

INDIVIDUAL PROJECT

A JOB-SHOP ACCOUNTING DATABASE SYSTEM

Name: Pedakolimi Ramya Sruthi

Course Name: Database Management Systems (DSA4513-995)

Semester: Fall-2023

Instructor: Dr. Le Gruenwald

Email id: sruthipedakolimi@ou.edu

TABLE OF CONTENTS

Task 1: ER Diagram

Task 2: Relational Database Schemas

Task 3:

3.1 Discussion of storage structures for tables

3.2 Discussion of storage structures for tables (Azure SQL)

Task 4: SQL statements and screenshots showing the creation of tables in Azure SQL Database

Task 5:

5.1 SQL statements (and Transact SQL stored procedures)
Implementing all queries (1-15 and error checking)

5.2 The Java source program and screenshots showing its successful compilation

Task 6: Java program Execution

6.1. Screenshots showing the testing of query 1

6.2. Screenshots showing the testing of query 2

6.3. Screenshots showing the testing of query 3

6.4. Screenshots showing the testing of query 4

6.5. Screenshots showing the testing of query 5

6.6. Screenshots showing the testing of query 6

6.7. Screenshots showing the testing of query 7

6.8. Screenshots showing the testing of query 8

6.9. Screenshots showing the testing of query 9

6.10. Screenshots showing the testing of query 10

6.11. Screenshots showing the testing of query 11

6.12. Screenshots showing the testing of query 12

6.13. Screenshots showing the testing of query 13

6.14. Screenshots showing the testing of query 14

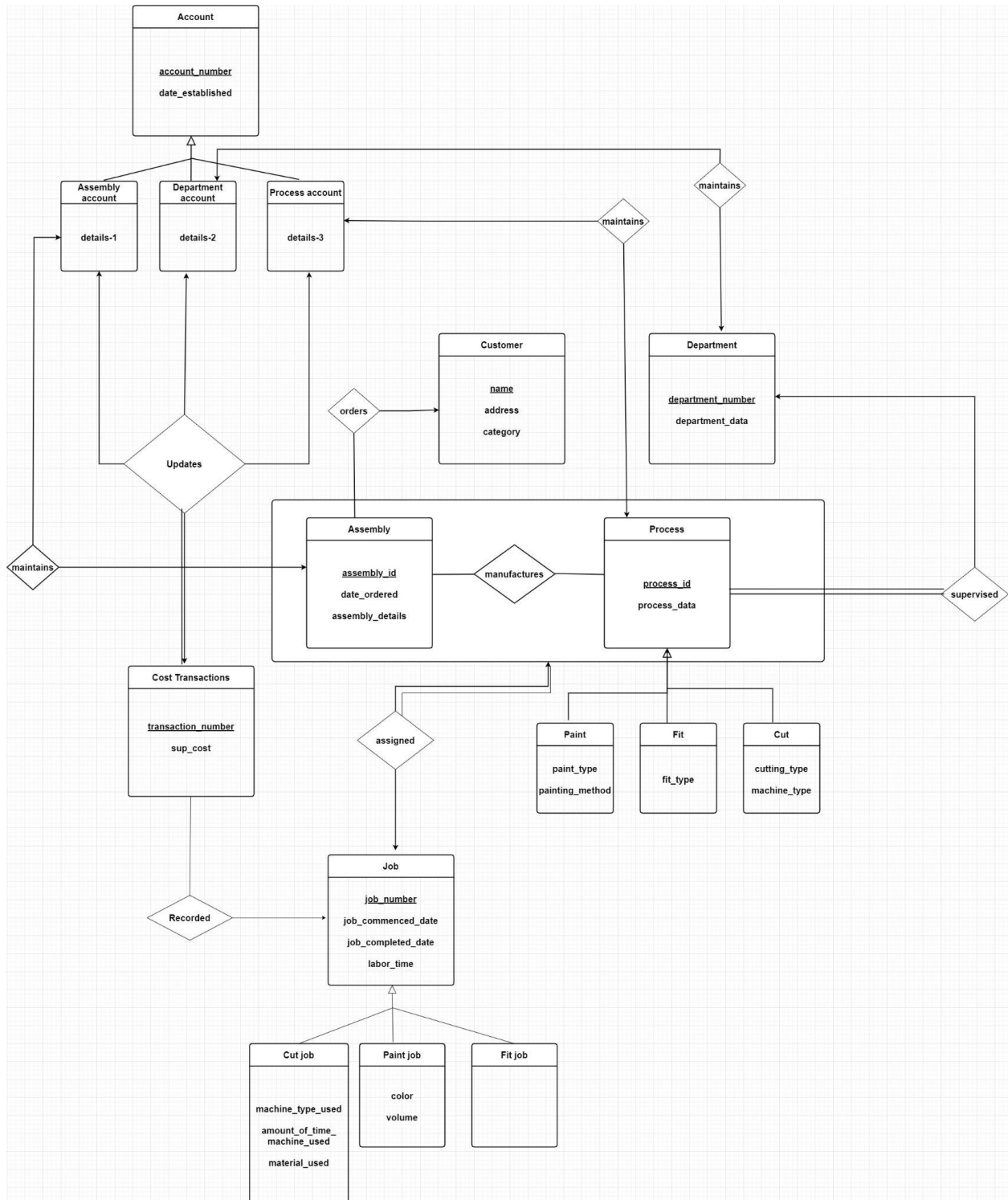
6.15. Screenshots showing the testing import and export

