

Name: Shahzad Ansari

Student ID: ansa0005

E-mail address: Shahzad.S.Ansari-1@ou.edu

Course: CS/DSA 4513

Section: Online Sections 995-999

Semester: Fall 2021

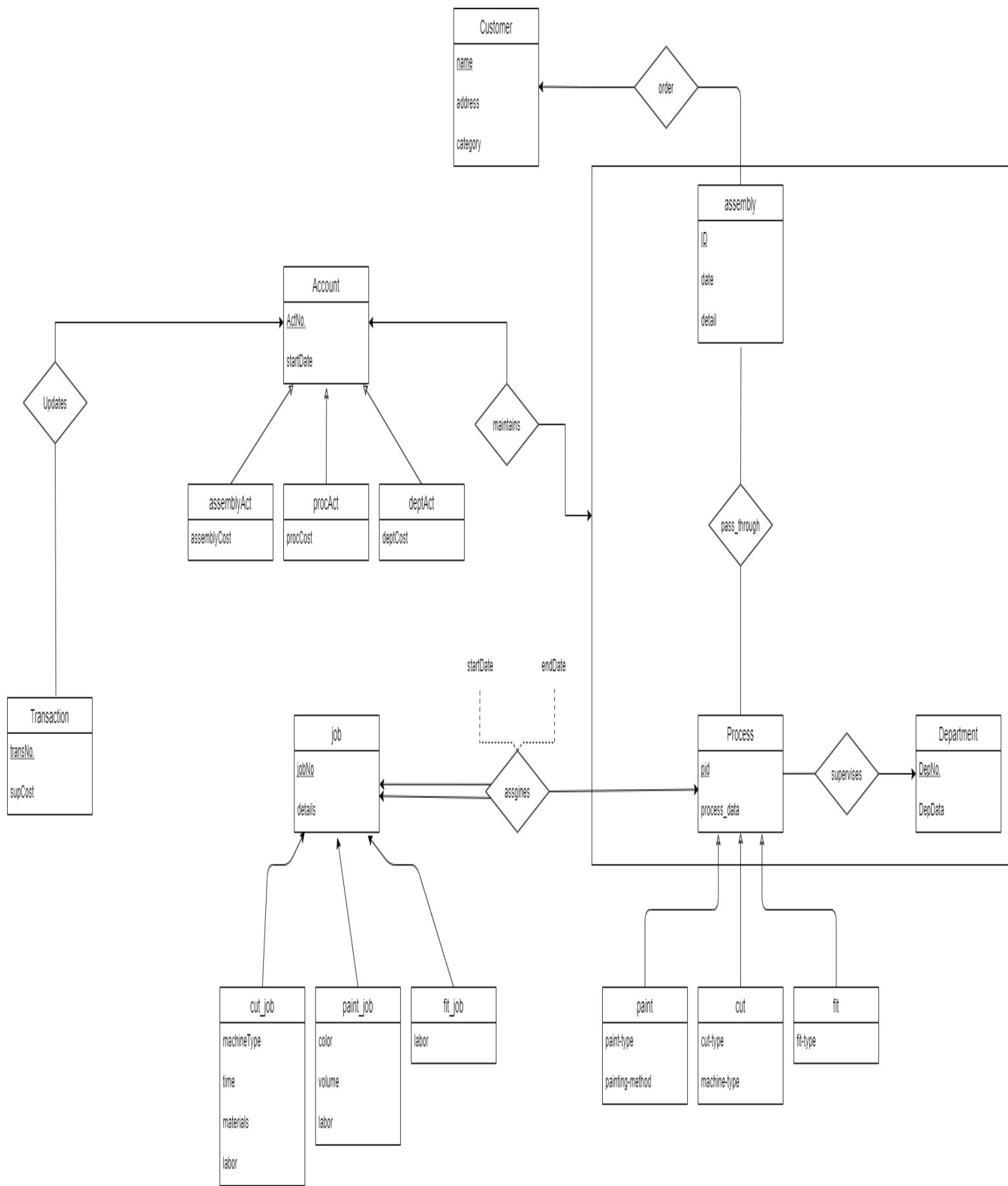
Instructor: Dr.Le Gruenwald

Project Title: JOB-SHOP ACCOUNTING SYSTEM

TABLE OF CONTENTS

Task	Subtask	Name	Page
1		ERD and Relational Database	3-4
	1.1	ERD	3
	1.2	RDB Schema	4
2		Data Dictionary	5-8
	2.1	Data Dictionary	5
3		Storage Structure	9-12
	3.1	Discussion of storage structures for tables	9
4		Creation of SQL tables	13-33
	4.1	SQL create tables	13
5		Transact SQL and Java	34-82
	5.1	TSQL Procedures and Execution	34
	5.2	Java Code	61
6		Testing Queries in Java	83-135
	6.1	Query 1	83
	6.2	Query 2	84
	6.3	Query 3	85-89
	6.4	Query 4	90-95
	6.5	Query 5	96-100
	6.6	Query 6	101-103
	6.7	Query 7	104-109
	6.8	Query 8	110-113
	6.9	Query 9	114-115
	6.10	Query 10	116-117
	6.11	Query 11	118-119
	6.12	Query 12	120-121
	6.13	Query 13	122-123
	6.14	Query 14	124-128
	6.15	Query 15	129

	6.16	Import customers from file	130- 131
	6.17	Export Customers to file	132
	6.18	Exit java program	134
	6.19	Error Examples	135
7		Running JSP and example proofs	136- 144
	7.1	Data handler	136- 137
	7.2	Get customer HTML	138- 139
	7.3	Get customer JSP	139- 140
	7.4	Insert Customer HTML	141- 142
	7.5	Insert Customer JSP	142- 143
	7.6	Execution Pictures	144- 145



Relational Database:

1. customer(name,address,category)
2. asmby(asmID,orderDate,details)
3. cust_order(asmID,name)
4. process(pid)
5. pass_through(pid,asmID)
6. dept(deptId,deptData)
7. account(actNo)
8. assemblyAct(actNo,startDate,assemblyCost)
9. assignes(startDate,endDate,jobNo,pid,asmld)
- 10.cutJob(jobId,details,materials,machineType,labor,timeTaken)
- 11.cutProcess(pid,process_data,cutType,machineType)
- 12.deptAct(actNo,startDate,deptCost)
- 13.fitJob(jobId,details,labor)
- 14.fitProcess(pid,process_data,fitType)
- 15.job(jobId)
- 16.jobTransaction(transNo,supCost)
- 17.maintains(pid,asmID,deptId,actNo)
- 18.paintjob(jobId,details,color,labor,volume)
- 19.paintProcess(pid,process_data,paintType,paintingMethod)
- 20.pass_through(pid,asmID)
- 21.procAct(actNo,startDate,procCost)
- 22.supervises(pid,deptId)
- 23.updates(transNo,actNo)

Data Dictionary

Table Name	keys	column	Type and size	nullable	Constraints
account	PK	actNo	int(10)	NOT NULL	
asmby	PK	asmID	int(10)	NOT NULL	
		orderDate	date(0)	NULL	
		details	varchar(64)	NULL	
assemblyAct	FK,PK	actNo	int(10)	NOT NULL	FK references asmby
		startDate	date(0)	NULL	
		assemblyCost	int(10)	NULL	
assigues		startDate	date(0)	NULL	
		endDate	date(0)	NULL	
	FK,PK	jobNo	int(10)	NOT NULL	References job
	FK,PK	pid	int(10)	NOT NULL	References process
	FK	asmID	int(10)	NULL	References asmby
cust_order	FK,PK	asmID	int(10)	NOT NULL	References asmby
	FK	name	varchar(64)	NULL	References customer
customer	PK	name	varchar(64)	NOT NULL	
		address	varchar(64)	NULL	
		category	varchar(64)	NULL	Must be in range 1 to 10

cutJob	FK,PK	jobId	int(10)	NOT NULL	References job
		details	varchar(64)	NULL	
		materials	varchar(64)	NULL	
		machineType	varchar(64)	NULL	
		labor	int(10)	NULL	
		timeTaken	int(10)	NULL	
cutProcess	FK,PK	pid	int(10)	NOT NULL	References job
		process_data	varchar(64)	NULL	
		cutType	varchar(64)	NULL	
		machineType	varchar(64)	NULL	
dept	PK	deptId	int(10)	NOT NULL	
		deptData	varchar(64)	NULL	
deptAct	FK,PK	actNo	int(10)	NOT NULL	References account
		startDate	date(0)	NULL	
		deptCost	int(10)	NULL	
fitJob	FK,PK	jobId	int(10)	NOT NULL	References job
		details	varchar(64)	NULL	
		labor	int(10)	NULL	
fitProcess	FK,PK	pid	int(10)	NOT NULL	References process
		process_data	varchar(64)	NULL	
		fitType	varchar(64)	NULL	
job	PK	jobId	int(10)	NOT NULL	
jobTransaction	PK	transNo	int(10)	NOT NULL	
		supCost	int(10)	NULL	

maintains	FK	pid	int(10)	NULL	References process
	FK	asmID	int(10)	NULL	References assembly
	FK	deptId	int(10)	NULL	References dept
	FK,PK	actNo	int(10)	NOT NULL	References account
paintJob	FK,PK	jobId	int(10)	NOT NULL	References job
		details	varchar(64)	NULL	
		color	varchar(64)	NULL	
		labor	int(10)	NULL	
		volume	int(10)	NULL	
paintProcess	FK,PK	pid	int(10)	NOT NULL	References process
		process_data	varchar(64)	NULL	
		paintType	varchar(64)	NULL	
		paintingMethod	varchar(64)	NULL	
pass_through	FK	pid	int(10)	NULL	References process
	FK	asmID	int(10)	NULL	References assembly
procAct	FK,PK	actNo	int(10)	NOT NULL	References account
		startDate	date(0)	NULL	
		procCost	int(10)	NULL	
process	PK	pid	int(10)	NOT NULL	
supervises	FK,PK	pid	int(10)	NOT NULL	References process

	FK	deptId	int(10)	NULL	References dept
updates	FK,PK	transNo	int(10)	NOT NULL	References transaction
	FK	actNo	int(10)	NULL	References account

Storage Structure

Table name	Query#	Type	Key	Frequency	File organization	Justification
customer	1	Insertion	Name	30/day	Unique clustered	Autogenerated index by SQL
dept	2	Insertion	deptId	Infrequent	Unique Clustered	Autogenerated Index by SQL
dept	3	Insertion	deptId	40/day	Unique Clustered	Autogenerated Index by SQL
process	3	Insertion	pid	40/day	Unique Clustered	Autogenerated Index by SQL
supervises	3	Insertion	pid	40/day	Unique Clustered	Autogenerated Index by SQL
cutProcess	3	Insertion	pid	40/day	Non-Unique Non-Clustered	Used for retrieval by other queries
paintProcess	3	Insertion	pid	40/day	Non-Unique Non-Clustered	Used for retrieval by other queries
fitProcess	3	Insertion	pid	40/day	Non-Unique Non-Clustered	Used for retrieval by other queries
cust_order	4	Insertion	asmID	40/day	Non-Unique Non-Clustered	Index on asmID because a name may have multiple assemblies
asmby	4	Insertion	asmID	40/day	Non-Unique Non-Clustered	When linking tables with asmby, they are linked on asmID
customer	4	Insertion	name	40/day	Unique clustered	Autogenerated index by SQL
pass_through	4	Insertion	asmID	40/day	Non-Unique Non-Clustered	When linking tables with asmby, they are linked on asmID
account	5	Insertion	actNo	10/day	Unique clustered	Autogenerated index by SQL

maintains	5	Insertion	actNo	10/day	Non-Unique Non-Clustered	When searching what type of account an actNo is it makes sense to order by actNo
assemblyAct	5	Insertion	actNo	10/day	Non-Unique Non-Clustered	When searching for account info index by actNo
deptAct	5	Insertion	actNo	10/day	Non-Unique Non-Clustered	When searching for account info index by actNo
procAct	5	Insertion	actNo	10/day	Non-Unique Non-Clustered	When searching for account info index by actNo
job	6	Insertion	jobID	50/day	Unique clustered	Autogenerated index by SQL
assignes	6	Insertion	jobNo,pid	50/day	Unique clustered	Autogenerated index by SQL
assignes	7	Insertion	jobNo,pid	50/day	Unique clustered	Autogenerated index by SQL
paintJob	7	Insertion	jobId	50/day	Unique clustered	Autogenerated index by SQL
cutJob	7	Insertion	jobId	50/day	Unique clustered	Autogenerated index by SQL
fitJob	7	Insertion	jobId	50/day	Unique clustered	Autogenerated index by SQL
updates	8	Insertion	transNo	50/day	Unique clustered	Autogenerated index by SQL
account	8	Insertion	actNo	50/day	Unique clustered	Autogenerated index by SQL
assemblyAct	8	Insertion	actNo	50/day	Non-Unique Non-Clustered	When retrieving an assembly account it makes sense to order by actNo

procAct	8	Insertion	actNo	50/day	Non-Unique Non-Clustered	When retrieving an assembly account it makes sense to order by actNo
deptAct	8	Insertion	actNo	50/day	Non-Unique Non-Clustered	When retrieving an assembly account it makes sense to order by actNo
assemblyAct	9	Random search	actNo	200/day	Unique clustered	Autogenerated index by SQL Since the pk is being indexed no need to add a extra index
maintains	9	Random search	asmID	200/day	Unique clustered	Autogenerated index by SQL Since the pk is being indexed no need to add a extra index
supervises	10	Random search	pid	20/day	Unique Clustered	Autogenerated Index by SQL
assigns	10	Random search	jobNo,pid	20/day	Unique Clustered	Autogenerated Index by SQL
paintJob	10	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
cutJob	10	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
fitJob	10	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
asmby	11	Random search	asmID	100/day	Non-Unique Non-Clustered	When populating pass_through it makes sense to order by asmID
pass_through	11	Random search	asmID	100/day	Non-Unique Non-Clustered	When linking tables with asmby, they are linked on asmID
supervises	11	Random search	pid	100/day	Unique Clustered	Autogenerated Index by SQL

assignes	12	Random search	jobNo,pid	20/day	Unique Clustered	Autogenerated Index by SQL
supervises	12	Random search	pid	20/day	Unique Clustered	Autogenerated Index by SQL
assignes	12	Random search	jobNo,pid	20/day	Unique Clustered	Autogenerated Index by SQL
jobs	12	Random search	jobID	20/day	Unique clustered	Autogenerated index by SQL
cutJob	12	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
fitJob	12	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
paintJob	12	Random search	jobId	20/day	Unique Clustered	Autogenerated Index by SQL
customer	13	Range search	category	100/day	Non-Unique Non-Clustered	Since we were searching within a range, we should sort by category
job	14	Random search	jobID	1/month	Unique clustered	Autogenerated index by SQL
assigns	14	Random search	jobNo,pid	1/month	Unique Clustered	Autogenerated Index by SQL
cutJob	14	Random search	jobId	1/month	Unique Clustered	Autogenerated Index by SQL
paintJob	15	Deletion	jobId	1/week	Unique Clustered	Autogenerated Index by SQL

CREATE TABLES

1. account

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[account](
    [actNo] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[account] ADD PRIMARY KEY CLUSTERED
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
```

2. asmbly

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[asmbly](
    [asmID] [int] NOT NULL,
    [orderDate] [date] NULL,
    [details] [varchar](64) NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[asmbly] ADD PRIMARY KEY CLUSTERED
(
    [asmID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
```

```
CREATE NONCLUSTERED INDEX [asmblyID] ON [dbo].[asmbly]
(
    [asmID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
```

3. customer

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[customer](
    [name] [varchar](64) NOT NULL,
    [address] [varchar](64) NULL,
    [category] [varchar](64) NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
ALTER TABLE [dbo].[customer] ADD PRIMARY KEY CLUSTERED
(
    [name] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
SET ANSI_PADDING ON
GO
CREATE NONCLUSTERED INDEX [customerCategoryIdx] ON
[dbo].[customer]
(
    [category] ASC
```

```
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =  
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
GO  
ALTER TABLE [dbo].[customer] WITH CHECK ADD CHECK  
(([category]>=(1) AND [category]<=(10)))  
GO
```

4. dept

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[dept](  
    [deptId] [int] NOT NULL,  
    [deptData] [varchar](64) NULL  
) ON [PRIMARY]  
GO  
ALTER TABLE [dbo].[dept] ADD PRIMARY KEY CLUSTERED  
(  
    [deptId] ASC  
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =  
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
GO
```

