

JOB-SHOP ACCOUNTING DATABASE SYSTEM

COURSE: CS/DSA-4513

Database Management Systems

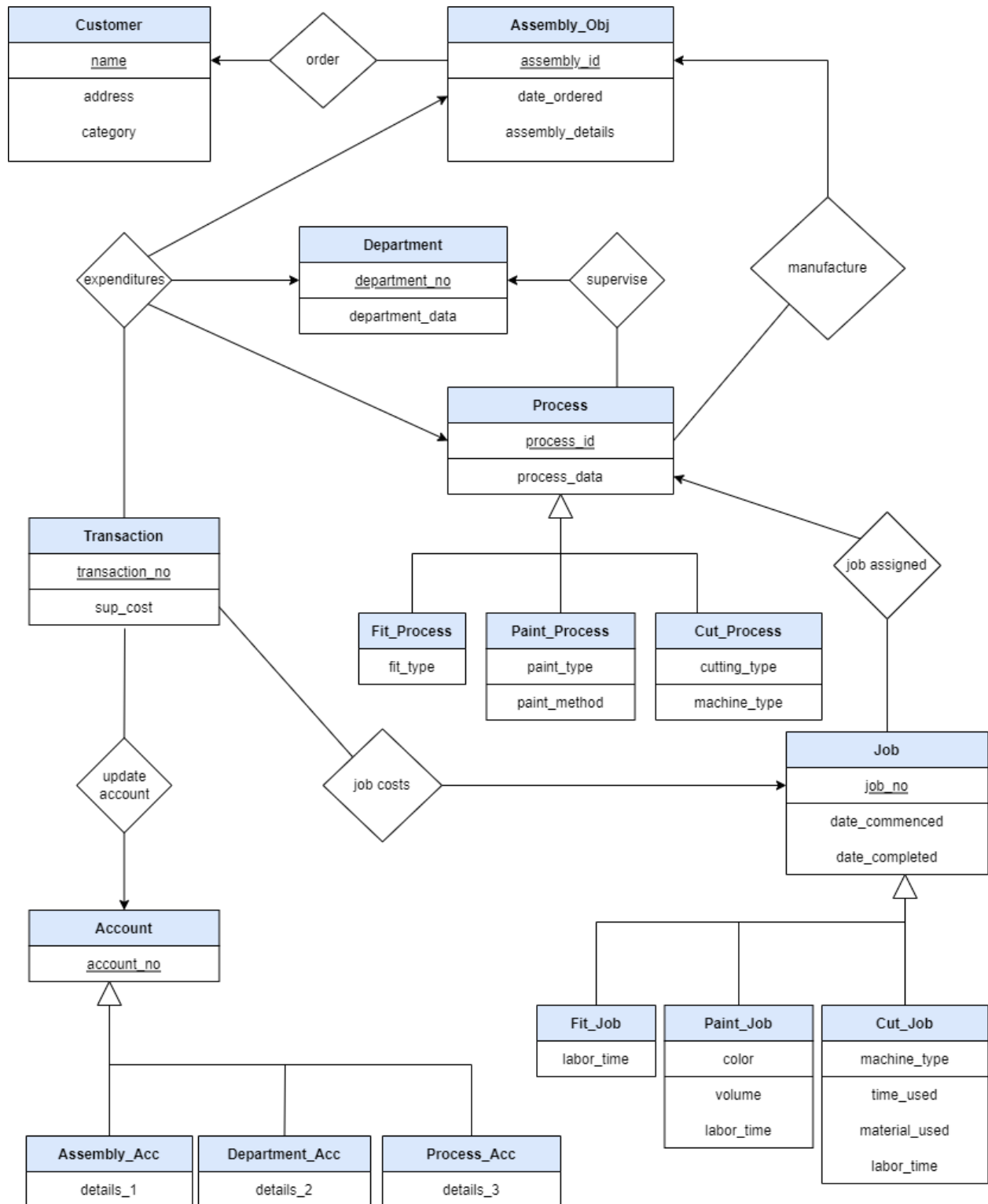
SECTION: 001

SEMESTER: Fall 2021

INSTRUCTOR: Dr. Le Gruenwald

AUTHOR: Steven Johnson, 113484715
steven.a.johnson-1@ou.edu

Tasks Performed	Page Number
Task 1. Conceptual Design	2-3
1.1. ER Diagram	2
1.2. Relational Database Schema	3
Task 2. Data Dictionary	4-5
Task 3. Storage Structures to Optimize Queries	6-9
3.1. Discussion of storage structures for tables	6-8
3.2. Discussion of storage structures for tables (Azure SQL Database)	9
Task 4. Table Creation in Azure SQL	10-15
Task 5. Queries 1-15 in Azure SQL	16-48
5.1. SQL statements	16-31
5.2. The Java source program and screenshots showing its successful compilation	32-48
Task 6. Query Testing in Java source program	49-86
6.1. Screenshots showing the testing of query 1	49-50
6.2. Screenshots showing the testing of query 2	51-53
6.3. Screenshots showing the testing of query 3	54-58
6.4. Screenshots showing the testing of query 4	59-63
6.5. Screenshots showing the testing of query 5	64-67
6.6. Screenshots showing the testing of query 6	68-70
6.7. Screenshots showing the testing of query 7	71-73
6.8. Screenshots showing the testing of query 8	74
6.9. Screenshots showing the testing of query 9	75
6.10. Screenshots showing the testing of query 10	76
6.11. Screenshots showing the testing of query 11	77
6.12. Screenshots showing the testing of query 12	78
6.13. Screenshots showing the testing of query 13	79
6.14. Screenshots showing the testing of query 14	80
6.15. Screenshots showing the testing of query 15	81-82
6.16. Screenshots showing the testing of the import and export options	83-84
6.17. Screenshots showing the testing of the quit option	85
6.18. Screenshots showing the testing of three types of errors	86
Task 7.	87-95
7.1. Web database application source program and screenshots showing its successful compilation	87-93
7.2. Screenshots showing the testing of the web database application	94-95



Customer(name, address, category)

Account(account_no)

Assembly_Acc(account_no, details_1)

Department_Acc(account_no, details_2)

Process_Acc(account_no, details_3)

Assembly_Obj(assembly_id, name, account_no, date_ordered, assembly_details)

Department(department_no, account_no, department_data)

Process(process_id, department_no, process_data)

Fit_Process(process_id, fit_type)

Paint_Process(process_id, paint_type, paint_method)

Cut_Process(process_id, cutting_type, machine_type)

Job(job_no, process_id, date_commenced, date_completed)

Fit_Job(job_no, labor_time)

Paint_Job(job_no, color, volume, labor_time)

Cut_Job(job_no, machine_type, time_used, material_used, labor_time)

Transaction_Obj(transaction_no, assembly_id, process_id, department_no, job_no, sup_cost)

Data Dictionary

Attribute	Type	Size(in bytes)	Constraints
name	varchar(64)	66	PK, FK
address	varchar(64)	66	
category	int	4	category >= 1 AND category <= 10
account_no	int	4	PK, FK
details_1	varchar(64)	66	
details_2	varchar(64)	66	
details_3	varchar(64)	66	
assembly_id	int	4	PK, FK
date_ordered	varchar(10)	12	
assembly_details	varchar(64)	66	
department_no	int	4	PK, FK
process_id	int	4	PK, FK
process_data	varchar(64)	66	
fit_type	varchar(64)	66	
paint_type	varchar(64)	66	
paint_method	varchar(64)	66	
cutting_type	varchar(64)	66	
machine_type	varchar(64)	66	
department_data	varchar(64)	66	
job_no	int	4	PK, FK
date_commenced	varchar(10)	12	
date_completed	varchar(10)	12	
labor_time	int	4	

color	varchar(64)	66	
volume	int	4	
time_used	int	4	
material_used	varchar(64)	66	
transaction_no	int	4	PK
sup_cost	int	4	

Table Name	Query # and Type	Search Key	Query Frequency	Selected File Organization	Justification
Customer	Q1 Insertion Q13 Range search	Q1 N/A Q13 category	Q1 30/day Q13 100/day	Index Sequential on search key category	This is the best type of file organization when you have many range searches, and although it is not very good with insertions, those are not nearly as frequent.
Assembly_Obj	Q4 Insertion Q12 Random search	Q4 N/A Q12 assembly_id	Q4 40/day Q12 20/day	Extendible Hashing with hashkey assembly_id	Hashing is good for a table with many random searches, and because there are also a lot of insertions, extendible was chosen over static.
Process	Q3 Insertion Q4 Random search Q5 Random search Q6 Random search Q9 Random search Q10 Random search Q11 Random search Q12 Random search	Q3 N/A Q4 process_id Q5 process_id Q6 process_id Q9 assembly_id Q10 process_id Q11 assembly_id Q12 department_no	Q3 Infrequent Q4 40/day Q5 10/day Q6 50/day Q9 200/day Q10 20/day Q11 100/day Q12 20/day	Static Hashing with hashkey assembly_id	Hashing is good for a table with many random searches, and there are not very many insertions happening, so static hashing works fine. The vast majority of the searches are done with a search key on assembly_id, so that attribute was chosen for the hashkey.
Fit_Process	Q3 Insertion Q11 Random search	Q3 N/A Q11 assembly_id	Q3 Infrequent Q11 100/day	Static Hashing with hashkey assembly_id	Same as "Process." Hashing is good for a table with many random searches, and there are not very many insertions happening, so static hashing works fine.
Paint_Process	Q3 Insertion Q11 Random search	Q3 N/A Q11 assembly_id	Q3 Infrequent Q11 100/day	Static Hashing with hashkey assembly_id	Same as "Process." Hashing is good for a table with many random searches, and there are not very many insertions happening, so static hashing works fine.
Cut_Process	Q3 Insertion Q11 Random search	Q3 N/A Q11 assembly_id	Q3 Infrequent Q11 100/day	Static Hashing with hashkey assembly_id	Same as "Process." Hashing is good for a table with many random searches, and there are not very many insertions happening, so static hashing works fine.
Department	Q2 Insertion	Q2 N/A	Q2 Infrequent	Heap file	Not many insertions means there is no reason to keep the table sorted. This will make the insertions very quick with few downsides since the table never needs to be searched.
Job	Q6 Insertion Q7 Random search Q9 Random search Q10 Random search Q12 Random search	Q6 N/A Q7 job_no Q9 job_no Q10 date_completed Q12 date_completed	Q6 50/day Q7 50/day Q9 200/day Q10 20/day Q12 20/day	Extendible hashing with hashkey job_no	When we first create a job, we don't know when it will complete, so sorting via date_completed is tricky. There are a lot of random searches happening and the job_no attribute is relevant to most of them. Since we always know the value of that attribute when first creating a job, that is the hashkey I chose.

	Q14 Deletion Q14 Range search Q15 Random search	Q14 job_no Q14 job_no Q15 job_no	Q14 1/month Q14 1/month Q15 1/week		
Fit_Job	Q7 Insertion Q10 Random search Q12 Random search	Q7 N/A Q10 date_completed Q12 date_completed	Q7 50/day Q10 20/day Q12 20/day	Extendible hashing with hashkey job_no	Same as "Job." When we first create a job, we don't know when it will complete, so sorting via date_completed is tricky. Since we always know the value of job_no, that is the hashkey I had to choose. Because there are quite a few insertions happening alongside these searches, I chose extendible hashing over static hashing.
Paint_Job	Q7 Insertion Q10 Random search Q12 Random search Q15 Random search	Q7 N/A Q10 date_completed Q12 date_completed Q15 job_no	Q7 50/day Q10 20/day Q12 20/day Q15 1/week	Extendible hashing with hashkey job_no	Once again, there are a lot of random searches happening and the job_no attribute is actually relevant to one of them. Since we always know the value of that attribute when first creating a job, that is the hashkey I chose. Because there are quite a few insertions happening alongside these searches, I chose extendible hashing over static hashing.
Cut_Job	Q7 Insertion Q10 Random search Q12 Random search Q14 Deletion Q14 Range search	Q7 N/A Q10 date_completed Q12 date_completed Q14 job_no Q14 job_no	Q7 50/day Q10 20/day Q12 20/day Q14 1/month Q14 1/month	Extendible hashing with hashkey job_no	This one is tough to choose since there is a little bit of every query type going on. When we first create a job, we don't know when it will complete, so sorting via date_completed is tricky. The job_no attribute is still relevant to many searches, and since we know what that attribute is when first creating a job, that is the hashkey I chose. Because there are quite a few insertions happening alongside these searches, I chose extendible hashing over static hashing.
Transaction_Obj	Q8 Insertion Q9 Random search	Q8 N/A Q9 job_no	Q8 50/day Q9 200/day	Extendible hashing with hashkey job_no	A lot of random searches alongside a lot of insertions is handled best by extendible hashing.
Account	Q5 Insertion Q8 Random search	Q5 N/A Q8 account_no	Q5 10/day Q8 50/day	Extendible hashing with hashkey account_no	There are a lot of random searches, but there are also quite a few insertions. This ratio is generally handled best by extendible hashing.
Assembly_Acc	Q5 Insertion Q8 Random search	Q5 N/A Q8 account_no	Q5 10/day Q8 50/day	Extendible hashing with hashkey account_no	Same as "Account." There are a lot of random searches, but there are also quite a few insertions. This ratio is generally handled best by extendible hashing.
Department_Acc	Q5 Insertion Q8 Random search	Q5 N/A Q8 account_no	Q5 10/day Q8 50/day	Extendible hashing with hashkey account_no	Same as "Account." There are a lot of random searches, but there are also quite a few insertions. This ratio is generally handled best by extendible hashing.
Process_Acc	Q5	Q5	Q5	Extendible hashing	Same as "Account." There are a lot of random

	Insertion Q8 Random search	N/A Q8 account_no	10/day Q8 50/day	with hashkey account_no	searches, but there are also quite a few insertions. This ratio is generally handled best by extendible hashing.
--	----------------------------------	-------------------------	------------------------	----------------------------	--

Storage Structures implemented in Azure SQL Database

By default, Azure uses B-trees to create indexes. After looking through the documentation on docs.microsoft.com it looked like there was a way to implement a hash table on an index. For each above table that was marked “Static hashing” or “Extendible hashing” I attempted to implement a hash table and stick to the default of B-trees for the rest.

