parseInt(cnsl.readLine("Please enter the job for which paint color should be
changed to Red: "));

                                                // Create the query to be sent to Oracle.
                                                setRed15 = "UPDATE pnt_job SET color = 'RED'
WHERE job_no = " +job16;

                                                // Fetch the results.

        conn.createStatement().executeQuery(setRed15);

                                                // Print success message.
```

```
                                              System.out.println();
                                              System.out.println("Color successfully
updated!");

                                              cnsl.readLine("Hit enter to
continue......");

                                              break;
                                      case 17:
                                              // Create the query to be sent to Oracle.
                                              String acAvg17 = "SELECT
TO_CHAR(AVG(expense),'999,999.99') FROM "
                                                      + " ( SELECT details1 AS expense
FROM asmbly_accnt "
                                                      + "UNION ALL "
                                                      + " SELECT details3 AS expense FROM
proc_accnt "
                                                      + "UNION ALL "
                                                      + "SELECT details2 AS expense FROM
dept_accnt )";

                                              // Fetch the results.
                                              ResultSet rs9 =
conn.createStatement().executeQuery(acAvg17);

                                              // Print output.
                                              System.out.println();
                                              System.out.println("Average of the acounts
are:");
                                              while(rs9.next()){

        System.out.println(rs9.getString(1));
                                              }

                                              cnsl.readLine("Hit enter to
continue......");
                                              break;
                                      case 18:
                                              String filename18 = cnsl.readLine("Please
enter the file name: ");

                                              String filepath18 =
"C:\\Users\\gaura\\Documents\\"+filename18;

                                              String  thisLine = null;
                                            try{
                                               // open input stream test.txt for reading
purpose.
                                               BufferedReader br = new BufferedReader(new
FileReader(filepath18));
                                               while ((thisLine = br.readLine()) != null)
{

        conn.createStatement().executeQuery("INSERT INTO cust VALUES ("+ thisLine +")");
                                               }
                                            }catch(Exception e){
                                               e.printStackTrace();
                                            }
                                              //Print success message

                                              System.out.println("Customers entered
successfully!!!");
```

```
                                             cnsl.readLine("Hit enter to
continue......");
                                             break;
                                     case 19:
                                             // Declare variables.
                                             String pmthd19 = null;
                                             String file19 = null;
                                             String filepath = null;
                                             String redAss19 = null;

                                             // Present user with prompt for the data to
be entered
                                             // accept the same using Console Reader.
                                             pmthd19 = cnsl.readLine("Please enter the
paint method to retreive the customers with Red assemblies: ");
                                             file19 = cnsl.readLine("Please enter the
file name (.csv): ");

                                             filepath = "C:\\Users\\gaura\\Documents\\" +
file19;

                                             // Create the query to be sent to Oracle.
                                             redAss19 = "SELECT * FROM cust WHERE cust_nm
IN (SELECT cust_nm FROM asmbly WHERE asmbly_id IN ( SELECT asmbly_id"
                                                     + " FROM job_assgn WHERE job_no IN (
SELECT job_no FROM pnt_job WHERE color='RED' )"
                                                     + " AND proc_id IN ( SELECT proc_id
FROM pnt_proc WHERE pnt_mthd='"+pmthd19+"' ) ) ) ORDER BY cust_nm";

                                             // Fetch the results.
                                             ResultSet rs10 =
conn.createStatement().executeQuery(redAss19);

                                             try{
                                                     FileWriter writer = new
FileWriter(filepath);
                                                     writer.append("'");
                                                     writer.append("Cust Name");
                                                     writer.append("'");
                                                     writer.append(",");
                                                     writer.append("'");
                                                     writer.append("Cust Add");
                                                     writer.append("'");
                                                     writer.append("\r\n");
                                                     // Write the result set to file
                                                     while(rs10.next()){
                                                     writer.append("'");
                                                     writer.append(rs10.getString(1));
                                                     writer.append("'");
                                                     writer.append(",");
                                                     writer.append("'");
                                                     writer.append(rs10.getString(2));
                                                     writer.append("'");
                                                     writer.append("\r\n");
                                                     }
                                                     writer.flush();
                                                     writer.close();
                                             } catch(IOException e)
{e.printStackTrace();}
```

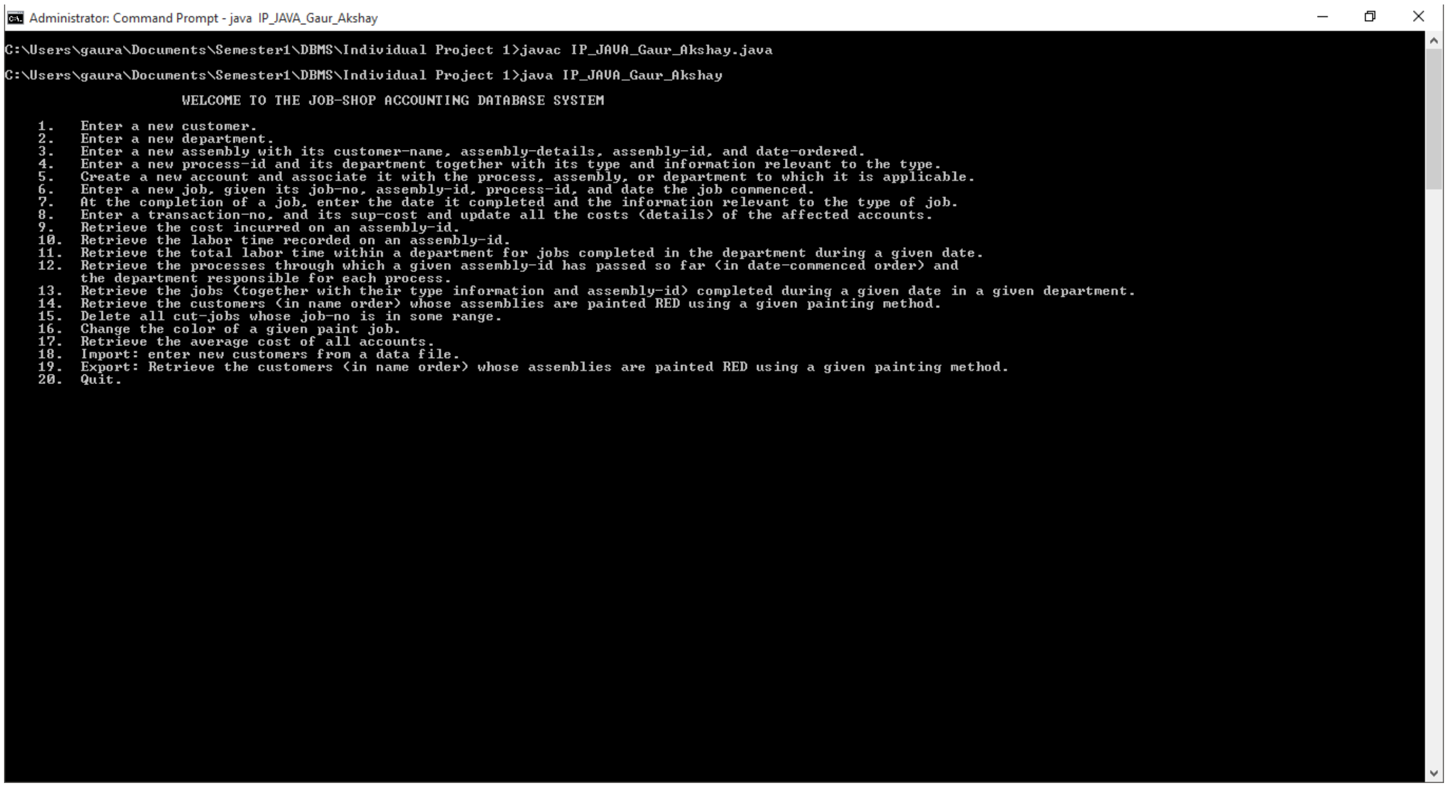
```
                                                        cnsl.readLine("Hit enter to 
continue......");
                                                        
                                                        break;
                                                case 20:
                                                        System.out.println();
                                                        System.out.println("Exiting the program!");
                                                        loop=1;
                                                        break;
                                                default :
                                                        //Ask the user to correct the choice 
entered.
                                                        
                                                        System.out.println("Incorrect option chosen, 
please choose again!");
                                                        break;
                                        }
                                }
                // Catch any exception.
                } catch(Exception e) {
                        System.out.println(e);
                // Close the connection before exiting the program.
                } finally {
                        if(conn != null){
                                try {
                                        conn.close();
                                } catch (Exception e) {System.out.println(e);}
                        }
                }
        }
}
```

SUCCESSFUL COMPILATION:



```
C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>javac IP_JAVA_Gaur_Akshay.java

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>
```

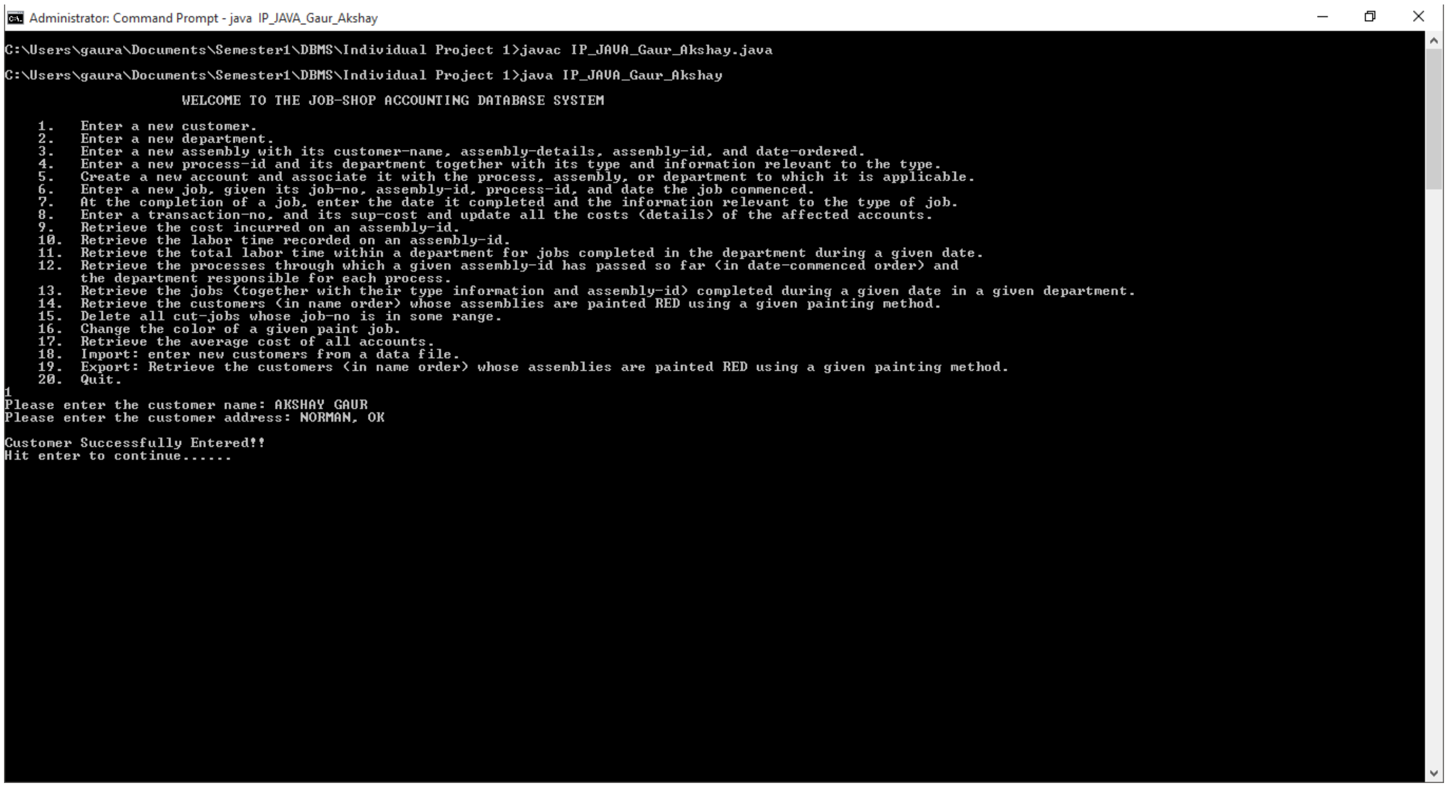
EXECUTION


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>javac IP_JAVA_Gaur_Akshay.java

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>java IP_JAVA_Gaur_Akshay

                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
```