5. job

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[job](
    [jobId] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[job] ADD PRIMARY KEY CLUSTERED
(
    [jobId] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
```

6. process

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[process](
    [pid] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[process] ADD PRIMARY KEY CLUSTERED
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
```

```
GO
```

7. jobTransaction

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[jobTransaction](
    [transNo] [int] NOT NULL,
    [supCost] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[jobTransaction] ADD PRIMARY KEY
CLUSTERED
(
    [transNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
```

8. assmeblyAct

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[assemblyAct](
    [actNo] [int] NOT NULL,
    [startDate] [date] NULL,
    [assemblyCost] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[assemblyAct] ADD PRIMARY KEY
CLUSTERED
```

```

(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [actNo] ON [dbo].[assemblyAct]
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[assemblyAct] WITH CHECK ADD
CONSTRAINT [FK_assemblyAct_actNo] FOREIGN KEY([actNo])
REFERENCES [dbo].[account] ([actNo])
GO
ALTER TABLE [dbo].[assemblyAct] CHECK CONSTRAINT
[FK_assemblyAct_actNo]
GO

```

9. assigues

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[assigues](
    [startDate] [date] NULL,
    [endDate] [date] NULL,
    [jobNo] [int] NOT NULL,
    [pid] [int] NOT NULL,
    [asmID] [int] NULL
) ON [PRIMARY]
GO

```

```
ALTER TABLE [dbo].[assinges] ADD CONSTRAINT
[PK_assinges] PRIMARY KEY CLUSTERED
(
    [jobNo] ASC,
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[assinges] WITH CHECK ADD CONSTRAINT
[FK_assinges_asmID] FOREIGN KEY([asmID])
REFERENCES [dbo].[asmbly] ([asmID])
GO
ALTER TABLE [dbo].[assinges] CHECK CONSTRAINT
[FK_assinges_asmID]
GO
ALTER TABLE [dbo].[assinges] WITH CHECK ADD CONSTRAINT
[FK_assinges_JobNo] FOREIGN KEY([jobNo])
REFERENCES [dbo].[job] ([jobId])
GO
ALTER TABLE [dbo].[assinges] CHECK CONSTRAINT
[FK_assinges_JobNo]
GO
ALTER TABLE [dbo].[assinges] WITH CHECK ADD CONSTRAINT
[FK_assinges_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[assinges] CHECK CONSTRAINT
[FK_assinges_pid]
GO
```

10. cust_order

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[cust_order](
    [asmID] [int] NOT NULL,
    [name] [varchar](64) NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cust_order] ADD PRIMARY KEY CLUSTERED
(
    [asmID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [Cust_OrderIDX] ON
[dbo].[cust_order]
(
    [asmID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[cust_order] WITH CHECK ADD
CONSTRAINT [FK_asmID] FOREIGN KEY([asmID])
REFERENCES [dbo].[asmbly] ([asmID])
GO
ALTER TABLE [dbo].[cust_order] CHECK CONSTRAINT
[FK_asmID]
GO
ALTER TABLE [dbo].[cust_order] WITH CHECK ADD
CONSTRAINT [FK_name] FOREIGN KEY([name])
REFERENCES [dbo].[customer] ([name])
```

```
GO  
ALTER TABLE [dbo].[cust_order] CHECK CONSTRAINT  
[FK_name]  
GO
```

11. deptAct

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[deptAct](  
    [actNo] [int] NOT NULL,  
    [startDate] [date] NULL,  
    [deptCost] [int] NULL  
) ON [PRIMARY]  
GO  
ALTER TABLE [dbo].[deptAct] ADD PRIMARY KEY CLUSTERED  
(  
    [actNo] ASC  
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =  
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
GO  
CREATE NONCLUSTERED INDEX [actNo] ON [dbo].[deptAct]  
(  
    [actNo] ASC  
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =  
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON  
[PRIMARY]  
GO  
ALTER TABLE [dbo].[deptAct] WITH CHECK ADD CONSTRAINT  
[FK_dept_actNo] FOREIGN KEY([actNo])  
REFERENCES [dbo].[account] ([actNo])  
GO  
ALTER TABLE [dbo].[deptAct] CHECK CONSTRAINT  
[FK_dept_actNo]
```

GO

12. fitJob

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[fitJob](
    [jobId] [int] NOT NULL,
    [details] [varchar](64) NULL,
    [labor] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[fitJob] ADD PRIMARY KEY CLUSTERED
(
    [jobId] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[fitJob] WITH CHECK ADD CONSTRAINT
[FK_fitJob_jobId] FOREIGN KEY([jobId])
REFERENCES [dbo].[job] ([jobId])
GO
ALTER TABLE [dbo].[fitJob] CHECK CONSTRAINT
[FK_fitJob_jobId]
GO
```

13. cutProcess

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[cutProcess](
    [pid] [int] NOT NULL,
    [process_data] [varchar](64) NULL,
    [cutType] [varchar](64) NULL,
    [machineType] [varchar](64) NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cutProcess] ADD PRIMARY KEY CLUSTERED
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [cutProcessIDX] ON
[dbo].[cutProcess]
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[cutProcess] WITH CHECK ADD
CONSTRAINT [FK_cutProcess_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[cutProcess] CHECK CONSTRAINT
[FK_cutProcess_pid]
GO
```

14. cutJob

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[cutJob](
    [jobId] [int] NOT NULL,
    [details] [varchar](64) NULL,
    [materials] [varchar](64) NULL,
    [machineType] [varchar](64) NULL,
    [labor] [int] NULL,
    [timeTaken] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[cutJob] ADD PRIMARY KEY CLUSTERED
(
    [jobId] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[cutJob] WITH CHECK ADD CONSTRAINT
[FK_cutJob_jobId] FOREIGN KEY([jobId])
REFERENCES [dbo].[job] ([jobId])
GO
ALTER TABLE [dbo].[cutJob] CHECK CONSTRAINT
[FK_cutJob_jobId]
GO
```

15. pass_through

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[pass_through](
    [pid] [int] NULL,
    [asmID] [int] NULL
) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [PassThroughIDX] ON
[dbo].[pass_through]
(
    [asmID] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[pass_through] WITH CHECK ADD
CONSTRAINT [FK_passThrough_asmID] FOREIGN KEY([asmID])
REFERENCES [dbo].[asmbly] ([asmID])
GO
ALTER TABLE [dbo].[pass_through] CHECK CONSTRAINT
[FK_passThrough_asmID]
GO
ALTER TABLE [dbo].[pass_through] WITH CHECK ADD
CONSTRAINT [FK_passThrough_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[pass_through] CHECK CONSTRAINT
[FK_passThrough_pid]
GO
```

16. paintProcess

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[paintProcess](
    [pid] [int] NOT NULL,
    [process_data] [varchar](64) NULL,
    [paintType] [varchar](64) NULL,
    [paintingMethod] [varchar](64) NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[paintProcess] ADD PRIMARY KEY
CLUSTERED
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [paintProcessIDX] ON
[dbo].[paintProcess]
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[paintProcess] WITH CHECK ADD
CONSTRAINT [FK_paintProcess_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[paintProcess] CHECK CONSTRAINT
[FK_paintProcess_pid]
GO
```

17. paintJob

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[paintJob](
    [jobId] [int] NOT NULL,
    [details] [varchar](64) NULL,
    [color] [varchar](64) NULL,
    [labor] [int] NULL,
    [volume] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[paintJob] ADD PRIMARY KEY CLUSTERED
(
    [jobId] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[paintJob] WITH CHECK ADD CONSTRAINT
[FK_paintJob_jobId] FOREIGN KEY([jobId])
REFERENCES [dbo].[job] ([jobId])
GO
ALTER TABLE [dbo].[paintJob] CHECK CONSTRAINT
[FK_paintJob_jobId]
GO
```

18. maintains

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[maintains](
    [pid] [int] NULL,
    [asmID] [int] NULL,
    [deptId] [int] NULL,
    [actNo] [int] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[maintains] ADD CONSTRAINT
[PK_maintains] PRIMARY KEY CLUSTERED
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [actNo] ON [dbo].[maintains]
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[maintains] WITH CHECK ADD
CONSTRAINT [FK_maintains_actNo] FOREIGN KEY([actNo])
REFERENCES [dbo].[account] ([actNo])
GO
ALTER TABLE [dbo].[maintains] CHECK CONSTRAINT
[FK_maintains_actNo]
GO
```

```
ALTER TABLE [dbo].[maintains] WITH CHECK ADD
CONSTRAINT [FK_maintains_asmID] FOREIGN KEY([asmID])
REFERENCES [dbo].[asmbly] ([asmID])
GO
ALTER TABLE [dbo].[maintains] CHECK CONSTRAINT
[FK_maintains_asmID]
GO
ALTER TABLE [dbo].[maintains] WITH CHECK ADD
CONSTRAINT [FK_maintains_deptID] FOREIGN KEY([deptId])
REFERENCES [dbo].[dept] ([deptId])
GO
ALTER TABLE [dbo].[maintains] CHECK CONSTRAINT
[FK_maintains_deptID]
GO
ALTER TABLE [dbo].[maintains] WITH CHECK ADD
CONSTRAINT [FK_maintains_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[maintains] CHECK CONSTRAINT
[FK_maintains_pid]
GO
```

19. updates

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[updates](
    [transNo] [int] NOT NULL,
    [actNo] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[updates] ADD PRIMARY KEY CLUSTERED
(
    [transNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[updates] WITH CHECK ADD CONSTRAINT
[FK_updates_actNo] FOREIGN KEY([actNo])
REFERENCES [dbo].[account] ([actNo])
GO
ALTER TABLE [dbo].[updates] CHECK CONSTRAINT
[FK_updates_actNo]
GO
ALTER TABLE [dbo].[updates] WITH CHECK ADD CONSTRAINT
[FK_updates_transNo] FOREIGN KEY([transNo])
REFERENCES [dbo].[jobTransaction] ([transNo])
GO
ALTER TABLE [dbo].[updates] CHECK CONSTRAINT
[FK_updates_transNo]
GO
```

20. supervises

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[supervises](
    [pid] [int] NOT NULL,
    [deptId] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[supervises] ADD PRIMARY KEY CLUSTERED
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[supervises] WITH CHECK ADD
CONSTRAINT [FK_supervises_deptID] FOREIGN KEY([deptId])
REFERENCES [dbo].[dept] ([deptId])
GO
ALTER TABLE [dbo].[supervises] CHECK CONSTRAINT
[FK_supervises_deptID]
GO
ALTER TABLE [dbo].[supervises] WITH CHECK ADD
CONSTRAINT [FK_supervises_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[supervises] CHECK CONSTRAINT
[FK_supervises_pid]
GO
```

21. procAct

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[procAct](
    [actNo] [int] NOT NULL,
    [startDate] [date] NULL,
    [procCost] [int] NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[procAct] ADD PRIMARY KEY CLUSTERED
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
CREATE NONCLUSTERED INDEX [actNo] ON [dbo].[procAct]
(
    [actNo] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, DROP_EXISTING =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[procAct] WITH CHECK ADD CONSTRAINT
[FK_procAct_actNo] FOREIGN KEY([actNo])
REFERENCES [dbo].[account] ([actNo])
GO
ALTER TABLE [dbo].[procAct] CHECK CONSTRAINT
[FK_procAct_actNo]
GO
```

22. fitProcess

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[fitProcess](
    [pid] [int] NOT NULL,
    [process_data] [varchar](64) NULL,
    [fitType] [varchar](64) NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[fitProcess] ADD PRIMARY KEY CLUSTERED
(
    [pid] ASC
)WITH (STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ONLINE = OFF, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
GO
ALTER TABLE [dbo].[fitProcess] WITH CHECK ADD
CONSTRAINT [FK_fitProcess_pid] FOREIGN KEY([pid])
REFERENCES [dbo].[process] ([pid])
GO
ALTER TABLE [dbo].[fitProcess] CHECK CONSTRAINT
[FK_fitProcess_pid]
GO
```

PROCEDURES

```
***** Object: StoredProcedure  
[dbo].[Associate_assembly_customer]     Script Date:  
11/21/2021 5:49:08 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[Associate_assembly_customer]  
@asmID INTEGER,  
@name VARCHAR(64)  
  
AS  
BEGIN  
    INSERT INTO cust_order VALUES(@asmID,@name)  
END  
  
***** Object: StoredProcedure  
[dbo].[Associate_assembly_process]     Script Date:  
11/21/2021 5:49:08 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[Associate_assembly_process]  
@asmID INTEGER,  
@pid INTEGER  
  
AS  
BEGIN  
    INSERT INTO pass_through VALUES(@pid,@asmID)  
  
END  
--
```

```
***** Object:  StoredProcedure [dbo].[changePaint]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[changePaint]
```

```
@jobId INTEGER,
@color VARCHAR(64)
```

```
AS
BEGIN
    UPDATE paintJob set color = @color WHERE jobId =
@jobId
END
```

```
***** Object:  StoredProcedure [dbo].[completed_job]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[completed_job]
@jobNo INTEGER,
@endDate DATE
```

```
AS
BEGIN
    UPDATE assignes
    SET endDate = @endDate
    WHERE jobNo = @jobNo
END
```

```
***** Object: StoredProcedure [dbo].[createAccount]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[createAccount]
@actNo INTEGER
AS
BEGIN
    INSERT INTO account VALUES (@actNo)
END

***** Object: StoredProcedure
[dbo].[createAssemblyAccount]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[createAssemblyAccount]
@actNo INTEGER,
@asmID INTEGER,
@startDate DATE
AS
BEGIN
    INSERT INTO maintains VALUES
(NULL,@asmID,NULL,@actNo)
    INSERT INTO assemblyAct VALUES(@actNo,@startDate,0)
END
```

```
***** Object: StoredProcedure  
[dbo].[createDeptAccount]      Script Date: 11/21/2021  
5:49:08 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[createDeptAccount]  
@actNo INTEGER,  
@deptId INTEGER,  
@startDate DATE  
  
AS  
BEGIN  
    INSERT INTO maintains VALUES  
(NULL,NULL,@deptId,@actNo)  
    INSERT INTO deptAct VALUES(@actNo,@startDate,0)  
END  
  
***** Object: StoredProcedure  
[dbo].[createProcessAccount]      Script Date: 11/21/2021  
5:49:08 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[createProcessAccount]  
@actNo INTEGER,  
@pid INTEGER,  
@startDate DATE  
AS  
BEGIN  
    INSERT INTO maintains VALUES (@pid,NULL,NULL,@actNo)  
    INSERT INTO procAct VALUES(@actNo,@startDate,0)  
END
```

```
***** Object: StoredProcedure [dbo].[cut_Job]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[cut_Job]
@jobNo INTEGER,
@details VARCHAR(64),
@materials VARCHAR(64),
@machineType VARCHAR(64),
@labor INTEGER,
@timeTaken INTEGER

AS
BEGIN
    INSERT INTO cutJob
VALUES(@jobNo,@details,@materials,@machineType,@labor,@t
imeTaken)
END
```

```
***** Object: StoredProcedure [dbo].[cutProcessTable]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[cutProcessTable]
@pid INTEGER,
@process_data VARCHAR(64),
@cutType VARCHAR(64),
@machineType VARCHAR(64)
AS
BEGIN
```

```
        INSERT INTO cutProcess VALUES (
@pid,@process_data,@cutType,@machineType)
END

/******** Object: StoredProcedure [dbo].[deleteCutJob]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[deleteCutJob]

@startRange INTEGER,
@endRange INTEGER

AS
BEGIN
    DELETE FROM cutJob WHERE jobID >= @startRange AND
jobID <= @endRange
    DELETE FROM assignes WHERE jobNo >= @startRange AND
jobNo <= @endRange
    DELETE FROM job WHERE jobID >= @startRange AND jobID
<= @endRange
END

/******** Object: StoredProcedure [dbo].[fit_Job]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[fit_Job]
@jobNo INTEGER,
@details VARCHAR(64),
@labor INTEGER
```

```
AS
BEGIN
    INSERT INTO fitJob VALUES(@jobNo,@details,@labor)
END

/***** Object: StoredProcedure [dbo].[fitProcessTable]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[fitProcessTable]
@pid INTEGER,
@process_data VARCHAR(64),
@fitType VARCHAR(64)
AS
BEGIN
    INSERT INTO fitProcess VALUES
    (@pid,@process_data,@fitType)
END

/***** Object: StoredProcedure
[dbo].[getAssembyProccesHistory]      Script Date:
11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[getAssembyProccesHistory]
@asmID INTEGER
AS
BEGIN

    SELECT DISTINCT
```

```
a.asmID,j.startDate,p.pid,s.deptId  
FROM  
    asmbly a, pass_through p, supervises s,assignes j  
WHERE  
    a.asmID = @asmID  
AND  
    p.asmID = a.asmID  
AND  
    s.pid = p.pid  
AND  
    j.pid = p.pid  
ORDER BY startDate
```

```
END
```

```
***** Object:  StoredProcedure [dbo].[getCustomers]  
Script Date: 11/21/2021 5:49:08 PM *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[getCustomers]  
  
@startRange INTEGER,  
@endRange INTEGER  
  
AS  
BEGIN  
SELECT * FROM customer WHERE category >= @startRange AND  
category <= @endRange ORDER BY name  
END
```

```
***** Object: StoredProcedure [dbo].[getJobHistory]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[getJobHistory]
@deptId INTEGER,
@endDate DATE

AS
BEGIN

DECLARE @jobs TABLE(
    deptId INTEGER,
    pid INTEGER,
    asmID INTEGER,
    endDate DATE,
    jobNo INTEGER
)

INSERT INTO @jobs(deptId,pid,asmID,endDate,jobNo)
SELECT DISTINCT
    s.deptId, s.pid , a.asmID,a.endDate,a.jobNo
FROM
    supervises s, assignes a
WHERE
    s.pid = a.pid
    AND
    s.deptId = @deptId
    AND
    a.endDate = @endDate

DECLARE @paintTest INTEGER
set @paintTest =
```

```
(SELECT count(1) where exists(  SELECT
deptId,pid,asmID,endDate,jobId,details,color,labor,volum
e
FROM
@jobs j, paintJob p
WHERE
j.jobNo = p.jobId
)
)

DECLARE @cutTest INTEGER
set @cutTest =
(SELECT count(1) where exists(  SELECT
deptId,pid,asmID,endDate,jobId,details,materials,labor,m
achineType,timeTaken
FROM
@jobs j, cutJob c
WHERE
j.jobNo = c.jobId
)
)

DECLARE @fitTest INTEGER
set @fitTest =
(SELECT count(1) where exists(  SELECT
deptId,pid,asmID,endDate,jobId,details,labor
FROM
@jobs j, fitJob f
WHERE
j.jobNo = f.jobId
)
)

IF @paintTest > 0
```

```
PRINT'PAINT'

IF @paintTest > 0
    SELECT

deptId,pid,asmID,endDate,jobId,details,color,labor,volum
e
    FROM
        @jobs j, paintJob p
WHERE
    j.jobNo = p.jobId


IF @cutTest > 0
    PRINT'cut'

IF @cutTest > 0
    SELECT

deptId,pid,asmID,endDate,jobId,details,materials,labor,m
achineType,timeTaken
    FROM
        @jobs j, cutJob c
WHERE
    j.jobNo = c.jobId


IF @fitTest > 0
    PRINT'FIT'

IF @fitTest > 0
    SELECT
        deptId,pid,asmID,endDate,jobId,details,labor
    FROM
        @jobs j, fitJob f
WHERE
    j.jobNo = f.jobId
```

