

Database Management System

CS-4513-001

Fall 2019

Dr. Le Gruenwald

Bao Le

113428176

[Baole@ou.edu](mailto:Baole@ou.edu)

Job-shop Accounting System

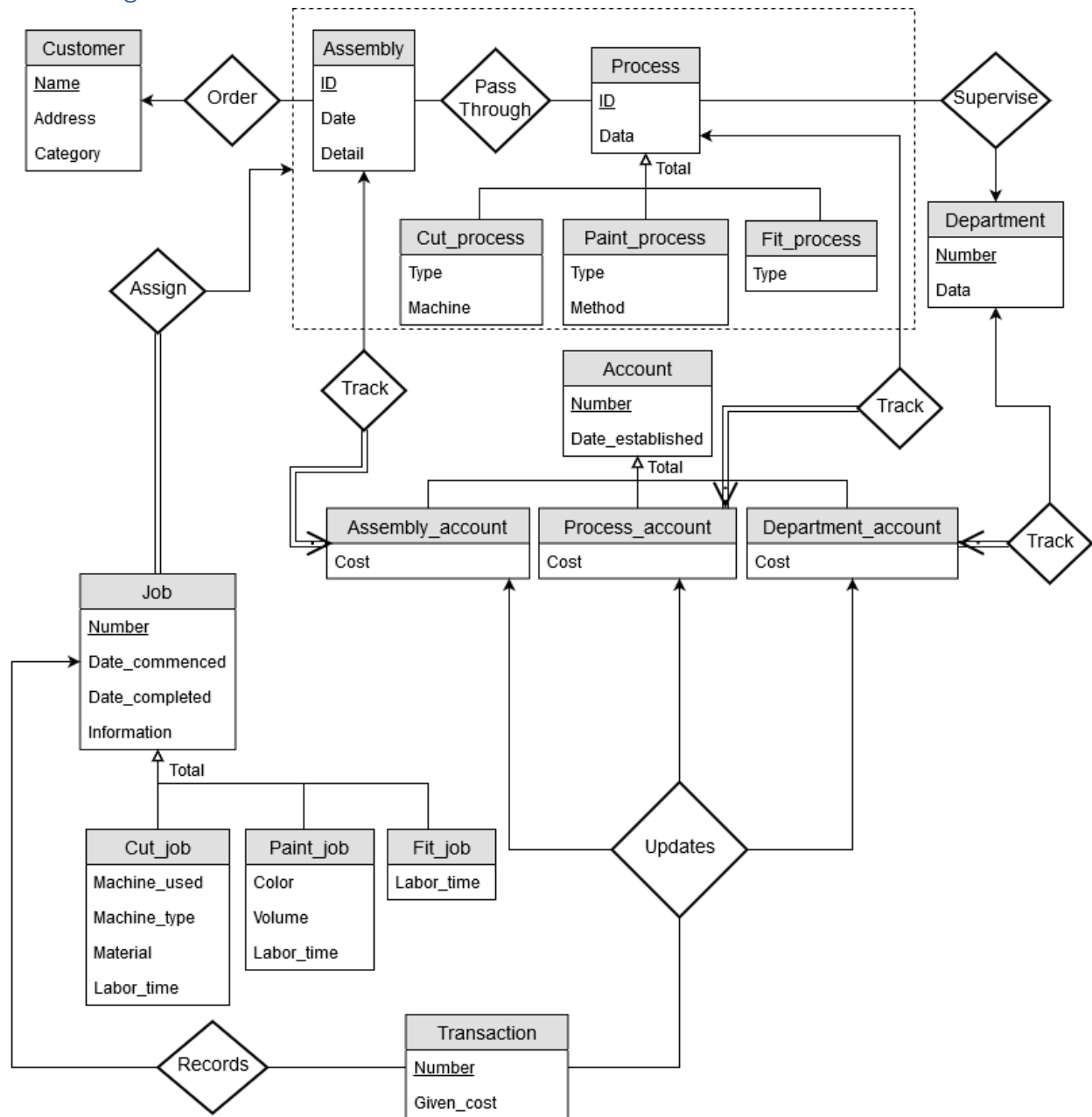
## Tasks Performed

Task 1 .....	4-6
1.1 ER Diagram .....	4-4
1.2 Relational Database Schema.....	4-5
1.3 Relational Database Schema Diagram .....	6-6
Task 2 Data Dictionary .....	7-12
Task 3 File Organization .....	12-18
3.1 Storage Structure .....	12-16
3.2 Storage Structure (Azure SQL Database) .....	16-18
Task 4 SQL Statements.....	18-37
4.1 Create Tables .....	18-27
4.2 Create Indexes .....	27-28
4.3 Create Queries .....	29-37
Task 5 Java Source Program and Compilation .....	37-50
Task 6 Java Program Execution .....	50-85
6.1 Query 1.....	50-51
6.2 Query 2.....	51-52
6.3 Query 3.....	52-55
6.4 Query 4.....	55-58
6.5 Query 5.....	58-62
6.6 Query 6.....	62-64
6.7 Query 7.....	65-68
6.8 Query 8.....	68-72
6.9 Query 9.....	72-72
6.10 Query 10.....	72-73
6.11 Query 11.....	73-73
6.12 Query 12.....	73-74
6.13 Query 13.....	74-75
6.14 Query 14.....	75-77
6.15 Query 15.....	77-78
6.16 Import Options.....	78-79
6.17 Export Options .....	79-81
6.18 Errors.....	81-85

6.19 Quit Option .....	85-85
Task 7 Web Application.....	85-94
7.1 Source Program.....	85-91
7.2 Successful Compilation .....	91-92
7.3 Web application testing.....	92-94

## Task 1

### 1.1 ER Diagram



### 1.2 Relational Database Schema

Customer(Name, Address, Category)

Assembly(ID, Date, Detail)

Order(Customer.name, Assembly.ID)

Process(ID, Data) // Data specify the type for the processes

Department(Number, Data)

Supervise(Process.id, Department.ID)

Pass\_through(Assembly.ID, Process.ID)

Cut\_process(Process.ID, Machine\_type)

Paint\_process(Process.ID, Method)

Fit\_process(Process.ID)

Job(Number, Date\_commenced, Date\_complete, Assembly.ID, Process.ID, Information)

Cut\_job(Job.number, Machine\_type, Machine\_time, Material, Labor\_time)

Paint\_job(Job.number, Color, Volume, Labor\_time)

Fit\_job(Job.number, Labor\_time)

Assembly\_account(Number, Date\_established, Cost, Assembly.ID)

Process\_account(Number, Date\_established, Cost, Process.ID)

Department\_account(Number, Date\_established, Cost, Department.number)

Transaction(Number, Sup-cost)

Record(Transaction.number, Job.number)

Updates (Transaction.number, Assembly\_account.number, Process\_account.number, Department\_account.number)

Assembly\_track(Assembly.ID, Assembly\_account.number)

Process\_track(Process.ID, Process\_account.number)

Department\_track(Department.number, Department\_account.number)



## Task 2 Data Dictionary

Table: Customer			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Name	Varchar	42	PK
Address	Varchar	42	Not null, can be blank though (traveler)
Category	Int	4	Not null, integer range from 1 to 10

Table: Assembly			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
ID	Int	4	PK
Date	Date	3	YYYY-MM-DD format, must be valid date
Detail	Varchar	42	Can be anything really, but preferably something useful

Table: Order			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Assembly.ID	Int	4	PK, FK references ID from assembly table
Customer.name	Varchar	42	FK references name from customer table

Table: Process			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
ID	Int	4	PK
Data	Varchar	7	Either Paint, Cut, or Fit

Table: Department			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK

Data	Varchar	42	Can be anything really, but preferably what process it's supervising
------	---------	----	--

Table: Supervise			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Process.ID	Int	4	PK, FK references ID from process table
Department.number	Int	4	FK references number from department table

Table: Pass_through			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Assembly.ID	Int	4	PK, FK references ID in assembly table
Process.ID	int	4	PK, FK references ID in process table

Table: Cut_process			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Process.ID	Int	4	PK, FK references ID in process table
Machine_type	Varchar	42	Machine type, can be empty

Table: Paint_process			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Process.ID	Int	4	PK, FK references ID in process table
Method	Varchar	42	Method type, can be empty



Table: Fit_process			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Process.ID	Int	4	PK, FK references ID in process table

Table: Job			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK
Date_commenced	Date	3	Not null, date must be correct format as well
Date_completed	Date	3	Can be null until completion, format must be correct
Assembly.ID	Int	4	FK references ID in assembly table
Process.ID	Int	4	FK references ID in process table
Information	Varchar	7	Can be Paint, Cut, or Fit

Table: Cut_job			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Job.number	Int	4	PK, FK references number in job table
Machine_type	Varchar	42	FK, should be references machine type in cut_process table, and not null
Machine_time	Int	4	Will be in minutes, cannot be negative
Material	Varchar	42	Can be empty, aka empty
Labor_time	Int	4	Will be in minutes, cannot be negative

Table: Paint_job			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Job.number	Int	4	PK,

			FK references number in job table
Color	Varchar	42	Can be empty, no color
Volume	Int	4	Cannot be negative, measure in Liter
Labor_time	int	4	Cannot be negative

Table: Fit_job			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Job.number	Int	4	PK, FK references number in job table
Labor_time	Int	4	Cannot be negative

Table: Assembly_account			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK
Date_established	Date	3	Not null, must be correct date and format
Cost	Float	4	Cannot be negative
Assembly.ID	Int	4	FK references ID in assembly table

Table: Process_account			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK
Date_established	Date	3	Not null, date must be correct and in format
Cost	Float	4	Cannot be negative
Process.ID	Int	4	FK references ID in process table

Table: Department_account			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK
Date_established	Date	3	Not null, format and date must be valid
Cost	Float	4	Cannot be negative
Department.number	Int	4	FK references number in department table

Table: Transaction			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Number	Int	4	PK
Sup-cost	Float	4	Cannot be negative

Table: Record			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Transaction.number	Int	4	PK, FK references number in transaction table
Job.number	Int	4	FK references number in job table

Table: Updates			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Transaction.number	Int	4	PK, FK references number in transaction table
Assembly_account.number	Int	4	FK references number in assembly_account table
Process_account.number	Int	4	FK references number in process_account table
Department_account.number	int	4	FK references number in process_account table

Table: Assembly_track			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Assembly.ID	Int	4	PK, FK references ID in assembly table
Assembly_account.number	Int	4	FK references number in Assembly_account table

Table: Department_track			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Department.number	Int	4	PK, FK references number in department table
Department_account.number	Int	4	FK references number in department_account table

Table: Process_track			
Attributes			
Name	Type	Size (bytes)	Constraint (if any)
Process.ID	Int	4	PK, FK references ID in process table
Process_account.number	Int	4	FK references number in process_account table

## Task 3 File Organization

### 3.1 Storage Structure

Table Name	Query# And Type	Search Key	Query Frequency	Select File Organization	Justification
Customer	#1, insertion		30/day	B+ tree on search key category	Category is more frequently used than name
	#3, random search	name	40/day		

	#13, range search	category	100/day		
Assembly	#3, insertio n			Extendable Hashing on search key ID	Majority of the query requires random search for some assembly ID
	#5, random search	ID	40/day		
	#6, random search	ID	50/day		
	#9, random search	ID	200/day		
Order	#3, insertio n		40/day	Heap File	Only used to show which customer order what assemblies
Process	#4, insertio n		infrequen t	Extendable Hashing on search key ID	Similar to Assembly, most of the query requires random search for process ID, and query 7 is only use if the job is a cut type
	#5, random search	ID	10/day		
	#6, random search	ID	50/day		
	#7, random search	ID	50/day		
	#10, random search	ID	20/day		
Department	#2, insertio n		30/day	Extendable Hashing on search key Number	Like assembly and process, most of the search requires us to find department number within the table
	#4, random search	Number	Infrequen t		
	#5, random search	Number	10/day		
	#10, random search	Number	20/day		

	#12, random search	Number	20/day		
Supervise	#4, insertio n		infrequen t	Extendable Hashing on search key Process.ID	Frequency on searching with process.ID is more frequent than the other 3 searches combined
	#8, random search	Department.numb er	50/day		
	#10, random search	Department.numb er	20/day		
	#11, random search	Process.ID	100/day		
	#12, random search	Department.numb er	20/day		
Pass_through	#6, insertio n		50/day	Heap File	Only insertion is used, query 11 can just use search on job table, as a job is assign every time an assembly pass through a process
Cut_process	#4, insertio n		infrequen t	Extendable Hashing on search key Machine_typ e	Used for cut_job machine type
	#7, random search	Machine_type	50/day		
Paint_process	#4, insertio n		infrequen t	Heap file	Only insertion
Fit_process	#4, insertio n		infrequen t	Heap file	Only insertion
Job	#6, insertio n		50/day	Extendable Hashing on search key Number	Majority of the query requires random search, and the only range search is used only about
	#7, insertio n		50/day		