6.16. Screenshots showing the testing of three types of errors

6.17. Screenshots showing the testing of the quit option

Task 7: Web database application and its execution

Task 1: ER Diagram



Task 2: Relational Database Schemas

customers(customer_name, customer_address, category)
 assemblies(assembly_id, date_ordered, assembly_details)
 orders(assembly_id, customer_name)
 department(department_number, department_data)
 process(process_id, process_data)
 paint_process(process_id, paint_type, painting_method)
 fit_process(process_id, fit_type)
 cut_process(process_id, cutting_type, machine_type)
 job(job_number, job_commenced_date, job_completed_date, labour_time)
 cut_job(job_number, machine_type_used, amount_of_time_machine_used,
 material_used)
 paint_job(job_number, color, volume)
 fit_job(job_number)
 supervised(process_id, department_number)
 assigns(assembly_id, process_id, job_number)
 manufactures(assembly_id, process_id)
 cost_transactions(transaction_number, account_number, sup_cost)
 recorded(transaction_number, job_number)
 account(account_number, date_established)
 assembly_account(account_number, cost_details_1)
 department_account(account_number, cost_details_2)
 process_account(account_number, cost_details_3)
 maintains_process(process_id, account_number)
 maintains_assembly(assembly_id, account_number)
 maintains_department(department_number, account_number)

Task 3

3.1: Storage structures for tables

Table Name	Query and Type	Search Key	Query Frequency	Selected File Organisation	Justifications
Customer	1(Insertions)	NA	30/day	B+ Tree	B+ Trees are well-suited for range queries and random searches, which aligns with the queries we are using frequently
Customer	12 (Range search)	Category	100/day		
assembly	4 (Insertion)	NA	40/day	Extendable Hashing with hash key on assembly_id	Since here the random search is taking place on assembly_id and the frequency of insertion and reandim search is more so we use Hashing here.
assembly	5 (Random Search)	assembly_id	10/day		
orders	4 (Insertion)	NA	40/day	Hash	Here this table is used for insertion frequently so hash would be more useful
department	2 (Insertion),	department_number	Infrequent	Extendable Hashing	Here the insertion is not frequent but we are using the random search daily so I planned to use the extendable hasing
department	5 (Random search)		10/day		
Process (paint/fit/cut)	3 (Insertion)		Infrequent	B+ Tree	Here we are having more frequency in noth the random and range search so it would be better if B+ tree is applied
paint/fit/cut Process	5 (random search)	Process_id	10/day		
paint/fit/cut Process	10 (random search)		50/day		

paint/fit/cut Process	11 (range search)	department_number	100/day		
supervised	3 (Insertion)		infrequent	Heap	For this I chose heap as the insertion in this table is infrequent
manufactures	4 (Insertion)		40/day	Hashing on the primary key	Here we are only inserting the data frequently so hashing would be better.
manufactures	6 (Insertion)		50/day		
job	6 (Insertion)		50/day	Hashing on the primary key	Here we are performing insertion and deletion. So hashing would be better
Job	13 (Deletion)		1/month		
assigns	6 (Insertion)		50/day	B+ Tree	Here we both the insertion and the random search is happening more frequently. So B+ Tree
assigns	11 (Random search)	assembly_id	100/day		
Fit job	6 (Insertion)		50/day	Hashing on the primary key	Here we are only inserting the data frequently so hashing would be better
Cut job	6 (Insertion)		50/day	B+ Tree	As here we are both inserting and deleting with range search the B+ Tree would be best
Cut job	13 (Deletion & Range Search)	job_number	1/month		
Paint Job	6 (Insertion)		50/day	B+ Tree	As here we are both inserting and deleting with range search the B+ Tree would be best
Paint job	14 (update & Random search)	job_number	1/week		

cost_transactions	8 (update & random search)	transaction_number	50/day	B+ Tree	Here we are updating the cost transactions more frequently and we also use both the range and random search so B+ Tree would be better choice
cost_transactions	9 (range search)	account_number	200/day		
recorded	9 (range search)		200/day	B + Tree	Here we are using the range search for the recorded more frequently. So B+ tree would be best
account (assembly_account/ department_account/ process_account)	5 (Insertion)		10/day	B+ Tree	Both the insertion and the updation with random search is happening frequently so B+ tree would be best fit
account (assembly_account/ department_account/ process_account)	8 (updation & random search)	account_number	50/day		
maintains _process/assembly/departm ent	5 (Insertion)	NA	10/day	B+ Tree	Here we are Inserting and using the random search . So B+ tree would be best
maintains _process/assembly/departm ent	9 (Random search)	assembly_id	200/day		

3.2 : Storage structures for tables (Azure SQL Database)

While I intend to utilize the storage structures mentioned above for my Azure SQL Database, it's worth noting that Azure automatically generates an index on the primary key, which proves beneficial for query performance. Specifically clustered Index should be used. we are using the range search for the recorded more frequently. So B+ tree would be best Given that most of my search keys align with the primary keys in the table, I plan to leverage the existing indexing provided by Azure. For search keys that aren't primary keys, I've created additional indexes to optimize query retrieval.

Specifically, I've added a secondary index on the "assembly_id" for the "JOB" table. Extending hashing wasn't feasible in this scenario due to the presence of a clustered primary key. Instead, I opted for a sequential index. It's essential to highlight that I'll still rely on Azure's default index for efficient query execution. While this approach may not be the most optimal, it meets my requirements adequately. The decision to create new indexes is primarily aimed at improving the speed of specific queries.