END

```
***** Object:  StoredProcedure
[dbo].[getJobHistory_cut]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[getJobHistory_cut]
@deptId INTEGER,
@endDate DATE
```

AS

BEGIN

```
DECLARE @jobs TABLE(
deptId INTEGER,
pid INTEGER,
asmID INTEGER,
endDate DATE,
jobNo INTEGER
)
```

```
INSERT INTO @jobs(deptId,pid,asmID,endDate,jobNo)
SELECT DISTINCT
s.deptId, s.pid , a.asmID,a.endDate,a.jobNo
FROM
supervises s, assignes a
WHERE
s.pid = a.pid
AND
```

```

s.deptId = @deptId
AND
a.endDate = @endDate

DECLARE @cutTest INTEGER
set @cutTest =
(SELECT count(1) where exists( SELECT
deptId,pid,asmID,endDate,jobId,details,materials,labor,m
achineType,timeTaken
FROM
@jobs j, cutJob
c
WHERE
j.jobNo =
c.jobId
)
)

IF @cutTest > 0
PRINT 'cut'

IF @cutTest > 0
SELECT

deptId,pid,asmID,endDate,jobId,details,materials,labor,m
achineType,timeTaken
FROM
@jobs j, cutJob c
WHERE
j.jobNo = c.jobId
ELSE
SELECT
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL

```

```
END

***** Object:  StoredProcedure
[dbo].[getJobHistory_fit]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[getJobHistory_fit]
@deptId INTEGER,
@endDate DATE

AS
BEGIN

    DECLARE @jobs TABLE(
        deptId INTEGER,
        pid INTEGER,
        asmID INTEGER,
        endDate DATE,
        jobNo INTEGER
    )

    INSERT INTO @jobs(deptId,pid,asmID,endDate,jobNo)
    SELECT DISTINCT
        s.deptId, s.pid , a.asmID,a.endDate,a.jobNo
    FROM
        supervises s, assignes a
    WHERE
        s.pid = a.pid
    AND
        s.deptId = @deptId
    AND
        a.endDate = @endDate
```

```
DECLARE @fitTest INTEGER
set @fitTest =
(SELECT count(1) where exists( SELECT
deptId,pid,asmID,endDate,jobId,details,labor
FROM
@jobs j, fitJob
f
WHERE
j.jobNo =
f.jobId
)
)

IF @fitTest > 0
PRINT'FIT'

IF @fitTest > 0
SELECT
deptId,pid,asmID,endDate,jobId,details,labor
FROM
@jobs j, fitJob f
WHERE
j.jobNo = f.jobId
ELSE
SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL

END
```

```
***** Object:  StoredProcedure
[dbo].[getJobHistory_paint]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[getJobHistory_paint]
@deptId INTEGER,
@endDate DATE

AS
BEGIN

    DECLARE @jobs TABLE(
        deptId INTEGER,
        pid INTEGER,
        asmID INTEGER,
        endDate DATE,
        jobNo INTEGER
    )

    INSERT INTO @jobs(deptId,pid,asmID,endDate,jobNo)
    SELECT DISTINCT
        s.deptId, s.pid , a.asmID,a.endDate,a.jobNo
    FROM
        supervises s, assignes a
    WHERE
        s.pid = a.pid
    AND
        s.deptId = @deptId
    AND
        a.endDate = @endDate
```

```
DECLARE @paintTest INTEGER
set @paintTest =
(SELECT count(1) where exists( SELECT
deptId,pid,asmID,endDate,jobId,details,color,labor,volum
e
FROM
@jobs j,
paintJob p
WHERE
j.jobNo =
p.jobId
)
)

IF @paintTest > 0
PRINT 'PAINT'

IF @paintTest > 0
SELECT

deptId,pid,asmID,endDate,jobId,details,color,labor,volum
e
FROM
@jobs j, paintJob p
WHERE
j.jobNo = p.jobId
ELSE
SELECT
NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL
END
```

```
***** Object:  StoredProcedure
[dbo].[New_assembly_customer]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[New_assembly_customer]
@name VARCHAR(64),
@address VARCHAR(64),
@category INTEGER,
@asmID INTEGER,
@orderDate DATE,
@details VARCHAR(64)

AS
BEGIN
    IF NOT EXISTS (SELECT * FROM asmbly WHERE asmID =
@asmID)
        INSERT INTO asmbly
VALUES(@asmID,@orderDate,@details);
    IF NOT EXISTS (SELECT * FROM customer WHERE name =
@name)
        INSERT INTO customer
VALUES(@name,@address,@category)
END
--
GO
```

```
***** Object: StoredProcedure [dbo].[New_Customer]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[New_Customer]
@name VARCHAR(64),
@address VARCHAR(64),
@category VARCHAR(64)

AS
BEGIN
    INSERT INTO customer VALUES (
    @name,@address,@category)
END
```

```
***** Object: StoredProcedure [dbo].[New_dept]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[New_dept]
@deptId INTEGER,
@deptData VARCHAR(64)

AS
BEGIN
    INSERT INTO dept VALUES ( @deptId,@deptData)
END
```

```
***** Object: StoredProcedure [dbo].[New_Job]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[New_Job]
@jobId INTEGER,
@pid INTEGER,
@asmID INTEGER,
@startDate DATE
AS
BEGIN
    IF EXISTS(SELECT * FROM pass_through WHERE pid
    =@pid AND asmID = @asmID)
        INSERT INTO job VALUES(@jobId)

    IF EXISTS(SELECT * FROM pass_through WHERE pid =@pid
    AND asmID = @asmID)
        INSERT INTO assignes
VALUES(@startDate,NULL,@jobId,@pid,@asmID)
    ELSE PRINT 'pid isnt associated with asmID'
END
```

```
***** Object: StoredProcedure [dbo].[New_ProcDept]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[New_ProcDept]
@pid INTEGER,
```

```
@deptData VARCHAR(64),
@deptId INTEGER
AS
BEGIN
    INSERT INTO dept VALUES ( @deptId,@deptData);
    INSERT INTO process VALUES (@pid)
END

/***** Object: StoredProcedure [dbo].[paint_Job]
Script Date: 11/21/2021 5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[paint_Job]
@jobNo INTEGER,
@details VARCHAR(64),
@color VARCHAR(64),
@labor INTEGER,
@volume INTEGER

AS
BEGIN
    INSERT INTO paintJob
VALUES(@jobNo,@details,@color,@labor,@volume)
END

GO
```

```
***** Object:  StoredProcedure
[dbo].[paintProcessTable]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[paintProcessTable]
@pid INTEGER,
@paintType VARCHAR(64),
@paintingMethod VARCHAR(64),
@process_data VARCHAR(64)
AS
BEGIN
    INSERT INTO paintProcess VALUES (
@pid,@process_data,@paintType,@paintingMethod)
END

***** Object:  StoredProcedure
[dbo].[recordTransaction]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[recordTransaction]
@transNo INTEGER,
@supCost INTEGER,
@actNo INTEGER

AS
BEGIN
    INSERT INTO jobTransaction VALUES
(@transNo,@supCost)
    INSERT INTO updates VALUES (@transNo,@actNo)
```

```
    IF EXISTS( SELECT * FROM assemblyAct WHERE actNo =  
@actNo)  
        UPDATE assemblyAct SET assemblyCost =  
assemblyCost + @supCost WHERE actNo = @actNo  
  
    IF EXISTS( SELECT * FROM procAct WHERE actNo =  
@actNo)  
        UPDATE procAct SET procCost = procCost +  
@supCost WHERE actNo = @actNo  
  
    IF EXISTS( SELECT * FROM deptAct WHERE actNo =  
@actNo)  
        UPDATE deptAct SET deptCost = deptCost +  
@supCost WHERE actNo = @actNo  
END
```

```
***** Object:  StoredProcedure  
[dbo].[supervisingTable]      Script Date: 11/21/2021  
5:49:08 PM *****
```

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE PROCEDURE [dbo].[supervisingTable]
```

```
@pid INTEGER,  
@deptId INTEGER
```

```
AS  
BEGIN  
    INSERT INTO supervises VALUES(@pid,@deptId)  
END
```

```
***** Object:  StoredProcedure
[dbo].[totalCostOnAssembly]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[totalCostOnAssembly]
@asmID INTEGER

AS
BEGIN
    SELECT sum(assemblyCost) FROM assemblyAct a INNER
JOIN maintains m on m.actNo = a.actNo WHERE m.asmID =
@asmID
END

GO

***** Object:  StoredProcedure
[dbo].[totalLaborOneDept]      Script Date: 11/21/2021
5:49:08 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[totalLaborOneDept]
@deptId INTEGER,
@endDate DATE

AS
BEGIN

DECLARE @paint INTEGER
SET @paint =
```

```
(SELECT SUM(labor)
FROM
supervises s, assignes a , paintJob p
WHERE
    a.endDate = @endDate
    AND
    s.deptId = @deptId
    AND
    a.pid = s.pid
    AND
    jobNo = p.jobId)
```

```
DECLARE @cut INTEGER
SET @cut =

(SELECT SUM(labor)
FROM
supervises s, assignes a , cutJob c
WHERE
    a.endDate = @endDate
    AND
    s.deptId = @deptId
    AND
    a.pid = s.pid
    AND
    jobNo = c.jobId)
```

```
DECLARE @fit INTEGER
SET @fit =
(SELECT SUM(labor)
FROM
supervises s, assignes a , fitJob f
WHERE
    a.endDate = @endDate
    AND
```

```
s.deptId = @deptId  
AND  
a.pid = s.pid  
AND  
jobNo = f.jobId)  
  
DECLARE @totalLabor INTEGER  
  
SET @totalLabor = IsNull(@fit,0) +IsNull(@cut,0)  
+IsNull(@paint,0)  
  
SELECT @totalLabor  
  
END
```

Running the procedures:

```

627 | INSERT INTO process VALUES (@pid)
628 | END
629 |
630 | ***** Object: StoredProcedure [dbo].[paint_Job] Script Date: 11/21/2021 5:49:08 PM *****/
631 | SET ANSI_NULLS ON
632 | GO

```

Messages

Time	Message
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 477
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 479
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 540
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 542
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 560
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 563
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 565
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 579
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 581
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 594
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 596
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 618
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 620
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 633
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 635
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 648
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 651
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 653
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 663
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 666
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 668
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 690
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 692
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 705
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 707
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 716
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 719
6:28:11 PM	Commands completed successfully.
6:28:11 PM	Started executing query at Line 721

JAVA CODE

1. Java sql driver

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class Shahzad_Ansari_IP_Task5b {
    // Database credentials
    final static String HOSTNAME = "";
    final static String DBNAME = " ";
    final static String USERNAME = " ";
    final static String PASSWORD = " ";

    // Database connection string
    final static String URL =
        String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30",
                      HOSTNAME, DBNAME, USERNAME, PASSWORD);
    // User input prompt/
    final static String PROMPT =
        "\nPlease select one of the options below: \n" +
        "1) Enter a new customer; \n" +
        "2) Enter a new department; \n" +
        "3) Enter a new process-id and its department together with its type and information relevant to the type; \n" +
        "4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered and associate it with one or more processes; \n" +
        "5) Create a new account and associate it with the process, assembly, or department to which it is \r\n" +
        "+ applicable; \n" +
        "6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced ; \n" +
        "7) At the completion of a job, enter the date it completed and the information relevant to the type \r\n" +
        "+ of job; \n" +
        "8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected \r\n" +
        "+ accounts by adding sup-cost to their current values of details; \n" +
        "9) Retrieve the total cost incurred on an assembly-id; \n" +
        "10) Retrieve the total labor time within a department for jobs completed in the department during a \r\n" +
        "+ given date; \n" +
        "11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process; \n" +
        "12) Retrieve the jobs (together with their type information and assembly-id) completed during a \r\n" +
        "+ given date in a given department; \n" +
        "13) Retrieve the customers (in name order) whose category is in a given range; \n" +
        "14) Delete all cut-jobs whose job-no is in a given range; \n" +
        "15) Change the color of a given paint job; \n" +
        "16) Import: enter new customers from a data file until the file is empty; \n" +
        "17) Export customers in a range; \n" +
        "18) Quit";
}

```

```

public static void main(String[] args) throws SQLException {
    System.out.println("Welcome to the sample application!");
    final Scanner mainSC = new Scanner(System.in);
    String option = "";
    while (!option.equals("18")) {
        System.out.println(PROMPT);
        option = mainSC.next();
        switch (option) {
            //=====
            case "1": // Insert a new customer
                insertCustomer();
                break;

            case "2": // Insert a new customer
                insertDept();
                break;

            case "3":
                Q3();
                break;
            case "4":
                Q4();
                break;
            case "5":
                Q5();
                break;
            case "6":
                Q6();
                break;
            case "7":
                Q7();
                break;
            case "8":
                Q8();
                break;
            case "9":
                Q9();
                break;
            case "10":
                Q10();
                break;
            case "11":
                Q11();
                break;
            case "12":
                Q12();
                break;
            case "13":
                Q13();
                break;
            case "14":
                Q14();
                break;
            case "15":
                Q15();
                break;
            case "16":
                Q16();
                break;
            case "17":
                Q17();
                break;
            case "18":
                System.out.println("Exiting! Goodbye!");
                break;
            default:
                System.out.println(String.format("invalid input: %s\n" + "try again!",option));
                break;
        }
    }
}