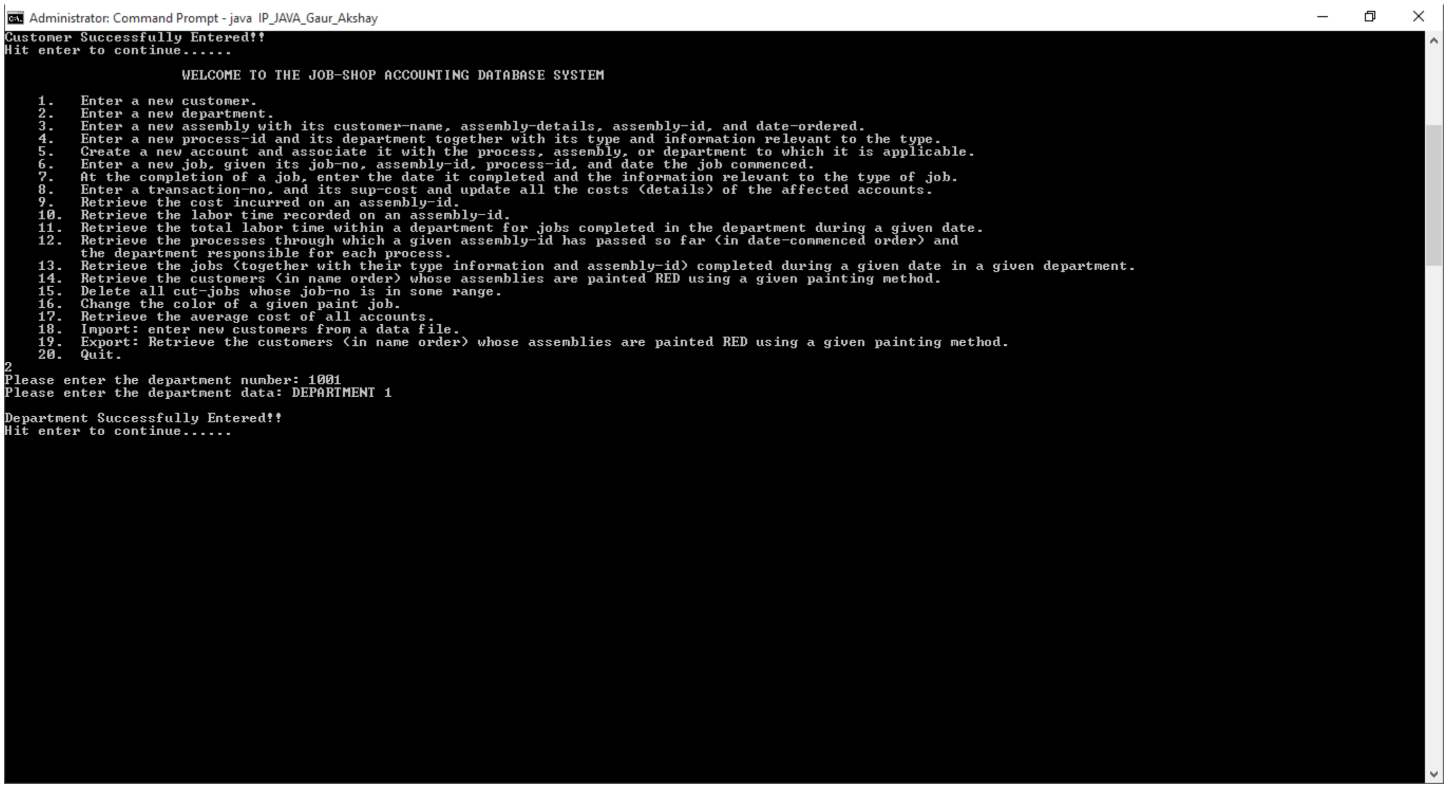
TASK 6 Java program Execution

QUERY 1


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>javac IP_JAVA_Gaur_Akshay.java

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>java IP_JAVA_Gaur_Akshay

                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
1
Please enter the customer name: AKSHAY GAUR
Please enter the customer address: NORMAN, OK

Customer Successfully Entered!!
Hit enter to continue......
```

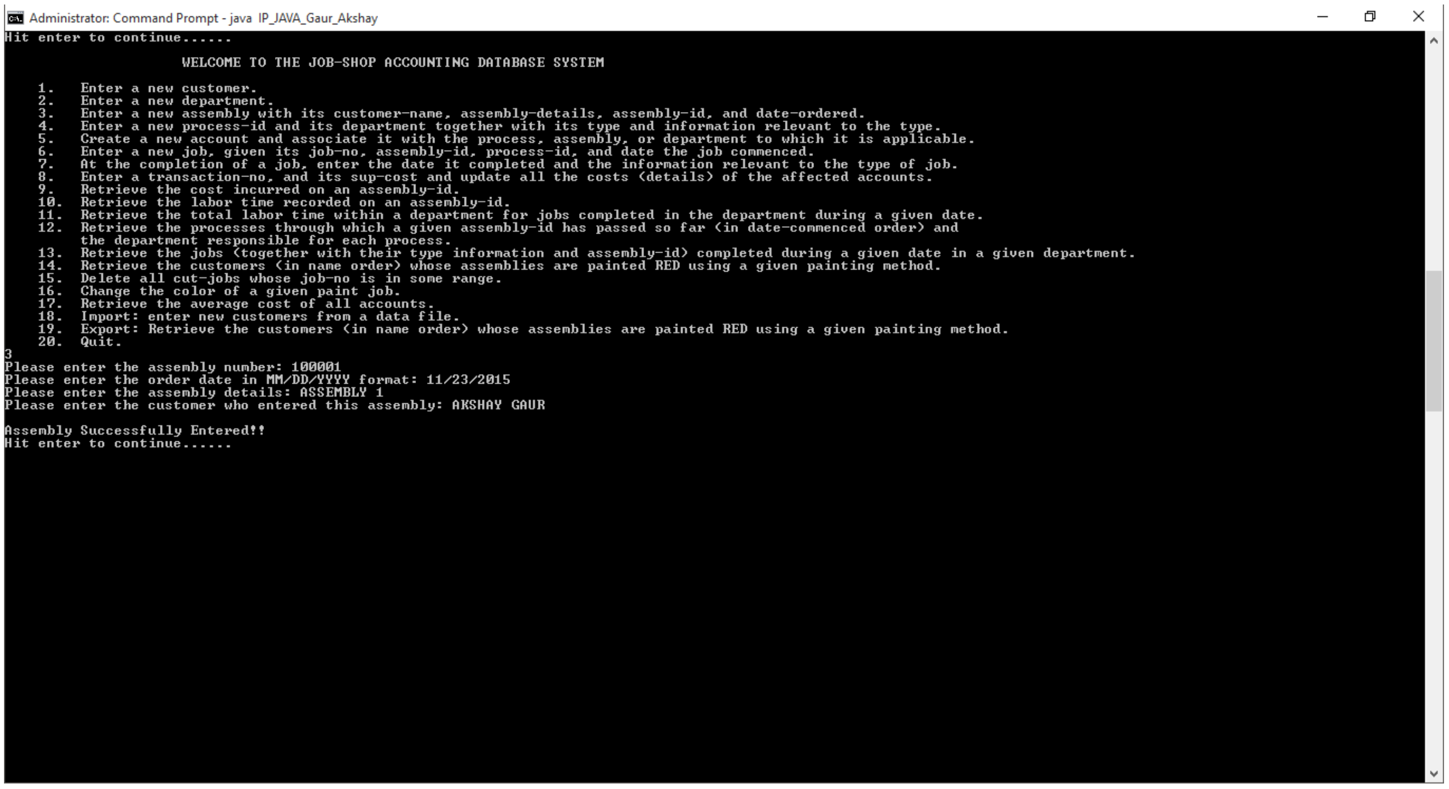
QUERY 2



Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
Customer Successfully Entered!!
Hit enter to continue......

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
2
Please enter the department number: 1001
Please enter the department data: DEPARTMENT 1

Department Successfully Entered!!
Hit enter to continue......
```

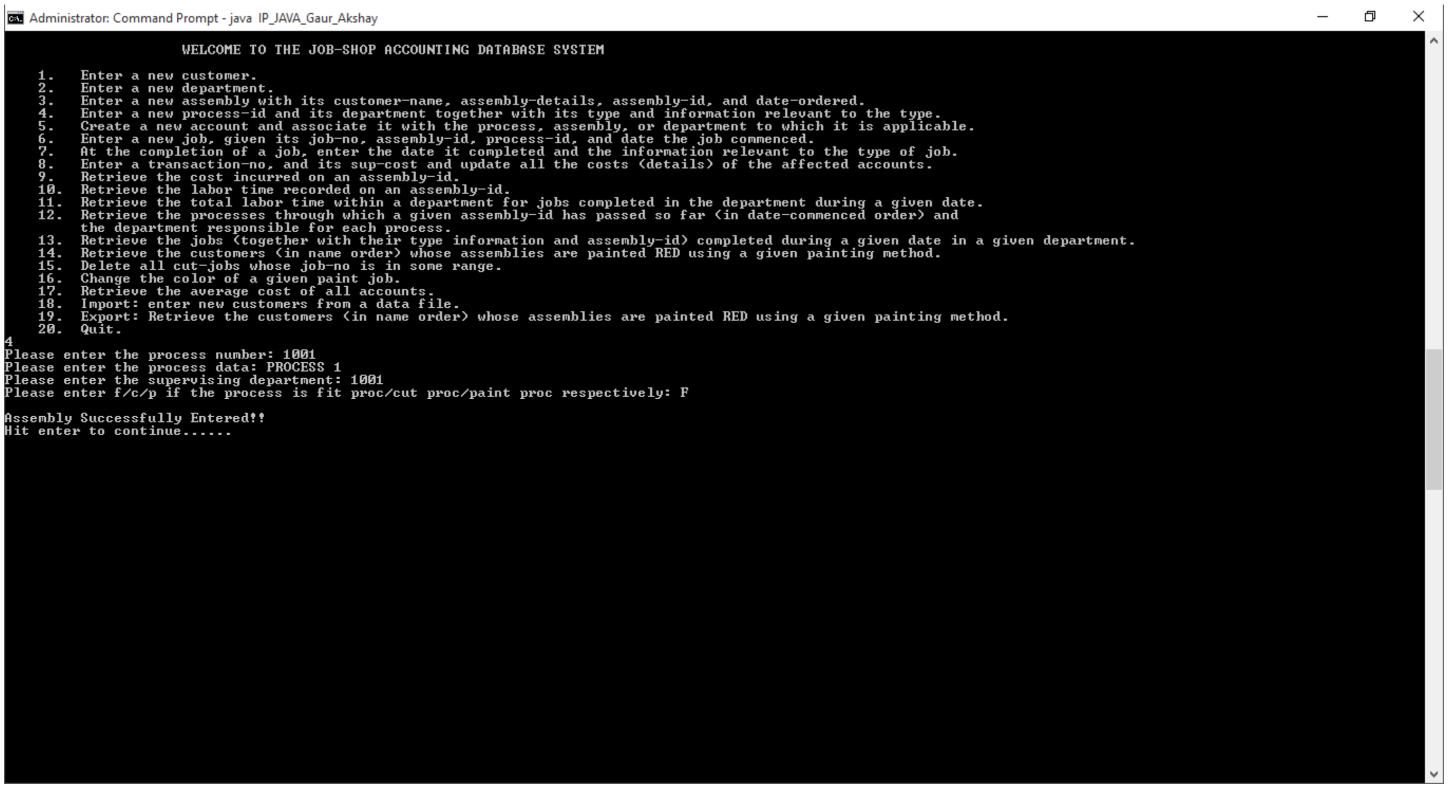
QUERY 3



Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
Hit enter to continue......