When I went to actually implement these tables with hashing, I ran into some problems. Creating a hash index in Azure Data Studio requires the corresponding table to be “memory optimized.” This would normally be an easy fix, but when I tried to make the table memory optimized, the console gave me the following error:

```
'MEMORY_OPTIMIZED tables' is not supported in this service tier of the database. See Books Online for more details on feature support in different service tiers of Windows Azure SQL Database.
```

It seems with the free student account that I use for this class, I cannot have Azure create a memory optimized table. I would need to subscribe to a higher service tier to do so.

Thus, I was unable to implement any hash tables for my indexes. When creating a primary key, an index is automatically created in the underlying B-tree structure that Azure defaults to. Because of this, I did not do any fancy index manipulation to optimize my queries.

Table Creation in Azure SQL

-- While working on the database design, it's useful to start from scratch every time
 -- Hence, we drop tables in reverse order they are created (so the foreign key constraints are not violated)

```
DROP TABLE IF EXISTS Transaction_Obj
DROP TABLE IF EXISTS Cut_Job
DROP TABLE IF EXISTS Paint_Job
DROP TABLE IF EXISTS Fit_Job
DROP TABLE IF EXISTS Job
DROP TABLE IF EXISTS Cut_Process
DROP TABLE IF EXISTS Paint_Process
DROP TABLE IF EXISTS Fit_Process
DROP TABLE IF EXISTS Process
DROP TABLE IF EXISTS Department
DROP TABLE IF EXISTS Assembly_Obj;
DROP TABLE IF EXISTS Process_Acc
DROP TABLE IF EXISTS Department_Acc
DROP TABLE IF EXISTS Assembly_Acc
DROP TABLE IF EXISTS Account
DROP TABLE IF EXISTS Customer;
```

-- Create tables

```
CREATE TABLE Customer (
  name VARCHAR(64) PRIMARY KEY,
  address VARCHAR(64),
  category INT CHECK (category >= 1 AND category <= 10),
);
```

```
CREATE TABLE Account (
  account_no INT PRIMARY KEY,
);
```

```
CREATE TABLE Assembly_Acc (
  account_no INT PRIMARY KEY,
  details_1 INT,
  CONSTRAINT FK_assembly_account_no FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Department_Acc (
  account_no INT PRIMARY KEY,
  details_2 INT,
  CONSTRAINT FK_dept_account_no FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Process_Acc (
  account_no INT PRIMARY KEY,
  details_3 INT,
  CONSTRAINT FK_proc_account_no FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Assembly_Obj (
  assembly_id INT PRIMARY KEY,
  date_ordered VARCHAR(10),
  assembly_details VARCHAR(64),
  name VARCHAR(64),
  account_no INT,
  CONSTRAINT FK_customer_name_on_order FOREIGN KEY (name) REFERENCES Customer,
  CONSTRAINT FK_assembly_account_no_to_use FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Department (
  department_no INT PRIMARY KEY,
  department_data VARCHAR(64),
  account_no INT,
  CONSTRAINT FK_dept_account_no_to_use FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Process (
  process_id INT PRIMARY KEY,
  process_data VARCHAR(64),
  assembly_id INT,
  department_no INT,
  account_no INT,
  CONSTRAINT FK_manufactured_assembly_id FOREIGN KEY (assembly_id) REFERENCES Assembly_Obj,
  CONSTRAINT FK_proc_supervisor_dept_no FOREIGN KEY (department_no) REFERENCES Department,
  CONSTRAINT FK_proc_account_no_to_use FOREIGN KEY (account_no) REFERENCES Account,
);
```

```
CREATE TABLE Fit_Process (
  process_id INT PRIMARY KEY,
  fit_type VARCHAR(64),
  CONSTRAINT FK_fit_proc_id FOREIGN KEY (process_id) REFERENCES Process,
);
```

```
CREATE TABLE Paint_Process (
  process_id INT PRIMARY KEY,
  paint_type VARCHAR(64),
  paint_method VARCHAR(64),
  CONSTRAINT FK_paint_proc_id FOREIGN KEY (process_id) REFERENCES Process,
);
```

```
CREATE TABLE Cut_Process (
  process_id INT PRIMARY KEY,
  cutting_type VARCHAR(64),
  machine_type VARCHAR(64),
  CONSTRAINT FK_cut_proc_id FOREIGN KEY (process_id) REFERENCES Process,
);
```

```
CREATE TABLE Job (
  job_no INT PRIMARY KEY,
  date_commenced VARCHAR(10),
  date_completed VARCHAR(10),
  process_id INT,
  CONSTRAINT FK_proc_id_for_parent_of_job FOREIGN KEY (process_id) REFERENCES Process,
);
```

```
CREATE TABLE Fit_Job (
  job_no INT PRIMARY KEY,
  labor_time INT,
  CONSTRAINT FK_fit_job_no FOREIGN KEY (job_no) REFERENCES Job,
);
```

```
CREATE TABLE Paint_Job (
  job_no INT PRIMARY KEY,
  color VARCHAR(64),
  volume INT,
  labor_time INT,
  CONSTRAINT FK_paint_job_no FOREIGN KEY (job_no) REFERENCES Job,
);
```

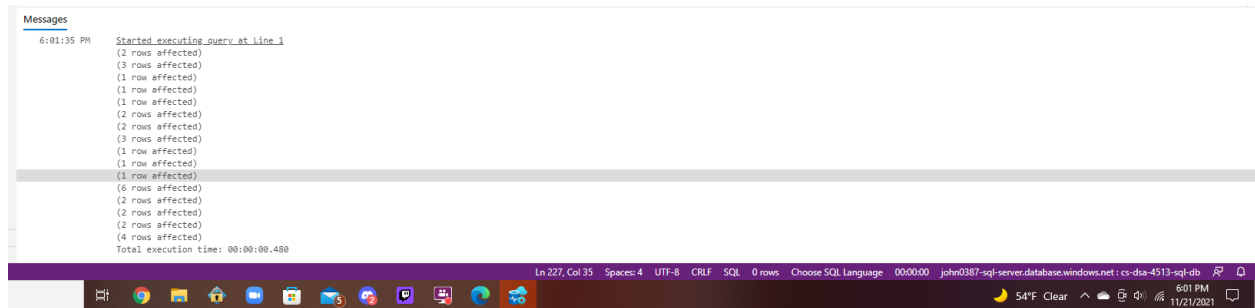
```
CREATE TABLE Cut_Job (
  job_no INT PRIMARY KEY,
  machine_type VARCHAR(64),
  time_used INT,
  material_used VARCHAR(64),
  labor_time INT,
  CONSTRAINT FK_cut_job_no FOREIGN KEY (job_no) REFERENCES Job,
);
```

```
CREATE TABLE Transaction_Obj (
  transaction_no INT,
  sup_cost INT,
  assembly_id INT,
  process_id INT,
  department_no INT,
  job_no INT,
```

```

CONSTRAINT FK_expenditures_from_assembly_id FOREIGN KEY (assembly_id) REFERENCES
Assembly_Obj,
CONSTRAINT FK_expenditures_from_proc_id FOREIGN KEY (process_id) REFERENCES Process,
CONSTRAINT FK_expenditures_from_dept_no FOREIGN KEY (department_no) REFERENCES Department,
CONSTRAINT FK_expenditures_from_job_no FOREIGN KEY (job_no) REFERENCES Job,
);

```



After the table declarations, I created some sample data to help with debugging queries 1-15, which were created as follows:

```
-- Create sample data for each table
```

```

INSERT INTO Customer
VALUES
('Jackson', '489 Stinson St', 5),
('Taylon', '331 NW 35th St', 3);

```

```
INSERT INTO Account
```

```
VALUES
```

```

(1),
(2),
(3);

```

```
INSERT INTO Assembly_Acc
```

```
VALUES
```

```
(1, 50);
```

```
INSERT INTO Department_Acc
```

```
VALUES
```

```
(2, 100000);
```

```
INSERT INTO Process_Acc
```

```
VALUES
```

```
(3, 150);
```

```
INSERT INTO Assembly_Obj
```

```
VALUES
```

```
(100, '11/19/2021', 'do this and that or something like that', 'Jackson', 1),  
(65, '11/20/2021', 'the best assembly around', 'Taylor', 1);
```

```
INSERT INTO Department
```

```
VALUES
```

```
(1, 'The first department', 2),  
(2, 'The second department', 2);
```

```
INSERT INTO Process
```

```
VALUES
```

```
(99, 'processes', 100, 1, 3),  
(10, 'are', 65, 2, 3),  
(26, 'cool', 65, 2, 3);
```

```
INSERT INTO Fit_Process
```

```
VALUES
```

```
(99, 'fit-type');
```

```
INSERT INTO Paint_Process
```

```
VALUES
```

```
(10, 'paint-type', 'paint-method');
```

```
INSERT INTO Cut_Process
```

```
VALUES
```

```
(26, 'cutting-type', 'machine-type');
```

```
INSERT INTO Job
```

```
VALUES
```

```
(1, '11/16/2021', '11/21/2021', 99),  
(2, '11/17/2021', '11/21/2021', 10),  
(3, '11/18/2021', '11/21/2021', 26),  
(4, '11/19/2021', NULL, 26),  
(5, '11/20/2021', NULL, 10),  
(6, '11/21/2021', NULL, 99);
```

```
INSERT INTO Fit_Job
```

```
VALUES
```

```
(1, 55),  
(6, 45);
```

```
INSERT INTO Paint_Job
```

```
VALUES
```

```
(2, 'color: red', 50, 135),  
(5, 'color: blue', 50, 160);
```

```
INSERT INTO Cut_Job
```

```
VALUES
```

```
(3, 'machine-type', 50, 'material used', 20),  
(4, 'machine-type', 40, 'material used', 30);
```

```
INSERT INTO Transaction_Obj
```

```
VALUES
```

```
(201, 50, NULL, NULL, NULL, 1),  
(202, 35, NULL, NULL, NULL, 2),  
(203, 25, NULL, NULL, NULL, 3),  
(204, 1300, NULL, NULL, 1, NULL);
```


Queries 1-15 in Azure SQL

-- Query 1

INSERT INTO Customer

VALUES

('Parker', '324 NW Pen St', 5);

Results		Messages	
	name ▾	address ▾	category ▾
1	Jackson	489 Stinson St	5
2	Parker	324 NW Pen St	5
3	Taylor	331 NW 35th St	3

-- Query 2

INSERT INTO Department

VALUES

(3, 'The third department', NULL);

Results		Messages	
	department_no ▾	department... ▾	account_no ▾
1	1	The first depa...	2
2	2	The second dep...	2
3	3	The third depa...	NULL

-- Query 3

INSERT INTO Process

VALUES

(16, 'very', NULL, 2, NULL);

Results		Messages			
	process_id	process_data	assembly_id	department_no	account_no
1	10	are	65	2	3
2	16	very	NULL	2	NULL
3	26	cool	65	2	3
4	99	processes	100	1	3

INSERT INTO Fit_Process

VALUES

(16, 'the best fit type');

Results		Messages	
	process_id	fit_type	
1	16	the best fit type	
2	99	fit-type	

-- Query 4

INSERT INTO Assembly_Obj

VALUES

(138, '11/21/2021', 'description of assembly', 'Parker', NULL);

Results		Messages			
	assembly_id	date_ordered	assembly_details	name	account_no
1	65	11/20/2021	the best assembly around	Taylor	1
2	100	11/19/2021	do this and that or somethin...	Jackson	1
3	138	11/21/2021	description of assembly	Parker	NULL

UPDATE Process

SET assembly_id = 138

WHERE process_id = 16;