```

```

public static void insertCustomer() throws SQLException {
    Scanner sc = new Scanner(System.in);
    boolean addmore1 = true;

    while(addmore1) {
        //sc.nextLine();
        System.out.println("Please enter a customer name:");
        final String CUSTOMER_NAME = sc.nextLine();

        // sc.nextLine();
        System.out.println("Please enter the address:");
        final String ADDRESS = sc.nextLine();

        //sc.nextLine();
        System.out.println("Please enter category from 1 to 10:");
        final Integer CATEGORY = sc.nextInt();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC New_Customer @name = ?,@address = ? , @category = ?;");
                statement.setString(1, CUSTOMER_NAME);
                statement.setString(2, ADDRESS);
                statement.setInt(3, CATEGORY);

                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection1 = sc.nextInt();
        if(selection1 == 0) {
            addmore1 = false;
            break;
        }
    }
}

public static void insertDept() throws SQLException{
    boolean addmore2 = true;
    @SuppressWarnings("resource")
    Scanner sc = new Scanner(System.in);
    while(addmore2) {

        //sc.nextLine();
        System.out.println("Please enter a ID:");
        final Integer DeptID = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the department data:");
        final String DeptData = sc.next();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC New_dept @deptId =? , @deptData = ?;");
                statement.setInt(1, DeptID);
                statement.setString(2, DeptData);
                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection2 = sc.nextInt();
        if(selection2 == 0) {
            addmore2 = false;
            break;
        }
    }
}

```

```

switch(Q3ProcessSelect) {
    case(1):
        sc.nextLine();
        System.out.println("Please enter the Paint type:");
        final String Q3PAINTTYPE = sc.next();

        sc.nextLine();
        System.out.println("Please enter the Paint method:");
        final String Q3PAINTMETHOD = sc.next();

        sc.nextLine();
        System.out.println("Please enter the process Data:");
        final String Q3PROCDATA1 = sc.next();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC paintProcessTable @pid = ?, @paintType = ?, @paintingMethod = ?, @process_data = ?");
                statement.setInt(1, Q3PID);
                statement.setString(2, Q3PAINTTYPE);
                statement.setString(3, Q3PAINTMETHOD);
                statement.setString(4, Q3PROCDATA1);

                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }
        break;
    case(2):
        sc.nextLine();
        System.out.println("Please enter the Machine Type:");
        final String Q3MACHINETYPE = sc.next();

        sc.nextLine();
        System.out.println("Please enter the Cut types:");
        final String Q3CUTTYPE = sc.next();

        sc.nextLine();
        System.out.println("Please enter the process Data:");
        final String Q3PROCDATA2 = sc.next();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC cutProcessTable @pid = ?, @process_data = ?, @cutType = ?, @machineType = ?");
                statement.setInt(1, Q3PID);
                statement.setString(2, Q3PROCDATA2);
                statement.setString(3, Q3CUTTYPE);
                statement.setString(4, Q3MACHINETYPE);

                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }
        break;
}

public static void Q3() throws SQLException{
    boolean addmore3 = true;
    Scanner sc = new Scanner(System.in);

    while(addmore3) {

        sc.nextLine();
        System.out.println("Please enter a Process ID:");
        final Integer Q3PID = sc.nextInt();

        sc.nextLine();
        System.out.println("Please enter the department ID:");
        final Integer Q3DID = sc.nextInt();

        sc.nextLine();
        System.out.println("Please enter the department Data:");
        final String Q3DEPTDATA = sc.next();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC New_ProcDept @pid = ?, @deptId = ?, @deptData = ?");
                statement.setInt(1, Q3PID);
                statement.setInt(2, Q3DID);
                statement.setString(3, Q3DEPTDATA);
                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("supervisingTable @pid = ?, @deptId = ?");
                statement.setInt(1, Q3PID);
                statement.setInt(2, Q3DID);
                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        sc.nextLine();
        System.out.println(" CHOOSE : \n"+ "1. Paint process \n 2. Cut Process \n 3. Fit Process");
        final Integer Q3ProcessSelect = sc.nextInt();
    }
}

```

```

15     switch(Q3ProcessSelect) {
16
17     case(1):
18
19         sc.nextLine();
20         System.out.println("Please enter the Paint type:");
21         final String Q3PAINTTYPE = sc.next();
22
23         sc.nextLine();
24         System.out.println("Please enter the Paint method:");
25         final String Q3PAINTMETHOD = sc.next();
26
27         sc.nextLine();
28         System.out.println("Please enter the process Data:");
29         final String Q3PROCDATA1 = sc.next();
30
31
32     try (final Connection connection = DriverManager.getConnection(URL)) {
33         try (
34             // Run the stored procedure
35             final PreparedStatement statement = connection.prepareStatement("EXEC paintProcessTable @pid = ? "
36             + ", @paintType = ?, @paintingMethod = ?, @process_data = ?;")) {
37             statement.setInt(1, Q3PID);
38             statement.setString(2, Q3PAINTTYPE);
39             statement.setString(3, Q3PAINTMETHOD);
40             statement.setString(4, Q3PROCDATA1);
41
42             int rows_inserted = statement.executeUpdate();
43             System.out.println("Updating the table with user input...");
44             System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
45         }
46     }
47     break;
48
49     case(2):
50
51         sc.nextLine();
52         System.out.println("Please enter the Machine Type:");
53         final String Q3MACHINETYPE = sc.next();
54
55         sc.nextLine();
56         System.out.println("Please enter the Cut type:");
57         final String Q3CUTYTYPE = sc.next();
58
59         sc.nextLine();
60         System.out.println("Please enter the process Data:");
61         final String Q3PROCDATA2 = sc.next();
62
63
64     try (final Connection connection = DriverManager.getConnection(URL)) {
65         try (
66             // Run the stored procedure
67             final PreparedStatement statement = connection.prepareStatement("EXEC cutProcessTable @pid = ? "
68             + ", @process_data = ?, @cutType = ?, @machineType = ?;")) {
69             statement.setInt(1, Q3PID);
70             statement.setString(2, Q3PROCDATA2);
71             statement.setString(3, Q3CUTYTYPE);
72             statement.setString(4, Q3MACHINETYPE);
73
74             int rows_inserted = statement.executeUpdate();
75             System.out.println("Updating the table with user input...");
76             System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
77         }
78     }
79     break;
80
81

```

```
case(3):
    sc.nextLine();
    System.out.println("Please enter the fit type"
        + ":");
    final String Q3FITTYPE = sc.next();

    sc.nextLine();
    System.out.println("Please enter the process Data:");
    final String Q3PROCDATA3 = sc.next();

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            // Run the stored procedure
            final PreparedStatement statement = connection.prepareStatement("EXEC fitProcessTable @pid = "
                + "?, @process_data = ?,@fitType = ?;")) {
            statement.setInt(1, Q3PID);
            statement.setString(2, Q3PROCDATA3);
            statement.setString(3, Q3FITTYPE);

            int rows_inserted = statement.executeUpdate();
            System.out.println("Updating the table with user input...");
            System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
        }
    }
    break;
}

System.out.println("Do you want to add more? type 0 or 1");
Integer selection3 = sc.nextInt();
if(selection3 == 0) {
    addmore3 = false;
    break;
}
}
```

```

372* public static void Q4() throws SQLException{
373    Scanner sc = new Scanner(System.in);
374    boolean addmore4 = true;
375
376    while(addmore4) {
377
378        sc.nextLine();
379        System.out.println("Please enter customer name:");
380        final String Q4NAME = sc.next();
381
382        sc.nextLine();
383        System.out.println("Please enter customer address:");
384        final String Q4ADDRESS = sc.next();
385
386        sc.nextLine();
387        System.out.println("Please enter a category from 1 to 10:");
388        final Integer Q4CATEGORY = sc.nextInt();
389
390        sc.nextLine();
391        System.out.println("Please enter an assembly ID:");
392        final Integer Q4ASMID = sc.nextInt();
393
394        sc.nextLine();
395        System.out.println("Please enter the date the order was made:");
396        final Date Q4ORDERDATE = Date.valueOf(sc.next());
397
398        sc.nextLine();
399        System.out.println("Please enter in the assmebly details:");
400        final String Q4ASMDTAILS = sc.next();
401
402        try (final Connection connection = DriverManager.getConnection(URL)) {
403            try {
404                // Run the stored procedure
405                final PreparedStatement statement = connection.prepareStatement("EXEC New_assembly_customer "
406                    + "@name = ? , @address = ?, @category = ?, @asmID =? , @orderDate = ?, @details = ?") {
407
408                    statement.setString(1,Q4NAME);
409                    statement.setString(2,Q4ADDRESS);
410                    statement.setInt(3,Q4CATEGORY);
411                    statement.setInt(4,Q4ASMID);
412                    statement.setDate(5,Q4ORDERDATE);
413                    statement.setString(6,Q4ASMDTAILS);
414
415                    int rows_inserted = statement.executeUpdate();
416                    System.out.println("Updating the table with user input...");
417                    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
418                }
419            }
420
421            try (final Connection connection = DriverManager.getConnection(URL)) {
422                try {
423                    // Run the stored procedure
424                    final PreparedStatement statement = connection.prepareStatement("EXEC Associate_assembly_customer "
425                        + "@asmID =?, @name = ?") {
426                        statement.setInt(1, Q4ASMID);
427                        statement.setString(2, Q4NAME);
428                        int rows_inserted = statement.executeUpdate();
429                        System.out.println("Updating the table with user input...");
430                        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
431                    }
432                }
433
434
435            boolean addmore = true;
436

```

```
438         while(addmore){
439             sc.nextLine();
440             System.out.println("Enter in the PID to associate to:");
441             final Integer Q4PID = sc.nextInt();
442
443
444
445             try (final Connection connection = DriverManager.getConnection(URL)) {
446                 try (
447                     // Run the stored procedure
448                     final PreparedStatement statement = connection.prepareStatement("EXEC Associate_assembly_process "
449                         + "@asmID =? , @pid = ?")) {
450                     statement.setInt(1, Q4ASMID);
451                     statement.setInt(2, Q4PID);
452                     int rows_inserted = statement.executeUpdate();
453                     System.out.println("Updating the table with user input...");
454                     System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
455                 }
456             }
457
458             sc.nextLine();
459             System.out.println("Do you want to add another association 1 or 0 for yes or no :");
460             Integer selection = sc.nextInt();
461
462             if(selection == 0) {
463                 addmore = false;
464             }
465
466
467         }
468
469
470         System.out.println("Do you want to add more? type 0 or 1");
471         Integer selection4 = sc.nextInt();
472         if(selection4 == 0) {
473             addmore4 = false;
474             break;
475         }
476     }
477 }
```

```

public static void Q5() throws SQLException{
    boolean addmore5 = true;
    Scanner sc = new Scanner(System.in);

    while(addmore5) {

        //sc.nextLine();
        System.out.println("Please enter a account number:");
        final String Q5ACCOUNTNUMBER = sc.next();

        //sc.nextLine();
        System.out.println("Please enter the start date:");
        final Date Q5STARTDATE = Date.valueOf(sc.next());

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC createAccount @actNo =?;");
                statement.setString(1, Q5ACCOUNTNUMBER);

                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        System.out.println(" CHOOSE : \n" + "1. department acct \n 2. process acct \n 3. assembly acct");

        //sc.nextLine();
        final Integer Q5ProcessSelect = sc.nextInt();

        switch(Q5ProcessSelect) {

            case(1):

                //sc.nextLine();
                System.out.println("Please enter a dept id:");
                final String Q5DEPTID = sc.next();

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try {
                        // Run the stored procedure
                        final PreparedStatement statement = connection.prepareStatement("EXEC createDeptAccount "
                                + "@actNo = ? , @deptId = ? , @startDate = ?;");

                        statement.setString(1, Q5ACCOUNTNUMBER);
                        statement.setString(2, Q5DEPTID);
                        statement.setDate(3, Q5STARTDATE);
                    }
                }
            }
        }
    }
}

```

```

1             int rows_inserted = statement.executeUpdate();
2             System.out.println("Updating the table with user input...");
3             System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
4         }
5     }
6     break;
7
8     case(2):
9
10    //sc.nextLine();
11    System.out.println("Please enter a process id:");
12    final String Q5PID = sc.next();
13
14    try (final Connection connection = DriverManager.getConnection(URL)) {
15        try (
16            // Run the stored procedure
17            final PreparedStatement statement = connection.prepareStatement("EXEC createProcessAccount "
18                + "@actNo = ? , @pid = ? , @startDate = ?;")) {
19
20            statement.setString(1, Q5ACOUNTNUMBER);
21            statement.setString(2, Q5PID);
22            statement.setDate(3, Q5STARTDATE);
23
24            int rows_inserted = statement.executeUpdate();
25            System.out.println("Updating the table with user input...");
26            System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
27        }
28    }
29    break;
30
31     case(3):
32
33
34    //sc.nextLine();
35    System.out.println("Please enter a assembly id:");
36    final String Q5ASSEMBLYID = sc.next();
37
38    try (final Connection connection = DriverManager.getConnection(URL)) {
39        try (
40            // Run the stored procedure
41            final PreparedStatement statement = connection.prepareStatement("EXEC createAssemblyAccount "
42                + "@actNo = ? , @asmID = ? , @startDate = ?;")) {
43            statement.setString(1, Q5ACOUNTNUMBER);
44            statement.setString(2, Q5ASSEMBLYID);
45            statement.setDate(3, Q5STARTDATE);
46
47            int rows_inserted = statement.executeUpdate();
48            System.out.println("Updating the table with user input...");
49            System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
50        }
51    }
52    break;
53
54
55    System.out.println("Do you want to add more? type 0 or 1");
56    Integer selection5 = sc.nextInt();
57    if(selection5 == 0) {
58        addmore5 = false;
59        break;
60    }
61
62 }
63
64 }
```

```

• public static void Q6() throws SQLException{
    Scanner sc = new Scanner(System.in);
    boolean addmore6 = true;

    while(addmore6) {

        //sc.nextLine();
        System.out.println("Please enter the job id:");
        final Integer Q6JOBID = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the process id:");
        final Integer Q6PID = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the assembly id:");
        final Integer Q6ASSEMBLYID = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the start date:");
        final Date Q6STARTDATE = Date.valueOf(sc.next());

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC New_Job @jobId = ?,"
                        + "@pid = ? , @smID = ? , @startDate = ?;") {
                    statement.setInt(1, Q6JOBID);
                    statement.setInt(2, Q6PID);
                    statement.setInt(3, Q6ASSEMBLYID);
                    statement.setDate(4, Q6STARTDATE);

                    int rows_inserted = statement.executeUpdate();
                    System.out.println("Updating the table with user input...");
                    if(rows_inserted == -1) {
                        System.out.println("There is no association between PID and assmbley ID in the passthrough "
                                + "[table, "
                                + "cannont enter in row");
                    }else {
                        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                    }
                }
            }
        }

        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection6 = sc.nextInt();
        if(selection6 == 0) {
            addmore6 = false;
            break;
        }
    }
}