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
3
Please enter the assembly number: 100001
Please enter the order date in MM/DD/YYYY format: 11/23/2015
Please enter the assembly details: ASSEMBLY 1
Please enter the customer who entered this assembly: AKSHAY GAUR

Assembly Successfully Entered!!
Hit enter to continue......
```

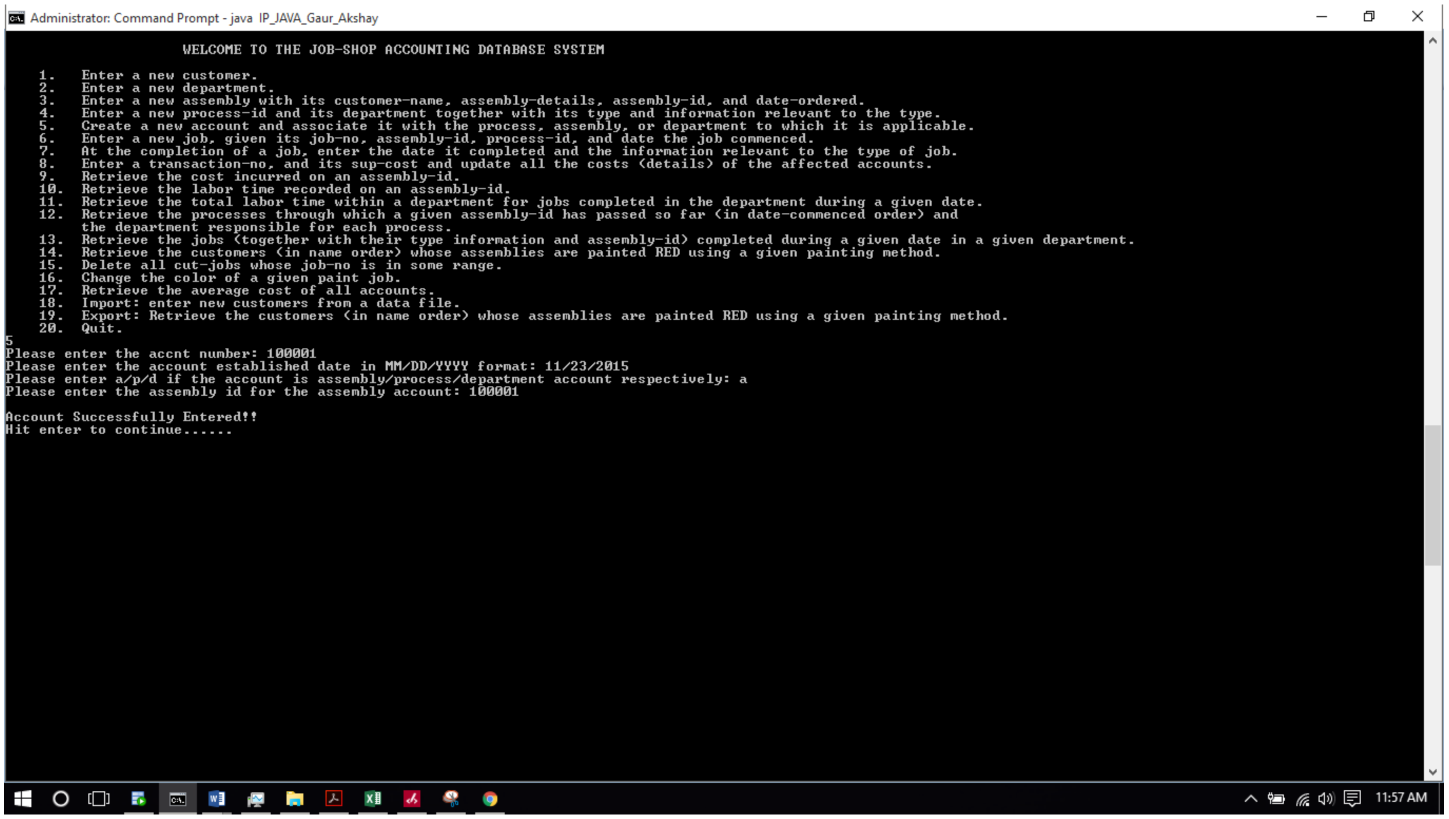
QUERY 4



Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
4
Please enter the process number: 1001
Please enter the process data: PROCESS 1
Please enter the supervising department: 1001
Please enter f/c/p if the process is fit proc/cut proc/paint proc respectively: F

Assembly Successfully Entered!!
Hit enter to continue......
```

QUERY 5

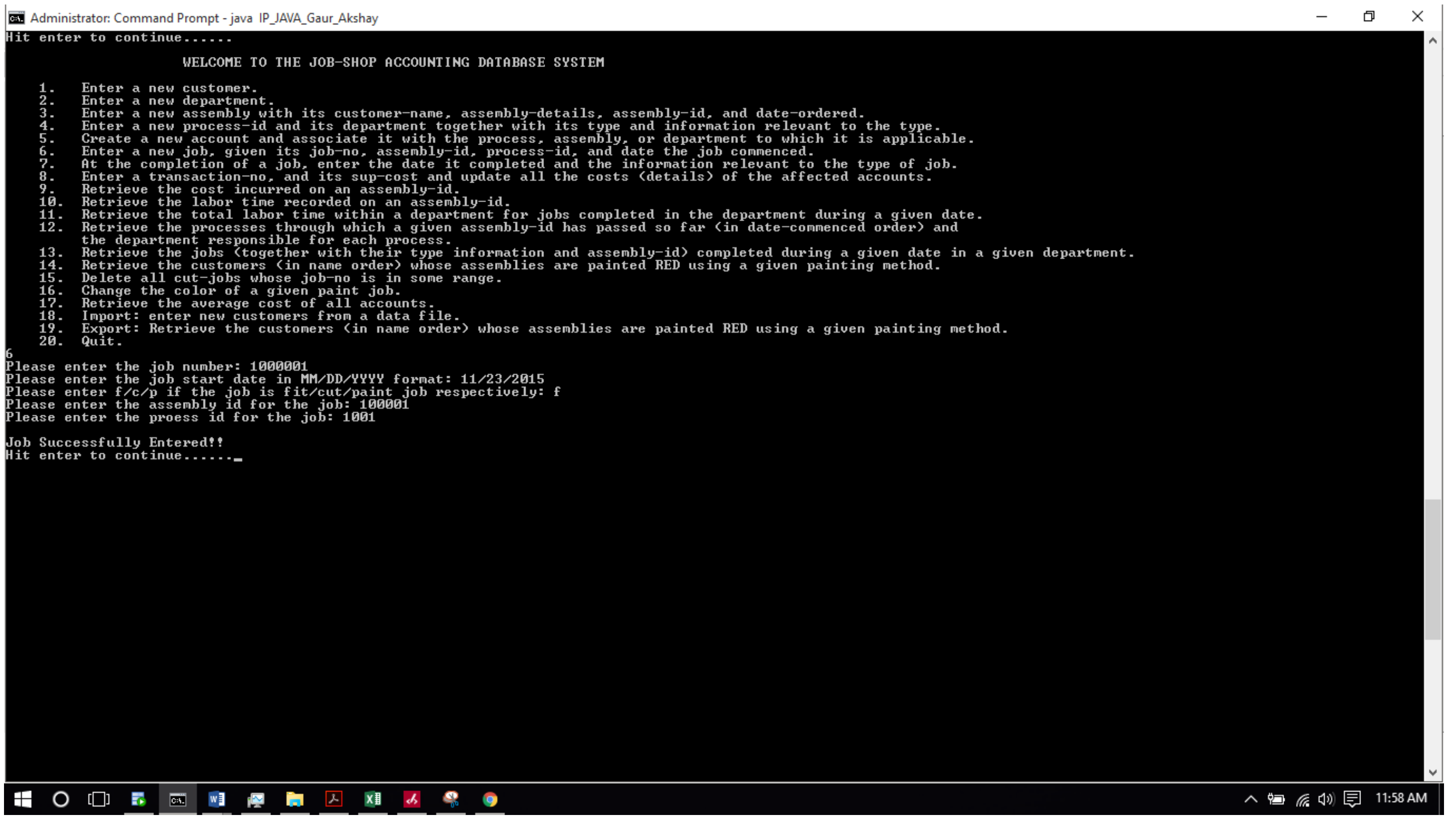


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
5
Please enter the accnt number: 100001
Please enter the account established date in MM/DD/YYYY format: 11/23/2015
Please enter a/p/d if the account is assembly/process/department account respectively: a
Please enter the assembly id for the assembly account: 100001