	#8, random search	Number	50/day		1 time a month and query 10 only occurs fewer time a day than the other query combined
	#10, random search	Process.ID, date_complete	20/day		
	#11, random search	Number	100/day		
	#12, random search	Number	20/day		
	#14, range search	Number	1/month		
	#15, random search	Number	1/week		
Cut_job	#7, insertio n		50/day	Extendable Hashing on search key Job.Number	The most frequently use query requires random searching Job.number
	#10, random search	Job.number	20/day		
	#14, range search	Job.number	1/month		
Paint_job	#7, insertio n		50/day	Extendable Hashing on search key Job.Number	both queries require Job.number for random search
	#10, random search	Job.number	20/day		
	#15, random search	Job.number	1/week		
Fit_job	#7, insertio n		50/day	Extendable Hashing on search key Job.Number	Similar to paint and paint_job, the queries only need job.number for random search
	#10, random search	Job.number	20/day		
Assembly_account	#5, insertio n		10/day	Extendable Hashing on	Queries only requires random search

	#5, #8, random search	Number	50/day	search key Number	
	#9, random search	Number	200/day		
Process_account	#5, insertio n		10/day	Extendable Hashing on search key Number	Queries only requires random search
	#5, #8, random search	Number	50/day		
Department_accou nt	#5, insertio n		10.day	Extendable Hashing on search key Number	Queries only requires random search
	#5, #8, random search	Number	50/day		
Transaction	#8, insertio n		50/day	Heap file	Only insertion
Record	#8, insertio n		50/day	Heap file	Only insertion
Updates	#8, insertio n		50/day	Heap file	Only insertion
Assembly_track	#5, insertio n		10/day	Heap File	Query 9 can just go straight to assembly_accou nt for assembly ID
Process_track	#5, insertio n		10/day	Heap file	Only insertion
Department_track	#5, insertio n		10/day	Heap file	Only insertion

### 3.2 Storage Structure (Azure SQL Database)

Using our knowledge of the table structure from 3.1, we can now construct a table for the storage structure within the SQL database

Table Name	Storage Structure	Justification
Customer	Non-clustered index on Category	Category is more frequently uses and it is used for a range



	Non-clustered index on Name	search, so a clustered index would be better but since azure SQL doesn't let u alter clustered index for primary key, we will have to settle with non-clustered
Assembly	Non-Clustered index on ID	Random search is used for majority of the query and non-clustered is faster on random search
Order	None	Heap file, no searches
Process	Non-clustered index on ID	Like assembly, majority of the queries requires random search on process.
Department	Non-clustered index on Number	Like assembly and process, most of the queries uses random search on Number, and for a single search non-clustered is faster
Supervise	Non-clustered index on process.ID Non-clustered index on assembly.ID	Both content within supervise requires frequent random searches, so non-clustered for both is ideal
Pass_through	None	Heap file, only insertion
Cut_process	Non-clustered index on Machine_type	Only Machine_type attribute is used for random search for cut job
Paint_process	None	Heap file, only insertion
Fit_process	None	Heap file, only insertion
Job	Non-clustered index on Number	While there is a range search, it is only 1/month which is very infrequent, so a non-clustered index on number will allow all the other query that uses random search to be fast
Cut_job	Non-clustered index on Job.number	Only uses random search on attribute Job.number
Paint_job	Non-clustered index on Job.number	Only uses random search on attribute Job.number
Fit_job	Non-clustered index on Job.number	Only uses random search on attribute Job.number
Assembly_account	Non-clustered index on Number	Only uses random search on attribute Number
Process_account	Non-clustered index on Number	Only uses random search on attribute Number
Department_account	Non-clustered index on Number	Only uses random search on attribute Number

Transaction	None	Heap file, only insertion
Record	None	Heap file, only insertion
Updates	None	Heap file, only insertion
Assembly_track	None	Heap file, only insertion
Process_track	None	Heap file, only insertion
Department_track	None	Heap file, only insertion

## Task 4 SQL Statements

### 4.1 Create Tables

Customer:

```
create table customer (
  cname varchar(40) primary key,
  caddress varchar(40) not null,
  category int not null

  constraint category_range_check
  check(category >= 1 and category <=10)
)
```

	cname	caddress	category
1	NULL	NULL	NULL

Assembly:

```
create table assemblies (
  aid int primary key,
  date_order DATE not null,
  detail varchar(40)
)
```

	aid	date_order	detail
1	NULL	NULL	NULL

Order:

```

create table orders (
  cname varchar(40) not null,
  aid int primary key

  constraint FK_customer_orders
  foreign key (cname) references customer(cname),

  constraint FK_assembly_orders
  foreign key (aid) references assemblies(aid)
)

```

	cname	aid
1	NULL	NULL

Process:

```

create table process (
  pid int primary key,
  pdata varchar(5)

  constraint process_type_check
  check(pdata = 'Paint' OR pdata = 'Fit' OR pdata = 'Cut')
)

```

	pid	pdata
1	NULL	NULL

Department:

```

create table department(
  dnum int primary key,
  ddata varchar(40)
)

```

	dnum	ddata
1	NULL	NULL

Supervise:

```

create table supervise (
  pid int primary key,
  dnum int not null

  constraint FK_process_supervise
  foreign key (pid) references process(pid),

  constraint FK_department_supervise
  foreign key (dnum) references department(dnum)
)

```

	cname	aid
1	NULL	NULL

Fit\_process:

```

create table fit_process (
  pid int primary key,
  fit_type varchar(40),

  constraint FK_process_fit_process
  foreign key (pid) references process(pid)
)

```

	pid	fit_type
1	NULL	NULL

Cut\_process:

```

create table cut_process (
  pid int primary key,
  cut_type varchar(40),
  machine_type varchar(40)

  constraint FK_process_cut_process
  foreign key (pid) references process(pid)
)

```

	pid	cut_type	machine_type
1	NULL	NULL	NULL

Paint\_process:

```

create table paint_process (
  pid int primary key,
  paint_type varchar(40),
  method varchar(40)

  constraint FK_process_paint_process
  foreign key (pid) references process(pid)
)

```

	pid	paint_type	method
1	NULL	NULL	NULL

Pass\_through:

```

create table pass_through (
  aid int not null,
  pid int not null
  primary key (aid, pid)

  constraint FK_assembly_pass_through
  foreign key (aid) references assemblies(aid),

  constraint FK_process_pass_through
  foreign key (pid) references process(pid)
)

```

	aid	pid
1	NULL	NULL

Job:

```

create table job (
  jnum int primary key,
  date_commence date not null,
  date_complete date,
  aid int not null,
  pid int not null,
  info varchar(5)

  constraint FK_assemblies_job
  foreign key (aid) references assemblies(aid),

  constraint FK_process_job
  foreign key (pid) references process(pid),

  constraint job_type_check
  check(info = 'Paint' OR info = 'Fit' OR info = 'Cut')
)

```

	jnum	date_commence	date_complete	aid	pid	info
1	NULL	NULL	NULL	NULL	NULL	NULL

Cut\_job:

```
create table cut_job (
  jnum int primary key,
  machine_type varchar(40) not null,
  machine_time int,
  material varchar(40),
  labor_time int

  constraint FK_job_cut_job
  foreign key (jnum) references job(jnum),

  constraint cut_job_time_range_check
  check(labor_time >= 0 and machine_time >= 0)
)
```

	jnum	machine_type	machine_time	material	labor_time
1	NULL	NULL	NULL	NULL	NULL

Paint\_job:

```
create table paint_job (
  jnum int primary key,
  color varchar(40),
  volume int,
  labor_time int

  constraint FK_job_paint_job
  foreign key (jnum) references job(jnum),

  constraint paint_job_time_range_check
  check(labor_time >= 0 and volume >= 0)
)
```

	jnum	color	volume	labor_time
1	NULL	NULL	NULL	NULL

Fit\_job:

```

create table fit_job (
  jnum int primary key,
  labor_time int

  constraint FK_job_fit_job
  foreign key (jnum) references job(jnum),

  constraint fit_job_time_range_check
  check(labor_time >= 0)
)

```

	jnum	labor_time
1	NULL	NULL

Assembly\_account:

```

create table assembly_account (
  aanum int primary key,
  date_established date not null,
  cost float,
  aid int not null

  constraint FK_assemblies_assembly_account
  foreign key (aid) references assemblies(aid),

  constraint assembly_account_cost_range_check
  check(cost >= 0)
)

```

	aanum	date_establish_	cost	aid
1	NULL	NULL	NULL	NULL

Process\_account:

```

create table process_account (
  panum int primary key,
  date_established date not null,
  cost float,
  pid int not null

  constraint FK_process_process_account
  foreign key (pid) references process(pid),

  constraint process_account_cost_range_check
  check(cost >= 0)
)

```

	panum	date_establish...	cost	pid
1	NULL	NULL	NULL	NULL

Department\_accouunt:

```
create table department_account (
  danum int primary key,
  date_established date not null,
  cost float,
  dnum int not null

  constraint FK_department_department_account
  foreign key (dnum) references department(dnum),

  constraint department_account_cost_range_check
  check(cost >= 0)
)
```

	danum	date_establish...	cost	dnum
1	NULL	NULL	NULL	NULL

Transactions:

```
create table transactions (
  tnum int primary key,
  sup_cost float not null,

  constraint transactions_cost_check
  check(sup_cost >= 0)
)
```

	tnum	sup_cost
1	NULL	NULL

Updates:



```

create table updates (
  tnum int primary key,
  aanum int not null,
  panum int not null,
  danum int not null

  constraint FK_transactions_updates
  foreign key (tnum) references transactions(tnum),

  constraint FK_assembly_account_updates
  foreign key (aanum) references assembly_account(aanum),

  constraint FK_process_account_updates
  foreign key (panum) references process_account(panum),

  constraint FK_department_account_updates
  foreign key (danum) references department_account(danum)
)

```

	tnum	aanum	panum	danum
1	NULL	NULL	NULL	NULL

Record:

```

create table record(
  tnum int primary key,
  jnum int not null

  constraint FK_transactions_record
  foreign key (tnum) references transactions(tnum),

  constraint FK_job_record
  foreign key (jnum) references job(jnum)
)

```

	tnum	jnum
1	NULL	NULL

Assembly\_track:

```

create table assembly_track (
  aid int primary key,
  aanum int not null

  constraint FK_assemblies_assembly_track
  foreign key (aid) references assemblies(aid),

  constraint FK_assembly_account_assembly_track
  foreign key (aanum) references assembly_account(aanum)
)

```

	aid	aanum
1	NULL	NULL

Process\_track:

```

create table process_track (
  pid int primary key,
  panum int not null

  constraint FK_process_process_track
  foreign key (pid) references process(pid),

  constraint FK_process_account_process_track
  foreign key (panum) references process_account(panum)
)

```

	pid	panum
1	NULL	NULL

Department\_track:

```

create table department_track (
  dnum int primary key,
  danum int not null

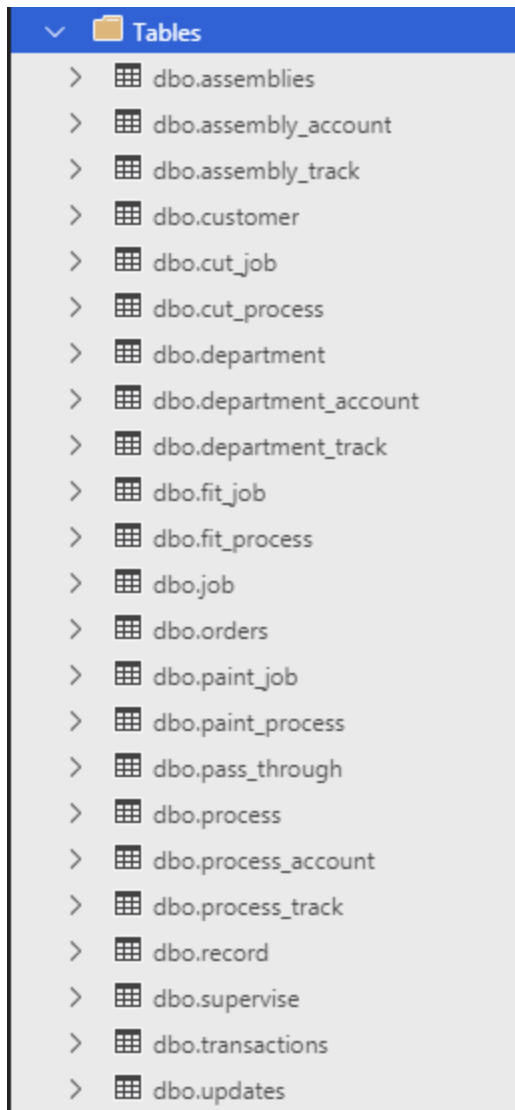
  constraint FK_department_department_track
  foreign key (dnum) references department(dnum),

  constraint FK_department_account_department_track
  foreign key (danum) references department_account(danum)
)

```

	dnum	danum
1	NULL	NULL

All table:



## 4.2 Create Indexes

Customer index:

```
create NONCLUSTERED INDEX cname
on customer (cname)
create NONCLUSTERED INDEX category
on customer (category)
```

Assembly index:

```
create NONCLUSTERED INDEX aid
on assemblies (aid)
```

Process index:

```
create NONCLUSTERED INDEX pid
on process (pid)
```

Department index:

```
create NONCLUSTERED INDEX dnum  
on department(dnum)
```

Supervise index:

```
create NONCLUSTERED INDEX pid  
on supervise(pid)  
create NONCLUSTERED INDEX dnum  
on supervise(dnum)
```