```

```
public static void Q7() throws SQLException{
    boolean addmore7 = true;
    Scanner sc = new Scanner(System.in);

    while(addmore7) {
        //sc.nextLine();
        System.out.println("Please enter the job id:");
        final Integer Q7JOBID = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the end date:");
        final Date Q7ENDDATE = Date.valueOf(sc.next());

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC completed_job "
                        + "@jobNo = ?, @endDate = ?") {
                    statement.setInt(1, Q7JOBID);
                    statement.setDate(2, Q7ENDDATE);

                    int rows_inserted = statement.executeUpdate();
                    System.out.println("Updating the table with user input...");
                    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
            }
        }

        System.out.println(" CHOOSE : \n" + "1. Paint job \n 2. Cut job \n 3. Fit job");

        //sc.nextLine();
        final Integer Q7ProcessSelect = sc.nextInt();

        switch(Q7ProcessSelect) {
            case(1):
                //sc.nextLine();
                System.out.println("Please enter the Paint color:");
                final String Q7PAINTCOLOR = sc.next();

                //sc.nextLine();
                System.out.println("Please enter the Paint volume:");
                final Integer Q7PAINTVOLUME = sc.nextInt();

                //sc.nextLine();
                System.out.println("Please enter the Paint labor :");
                final Integer Q7PAINTLABOR = sc.nextInt();

                //sc.nextLine();
                System.out.println("Please enter the department Data:");
                final String Q7PAINTDETAILS = sc.next();

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try {
                        // Run the stored procedure
                        final PreparedStatement statement = connection.prepareStatement("EXEC paint_Job "
                                + "@jobNo = ?, @details = ?, @color = ?, @labor =? , @volume =? ;") {
                            statement.setInt(1, Q7JOBID);
                            statement.setString(2, Q7PAINTDETAILS);
                            statement.setString(3, Q7PAINTCOLOR);
                            statement.setInt(4, Q7PAINTLABOR);
                            statement.setInt(5, Q7PAINTVOLUME);
                        }
                    }
                }
            }
        }
    }
}
```

```

        int rows_inserted = statement.executeUpdate();
        System.out.println("Updating the table with user input...");
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
    }
}
break;

case(2):
    //sc.nextLine();
    System.out.println("Please enter the machine type:");
    final String Q7CUTMACHINETYPE = sc.next();

    //sc.nextLine();
    System.out.println("Please enter the materials used:");
    final String Q7CUTMATERIALS = sc.next();

    //sc.nextLine();
    System.out.println("Please enter the Cut labor :");
    final Integer Q7CUTLABOR = sc.nextInt();

    //sc.nextLine();
    System.out.println("Please enter the Cut time:");
    final Integer Q7CUTTIME = sc.nextInt();

    //sc.nextLine();
    System.out.println("Please enter the Cut Data:");
    final String Q7CUTDETAILS = sc.next();

try (final Connection connection = DriverManager.getConnection(URL)) {
    try (
        // Run the stored procedure
        final PreparedStatement statement = connection.prepareStatement("EXEC cut_Job "
            + "@jobNo = ? , @details = ? , @materials = ?,@machineType = ?,@labor = ? , "
            + "@timeTaken = ? ;"))
    {
        statement.setInt(1, Q7JOBID);
        statement.setString(2, Q7CUTDETAILS);
        statement.setString(3, Q7CUTMATERIALS);
        statement.setString(4, Q7CUTMACHINETYPE);
        statement.setInt(5, Q7CUTLABOR);
        statement.setInt(6, Q7CUTTIME);

        int rows_inserted = statement.executeUpdate();
        System.out.println("Updating the table with user input...");
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
    }
}
break;

case(3):
    //sc.nextLine();
    System.out.println("Please enter the Fit labor:");
    final Integer Q7FITLABOR = sc.nextInt();

    //sc.nextLine();
    System.out.println("Please enter the Fit details:");
    final String Q7FITDETAILS = sc.next();
}

```

```

try (final Connection connection = DriverManager.getConnection(URL)) {
    try (
        // Run the stored procedure
        final PreparedStatement statement = connection.prepareStatement("EXEC fit_Job @jobNo = ? "
            + ", @details = ? , @labor = ?;")) {
        statement.setInt(1, Q7JOBID);
        statement.setString(2, Q7FITDETAILS);
        statement.setInt(3, Q7FITALBOR);

        int rows_inserted = statement.executeUpdate();
        System.out.println("Updating the table with user input...");
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
    }
    break;
}

System.out.println("Do you want to add more? type 0 or 1");
Integer selection7 = sc.nextInt();
if(selection7 == 0) {
    addmore7 = false;
    break;
}

public static void Q8() throws SQLException{
    boolean addmore8 = true;
    Scanner sc = new Scanner(System.in);
    while(addmore8) {
        System.out.println("-----");
        //sc.nextLine();
        System.out.println("Please enter a transaction number:");
        final Integer Q8TRANSPNUM = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the sup cost:");
        final Integer Q8SUPCOST = sc.nextInt();

        //sc.nextLine();
        System.out.println("Please enter the account number:");
        final Integer Q8ACTNUM = sc.nextInt();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC recordTransaction "
                    + "|@transNo = ? , @supCost = ? , @actNo = ?;")) {
                statement.setInt(1, Q8TRANSPNUM);
                statement.setInt(2, Q8SUPCOST);
                statement.setInt(3, Q8ACTNUM);

                int rows_inserted = statement.executeUpdate();
                System.out.println("Updating the table with user input...");
                System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
            }
        }

        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection8 = sc.nextInt();
        if(selection8 == 0) {
            addmore8 = false;
            break;
        }
    }
}

```

```

    public static void Q9() throws SQLException{
        boolean addmore9 = true;
        Scanner sc = new Scanner(System.in);
        while(addmore9) {
            System.out.println("-----");
            //sc.nextLine();
            System.out.println("Please enter a assembly ID:");
            final Integer Q9ASSEMBLYID = sc.nextInt();

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try {
                    // Run the stored procedure
                    final PreparedStatement statement = connection.prepareStatement("EXEC totalCostOnAssembly "
                            + "|asmID = ?;");
                    statement.setInt(1, Q9ASSEMBLYID);
                    ResultSet result = statement.executeQuery();
                    while (result.next()) {
                        System.out.println("Cost");
                        System.out.println(String.format("%s ", result.getString(1)));
                    }
                }
            }

            System.out.println("Do you want to add more? type 0 or 1");
            Integer selection9 = sc.nextInt();
            if(selection9 == 0) {
                addmore9 = false;
                break;
            }
        }
    }

    public static void Q10() throws SQLException{
        Scanner sc = new Scanner(System.in);

        boolean addmore10 = true;

        while(addmore10) {
            System.out.println("-----");
            System.out.println("Please enter a department ID:");
            final Integer Q10DEPTID = sc.nextInt();

            System.out.println("Please enter the end date:");
            final Date Q10ENDDATE = Date.valueOf(sc.next());

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try {
                    // Run the stored procedure
                    final PreparedStatement statement = connection.prepareStatement("EXEC totalLaborOneDept "
                            + "|deptId = ?, @endDate = ?)");
                    statement.setInt(1, Q10DEPTID);
                    statement.setDate(2, Q10ENDDATE);
                    ResultSet result = statement.executeQuery();
                    while (result.next()) {
                        System.out.println("labor");
                        System.out.println(String.format("%s ", result.getString(1)));
                    }
                }
            }

            System.out.println("Do you want to add more? type 0 or 1");
            Integer selection10 = sc.nextInt();
            if(selection10 == 0) {
                addmore10 = false;
                break;
            }
        }
    }
}

```

```
• public static void Q11() throws SQLException{
  boolean addmore11 = true;
  Scanner sc = new Scanner(System.in);

  while(addmore11) {
    System.out.println("-----");
    System.out.println("Please enter a assembly ID:");
    final Integer Q10ASSEMBLYID = sc.nextInt();

    try (final Connection connection = DriverManager.getConnection(URL)) {
      try {
        // Run the stored procedure
        final PreparedStatement statement = connection.prepareStatement("EXEC getAssemblyProcesHistory"
          + "|asmID = ?;");
        statement.setInt(1, Q10ASSEMBLYID);
        ResultSet result = statement.executeQuery();
        System.out.println("asmID | start date | pid | did");
        while (result.next()) {
          System.out.println(String.format("%s | %s | %s | %s",
            result.getString(1),
            result.getString(2),
            result.getString(3),
            result.getString(4)
          ));
        }
      }
    }
    System.out.println("Do you want to add more? type 0 or 1");
    Integer selection11 = sc.nextInt();
    if(selection11 == 0) {
      addmore11 = false;
      break;
    }
  }
}
```

```
public static void Q12() throws SQLException{
    Scanner sc = new Scanner(System.in);
    boolean addmore12 = true;
    while(addmore12) {
        System.out.println("-----");

        System.out.println("Please enter a department ID:");
        final Integer Q12DEPTID = sc.nextInt();

        System.out.println("Please enter the end date:");
        final Date Q12ENDDATE = Date.valueOf(sc.next());

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC getJobHistory_paint "
                        + "@deptId = ?, @endDate = ?;");
                statement.setInt(1, Q12DEPTID);
                statement.setDate(2, Q12ENDDATE);

                ResultSet result = statement.executeQuery();
                System.out.println("=====");

                System.out.println("PAINT JOBTS\n"
                        + "ID|PID|asmID|end date|job ID|Details|color|labor|volume");
                while (result.next()) {

                    if(result.getString(1)=="NULL") {
                        System.out.println("NO ROWS");
                        break;
                    }else {
                        System.out.println(String.format("%s | %s ", 
                                result.getString(1),
                                result.getString(2),
                                result.getString(3),
                                result.getString(4),
                                result.getString(5),
                                result.getString(6),
                                result.getString(7),
                                result.getString(8),
                                result.getString(9)
                            ));
                    }
                    break;
                }
            }
        }
    }
}
```

```

try (final Connection connection = DriverManager.getConnection(URL)) {
    try {
        // Run the stored procedure
        final PreparedStatement statement = connection.prepareStatement("EXEC getJobHistory_cut "
                + "@deptId = ?, @endDate = ?;") {
            statement.setInt(1, Q12DEPTID);
            statement.setDate(2, Q12ENDDATE);

            ResultSet result = statement.executeQuery();
            System.out.println("=====");
            System.out.println("CUT JOBS\n"
                    + "DID|PID|asmID|end date|job ID|Materials|labor|machine Type|time taken");
            while (result.next()) {

                if(result.getString(1)=="null") {
                    System.out.println("NO ROWS");
                    break;
                }else {
                    System.out.println(String.format("%s | %s ",
                        result.getString(1),
                        result.getString(2),
                        result.getString(3),
                        result.getString(4),
                        result.getString(5),
                        result.getString(6),
                        result.getString(7),
                        result.getString(8),
                        result.getString(9)
                    ));
                }
            }
            break;
        }
    }
}

try (final Connection connection = DriverManager.getConnection(URL)) {
    try {
        // Run the stored procedure
        final PreparedStatement statement = connection.prepareStatement("EXEC getJobHistory_fit "
                + "@deptId = ?, @endDate = ?;") {
            statement.setInt(1, Q12DEPTID);
            statement.setDate(2, Q12ENDDATE);

            ResultSet result = statement.executeQuery();
            System.out.println("=====");

            System.out.println("FIT JOBTS\n"
                    + "DID|PID|asmID|end date|job ID|Detials|labor");
            while (result.next()) {
                if(result.getString(1) == "null") {
                    System.out.println("NO ROWS");
                }else {
                    System.out.println(String.format("%s | %s | %s | %s | %s | %s | %s",
                        result.getString(1),
                        result.getString(2),
                        result.getString(3),
                        result.getString(4),
                        result.getString(5),
                        result.getString(6),
                        result.getString(7)
                    ));
                }
            }
        }
    }
}

```

```

        });
    }
}

System.out.println("Do you want to add more? type 0 or 1");
Integer selection12 = sc.nextInt();
if(selection12 == 0) {
    addmore12 = false;
    break;
}
}

public static void Q13() throws SQLException{
    Scanner sc = new Scanner(System.in);

    boolean addmore13 = true;

    while(addmore13) {
        System.out.println("-----");

        System.out.println("Please enter a start range:");
        final Integer Q13STARTRANGE = sc.nextInt();

        System.out.println("Please enter a end range:");
        final Integer Q13ENDRANGE = sc.nextInt();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("getCustomers " +
                    + "@startRange = ? ,@endRange = ?;");
                statement.setInt(1, Q13STARTRANGE);
                statement.setInt(2, Q13ENDRANGE );
                ResultSet result = statement.executeQuery();
                System.out.println("name | address | category");
                while (result.next()) {
                    System.out.println(String.format("%s | %s | %s ",
                        result.getString(1),
                        result.getString(2),
                        result.getString(3)
                    ));
                }
            }
        }
        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection13 = sc.nextInt();
        if(selection13 == 0) {
            addmore13 = false;
            break;
        }
    }
}
}

```

```

public static void Q14() throws SQLException{
    boolean addmore14 = true;
    Scanner sc = new Scanner(System.in);

    while(addmore14) {
        System.out.println("-----");
        System.out.println("Please enter a start range:");
        final Integer Q14STARTRANGE = sc.nextInt();

        System.out.println("Please enter a end range:");
        final Integer Q14ENDRANGE = sc.nextInt();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC deleteCutJob "
                    + "@startRange = ?, @endRange = ?;");
                statement.setInt(1, Q14STARTRANGE);
                statement.setInt(2, Q14ENDRANGE );

                statement.executeUpdate();
            }
        }

        System.out.println("Do you want to delete more? type 0 or 1");
        Integer selection14 = sc.nextInt();
        if(selection14 == 0) {
            addmore14 = false;
            break;
        }
    }
}

public static void Q15() throws SQLException{
    Scanner sc = new Scanner(System.in);

    boolean addmore15 = true;

    while(addmore15) {

        System.out.println("Please enter a job ID:");
        final Integer Q15JOBID = sc.nextInt();

        System.out.println("Please enter a color:");
        final String Q15COLOR = sc.next();

        try (final Connection connection = DriverManager.getConnection(URL)) {
            try {
                // Run the stored procedure
                final PreparedStatement statement = connection.prepareStatement("EXEC changePaint "
                    + "@jobId = ?, @color = ?;");
                statement.setInt(1, Q15JOBID);
                statement.setString(2, Q15COLOR );

                statement.executeUpdate();
            }
        }

        System.out.println("Do you want to add more? type 0 or 1");
        Integer selection15 = sc.nextInt();
        if(selection15 == 0) {
            addmore15 = false;
            break;
        }
    }
}

```

```

public static void Q16() throws SQLException{
    System.out.println("Please enter the file info in this format location\\filename.csv format");
    Scanner sc = new Scanner(System.in);

    final String csvRead = sc.next();
    BufferedReader br = null;
    String line = "";
    String cvsSplitBy = ",";
    try {//go through lines in the provided file
        br = new BufferedReader(new FileReader(csvRead));
        while ((line = br.readLine()) != null) {
            //split the columns
            String[] CUSTOMER_DATA = line.split(cvsSplitBy);
            //set attribute values from the csv file
            String name = CUSTOMER_DATA[0];
            String address = CUSTOMER_DATA[1];
            String category = CUSTOMER_DATA[2];

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try {
                    // Run the stored procedure
                    final PreparedStatement statement = connection.prepareStatement("EXEC New_Customer "
                            + "[@name = ?, @address = ? , @category = ?]");
                    statement.setString(1, name);
                    statement.setString(2, address);
                    statement.setInt(3, Integer.parseInt(category));

                    int rows_inserted = statement.executeUpdate();
                    System.out.println("Updating the table with user input...");
                    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
            }
        }
    } //check for file availability
    catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

public static void Q17() throws SQLException{
    Scanner sc = new Scanner(System.in);

    System.out.println("Please enter a start range:");
    final Integer Q13STARTRANGE = sc.nextInt();

    System.out.println("Please enter a end range:");
    final Integer Q13ENDRANGE = sc.nextInt();

    System.out.println("Please enter the export location and file name in location\\filename.csv format");

    final String csvInfo = sc.next();

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try {
            // Run the stored procedure
            final PreparedStatement statement = connection.prepareStatement("getCustomers @startRange = "
                    + " ? ,@endRange = ?;") {
                statement.setInt(1, Q13STARTRANGE);
                statement.setInt(2, Q13ENDRANGE );
                ResultSet result = statement.executeQuery();

                BufferedWriter fileWriter = new BufferedWriter(new FileWriter(csvInfo));

                // write header line containing column names
                fileWriter.write("name,address,category");

                while (result.next()) {
                    String name = result.getString("name");
                    String address = result.getString("address");
                    Integer category = result.getInt("category");

                    String line = String.format("%s,%s,%s",
                        name, address,category);

                    fileWriter.newLine();
                    fileWriter.write(line);
                }

                statement.close();
                fileWriter.close();
            } catch (SQLException e) {
                System.out.println("Database error:");
                e.printStackTrace();
            } catch (IOException e) {
                System.out.println("File IO error:");
                e.printStackTrace();
            }
        }
    }
}

```

Queries

1.

```

1
Please enter a customer name:
shahzad

Please enter the address:
test dr

Please enter category from 1 to 10:
1

Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a customer name:
tim

Please enter the address:
red st

Please enter category from 1 to 10:
2

Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a customer name:
drake

Please enter the address:
green ave

Please enter category from 1 to 10:
3

Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a customer name:
ronald

Please enter the address:
toot st

Please enter category from 1 to 10:
4

Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a customer name:
reggy

Please enter the address:
apple ct

Please enter category from 1 to 10:
5

Updating the table with user input...