Results		Messages			
	process_id	process_data	assembly_id	dep...	account_no
1	10	are	65	2	3
2	16	very	138	2	NULL
3	26	cool	65	2	3
4	99	processes	100	1	3

```
-- Query 5
INSERT INTO Account
VALUES
(4);
```

Results		Messages	
	account_no		
1	1		
2	2		
3	3		
4	4		

```
INSERT INTO Process_Acc
VALUES
(4, 10000);
```

Results		Messages	
	account_no	details_3	
1	3	150	
2	4	10000	

```
UPDATE Process
SET account_no = 3
WHERE process_id = 16;
```

Results		Messages			
	process_id	process_data	assembly_id	department_no	account_no
1	10	are	65	2	3
2	16	very	138	2	3
3	26	cool	65	2	3
4	99	processes	100	1	3

-- Query 6

INSERT INTO Job

VALUES

(7, '11/20/2021', NULL, 16);

Results		Messages		
	job_no	date_commenced	date_completed	process_id
1	1	11/16/2021	11/21/2021	99
2	2	11/17/2021	11/21/2021	10
3	3	11/18/2021	11/21/2021	26
4	4	11/19/2021	NULL	26
5	5	11/20/2021	NULL	10
6	6	11/21/2021	NULL	99
7	7	11/20/2021	NULL	16

UPDATE Process

SET assembly_id = 138

WHERE process_id = 16;

Results		Messages			
	process_id	process_data	assembly_id	department_no	account_no
1	10	are	65	2	3
2	16	very	138	2	3
3	26	cool	65	2	3
4	99	processes	100	1	3

```
-- Query 7
UPDATE Job
SET date_completed = '11/21/2021'
WHERE job_no = 7;
```

Results		Messages		
	job_no	date_commenced	date_completed	process_id
1	1	11/16/2021	11/21/2021	99
2	2	11/17/2021	11/21/2021	10
3	3	11/18/2021	11/21/2021	26
4	4	11/19/2021	NULL	26
5	5	11/20/2021	NULL	10
6	6	11/21/2021	NULL	99
7	7	11/20/2021	11/21/2021	16

```
INSERT INTO Fit_Job
VALUES
(7, 55);
```

Results

Messages

	job_no	labor_time
1	1	55
2	6	45
3	7	55

```
-- Query 8
INSERT INTO Transaction_Obj
VALUES
(205, 120, NULL, NULL, NULL, 7);
```

Results		Messages				
	transaction_no	sup_cost	assembly_id	process_id	department_no	job_no
1	201	50	NULL	NULL	NULL	1
2	202	35	NULL	NULL	NULL	2
3	203	25	NULL	NULL	NULL	3
4	204	1300	NULL	NULL	1	NULL
5	205	120	NULL	NULL	NULL	7

```
UPDATE Process_Acc
SET details_3 = details_3 +
(SELECT sup_cost
FROM Transaction_Obj
WHERE transaction_no = 205)
WHERE account_no = 3;
```

Results		Messages	
	account_no	details_3	
1	3	270	
2	4	10000	

```
-- Query 9
SELECT SUM(ISNULL(S.sup_cost, 0))
FROM
  (
    SELECT sup_cost
    FROM Transaction_Obj
    WHERE
      job_no IN(
        SELECT job_no
        FROM Job
        WHERE process_id IN(
          SELECT process_id
          FROM Process
          WHERE
            assembly_id = 138)
      )
  ) AS S;
```

	(No c... ▼
1	120


```
-- Query 10
SELECT SUM(S.labor_time)
FROM
(
  SELECT job_no, labor_time FROM Fit_Job
  UNION
  SELECT job_no, labor_time FROM Paint_Job
  UNION
  SELECT job_no, labor_time FROM Cut_Job
) AS S
WHERE S.job_no IN(
  SELECT job_no
  FROM Job
  WHERE
    date_completed = '11/21/2021' AND process_id IN(
      SELECT process_id
      FROM Process
      WHERE department_no = 2
    )
);
```

	(No column name) ▾
1	210

```
-- Query 11
-- DOESN'T QUITE NOT WORK AS INTENDED
-- Does not sort by date, it sorts by process_id
SELECT process_id, department_no
FROM Process
WHERE assembly_id = 65;
```

	process_id	department_no
1	10	2
2	26	2

```

-- Query 12
SELECT job_no, labor_time
FROM
    Fit_Job AS F
WHERE F.job_no IN(
    SELECT job_no
    FROM Job
    WHERE
        date_completed = '11/21/2021' AND process_id IN(
            SELECT process_id
            FROM Process
            WHERE department_no = 2
        )
);

SELECT job_no, labor_time, color, volume
FROM
    Paint_Job AS P
WHERE P.job_no IN(
    SELECT job_no
    FROM Job
    WHERE
        date_completed = '11/21/2021' AND process_id IN(
            SELECT process_id
            FROM Process
            WHERE department_no = 2
        )
);

SELECT job_no, labor_time, machine_type, time_used, material_used
FROM
    Cut_Job AS C
WHERE C.job_no IN(
    SELECT job_no
    FROM Job
    WHERE
        date_completed = '11/21/2021' AND process_id IN(
            SELECT process_id
            FROM Process
            WHERE department_no = 2
        )
);

```

Results

Messages

	job_no	labor_time
1	7	55

	job_no	labor_time	color	volume
1	2	135	color: red	50

	job_no	labor_time	machine_type	time_used	material_used
1	3	20	machine-type	time-used	material used

```
-- Query 13
SELECT *
FROM Customer
WHERE category BETWEEN 4 AND 6;
```

	name ▾	address ▾	category ▾
1	Jackson	489 Stinson St	5
2	Parker	324 NW Pen St	5

```
-- Query 14
```

```
DROP TABLE IF EXISTS temp_cut_jobs;
```

```
SELECT *
```

```
INTO temp_cut_jobs
```

```
FROM Cut_Job;
```

```
DELETE FROM Cut_Job
```

```
WHERE job_no BETWEEN 4 AND 7;
```

Results		Messages			
	job_no	machine_type	time_used	material_used	labor_time
1	3	machine-type	time-used	material used	20
2	4	machine-type	time-used	material used	30

```
DELETE FROM Transaction_Obj
```

```
WHERE job_no IN(
```

```
    SELECT job_no
```

```
    FROM temp_cut_jobs
```

```
    WHERE job_no BETWEEN 4 AND 7
```

```
);
```

Results		Messages				
	transaction_no	sup_cost	assembly_id	process_id	department_no	job_no
1	201	50	NULL	NULL	NULL	1
2	202	35	NULL	NULL	NULL	2
3	203	25	NULL	NULL	NULL	3
4	204	1300	NULL	NULL	1	NULL
5	205	120	NULL	NULL	NULL	7

```
DELETE FROM Job
```

```
WHERE job_no IN(
```

```
    SELECT job_no
```

```
    FROM temp_cut_jobs
```

```
    WHERE job_no BETWEEN 4 AND 7
```

```
);
```

Results Messages				
	job_no ▼	date_commenced ▼	date_completed ▼	process_id ▼
1	1	11/16/2021	11/21/2021	99
2	2	11/17/2021	11/21/2021	10
3	3	11/18/2021	11/21/2021	26
4	4	11/19/2021	NULL	26
5	5	11/20/2021	NULL	10
6	6	11/21/2021	NULL	99
7	7	11/20/2021	11/21/2021	16

```
-- Query 15
UPDATE Paint_Job
SET color = 'color: purple'
WHERE job_no = 2;
```

Results		Messages		
	job_no	color	volume	labor_time
1	2	color: purple	50	135

Java Source Program

***Copy-paste screwed up my nesting. Looks much better in the attached file**

```
import java.sql.Connection;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class Johnson_Steven_IP_Task5b {
    // Database credentials
    final static String HOSTNAME = "john0387-sql-server.database.windows.net";
    final static String DBNAME = "cs-dsa-4513-sql-db";
    final static String USERNAME = "john0387";
    final static String PASSWORD = "hootnanny111.OU";

    // Database connection string
    final static String URL =

String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameIn
Certificate=*.database.windows.net;loginTimeout=30;",
                HOSTNAME, DBNAME, USERNAME, PASSWORD);

    // Query templates for each query 1-15
    // Every statement in each query got its own template
    // Despite how inefficient this method is, everything that I tried to implement works!
    // Note: some queries had to be split into multiple steps hence the notation: a, b, c / i, ii, iii
    final static String QUERY_TEMPLATE_1 =
        "INSERT INTO Customer " +
        "VALUES (?, ?, ?);";

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

    final static String QUERY_TEMPLATE_3 =
        "INSERT INTO Process " +
        "VALUES (?, ?, NULL, ?, NULL);";

    final static String QUERY_TEMPLATE_3a =
        "INSERT INTO Fit_Process " +
        "VALUES (?, ?);";

    final static String QUERY_TEMPLATE_3b =
        "INSERT INTO Paint_Process " +
        "VALUES (?, ?, ?);";

    final static String QUERY_TEMPLATE_3c =
        "INSERT INTO Cut_Process " +
        "VALUES (?, ?, ?);";

    final static String QUERY_TEMPLATE_4a =
        "INSERT INTO Assembly_Obj " +
        "VALUES (?, ?, ?, ?, NULL);";

    final static String QUERY_TEMPLATE_4b =
        "UPDATE Process " +
        "SET assembly_id = ? " +
```

```

        "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_5 =
    "INSERT INTO Account " +
    "VALUES (?);";

final static String QUERY_TEMPLATE_5ai =
    "INSERT INTO Assembly_Acc " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_5aii =
    "UPDATE Assembly_Obj " +
    "SET account_no = ? " +
    "WHERE assembly_id = ?;";

final static String QUERY_TEMPLATE_5bi =
    "INSERT INTO Department_Acc " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_5bii =
    "UPDATE Department " +
    "SET account_no = ? " +
    "WHERE department_no = ?;";

final static String QUERY_TEMPLATE_5ci =
    "INSERT INTO Process_Acc " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_5cii =
    "UPDATE Process " +
    "SET account_no = ? " +
    "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_6a =
    "INSERT INTO Job " +
    "VALUES (?, ?, NULL, ?);";

final static String QUERY_TEMPLATE_6b =
    "UPDATE Process " +
    "SET assembly_id = ? " +
    "WHERE process_id = ?;";

final static String QUERY_TEMPLATE_7 =
    "UPDATE Job " +
    "SET date_completed = ? " +
    "WHERE job_no = ?;";

final static String QUERY_TEMPLATE_7a =
    "INSERT INTO Fit_Job " +
    "VALUES (?, ?);";

final static String QUERY_TEMPLATE_7b =
    "INSERT INTO Paint_Job " +
    "VALUES (?, ?, ?, ?);";

final static String QUERY_TEMPLATE_7c =
    "INSERT INTO Cut_Job " +
    "VALUES (?, ?, ?, ?, ?);";

//placeholder for query 8, DID NOT implement

```

```

final static String QUERY_TEMPLATE_9 =
    "SELECT SUM(ISNULL(S.sup_cost, 0)) " +
    "FROM (" +
        "SELECT sup_cost " +
        "FROM Transaction_Obj " +
        "WHERE " +
            "job_no IN(" +
                "SELECT job_no " +
                "FROM Job " +
                "WHERE process_id IN(" +
                    "SELECT process_id " +
                    "FROM Process " +
                    "WHERE " +
                        "assembly_id = ?)" +
                ")" +
            ")" +
    ") AS S;";

```

```

final static String QUERY_TEMPLATE_10 =
    "SELECT SUM(S.labor_time) " +
    "FROM (" +
        "SELECT job_no, labor_time FROM Fit_Job " +
        "UNION " +
        "SELECT job_no, labor_time FROM Paint_Job " +
        "UNION " +
        "SELECT job_no, labor_time FROM Cut_Job" +
    ") AS S " +
    "WHERE S.job_no IN(" +
        "SELECT job_no " +
        "FROM Job " +
        "WHERE " +
            "date_completed = ? AND process_id IN(" +
                "SELECT process_id " +
                "FROM Process " +
                "WHERE department_no = ?" +
            ")" +
    ");";

```

```

final static String QUERY_TEMPLATE_11 =
    "SELECT process_id, department_no " +
    "FROM Process " +
    "WHERE assembly_id = ?;";

```

//placeholder for query 12, DID NOT implement!

```

final static String QUERY_TEMPLATE_13 =
    "SELECT * " +
    "FROM Customer " +
    "WHERE category BETWEEN ? AND ?;";

```

```

final static String QUERY_TEMPLATE_14a =
    "DROP TABLE IF EXISTS temp_cut_jobs;";

```

```

final static String QUERY_TEMPLATE_14b =
    "SELECT * " +
    "INTO temp_cut_jobs " +
    "FROM Cut_Job;";

```

```

final static String QUERY_TEMPLATE_14c =
    "DELETE FROM Cut_Job " +
    "WHERE job_no BETWEEN ? AND ?;";