Cut\_process index:

```
create NONCLUSTERED INDEX machine_type  
on cut_process (machine_type)
```

Job index:

```
create NONCLUSTERED INDEX jnum  
on job (jnum)
```

Cut\_job index:

```
create NONCLUSTERED INDEX jnum  
on cut_job(jnum)
```

Paint\_job index:

```
create NONCLUSTERED INDEX jnum  
on paint_job(jnum)
```

Fit\_job index:

```
create NONCLUSTERED INDEX jnum  
on fit_job(jnum)
```

Assembly\_account index:

```
create NONCLUSTERED INDEX aanum  
on assembly_account(aanum)
```

Process\_account index:

```
create NONCLUSTERED INDEX panum  
on process_account(panum)
```

Department\_account index:

```
create NONCLUSTERED INDEX danum  
on department_account(danum)
```

### 4.3 Create Queries

*Create as procedures...*

Query 1:

```
GO
CREATE PROCEDURE insert_customer
|   @cname VARCHAR(40),
|   @caddress VARCHAR(40),
|   @category INT
AS
BEGIN
|   INSERT INTO customer(
|       cname, address, category
|   ) values
|       (@cname, @caddress, @category)
END
```

Query 2:

```
GO
CREATE PROCEDURE insert_department
|   @dnum INT,
|   @ddata VARCHAR(40)
AS
BEGIN
|   INSERT INTO department(
|       dnum, ddata
|   ) values
|       (@dnum, @ddata)
END
```

Query 3:

```
GO
CREATE PROCEDURE insert_assembly
    @aid INT,
    @date_order VARCHAR(40),
    @detail VARCHAR(40),
    @cname VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM customer WHERE cname = @cname)
        THROW 50000, 'Customer does not exist',1
    ELSE
        BEGIN TRY
            INSERT INTO assemblies(
                aid, date_order, detail
            ) values
                (@aid, @date_order, @detail)
            INSERT INTO orders(
                aid, cname
            ) values
                (@aid, @cname)
        END TRY
        BEGIN CATCH
            THROW 50000, 'Assembly ID already exist', 1
        END CATCH
END
```

Query 4:

```

GO
CREATE PROCEDURE insert_process
    @pid INT,
    @pdata VARCHAR(40),
    @dnum INT,
    @info VARCHAR(40)
AS
BEGIN
    -- check if department exists
    IF NOT EXISTS (SELECT 1 FROM department WHERE dnum = @dnum)
        THROW 50001, 'Department number not found', 1
    ELSE
        BEGIN TRY
            IF @pdata = '1'
            BEGIN
                INSERT INTO process (pid, pdata) values (@pid, 'Cut')
                INSERT INTO supervise (pid, dnum) values (@pid, @dnum)
                INSERT INTO cut_process (pid, machine_type) values (@pid, @info)
            END
            IF @pdata = '2'
            BEGIN
                INSERT INTO process (pid, pdata) values (@pid, 'Paint')
                INSERT INTO supervise (pid, dnum) values (@pid, @dnum)
                INSERT INTO paint_process (pid, method) values (@pid, @info)
            END
            IF @pdata = '3'
            BEGIN
                INSERT INTO process (pid, pdata) values (@pid, 'Fit')
                INSERT INTO supervise (pid, dnum) values (@pid, @dnum)
                INSERT INTO fit_process (pid) values (@pid)
            END
        END TRY
        BEGIN CATCH
            THROW 50001, 'Process ID already exists', 1
        END CATCH
    END
END

```

Query 5:

```

GO
CREATE PROCEDURE insert_accounts
    @account_id INT,
    @assoc INT,
    @pk INT,
    @date VARCHAR(40)
AS
BEGIN
    -- make sure other accounts aren't using this id
    IF EXISTS (SELECT 1 FROM assembly_track WHERE aanum = @account_id)
        THROW 50003, 'This Account number has already been used for an Assembly Account',1
    ELSE IF EXISTS (SELECT 1 FROM process_track WHERE panum = @account_id)
        THROW 50004, 'This Account number has already been used for a Process Account',1
    ELSE IF EXISTS (SELECT 1 FROM department_track WHERE danum = @account_id)
        THROW 50005, 'This Account number has already been used for a Department Account',1
    ELSE
        IF @assoc = 1
            EXEC insert_assembly_account @account_id, @pk, @date
        ELSE IF @assoc = 2
            EXEC insert_process_account @account_id, @pk, @date
        ELSE
            EXEC insert_department_account @account_id, @pk, @date
END

-- support assembly
GO
CREATE PROCEDURE insert_assembly_account
    @account_id INT,
    @pk INT,
    @date VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM assemblies WHERE aid = @pk)
        THROW 50003, 'Assembly ID does not exist',1
    ELSE IF EXISTS (SELECT 1 FROM assembly_track WHERE aid = @pk)
        THROW 50003, 'Assembly ID already have an associated account',1
    ELSE
        BEGIN
            INSERT INTO assembly_account (aanum, date_established, cost, aid) VALUES (@account_id, @date, '0.00', @pk)
            INSERT INTO assembly_track (aanum, aid) VALUES (@account_id, @pk)
        END
END

--support process
GO
CREATE PROCEDURE insert_process_account
    @account_id INT,
    @pk INT,
    @date VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM process WHERE pid = @pk)
        THROW 50004, 'Process ID does not exist',1
    ELSE IF EXISTS (SELECT 1 FROM process_track WHERE pid = @pk)
        THROW 50004, 'Process ID already have an associated account',1
    ELSE
        BEGIN
            INSERT INTO process_account (panum, date_established, cost, pid) VALUES (@account_id, @date, '0.00', @pk)
            INSERT INTO process_track (panum, pid) VALUES (@account_id, @pk)
        END
END

```



```
--support department
GO
CREATE PROCEDURE insert_department_account
    @account_id INT,
    @pk INT,
    @date VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM department WHERE dnum = @pk)
        THROW 50005, 'Department number does not exist',1
    ELSE IF EXISTS (SELECT 1 FROM department_track WHERE dnum = @pk)
        THROW 50005, 'Department number already have an associated account',1
    ELSE
        BEGIN
            INSERT INTO department_account (danum, date_established, cost, dnum) VALUES (@account_id, @date, '0.00', @pk)
            INSERT INTO department_track (danum, dnum) VALUES (@account_id, @pk)
        END
    END
END
```

Query 6:

```
GO
CREATE PROCEDURE insert_job
    @jnum INT,
    @aid INT,
    @pid INT,
    @date VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM assemblies WHERE aid = @aid)
        THROW 50006, 'Assembly ID does not exists',1
    ELSE IF NOT EXISTS (SELECT 1 FROM process WHERE pid = @pid)
        THROW 50006, 'Process ID does not exists',1
    ELSE
        BEGIN TRY
            INSERT INTO job (jnum, date_commence, aid, pid) VALUES (@jnum, @date, @aid, @pid)
        END TRY
        BEGIN CATCH
            THROW 50006, 'Job number already exists',1
        END CATCH
        BEGIN TRY
            INSERT INTO pass_through(pid, aid) VALUES (@pid, @aid)
        END TRY
        BEGIN CATCH
            THROW 50006, 'Assembly passing through this process once more',1
        END CATCH
    END
END
```

Query 7:

```

GO
CREATE PROCEDURE insert_job_done
    @jnum INT,
    @date DATE
AS
BEGIN
    IF EXISTS (SELECT date_complete FROM job WHERE jnum = @jnum AND date_complete IS NOT NULL)
        THROW 50007, 'This job has already been completed',1
    ELSE
        BEGIN TRY
            UPDATE job SET date_complete = @date WHERE jnum = @jnum
            SELECT process.pdata FROM job, process WHERE @jnum = job.jnum AND job.pid = process.pid
        END TRY
        BEGIN CATCH
            THROW 50007, 'Job number does not exists',1
        END CATCH
END

```

-- update remaining job information after clarification on type of process

```

GO
CREATE PROCEDURE update_job_info
    @jnum INT,
    @type VARCHAR(40),
    @labor INT,
    @info1 VARCHAR(40),
    @info2 VARCHAR(40)
AS
BEGIN
    BEGIN TRY
        UPDATE job SET info = @type WHERE jnum = @jnum
        IF @type = 'Cut'
            INSERT INTO cut_job (jnum, machine_type, machine_time, material, labor_time)
                VALUES (@jnum, (SELECT machine_type FROM job, cut_process WHERE @jnum = job.jnum AND job.pid = cut_process.pid), @info1, @info2, @labor)
        ELSE IF @type = 'Paint'
            INSERT INTO paint_job (jnum, color, volume, labor_time) VALUES (@jnum, @info1, @info2, @labor)
        ELSE IF @type = 'Fit'
            INSERT INTO fit_job (jnum, labor_time) VALUES (@jnum, @labor)
        END TRY
        BEGIN CATCH
            THROW 50008, 'Odd, how did we end up here?',1
        END CATCH
END

```

## Query 8:

```

GO
CREATE PROCEDURE insert_trans
    @tnum INT,
    @jnum INT,
    @cost FLOAT
AS
BEGIN
    DECLARE @aenum INT = (SELECT assembly_account.aenum FROM job, assembly_account WHERE @jnum = job.jnum AND job.pid = assembly_account.pid)
    DECLARE @penum INT = (SELECT process_account.penum FROM job, process_account WHERE @jnum = job.jnum AND job.pid = process_account.pid)
    DECLARE @denum INT = (SELECT department_account.denum FROM job, department_account, supervise WHERE @jnum = job.jnum AND job.pid = supervise.pid AND supervise.denum = department_account.denum)

    IF NOT EXISTS (SELECT 1 FROM job WHERE @jnum = jnum)
        THROW 50009, 'Job number does not exists',1
    ELSE IF @aenum IS NULL
        THROW 50009, 'Assembly for this job does not have a registered account',1
    ELSE IF @penum IS NULL
        THROW 50009, 'Process for this job does not have a registered account',1
    ELSE IF @denum IS NULL
        THROW 50009, 'Department for the process of this job does not have a registered account',1
    ELSE
        BEGIN TRY
            INSERT INTO transactions (tnum, sup_cost) VALUES (@tnum, @cost)
            INSERT INTO record (tnum, jnum) VALUES (@tnum, @jnum)
            INSERT INTO updates (tnum, aenum, penum, denum) VALUES (@tnum, @aenum, @penum, @denum)
            UPDATE assembly_account SET cost = cost + @cost WHERE aenum = @aenum
            UPDATE process_account SET cost = cost + @cost WHERE penum = @penum
            UPDATE department_account SET cost = cost + @cost WHERE denum = @denum
        END TRY
        BEGIN CATCH
            THROW 50009, 'Transaction number already existed',1
        END CATCH
END

```

## Query 9:

```

GO
CREATE PROCEDURE assembly_cost
    @aid INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM assemblies WHERE aid = @aid)
        THROW 50010, 'Assembly ID does not exist',1
    ELSE IF NOT EXISTS (SELECT 1 FROM assembly_account WHERE aid = @aid)
        THROW 50010, 'Assembly ID does not have an account',1
    ELSE
        SELECT cost FROM assembly_account WHERE aid = @aid
END

```

## Query 10:

```

GO
CREATE PROCEDURE labor_time
    @dnum INT,
    @date DATE
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM department WHERE dnum = @dnum)
        THROW 50011, 'Department number does not exist',1
    ELSE
        DECLARE @fit_sum INT =
            (SELECT SUM(fit_job.labor_time)
             FROM supervise, job, fit_job
             WHERE supervise.dnum = @dnum AND supervise.pid = job.pid AND job.date_complete = @date AND
                   fit_job.jnum = job.jnum);

        DECLARE @paint_sum INT =
            (SELECT DISTINCT SUM(cut_job.labor_time)
             FROM supervise, job, cut_job
             WHERE supervise.dnum = @dnum AND supervise.pid = job.pid AND job.date_complete = @date AND
                   cut_job.jnum = job.jnum);

        DECLARE @cut_sum INT =
            (SELECT DISTINCT SUM(paint_job.labor_time)
             FROM supervise, job, paint_job
             WHERE supervise.dnum = @dnum AND supervise.pid = job.pid AND job.date_complete = @date AND
                   paint_job.jnum = job.jnum);

        DECLARE @labor_sum INT = 0;
        IF @fit_sum IS NOT NULL
            SET @labor_sum = @labor_sum + @fit_sum;
        IF @paint_sum IS NOT NULL
            SET @labor_sum = @labor_sum + @paint_sum;
        IF @cut_sum IS NOT NULL
            SET @labor_sum = @labor_sum + @cut_sum;

        SELECT @labor_sum AS RESULT
END

```

## Query 11:

```

GO
CREATE PROCEDURE assembly_pass
    @aid INT
AS
BEGIN
    /*
        IF NOT EXISTS (SELECT 1 FROM assemblies WHERE aid = @aid)
            THROW 50012, 'Assembly ID does not exists', 1
        ELSE
    */
    SELECT job.date_commence, job.pid, supervise.dnum
    FROM job, supervise
    WHERE job.aid = @aid AND job.pid = supervise.pid
    ORDER BY job.date_commence ASC
END