Task 4: SQL statements and screenshots

```

1  -- Drop statements for maintaining referential integrity
2  DROP TABLE IF EXISTS maintains_assembly;
3  DROP TABLE IF EXISTS maintains_department;
4  DROP TABLE IF EXISTS maintains_process;
5  DROP TABLE IF EXISTS recorded;
6  DROP TABLE IF EXISTS cost_transactions;
7  DROP TABLE IF EXISTS manufactures;
8  DROP TABLE IF EXISTS supervised;
9  -- Drop statements for job-related tables
10 DROP TABLE IF EXISTS fit_job;
11 DROP TABLE IF EXISTS paint_job;
12 DROP TABLE IF EXISTS cut_job;
13 -- Drop statements for process-related tables
14 DROP TABLE IF EXISTS fit_process;
15 DROP TABLE IF EXISTS cut_process;
16 DROP TABLE IF EXISTS paint_process;
17 DROP TABLE IF EXISTS assigns;
18 -- Drop statements for assembly-related tables
19 DROP TABLE IF EXISTS assemblies;
20 DROP TABLE IF EXISTS customers;
21 -- Drop statements for department-related tables
22 DROP TABLE IF EXISTS departments;
23 DROP TABLE IF EXISTS department_account;
24 -- Drop statements for account-related tables
25 DROP TABLE IF EXISTS process_account;
26 DROP TABLE IF EXISTS assembly_account;
27 -- Drop statements for general tables
28 DROP TABLE IF EXISTS orders;
29 DROP TABLE IF EXISTS process;
30 DROP TABLE IF EXISTS account;
31 DROP TABLE IF EXISTS job;
32
33
34 -- Creating customer table
35 CREATE TABLE customers (
36     customer_name VARCHAR(250),
37     customer_address VARCHAR(250),
38     category INT NOT NULL CHECK (category BETWEEN 1 AND 10)
39     PRIMARY KEY (customer_name)
40 );
41
42 -- Creating assembly table
43 CREATE TABLE assemblies (
44     assembly_id INT,
45     date_ordered DATE,
46     assembly_details VARCHAR(250),
47     PRIMARY KEY (assembly_id),
48 );
49
50 -- Create orders table
51 CREATE TABLE orders (
52     assembly_id INT,
53     customer_name VARCHAR(250),
54     PRIMARY KEY (assembly_id),
55     FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
56     FOREIGN KEY (customer_name) REFERENCES customers(customer_name)
57 );
58

```

```

58
59 --Create departments table
60 CREATE TABLE departments (
61     department_number INT,
62     department_data VARCHAR(2500),
63     PRIMARY KEY (department_number),
64 );
65
66 -- Create process table
67 CREATE TABLE process (
68     process_id INT,
69     process_data VARCHAR(250),
70     PRIMARY KEY (process_id),
71 );
72
73 -- Create paint process
74 CREATE TABLE paint_process (
75     process_id INT,
76     paint_type VARCHAR(250),
77     painting_method VARCHAR(250),
78     PRIMARY KEY (process_id),
79     FOREIGN KEY (process_id) REFERENCES process(process_id)
80 );
81
82 -- Create fit process
83 CREATE TABLE fit_process (
84     process_id INT,
85     fit_type VARCHAR(250),
86     PRIMARY KEY (process_id),
87     FOREIGN KEY (process_id) REFERENCES process(process_id)
88 );
89
90 -- Create cut process
91 CREATE TABLE cut_process (
92     process_id INT,
93     cutting_type VARCHAR(250),
94     machine_type VARCHAR(250),
95     PRIMARY KEY (process_id),
96     FOREIGN KEY (process_id) REFERENCES process(process_id)
97 );
98 );
99
100 CREATE TABLE supervised (
101     process_id INT,
102     department_number INT,
103     PRIMARY KEY (process_id),
104     FOREIGN KEY (process_id) REFERENCES process(process_id),
105     FOREIGN KEY (department_number) REFERENCES departments(department_number)
106 )
107
108 CREATE INDEX Supervised_DepartmentName ON Supervised (department_number);
109
110 CREATE TABLE job (
111     job_number INT,
112     job_commenced_date DATE,
113     job_completed_date DATE,
114     labor_time DECIMAL(10,2),
115     PRIMARY KEY (job_number)
116 )
117
118 CREATE INDEX Job_Completed_Date_Index ON job (job_completed_date);
119 CREATE INDEX JOB_JOB_NUMBER_INDEX ON job(job_number)

```

```

119
120 CREATE TABLE cut_job (
121     job_number INT,
122     machine_type_used VARCHAR(250),
123     amount_of_time_machine_used DECIMAL(10,2),
124     material_used VARCHAR(250),
125     PRIMARY KEY (job_number),
126     FOREIGN KEY (job_number) REFERENCES job(job_number),
127 );
128 CREATE INDEX CUT_JOB_NUMBER_INDEX ON cut_job(job_number)
129
130 CREATE TABLE paint_job (
131     job_number INT,
132     color VARCHAR(250),
133     volume DECIMAL(10,2),
134     PRIMARY KEY (job_number),
135     FOREIGN KEY (job_number) REFERENCES job(job_number)
136 );
137 CREATE INDEX PAINT_JOB_NUMBER_INDEX ON paint_job(job_number)
138
139 CREATE TABLE fit_job (
140     job_number INT,
141     PRIMARY KEY (job_number),
142     FOREIGN KEY (job_number) REFERENCES job(job_number)
143 );
144
145 CREATE TABLE assigns(
146     assembly_id INT,
147     process_id INT,
148     job_number INT,
149     PRIMARY KEY (assembly_id, process_id),
150     FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
151     FOREIGN KEY (process_id) REFERENCES process(process_id),
152     FOREIGN KEY (job_number) REFERENCES job(job_number),
153 );
154 CREATE INDEX Assigns_Assembly_ID ON assigns (assembly_id);
155
156 CREATE TABLE manufactures (
157     assembly_id INT,
158     process_id INT,
159     PRIMARY KEY (assembly_id, process_id),
160     FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
161     FOREIGN KEY (process_id) REFERENCES process(process_id),
162 );
163
164 CREATE TABLE account(
165     account_number INT,
166     date_established DATE,
167     PRIMARY KEY (account_number),
168 )
169
170 CREATE TABLE assembly_account (
171     account_number INT,
172     cost_details_1 VARCHAR(250),
173     PRIMARY KEY (account_number),
174     FOREIGN KEY (account_number) REFERENCES account(account_number)
175 );
176 CREATE INDEX Assembly_Account_Acc_Number_Index ON assembly_account(account_number)
177