```

```

final static String QUERY_TEMPLATE_14d =
    "DELETE " +
    "FROM Transaction_Obj " +
    "WHERE job_no IN(" +
        "SELECT job_no " +
        "FROM temp_cut_jobs" +
        "WHERE job_no BETWEEN ? AND ?" +
    ");";

final static String QUERY_TEMPLATE_14e =
    "DELETE" +
    "FROM Job " +
    "WHERE job_no IN(" +
        "SELECT job_no " +
        "FROM temp_cut_jobs " +
        "WHERE job_no BETWEEN ? AND ?" +
    ");";

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

// User input prompt//
final static String PROMPT =
    "(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
applicable \n" +
    "(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced \n" +
    "(7) At the completion of a job, enter the date it completed and the information relevant to the type of
job \n" +
    "(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts
by adding sup-cost to their current values of details \n" +
    "(9) Retrieve the total cost incurred on an assembly-id \n" +
    "(10) Retrieve the total labor time within a department for jobs completed in the department during a
given date \n" +
    "(11) Retrieve the processes through which a given assembly-id has passed so far (in
date-commenced order) and the department responsible for each process \n" +
    "(12) Retrieve the jobs (together with their type information and assembly-id) completed during a
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 \n" +
    "(17) Export \n" +
    "(18) Quit";

public static void main(String[] args) throws SQLException {
    System.out.println("Welcome to the Job-Shop Accounting Database System");
    final Scanner sc = new Scanner(System.in); // Scanner is used to collect the user input
    String option = ""; // Initialize user option selection as nothing

    while (!option.equals("18")) { // Ask user for options until options 16, 17, or 18 are selected
        System.out.println(PROMPT); // Print the available options
        option = sc.next(); // Read in the user option selection
    }
}

```

```

switch (option) { // Switch between different options
    case "1": // Insert a new Customer
        // Collect the new data from the user
        System.out.println("Please enter customer name:");
        sc.nextLine();
        final String name = sc.nextLine();

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

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

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_1)) {
                // Populate the query template with the data collected from
                // the user
                statement.setString(1, name);
                statement.setString(2, address);
                statement.setInt(3, category);

                System.out.println("Dispatching the query...");
                // Actually execute the populated query
                final int rows_inserted = statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)
                    inserted.\n", rows_inserted));
            }
        }

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

        System.out.println("Please enter department data(64 characters MAX):");
        sc.nextLine();
        final String department_data = sc.nextLine();

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_2)) {
                // Populate the query template with the data collected from
                // the user
                statement.setInt(1, department_no);
                statement.setString(2, department_data);

                System.out.println("Dispatching the query...");
                // Actually execute the populated query
                final int rows_inserted = statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)

```

```

inserted.\n", rows_inserted));
    }
}

break;
case "3": // Insert a new Process given its department, type, and type information
    // Collect the new data from the user
    System.out.println("Please enter process ID:");
    final int process_id = sc.nextInt();

    System.out.println("Please enter process data(64 characters MAX):");
    sc.nextLine();
    final String process_data = sc.nextLine();

    System.out.println("Please enter department number:");
    final int proc_supervisor_dept_no = sc.nextInt();

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            final PreparedStatement statement =
                connection.prepareStatement(QUERY_TEMPLATE_3)) {
            // Populate the query template with the data collected from
            // the user
            statement.setInt(1, process_id);
            statement.setString(2, process_data);
            statement.setInt(3, proc_supervisor_dept_no);
            statement.executeUpdate();
        }
    }

    //Depending on the type of process, ask for different inputs and perform a
    //different query
    System.out.println("Please enter process type(fit, paint, or cut):");
    sc.nextLine();
    final String process_type = sc.nextLine();

    if(process_type.equalsIgnoreCase("fit")) {
        System.out.println("Please enter fit-type(64 characters MAX):");
        final String fit_type = sc.nextLine();

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL))
        {
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_3a)) {
                // Populate the query template with the data
                // collected from the user
                statement.setInt(1, process_id);
                statement.setString(2, fit_type);
                final int rows_inserted =
                    statement.executeUpdate();

                System.out.println(String.format("Done. %d row(s)
                    inserted.\n", rows_inserted));
            }
        }
    }
    else if(process_type.equalsIgnoreCase("paint")) {
        System.out.println("Please enter paint-type(64 characters MAX):");
        final String paint_type = sc.nextLine();
    }
}

```

```

        System.out.println("Please enter paint-method(64 characters MAX):");
        final String paint_method = sc.nextLine();

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL))
        {
            try (
                final PreparedStatement statement =
                    // Populate the query template with the data
                    statement.setInt(1, process_id);
                    statement.setString(2, paint_type);
                    statement.setString(3, paint_method);
                    final int rows_inserted =

                    System.out.println(String.format("Done. %d row(s)
                        inserted.\n", rows_inserted));
            )
            {
                connection.prepareStatement(QUERY_TEMPLATE_3b)) {
                    collected from the user

                    statement.executeUpdate();
                    inserted.\n", rows_inserted));
            }
        }
        else if(process_type.equalsIgnoreCase("cut")) {
            System.out.println("Please enter cutting-type(64 characters MAX):");
            final String cutting_type = sc.nextLine();

            System.out.println("Please enter machine-type(64 characters MAX):");
            final String machine_type = sc.nextLine();

            System.out.println("Connecting to the database...");
            // Get a database connection and prepare a query statement
            try (final Connection connection = DriverManager.getConnection(URL))
            {
                try (
                    final PreparedStatement statement =
                        // Populate the query template with the data
                        statement.setInt(1, process_id);
                        statement.setString(2, cutting_type);
                        statement.setString(3, machine_type);
                        final int rows_inserted =

                        System.out.println(String.format("Done. %d row(s)
                            inserted.\n", rows_inserted));
                )
                {
                    connection.prepareStatement(QUERY_TEMPLATE_3c)) {
                        collected from the user

                        statement.executeUpdate();
                        inserted.\n", rows_inserted));
                    }
                }
            }
            else {
                System.out.println("Error: Invalid type");
                break;
            }
        }
        break;
    case "4": // Insert a new Assembly_Obj given various inputs
        // Collect the new data from the user
        System.out.println("Please enter assembly ID:");
        final int assembly_id = sc.nextInt();

        System.out.println("Please enter assembly details(64 characters MAX):");

```

```

        sc.nextLine();
        final String assembly_details = sc.nextLine();

        System.out.println("Please enter customer name:");
        final String customer_name_on_order = sc.nextLine();

        System.out.println("Please enter the date this assembly was
ordered(MM/DD/YYYY):");

        final String date_ordered = sc.nextLine();

        System.out.println("Please enter the process associated with this assembly:");
        final int parent_process_id = sc.nextInt();

        System.out.println("Connecting to the database...");
        // Get the database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_4a)) {
                // Populate the query template with the data collected from
                the user
                statement.setInt(1, assembly_id);
                statement.setString(2, date_ordered);
                statement.setString(3, assembly_details);
                statement.setString(4, customer_name_on_order);

                System.out.println("Dispatching the first query...");
                final int rows_inserted = statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)
inserted.\n", rows_inserted));
            }
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_4b)) {
                // Populate the query template with the data
                collected from the user
                statement.setInt(1, assembly_id);
                statement.setInt(2, parent_process_id);

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

                final int rows_updated =
                    statement.executeUpdate();

                System.out.println(String.format("Done. %d row(s)
updated.\n", rows_updated));
            }
        }
    }
    break;
    case "5": // Insert a new Account given various inputs
        // Collect the new data from the user
        System.out.println("Please enter the account number:");
        final int account_no = sc.nextInt();

        System.out.println("Connecting to the database...");
        // Get the database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (
                final PreparedStatement statement =
                    connection.prepareStatement(QUERY_TEMPLATE_5)) {

```



```

the user

// Populate the query template with the data collected from
statement.setInt(1, account_no);

System.out.println("Dispatching the query...");
final int rows_inserted = statement.executeUpdate();
System.out.println(String.format("Done. %d row(s)

inserted.\n", rows_inserted));

}

}

System.out.println("What type of account is this?");
sc.nextLine();
final String account_type = sc.nextLine();
if(account_type.equalsIgnoreCase("assembly")) {
    System.out.println("Please enter the associated assembly ID:");
    final int assembly_id_for_account = sc.nextInt();

    System.out.println("Please enter the initial balance:");
    final int details_1 = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get a database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL))
    {
        try (
            final PreparedStatement statement =
                connection.prepareStatement(QUERY_TEMPLATE_5ai)) {
            // Populate the query template with the data
            // collected from the user
            statement.setInt(1, account_no);
            statement.setInt(2, details_1);
            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done. %d row(s)
            inserted.\n", rows_inserted));
        }

        try (
            final PreparedStatement statement =
                connection.prepareStatement(QUERY_TEMPLATE_5aai)) {
            // Populate the query template with the
            // data collected from the user
            statement.setInt(1, account_no);
            statement.setInt(2,
                assembly_id_for_account);
            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done.
            %d row(s) inserted.\n", rows_inserted));
        }
    }
}
else if(account_type.equalsIgnoreCase("department")){
    System.out.println("Please enter the associated department number:");
    final int department_no_for_account = sc.nextInt();

    System.out.println("Please enter the initial balance:");
    final int details_2 = sc.nextInt();

```

```

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL))
        {
            try (
                final PreparedStatement statement =
                    // Populate the query template with the data
                    connection.prepareStatement(QUERY_TEMPLATE_5bi)) {
                statement.setInt(1, account_no);
                statement.setInt(2, details_2);
                final int rows_inserted =
                    statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)
                    inserted.\n", rows_inserted));
            }

            try (
                final PreparedStatement statement =
                    // Populate the query template with the
                    connection.prepareStatement(QUERY_TEMPLATE_5bii)) {
                statement.setInt(1, account_no);
                statement.setInt(2,
                    department_no_for_account);
                final int rows_inserted =
                    statement.executeUpdate();
                System.out.println(String.format("Done.
                    %d row(s) inserted.\n", rows_inserted));
            }
        }
    }
    else if(account_type.equalsIgnoreCase("process")) {
        System.out.println("Please enter the associated process ID:");
        final int process_id_for_account = sc.nextInt();

        System.out.println("Please enter the initial balance:");
        final int details_3 = sc.nextInt();

        System.out.println("Connecting to the database...");
        // Get a database connection and prepare a query statement
        try (final Connection connection = DriverManager.getConnection(URL))
        {
            try (
                final PreparedStatement statement =
                    // Populate the query template with the data
                    connection.prepareStatement(QUERY_TEMPLATE_5ci)) {
                statement.setInt(1, account_no);
                statement.setInt(2, details_3);
                final int rows_inserted =
                    statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)
                    inserted.\n", rows_inserted));
            }

            try (
                final PreparedStatement statement =

```

```

connection.prepareStatement(QUERY_TEMPLATE_5cii) {
    // Populate the query template with the
    data collected from the user
    statement.setInt(1, account_no);
    statement.setInt(2,
    process_id_for_account);
    final int rows_inserted =
    statement.executeUpdate();
    System.out.println(String.format("Done.
    %d row(s) inserted.\n", rows_inserted));
}
}
else {
    System.out.println("Error: Invalid type");
    break;
}

break;
case "6": // Insert a new Job given various inputs
    // Collect the new data from the user
    System.out.println("Please enter the job number.");
    final int job_no = sc.nextInt();

    System.out.println("Please enter the date commenced(MM/DD/YYYY):");
    sc.nextLine();
    final String date_commenced = sc.nextLine();

    System.out.println("Please enter the ID of the parent process.");
    final int proc_id_for_parent_of_job = sc.nextInt();

    System.out.println("Please enter the ID of the assembly that this job is working
on:");

    final int assembly_id_for_job = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get the database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            final PreparedStatement statement =
            connection.prepareStatement(QUERY_TEMPLATE_6a)) {
                // Populate the query template with the data collected from
                the user
                statement.setInt(1, job_no);
                statement.setString(2, date_commenced);
                statement.setInt(3, proc_id_for_parent_of_job);

                System.out.println("Dispatching the query...");
                final int rows_inserted = statement.executeUpdate();
                System.out.println(String.format("Done. %d row(s)
                inserted.\n", rows_inserted));
            }
        try (
            final PreparedStatement statement =
            connection.prepareStatement(QUERY_TEMPLATE_6b)) {
                // Populate the query template with the data collected from

```

the user

```

statement.setInt(1, assembly_id_for_job);
statement.setInt(2, proc_id_for_parent_of_job);

System.out.println("Dispatching the query...");
final int rows_inserted = statement.executeUpdate();
System.out.println(String.format("Done. %d row(s)

updated.\n", rows_inserted));

        }
    }

    break;
case "7":
    // Collect the new data from the user
    System.out.println("Which job was completed?:");
    final int completed_job_no = sc.nextInt();

    System.out.println("Please enter the date the job was

completed(MM/DD/YYYY):");

    sc.nextLine();
    final String completion_date = sc.nextLine();

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

            // Populate the query template with the data collected from

            statement.setString(1, completion_date);
            statement.setInt(2, completed_job_no);

            System.out.println("Dispatching the query...");
            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done. %d row(s)

updated.\n", rows_inserted));

        }
    }

    System.out.println("What type of job wat it?");
    final String job_type = sc.nextLine();
    if(job_type.equalsIgnoreCase("fit")) {
        System.out.println("Please enter the labor time(in minutes):");
        final int labor_time = sc.nextInt();

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

            // Populate the query template with the data

            statement.setInt(1, completed_job_no);