```

Query 12:

```

GO
CREATE PROCEDURE job_complete
    @dnum INT,
    @date DATE
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM department WHERE dnum = @dnum)
        THROW 50013, 'Department number does not exists', 1
    ELSE
        SELECT ALL job.jnum, job.info, job.aid
        FROM job, supervise
        WHERE supervise.dnum = @dnum AND supervise.pid = job.pid AND job.date_complete = @date
END

```

Query 13:

```

GO
CREATE PROCEDURE get_customer
    @lb INT,
    @ub INT
AS
BEGIN
    IF @lb > @ub OR @lb < 1 OR @ub > 10
        THROW 50014, 'Requested bound is not correct', 1
    ELSE
        SELECT customer.cname, customer.caddress, customer.category
        FROM customer
        WHERE customer.category >= @lb AND customer.category <= @ub
        ORDER BY customer.cname ASC
END

```

Query 14:

```

GO
CREATE PROCEDURE delete_cut
    @joblb INT,
    @jobub INT
AS
BEGIN
    IF @joblb > @jobub
        THROW 50015, 'Job number lower bound cannot be larger than job number upper bound', 1
    ELSE
        ALTER TABLE record nocheck constraint all

        DELETE FROM cut_job
        WHERE cut_job.jnum >= @joblb AND cut_job.jnum <= @jobub
        DELETE FROM job
        WHERE job.jnum >= @joblb AND job.jnum <= @jobub AND job.info = 'Cut'

        ALTER TABLE record check constraint all
END

```

Query 15:

```

GO
CREATE PROCEDURE change_color
    @jnum INT,
    @color VARCHAR(40)
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM job WHERE jnum = @jnum AND info = 'Paint')
        THROW 50016, 'Paint job with this job number does not exist', 1
    ELSE
        UPDATE paint_job SET color = @color WHERE jnum = @jnum
END

```

## Task 5 Java Source Program and Compilation

Query 1:

```

public static void queryOne(Connection connection) {
    String name, address;
    int category;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter customer name: ");
        name = input.nextLine();
        System.out.print("Enter customer address: ");
        address = input.nextLine();
        System.out.print("Enter customer category: ");
        category = input.nextInt();

        // execute procedure
        final String insertCustomer = "EXEC insert_customer '" + name + "', '" + address + "', '" + category + "'";
        final ResultSet rs = statement.executeQuery(insertCustomer);

    } catch (Exception ae) {
        System.out.println(ae);
    }
}

```

Query 2:

```

public static void queryTwo(Connection connection) {
    String ddata;
    int dnum;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter the Department number: ");
        dnum = input.nextInt();
        ddata = input.nextLine();
        System.out.print("Enter the Department data: ");
        ddata = input.nextLine();

        // execute procedure
        final String insertDepartment = "EXEC insert_department '" + dnum + "', '" + ddata + "'";
        final ResultSet rs = statement.executeQuery(insertDepartment);

    } catch (Exception ae) {
        System.out.println(ae);
    }
}

```

Query 3:

```

try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter new Assembly ID: ");
    aid = input.nextInt();
    cname = input.nextLine();
    System.out.print("Enter the customer name who order the assembly: ");
    cname = input.nextLine();
    System.out.print("Enter Assembly detail: ");
    adetail = input.nextLine();
    System.out.print("Enter day order(DD): ");
    day = input.nextInt();
    System.out.print("Enter month order(MM): ");
    month = input.nextInt();
    System.out.print("Enter year order(YYYY): ");
    year = input.nextInt();

    // make date
    date = year + "-" + month + "-" + day;

    // execute procedure
    final String insertAssembly = "EXEC insert_assembly '" + aid + "', '" + date + "', '" + adetail + "', '" + cname + "'";
    final ResultSet rs = statement.executeQuery(insertAssembly);

} catch (Exception ae) {
    System.out.println(ae);
}

```

#### Query 4:

```

try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter new process ID: ");
    pid = input.nextInt();
    System.out.print("Enter department number for this process: ");
    dept = input.nextInt();
    System.out.print("Enter type of process{Cut(1), Paint(2), Fit(3)}: ");
    pdata = input.nextInt();

    info2 = input.nextLine(); // clear the end line

    // extra information depending on type of process
    if (pdata == 1) {
        System.out.print("Enter machine type: ");
        info2 = input.nextLine();
    } else if (pdata == 2) {
        System.out.print("Enter paint method: ");
        info2 = input.nextLine();
    } else if (pdata == 3) {
        // nothing
    } else {
        System.out.println("Invalid choice");
        return;
    }

    // execute procedure
    final String insertProcess = "EXEC insert_process '" + pid + "', '" + pdata + "', '" + dept + "', '" + info2 + "'";
    final ResultSet rs = statement.executeQuery(insertProcess);
}

```

#### Query 5:

```

try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter new Account number: ");
    account = input.nextInt();
    System.out.print("Is this for Assembly(1), Process(2), or Department(3)? ");
    assoc = input.nextInt();

    // case prompt for assembly, process, and department
    if (assoc == 1) {
        System.out.print("Enter Assembly ID for this account to track: ");
        pk = input.nextInt();
    } else if (assoc == 2) {
        System.out.print("Enter Process ID for this account to track: ");
        pk = input.nextInt();
    } else if (assoc == 3) {
        System.out.print("Enter Department number for this account to track: ");
        pk = input.nextInt();
    } else {
        System.out.println("Invalid choice");
        return;
    }

    // ask for date
    System.out.print("Enter day made(DD): ");
    day = input.nextInt();
    System.out.print("Enter month made(MM): ");
    month = input.nextInt();
    System.out.print("Enter year made(YYYY): ");
    year = input.nextInt();

    // make date
    date = year + "-" + month + "-" + day;

    //execute query
    final String insertAccount = "EXEC insert_accounts '" + account + "', '" + assoc + "', '" + pk + "', '" + date + "'";
    final ResultSet rs = statement.executeQuery(insertAccount);
}

```

#### Query 6:

```

try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter a Job number: ");
    job = input.nextInt();
    System.out.print("Enter associated Assembly ID: ");
    aid = input.nextInt();
    System.out.print("Enter associated Process ID: ");
    pid = input.nextInt();
    System.out.print("Enter day commenced(DD): ");
    day = input.nextInt();
    System.out.print("Enter month commenced(MM): ");
    month = input.nextInt();
    System.out.print("Enter year commenced(YYYY): ");
    year = input.nextInt();

    // make date
    date = year + "-" + month + "-" + day;

    //execute query
    final String insertJob = "EXEC insert_job '" + job + "', '" + aid + "', '" + pid + "', '" + date + "'";
    final ResultSet rs = statement.executeQuery(insertJob);
}

```



Query 7:

```
try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter a Job number: ");
    job = input.nextInt();
    System.out.print("Enter day completed(DD): ");
    day = input.nextInt();
    System.out.print("Enter month completed(MM): ");
    month = input.nextInt();
    System.out.print("Enter year completed(YYYY): ");
    year = input.nextInt();

    // make date
    date = year + "-" + month + "-" + day;

    // find and update table and retrieve info for information
    final String insertJob = "EXEC insert_job_done '" + job + "', '" + date + "'";
    final ResultSet rs = statement.executeQuery(insertJob);

    // check if table with the job number exist
    if (!rs.next()) {
        System.out.println("Job number does not exist");
        return;
    }

    // get the job type
    String type = input.nextLine();
    type = rs.getString(1);

    // get the other information depending on job type
    if (type.equals("Cut")) {
        System.out.print("Enter machine time(minutes): ");
        info1 = input.nextLine();
        System.out.print("Enter materials: ");
        info2 = input.nextLine();
    } else if (type.equals("Paint")) {
        System.out.print("Enter Color: ");
        info1 = input.nextLine();
        System.out.print("Enter Volume(Litres): ");
        info2 = input.nextLine();
    } else if (type.equals("Fit")) {
        // nothing
    } else {
        System.out.println("Table malfunction");
        return;
    }
}
```

```
// labor time
System.out.print("Enter labor time(minutes): ");
labor = input.nextInt();

// execute procedure
final String updateInfo = "EXEC update_job_info '" + job + "', '" + type + "', '" + labor + "', '" + info1 + "', '" + info2 + "'";
final ResultSet rs2 = statement.executeQuery(updateInfo);
```

#### Query 8:

```
public static void queryEight(Connection connection) {
    int trans, job;
    float cost;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter transaction number: ");
        trans = input.nextInt();
        System.out.print("Enter a job number for this transaction: ");
        job = input.nextInt();
        System.out.print("Enter the transaction cost $: ");
        cost = input.nextFloat();

        // execute procedure
        final String insertTrans = "EXEC insert_trans '" + trans + "', '" + job + "', '" + cost + "'";
        final ResultSet rs = statement.executeQuery(insertTrans);

    } catch (Exception ae) {
        System.out.println(ae);
    }
}
```

#### Query 9:

```
public static void queryNine(Connection connection) {
    int aid;

    Scanner input = new Scanner(System.in);
    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter Assembly ID to retrieve cost: ");
        aid = input.nextInt();

        // execute procedure
        final String assemblyCost = "EXEC assembly_cost '" + aid + "'";
        final ResultSet rs = statement.executeQuery(assemblyCost);

        // print cost
        if (rs.next()) {
            System.out.println("\n\tAssembly ID: " + aid + "\n\tCost for this Assembly: $" + rs.getFloat(1) + "\n");
        }

    } catch (Exception ae) {
        System.out.println(ae);
    }
}
```

#### Query 10:

```

public static void queryTen(Connection connection) {
    int dnum, day, month, year;
    String date;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter department to retrieve labor time from: ");
        dnum = input.nextInt();
        System.out.print("Enter day(DD): ");
        day = input.nextInt();
        System.out.print("Enter month(MM): ");
        month = input.nextInt();
        System.out.print("Enter year(YYYY): ");
        year = input.nextInt();

        // make date
        date = year + "-" + month + "-" + day;

        // execute procedure
        final String laborTime = "EXEC labor_time '" + dnum + "', '" + date + "'";
        PreparedStatement pst = connection.prepareStatement(laborTime);
        final ResultSet rs = pst.executeQuery();

        // print total labor time
        if (rs.next()) {
            System.out.println("\n\tTotal Labor time on " + date + " for Department "
                               + dnum + ": " + rs.getString(1) + " minutes\n");
        }

    } catch (Exception ae) {
        System.out.println(ae);
    }
}

```

Query 11:

```

public static void queryEleven(Connection connection) {
    int aid;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter assembly ID: ");
        aid = input.nextInt();

        // execute procedure
        final String assemblyPass = "EXEC assembly_pass '" + aid + "'";
        final ResultSet rs = statement.executeQuery(assemblyPass);

        // print out all process that have that assembly passed and department
        System.out.println();
        while (rs.next()) {
            System.out.println(String.format("\tDate: %s | Process: %s | Department: %s",
                rs.getString(1),
                rs.getString(2),
                rs.getString(3)));
        }
        System.out.println();

    } catch (Exception ae) {
        System.out.println(ae);
    }
}

```

Query 12:

```

public static void queryTwelve(Connection connection) {
    int dnum, day, month, year;
    String date;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter Department number: ");
        dnum = input.nextInt();
        System.out.print("Enter day completed(DD): ");
        day = input.nextInt();
        System.out.print("Enter month completed(MM): ");
        month = input.nextInt();
        System.out.print("Enter year completed(YYYY): ");
        year = input.nextInt();

        // date
        date = year + "-" + month + "-" + day;
    }
}

```

```

// execute procedure
final String jobComplete = "EXEC job_complete '" + dnum + "', '" + date + "'";
final ResultSet rs = statement.executeQuery(jobComplete);

// print out the job number and types with assembly id
System.out.println();
while (rs.next()) {
    System.out.println(String.format("\tJob number: %s | Job Type: %s | Assembly ID: %s",
        rs.getString(1),
        rs.getString(2),
        rs.getString(3)));
}
System.out.println();

} catch (Exception ae) {
    System.out.println(ae);
}
}

```

Query 13:

```

int low, upp;

Scanner input = new Scanner(System.in);

try (final Statement statement = connection.createStatement()) {

    // input prompt
    System.out.print("Enter Customer category lower bound(1-10): ");
    low = input.nextInt();
    System.out.print("Enter Customer category upper bound(1-10, can be the same): ");
    upp = input.nextInt();

    // execute procedure
    final String getCustomer = "EXEC get_customer '" + low + "', '" + upp + "'";
    final ResultSet rs = statement.executeQuery(getCustomer);

    // print out customer name
    System.out.println();
    while (rs.next()) {
        System.out.println(String.format("\tCustomer name: %s | Address: %s | Category: %s",
            rs.getString(1),
            rs.getString(2),
            rs.getString(3)));
    }
    System.out.println();

} catch (Exception ae) {
    System.out.println(ae);
}

```

---

Query 14:

```

public static void queryFourteen(Connection connection) {
    int joblb, jobup;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter Job number lower bound: ");
        joblb = input.nextInt();
        System.out.print("Enter Job number upper bound: ");
        jobup = input.nextInt();

        // execute procedure
        final String deleteCut = "EXEC delete_cut '" + joblb + "', '" + jobup + "'";
        final ResultSet rs = statement.executeQuery(deleteCut);

    } catch (Exception ae) {
        System.out.println(ae);
    }

}