```

14 | `SELECT * FROM [dbo].[customer];`

name	address	category

13 | --SELECT * FROM [dbo].[assignes];
 14 | `SELECT * FROM [dbo].[customer];`
 15 | `SELECT * FROM [dbo].[customer];`

	name	address	category
1	drake	green ave	3
2	reggy	apple ct	5
3	ronald	toot st	4
4	shahzad	test dr	1
5	tim	red st	2

2.

```

2
Please enter a ID:
10
Please enter the department data:
d1
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a ID:
11
Please enter the department data:
d2
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a ID:
12
Please enter the department data:
d3
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a ID:
13
Please enter the department data:
d4
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a ID:
14
Please enter the department data:
d5
Updating the table with user input...
Done. 1 rows inserted.

```

19 `SELECT * FROM [dbo].[dept];`
 20 `--SELECT * FROM [dbo].[supervises];`
 21 `--SELECT * FROM [dbo].[paintProcess];`

Results **Messages**

	deptId	deptData

18 `--SELECT * FROM [dbo].[pass_through];`
 19 `SELECT * FROM [dbo].[dept];`
 20 `--SELECT * FROM [dbo].[supervises];`
 21 `--SELECT * FROM [dbo].[paintProcess];`
 22 `--SELECT * FROM [dbo].[cutProcess];`

Results **Messages**

	deptId	deptData
1	10	d1
2	11	d2
3	12	d3
4	13	d4

3.

	pid	
1	10	d1
2	11	d2
3	12	d3
4	13	d4
5	14	d5

	pid	deptId		
1	10	d1		
	pid	process_data	paintType	paintingMethod
2	11	d2	c1	m1
3	12	d3	c2	m2
4	13	d4	c3	m3
5	14	d5	c4	m4

	pid	process_data	cutType	machineType	
1	10	d1	c1	m1	
	pid	process_data	fitType		
2	11	d2	c2	m2	
3	12	d3	c3	m3	
4	13	d4	c4	m4	
5	14	d5	c5	m5	

```

3
Please enter a Process ID:
20
Please enter the department ID:
15
Please enter the department Data:
asdsdf
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
1
Please enter the Paint type:
smooth
Please enter the Paint method:
brush
Please enter the process Data:
asdfsd
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
21
Please enter the department ID:
16
Please enter the department Data:
asdfsdf
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
1
Please enter the Paint type:
glossy
Please enter the Paint method:
brush
Please enter the process Data:
adfsfs
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
22
Please enter the department ID:
17
Please enter the department Data:
asdfsdf
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
1
Please enter the Paint type:
shiny
Please enter the Paint method:
spray
Please enter the process Data:
asdfs
Please enter a Process ID:
23
Please enter the department ID:
18
Please enter the department Data:
ssdfd
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
2
Please enter the Machine Type:
large
Please enter the Cut type:
fast
Please enter the process Data:
sdffs
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
24
Please enter the department ID:
19
Please enter the department Data:
asdfsdf
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
2
Please enter the Machine Type:
small
Please enter the Cut type:
slow
Please enter the process Data:
asdfsd
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
25
Please enter the department ID:
20
Please enter the department Data:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
2
Please enter the Machine Type:
medium
Please enter the Cut type:
fast
Please enter the process Data:
asdfs

```

```

Please enter a Process ID:
26
Please enter the department ID:
21
Please enter the department Data:
asdfd
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
3
Please enter the fit type:
small
Please enter the process Data:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
27
Please enter the department ID:
22
Please enter the department Data:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
3
Please enter the fit type:
large
Please enter the process Data:
asdsd
Updating the table with user input...
Done. 1 rows inserted.

```

```

Do you want to add more? type 0 or 1
1
Please enter a Process ID:
28
Please enter the department ID:
23
Please enter the department Data:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
3
Please enter the fit type:
medium
Please enter the process Data:
asdfsd
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a Process ID:
29
Please enter the department ID:
24
Please enter the department Data:
sdfd
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint process
2. Cut Process
3. Fit Process
3
Please enter the fit type:
small
Please enter the process Data:
asdfs
Updating the table with user input...
Done. 1 rows inserted.

```

```

20  SELECT * FROM [dbo].[supervises];
21  SELECT * FROM [dbo].[paintProcess];
22  SELECT * FROM [dbo].[cutProcess];
23  SELECT * FROM [dbo].[fitProcess];
24

```

Results Messages

	deptId	deptData
3	12	d3
4	13	d4
5	14	d5
6	15	asdssdf
7	16	asdfsdf
8	17	asdfsdf
9	18	ssdfd
10	19	asdfsd
11	20	asdfs
12	21	asdfd

	pid	deptId
1	20	15
2	21	16
3	22	17
4	23	18
5	24	19
6	25	20
7	26	21
8	27	22
9	28	23
10	29	24

	pid	process_data	paintType	paintingMethod
1	20	asdfsd	smooth	brush
2	21	adfsfs	glossy	brush
3	22	sdfss	shiny	spray

	pid	process_data	cutType	machineType
1	23	sdfss	fast	large
2	24	asdfsd	slow	small
3	25	asdfs	fast	medium

	pid	process_data	fitType
1	26	asdfs	small
2	27	asdssd	large
3	28	asdfsdf	medium
4	29	asdfs	small

```
14    --SELECT * FROM [dbo].[customer];
15    --SELECT * FROM [dbo].[asmbly];
16    SELECT * FROM [dbo].[process];
17    --SELECT * FROM [dbo].[cust_order];
18    --SELECT * FROM [dbo].[pass_through];
19    SELECT * FROM [dbo].[dept];
20    SELECT * FROM [dbo].[supervises];
21    SELECT * FROM [dbo].[paintProcess];
22    SELECT * FROM [dbo].[cutProcess];
23    SELECT * FROM [dbo].[fitProcess];
24
```

Results Messages

	pid
1	20
2	21
3	22
4	23
5	24
6	25
7	26
8	27
9	28
10	29

	deptId	deptData
1	10	d1
2	11	d2
3	12	d3
4	13	d4
5	14	d5
6	15	asdsdf
7	16	asdfasdf
8	17	asdfasdf
9	18	ssdfsd
10	19	asdfasd

4.

```

9   --SELECT * FROM [dbo].[assemblyAct];
10  --SELECT * FROM [dbo].[procAct];
11  --SELECT * FROM [dbo].[deptAct];
12  --SELECT * FROM [dbo].[maintains];
13  --SELECT * FROM [dbo].[assignes];
14  SELECT * FROM [dbo].[customer];
15  --SELECT * FROM [dbo].[asmbly];
16  SELECT * FROM [dbo].[process];
17  SELECT * FROM [dbo].[cust_order];
18  SELECT * FROM [dbo].[pass_through];

```

Results **Messages**

	name	address	category
1	drake	green ave	3
2	reggy	apple ct	5
3	ronald	toot st	4
4	shahzad	test dr	1
5	tim	red st	2

	pid
1	20
2	21
3	22
4	23
5	24
6	25
7	26
8	27
9	28
10	29

	asmID	name

	pid	asmID

```
Please enter customer name:  
drake  
Please enter customer address:  
green ave  
Please enter a category from 1 to 10:  
3  
Please enter an assembly ID:  
5  
Please enter the date the order was made:  
2018-10-11  
Please enter in the assmebly details:  
adsfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
20  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
0  
Do you want to add more? type 0 or 1  
1  
Please enter customer name:  
shah  
Please enter customer address:  
green st  
Please enter a category from 1 to 10:  
4  
Please enter an assembly ID:  
6  
Please enter the date the order was made:  
2018-11-11  
Please enter in the assmebly details:  
asdfsdf  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
27  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
0  
Do you want to add more? type 0 or 1  
1  
Please enter customer name:  
sam  
Please enter customer address:  
toot ct  
Please enter a category from 1 to 10:  
5  
Please enter an assembly ID:  
7  
Please enter the date the order was made:  
2018-9-9  
Please enter in the assmebly details:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
23  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
0  
Do you want to add more? type 0 or 1
```

```
4
Please enter customer name:
ronald
Please enter customer address:
toot st
Please enter a category from 1 to 10:
4
Please enter an assembly ID:
3
Please enter the date the order was made:
2018-3-1
Please enter in the assmebly details:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
Enter in the PID to associate to:
25
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
0
Do you want to add more? type 0 or 1
1
Please enter customer name:
reggy
Please enter customer address:
apple ct
Please enter a category from 1 to 10:
5
Please enter an assembly ID:
4
Please enter the date the order was made:
2018-8-11
Please enter in the assmebly details:
asdfds
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
Enter in the PID to associate to:
29
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
1
Enter in the PID to associate to:
22
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
0
```

```
4
Please enter customer name:
shahzad
Please enter customer address:
test dr
Please enter a category from 1 to 10:
1
Please enter an assembly ID:
1
Please enter the date the order was made:
2018-1-1
Please enter in the assmебly details:
adfsdfs
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
Enter in the PID to associate to:
20
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
1
Enter in the PID to associate to:
21
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
0
Do you want to add more? type 0 or 1
1
Please enter customer name:
tim
Please enter customer address:
red st
Please enter a category from 1 to 10:
2
Please enter an assembly ID:
2
Please enter the date the order was made:
2018-1-10
Please enter in the assmебly details:
asdsfd
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
Enter in the PID to associate to:
20
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add another association 1 or 0 for yes or no :
1
Enter in the PID to associate to:
22
Updating the table with user input...
Done. 1 rows inserted.
```

```
Please enter customer name:  
tabby  
Please enter customer address:  
pop st  
Please enter a category from 1 to 10:  
6  
Please enter an assembly ID:  
8  
Please enter the date the order was made:  
2018-1-3  
Please enter in the assmebly details:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
28  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
0  
Do you want to add more? type 0 or 1  
1  
Please enter customer name:  
frank  
Please enter customer address:  
s ave  
Please enter a category from 1 to 10:  
8  
Please enter an assembly ID:  
9  
Please enter the date the order was made:  
2018-4-3  
Please enter in the assmebly details:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
22  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
1  
Enter in the PID to associate to:  
27  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add another association 1 or 0 for yes or no :  
0  
Do you want to add more? type 0 or 1  
1  
Please enter customer name:  
lexi  
Please enter customer address:  
cat dr  
Please enter a category from 1 to 10:  
5  
Please enter an assembly ID:  
10  
Please enter the date the order was made:  
2018-4-21  
Please enter in the assmebly details:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Updating the table with user input...  
Done. 1 rows inserted.  
Enter in the PID to associate to:  
22
```

```
2 --SELECT * FROM [dbo].[job];
3 --SELECT * FROM [dbo].[cutJob];
4 --SELECT * FROM [dbo].[paintJob];
5 --SELECT * FROM [dbo].[fitJob];
6 --SELECT * FROM [dbo].[jobTransaction];
7 SELECT * FROM [dbo].[account];
8 --SELECT * FROM [dbo].[updates];
9 SELECT * FROM [dbo].[assemblyAct];
10 SELECT * FROM [dbo].[procAct];
11 SELECT * FROM [dbo].[deptAct];
12 SELECT * FROM [dbo].[maintains];
13 --SELECT * FROM [dbo].[assignes];
```

Results Messages

actNo

	actNo	startDate	assemblyCost
--	-------	-----------	--------------

	actNo	startDate	procCost
--	-------	-----------	----------

	actNo	startDate	deptCost
--	-------	-----------	----------

pid	asmID	deptId	actNo
-----	-------	--------	-------

```
14  SELECT * FRO  
15  SELECT * FRO  
16  SELECT * FRO  
17  SELECT * FRO  
18  SELECT * FRO
```

Results Messages

	name	▼	add
1	drake		gr
2	frank		s
3	lexi		ca
4	reggy		app
5	ronald		too
6	sam		too
7	shah		gr
8	shahzad		te
9	tabby		pop
10	tim		re

	asmID	✓	or
1	1		20
2	2		20
3	3		20
4	4		20
5	5		20
6	6		20
7	7		20
8	8		20
9	9		20
10	10		20

	pid	▽
1	20	
2	21	
3	22	
4	23	
5	24	
6	25	
7	26	
8	27	
9	28	
10	29	

5.

14	SELECT * FROM [dbo].[customer];			
15	SELECT * FROM [dbo].[asmbly];			
16	SELECT * FROM [dbo].[process];			
17	SELECT * FROM [dbo].[cust_order];			
18	SELECT * FROM [dbo].[pass_through];			
Results Messages				
6	6	2018-11-11	asdfsdf	
7	7	2018-09-09	asdfsd	
8	8	2018-01-03	asdfsd	
9	9	2018-04-03	asdfs	
10	10	2018-04-21	asdfsd	
	pid	▼		
1	20			
2	21			
3	22			
4	23			
5	24			
6	25			
7	26			
8	27			
9	28			
10	29			
	asmID	▼	name	▼
1	1		shahzad	
2	2		tim	
3	3		ronald	
4	4		reggy	
5	5		drake	
6	6		shah	
7	7		sam	
8	8		tabby	
9	9		frank	
10	10		lexi	
	pid	▼	asmID	▼
1	20		1	
2	21		1	
3	20		2	
4	22		2	
5	23		7	
6	22		9	
7	27		9	
8	22		10	
9	25		3	
10	29		4	

```

5
Please enter a account number:
1
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
1
Please enter a dept id:
10
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a account number:
2
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
1
Please enter a dept id:
11
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a account number:
3
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
1
Please enter a dept id:
12
Updating the table with user input...
Done. 1 rows inserted.

```

```

Please enter a account number:
4
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
2
Please enter a process id:
20
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a account number:
5
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
2
Please enter a process id:
21
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter a account number:
6
Please enter the start date:
2019-1-1
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. department acct
2. process acct
3. assembly acct
2
Please enter a process id:
22
Updating the table with user input...
Done. 1 rows inserted.

```

```
Please enter a account number:  
7  
Please enter the start date:  
2019-1-1  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. department acct  
2. process acct  
3. assembly acct  
3  
Please enter a assembly id:  
1  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
Please enter a account number:  
8  
Please enter the start date:  
2019-1-1  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. department acct  
2. process acct  
3. assembly acct  
3  
Please enter a assembly id:  
2  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
Please enter a account number:  
9  
Please enter the start date:  
2019-1-1  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. department acct  
2. process acct  
3. assembly acct  
3  
Please enter a assembly id:  
2  
Updating the table with user input...  
Done. 1 rows inserted.
```

```
Please enter a account number:  
10  
Please enter the start date:  
2019-1-1  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. department acct  
2. process acct  
3. assembly acct  
3  
Please enter a assembly id:  
3  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
0
```

```
7  SELECT * FROM [dbo].[account];
8  --SELECT * FROM [dbo].[updates];
9  SELECT * FROM [dbo].[assemblyAct];
10 SELECT * FROM [dbo].[procAct];
11 SELECT * FROM [dbo].[deptAct];
12 SELECT * FROM [dbo].[maintains];
13 --SELECT * FROM [dbo].[assignes];
14 --SELECT * FROM [dbo].[customer];
15 SELECT * FROM [dbo].[asmbly];
```

Results **Messages**

	actNo
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

	actNo	startDate	assemblyCost
1	7	2019-01-01	0
2	8	2019-01-01	0
3	9	2019-01-01	0
4	10	2019-01-01	0

	actNo	startDate	procCost
1	4	2019-01-01	0
2	5	2019-01-01	0
3	6	2019-01-01	0

	actNo	startDate	deptCost
1	1	2019-01-01	0
2	2	2019-01-01	0
3	3	2019-01-01	0

```

7  SELECT * FROM [dbo].[account];
8  --SELECT * FROM [dbo].[updates];
9  SELECT * FROM [dbo].[assemblyAct];
10 SELECT * FROM [dbo].[procAct];
11 SELECT * FROM [dbo].[deptAct];
12 SELECT * FROM [dbo].[maintains];
13 --SELECT * FROM [dbo].[assignes];
14 --SELECT * FROM [dbo].[customer];
15 SELECT * FROM [dbo].[asmbly];

```

Results **Messages**

3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

	actNo	startDate	assemblyCost
1	7	2019-01-01	0
2	8	2019-01-01	0
3	9	2019-01-01	0
4	10	2019-01-01	0

	actNo	startDate	procCost
1	4	2019-01-01	0
2	5	2019-01-01	0
3	6	2019-01-01	0

	actNo	startDate	deptCost
1	1	2019-01-01	0
2	2	2019-01-01	0
3	3	2019-01-01	0

	pid	asmID	deptId	actNo
1	NULL	NULL	10	1
2	NULL	NULL	11	2
3	NULL	NULL	12	3
4	20	NULL	NULL	4
5	21	NULL	NULL	5
6	22	NULL	NULL	6
7	NULL	1	NULL	7
8	NULL	2	NULL	8
9	NULL	2	NULL	9
10	NULL	3	NULL	10

6.

```
25 SELECT * from job  
26  
27 SELECT * from assignes  
28
```

Results Messages

jobId

startDate	endDate	jobNo	pid	asmID
-----------	---------	-------	-----	-------

```

Do you want to add more? type 0 or 1
1
Please enter the job id:
6
Please enter the process id:
22
Please enter the assembly id:
4
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
7
Please enter the process id:
27
Please enter the assembly id:
9
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
8
Please enter the process id:
20
Please enter the assembly id:
5
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
9
Please enter the process id:
20
Please enter the assembly id:
1
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
10
Please enter the process id:
27
Please enter the assembly id:
6
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
0

```

```

18) Quit
6
Please enter the job id:
1
Please enter the process id:
20
Please enter the assembly id:
2
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
2
Please enter the process id:
21
Please enter the assembly id:
1
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
3
Please enter the process id:
22
Please enter the assembly id:
2
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
4
Please enter the process id:
25
Please enter the assembly id:
3
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
5
Please enter the process id:
27
Please enter the assembly id:
6
Please enter the start date:
2019-9-5
Updating the table with user input...
Done. 1 rows inserted.