Account Successfully Entered!!
Hit enter to continue......
```

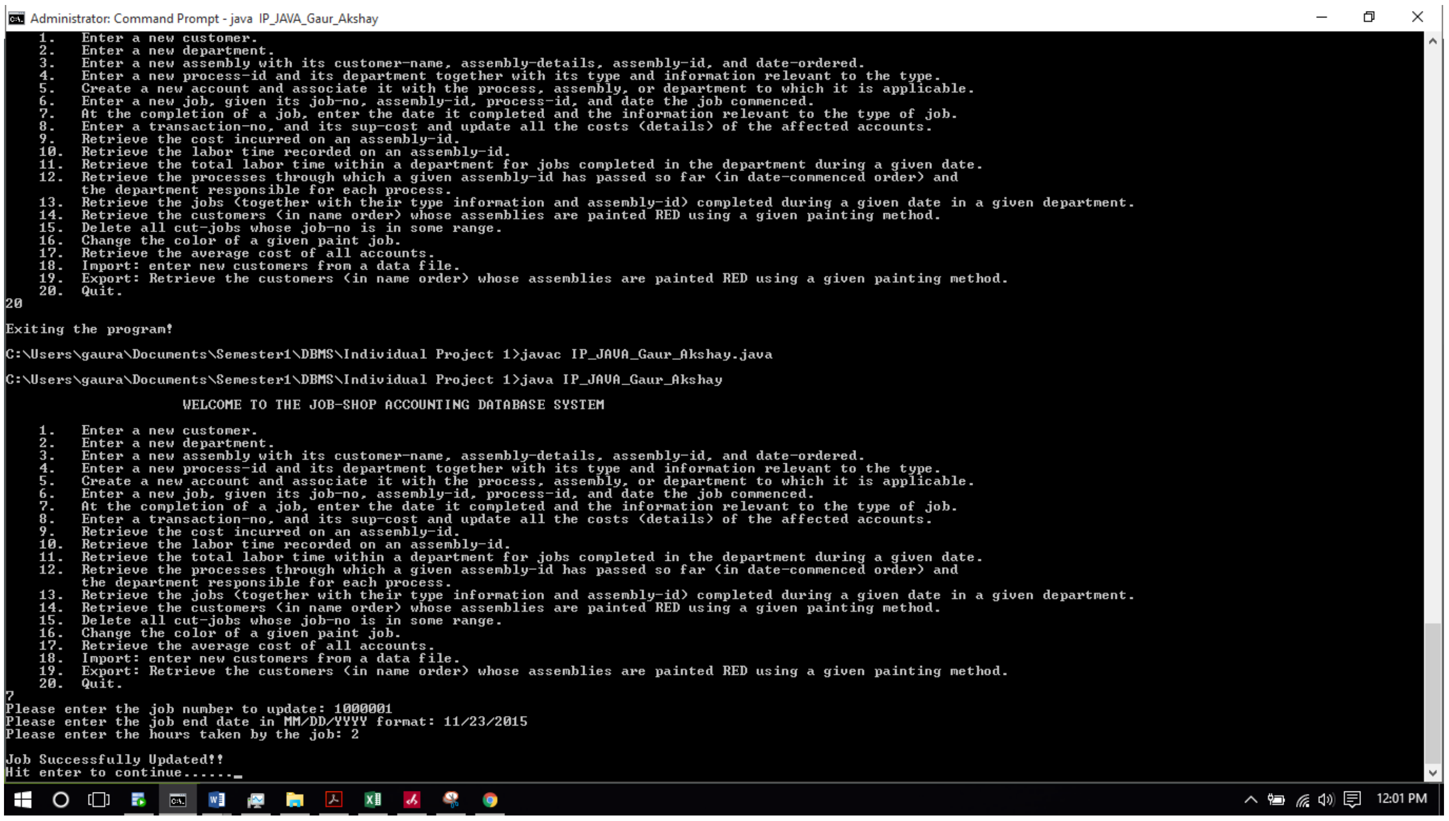
11:57 AM

QUERY 6



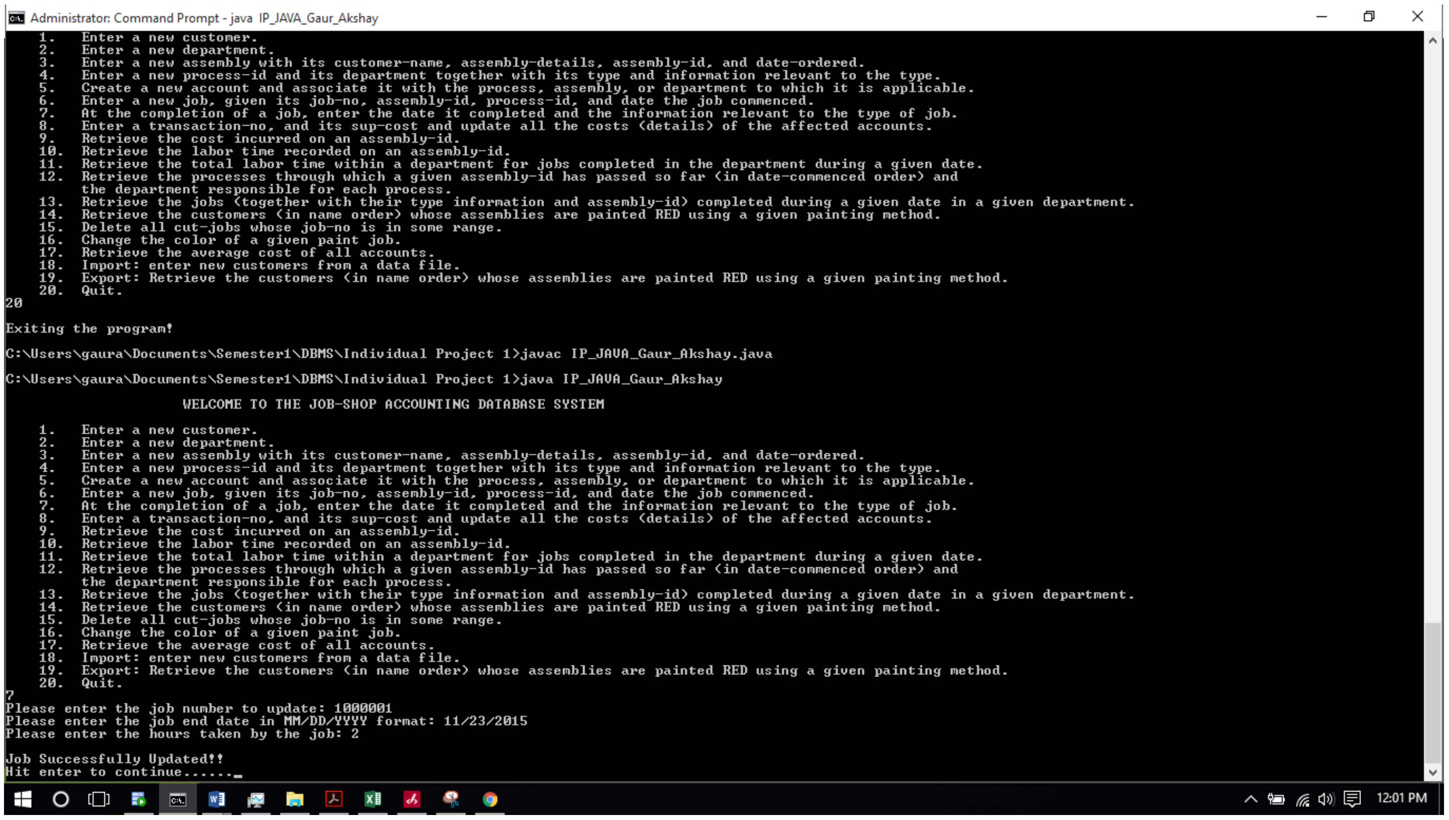
```
Hit enter to continue......

                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
6
Please enter the job number: 1000001
Please enter the job start date in MM/DD/YYYY format: 11/23/2015
Please enter f/c/p if the job is fit/cut/paint job respectively: f
Please enter the assembly id for the job: 100001
Please enter the proess id for the job: 1001

Job Successfully Entered!!
Hit enter to continue......_
```

QUERY 7

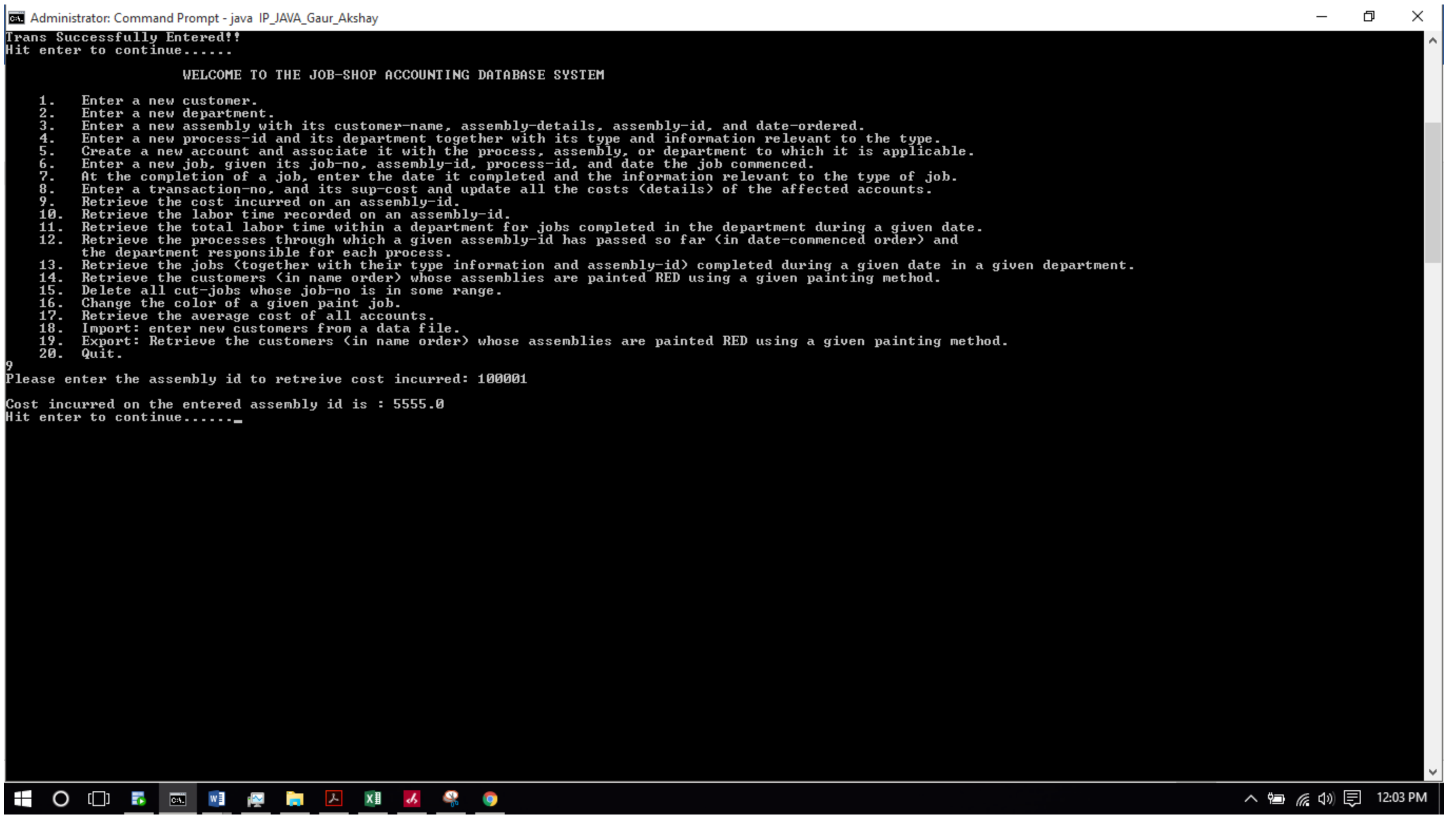


```
    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
20

Exiting the program!

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>javac IP_JAVA_Gaur_Akshay.java

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>java IP_JAVA_Gaur_Akshay

                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
7
Please enter the job number to update: 1000001
Please enter the job end date in MM/DD/YYYY format: 11/23/2015
Please enter the hours taken by the job: 2