```

Query 15:

```

public static void queryFifteen(Connection connection) {
    int job;
    String color;

    Scanner input = new Scanner(System.in);

    try (final Statement statement = connection.createStatement()) {

        // input prompt
        System.out.print("Enter Paint Job ID: ");
        job = input.nextInt();
        color = input.nextLine();
        System.out.print("Enter new color: ");
        color = input.nextLine();

        // execute procedure
        final String changeColor = "EXEC change_color '" + job + "', '" + color + "'";
        final ResultSet rs = statement.executeQuery(changeColor);

    } catch (Exception ae) {
        System.out.println(ae);
    }

}

```

Import customer:

```

public static void importCustomer(Connection connection) {
    String fileName, cname, address, skipLine;
    String category;

    Scanner input = new Scanner(System.in);

    try {

        // prompt file name
        System.out.print("Enter file name: ");
        fileName = input.nextLine();

        // make scanner for it
        Scanner f = new Scanner(new File(fileName));
        // delimiter to remove tab commas and spaces (requires file to have spaces)
        f.useDelimiter(";|\\t|\\r\\n");

        // while the file has more line, we only get first 3 string between each semi colon
        while (f.hasNext()) {
            // get the first 3 string
            cname = f.next();
            address = f.next();
            category = f.next();

            try (final Statement statement = connection.createStatement()) {

                // execute procedure
                String insertCustomer = "EXEC insert_customer '" + cname + "', '" + address + "', '" + category + "'";
                ResultSet rs = statement.executeQuery(insertCustomer);

            } catch (Exception ae) {
                System.out.println(ae);
            }

        }

    } catch (Exception ae) {
        System.out.println(ae);
    }
}

```

Export customer:

```

public static void exportCustomer(Connection connection) {
    String fileName, cname, address, category;
    int low, upp;

    Scanner input = new Scanner(System.in);

    // ask file name
    System.out.print("Enter file name: ");
    fileName = input.nextLine();

    try (final Statement statement = connection.createStatement()) {

        // make printwriter
        PrintWriter out = new PrintWriter(fileName, "UTF-8");

        // prompt lower and upper bound
        System.out.print("Enter Customer category lower bound(1-10): ");
        low = input.nextInt();
        System.out.print("Enter Customer category upper bound(1-10, can be the same): ");
        upp = input.nextInt();
    }
}

```

```

// execute procedure
final String getCustomer = "EXEC get_customer '" + low + "', '" + upp + "'";
final ResultSet rs = statement.executeQuery(getCustomer);

// use result set and write to file
while (rs.next()) {
    out.write(rs.getString(1) + ";" + rs.getString(2) + ";" + rs.getString(3) + "\r\n");
}
out.close();

} catch (Exception ae) {
    System.out.println(ae);
}
}

```

Main/ choice selection:

```

public static void main(String[] args) {
    // Connect to database
    final String hostName = "1e0058-sql-server.database.windows.net";
    final String dbName = "cs-dss-4513-sql-db";
    final String user = "1e0058";
    final String password = "Memelord1";
    final String url = String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;hostNameInCertificate=");

    try (final Connection connection = DriverManager.getConnection(url)) {
        final String schema = connection.getSchema();
        System.out.println("Successful connection - Schema: " + schema);
        System.out.println();

        Scanner input = new Scanner(System.in);
        int choice;

        while (true) {
            System.out.println("WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM");
            System.out.println("Choice 1: Enter a new customer");
            System.out.println("Choice 2: Enter a new department");
            System.out.println("Choice 3: Enter a new assembly with its customer-name, assembly-details, assembly-id, and");
            System.out.println("Choice 4: Enter a new process-id and its department together with its type and information relevant");
            System.out.println("Choice 5: Create a new account and associate it with the process, assembly, or department to which");
            System.out.println("Choice 6: Enter a new job, given its job-no, assembly-id, process-id, and date the job");
            System.out.println("Choice 7: At the completion of a job, enter the date it completed and the information relevant to");
            System.out.println("Choice 8: Enter a transaction-no and its sup-cost and update all the costs (details) of");
            System.out.println("Choice 9: Retrieve the cost incurred on an assembly-id");
            System.out.println("Choice 10: Retrieve the total labor time within a department for jobs completed in the department");
            System.out.println("Choice 11: Retrieve the processes through which a given assembly-id has passed so far");
            System.out.println("Choice 12: Retrieve the jobs (together with their type information and assembly-id) complet

```



```

System.out.println("Choice 12: Retrieve the jobs (together with their type information and assembly-id) comp!
System.out.println("Choice 13: Retrieve the customers (in name order) whose category is in a givenrange");
System.out.println("Choice 14: Delete all cut-jobs whose job-no is in a givenrange ");
System.out.println("Choice 15: Change the color of a given paint job");
System.out.println("Choice 16: Import: enter new customers from a data file until the file is empty");
System.out.println("Choice 17: Export: Retrieve the customers (in name order) whose category is in a given range and cu
System.out.println("Choice 18: Quit");
System.out.print("Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: ");

// input
choice = input.nextInt();

// input evaluation
if (choice == 18) {
    break;
} else if (choice == 1) {
    queryOne(connection);
} else if (choice == 2) {
    queryTwo(connection);
} else if (choice == 3) {
    queryThree(connection);
} else if (choice == 4) {
    queryFour(connection);
} else if (choice == 5) {
    queryFive(connection);
} else if (choice == 6) {
    querySix(connection);
} else if (choice == 7) {
    querySeven(connection);

} else if (choice == 8) {
    queryEight(connection);
} else if (choice == 9) {
    queryNine(connection);
} else if (choice == 10) {
    queryTen(connection);
} else if (choice == 11) {
    queryEleven(connection);
} else if (choice == 12) {
    queryTwelve(connection);
} else if (choice == 13) {
    queryThirteen(connection);
} else if (choice == 14) {
    queryFourteen(connection);
} else if (choice == 15) {
    queryFifteen(connection);
} else if (choice == 16) {
    importCustomer(connection);
} else if (choice == 17) {
    exportCustomer(connection);
} else {
    System.out.println("Choices not in range.");
}

}
} catch (Exception ae) {
    System.out.println(ae);
}
System.out.println("Fin");
}

```

Successful compilation:

```

Problems | Declaration | Console 12
Project [Java Application] C:\Program Files\Java\jdk-13\bin\javaw.exe (Nov 13, 2019, 7:44:24 PM)
Successful connection - Schema: dbo

WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM
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
Choice 9: Retrieve the cost incurred on an assembly-id
Choice 10: Retrieve the total labor time within a department for jobs completed in the department during a given date
Choice 11: Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the depart
Choice 12: Retrieve the jobs (together with their type information and assembly-id) completed during a given date in a given de
Choice 13: Retrieve the customers (in name order) whose category is in a givenrange
Choice 14: Delete all out-jobs whose job-no is in a givenrange
Choice 15: Change the color of a given paint job
Choice 16: Import: enter new customers from a data file until the file is empty
Choice 17: Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of scre
Choice 18: Quit
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18:

```

## Task 6 Java Program Execution

### 6.1 Query 1

1)

```

Choice 18: Quit
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: gon freecs
Enter customer address: whale island
Enter customer category: 1
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

```

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: killua zoldyck
Enter customer address: kukuroo mountain
Enter customer category: 10
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: kurapika
Enter customer address: kurta village
Enter customer category: 5
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: leorio paladiknight
Enter customer address: hunter university
Enter customer category: 3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: pariston hill
Enter customer address: hunter association
Enter customer category: 8
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Table:

Customer:

	cname	caddress	category
1	gon freecs	whale island	1
2	killua zoldyck	kukuroo mounta...	10
3	kurapika	kurta village	5
4	leorio paladik...	hunter univers...	3
5	pariston hill	hunter associat...	8
6	NULL	NULL	NULL

## 6.2 Query 2

1)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 2
Enter the Department number: 1
Enter the Department data: process 1/2/3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 2
Enter the Department number: 2
Enter the Department data: process 4/5/6
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 2
Enter the Department number: 3
Enter the Department data: TBA
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 2
Enter the Department number: 10
Enter the Department data: process 99/100
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 2
Enter the Department number: 100
Enter the Department data: process 10
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Table:

Department:

	dnum	ddata
1	1	process 1/2/3
2	2	process 4/5/6
3	3	TBA
4	10	process 99/100
5	100	process 10
6	NULL	NULL

### 6.3 Query 3

1)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 1
Enter the customer name who order the assembly: gon freecs
Enter Assembly detail: process 1/2
Enter day order(DD): 1
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 2
Enter the customer name who order the assembly: kurapika
Enter Assembly detail: process 2
Enter day order(DD): 2
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 3
Enter the customer name who order the assembly: kurapika
Enter Assembly detail: process 3
Enter day order(DD): 3
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 4
Enter the customer name who order the assembly: killua zoldyck
Enter Assembly detail: none
Enter day order(DD): 4
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 5
Enter the customer name who order the assembly: leorio paladiknight
Enter Assembly detail: process 5
Enter day order(DD): 5
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

6)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 6
Enter the customer name who order the assembly: pariston hill
Enter Assembly detail: process 4
Enter day order(DD): 6
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

7)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 7
Enter the customer name who order the assembly: pariston hill
Enter Assembly detail: process 9/10
Enter day order(DD): 7
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

8)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 8
Enter the customer name who order the assembly: killua zoldyck
Enter Assembly detail: none
Enter day order(DD): 8
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

```

9)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 100
Enter the customer name who order the assembly: gon freecs
Enter Assembly detail: process 100
Enter day order(DD): 31
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

10)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 3
Enter new Assembly ID: 10
Enter the customer name who order the assembly: kurapika
Enter Assembly detail: process 99
Enter day order(DD): 10
Enter month order(MM): 1
Enter year order(YYYY): 1991
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Tables:

Assemblies:

	aid	date_order	detail
1	1	1991-01-01	process 1/2
2	2	1991-01-02	process 2
3	3	1991-01-03	process 3
4	4	1991-01-04	none
5	5	1991-01-05	process 5
6	6	1991-01-06	process 4
7	7	1991-01-07	process 9/10
8	8	1991-01-08	none
9	10	1991-01-10	process 99
-	100	1991-01-31	process 100
-	NULL	NULL	NULL

Orders:

	cname	aid
1	gon freecs	1
2	kurapika	2
3	kurapika	3
4	killua zoldyck	4
5	leorio paladik...	5
6	pariston hill	6
7	pariston hill	7
8	killua zoldyck	8
9	kurapika	10
-	gon freecs	100
-	NULL	NULL

## 6.4 Query 4

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 1
Enter department number for this process: 1
Enter type of process{Cut(1), Paint(2), Fit(3)}: 3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 2
Enter department number for this process: 1
Enter type of process{Cut(1), Paint(2), Fit(3)}: 3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

3)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 3
Enter department number for this process: 1
Enter type of process{Cut(1), Paint(2), Fit(3)}: 2
Enter paint method: acrylic
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 4
Enter department number for this process: 2
Enter type of process{Cut(1), Paint(2), Fit(3)}: 1
Enter machine type: circular
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 5
Enter department number for this process: 2
Enter type of process{Cut(1), Paint(2), Fit(3)}: 1
Enter machine type: round
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

6)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 6
Enter department number for this process: 2
Enter type of process{Cut(1), Paint(2), Fit(3)}: 2
Enter paint method: oil
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

7)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 99
Enter department number for this process: 10
Enter type of process{Cut(1), Paint(2), Fit(3)}: 1
Enter machine type: base
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

8)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 100
Enter department number for this process: 10
Enter type of process{Cut(1), Paint(2), Fit(3)}: 3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

9)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 10
Enter department number for this process: 100
Enter type of process{Cut(1), Paint(2), Fit(3)}: 2
Enter paint method: water color
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

10)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 9
Enter department number for this process: 3
Enter type of process{Cut(1), Paint(2), Fit(3)}: 3
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```



Tables:

Process:

	pid	pdata
1	1	Fit
2	2	Fit
3	3	Paint
4	4	Cut
5	5	Cut
6	6	Paint
7	9	Fit
8	10	Paint
9	99	Cut
—	100	Fit
—	NULL	NULL

Supervise:

	pid	dnum
1	1	1
2	2	1
3	3	1
4	4	2
5	5	2
6	6	2
7	9	3
8	99	10
9	100	10
—	10	100
—	NULL	NULL

Fit\_process:

	pid
1	1
2	2
3	9
4	100
5	NULL

Paint\_process:

	pid	method
1	3	acrylic
2	6	oil
3	10	water color
4	NULL	NULL

Cut\_process:

	pid	machine_type
1	99	base
2	4	circular
3	5	round
4	NULL	NULL

## 6.5 Query 5

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 10
Is this for Assembly(1), Process(2), or Department(3)? 3
Enter Department number for this account to track: 1
Enter day made (DD): 1
Enter month made (MM): 1
Enter year made (YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 5
Is this for Assembly(1), Process(2), or Department(3)? 1
Enter Assembly ID for this account to track: 1
Enter day made(DD): 20
Enter month made(MM): 1
Enter year made(YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 1
Is this for Assembly(1), Process(2), or Department(3)? 2
Enter Process ID for this account to track: 2
Enter day made(DD): 2
Enter month made(MM): 2
Enter year made(YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 3
Is this for Assembly(1), Process(2), or Department(3)? 3
Enter Department number for this account to track: 2
Enter day made(DD): 3
Enter month made(MM): 2
Enter year made(YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 2
Is this for Assembly(1), Process(2), or Department(3)? 1
Enter Assembly ID for this account to track: 5
Enter day made(DD): 2
Enter month made(MM): 5
Enter year made(YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

6)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 9
Is this for Assembly(1), Process(2), or Department(3)? 2
Enter Process ID for this account to track: 5
Enter day made(DD): 2
Enter month made(MM): 5
Enter year made(YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

7)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 4
Is this for Assembly(1), Process(2), or Department(3)? 3
Enter Department number for this account to track: 100
Enter day made (DD): 4
Enter month made (MM): 3
Enter year made (YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

8)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 100
Is this for Assembly(1), Process(2), or Department(3)? 1
Enter Assembly ID for this account to track: 7
Enter day made (DD): 1
Enter month made (MM): 7
Enter year made (YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

9)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 8
Is this for Assembly(1), Process(2), or Department(3)? 2
Enter Process ID for this account to track: 10
Enter day made (DD): 2
Enter month made (MM): 10
Enter year made (YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

10)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 5
Enter new Account number: 99
Is this for Assembly(1), Process(2), or Department(3)? 1
Enter Assembly ID for this account to track: 2
Enter day made (DD): 1
Enter month made (MM): 2
Enter year made (YYYY): 1990
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Tables:

Assembly\_account:

	aanum	date_establish_	cost	aid
1	2	1990-05-02	0	5
2	5	1990-01-20	0	1
3	99	1990-02-01	0	2
4	100	1990-07-01	0	7
5	NULL	NULL	NULL	NULL

Assembly\_track:

	aid	aanum
1	1	5
2	2	99
3	5	2
4	7	100
5	NULL	NULL

Process\_account:

	panum	date_establish_	cost	pid
1	1	1990-02-02	0	2
2	8	1990-10-02	0	10
3	9	1990-05-02	0	5
4	NULL	NULL	NULL	NULL

Process\_track:

	pid	panum
1	2	1
2	5	9
3	10	8
4	NULL	NULL

Department\_account:

	danum	date_establish_	cost	dnum
1	3	1990-02-03	0	2
2	4	1990-03-04	0	100
3	10	1990-01-01	0	1
4	NULL	NULL	NULL	NULL

Department\_track:

	dnum	danum
1	1	10
2	2	3
3	100	4
4	NULL	NULL

## 6.6 Query 6

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 10
Enter associated Assembly ID: 1
Enter associated Process ID: 2
Enter day commenced(DD): 1
Enter month commenced(MM): 2
Enter year commenced(YYYY): 1999
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 9
Enter associated Assembly ID: 2
Enter associated Process ID: 2
Enter day commenced(DD): 2
Enter month commenced(MM): 2
Enter year commenced(YYYY): 1999
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

3)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 8
Enter associated Assembly ID: 5
Enter associated Process ID: 5
Enter day commenced(DD): 5
Enter month commenced(MM): 5
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

4)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 7
Enter associated Assembly ID: 7
Enter associated Process ID: 10
Enter day commenced(DD): 7
Enter month commenced(MM): 10
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

5)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 1
Enter associated Assembly ID: 2
Enter associated Process ID: 2
Enter day commenced(DD): 3
Enter month commenced(MM): 2
Enter year commenced(YYYY): 1999
com.microsoft.sqlserver.jdbc.SQLServerException: Assembly passing through this process once more