```

```
... 23 --SELECT * FROM [dbo].[fitProcess];
24
25 SELECT * from job
26
27 SELECT * from assignes
28
```

Results **Messages**

	jobId
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

	startDate	endDate	jobNo	pid	asmID
1	2019-09-05	NULL	1	20	2
2	2019-09-05	NULL	2	21	1
3	2019-09-05	NULL	3	22	2
4	2019-09-05	NULL	4	25	3
5	2019-09-05	NULL	5	27	6
6	2019-09-05	NULL	6	22	4
7	2019-09-05	NULL	7	27	9
8	2019-09-05	NULL	8	20	5
9	2019-09-05	NULL	9	20	1
10	2019-09-05	NULL	10	27	6

7.

```

24
25   SELECT * from job
26   SELECT * from assignes
27   SELECT * FROM [dbo].[cutJob];
28   SELECT * FROM [dbo].[paintJob];
29   SELECT * FROM [dbo].[fitJob];
30
y...
y...
to...
t
unt
Results    Messages


|    | startDate  | endDate | jobNo | pid | asmID |
|----|------------|---------|-------|-----|-------|
| 1  | 2019-09-05 | NULL    | 1     | 20  | 2     |
| 2  | 2019-09-05 | NULL    | 2     | 21  | 1     |
| 3  | 2019-09-05 | NULL    | 3     | 22  | 2     |
| 4  | 2019-09-05 | NULL    | 4     | 25  | 3     |
| 5  | 2019-09-05 | NULL    | 5     | 27  | 6     |
| 6  | 2019-09-05 | NULL    | 6     | 22  | 4     |
| 7  | 2019-09-05 | NULL    | 7     | 27  | 9     |
| 8  | 2019-09-05 | NULL    | 8     | 20  | 5     |
| 9  | 2019-09-05 | NULL    | 9     | 20  | 1     |
| 10 | 2019-09-05 | NULL    | 10    | 27  | 6     |



|  | jobId | details | materials | machineType | labor | timeTaken |
|--|-------|---------|-----------|-------------|-------|-----------|
|--|-------|---------|-----------|-------------|-------|-----------|


|  | jobId | details | color | labor | volume |
|--|-------|---------|-------|-------|--------|
|--|-------|---------|-------|-------|--------|


|  | jobId | details | labor |
|--|-------|---------|-------|
|--|-------|---------|-------|


```

```
Please enter the job id:  
3  
Please enter the end date:  
2019-10-05  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. Paint job  
2. Cut job  
3. Fit job  
1  
Please enter the Paint color:  
blue  
Please enter the Paint volume:  
5  
Please enter the Paint labor :  
44  
Please enter the department Data:  
asdfsdf  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
Please enter the job id:  
4  
Please enter the end date:  
2019-10-05  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. Paint job  
2. Cut job  
3. Fit job  
2  
Please enter the machine type:  
large  
Please enter the materials used:  
44  
Please enter the Cut labor :  
2  
Please enter the Cut time:  
23  
Please enter the Cut Data:  
adsfdf  
Updating the table with user input...  
Done. 1 rows inserted.
```

```
Please enter the job id:  
5  
Please enter the end date:  
2019-10-05  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. Paint job  
2. Cut job  
3. Fit job  
2  
Please enter the machine type:  
small  
Please enter the materials used:  
33  
Please enter the Cut labor :  
2  
Please enter the Cut time:  
33  
Please enter the Cut Data:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
Please enter the job id:  
6  
Please enter the end date:  
2019-10-05  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. Paint job  
2. Cut job  
3. Fit job  
3  
Please enter the Fit labor:  
22  
Please enter the Fit details:  
asdsf  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
Please enter the job id:  
7  
Please enter the end date:  
2019-10-05  
Updating the table with user input...  
Done. 1 rows inserted.  
CHOOSE :  
1. Paint job  
2. Cut job  
3. Fit job  
3  
Please enter the Fit labor:  
22  
Please enter the Fit details:  
asdfs  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1
```

```
HW3test [Java Application] C:\Users\ShahzadAnsari\p2\pool\plugins\org.eclipse.justj.openjdk.h
Please enter the job id:
8
Please enter the end date:
2019-10-05
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint job
2. Cut job
3. Fit job
3
Please enter the Fit labor:
54
Please enter the Fit details:
asdf
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
9
Please enter the end date:
2019-10-05
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint job
2. Cut job
3. Fit job
3
Please enter the Fit labor:
87
Please enter the Fit details:
asdfsds
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
10
Please enter the end date:
2019-10-05
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint job
2. Cut job
3. Fit job
3
Please enter the Fit labor:
211
Please enter the Fit details:
asdfs
Updating the table with user input...
Done. 1 rows inserted.
```

```
7
Please enter the job id:
1
Please enter the end date:
2019-10-05
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint job
2. Cut job
3. Fit job
1
Please enter the Paint color:
red
Please enter the Paint volume:
10
Please enter the Paint labor :
20
Please enter the department Data:
asdf
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
Please enter the job id:
2
Please enter the end date:
2019-10-05
Updating the table with user input...
Done. 1 rows inserted.
CHOOSE :
1. Paint job
2. Cut job
3. Fit job
1
Please enter the Paint color:
green
Please enter the Paint volume:
2
Please enter the Paint labor :
25
Please enter the department Data:
asdfdfsf
Updating the table with user input...
Done. 1 rows inserted.
```

```

23 --SELECT * FROM [dbo].[fitProcess];
24
25 SELECT * from job
26 SELECT * from assignes
27 SELECT * FROM [dbo].[cutJob];
28 SELECT * FROM [dbo].[paintJob];
29 SELECT * FROM [dbo].[fitJob];
30
31

```

Results **Messages**

	startDate	endDate	jobNo	pid	asmID
1	2019-09-05	2019-10-05	1	20	2
2	2019-09-05	2019-10-05	2	21	1
3	2019-09-05	2019-10-05	3	22	2
4	2019-09-05	2019-10-05	4	25	3
5	2019-09-05	2019-10-05	5	27	6
6	2019-09-05	2019-10-05	6	22	4
7	2019-09-05	2019-10-05	7	27	9
8	2019-09-05	2019-10-05	8	20	5
9	2019-09-05	2019-10-05	9	20	1
10	2019-09-05	2019-10-05	10	27	6

	jobId	details	materials	machineType	labor	timeTaken
1	4	adsfd	44	large	2	23
2	5	asdfs	33	small	2	33

	jobId	details	color	labor	volume
1	1	asdf	red	20	10
2	2	asfdfsfdf	green	25	2
3	3	asdfsdsd	blue	44	5

	jobId	details	labor
1	6	asdsf	22
2	7	asdfs	22
3	8	asdf	54
4	9	asdfsd	87
5	10	asdfs	211

8.

```

23 --SELECT * FROM [dbo].[fitProcess];
24 SELECT * from account
25 SELECT * from assemblyAct
26 SELECT * from procAct
27 SELECT * from deptAct
28 SELECT * from jobTransaction
29 SELECT * from updates

```

Results **Messages**

	actNo
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

	actNo	startDate	assemblyCost
1	7	2019-01-01	0
2	8	2019-01-01	0
3	9	2019-01-01	0
4	10	2019-01-01	0

	actNo	startDate	procCost
1	4	2019-01-01	0
2	5	2019-01-01	0
3	6	2019-01-01	0

	actNo	startDate	deptCost
1	1	2019-01-01	0
2	2	2019-01-01	0
3	3	2019-01-01	0

	transNo	supCost

```
[INFO] test Java Application]
1
Please enter the sup cost:
10
Please enter the account number:
1
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
2
-----
Please enter a transation number:
2
Please enter the sup cost:
30
Please enter the account number:
2
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
-----
Please enter a transation number:
3
Please enter the sup cost:
22
Please enter the account number:
3
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
-----
Please enter a transation number:
4
Please enter the sup cost:
252
Please enter the account number:
4
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
-----
Please enter a transation number:
5
Please enter the sup cost:
252
Please enter the account number:
5
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
-----
Please enter a transation number:
6
Please enter the sup cost:
595
Please enter the account number:
6
Updating the table with user input...
Done. 1 rows inserted.
Do you want to add more? type 0 or 1
1
```

```
-----  
Please enter a transaction number:  
7  
Please enter the sup cost:  
889  
Please enter the account number:  
7  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
-----  
Please enter a transaction number:  
8  
Please enter the sup cost:  
515  
Please enter the account number:  
8  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
-----  
Please enter a transaction number:  
9  
Please enter the sup cost:  
659  
Please enter the account number:  
9  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
1  
-----  
Please enter a transaction number:  
10  
Please enter the sup cost:  
556  
Please enter the account number:  
10  
Updating the table with user input...  
Done. 1 rows inserted.  
Do you want to add more? type 0 or 1  
0
```

```

25  SELECT * from assemblyAct
26  SELECT * from procAct
27  SELECT * from deptAct
28  SELECT * from jobTransaction
29  SELECT * from updates

```

Results **Messages**

	actNo	startDate	assemblyCost
1	7	2019-01-01	889
2	8	2019-01-01	515
3	9	2019-01-01	659
4	10	2019-01-01	556

	actNo	startDate	procCost
1	4	2019-01-01	252
2	5	2019-01-01	252
3	6	2019-01-01	595

	actNo	startDate	deptCost
1	1	2019-01-01	10
2	2	2019-01-01	30
3	3	2019-01-01	22

	transNo	supCost
1	1	10
2	2	30
3	3	22
4	4	252
5	5	252
6	6	595
7	7	889
8	8	515
9	9	659
10	10	556

	transNo	actNo
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

9.

```
EXEC totalCostOnAssembly 'Customer'
```

30
31 select*FROM maintains
32
33
34
35 SELECT * FROM assemblyAct
36

Results Messages

	pid	asmID	deptId	actNo
1	NULL	NULL	10	1
2	NULL	NULL	11	2
3	NULL	NULL	12	3
4	20	NULL	NULL	4
5	21	NULL	NULL	5
6	22	NULL	NULL	6
7	NULL	1	NULL	7
8	NULL	2	NULL	8
9	NULL	2	NULL	9
10	NULL	3	NULL	10

	actNo	startDate	assemblyCost
1	7	2019-01-01	889
2	8	2019-01-01	515
3	9	2019-01-01	659
4	10	2019-01-01	556

```
18) Quite
9
-----
Please enter a assembly ID:
1
Cost
889
Do you want to add more? type 0 or 1
1
-----
Please enter a assembly ID:
2
Cost
1174
Do you want to add more? type 0 or 1
1
-----
Please enter a assembly ID:
3
Cost
556
Do you want to add more? type 0 or 1
```

10.

```

27  SELECT * FROM assignes
28  SELECT * FROM supervises
29  SELECT * FROM paintJob
30  SELECT * FROM cutJob
31  SELECT * FROM fitJob
32

```

Results **Messages**

	startDate	endDate	jobNo	pid	asmID
1	2019-09-05	2019-10-05	1	20	2
2	2019-09-05	2019-10-05	8	20	5
3	2019-09-05	2019-10-05	9	20	1

	pid	deptId
1	20	15
2	21	16
3	22	17
4	23	18
5	24	19
6	25	20
7	26	21
8	27	22
9	28	23
10	29	24

	jobId	details	color	labor	volume
1	1	asdf	red	20	10
2	2	asfdfsfdf	green	25	2
3	3	asdfsd	blue	44	5

	jobId	details	materials	machineType	labor	timeTaken
1	4	adsfd	44	large	2	23
2	5	asdfs	33	small	2	33

	jobId	details	labor
1	6	asdsf	22
2	7	asdfs	22
3	8	asdf	54
4	9	asdfsd	87
5	10	asdfs	211

```
17) Export customers in a range;
18) Quit
10
-----
Please enter a department ID:
15
Please enter the end date:
2019-10-5
labor
161
Do you want to add more? type 0 or 1
1
-----
Please enter a department ID:
16
Please enter the end date:
2019-10-5
labor
25
Do you want to add more? type 0 or 1
1
-----
Please enter a department ID:
17
Please enter the end date:
2019-10-5
labor
66
Do you want to add more? type 0 or 1
```

11.

```
40
41 UPDATE assignes
42 SET startDate = '2018-10-3'
43 WHERE jobNo = 9;
44
45 SELECT * FROM assignes|
```

Results Messages

	startDate	endDate	jobNo	pid	asmID
1	2018-03-12	2019-10-05	1	20	2
2	2018-03-02	2019-10-22	2	21	1
3	2018-04-02	2019-11-22	3	22	2
4	2018-05-02	2020-01-02	4	25	3
5	2018-06-24	2020-01-22	5	27	6
6	2018-07-14	2020-03-22	6	22	4
7	2018-11-14	2020-03-12	7	27	9
8	2019-09-05	2019-10-05	8	20	5
9	2018-10-03	2020-03-12	9	20	1
10	2019-09-05	2019-10-05	10	27	6

```
11
-----
Please enter a assembly ID:
1
asmID | start date | pid | did
1     | 2018-03-02 | 21  | 16
1     | 2018-03-12 | 20  | 15
1     | 2018-10-03 | 20  | 15
1     | 2019-09-05 | 20  | 15
Do you want to add more? type 0 or 1
1
-----
Please enter a assembly ID:
2
asmID | start date | pid | did
2     | 2018-03-12 | 20  | 15
2     | 2018-04-02 | 22  | 17
2     | 2018-07-14 | 22  | 17
2     | 2018-10-03 | 20  | 15
2     | 2019-09-05 | 20  | 15
Do you want to add more? type 0 or 1
1
-----
Please enter a assembly ID:
3
asmID | start date | pid | did
3     | 2018-05-02 | 25  | 20
Do you want to add more? type 0 or 1
```

12. ..

```

26  SELECT * FROM supervises
27  SELECT * FROM assignes
28  SELECT * FROM paintJob
29  SELECT * FROM cutJob
30  SELECT * FROM fitJob