Job Successfully Updated!!
Hit enter to continue......_
```

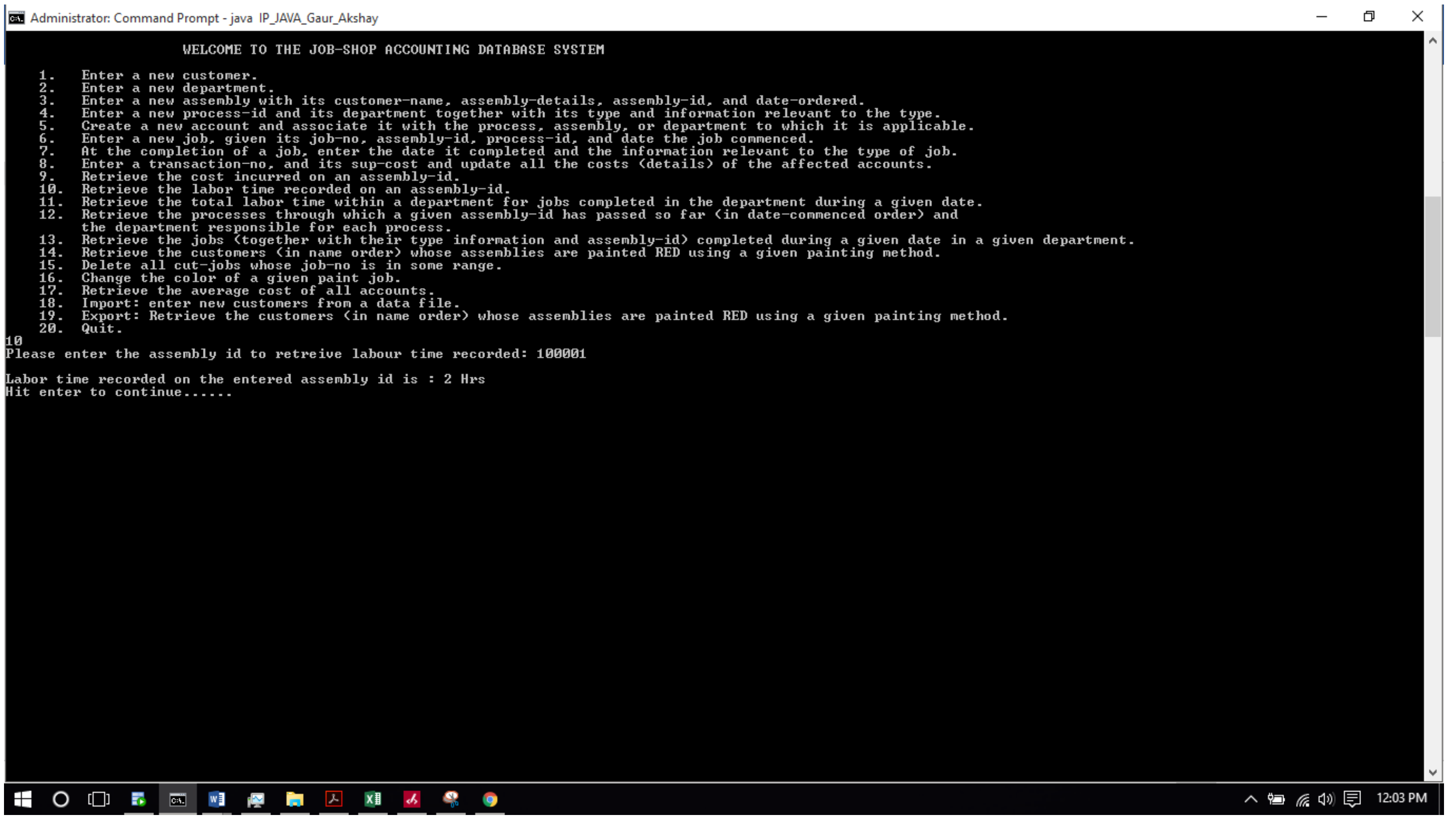
QUERY 8


```
Administrator: Command Prompt - java  IP_JAVA_Gaur_Akshay
    1.    Enter a new customer.
    2.    Enter a new department.
    3.    Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.    Enter a new process-id and its department together with its type and information relevant to the type.
    5.    Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.    Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.    At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.    Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.    Retrieve the cost incurred on an assembly-id.
    10.   Retrieve the labor time recorded on an assembly-id.
    11.   Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.   Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
          the department responsible for each process.
    13.   Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.   Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.   Delete all cut-jobs whose job-no is in some range.
    16.   Change the color of a given paint job.
    17.   Retrieve the average cost of all accounts.
    18.   Import: enter new customers from a data file.
    19.   Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.   Quit.
20

Exiting the program!

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>javac IP_JAVA_Gaur_Akshay.java

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>java IP_JAVA_Gaur_Akshay

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.    Enter a new customer.
    2.    Enter a new department.
    3.    Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.    Enter a new process-id and its department together with its type and information relevant to the type.
    5.    Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.    Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.    At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.    Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.    Retrieve the cost incurred on an assembly-id.
    10.   Retrieve the labor time recorded on an assembly-id.
    11.   Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.   Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
          the department responsible for each process.
    13.   Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.   Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.   Delete all cut-jobs whose job-no is in some range.
    16.   Change the color of a given paint job.
    17.   Retrieve the average cost of all accounts.
    18.   Import: enter new customers from a data file.
    19.   Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.   Quit.
7
Please enter the job number to update: 1000001
Please enter the job end date in MM/DD/YYYY format: 11/23/2015
Please enter the hours taken by the job: 2

Job Successfully Updated!!
Hit enter to continue......_
```
12:01 PM

QUERY 9


```
Administrator: Command Prompt - java  IP_JAVA_Gaur_Akshay
Trans Successfully Entered!!
Hit enter to continue......

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.    Enter a new customer.
    2.    Enter a new department.
    3.    Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.    Enter a new process-id and its department together with its type and information relevant to the type.
    5.    Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.    Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.    At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.    Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.    Retrieve the cost incurred on an assembly-id.
    10.   Retrieve the labor time recorded on an assembly-id.
    11.   Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.   Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
          the department responsible for each process.
    13.   Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.   Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.   Delete all cut-jobs whose job-no is in some range.
    16.   Change the color of a given paint job.
    17.   Retrieve the average cost of all accounts.
    18.   Import: enter new customers from a data file.
    19.   Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.   Quit.
9
Please enter the assembly id to retreive cost incurred: 100001

Cost incurred on the entered assembly id is : 5555.0
Hit enter to continue......_
```
12:03 PM

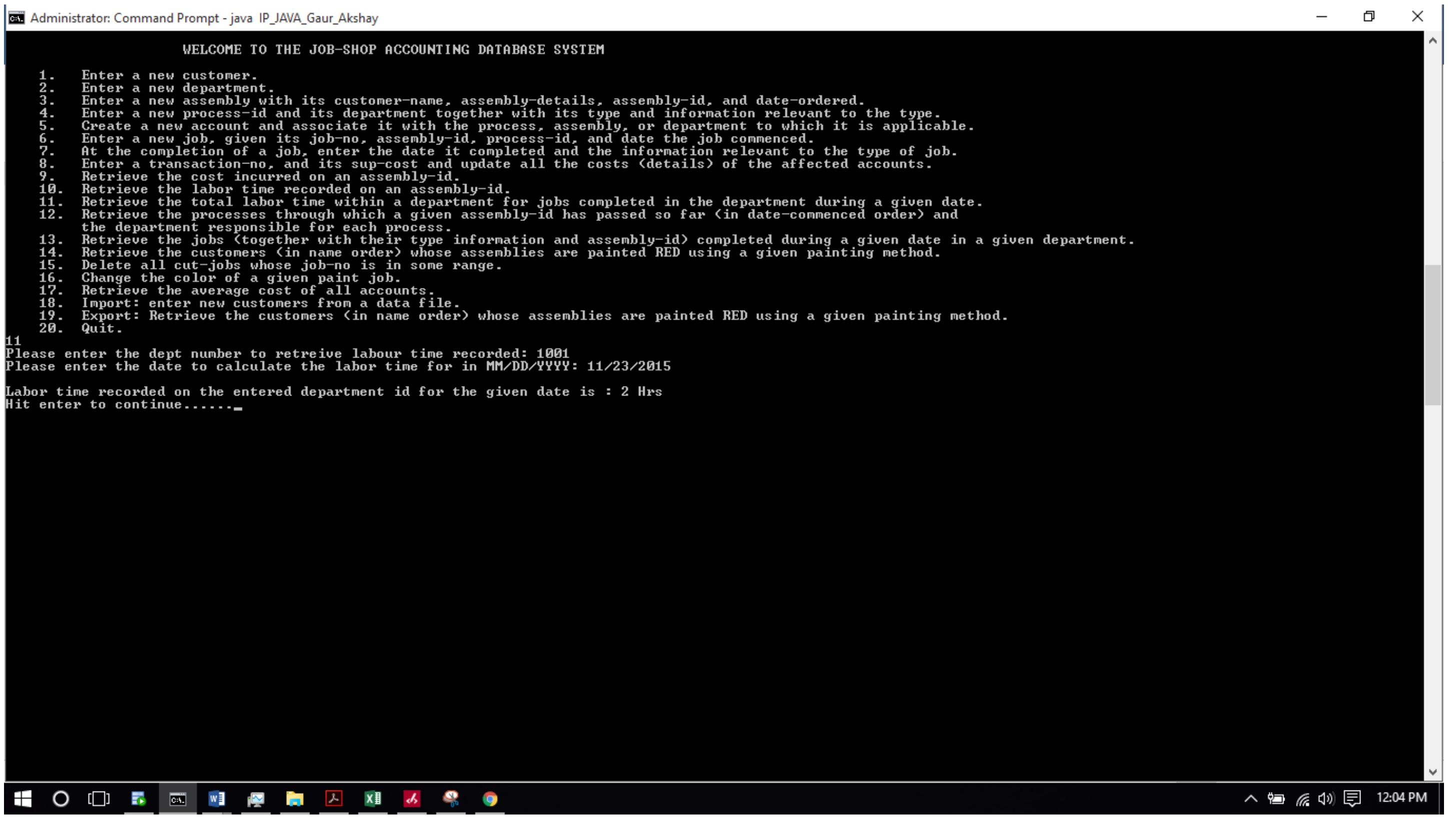
QUERY 10


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
10
Please enter the assembly id to retreive labour time recorded: 100001

Labor time recorded on the entered assembly id is : 2 Hrs
Hit enter to continue......
```

12:03 PM

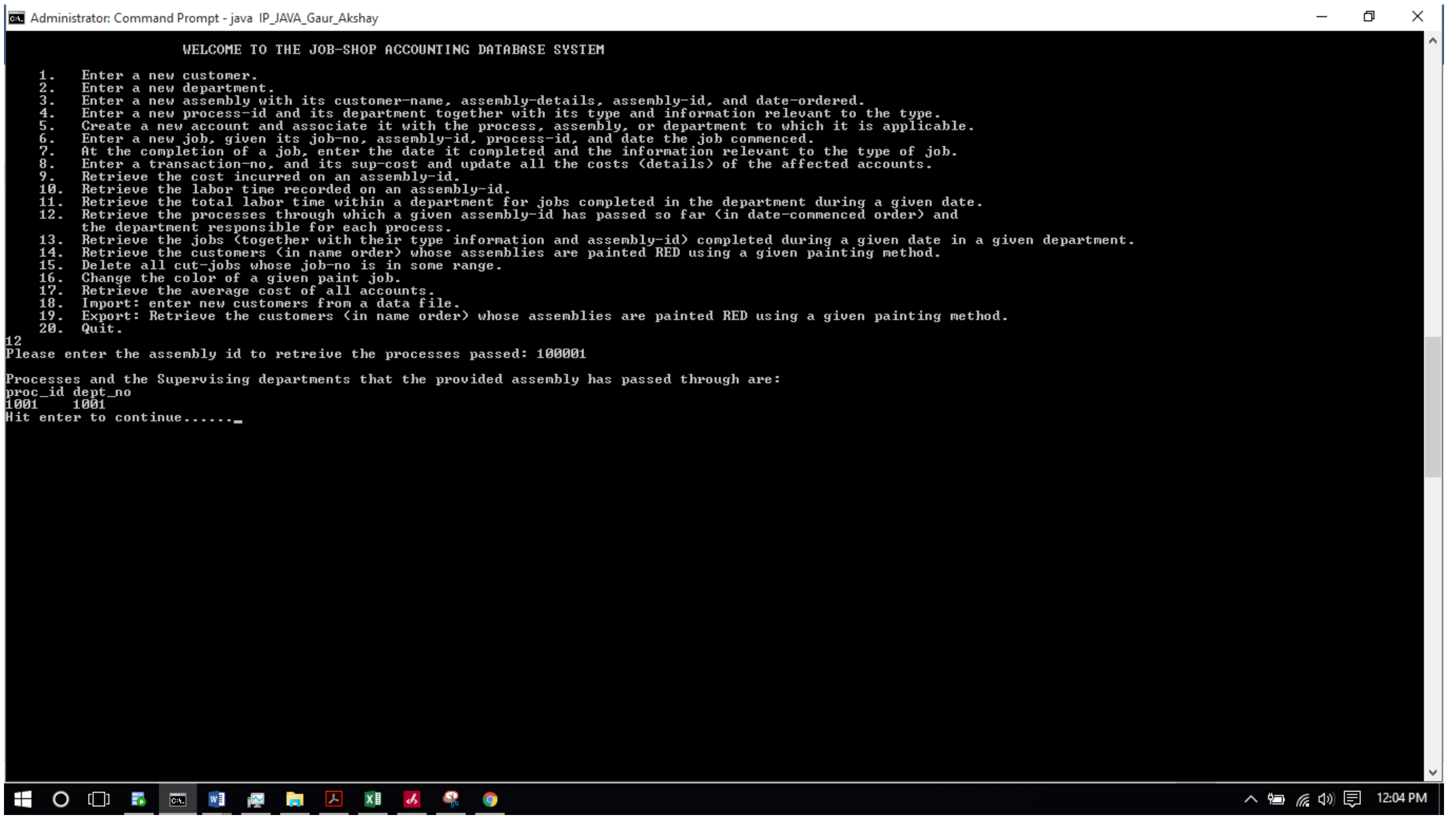
QUERY 11


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
11
Please enter the dept number to retreive labour time recorded: 1001
Please enter the date to calculate the labor time for in MM/DD/YYYY: 11/23/2015