```

```

statement.setInt(2, labor_time);
final int rows_inserted =

statement.executeUpdate();

inserted.\n", rows_inserted));

}

}

else if(job_type.equalsIgnoreCase("paint")){
    System.out.println("Please enter the color of paint used in the job:");
    final String color_used_on_completed = sc.nextLine();

    System.out.println("Please enter the volume of paint used(in

gallons):");

    final int volume = sc.nextInt();

    System.out.println("Please enter the labor time(in minutes):");
    final int labor_time = sc.nextInt();

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

            // Populate the query template with the data

            statement.setInt(1, completed_job_no);
            statement.setString(2, color_used_on_completed);
            statement.setInt(3, volume);
            statement.setInt(4, labor_time);
            final int rows_inserted =

            System.out.println(String.format("Done. %d row(s)

            }

        }

    }

else if(job_type.equalsIgnoreCase("cut")) {
    System.out.println("Please enter the machine type:");
    final String machine_type = sc.nextLine();

    System.out.println("Please enter the time machine was used(in

minutes):");

    final int time_used = sc.nextInt();

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

    System.out.println("Please enter the labor time(in minutes):");
    final int labor_time = sc.nextInt();

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

```

```

connection.prepareStatement(QUERY_TEMPLATE_7c)) {
    // Populate the query template with the data
    statement.setInt(1, completed_job_no);
    statement.setString(2, machine_type);
    statement.setInt(3, time_used);
    statement.setString(4, cutting_type);
    statement.setInt(5, labor_time);
    final int rows_inserted =
        statement.executeUpdate();
    System.out.println(String.format("Done. %d row(s)
    inserted.\n", rows_inserted));
}
}
else {
    System.out.println("Error: Invalid type");
    break;
}
break;
case "8":
    System.out.println("Did not implement query 8. Type 18 to exit. Thank you!");
    break;
case "9":
    System.out.println("Please enter the assembly ID to find the cost of:");
    final int assembly_id_to_sum = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get the database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            final PreparedStatement statement =
                connection.prepareStatement(QUERY_TEMPLATE_9)) {
            // Populate the query template with the data collected from
            // the user
            statement.setInt(1, assembly_id_to_sum);

            System.out.println("Dispatching the query...");
            final ResultSet resultSet = statement.executeQuery();
            System.out.println(String.format("Sum of cost on assembly
            ID: %d", assembly_id_to_sum));
            System.out.println(String.format("%d", resultSet.getInt(1)));
        }
    }
    break;
case "10":
    System.out.println("Please enter the date of completion for the jobs:");
    sc.nextLine();
    final String date_to_check = sc.nextLine();

    System.out.println("Please enter the department to check:");
    final int dept_no_to_check = sc.nextInt();

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

```

```

        connection.prepareStatement(QUERY_TEMPLATE_10)) {
            // Populate the query template with the data collected from
            the user
            statement.setString(1, date_to_check);
            statement.setInt(2, dept_no_to_check);

            System.out.println("Dispatching the query...");
            final ResultSet resultSet = statement.executeQuery();
            System.out.println(String.format("Total labor time for jobs
within department %d that completed on %s", dept_no_to_check, date_to_check));

            System.out.println(String.format("%d", resultSet.getInt(1)));
        }
    }
    break;
case "11":
    System.out.println("Please enter an assembly ID to find which processes and
departments worked on it:");

    final int assembly_id_path = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get the database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            final PreparedStatement statement =
                // Populate the query template with the data collected from
                the user
                statement.setInt(1, assembly_id_path);

                System.out.println("Dispatching the query...");
                final ResultSet resultSet = statement.executeQuery();
                System.out.println(String.format("process_id | corresponding
department_no"));

                while (resultSet.next()) {
                    System.out.println(String.format("%d | %d",
                        resultSet.getInt(1),
                        resultSet.getInt(2)));
                }
            }
        ) {
    }
    break;
case "12":
    System.out.println("Did not implement query 12. Type 18 to exit. Thank you!");
    break;
case "13":
    System.out.println("Please enter the lower bound on category:");
    final int lower = sc.nextInt();

    System.out.println("Please enter the upper bound on category:");
    final int upper = sc.nextInt();

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

```

```

                                try (
connection.prepareStatement(QUERY_TEMPLATE_13)) {
                                // Populate the query template with the data collected from
the user
                                statement.setInt(1, lower);
                                statement.setInt(2, upper);

                                System.out.println("Dispatching the query...");
                                final ResultSet resultSet = statement.executeQuery();
                                System.out.println(String.format("Customers with category
                                >= %d and <= %d", lower, upper));

                                while (resultSet.next()) {
                                    System.out.println(String.format("%s | %s | %d",
                                        resultSet.getString(1),
                                        resultSet.getString(2),
                                        resultSet.getInt(3)));
                                }
                                }

                                break;
case "14":
    System.out.println("Please enter the lower bound on the job number:");
    final int lower_job_no = sc.nextInt();

    System.out.println("Please enter the upper bound on the job number:");
    final int upper_job_no = sc.nextInt();

    System.out.println("Connecting to the database...");
    // Get the database connection, create statement and execute it right away, as no
user input need be collected

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final Statement statement = connection.createStatement()){
            statement.executeUpdate(QUERY_TEMPLATE_14a);
        }
        try (final Statement statement = connection.createStatement()){
            statement.executeUpdate(QUERY_TEMPLATE_14b);
        }
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_14c)) {
            // Populate the query template with the data collected from
the user

            statement.setInt(1, lower_job_no);
            statement.setInt(2, upper_job_no);
            statement.executeUpdate();
        }
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_14d)) {
            // Populate the query template with the data collected from
the user

            statement.setInt(1, lower_job_no);
            statement.setInt(2, upper_job_no);
            statement.executeUpdate();
        }
        try (final PreparedStatement statement =
connection.prepareStatement(QUERY_TEMPLATE_14e)) {
            // Populate the query template with the data collected from
the user

            statement.setInt(1, lower_job_no);

```



```

        statement.setInt(2, upper_job_no);
        statement.executeUpdate();
    }
}

break;
case "15":
    System.out.println("Please enter the job number of the paint job you wish to
edit:");

    final int paint_job_no_to_edit = sc.nextInt();

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

    System.out.println("Connecting to the database...");
    // Get the database connection and prepare a query statement
    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (
            final PreparedStatement statement =
                connection.prepareStatement(QUERY_TEMPLATE_15)) {
            // Populate the query template with the data collected from
            the user

            statement.setString(1, new_color);
            statement.setInt(2, paint_job_no_to_edit);

            System.out.println("Dispatching the query...");
            final int rows_inserted = statement.executeUpdate();
            System.out.println(String.format("Done. %d row(s)
updated.\n", rows_inserted));
        }
    }

    break;
case "16":
    System.out.println("Did not implement the import function. Type 18 to exit. Thank
you!");

    break;
case "17":
    System.out.println("Did not implement the export function. Type 18 to exit. Thank
you!");

    break;
case "18": // Do nothing, the while loop will terminate upon the next iteration
    System.out.println("Exiting! Good-bye!");

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

    break;
}
}

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

```

Screenshots showing testing of queries 1-15

Query 1

```

JOHNSON_REVIEW_123456 Java Application 1: C:\Users\JOHNSON\AppData\Local\Programs\Java\jdk-11.0.2\bin\java.exe (1100 66, 2021, 12-24-21 PM)
Welcome to the Job-Shop Accounting Database System
(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 more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit
1
Please enter customer name:
Parker
Please enter customer address:
324 NW Pen St
Please enter customer category:
5
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(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 more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit
1
Please enter customer name:
Johnson
Please enter customer address:
301 F St
Please enter customer category:
7
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

```

```

johnson_steven_ip_laskob [Java Application] C:\Users\Steven.P\pools\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_04_13.0.2\j2se-13.0.2\bin\javaw.exe (NOV 22, 2021, 1:23:42 PM)
1
Please enter customer name:
Johnson
Please enter customer address:
301 F St
Please enter customer category:
7
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(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 more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit
1
Please enter customer name:
Lee
Please enter customer address:
2395 Yes Ave
Please enter customer category:
10
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(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 more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit

```

Result

Results		Messages		
	name	address	category	
1	Jackson	489 Stinson St	5	
2	Johnson	301 F St	7	
3	Lee	2395 Yes Ave	10	
4	Parker	324 NW Pen St	5	
5	Taylor	331 NW 35th St	3	

Query 2

```

Welcome to the Job-Shop Accounting Database System
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and information
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date
(5) Create a new account and associate it with the process, assembly, or department to
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department
(11) Retrieve the processes through which a given assembly-id has passed so far (in department)
(12) Retrieve the jobs (together with their type information and assembly-id) completed
(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
(17) Export
(18) Quit
2
Please enter department number:
3
Please enter department data(64 characters MAX):
data for department 3
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(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
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date
(5) Create a new account and associate it with the process, assembly, or department to
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department
(11) Retrieve the processes through which a given assembly-id has passed so far (in department)
(12) Retrieve the jobs (together with their type information and assembly-id) completed
(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
(17) Export
(18) Quit
2
Please enter department number:
4
Please enter department data(64 characters MAX):
data for the fourth dept
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(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
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, and date

```

```

johnson_steven@_task00: /java Application/ C:\Users\Steven\AppData\Local\Programs\org.eclipse.jdt
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
2
Please enter department number:
4
Please enter department data(64 characters MAX):
data for the fourth dept
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type ar
(4) Enter a new assembly with its customer-name, assembly-details, asse
(5) Create a new account and associate it with the process, assembly, c
(6) Enter a new job, given its job-no, assembly-id, process-id, and dat
(7) At the completion of a job, enter the date it completed and the inf
(8) Enter a transaction-no and its sup-cost and update all the costs (c
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs complet
(11) Retrieve the processes through which a given assembly-id has passe
(12) Retrieve the jobs (together with their type information and assem
(13) Retrieve the customers (in name order) whose category is in a give
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
2
Please enter department number:
5
Please enter department data(64 characters MAX):
fifth dept data and stuff
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type ar
(4) Enter a new assembly with its customer-name, assembly-details, asse
(5) Create a new account and associate it with the process, assembly, c
(6) Enter a new job, given its job-no, assembly-id, process-id, and dat
(7) At the completion of a job, enter the date it completed and the inf
(8) Enter a transaction-no and its sup-cost and update all the costs (c
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs complet
(11) Retrieve the processes through which a given assembly-id has passe
(12) Retrieve the jobs (together with their type information and assem
(13) Retrieve the customers (in name order) whose category is in a give
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit

```

Result

Results		Messages	
	department_no	department...	account_no
1	1	The first depa...	2
2	2	The second dep...	2
3	3	data for depar...	NULL
4	4	data for the f...	NULL
5	5	fifth dept dat...	NULL

Query 3

```

(18) quit
3
Please enter process ID:
75
Please enter process data(64 characters MAX):
this is process 75
Please enter department number:
3
Please enter process type(fit, paint, or cut):
fit
Please enter fit-type(64 characters MAX):
best-fit
Connecting to the database...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assemb
(5) Create a new account and associate it with the proc
(6) Enter a new job, given its job-no, assembly-id, pro
(7) At the completion of a job, enter the date it compl
(8) Enter a transaction-no and its sup-cost and update
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given assem
(12) Retrieve the jobs (together with their type inform
(13) Retrieve the customers (in name order) whose categ
(14) Delete all cut-jobs whose job-no is in a given ran
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
3
Please enter process ID:
66
Please enter process data(64 characters MAX):
process 66 is cool!
Please enter department number:
5
Please enter process type(fit, paint, or cut):
paint
Please enter paint-type(64 characters MAX):
acrylic
Please enter paint-method(64 characters MAX):
idk any methods
Connecting to the database...
Done. 1 row(s) inserted.

```

```

(18) Quit
3
Please enter process ID:
32
Please enter process data(64 characters MAX):
proc32.0.0
Please enter department number:
4
Please enter process type(fit, paint, or cut):
cut
Please enter cutting-type(64 characters MAX):
cutcut
Please enter machine-type(64 characters MAX):
ME! I'm the machine!
Connecting to the database...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assem
(5) Create a new account and associate it with the pro
(6) Enter a new job, given its job-no, assembly-id, pr
(7) At the completion of a job, enter the date it comp
(8) Enter a transaction-no and its sup-cost and update
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given asse
(12) Retrieve the jobs (together with their type infor
(13) Retrieve the customers (in name order) whose cate
(14) Delete all cut-jobs whose job-no is in a given ra
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
3
Please enter process ID:
44
Please enter process data(64 characters MAX):
rtdiorhd
Please enter department number:
4
Please enter process type(fit, paint, or cut):
paint
Please enter paint-type(64 characters MAX):
water-colors
Please enter paint-method(64 characters MAX):
idk
Connecting to the database...
Done. 1 row(s) inserted.