```

6)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 2
Enter associated Assembly ID: 1
Enter associated Process ID: 10
Enter day commenced(DD): 1
Enter month commenced(MM): 10
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

7)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 3
Enter associated Assembly ID: 5
Enter associated Process ID: 2
Enter day commenced(DD): 5
Enter month commenced(MM): 2
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

8)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 5
Enter associated Assembly ID: 2
Enter associated Process ID: 10
Enter day commenced(DD): 1
Enter month commenced(MM): 2
Enter year commenced(YYYY): 1999
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

9)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 4
Enter associated Assembly ID: 1
Enter associated Process ID: 5
Enter day commenced(DD): 5
Enter month commenced(MM): 6
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

10)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 100
Enter associated Assembly ID: 6
Enter associated Process ID: 4
Enter day commenced(DD): 6
Enter month commenced(MM): 4
Enter year commenced(YYYY): 1992
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Tables:

Jobs:

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	NULL	2	2	NULL
2	2	1992-10-01	NULL	1	10	NULL
3	3	1992-02-05	NULL	5	2	NULL
4	4	1992-06-05	NULL	1	5	NULL
5	5	1999-02-01	NULL	2	10	NULL
6	7	1992-10-07	NULL	7	10	NULL
7	8	1992-05-05	NULL	5	5	NULL
8	9	1999-02-02	NULL	2	2	NULL
9	10	1999-02-01	NULL	1	2	NULL
...	100	1992-04-06	NULL	6	4	NULL
...	NULL	NULL	NULL	NULL	NULL	NULL

Pass\_through:

	aid	pid
1	1	2
2	1	5
3	1	10
4	2	2
5	2	10
6	5	2
7	5	5
8	6	4
9	7	10
...	NULL	NULL



## 6.7 Query 7

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 100
Enter day completed(DD): 6
Enter month completed(MM): 5
Enter year completed(YYYY): 1992
Enter machine time(minutes): 30
Enter materials: iron, wood, sleet
Enter labor time(minutes): 10
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 4
Enter day completed(DD): 5
Enter month completed(MM): 7
Enter year completed(YYYY): 1992
Enter machine time(minutes): 25
Enter materials: steel
Enter labor time(minutes): 15
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

3)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 10
Enter day completed(DD): 1
Enter month completed(MM): 3
Enter year completed(YYYY): 1999
Enter labor time(minutes): 30
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

4)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 1
Enter day completed(DD): 1
Enter month completed(MM): 3
Enter year completed(YYYY): 1999
Enter labor time(minutes): 100
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

5)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 9
Enter day completed(DD): 2
Enter month completed(MM): 3
Enter year completed(YYYY): 1999
Enter labor time(minutes): 45
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

6)

```

Enter a Job number: 8
Enter day completed(DD): 5
Enter month completed(MM): 5
Enter year completed(YYYY): 2000
Enter machine time(minutes): 10
Enter materials: stones, marbles
Enter labor time(minutes): 25
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

7)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 3
Enter day completed(DD): 1
Enter month completed(MM): 3
Enter year completed(YYYY): 1999
Enter labor time(minutes): 50
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

8)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 7
Enter day completed(DD): 10
Enter month completed(MM): 10
Enter year completed(YYYY): 1993
Enter Color: red
Enter Volume(Litres): 100
Enter labor time(minutes): 70
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

9)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 2
Enter day completed(DD): 10
Enter month completed(MM): 10
Enter year completed(YYYY): 1993
Enter Color: purple
Enter Volume(Litres): 50
Enter labor time(minutes): 35
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

10)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 7
Enter a Job number: 5
Enter day completed(DD): 1
Enter month completed(MM): 3
Enter year completed(YYYY): 1999
Enter Color: orange
Enter Volume(Litres): 120
Enter labor time(minutes): 90
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Tables:

Job:

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	1999-03-01	2	2	Fit
2	2	1992-10-01	1993-10-10	1	10	Paint
3	3	1992-02-05	1999-03-01	5	2	Fit
4	4	1992-06-05	1992-07-05	1	5	Cut
5	5	1999-02-01	1999-03-01	2	10	Paint
6	7	1992-10-07	1993-10-10	7	10	Paint
7	8	1992-05-05	2000-05-05	5	5	Cut
8	9	1999-02-02	1999-03-02	2	2	Fit
9	10	1999-02-01	1999-03-01	1	2	Fit
—	100	1992-04-06	1992-05-06	6	4	Cut
—	NULL	NULL	NULL	NULL	NULL	NULL

Cut\_job:

	jnum	machine_type	machine_time	material	labor_time
1	4	round	25	steel	15
2	8	round	10	stones, marbles	25
3	100	circular	30	iron, wood, sl...	10
4	NULL	NULL	NULL	NULL	NULL

Fit\_job:

	jnum	labor_time
1	1	100
2	3	50
3	9	45
4	10	30
5	NULL	NULL

Paint\_job:

	jnum	color	volume	labor_time
1	2	purple	50	35
2	5	orange	120	90
3	7	red	100	70
4	NULL	NULL	NULL	NULL

## 6.8 Query 8

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 100
Enter a job number for this transaction: 10
Enter the transaction cost $: 100
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 101
Enter a job number for this transaction: 9
Enter the transaction cost $: 150
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

3)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 102
Enter a job number for this transaction: 8
Enter the transaction cost $: 75
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

4)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 103
Enter a job number for this transaction: 7
Enter the transaction cost $: 50
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

5)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 2
Enter a job number for this transaction: 2
Enter the transaction cost $: 150.50
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

6)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 1
Enter a job number for this transaction: 9
Enter the transaction cost $: 50
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

7)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 3
Enter a job number for this transaction: 3
Enter the transaction cost $: 275.50
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

8)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 4
Enter a job number for this transaction: 5
Enter the transaction cost $: 20
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

9)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 5
Enter a job number for this transaction: 4
Enter the transaction cost $: 85.25
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

10)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 8
Enter transaction number: 10
Enter a job number for this transaction: 7
Enter the transaction cost $: 250
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

Tables:

Transactions:

	tnum	sup_cost
1	1	50
2	2	150.5
3	3	275.5
4	4	20
5	5	85.25
6	10	250
7	100	100
8	101	150
9	102	75
—	103	50
—	NULL	NULL

Record:

	tnum	jnum
1	1	9
2	2	2
3	3	3
4	4	5
5	5	4
6	10	7
7	100	10
8	101	9
9	102	8
—	103	7
—	NULL	NULL

Updates:

	tnum	aanum	panum	danum
1	1	99	1	10
2	2	5	8	4
3	3	2	1	10
4	4	99	8	4
5	5	5	9	3
6	10	100	8	4
7	100	5	1	10
8	101	99	1	10
9	102	2	9	3
—	103	100	8	4
—	NULL	NULL	NULL	NULL

Assembly\_account:

	aanum	date_establish_	cost	aid
1	2	1990-05-02	350.5	5
2	5	1990-01-20	335.75	1
3	99	1990-02-01	220	2
4	100	1990-07-01	300	7
5	NULL	NULL	NULL	NULL

Process\_account:

	panum	date_establish_	cost	pid
1	1	1990-02-02	575.5	2
2	8	1990-10-02	470.5	10
3	9	1990-05-02	160.25	5
4	NULL	NULL	NULL	NULL

Department\_account:

	danum	date_establish_	cost	dnum
1	3	1990-02-03	160.25	2
2	4	1990-03-04	470.5	100
3	10	1990-01-01	575.5	1
4	NULL	NULL	NULL	NULL

### 6.9 Query 9

1)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 9

Enter Assembly ID to retrieve cost: 1

|

Assembly ID: 1

Cost for this Assembly: \$335.75

2)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 9

Enter Assembly ID to retrieve cost: 7

|

Assembly ID: 7

Cost for this Assembly: \$300.0

3)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 9

Enter Assembly ID to retrieve cost: 5

|

Assembly ID: 5

Cost for this Assembly: \$350.5

### 6.10 Query 10

1)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 10

Enter department to retrieve labor time from: 1

Enter day(DD): 1

Enter month(MM): 3

Enter year(YYYY): 1999

Total Labor time on 1999-3-1 for Department 1: 180 minutes

2)



```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 10
Enter department to retrieve labor time from: 100
Enter day(DD): 1
Enter month(MM): 3
Enter year(YYYY): 1999
|
    Total Labor time on 1999-3-1 for Department 100: 90 minutes

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 10
Enter department to retrieve labor time from: 100
Enter day(DD): 10
Enter month(MM): 10
Enter year(YYYY): 1993
|
    Total Labor time on 1993-10-10 for Department 100: 105 minutes

```

## 6.11 Query 11

1)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 11
Enter assembly ID: 1

    Date: 1992-06-05 | Process: 5 | Department: 2
    Date: 1992-10-01 | Process: 10 | Department: 100
    Date: 1999-02-01 | Process: 2 | Department: 1

```

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 11
Enter assembly ID: 2
|
    Date: 1999-02-01 | Process: 10 | Department: 100
    Date: 1999-02-02 | Process: 2 | Department: 1
    Date: 1999-02-03 | Process: 2 | Department: 1