```

Results **Messages**

	pid	deptId
1	20	15
2	21	16
3	22	17
4	23	18
5	24	19
6	25	20
7	26	21
8	27	22
9	28	23
10	29	24

	startDate	endDate	jobNo	pid	asmID
1	2018-03-12	2019-10-05	1	20	2
2	2018-03-02	2019-10-22	2	21	1
3	2018-04-02	2019-11-22	3	22	2
4	2018-05-02	2020-01-02	4	25	3
5	2018-06-24	2020-01-22	5	27	6
6	2018-07-14	2020-03-22	6	22	4
7	2018-11-14	2020-03-12	7	27	9
8	2019-09-05	2019-10-05	8	20	5
9	2018-10-03	2020-03-12	9	20	1
10	2019-09-05	2019-10-05	10	27	6

	jobId	details	color	labor	volume
1	1	asdf	red	20	10
2	2	asfdfsfdf	green	25	2
3	3	asdfsdf	blue	44	5

	jobId	details	materials	machineType	labor	timeTaken
1	4	adsfd	44	large	2	23
2	5	asdfs	33	small	2	33

	jobId	details	labor
1	6	asdsf	22
2	7	asdfs	22
3	8	asdf	54
4	9	asdfsdf	87

```
12
-----
Please enter a department ID:
15
Please enter the end date:
2019-10-05
=====
PAINT JOBS
DID|PID|asmID|end date|job ID|Details|color|labor|volume
15 | 20 | 2 | 2019-10-05 |1 | asdf | red | 20 | 10
=====
CUT JOBS
DID|PID|asmID|end date|job ID|Materials|labor|machine Type|time taken
null | null
=====
FIT JOBS
DID|PID|asmID|end date|job ID|Detials|labor
15 | 20 | 5 | 2019-10-05 | 8 | asdf | 54
Do you want to add more? type 0 or 1
1
-----
Please enter a department ID:
16
Please enter the end date:
2019-10-22
=====
PAINT JOBS
DID|PID|asmID|end date|job ID|Details|color|labor|volume
16 | 21 | 1 | 2019-10-22 |2 | asfdfdsdf | green | 25 | 2
=====
CUT JOBS
DID|PID|asmID|end date|job ID|Materials|labor|machine Type|time taken
null | null
=====
FIT JOBS
DID|PID|asmID|end date|job ID|Detials|labor
null | null | null | null | null | null | null
Do you want to add more? type 0 or 1
1
-----
Please enter a department ID:
22
Please enter the end date:
2020-03-12
=====
PAINT JOBS
DID|PID|asmID|end date|job ID|Details|color|labor|volume
null | null
=====
CUT JOBS
DID|PID|asmID|end date|job ID|Materials|labor|machine Type|time taken
null | null
=====
FIT JOBS
DID|PID|asmID|end date|job ID|Detials|labor
22 | 27 | 9 | 2020-03-12 | 7 | asdfs | 22
Do you want to add more? type 0 or 1
```

13.

```

tProcessTable          47
deleteCutJob           48   SELECT * FROM customer ORDER BY category
Job                    49
ProcessTable           50
tAssemblyProces...     51
tCustomers              52

tJobHistory             53
tJobHistory_cut         54
tJobHistory_fit         55
tJobHistory_paint       56
new_assembly_custo...    57
new_Customer            58
new_dept                 59
new_Job                  60
new_ProcDept             61
int_Job                  62
intProcessTable          63
cordTransaction           64
pervisingTable            65
talCostOnAssembly        66
talLaborOneDept           67
s                         68
e Triggers                69
es                        70

```

	Results	Messages	
	name	address	category
1	shahzad	test dr	1
2	tim	red st	2
3	tootoo	11 2	2
4	drake	green ave	3
5	ronald	toot st	4
6	shah	green	4
7	sam	toot	5
8	lexi	cat	5
9	reggy	apple ct	5
10	tabby	pop	6
11	frank	s	8

```
.p2\pri 18) Quit
.p2\pri 13
-----
.p2\pri Please enter a start range:
1
Please enter a end range:
4
name | address | category
drake | green ave | 3
ronald | toot st | 4
shah | green | 4
shahzad | test dr | 1
tim | red st | 2
tootoo | 11 2 | 2
Do you want to add more? type 0 or 1
1
-----
Please enter a start range:
3
Please enter a end range:
5
name | address | category
drake | green ave | 3
lexi | cat | 5
reggy | apple ct | 5
ronald | toot st | 4
sam | toot | 5
shah | green | 4
Do you want to add more? type 0 or 1
1
-----
Please enter a start range:
2
Please enter a end range:
8
name | address | category
drake | green ave | 3
frank | s | 8
lexi | cat | 5
reggy | apple ct | 5
ronald | toot st | 4
sam | toot | 5
shah | green | 4
tabby | pop | 6
tim | red st | 2
tootoo | 11 2 | 2
Do you want to add more? type 0 or 1
```

14.

```
14
-----
Please enter a start range:
0
Please enter a end range:
4
Do you want to delete more? type 0 or 1
1
-----
Please enter a start range:
0
Please enter a end range:
5
Do you want to delete more? type 0 or 1
1
-----
Please enter a start range:
0
Please enter a end range:
100
Do you want to delete more? type 0 or 1
```

```

47
48   SELECT * FROM cutJob
49   SELECT * FROM assignes
50   SELECT * FROM job
51

```

Results **Messages**

	jobId	details	materials	machineType	labor	timeTaken
1	5	asdfs	33	small	2	33
2	20	asdfs	10	small	22	16

	startDate	endDate	jobNo	pid	asmID
1	2018-06-24	2020-01-22	5	27	6
2	2018-07-14	2020-03-22	6	22	4
3	2018-11-14	2020-03-12	7	27	9
4	2019-09-05	2019-10-05	8	20	5
5	2018-10-03	2020-03-12	9	20	1
6	2019-09-05	2019-10-05	10	27	6
7	2010-03-11	2010-03-16	20	22	2

	jobId
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	20

```

45
46
47
48     SELECT * FROM cutJob
49     SELECT * FROM assignes
50     SELECT * FROM job

```

Results **Messages**

	jobId	details	materials	machineType	labor	timeTaken
1	4	adsfd	44	large	2	23
2	5	asdfs	33	small	2	33
3	20	asdfs	10	small	22	16

	startDate	endDate	jobNo	pid	asmID
1	2018-03-12	2019-10-05	1	20	2
2	2018-03-02	2019-10-22	2	21	1
3	2018-04-02	2019-11-22	3	22	2
4	2018-05-02	2020-01-02	4	25	3
5	2018-06-24	2020-01-22	5	27	6
6	2018-07-14	2020-03-22	6	22	4
7	2018-11-14	2020-03-12	7	27	9
8	2019-09-05	2019-10-05	8	20	5
9	2018-10-03	2020-03-12	9	20	1
10	2019-09-05	2019-10-05	10	27	6
11	2010-03-11	2010-03-16	20	22	2

	jobId
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	20

```

SS... ↗ Export as Notebook
ab...
47
48   SELECT * FROM cutJob
49   SELECT * FROM assignes
50   SELECT * FROM job
51

Results Messages
y...
y...    jobId  ↴  details  ↴  materials  ↴  machineType  ↴  labor  ↴  timeTaken  ↴
y...    1      20       asdfs        10          small        22          16

co...
t
unt    startDate  ↴  endDate  ↴  jobNo  ↴  pid  ↴  asmID  ↴
unt    1          2018-07-14  2020-03-22  6      22          4
unt    2          2018-11-14  2020-03-12  7      27          9
unt    3          2019-09-05  2019-10-05  8      20          5
unt    4          2018-10-03  2020-03-12  9      20          1
unt    5          2019-09-05  2019-10-05  10     27          6
unt    6          2010-03-11  2010-03-16  20     22          2

es...
    jobId  ↴
1      1
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9
10     10
11     20

```

bly

```
47
48 SELECT * FROM cutJob
49 SELECT * FROM assignes
50 SELECT * FROM job
51
```

Results Messages

jobId	details	materials	machineType	labor	timeTaken
-------	---------	-----------	-------------	-------	-----------

startDate	endDate	jobNo	pid	asmID
-----------	---------	-------	-----	-------

	jobId ▼
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	20

15.

```

47
48     SELECT * FROM paintJob
49

```

Results **Messages**

	jobId	details	color	labor	volume
1	1	asdf	red	20	10
2	2	asdfdfsdf	green	25	2
3	3	asdfsdsd	blue	44	5

```

izadAnsari\p2\pa
application] 16) Import: enter new customers from a data file until the file is empty;
application] 17) Export customers in a range;
application] 18) Quit
application] 15
Please enter a job ID:
1
Please enter a color:
chartruse
Do you want to add more? type 0 or 1
1
Please enter a job ID:
2
Please enter a color:
purple
Do you want to add more? type 0 or 1
1
Please enter a job ID:
3
Please enter a color:
indigo
Do you want to add more? type 0 or 1
0

```

```

47
48     SELECT * FROM paintJob
49

```

Results **Messages**

	jobId	details	color	labor	volume
1	1	asdf	chartruse	20	10
2	2	asdfdfsdf	purple	25	2
3	3	asdfsdsd	indigo	44	5

16.

```
50      SELECT * FROM customer
51

Results    Messages
```

	name	address	category
1	drake	green ave	3
2	frank	s	8
3	lexi	cat	5
4	reggy	apple ct	5
5	ronald	toot st	4
6	sam	toot	5
7	shah	green	4
8	shahzad	test dr	1
9	tabby	pop	6
10	tim	red st	2
11	tootoo	11 2	2

```
16
Please enter the file info in this format location\filename.csv format
C:\\Users\\ShahzadAnsari\\Documents\\School\\DSA-4513\\Project\\customer.csv
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
Updating the table with user input...
Done. 1 rows inserted.
```

Export as Notebook

... 52 SELECT * FROM customer 53

Results Messages

	name	address	category
1	drake	green ave	3
2	frank	s	8
3	frenchie	121 st.	2
4	karl	loopy ave	3
5	lexi	cat	5
6	marx	froo ave	4
7	reggy	apple ct	5
8	ronald	toot st	4
9	sam	toot	5
10	shah	green	4
11	shahzad	test dr	1
12	tabby	pop	6
13	tim	red st	2
14	tootoo	11 2	2

17.

```
18) Quit
17
Please enter a start range:
3
Please enter a end range:
8
Please enter the export location and file name in location\filename.csv format
C:\\Users\\ShahzadAnsari\\Documents\\School\\DSA-4513\\Project\\customer1.csv
```

A	B	C	D
1	name	address	category
2	drake	green ave	3
3	frank	s	8
4	karl	loopy ave	3
5	lexi	cat	5
6	marx	froo ave	4
7	reggy	apple ct	5
8	ronald	toot st	4
9	sam	toot	5
10	shah	green	4
11	tabby	pop	6
12			
13			

18.

```
a  
invalid input: a  
try again!  
  
Please select one of the options below:  
1) Enter a new customer;  
2) Enter a new department;  
3) Enter a new process-id and its department together with its type and information relevant to the type;  
4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered and associate it with one or  
5) Create a new account and associate it with the process, assembly, or department to which it is  
applicable;  
6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced ;  
7) At the completion of a job, enter the date it completed and the information relevant to the type  
of job;  
8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected  
accounts by adding sup-cost to their current values of details;  
9) Retrieve the total cost incurred on an assembly-id;  
10) Retrieve the total labor time within a department for jobs completed in the department during a  
given date;  
11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department ,  
12) Retrieve the jobs (together with their type information and assembly-id) completed during a  
given date in a given department;  
13) Retrieve the customers (in name order) whose category is in a given range;  
14) Delete all cut-jobs whose job-no is in a given range;  
15) Change the color of a given paint job;  
16) Import: enter new customers from a data file until the file is empty;  
17) Export customers in a range;  
18) Quit  
18  
Exiting! Goodbye!
```

Error Examples

Duplicate PK in Dept.

```
18) Quit
2
Please enter a ID:
10
Please enter the department data:
asdfs
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_
```

```
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:197)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:100)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:150)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PreparedStmtProxy.execute(IOBuffer.java:74)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:74)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:370)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:214)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:197)
```

Invalid category range

```
18) Quit
1
Please enter a customer name:
33
Please enter the address:
1
Please enter category from 1 to 10:
33
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint
```

```
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:197)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:100)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:150)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PreparedStmtProxy.execute(IOBuffer.java:74)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:74)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:370)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:214)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:197)
```

Invalid Date

```
18) Quit
12
-----
Please enter a department ID:
999
Please enter the end date:
3
Exception in thread "main" java.lang.IllegalArgumentException
at java.sql/java.sql.Date.valueOf(Date.java:141)
at HW3test.Q12(HW3test.java:986)
at HW3test.main(HW3test.java:98)
```

JSP CODE**Data Handler**

```

package jsp_project;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DataHandler {

    private Connection conn;

    // Azure SQL connection credentials
    final static String HOSTNAME = " ";
    final static String DBNAME = " ";
    final static String USERNAME = " ";
    final static String PASSWORD = "";

    // Resulting connection string
    final private String url =
        String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
                      HOSTNAME, DBNAME, USERNAME,
                      PASSWORD);

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

    //-----
    public boolean addCustomer(

```

```
String name, String address, Integer category) throws SQLException {
    getDBConnection();
    final String sqlQuery =
        "INSERT INTO customer " +
        "(name, address , category) " +
        "VALUES " +
        "(?, ?, ?)";
    final PreparedStatement statement = conn.prepareStatement(sqlQuery);

    statement.setString(1, name);
    statement.setString(2, address);
    statement.setInt(3, category);
    return statement.executeUpdate() == 1;
}

//-----
public ResultSet getCustomers(Integer start, Integer end) throws SQLException {
    getDBConnection();
    final PreparedStatement statement = conn.prepareStatement("SELECT * FROM customer where category >= ? AND
category <= ? ORDER BY name ;");
    statement.setInt(1, start);
    statement.setInt(2, end);
    ResultSet result = statement.executeQuery();
    return result ;
}
}
```

Get Customers HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Get Customers</title>
</head>
<body>
<h2>Get Customers</h2>
<!--
Page to collect Customer info.

-->
<form action="Shahzad_Ansari_IP_Task7_getCustomers_Confermation.jsp">
<!-- The form organized in an HTML table for better clarity. -->
<table border=1>
<tr>
<th colspan="2">Enter the start category:</th>
</tr>
<tr>
<td>Start:</td>
<td><div style="text-align: center;">
<input type=text name=start>
</div></td>
</tr>
<tr>
<td>End:</td>
<td><div style="text-align: center;">
<input type=text name=end>
</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=Search>
</div></td>
</tr>
</table>
</form>
</body>
</html>

```

GetCustomer Confirmation

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
<%@page import="jsp_project.Shahzad_Ansari_IP_Task7_DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
// The handler is the one in charge of establishing the connection.
DataHandler handler = new DataHandler();
// Get the attribute values passed from the input form.
String start = request.getParameter("start");
String end = request.getParameter("end");

// Now perform the query with the data from the form.
ResultSet rs = handler.getCustomers(Integer.valueOf(start),Integer.valueOf(end));
%>

```

```

<!-- The table for customer records -->





```

insertCustomer HTML

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="UTF-8">
    <title>Add Customer</title>
  </head>
  <body>
    <h2>Add Customer</h2>

    <form action="Shahzad_Ansari_IP_Task7_insertCustomer_Confirmation.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>name:</td>
          <td><div style="text-align: center;">
            <input type=text name=name>
          </div></td>
        </tr>
        <tr>
          <td>address:</td>
          <td><div style="text-align: center;">
            <input type=text name=address>
          </div></td>
        </tr>
        <tr>
          <td>category:</td>
          <td><div style="text-align: center;">
            <input type=text name=category>
          </div></td>
        </tr>
        <tr>
```

```

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

```

InsertCustomer Confirmation

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
<%@page import="jsp_project.Shahzad_Ansari_IP_Task7_DataHandler"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Array"%>
<%
DataHandler handler = new DataHandler();
String name = request.getParameter("name");
String address = request.getParameter("address");
String category = request.getParameter("category");

boolean inserted = handler.addCustomer(name, address, Integer.valueOf(category));
if (!inserted) {
%>

```

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

<%
}

%>
</body>
</html>
```

JSP EXECUTION

Get Customers

Enter the start category:		
Start:	2	
End:	8	<input type="button" value="x"/>
<input type="button" value="Clear"/>	<input type="button" value="Search"/>	

Name	Address	Category
drake	green ave	3
frank	s	8
frenchie	121 st.	2
karl	loopy ave	3
lexi	cat	5
marx	froo ave	4
reggy	apple ct	5
ronald	toot st	4
sam	toot	5
shah	green	4
tabby	pop	6
tim	red st	2
tootoo	11 2	2

Name	Address	Category
drake	green ave	3
frank	s	8
frenchie	121 st.	2
garthBrooks	123 pensicola ave	5
karl	loopy ave	3
lexi	cat	5
marx	froo ave	4
reggy	apple ct	5
ronald	toot st	4
sam	toot	5
shah	green	4
tabby	pop	6
tim	red st	2
tootoo	11 2	2

Add Customer

Enter the customer data:	
name:	garthBrooks
address:	123 pensicola ave
category:	5 <input type="button" value="x"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

- Name: garthBrooks
- Address: 123 pensicola ave
- Category: 5

Was successfully inserted.