```

```

178 CREATE TABLE department_account (
179     account_number INT,
180     cost_details_2 VARCHAR(250),
181     PRIMARY KEY (account_number),
182     FOREIGN KEY (account_number) REFERENCES account(account_number)
183 );
184
185 CREATE INDEX Department_Account_Acc_Number_Index ON department_account(account_number)
186
187 CREATE TABLE process_account (
188     account_number INT,
189     cost_details_3 VARCHAR(250),
190     PRIMARY KEY (account_number),
191     FOREIGN KEY (account_number) REFERENCES account(account_number)
192 );
193
194 --CREATE INDEX Process_Account_Acc_Number_Index ON process_account(account_number)
195
196 CREATE TABLE cost_transactions (
197     transaction_number INT,
198     account_number INT,
199     sup_cost DECIMAL(10,2),
200     PRIMARY KEY (transaction_number),
201     FOREIGN KEY (account_number) REFERENCES account(account_number)
202 );
203
204 CREATE TABLE recorded (
205     transaction_number INT,
206     job_number INT,
207     PRIMARY KEY (transaction_number),
208     FOREIGN KEY (transaction_number) REFERENCES cost_transactions(transaction_number),
209     FOREIGN KEY (job_number) REFERENCES job(job_number)
210 );
211
212 CREATE TABLE maintains_process (
213     process_id INT,
214     account_number INT,
215     PRIMARY KEY (process_id),
216     FOREIGN KEY (account_number) REFERENCES account(account_number),
217     FOREIGN KEY (process_id) REFERENCES process(process_id)
218 );
219
220 CREATE TABLE maintains_assembly (
221     assembly_id INT,
222     account_number INT,
223     PRIMARY KEY (assembly_id),
224     FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
225     FOREIGN KEY (account_number) REFERENCES account(account_number),
226 );
227
228 CREATE INDEX Maintains_Assembly_AssemblyID_INDEX ON maintains_assembly(assembly_id)
229
230 CREATE TABLE maintains_department (
231     department_number INT,
232     account_number INT,
233     PRIMARY KEY (department_number),
234     FOREIGN KEY (department_number) REFERENCES departments(department_number),
235     FOREIGN KEY (account_number) REFERENCES account(account_number),
236 );
237
238

```

Messages

2:35:45 PM Started executing query at line 1
 Commands completed successfully.
 Total execution time: 00:00:00.386

Task 5

5.1: SQL Statements (Stored Procedures) and Error Handling

SQL Statements (Stored Procedures)

```

1  DROP PROCEDURE IF EXISTS addCustomerDetails
2  DROP PROCEDURE IF EXISTS addDepartmentDetails
3  DROP PROCEDURE IF EXISTS addNewProcess
4  DROP PROCEDURE IF EXISTS addNewAssembly
5  DROP PROCEDURE IF EXISTS addAccountDetailsAndProcess
6  DROP PROCEDURE IF EXISTS addJobDetails
7  DROP PROCEDURE IF EXISTS ChangeColorPaintJob;
8  DROP PROCEDURE IF EXISTS deleteCutJobs;
9  DROP PROCEDURE IF EXISTS getCustomersForCategories;
10 DROP PROCEDURE IF EXISTS getProcessDepartmentDetails;
11 DROP PROCEDURE IF EXISTS getTotalCostIncurred;
12 DROP PROCEDURE IF EXISTS updateCostDetails;
13 DROP PROCEDURE IF EXISTS GetTotalLaborTimeInDepartment
14 DROP PROCEDURE IF EXISTS updateJobDetails;
15 -----
16 -- 1. Enter a new customer (30/day).
17 -----
18 GO
19 CREATE PROCEDURE addCustomerDetails
20     @customer_name VARCHAR(250),
21     @customer_address VARCHAR(250),
22     @category INT
23 AS
24 BEGIN
25     -- Check if the customer already exists
26
27     INSERT INTO customers(customer_name, customer_address, category)
28     VALUES (@customer_name, @customer_address, @category);
29
30 END
31 -----
32 -- 2. Enter a new department (infrequent).
33 -----
34 GO
35 CREATE PROCEDURE addDepartmentDetails
36     @department_number INT,
37     @department_data VARCHAR(255)
38 AS
39 BEGIN
40     -- Insert the new department without checking for duplicates
41     INSERT INTO departments (department_number, department_data)
42     VALUES (@department_number, @department_data);

```

```

44 -----
45 -- 3. Enter a new process-id and its department together with its type and information relevant to the type (i
46 -----
47 GO
48 CREATE PROCEDURE addNewProcess
49     @process_id INT,
50     @process_data VARCHAR(250),
51     @paint_type VARCHAR(250),
52     @painting_method VARCHAR(250),
53     @fit_type VARCHAR(250),
54     @cutting_type VARCHAR(250),
55     @machine_type VARCHAR(250),
56     @department_number INT,
57     @process_type VARCHAR(250)
58 AS
59 BEGIN
60     -- Insert into process and supervised tables
61     INSERT INTO process (process_id, process_data) VALUES (@process_id, @process_data);
62     INSERT INTO supervised (process_id, department_number ) VALUES (@process_id, @department_number)
63     -- Inserting into respective process types based on the user selection
64     IF @process_type = 'fit'
65     BEGIN
66         INSERT INTO fit_process (process_id, fit_type)
67         VALUES (@process_id, @fit_type);
68     END
69     ELSE IF @process_type = 'paint'
70     BEGIN
71         INSERT INTO paint_process (process_id, paint_type, painting_method)
72         VALUES (@process_id, @paint_type, @painting_method);
73     END
74     ELSE IF @process_type = 'cut'
75     BEGIN
76         INSERT INTO cut_process (process_id, cutting_type, machine_type)
77         VALUES (@process_id, @cutting_type, @machine_type);
78     END
79 END;
80