Labor time recorded on the entered department id for the given date is : 2 Hrs
Hit enter to continue......_
```

12:04 PM

QUERY 12

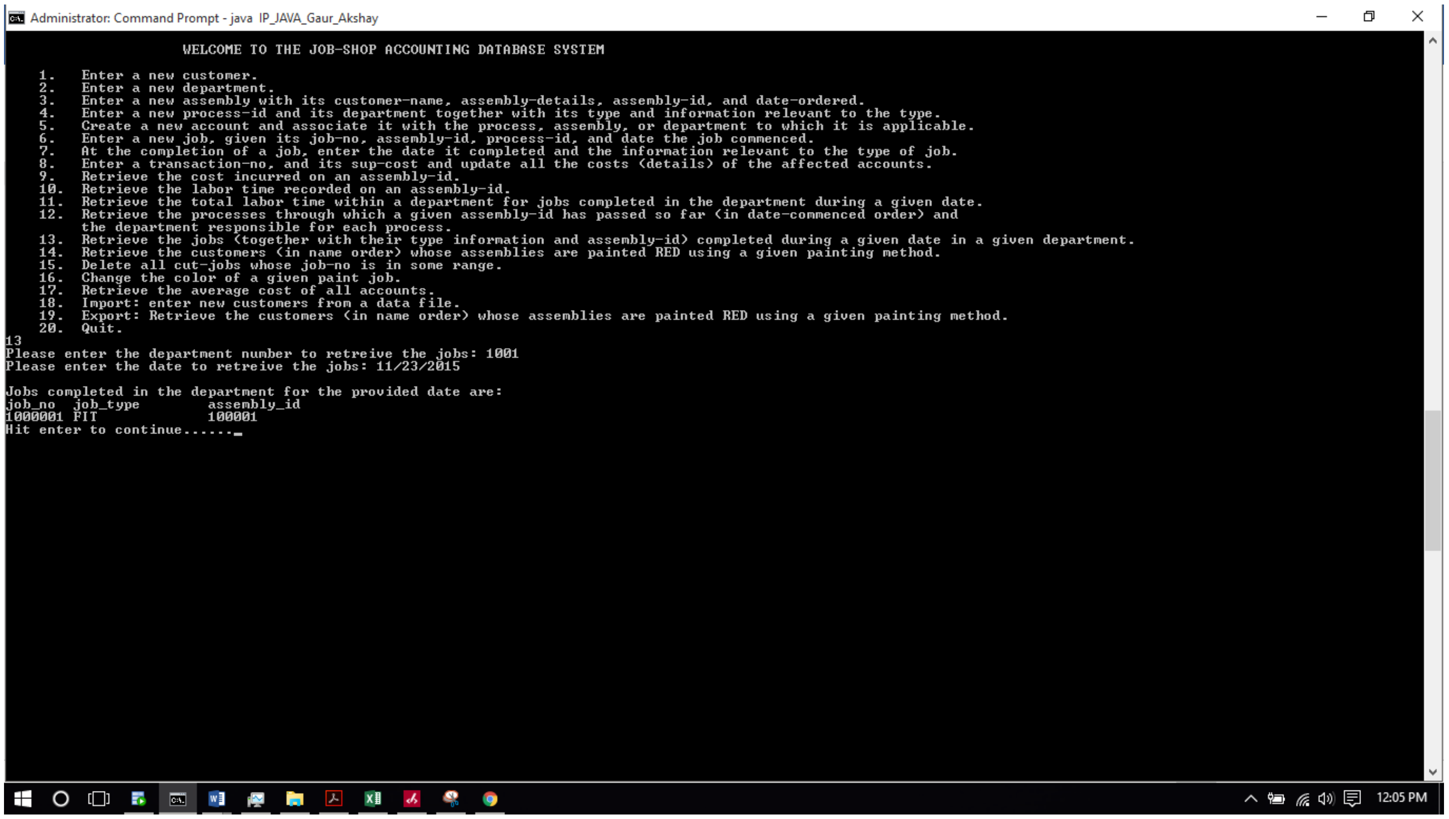


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
12
Please enter the assembly id to retreive the processes passed: 100001

Processes and the Supervising departments that the provided assembly has passed through are:
proc_id dept_no
1001    1001
Hit enter to continue......_
```

12:04 PM

QUERY 13



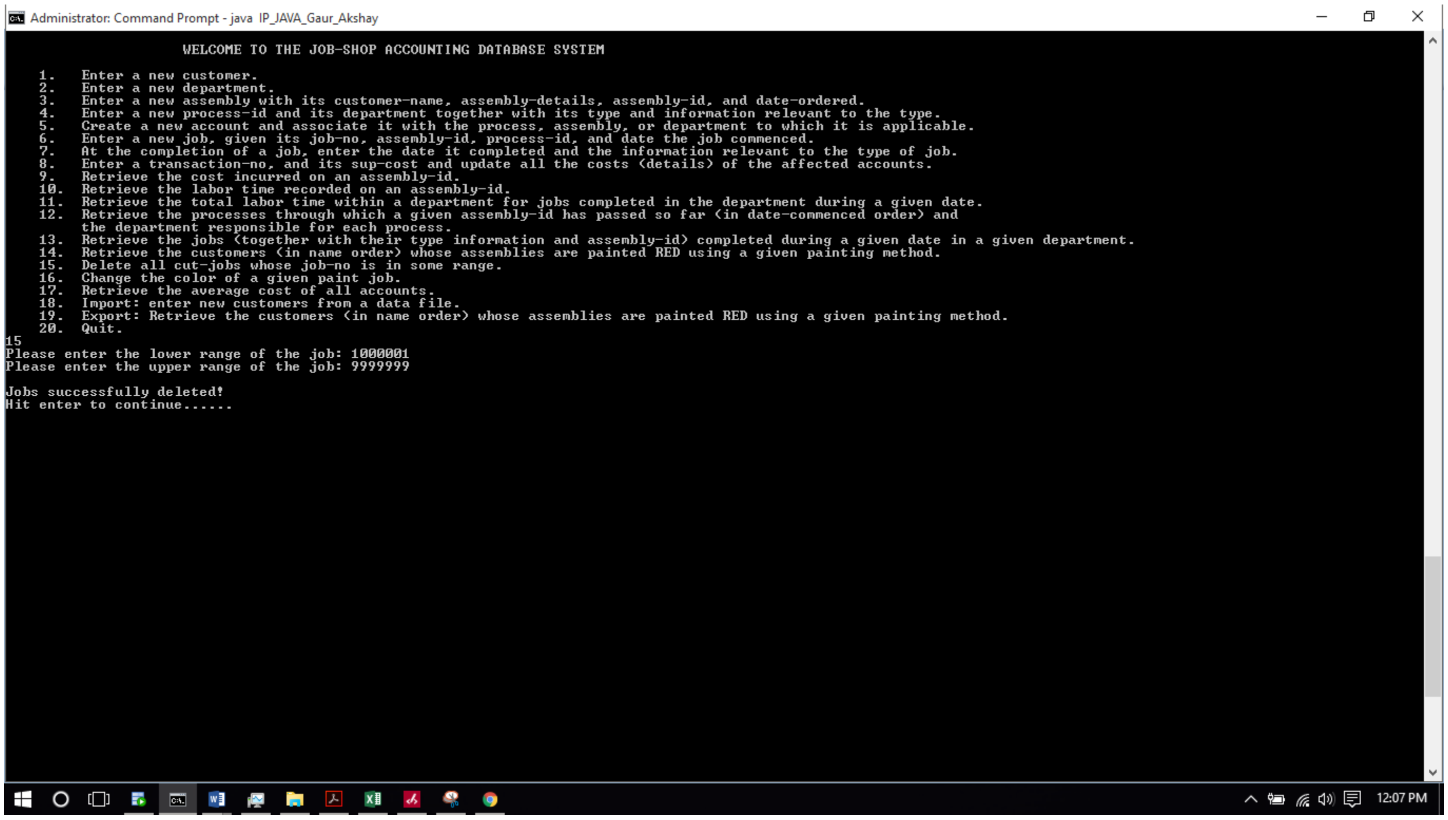
Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
13
Please enter the department number to retreive the jobs: 1001
Please enter the date to retreive the jobs: 11/23/2015

Jobs completed in the department for the provided date are:
job_no  job_type        assembly_id
1000001 FIT             100001
Hit enter to continue......_
```

12:05 PM

QUERY 14

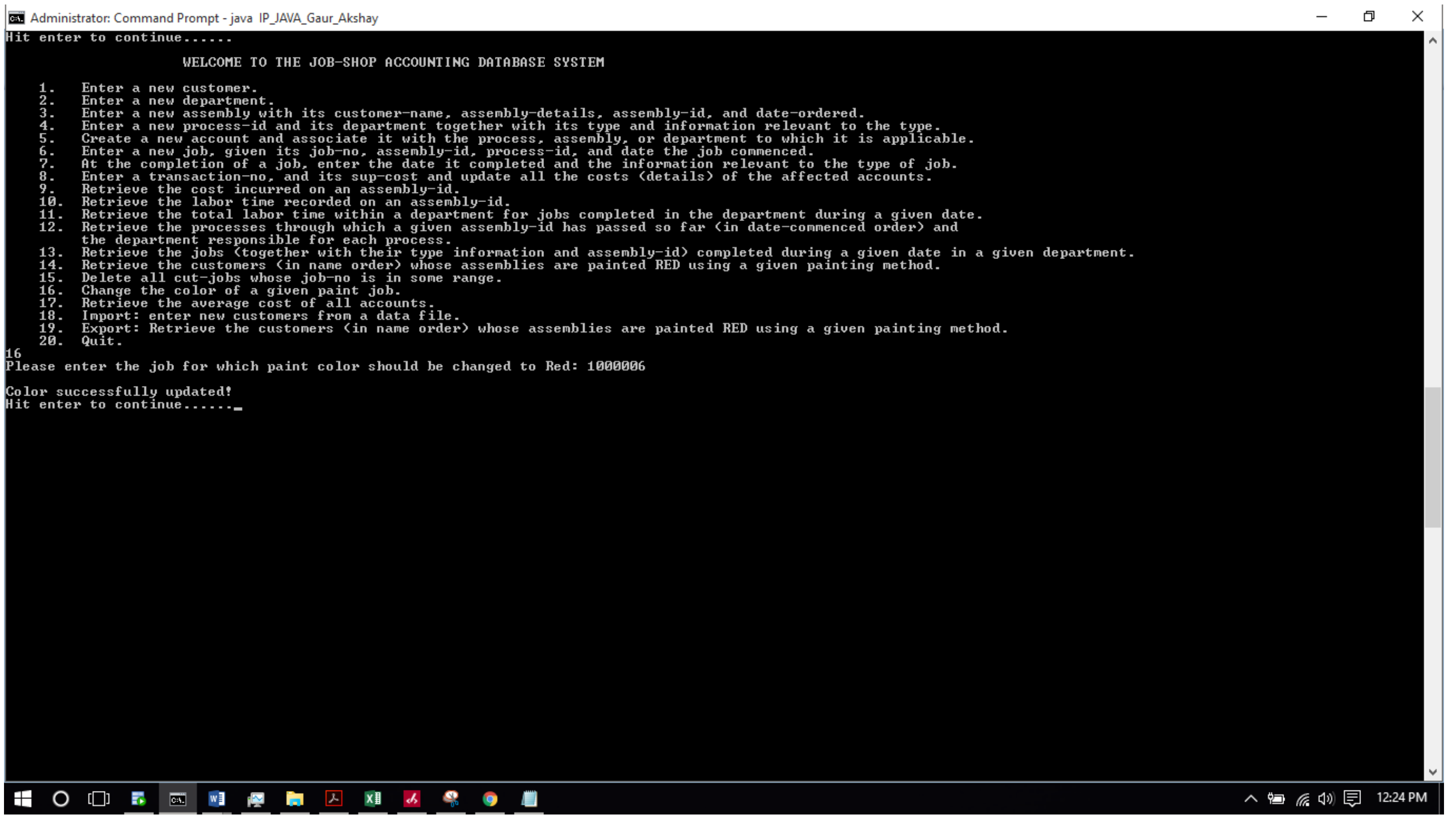
QUERY 15



Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
15
Please enter the lower range of the job: 1000001
Please enter the upper range of the job: 9999999

Jobs successfully deleted!
Hit enter to continue......
```