```



```

(10) Quit
3
Please enter process ID:
86
Please enter process data(64 characters MAX):
sdvtugrtvcs
Please enter department number:
4
Please enter process type(fit, paint, or cut):
fit
Please enter fit-type(64 characters MAX):
good-fit
Connecting to the database...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with :
(4) Enter a new assembly with its customer-name, assembly-de
(5) Create a new account and associate it with the process, i
(6) Enter a new job, given its job-no, assembly-id, process-i
(7) At the completion of a job, enter the date it completed i
(8) Enter a transaction-no and its sup-cost and update all tl
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jo
(11) Retrieve the processes through which a given assembly-id
(12) Retrieve the jobs (together with their type information
(13) Retrieve the customers (in name order) whose category is
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
3
Please enter process ID:
33
Please enter process data(64 characters MAX):
33 bottles of milk on the wall
Please enter department number:
3
Please enter process type(fit, paint, or cut):
fit
Please enter fit-type(64 characters MAX):
333333
Connecting to the database...
Done. 1 row(s) inserted.

```

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department tog
- (4) Enter a new assembly with its customer-name,
- (5) Create a new account and associate it with th
- (6) Enter a new job, given its job-no, assembly-i
- (7) At the completion of a job, enter the date it
- (8) Enter a transaction-no and its sup-cost and u
- (9) Retrieve the total cost incurred on an assembl
- (10) Retrieve the total labor time within a depar
- (11) Retrieve the processes through which a given
- (12) Retrieve the jobs (together with their type
- (13) Retrieve the customers (in name order) whose
- (14) Delete all cut-jobs whose job-no is in a giv
- (15) Change the color of a given paint job
- (16) Import
- (17) Export
- (18) Quit

3

Please enter process ID:

21

Please enter process data(64 characters MAX):

whats 9+10

Please enter department number:

2

Please enter process type(fit, paint, or cut):

fit

Please enter fit-type(64 characters MAX):

21!!!

Connecting to the database...

Done. 1 row(s) inserted.

- (1) Enter a new customer
- (2) Enter a new department
- (3) Enter a new process-id and its department tog

Result

Results		Messages			
	process_id	process_da...	assembly_id	department...	account_no
1	10	are	65	2	3
2	21	whats 9+10	NULL	2	NULL
3	26	cool	65	2	3
4	32	proc32.0.0	NULL	4	NULL
5	33	33 bottles of ...	NULL	3	NULL
6	44	rtdiorhd	NULL	4	NULL
7	66	process 66 is ...	NULL	5	NULL
8	75	this is proces...	NULL	3	NULL
9	86	sdvtugrtvcs	NULL	4	NULL
10	99	processes	100	1	3

Query 4

```

(18) Quit
4
Please enter assembly ID:
200
Please enter assembly details(64 characters MAX):
100 more than 100
Please enter customer name:
Jackson
Please enter the date this assembly was ordered(MM/DD/YYYY):
10/09/2021
Please enter the process associated with this assembly:
21
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

Dispatching the second query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its
(4) Enter a new assembly with its customer-name, assembly-detail
(5) Create a new account and associate it with the process, assembly-id
(6) Enter a new job, given its job-no, assembly-id, process-id,
(7) At the completion of a job, enter the date it completed and
(8) Enter a transaction-no and its sup-cost and update all the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs
(11) Retrieve the processes through which a given assembly-id has
(12) Retrieve the jobs (together with their type information and
(13) Retrieve the customers (in name order) whose category is in
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
4
Please enter assembly ID:
150
Please enter assembly details(64 characters MAX):
50 more than 100
Please enter customer name:
Lee
Please enter the date this assembly was ordered(MM/DD/YYYY):
09/23/2021
Please enter the process associated with this assembly:
66
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

```

```

(17) Export
(18) Quit
4
Please enter assembly ID:
250
Please enter assembly details(64 characters MAX):
50 more than 200
Please enter customer name:
Parker
Please enter the date this assembly was ordered(MM/DD/YYYY):
04/20/2021
Please enter the process associated with this assembly:
86
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

Dispatching the second query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its
(4) Enter a new assembly with its customer-name, assembly-details
(5) Create a new account and associate it with the process, as
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed and
(8) Enter a transaction-no and its sup-cost and update all the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for job
(11) Retrieve the processes through which a given assembly-id
(12) Retrieve the jobs (together with their type information)
(13) Retrieve the customers (in name order) whose category is
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
4
Please enter assembly ID:
300
Please enter assembly details(64 characters MAX):
100 more than 200
Please enter customer name:
Taylor
Please enter the date this assembly was ordered(MM/DD/YYYY):
9/12/2021
Please enter the process associated with this assembly:
99
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

```

```

(18) Quit
4
Please enter assembly ID:
350
Please enter assembly details(64 characters MAX):
50 more than 300
Please enter customer name:
Johnson
Please enter the date this assembly was ordered(MM/DD/YYYY):
8/27/2021
Please enter the process associated with this assembly:
32
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

Dispatching the second query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its
(4) Enter a new assembly with its customer-name, assembly-details
(5) Create a new account and associate it with the process, assembly-id
(6) Enter a new job, given its job-no, assembly-id, process-id,
(7) At the completion of a job, enter the date it completed and
(8) Enter a transaction-no and its sup-cost and update all the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs
(11) Retrieve the processes through which a given assembly-id has
(12) Retrieve the jobs (together with their type information and
(13) Retrieve the customers (in name order) whose category is :
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
4
Please enter assembly ID:
400
Please enter assembly details(64 characters MAX):
100 more than 300
Please enter customer name:
Jackson
Please enter the date this assembly was ordered(MM/DD/YYYY):
07/07/2021

Please enter the process associated with this assembly:
10
Connecting to the database...
Dispatching the first query...

```

```
(16) Import
(17) Export
(18) Quit
4
Please enter assembly ID:
450
Please enter assembly details(64 characters MAX):
200 more than 250
Please enter customer name:
Taylor
Please enter the date this assembly was ordered(MM/DD/YYYY):
11/01/2021
Please enter the process associated with this assembly:
44
Connecting to the database...
Dispatching the first query...
Done. 1 row(s) inserted.

Dispatching the second query...
Done. 1 row(s) updated.

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

Result

Results Messages					
	assembly_id	date_order...	assembly_d...	name	account_no
1	65	11/20/2021	the best assem...	Taylor	1
2	100	11/19/2021	do this and th...	Jackson	1
3	150	09/23/2021	50 more than 1...	Lee	NULL
4	200	10/09/2021	100 more than ...	Jackson	NULL
5	250	04/20/2021	50 more than 2...	Parker	NULL
6	300	9/12/2021	100 more than ...	Taylor	NULL
7	350	8/27/2021	50 more than 3...	Johnson	NULL
8	400	07/07/2021	100 more than ...	Jackson	NULL
9	450	11/01/2021	200 more than ...	Taylor	NULL

Results Messages					
	process_id	process_da...	assembly_id	department...	account_no
1	10	are	400	2	3
2	21	whats 9+10	200	2	NULL
3	26	cool	65	2	3
4	32	proc32.0.0	350	4	NULL
5	33	33 bottles of ...	NULL	3	NULL
6	44	rtdiorhd	450	4	NULL
7	66	process 66 is ...	150	5	NULL
8	75	this is proces...	NULL	3	NULL
9	86	sdvtugrtvcs	250	4	NULL
1...	99	processes	300	1	3

Query 5

```

(18) Quit
5
Please enter the account number:
5
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
department
Please enter the associated department number:
3
Please enter the initial balance:
10000
Connecting to the database...
Done. 1 row(s) inserted.

Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department to
(4) Enter a new assembly with its customer-name,
(5) Create a new account and associate it with tl
(6) Enter a new job, given its job-no, assembly-:
(7) At the completion of a job, enter the date i
(8) Enter a transaction-no and its sup-cost and
(9) Retrieve the total cost incurred on an assem
(10) Retrieve the total labor time within a depa
(11) Retrieve the processes through which a giv
(12) Retrieve the jobs (together with their typ
(13) Retrieve the customers (in name order) whos
(14) Delete all cut-jobs whose job-no is in a gi
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
5
Please enter the account number:
6
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
process
Please enter the associated process ID:
66
Please enter the initial balance:
175
Connecting to the database...

```

```

(18) Quit
5
Please enter the account number:
7
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
assembly
Please enter the associated assembly ID:
150
Please enter the initial balance:
70
Connecting to the database...
Done. 1 row(s) inserted.

Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assembly-id, and process-id
(5) Create a new account and associate it with the process-id
(6) Enter a new job, given its job-no, assembly-id, process-id, and date
(7) At the completion of a job, enter the date it completed
(8) Enter a transaction-no and its sup-cost and update the job
(9) Retrieve the total cost incurred on an assembly-:
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given assembly has been processed
(12) Retrieve the jobs (together with their type info) for a given assembly
(13) Retrieve the customers (in name order) whose cars have been painted
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
5
Please enter the account number:
8
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
assembly
Please enter the associated assembly ID:
350
Please enter the initial balance:
90
Connecting to the database...
Done. 1 row(s) inserted.

```

```

5
Please enter the account number:
9
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
department
Please enter the associated department number:
4
Please enter the initial balance:
20000
Connecting to the database...
Done. 1 row(s) inserted.

Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together w
(4) Enter a new assembly with its customer-name, assembl
(5) Create a new account and associate it with the proce
(6) Enter a new job, given its job-no, assembly-id, proc
(7) At the completion of a job, enter the date it comple
(8) Enter a transaction-no and its sup-cost and update a
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department f
(11) Retrieve the processes through which a given assembl
(12) Retrieve the jobs (together with their type informa
(13) Retrieve the customers (in name order) whose categ
(14) Delete all cut-jobs whose job-no is in a given rang
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
5
Please enter the account number:
10
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

What type of account is this?
process
Please enter the associated process ID:
33
Please enter the initial balance:
165
Connecting to the database...

```

Result

Results Messages	
	account_no ▾
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
1...	10

Results Messages		
	account_no ▾	details_1 ▾
1	1	50
2	7	70
3	8	90

Results Messages		
	account_no ▾	details_2 ▾
1	2	100000
2	5	10000
3	9	20000

Results Messages		
	account_no ▾	details_3 ▾
1	3	150
2	6	175
3	10	165

Query 6

*Note: When creating my tables for task 4 I populated them with some data. I already had a bunch of jobs in there, so I only added enough to get 10 total jobs rather than do 10 iterations of query 10. That's why there are only 3 queries shown below even though we were asked for 10. Query 7 also shares this complication since it also deals with jobs.

```
(17) export
(18) Quit
6
Please enter the job number:
8
Please enter the date commenced(MM/DD/YYYY):
11/21/2021
Please enter the ID of the parent process:
86
Please enter the ID of the assembly that this job is working on:
200
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

Dispatching the query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and
(4) Enter a new assembly with its customer-name, assembly-details, assemb
(5) Create a new account and associate it with the process, assembly, or
(6) Enter a new job, given its job-no, assembly-id, process-id, and date
(7) At the completion of a job, enter the date it completed and the inform
(8) Enter a transaction-no and its sup-cost and update all the costs (det
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed
(11) Retrieve the processes through which a given assembly-id has passed
(12) Retrieve the jobs (together with their type information and assembly
(13) Retrieve the customers (in name order) whose category is in a given
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
6
Please enter the job number:
9
Please enter the date commenced(MM/DD/YYYY):
11/21/2021
Please enter the ID of the parent process:
44
Please enter the ID of the assembly that this job is working on:
350
Connecting to the database...
Dispatching the query...
```

```
(17) Export
(18) Quit
6
Please enter the job number:
10
Please enter the date commenced(MM/DD/YYYY):
11/22/1021
Please enter the ID of the parent process:
33
Please enter the ID of the assembly that this job is working on:
100
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted.