```

```

83 -- 4 Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered
84 -- and associate it with one or more processes (40/day)
85 -----
86 GO
87 CREATE PROCEDURE addNewAssembly(
88     @assembly_id INT,
89     @date_ordered VARCHAR(30),
90     @assembly_details VARCHAR(250),
91     @customer_name VARCHAR(250),
92     @process_id VARCHAR(250)
93 )
94 AS
95 BEGIN
96     -- Insert in assembly, order and manufactures
97     INSERT INTO assemblies (assembly_id, date_ordered, assembly_details)
98     VALUES (@assembly_id, @date_ordered, @assembly_details);
99
100     INSERT INTO orders (assembly_id, customer_name)
101     VALUES (@assembly_id, @customer_name);
102
103     INSERT INTO manufactures (assembly_id, process_id)
104     VALUES (@assembly_id, @process_id)
105
106 END
107

```

```

109 -----
110 -- 5. Create a new account and associate it with the process, assembly, or department to which it is applicab
111 -----
112 GO
113 -- Create the procedure
114 CREATE PROCEDURE addAccountDetailsAndProcess
115     @account_type VARCHAR(50),
116     @account_number INT,
117     @cost_details DECIMAL(10, 2),
118     @date_established VARCHAR(30),
119     @process_id INT,
120     @assembly_id INT,
121     @department_number INT
122 AS
123 BEGIN
124
125     INSERT INTO account (account_number, date_established)
126     VALUES (@account_number, @date_established)
127     -- Create the account for the account type the user inputs.
128     IF @account_type = 'process'
129     BEGIN
130         INSERT INTO process_account (account_number, cost_details_3)
131         VALUES (@account_number, @cost_details);
132         INSERT INTO maintains_process (process_id, account_number)
133         VALUES (@process_id, @account_number);
134     END
135     ELSE IF @account_type = 'assembly'
136     BEGIN
137         INSERT INTO assembly_account (account_number, cost_details_1)
138         VALUES (@account_number, @cost_details);
139
140         INSERT INTO maintains_assembly (assembly_id, account_number)
141         VALUES (@assembly_id, @account_number);
142     END
143     ELSE IF @account_type = 'department'
144     BEGIN
145         INSERT INTO department_account (account_number, cost_details_2)
146         VALUES (@account_number, @cost_details);
147
148         INSERT INTO maintains_department (department_number, account_number)
149         VALUES (@department_number, @account_number);
150     END
151     ELSE
152     BEGIN
153         -- Handle invalid account type
154         SELECT 'Invalid account type. Supported types are process_account, assembly_account, and department_account.' AS Message;
155         RETURN;
156     END
157 END;
158 GO
159
160 -----
161 -- 6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
162 -----
163 GO
164 CREATE PROCEDURE addJobDetails
165     @job_number INT,
166     @assembly_id INT,
167     @process_id INT,
168     @job_commenced_date VARCHAR(30),
169     @job_type VARCHAR(20),
170     @machine_type_used VARCHAR(250),
171     @material_used VARCHAR(250),
172     @color VARCHAR(250),
173     @volume VARCHAR(250)

```

```

174 AS
175 BEGIN
176
177     INSERT INTO job(job_number, job_commenced_date, job_completed_date, labor_time) VALUES (@job_number, @job_commenced_date, null, null)
178     INSERT INTO assigns (assembly_id, process_id, job_number) VALUES (@assembly_id, @process_id, @job_number);
179     IF @job_type = 'cut'
180     BEGIN
181         INSERT INTO cut_job (job_number, machine_type_used, amount_of_time_machine_used, material_used)
182         VALUES (@job_number, @machine_type_used, null, @material_used);
183     END
184     ELSE IF @job_type = 'paint'
185     BEGIN
186         INSERT INTO paint_job (job_number, color, volume)
187         VALUES (@job_number, @color, @volume);
188     END
189     ELSE IF @job_type = 'fit'
190     BEGIN
191         INSERT INTO fit_job (job_number)
192         VALUES (@job_number);
193     END
194
195 END;
196
197 END;
198
199 -----
200 -- 7. At the completion of a job, enter the date it completed and the information relevant to the type of job (50/day).
201 -----
202 GO
203 CREATE PROCEDURE updateJobDetails
204     @job_number INT,
205     @job_completed_date VARCHAR(15),
206     @amount_of_time_machine_used DECIMAL(10,2),
207     @labor_time DECIMAL(10,2)
208

```

Notebooks (Ctrl+Shift+B)

```

210 BEGIN
211     DECLARE @job_type VARCHAR
212     UPDATE Job
213     SET job_completed_date = @job_completed_date,
214     labor_time = @labor_time
215     WHERE job_number= @job_number;
216
217     SELECT @job_type =
218     CASE
219         WHEN EXISTS (SELECT 1 FROM fit_job WHERE job_number= @job_number) THEN 'fit'
220         WHEN EXISTS (SELECT 1 FROM cut_job WHERE job_number= @job_number) THEN 'cut'
221         WHEN EXISTS (SELECT 1 FROM paint_job WHERE job_number= @job_number) THEN 'paint'
222         ELSE NULL
223     END;
224
225     IF @job_type = 'cut'
226     BEGIN
227
228         Update Cut_Job SET
229         amount_of_time_machine_used = @amount_of_time_machine_used
230         WHERE job_number = @job_number;
231     END
232 END;
233
234 -----
235 -- 8. Enter a transaction-no and its sup-cost and update all the costs
236 --(details) of the affected accounts by adding sup-cost to their current values of details (50/day)
237 -----
238 GO
239 CREATE PROCEDURE updateCostDetails
240     @transaction_number INT,
241     @sup_cost DECIMAL,
242     @account_number INT