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 11
Enter assembly ID: 7

    Date: 1992-10-07 | Process: 10 | Department: 100

```

## 6.12 Query 12

1)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 12  
 Enter Department number: 1  
 Enter day completed(DD): 1  
 Enter month completed(MM): 3  
 Enter year completed(YYYY): 1999

|  
 Job number: 1 | Job Type: Fit | Assembly ID: 2  
 Job number: 3 | Job Type: Fit | Assembly ID: 5  
 Job number: 10 | Job Type: Fit | Assembly ID: 1

2)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 12  
 Enter Department number: 2  
 Enter day completed(DD): 5  
 Enter month completed(MM): 5  
 Enter year completed(YYYY): 2000

|  
 Job number: 8 | Job Type: Cut | Assembly ID: 5

3)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 12  
 Enter Department number: 100  
 Enter day completed(DD): 10  
 Enter month completed(MM): 10  
 Enter year completed(YYYY): 1993

Job number: 2 | Job Type: Paint | Assembly ID: 1  
 Job number: 7 | Job Type: Paint | Assembly ID: 7

### 6.13 Query 13

1)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 13  
 Enter Customer category lower bound(1-10): 1  
 Enter Customer category upper bound(1-10, can be the same): 10

|  
 Customer name: gon freecs | Address: whale island | Category: 1  
 Customer name: killua zoldyck | Address: kukuroo mountain | Category: 10  
 Customer name: kurapika | Address: kurta village | Category: 5  
 Customer name: leorio paladiknight | Address: hunter university | Category: 3  
 Customer name: pariston hill | Address: hunter association | Category: 8

2)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 13
Enter Customer category lower bound(1-10): 1
Enter Customer category upper bound(1-10, can be the same): 5
|
    Customer name: gon freecs | Address: whale island | Category: 1
    Customer name: kurapika | Address: kurta village | Category: 5
    Customer name: leorio paladiknight | Address: hunter university | Category: 3

```

3)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 13
Enter Customer category lower bound(1-10): 10
Enter Customer category upper bound(1-10, can be the same): 10
|
    Customer name: killua zoldyck | Address: kukuroo mountain | Category: 10

```

## 6.14 Query 14

1)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 14
Enter Job number lower bound: 1
Enter Job number upper bound: 5
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

```

Job:

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	1999-03-01	2	2	Fit
2	2	1992-10-01	1993-10-10	1	10	Paint
3	3	1992-02-05	1999-03-01	5	2	Fit
4	5	1999-02-01	1999-03-01	2	10	Paint
5	7	1992-10-07	1993-10-10	7	10	Paint
6	8	1992-05-05	2000-05-05	5	5	Cut
7	9	1999-02-02	1999-03-02	2	2	Fit
8	10	1999-02-01	1999-03-01	1	2	Fit
9	100	1992-04-06	1992-05-06	6	4	Cut
—	NULL	NULL	NULL	NULL	NULL	NULL

Cut\_job:

	jnum	machine_type	machine_time	material	labor_time
1	8	round	10	stones, marbles	25
2	100	circular	30	iron, wood, sl...	10
3	NULL	NULL	NULL	NULL	NULL

2)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 14

Enter Job number lower bound: 5

Enter Job number upper bound: 10

[com.microsoft.sqlserver.jdbc.SQLServerException](#): The statement did not return a result set.

Job:

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	1999-03-01	2	2	Fit
2	2	1992-10-01	1993-10-10	1	10	Paint
3	3	1992-02-05	1999-03-01	5	2	Fit
4	5	1999-02-01	1999-03-01	2	10	Paint
5	7	1992-10-07	1993-10-10	7	10	Paint
6	9	1999-02-02	1999-03-02	2	2	Fit
7	10	1999-02-01	1999-03-01	1	2	Fit
8	100	1992-04-06	1992-05-06	6	4	Cut
9	NULL	NULL	NULL	NULL	NULL	NULL

Cut\_job:

	jnum	machine_type	machine_time	material	labor_time
1	100	circular	30	iron, wood, sl...	10
2	NULL	NULL	NULL	NULL	NULL

3)

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 14

Enter Job number lower bound: 99

Enter Job number upper bound: 101

[com.microsoft.sqlserver.jdbc.SQLServerException](#): The statement did not return a result set.

Job:

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	1999-03-01	2	2	Fit
2	2	1992-10-01	1993-10-10	1	10	Paint
3	3	1992-02-05	1999-03-01	5	2	Fit
4	5	1999-02-01	1999-03-01	2	10	Paint
5	7	1992-10-07	1993-10-10	7	10	Paint
6	9	1999-02-02	1999-03-02	2	2	Fit
7	10	1999-02-01	1999-03-01	1	2	Fit
8	NULL	NULL	NULL	NULL	NULL	NULL

Cut\_job:

	jnum	machine_type	machine_time	material	labor_time
1	NULL	NULL	NULL	NULL	NULL

## 6.15 Query 15

1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 15
Enter Paint Job ID: 7
Enter new color: white
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

2)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 15
Enter Paint Job ID: 2
Enter new color: yellow
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

3)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 15
Enter Paint Job ID: 5
Enter new color: black
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
```

Table:

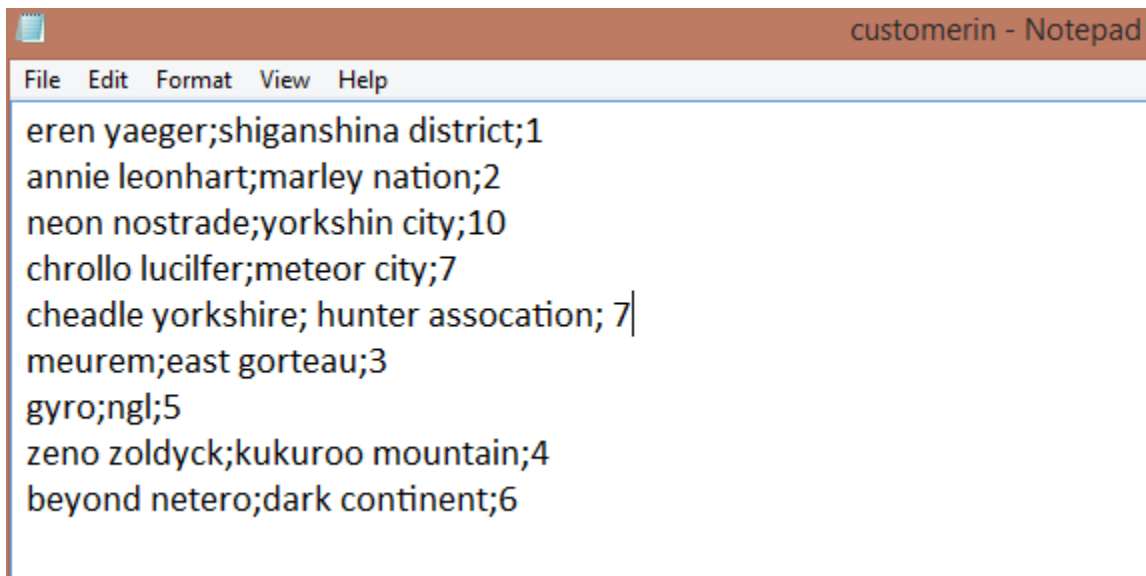
Paint\_job:

	jnum	color	volume	labor_time
1	2	yellow	50	35
2	5	black	120	90
3	7	white	100	70
4	NULL	NULL	NULL	NULL

(Was purple, orange, red before)

## 6.16 Import Options

Customerin.txt:



```

eren yaeger;shiganshina district;1
annie leonhart;marley nation;2
neon nostrade;yorkshin city;10
chrollo lucilfer;meteor city;7
cheadle yorkshire; hunter association; 7|
meurem;east gorteau;3
gyro;ngl;5
zeno zoldyck;kukuroo mountain;4
beyond netero;dark continent;6

```

Import (Option 16):

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 16
Enter file name: customerin.txt
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.
com.microsoft.sqlserver.jdbc.SQLServerException: The statement did not return a result set.

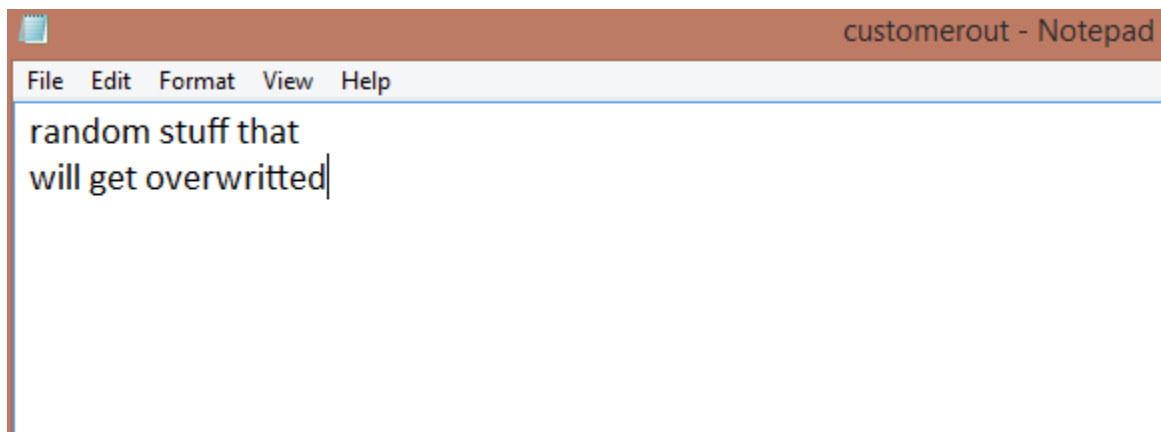
```

Customer Table after:

	cname	caddress	category
1	annie leonhart	marley nation	2
2	beyond netero	dark continent	6
3	cheadle yorksh...	hunter associa...	7
4	chrollo lucilf...	meteor city	7
5	eren yaeger	shiganshina di...	1
6	gon freecs	whale island	1
7	gyro	ngl	5
8	killua zoldyck	kukuroo mounta...	10
9	kurapika	kurta village	5
-	leorio paladik...	hunter univers...	3
-	meurem	east gorteau	3
-	neon nostrade	yorkshin city	10
-	pariston hill	hunter associa...	8
-	zeno zoldyck	kukuroo mounta...	4
-	NULL	NULL	NULL

## 6.17 Export Options

Customerout.txt:



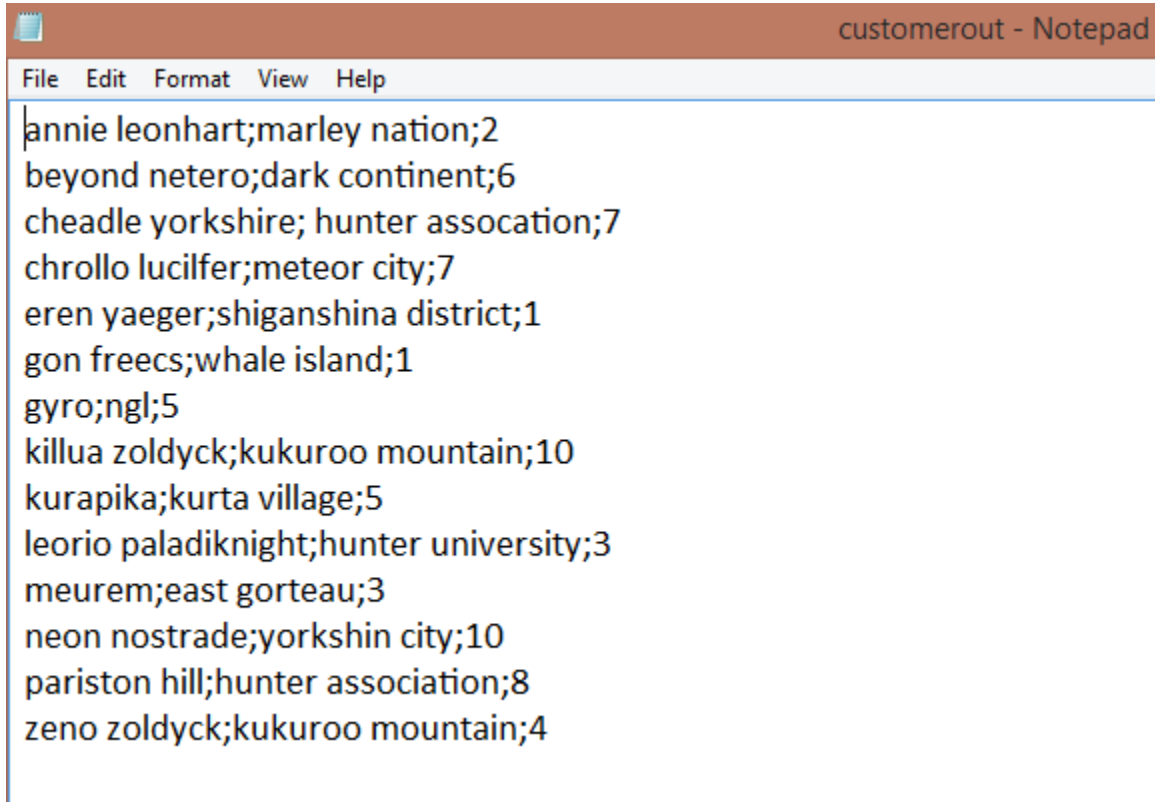
Export (Option 17):

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 17
Enter file name: customerout.txt
Enter Customer category lower bound(1-10): 1
Enter Customer category upper bound(1-10, can be the same): 10
WELCOME TO THE JOB-SHOP ACCOUNTING DATABASE SYSTEM

```

Customerout.txt After:



```

annie leonhart;marley nation;2
beyond netero;dark continent;6
cheadle yorkshire; hunter association;7
chrollo lucilfer;meteor city;7
eren yaeger;shiganshina district;1
gon freecs;whale island;1
gyro;ngl;5
killua zoldyck;kukuroo mountain;10
kurapika;kurta village;5
leorio paladiknight;hunter university;3
meurem;east gorteau;3
neon nostrade;yorkshin city;10
pariston hill;hunter association;8
zeno zoldyck;kukuroo mountain;4

```

Customer Table After (Unchanged):



	cname	caddress	category
1	annie leonhart	marley nation	2
2	beyond netero	dark continent	6
3	cheadle yorksh...	hunter associa...	7
4	chrollo lucilf...	meteor city	7
5	eren yaeger	shiganshina di...	1
6	gon freecs	whale island	1
7	gyro	ngl	5
8	killua zoldyck	kukuroo mounta...	10
9	kurapika	kurta village	5
-	leorio paladik...	hunter univers...	3
-	meurem	east gorteau	3
-	neon nostrade	yorkshin city	10
-	pariston hill	hunter associa...	8
-	zeno zoldyck	kukuroo mounta...	4
-	NULL	NULL	NULL

## 6.18 Errors

1) Making a process and associating with a non-existence department number (Query 4)

```

Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 4
Enter new process ID: 999
Enter department number for this process: 999
Enter type of process{Cut(1), Paint(2), Fit(3)}: 3
com.microsoft.sqlserver.jdbc.SQLServerException: Department number not found

```

Process Table (Unchanged):

	pid	pdata
1	1	Fit
2	2	Fit
3	3	Paint
4	4	Cut
5	5	Cut
6	6	Paint
7	9	Fit
8	10	Paint
9	99	Cut
—	100	Fit
—	NULL	NULL

Department Table(Unchanged, although this table isn't really affected at all):

	dnum	ddata
1	1	process 1/2/3
2	2	process 4/5/6
3	3	TBA
4	10	process 99/100
5	100	process 10
6	NULL	NULL

Supervise Table (Unchanged):

	pid	dnum
1	1	1
2	2	1
3	3	1
4	4	2
5	5	2
6	6	2
7	9	3
8	99	10
9	100	10
—	10	100
—	NULL	NULL

Fit\_process (Unchanged, but I show this because I chose fit-type):

	pid
1	1
2	2
3	9
4	100
5	NULL

2) Create a new customer for Customer table with category not between 1 to 10 (Query 1)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 1
Enter customer name: hlaaka sorrow
Enter customer address: unknown
Enter customer category: 15
com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint "category_range_check".
```

Customer Table (Unchanged):

	cname	caddress	category
1	annie leonhart	marley nation	2
2	beyond netero	dark continent	6
3	theadle yorksh...	hunter associa...	7
4	chrollo lucilf...	meteor city	7
5	eren yaeger	shiganshina di...	1
6	gon freecs	whale island	1
7	gyro	ngl	5
8	killua zoldyck	kukuroo mounta...	10
9	kurapika	kurta village	5
-	leorio paladik...	hunter univers...	3
-	meurem	east gorteau	3
-	neon nostrade	yorkshin city	10
-	pariston hill	hunter associa...	8
-	zeno zoldyck	kukuroo mounta...	4
-	NULL	NULL	NULL

(Doesn't have hisoka morrow)

3) Insert a new job with invalid date for date\_commence (Query 6)

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 6
Enter a Job number: 999
Enter associated Assembly ID: 1
Enter associated Process ID: 1
Enter day commenced(DD): 50
Enter month commenced(MM): 50
Enter year commenced(YYYY): 20005
com.microsoft.sqlserver.jdbc.SQLServerException: Error converting data type varchar to date.
```

Job table (Unchanged):

	jnum	date_commence	date_complete	aid	pid	info
1	1	1999-02-03	1999-03-01	2	2	Fit
2	2	1992-10-01	1993-10-10	1	10	Paint
3	3	1992-02-05	1999-03-01	5	2	Fit
4	5	1999-02-01	1999-03-01	2	10	Paint
5	7	1992-10-07	1993-10-10	7	10	Paint
6	9	1999-02-02	1999-03-02	2	2	Fit
7	10	1999-02-01	1999-03-01	1	2	Fit
8	NULL	NULL	NULL	NULL	NULL	NULL

(No job number 999)

## 6.19 Quit Option

Quit/Fin (Option 18):

```
Enter choices 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18: 18
Fin
```

## Task 7 Web Application

### 7.1 Source Program

Data Handler getDB:

```
private void getDBConnection() throws SQLException {
    if (conn != null) {
        return;
    }

    this.conn = DriverManager.getConnection(url);
}
```

Data Handler addCustomer:

```
public boolean addCustomer (String cname, String address, int category) throws SQLException {
    getDBConnection();

    final Statement statement = conn.createStatement();

    final String insertCustomer = "EXEC insert_customer '" + cname + "', '" + address + "', '" + category + "'";

    return(statement.executeUpdate(insertCustomer) == 1);
}
```

Data Handler getCustomer:

```

public ResultSet getCustomer(String lb, String ub) throws SQLException {
    getDBConnection();

    final Statement statement = conn.createStatement();

    final String getCustomer = "EXEC get_customer '" + lb + "', '" + ub + "'";

    return(statement.executeQuery(getCustomer));
}
}

```

Get Customer form:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Get From Customer Table</title>
</head>
<body>
    <h2>Get Customer from the Job-Shop Accounting Database System</h2>
    <!--
        Form for collecting user input to get the customer record
        Upon form submission, IP_Task7_Le_Bao_get_customer.jsp file will be invoked.
    -->

```

```

<form action="IP_Task7_Le_Bao_get_customer.jsp">
  <!-- The form organized in an HTML table for better clarity. -->
  <table border=1>
    <tr>
      <th colspan="2">Enter category lower and upper bound:</th>
    </tr>
    <tr>
      <td>Category lower bound(1-10) :</td>
      <td><div style="text-align: center;">
        <input type=text name=lb>
      </div></td>
    </tr>
    <tr>
      <td>Category upper bound(1-10, can be the same) :</td>
      <td><div style="text-align: center;">
        <input type=text name=ub>
      </div></td>
    </tr>
    <tr>
      <td><div style="text-align: center;">
        <input type=reset value=Clear>
      </div></td>
      <td><div style="text-align: center;">
        <input type=submit value=Get>
      </div></td>
    </tr>
  </table>
</form>
</body>
</html>

```

Insert Customer form:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Fill Customer Table</title>
  </head>
  <body>
    <h2>Add Customer to the Job-Shop Accounting Database System</h2>
    <!--
      Form for collecting user input for the customer record
      Upon form submission, IP_Task7_Le_Bao_insert_customer.jsp file will be invoked.
    -->
    <form action="IP_Task7_Le_Bao_insert_customer.jsp">
      <!-- The form organized in an HTML table for better clarity. -->
      <table border=1>
        <tr>
          <th colspan="2">Enter the Customer data:</th>
        </tr>
        <tr>
          <td>Customer name:</td>
          <td><div style="text-align: center;">
            <input type=text name=cname>
          </div></td>
        </tr>
        <tr>
          <td>Address:</td>
          <td><div style="text-align: center;">
            <input type=text name=address>
          </div></td>
        </tr>
        <tr>
          <td>Category(1-10) :</td>
          <td><div style="text-align: center;">
            <input type=text name=category>
          </div></td>
        </tr>
        <tr>
          <td><div style="text-align: center;">
            <input type=reset value=Clear>
          </div></td>
          <td><div style="text-align: center;">
            <input type=submit value=Insert>
          </div></td>
        </tr>
      </table>
    </form>
  </body>
</html>

```

Get Customer page that link with Java code:



```

<body>
<%@page import="ProjectTask7.IP_Task7_Le_Bao_DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
IP_Task7_Le_Bao_DataHandler handler = new IP_Task7_Le_Bao_DataHandler();

// Get the attribute values passed from the input form.
String lb = request.getParameter("lb");
String ub = request.getParameter("ub");

/*
 * If the user hasn't filled out lower or upper bound.
 */

if (ub.equals("") || lb.equals("")) {
    response.sendRedirect("IP_Task7_Le_Bao_insert_customer_form.jsp");
} else {

    final ResultSet rs = handler.getCustomer(lb, ub);

%>
<table cellpadding="2" cellspacing="2" border="1">
    <tr> <!-- The table headers row -->
        <td align="center">
            <h4>Customer name</h4>
        </td>
        <td align="center">
            <h4>Customer Address</h4>

```

```

        <h4>Customer Address</h4>
    </td>
    <td align="center">
        <h4>Customer Category</h4>
    </td>
</tr>
<%
while(rs.next()) { // For each customer record returned...
    // Extract the attribute values for every row returned
    final String cname = rs.getString(1);
    final String caddress = rs.getString(2);
    final String category = rs.getString(3);

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

<a href="IP_Task7_Le_Bao_insert_customer_form.jsp">Insert customer.</a><br>
<a href="IP_Task7_Le_Bao_get_customer_form.jsp">Retrieve customer.</a>
<%
}
%>
</body>

```

Insertion Customer page that link with Java code:

```

<body>
<%@page import="ProjectTask7.IP_Task7_Le_Bao_DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
IP_Task7_Le_Bao_DataHandler handler = new IP_Task7_Le_Bao_DataHandler();

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

/*
 * If the user hasn't filled out name, category, or address.
 */
if (cname.equals("") || address.equals("") || cat.equals("")) {
    response.sendRedirect("IP_Task7_Le_Bao_insert_customer_form.jsp");
} else {
    int category = Integer.parseInt(cat);

    // Now perform the query with the data from the form.
    boolean success = handler.addCustomer(cname, address, category);
    if (!success) { // Something went wrong
        %>
        <h2>There was a problem inserting into the table</h2>
        <%

    } else { // Confirm success to the user
        %>
        <h2>Customer:</h2>

        <ul>
            <li>Name: <%= cname %></li>
            <li>Address: <%= address %></li>
            <li>Category: <%= category %></li>
        </ul>

        <h2>Was successfully inserted.</h2>

        <a href="IP_Task7_Le_Bao_get_customer_form.jsp">Retrieve customer.</a><br>
        <a href="IP_Task7_Le_Bao_insert_customer_form.jsp">Insert customer.</a>
        <%
    }
}
%>
</body>

```

## 7.2 Successful Compilation

Query 1:

← → 🚫 🔄 http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_insert\_customer\_form.jsp

### Add Customer to the Job-Shop Accounting Database System

Enter the Customer data:	
Customer name:	<input type="text"/>
Address:	<input type="text"/>
Category(1-10):	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

Query 13:

← → 🚫 🔄 http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_get\_customer\_form.jsp

### Get Customer from the Job-Shop Accounting Database System

Enter category lower and upper bound:	
Category lower bound(1-10):	<input type="text"/>
Category upper bound(1-10, can be the same):	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="Get"/>

### 7.3 Web application testing

Query 13:

← → 🚫 🔄 http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_get\_customer\_form.jsp

### Get Customer from the Job-Shop Accounting Database System

Enter category lower and upper bound:	
Category lower bound(1-10):	<input type="text" value="1"/>
Category upper bound(1-10, can be the same):	<input type="text" value="5"/> ×
<input type="button" value="Clear"/>	<input type="button" value="Get"/>

http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_get\_customer.jsp?lb=1&ub=5

Customer name	Customer Address	Customer Category
annie leonhart	marley nation	2
eren yaeger	shiganshina district	1
gon freecs	whale island	1
gyro	ngl	5
kurapika	kurta village	5
leorio paladiknight	hunter university	3
meurem	east gorteau	3
zeno zoldyck	kukuroo mountain	4

[Insert customer.](#)

[Retrieve customer.](#)

Query 1:

http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_insert\_customer\_form.jsp

## Add Customer to the Job-Shop Accounting Database System

Enter the Customer data:	
Customer name:	<input type="text" value="hisoka morrow"/>
Address:	<input type="text" value="unknown"/>
Category(1-10):	<input type="text" value="4"/> ×
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_insert\_customer.jsp?cname=hisoka+morrow&address=unknown&category=4

### Customer:

- Name: hisoka morrow
- Address: unknown
- Category: 4

Was successfully inserted.

[Retrieve customer.](#)

[Insert customer.](#)

Query 13:

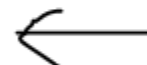
http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_get\_customer\_form.jsp

### Get Customer from the Job-Shop Accounting Database System

Enter category lower and upper bound:	
Category lower bound(1-10):	<input type="text" value="1"/>
Category upper bound(1-10, can be the same):	<input type="text" value="5"/>
<input type="button" value="Clear"/>	<input type="button" value="Get"/>

http://localhost:8080/ProjectTask7/IP\_Task7\_Le\_Bao\_get\_customer.jsp?lb=1&ub=5

Customer name	Customer Address	Customer Category
annie leonhart	marley nation	2
eren yaeger	shiganshina district	1
gon freecs	whale island	1
gyro	ngl	5
hisoka morrow	unknown	4
kurapika	kurta village	5
leorio paladiknight	hunter university	3
meurem	east gorteau	3
zeno zoldyck	kukuroo mountain	4



[Insert customer.](#)

[Retrieve customer.](#)