Dispatching the query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type
(4) Enter a new assembly with its customer-name assembly-details as
```

Results

Results Messages				
	job_no	date_comme...	date_compl...	process_id
1	1	11/16/2021	11/21/2021	99
2	2	11/17/2021	11/21/2021	10
3	3	11/18/2021	11/21/2021	26
4	4	11/19/2021	NULL	26
5	5	11/20/2021	NULL	10
6	6	11/21/2021	NULL	99
7	7	11/20/2021	NULL	75
8	8	11/21/2021	NULL	86
9	9	11/21/2021	NULL	44
1...	10	11/22/1021	NULL	33

Results Messages					
	process_id	process_da...	assembly_id	department...	account_no
1	10	are	400	2	3
2	21	whats 9+10	200	2	NULL
3	26	cool	65	2	3
4	32	proc32.0.0	350	4	NULL
5	33	33 bottles of ...	100	3	10
6	44	rtdiorhd	350	4	NULL
7	66	process 66 is ...	150	5	6
8	75	this is proces...	450	3	NULL
9	86	sdvtugrtvcs	200	4	NULL
1...	99	processes	300	1	3

Query 7

```

(18) quit
7
Which job was completed?:
7
Please enter the date the job was completed(MM/DD/YYYY):
11/21/2021
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

What type of job wat it?
fit
Please enter the labor time(in minutes):
50
Connecting to the database...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its name
(4) Enter a new assembly with its customer-name, assembly-details
(5) Create a new account and associate it with the process, assembly-id
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed and its cost
(8) Enter a transaction-no and its sup-cost and update all the jobs
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs
(11) Retrieve the processes through which a given assembly-id
(12) Retrieve the jobs (together with their type information)
(13) Retrieve the customers (in name order) whose category is
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
7
Which job was completed?:
8
Please enter the date the job was completed(MM/DD/YYYY):
11/22/2021
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

What type of job wat it?
paint
Please enter the color of paint used in the job:
blue
Please enter the volume of paint used(in gallons):
3
Please enter the labor time(in minutes):
56
Connecting to the database...
Done. 1 row(s) inserted.

```



```

7
Which job was completed?:
9
Please enter the date the job was completed(MM/DD/YYYY):
1/22/2021
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

What type of job wat it?
cut
Please enter the machine type:
mashingmachine
Please enter the time machine was used(in minutes):
43
Please enter the cutting type:
Please enter the labor time(in minutes):
34
Connecting to the database...
Done. 1 row(s) inserted.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its
(4) Enter a new assembly with its customer-name, assembly-data
(5) Create a new account and associate it with the process, as
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed and
(8) Enter a transaction-no and its sup-cost and update all the
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for job
(11) Retrieve the processes through which a given assembly-id l
(12) Retrieve the jobs (together with their type information an
(13) Retrieve the customers (in name order) whose category is :
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
7
Which job was completed?:
10
Please enter the date the job was completed(MM/DD/YYYY):
11/22/2021
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

What type of job wat it?
fit
Please enter the labor time(in minutes):
45
Connecting to the database...
Done. 1 row(s) inserted.

```

Result

Results Messages				
	job_no	date_commenced	date_completed	process_id
1	1	11/16/2021	11/21/2021	99
2	2	11/17/2021	11/21/2021	10
3	3	11/18/2021	11/21/2021	26
4	4	11/19/2021	NULL	26
5	5	11/20/2021	NULL	10
6	6	11/21/2021	NULL	99
7	7	11/20/2021	11/21/2021	75
8	8	11/21/2021	11/22/2021	86
9	9	11/21/2021	1/22/2021	44
10	10	11/22/1021	11/22/2021	33

Results Messages		
	job_no	labor_time
1	1	55
2	6	45
3	7	50
4	10	45

Results Messages					
	job_no	machine_type	time_used	material_used	labor_time
1	3	machine-type	50	material used	20
2	4	machine-type	40	material used	30
3	9	mashingmachine	43		34

Results Messages				
	job_no	color	volume	labor_time
1	2	color: red	50	135
2	5	color: blue	50	160
3	8	blue	3	56

Query 8

*Note: did not implement this query

Query 9

*Note: this query repeatedly produced empty tables because I never implemented query 8. With no transactions to sum up, there is no way for the query to return a populated table. As seen in Task 4, the logic for the query works if there are transactions present in the database.

```

9
Please enter the assembly ID to find the cost of:
65
Connecting to the database...
Dispatching the query...
Sum of cost on assembly ID: 65
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The result set has no current row.
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDriverError(SQLServerException.java:237)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.verifyResultSetHasCurrentRow(SQLServerResultSet.java:563)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getterGetColumn(SQLServerResultSet.java:2046)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2077)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2063)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getInt(SQLServerResultSet.java:2349)
    at Johnson_Steven_IP_Task5b.main(Johnson_Steven_IP_Task5b.java:709)

```

Query 10

*Note: could not get this query to work. Always returned empty value.

```

Welcome to the Job-Shop Accounting Database System
(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 more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit
10
Please enter the date of completion for the jobs:
11/22/2021
Please enter the department to check:
3
Connecting to the database...
Dispatching the query...
Total labor time for jobs within department 3 that completed on 11/22/2021
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The result set has no current row.
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDriverError(SQLServerException.java:237)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.verifyResultSetHasCurrentRow(SQLServerResultSet.java:563)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getterGetColumn(SQLServerResultSet.java:2046)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2077)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2063)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getInt(SQLServerResultSet.java:2349)
    at Johnson_Steven_IP_Task5b.main(Johnson Steven IP Task5b.java:735)

(17) Export
(18) Quit
10
Please enter the date of completion for the jobs:
11/21/2021
Please enter the department to check:
2
Connecting to the database...
Dispatching the query...
Total labor time for jobs within department 2 that completed on 11/21/2021
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The result set has no current row.
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDriverError(SQLServerException.java:237)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.verifyResultSetHasCurrentRow(SQLServerResultSet.java:563)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getterGetColumn(SQLServerResultSet.java:2046)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2077)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2063)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getInt(SQLServerResultSet.java:2349)
    at Johnson_Steven_IP_Task5b.main(Johnson Steven IP Task5b.java:735)

(17) Export
(18) Quit
10
Please enter the date of completion for the jobs:
11/21/2021
Please enter the department to check:
1
Connecting to the database...
Dispatching the query...
Total labor time for jobs within department 1 that completed on 11/21/2021
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The result set has no current row.
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDriverError(SQLServerException.java:237)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.verifyResultSetHasCurrentRow(SQLServerResultSet.java:563)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getterGetColumn(SQLServerResultSet.java:2046)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2077)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getValue(SQLServerResultSet.java:2063)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerResultSet.getInt(SQLServerResultSet.java:2349)
    at Johnson_Steven_IP_Task5b.main(Johnson Steven IP Task5b.java:735)

```

Query 11

```

11
Please enter an assembly ID to find which processes
350
Connecting to the database...
Dispatching the query...
process_id | corresponding department_no
32 | 4
44 | 4
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department togeth
(4) Enter a new assembly with its customer-name, ass
(5) Create a new account and associate it with the p
(6) Enter a new job, given its job-no, assembly-id,
(7) At the completion of a job, enter the date it co
(8) Enter a transaction-no and its sup-cost and upda
(9) Retrieve the total cost incurred on an assembly-
(10) Retrieve the total labor time within a departme
(11) Retrieve the processes through which a given as
(12) Retrieve the jobs (together with their type inf
(13) Retrieve the customers (in name order) whose ca
(14) Delete all cut-jobs whose job-no is in a given
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
11
Please enter an assembly ID to find which processes
200
Connecting to the database...
Dispatching the query...
process_id | corresponding department_no
21 | 2
86 | 4
(1) Enter a new customer
(18) Quit
11
Please enter an assembly ID to find which pr
65
Connecting to the database...
Dispatching the query...
process_id | corresponding department_no
26 | 2
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its departmen

```

Query 12

*Note: did not implement query 12

Query 13

```

13
Please enter the lower bound on category:
1
Please enter the upper bound on category:
4
Connecting to the database...
Dispatching the query...
Customers with category >= 1 and <= 4
Taylor | 331 NW 35th St | 3
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assembly-id, process-id
(5) Create a new account and associate it with the process-id
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed
(8) Enter a transaction-no and its sup-cost and update-cost
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given assembly has been processed
(12) Retrieve the jobs (together with their type info) for a given assembly-id
(13) Retrieve the customers (in name order) whose category is in the 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
(17) Export
(18) Quit
13
Please enter the lower bound on category:
3
Please enter the upper bound on category:
7
Connecting to the database...
Dispatching the query...
Customers with category >= 3 and <= 7
Jackson | 489 Stinson St | 5
Johnson | 301 F St | 7
Parker | 324 NW Pen St | 5
Taylor | 331 NW 35th St | 3
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assembly-id, process-id
(5) Create a new account and associate it with the process-id
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed
(8) Enter a transaction-no and its sup-cost and update-cost
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given assembly has been processed
(12) Retrieve the jobs (together with their type info) for a given assembly-id
(13) Retrieve the customers (in name order) whose category is in the 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
(17) Export
(18) Quit
13
Please enter the lower bound on category:
6
Please enter the upper bound on category:
10
Connecting to the database...
Dispatching the query...
Customers with category >= 6 and <= 10
Johnson | 301 F St | 7
Lee | 2395 Yes Ave | 10
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together
(4) Enter a new assembly with its customer-name, assembly-id, process-id
(5) Create a new account and associate it with the process-id
(6) Enter a new job, given its job-no, assembly-id, process-id
(7) At the completion of a job, enter the date it completed
(8) Enter a transaction-no and its sup-cost and update-cost
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department
(11) Retrieve the processes through which a given assembly has been processed
(12) Retrieve the jobs (together with their type info) for a given assembly-id
(13) Retrieve the customers (in name order) whose category is in the 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
(17) Export
(18) Quit

```


Query 14

*Note: could not get the query to work.

```

14
Please enter the lower bound on the job number:
6
Please enter the upper bound on the job number:
9
Connecting to the database...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Incorrect syntax near the keyword 'BETWEEN'.
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLSer
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePrep
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExec
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7417)
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLSe
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLSe
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQL
    at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate
    at Johnson_Steven_IP_Task5b.main(Johnson Steven IP_Task5b.java:830)

```

Query 15

```

(10) Quit
15
Please enter the job number of the paint job you wish to ed
2
Please enter the color you wish to change it to:
orange
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with
(4) Enter a new assembly with its customer-name, assembly-d
(5) Create a new account and associate it with the process,
(6) Enter a new job, given its job-no, assembly-id, process
(7) At the completion of a job, enter the date it completed
(8) Enter a transaction-no and its sup-cost and update all
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for
(11) Retrieve the processes through which a given assembly-
(12) Retrieve the jobs (together with their type informatio
(13) Retrieve the customers (in name order) whose category
(14) Delete all cut-jobs whose job-no is in a given range
(15) Change the color of a given paint job
(16) Import
(17) Export
(18) Quit
15
Please enter the job number of the paint job you wish to ed
5
Please enter the color you wish to change it to:
red
Connecting to the database...
(17) Export
(18) Quit
15
Please enter the job number of the paint job you wish to ed
8
Please enter the color you wish to change it to:
black
Connecting to the database...
Dispatching the query...
Done. 1 row(s) updated.

(1) Enter a new customer

```

Result

<div>Results Messages</div>				
	job_no	color	volume	labor_time
1	2	orange	50	135
2	5	red	50	160
3	8	black	3	56

Import

*Note: did not implement

Export

*Note: did not implement

Quit

clipse-workspace - Johnson_Steven_IP_Task5b/src/Johnson_Steven_IP_Task5b.java - Eclipse IDE

Edit Source Refactor Navigate Search Project Run Window Help

```

Problems Javadoc Declaration Console
<terminated> Johnson_Steven_IP_Task5b [Java Application] C:\Users\Steven\AppData\Local\Temp\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (Nov 22, 2021, 6:32:34 PM)
Welcome to the Job-Shop Accounting Database System
(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-detail, assembly-id, and date-ordered and associate it with one or more processes
(5) Create a new account and associate it with the process, assembly, or department to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the affected accounts by adding sup-cost to their current values of details
(9) Retrieve the total cost incurred on an assembly-id
(10) Retrieve the total labor time within a department for jobs completed in the department during a given date
(11) Retrieve the processes through which a given assembly-id has passed so far (in date-commenced order) and the department responsible for each process
(12) Retrieve the 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
(17) Export
(18) Quit
18
Exiting! Good-bye!

```

Error Detection

Category out of bounds

```
(18) Quit
1
Please enter customer name:
Jacobs
Please enter customer address:
909 S Ram Blvd
Please enter customer category:
11
Connecting to the database...
Dispatching the query...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint "CK_Customer_catgo_7D6461A5". The (
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1662)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:615)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:537)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7417)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3488)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:262)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:237)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:483)
at Johnson_Steven_IP_Task5b.main(Johnson_Steven_IP_Task5b.java:257)
```

Insert duplicate primary key

```
(18) Quit
2
Please enter department number:
1
Please enter department data(64 characters MAX):
the first department, but there's a twist
Connecting to the database...
Dispatching the query...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Departme_C2230C91DB28D94C'. Cannot insert duplicate key i
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1662)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:615)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:537)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7417)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3488)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:262)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:237)
at com.microsoft.sqlserver.jdbc@9.4.0.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStatement.java:483)
at Johnson_Steven_IP_Task5b.main(Johnson_Steven_IP_Task5b.java:285)
```

Edit a record that doesn't exist

```
(18) Quit
15
Please enter the job number of the paint job you wish to edit:
11
Please enter the color you wish to change it to:
red
Connecting to the database...
Dispatching the query...
Done. 0 row(s) updated.