```



```

242
243 AS
244 BEGIN
245     INSERT INTO cost_transactions(transaction_number, sup_cost, account_number)
246     VALUES (@transaction_number,@sup_cost, @account_number)
247
248     -- Update assembly account details
249     IF EXISTS (SELECT 1 FROM assembly_account WHERE account_number = @account_number)
250     BEGIN
251         UPDATE assembly_account
252         SET cost_details_1 = cost_details_1 + @sup_cost
253         WHERE account_number = @account_number;
254     END
255
256     -- Update department account details
257     ELSE IF EXISTS (SELECT 1 FROM department_account WHERE account_number = @account_number)
258     BEGIN
259         UPDATE department_account
260         SET cost_details_2 = cost_details_2 + @sup_cost
261         WHERE account_number = @account_number;
262     END
263     ELSE IF EXISTS (SELECT 1 FROM process_account WHERE account_number = @account_number)
264     BEGIN
265         UPDATE process_account
266         SET cost_details_3 = cost_details_3 + @sup_cost
267         WHERE account_number = @account_number;
268     END
269
270 END;
271

```

```

272
273 -- 9. Retrieve the total cost incurred on an assembly-id (200/day)
274 -----
275
276 GO
277 CREATE PROCEDURE getTotalCostIncurred
278     @assembly_id INT,
279     @total_cost DECIMAL(10, 2) OUTPUT
280 AS
281 BEGIN
282     SELECT @total_cost = SUM(ct.sup_cost)
283     FROM cost_transactions ct
284     JOIN maintains_assembly ma ON ct.account_number = ma.account_number
285     WHERE ma.assembly_id = @assembly_id;
286 END
287 PRINT 'total_sup_code: ' + CAST(@total_cost AS VARCHAR);
288 GO
289 -----
290 -- 10. Retrieve the total labor time within a department for jobs completed in the department during a given
291 -----
292 GO
293 CREATE PROCEDURE GetTotalLaborTimeInDepartment
294     @department_number INT,
295     @job_completed_date VARCHAR(30),
296     @total_labor_time DECIMAL(10,2) OUTPUT
297 AS
298 BEGIN
299     -- Calculate total labor time for jobs completed on the given date
300     SELECT @total_labor_time = ISNULL(SUM(j.labor_time), 0)
301     FROM job AS j
302     WHERE j.job_number IN (
303         SELECT job_number
304         FROM assigns
305         WHERE process_id IN (
306             SELECT process_id
307             FROM supervised
308             WHERE department_number = @department_number
309         )
310     )
311

```

```

309
310 Notebooks (Ctrl+Shift+B) job_completed_date = @job_completed_date;
311     SELECT @total_labor_time AS TotalLaborTime;
312 END;
313
314 -----
315 -- 11. Retrieve the processes through which a given assembly-id has passed so far (in date-
316 -- commenced order) and the department responsible for each process
317 -----
318
319 GO
320 CREATE PROCEDURE getProcessDepartmentDetails
321     @assembly_id INT
322 AS
323 BEGIN
324     SELECT s.process_id, s.department_number FROM supervised as s where
325     s.process_id in (SELECT a.process_id FROM assigns as a where a.assembly_id = @assembly_id)
326     ORDER BY (SELECT job_commenced_date
327     FROM job AS j
328     WHERE j.job_number = (SELECT a.job_number FROM assigns AS a WHERE a.assembly_id = @assembly_id))
329
330 END
331 GO
332 -----
333 -- 12. Retrieve the customers (in name order) whose category is in a given range (100/day)
334 -----
335
336 GO
337 CREATE PROCEDURE getCustomersForCategories
338     @range_start INT,
339     @range_end INT
340 AS
341 BEGIN
342     SELECT * FROM customers
343     WHERE category BETWEEN @range_start AND @range_end ORDER BY customer_name;
344 END
345 GO
346
347

```

```

347
348 Notebooks (Ctrl+Shift+B) all cut-jobs whose job-no is in a given range (1/month).
349 -----
350
351 GO
352 CREATE PROCEDURE deleteCutJobs
353     @job_number_start INT,
354     @job_number_end INT
355 AS
356 BEGIN
357     DELETE FROM cut_job
358     WHERE job_number BETWEEN @job_number_start AND @job_number_end;
359
360     DELETE FROM job
361     WHERE job_number BETWEEN @job_number_start AND @job_number_end;
362 END
363 GO
364

```

```

365 -----
366 -- 14.Change the color of a given paint job (1/week).
367 -----
368
369 GO
370 CREATE PROCEDURE ChangeColorPaintJob
371     @paint_job_number INT,
372     @color VARCHAR(50)
373 AS
374 BEGIN
375     UPDATE paint_job
376     SET color = @color
377     WHERE job_number = @paint_job_number;
378 END;

```

Messages

```

5:02:26 PM      Started executing query at line 1
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 19
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 34
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 48
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 87
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 113
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 159
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 164
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 203
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 238
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 277
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 289
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 293