QUERY 16

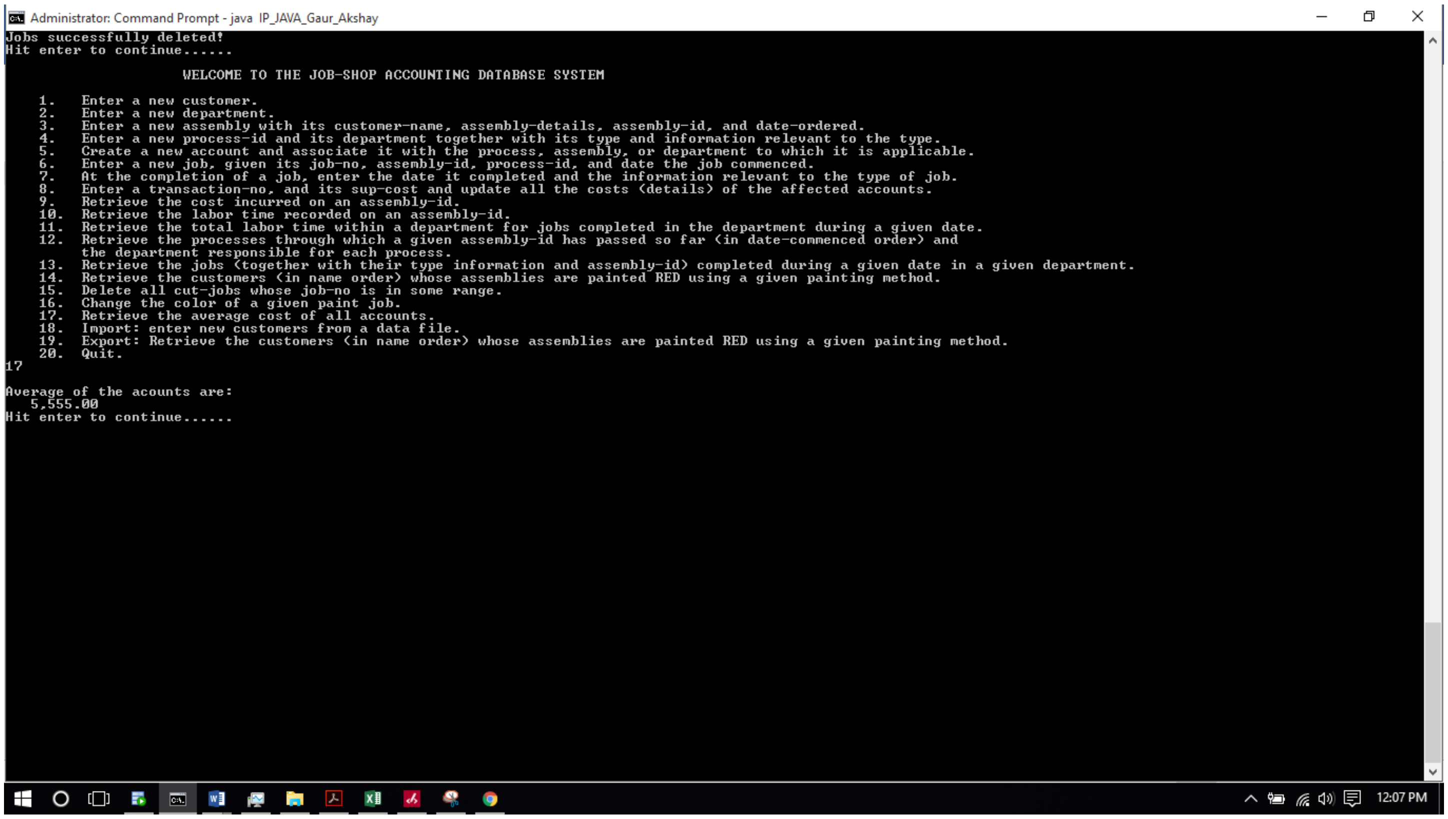


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
Hit enter to continue......

                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.    Enter a new customer.
    2.    Enter a new department.
    3.    Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.    Enter a new process-id and its department together with its type and information relevant to the type.
    5.    Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.    Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.    At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.    Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.    Retrieve the cost incurred on an assembly-id.
    10.   Retrieve the labor time recorded on an assembly-id.
    11.   Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.   Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
          the department responsible for each process.
    13.   Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.   Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.   Delete all cut-jobs whose job-no is in some range.
    16.   Change the color of a given paint job.
    17.   Retrieve the average cost of all accounts.
    18.   Import: enter new customers from a data file.
    19.   Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.   Quit.
16
Please enter the job for which paint color should be changed to Red: 1000006

Color successfully updated!
Hit enter to continue......_
```

12:24 PM

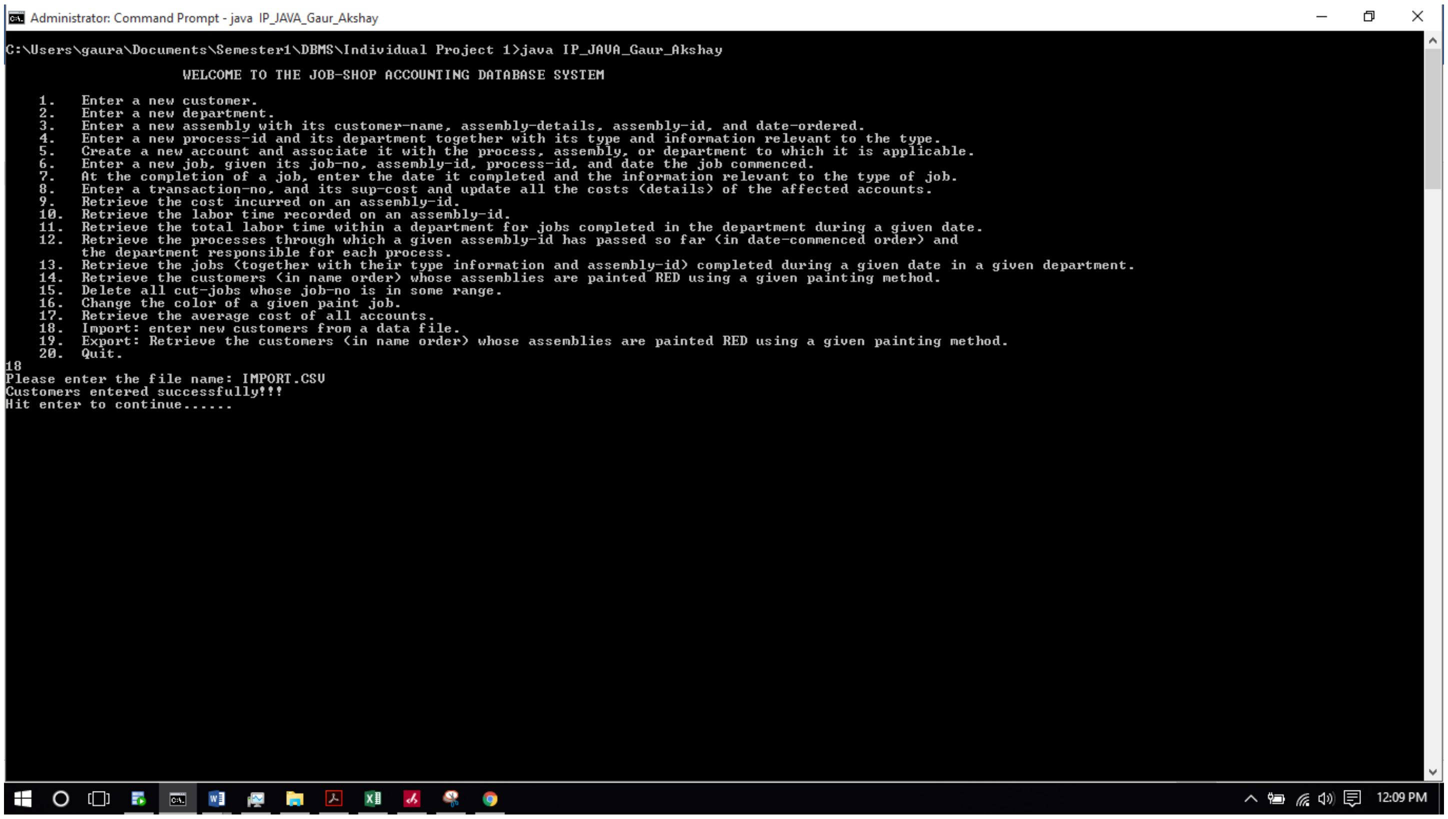
QUERY 17


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
Jobs successfully deleted!
Hit enter to continue......

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
17

Average of the acounts are:
   5,555.00
Hit enter to continue......
```