(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type
(4) Enter a new assembly with its customer-name, assembly-details, as
(5) Create a new account and associate it with the process, assembly,
(6) Enter a new iob. given its iob-no. assembly-id. process-id. and d
```

DataHandler.java

```

package jsp_azure_test;

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

public class DataHandler {

    private Connection conn;

    // Azure SQL connection credentials
    private String server = "john0387-sql-server.database.windows.net";
    private String database = "cs-dsa-4513-sql-db";
    private String username = "john0387";
    private String password = "hootnanny111.OU";

    // Resulting connection string
    final private String url =

String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameIn
Certificate=*.database.windows.net;loginTimeout=30;",
        server, database, username, password);

    // Initialize and save the database connection
    private void getDBConnection() throws SQLException {
        if (conn != null) {
            return;
        }

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

    // Perform Query 13
    public ResultSet queryThirteen(int lowerBound, int upperBound) throws SQLException {
        getDBConnection();

        final String sqlQuery =
            "SELECT * " +
            "FROM Customer " +
            "WHERE category BETWEEN ? AND ?;";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        stmt.setInt(1, lowerBound);
        stmt.setInt(2, upperBound);

        return stmt.executeQuery();
    }

    // Perform Query 1
    public boolean queryOne(String customerName, String customerAddress, int customerCategory) throws SQLException {
        getDBConnection();

        final String sqlQuery =
            "INSERT INTO Customer " +
            "VALUES (?, ?, ?);";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        stmt.setString(1, customerName);

```



```
stmt.setString(2, customerAddress);  
stmt.setInt(3, customerCategory);  
  
return stmt.executeUpdate() == 1;  
}  
}
```

get_customers

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Movie Nights</title>
  </head>
  <body>
    <%@page import="jsp_azure_test.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
      // We instantiate the data handler here, and get all the movies from the database
      final DataHandler handler = new DataHandler();

      String lowerInputString = request.getParameter("lowerBound");
      String upperInputString = request.getParameter("upperBound");
      int lowerBound = Integer.parseInt(lowerInputString);
      int upperBound = Integer.parseInt(upperInputString);

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

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

```

get_customers_form

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Input for Query 13</title>
  </head>
  <body>
    <h2>Input for query 13</h2>
    <!--
      Form for collecting user input for the new movie_night record.
      Upon form submission, add_movie.jsp file will be invoked.
    -->
    <form action="Johnson_Steven_IP_Task7_get_customers.jsp">
      <!-- The form organized in an HTML table for better clarity. -->
      <table border=1>
        <tr>
          <th colspan="2">Enter the Desired Range:</th>
        </tr>
        <tr>
          <td>Lower Bound</td>
          <td><div style="text-align: center;">
            <input type="text" name="lowerBound">
          </div></td>
        </tr>
        <tr>
          <td>Upper Bound</td>
          <td><div style="text-align: center;">
            <input type="text" name="upperBound">
          </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="GO">
          </div></td>
        </tr>
      </table>
    </form>
  </body>
</html>

```

add_customer

```

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

// Get the attribute values passed from the input form.
String customerName = request.getParameter("name");
String customerAddress = request.getParameter("address");
String customerCategory = request.getParameter("category");

/*
 * If the user hasn't filled out all the time, movie name and duration. This is very simple checking.
 */
if (customerName.equals("") || customerAddress.equals("") || customerCategory.equals("")) {
    response.sendRedirect("Johnson_Steven_IP_Task7_add_customer_form.jsp");
} else {
    int category = Integer.parseInt(customerCategory);

    // Now perform the query with the data from the form.
    boolean success = handler.queryOne(customerName, customerAddress, category);
    if (!success) { // Something went wrong
        %>
        <h2>There was a problem inserting the course</h2>
        <%
    } else { // Confirm success to the user
        %>
        <h2>New Customer</h2>

        <ul>
            <li>Name <%=customerName%></li>
            <li>Address <%=customerAddress%></li>
            <li>Category <%=category%></li>
        </ul>

        <h2>Was successfully inserted.</h2>

        <a href="Johnson_Steven_IP_Task7_get_customers_form.jsp">Call Query 13 here to find customers within a category
range.</a>
        <%
    }
}
%>
</body>
</html>

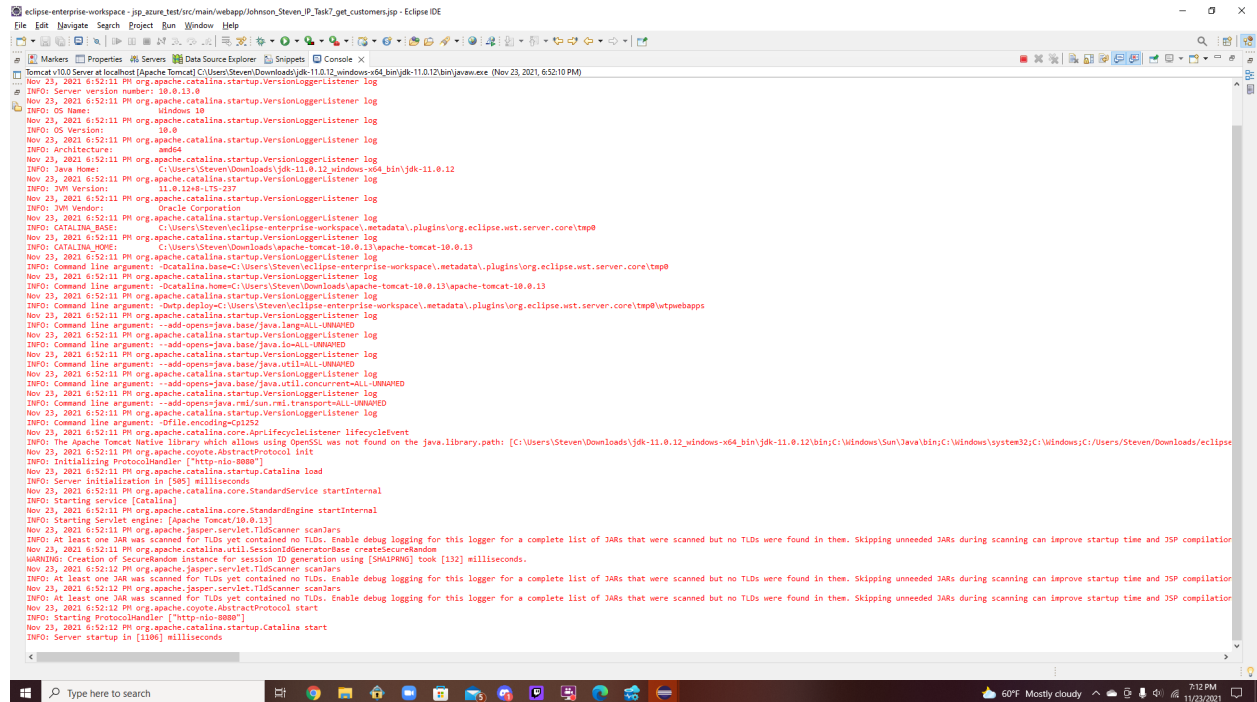
```

add_customer_form

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Add a New Customer</title>
  </head>
  <body>
    <h2>Add a New Customer</h2>
    <!--
      Form for collecting user input for the new movie_night record.
      Upon form submission, add_movie.jsp file will be invoked.
    -->
    <form action="Johnson_Steven_IP_Task7_add_customer.jsp">
      <!-- The form organized in an HTML table for better clarity. -->
      <table border=1>
        <tr>
          <th colspan="2">Enter the Customer Data:</th>
        </tr>
        <tr>
          <td>Customer Name</td>
          <td><div style="text-align: center;">
            <input type="text" name="name">
          </div></td>
        </tr>
        <tr>
          <td>Customer Address</td>
          <td><div style="text-align: center;">
            <input type="text" name="address">
          </div></td>
        </tr>
        <tr>
          <td>Category</td>
          <td><div style="text-align: center;">
            <input type="text" name="category">
          </div></td>
        </tr>
        <tr>
          <td><div style="text-align: center;">
            <input type="reset" value="Clear">
          </div></td>
          <td><div style="text-align: center;">
            <input type="submit" value="GO">
          </div></td>
        </tr>
      </table>
    </form>
  </body>
</html>

```



```
Tomcat v10.0 Server at localhost [Apache Tomcat/C:\Users\Steven\Downloads\jdk-11.0.12_windows-x64_bin\jdk-11.0.12\bin\java.exe (Nov 23, 2021, 6:52:10 PM)
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Server version number: 10.0.13.0
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: OS Name: Windows 10
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: OS Version: 10.0
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Architecture: amd64
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Java Home: C:\Users\Steven\Downloads\jdk-11.0.12_windows-x64_bin\jdk-11.0.12
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: JVM Version: 11.0.12-b15-237
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: JVM Vendor: Oracle Corporation
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: CATALINA_BASE: C:\Users\Steven\workspace\enterprise-workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: CATALINA_HOME: C:\Users\Steven\Downloads\apache-tomcat-10.0.13\apache-tomcat-10.0.13
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dcatalina.base=C:\Users\Steven\workspace\enterprise-workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dcatalina.home=C:\Users\Steven\Downloads\apache-tomcat-10.0.13\apache-tomcat-10.0.13
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dtp.deploy=C:\Users\Steven\workspace\enterprise-workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.lang=ALL-UNNAMED
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.io=ALL-UNNAMED
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.base/java.util=ALL-UNNAMED
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.VersionLoggerListener log
INFO: Command line argument: -Dfile.encoding=cp1252
Nov 23, 2021 6:52:11 PM org.apache.catalina.core.AprLifecycleListener lifecycleEvent
INFO: The Apache Tomcat Native library which allows using OpenSSL, was not found on the java.library.path: [C:\Users\Steven\Downloads\jdk-11.0.12_windows-x64_bin\jdk-11.0.12\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Users\Steven\Downloads\eclipse
Nov 23, 2021 6:52:11 PM org.apache.coyote.AbstractProtocol init
INFO: Initializing ProtocolHandler ["http-nio-8080"]
Nov 23, 2021 6:52:11 PM org.apache.catalina.startup.Catalina load
INFO: Server initialization in [565] milliseconds
Nov 23, 2021 6:52:11 PM org.apache.catalina.core.StandardService startInternal
INFO: Starting service [Catalina]
Nov 23, 2021 6:52:11 PM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet engine: [Apache Tomcat/10.0.13]
Nov 23, 2021 6:52:11 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation
Nov 23, 2021 6:52:11 PM org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
WARNING: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [132] milliseconds.
Nov 23, 2021 6:52:12 PM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation
Nov 23, 2021 6:52:12 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Nov 23, 2021 6:52:12 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [1106] milliseconds
```

A lot of red text, but they weren't errors, it was just feedback from the server.

Input for query 13

Enter the Desired Range:	
Lower Bound	<input type="text" value="1"/>
Upper Bound	<input type="text" value="5"/> ×
<input type="button" value="Clear"/>	<input type="button" value="GO"/>

Customer Name	Customer Address	Category
Bell	244 Ding Ave	4
Jackson	489 Stinson St	5
Parker	324 NW Pen St	5
Taylor	331 NW 35th St	3

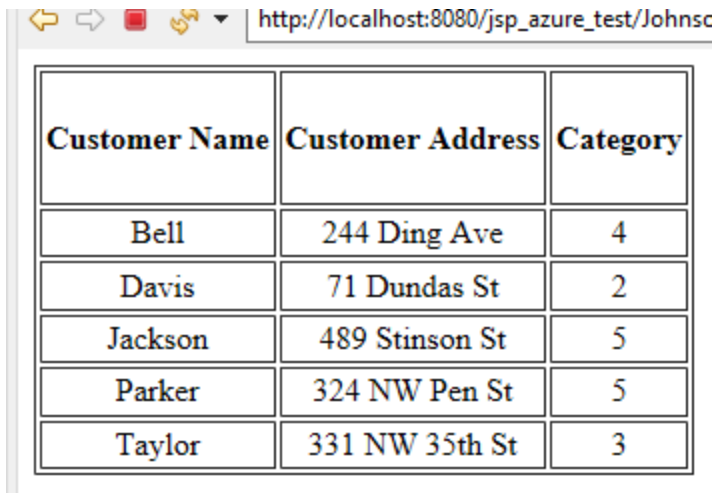
http://localhost:8080/jsp_azure_test/Johnson_Steven_IP_Task

New Customer

- Name Davis
- Address 71 Dundas St
- Category 2

Was successfully inserted.

[Call Query 13 here to find customers within a category range.](#)



A screenshot of a web browser window. The address bar shows the URL `http://localhost:8080/jsp_azure_test/Johnsc`. The browser window displays a table with three columns: **Customer Name**, **Customer Address**, and **Category**. The table contains five rows of data.

Customer Name	Customer Address	Category
Bell	244 Ding Ave	4
Davis	71 Dundas St	2
Jackson	489 Stinson St	5
Parker	324 NW Pen St	5
Taylor	331 NW 35th St	3