```

```

                  Commands completed successfully.
5:02:26 PM      Started executing query at line 320
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 332
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 337
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 346
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 352
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 364
                  Commands completed successfully.
5:02:26 PM      Started executing query at line 370
                  Commands completed successfully.
Total execution time: 00:00:01.442

```

Error Handling

1. Primary Key Constraint

```
1 insert into departments(department_number, department_data) Values (1006, 'Data Science and Analytics')
```

Messages

```
7:18:09 PM Started executing query at Line 1
Msg 2627, Level 14, State 1, Line 1
Violation of PRIMARY KEY constraint 'PK__departme__9265CB0998BEC5BF'. Cannot insert duplicate key in object 'dbo.departments'. The duplicate key value is (1006).
The statement has been terminated.
Total execution time: 00:00:00.058
```

2. Category Constraint Error Handling

Run Cancel Disconnect Change Database: cs-dsa-4513-sql-db Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

```
2 insert into customers(customer_name, customer_address, category) values ('Dhana', 'Beaumont Drive', 16)
```

Messages

```
7:03:28 PM Started executing query at Line 1
Msg 547, Level 16, State 0, Line 1
The INSERT statement conflicted with the CHECK constraint "CK__customers__categ__110B679F". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.customers", column 'category'.
The statement has been terminated.
Total execution time: 00:00:00.070
```

3. Foreign Key Constraint

```
1 insert into supervised(process_id, department_number) Values (7002, 1006)
2
```

Messages

```
7:24:35 PM Started executing query at Line 1
Msg 547, Level 16, State 0, Line 1
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__supervise__proce__26068485". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.process", column 'process_id'.
The statement has been terminated.
Total execution time: 00:00:00.080
```

5.2 Java Source Program

```

1
2=import java.sql.Connection;
3 import java.util.Scanner;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.sql.Types;
7 import java.sql.DriverManager;
8 import java.sql.PreparedStatement;
9 import java.time.format.DateTimeFormatter;
10 import java.io.*;
11 import java.math.BigDecimal;
12 import java.sql.CallableStatement;
13
14 public class Pedakolimi_Ramya_Sruthi_IP_Task5b {
15     // Database credentials
16     final static String HOSTNAME = "peda0001-sql-server.database.windows.net";
17     final static String DBNAME = "cs-dsa-4513-sql-db";
18     final static String USERNAME = "peda0001";
19     final static String PASSWORD = "Laddu@2023";
20     // Database connection string
21     final static String URL = String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;"
22         + "password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;"
23         + "loginTimeout=30;",
24         HOSTNAME, DBNAME, USERNAME, PASSWORD);
25     // Query templates
26     final static String QUERY_TEMPLATE_1 = "EXEC addCustomerDetails @customer_name = ?, @customer_address = ?, @category = ?";
27     final static String QUERY_TEMPLATE_2 = "EXEC addDepartmentDetails @department_number = ?, @department_data = ?";
28     final static String QUERY_TEMPLATE_3 = "EXEC addNewProcess @process_id = ?, @process_data = ?, @paint_type = ?, @painting_method = ?, @fit_type = ?,"
29         + "@cutting_type = ?, @machine_type = ?, @department_number = ?, @process_type = ?";
30     final static String QUERY_TEMPLATE_4 = "EXEC addNewAssembly @assembly_id = ?, @date_ordered = ?, @assembly_details = ?, @customer_name = ?, @process_id
31     final static String QUERY_TEMPLATE_5 = "EXEC addAccountDetailsAndProcess @account_type = ?, @account_number = ?, @cost_details = ?, @date_established = ?, "
32         + "@process_id = ?, @assembly_id = ?, @department_number = ?";
33
34     final static String QUERY_TEMPLATE_6 = "EXEC addJobDetails @job_number = ?, @assembly_id = ?, @process_id = ?, "
35         + "@job_commenced_date = ?, @job_type = ?, @machine_type_used = ?, @material_used = ?, @color = ?, @volume = ?";
36     final static String QUERY_TEMPLATE_7 = "EXEC updateJobDetails @job_number = ?, @job_completed_date = ?,"
37         + " @amount_of_time_machine_used = ?, @labor_time = ?";
38     final static String QUERY_TEMPLATE_8 = "EXEC updateCostDetails @transaction_number = ?, @sup_cost = ?, @account_number = ?";
39     final static String QUERY_TEMPLATE_9 = "EXEC getTotalCostIncurred @assembly_id = ?, @total_cost = ?";
40     final static String QUERY_TEMPLATE_10 = "EXEC GetTotalLaborTimeInDepartment @department_number = ?, @job_completed_date = ?, @total_labor_time = ?";
41     final static String QUERY_TEMPLATE_11 = "EXEC getProcessDepartmentDetails @assembly_id = ?";
42     final static String QUERY_TEMPLATE_12 = "EXEC getCustomersForCategories @range_start = ?, @range_end = ?";
43     final static String QUERY_TEMPLATE_13 = "EXEC deleteCutJobs @job_number_start = ?, @job_number_end = ?";
44     final static String QUERY_TEMPLATE_14 = "EXEC ChangeColorPaintJob @paint_job_number = ?, @color = ?";
45     final static String QUERY_TEMPLATE_16 = "EXEC getCustomerData @start_range = ?, @end_range = ?";
46
47     final static DateTimeFormatter dtf = DateTimeFormatter.ofPattern("MM-dd-yyyy");
48     // User input prompt
49     final static String PROMPT =
50         "\nPlease select one of the options below: \n" +
51         "(1) Enter a new customer \n" +
52         "(2) Enter a new department \n" +
53         "(3) Enter a new process-id and its department together with its type and information \n" +
54         "\trelevant to the type\n" +
55         "(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, \n" +
56         "\tand dateordered and associate it with one or more processes\n" +
57         "(5) Create a new account and associate it with the process, assembly, or department \n" +
58         "\tto which it is applicable\n" +
59         "(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced\n" +
60         "(7) At the completion of a job, enter the date it completed and the information \n" +
61         "\trelevant to the type of job \n" +
62         "(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the \n" +
63         "\ttagged accounts by adding sup-cost to their current values of details \n" +
64         "(9) Retrieve the total cost incurred on an assembly-id \n" +
65         "(10) Retrieve the total labor time within a department for jobs completed in the \n" +
66         "\tdepartment during a given date\n" +
67         "(11) Retrieve the processes through which a given assembly-id has passed so far \n" +
68         "\t(in datecommenced order) and the department responsible for each process\n" +
69         "(12) Retrieve the customers (in name order) whose category is in a given range\n" +
70         "(13) Delete all cut-jobs whose job-no is in a given range\n" +
71         "(14) Change the color of a given paint job\n" +
72         "(15) Import: enter new customers from a data file until the file is empty \n" +
73         "\t(the user must be asked to enter the input file name). \n" +
74         "(16) Export: Retrieve the customers (in name order) whose category is in a given range \n" +
75         "\tand output them to a data file instead of
76         "(17) Quit\n";
77
78     public static void main(String[] args) throws SQLException, IOException {
79         System.out.println("WELCOME TO JOB-SHOP ACCOUNTING SYSTEM !");
80         final Scanner sc = new Scanner(System.in);
81
82         String option = "";
83
84         while (!option.equals("17")) {
85             System.out.println(PROMPT);
86             System.out.println("Enter your option : ");
87             option = sc.next();
88
89             switch (option) {
90                 case "1":
91                     System.out.println("Enter customer name:");
92                     sc.nextLine();
93                     final String customer_name = sc.nextLine();
94                     System.out.println("Enter customer address:");
95                     final String customer_address = sc.nextLine();
96