12:07 PM

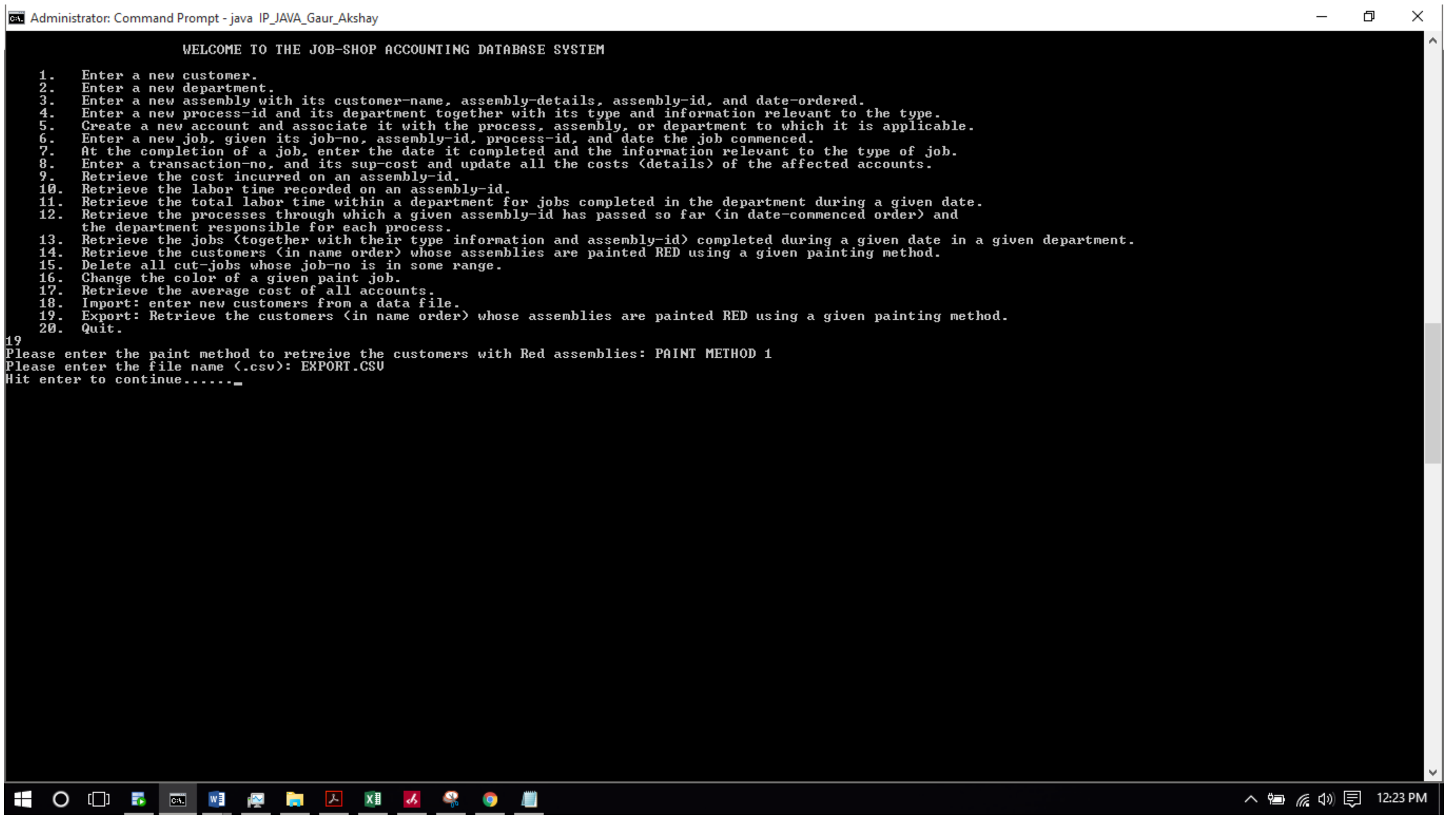
QUERY 18


Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

```
C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>java IP_JAVA_Gaur_Akshay

                     WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
18
Please enter the file name: IMPORT.CSV
Customers entered successfully!!!
Hit enter to continue......
```

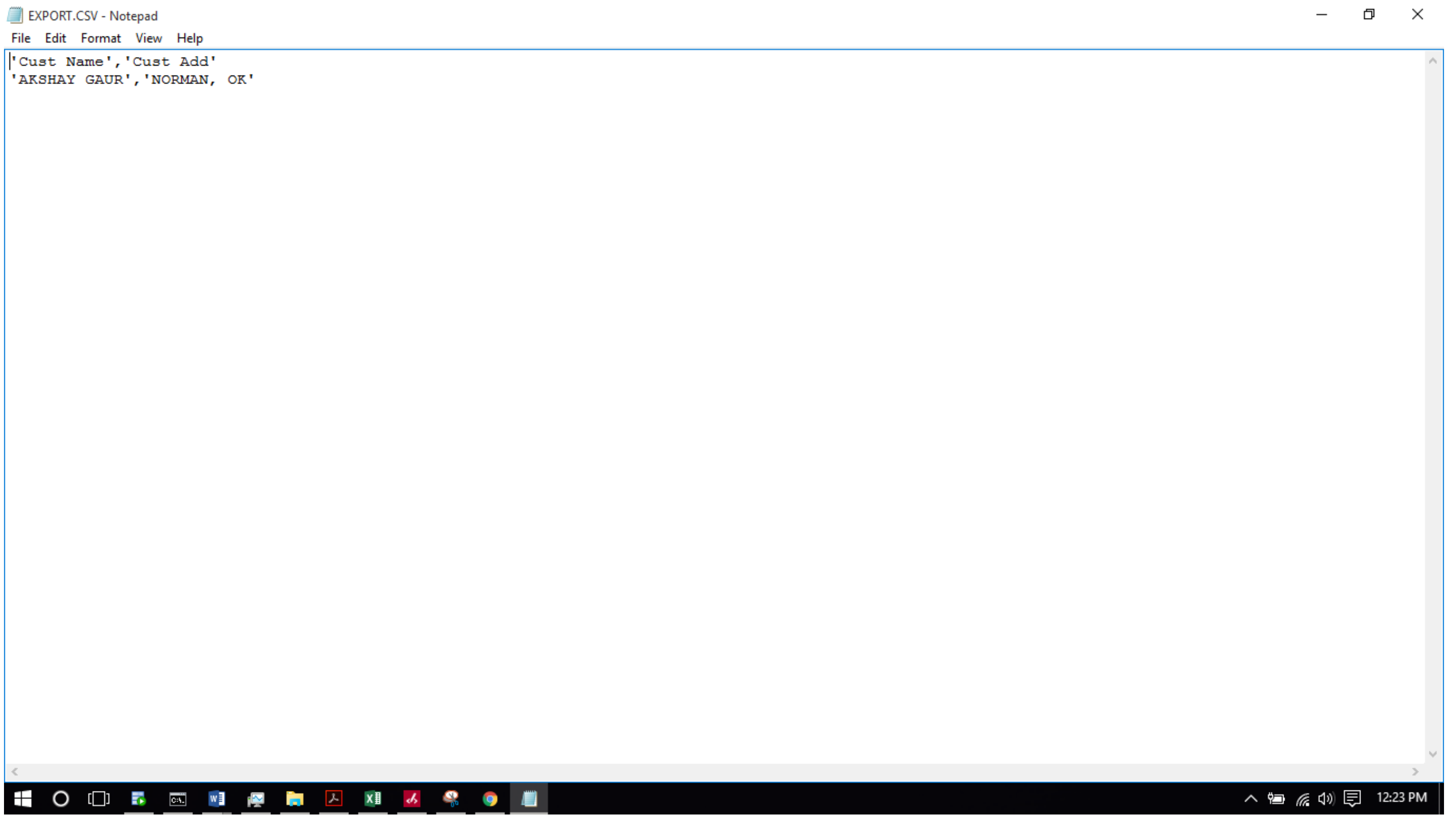
12:09 PM

OPTION 19



Administrator: Command Prompt - java IP_JAVA_Gaur_Akshay

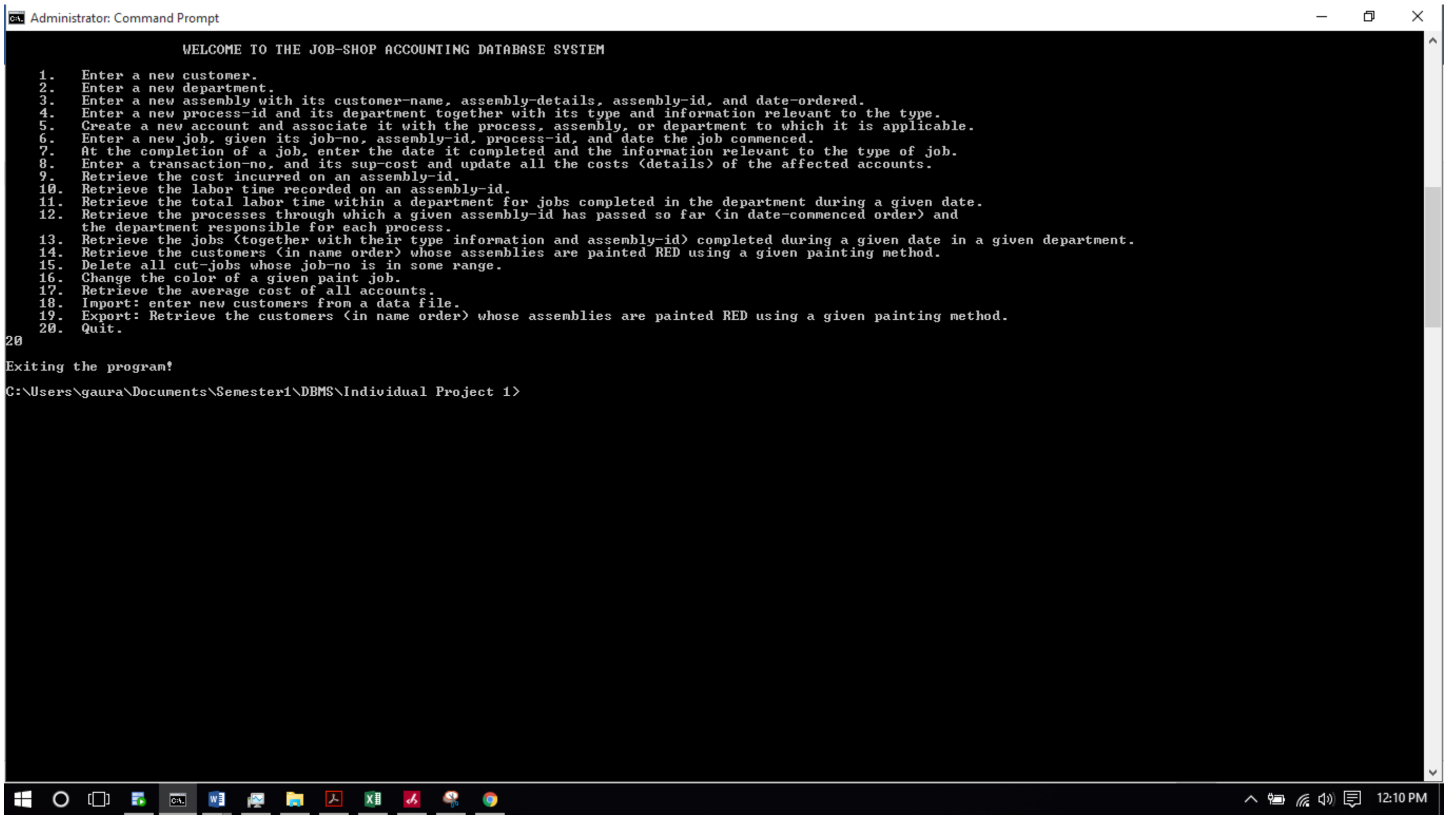
```
                    WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
19
Please enter the paint method to retreive the customers with Red assemblies: PAINT METHOD 1
Please enter the file name (.csv): EXPORT.CSV
Hit enter to continue......_
```



EXPORT.CSV - Notepad

```
'Cust Name','Cust Add'
'AKSHAY GAUR','NORMAN, OK'
```

OPTION 20



Administrator: Command Prompt

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

    1.   Enter a new customer.
    2.   Enter a new department.
    3.   Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered.
    4.   Enter a new process-id and its department together with its type and information relevant to the type.
    5.   Create a new account and associate it with the process, assembly, or department to which it is applicable.
    6.   Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced.
    7.   At the completion of a job, enter the date it completed and the information relevant to the type of job.
    8.   Enter a transaction-no, and its sup-cost and update all the costs (details) of the affected accounts.
    9.   Retrieve the cost incurred on an assembly-id.
    10.  Retrieve the labor time recorded on an assembly-id.
    11.  Retrieve the total labor time within a department for jobs completed in the department during a given date.
    12.  Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and
         the department responsible for each process.
    13.  Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given department.
    14.  Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    15.  Delete all cut-jobs whose job-no is in some range.
    16.  Change the color of a given paint job.
    17.  Retrieve the average cost of all accounts.
    18.  Import: enter new customers from a data file.
    19.  Export: Retrieve the customers (in name order) whose assemblies are painted RED using a given painting method.
    20.  Quit.
20

Exiting the program!

C:\Users\gaura\Documents\Semester1\DBMS\Individual Project 1>
```

12:10 PM