```

```

99      System.out.println("Enter category:");
100      final int category = sc.nextInt();
101      System.out.println("Connecting to the database...");
102
103      try (final Connection connection = DriverManager.getConnection(URL)) {
104          try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_1)) {
105              statement.setString(1, customer_name);
106              statement.setString(2, customer_address);
107              statement.setInt(3, category);
108              statement.execute();
109          }
110      }
111      break;
112  case "2":
113      System.out.println("Enter Department number:");
114      final int department_number = sc.nextInt();
115
116      System.out.println("Enter Department data:");
117      sc.nextLine();
118      final String department_data = sc.nextLine();
119      System.out.println("Connecting to the database...");
120
121      try (final Connection connection = DriverManager.getConnection(URL)) {
122          try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_2)) {
123              statement.setInt(1, department_number);
124              statement.setString(2, department_data);
125              statement.execute();
126          }
127      }
128      break;
129
130
131  case "3":
132
133      System.out.println("Enter the Process ID:");
134      final int process_id = sc.nextInt();
135
136      System.out.println("Enter the Department Number:");
137      final int department_number3 = sc.nextInt();
138
139      System.out.println("Enter the Process Data:");
140      sc.nextLine();
141
142      final String process_data = sc.nextLine();
143
144      System.out.println("Please enter the process_type (cut/paint/fit) to insert");
145      final String process_type = sc.nextLine();
146      String cutting_type = "";
147      String machine_type = "";
148      String paint_type = "";
149      String painting_method = "";
150      String fit_type = "";
151
152      if(process_type.equals("paint")) {
153
154          System.out.println("Enter the paint type:");
155          paint_type = sc.nextLine();
156
157          System.out.println("Enter the painting method:");
158          painting_method = sc.nextLine();
159
160      }
161

```

```

162         else if (process_type.equals("cut")) {
163
164             System.out.println("cutting type:");
165             cutting_type = sc.nextLine();
166
167             System.out.println("Enter the machine type:");
168             machine_type = sc.nextLine();
169         }
170     else if (process_type.equals("fit")) {
171
172         System.out.println("Enter the fit type:");
173         fit_type = sc.nextLine();
174     }
175 }
176 System.out.println("Connecting to the database...");
177
178 try (final Connection connection = DriverManager.getConnection(URL)) {
179     System.out.println("Dispatching the query...");
180     try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_3)) {
181         statement.setInt(1, process_id);
182         statement.setString(2, process_data);
183         statement.setString(3, paint_type);
184         statement.setString(4, painting_method);
185         statement.setString(5, fit_type);
186         statement.setString(6, cutting_type);
187         statement.setString(7, machine_type);
188         statement.setInt(8, department_number3);
189         statement.setString(9, process_type);
190         statement.execute();
191     }
192 }
193 break;

```

```

195 case "4":
196     System.out.println("Enter the new assembly id:");
197     final int assembly_id = sc.nextInt();
198
199     System.out.println("Enter the date ordered (MM-dd-yyyy):");
200     sc.nextLine();
201     final String date_ordered = sc.nextLine();
202
203
204     System.out.println("Enter Assembly Details:");
205     final String assembly_details = sc.nextLine();
206
207
208     System.out.println("Enter Customer Name");
209     final String customer_name1 = sc.nextLine();
210
211     System.out.println("Enter Existing process id");
212     final String process_ids = sc.nextLine();
213
214     System.out.println("Connecting to the database...");
215
216     try (final Connection connection = DriverManager.getConnection(URL)) {
217         try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_4)) {
218             statement.setInt(1, assembly_id);
219             statement.setString(2, date_ordered);
220             statement.setString(3, assembly_details);
221             statement.setString(4, customer_name1);
222             statement.setString(5, process_ids);
223             statement.execute();
224         }
225     }
226     break;

```

```

228         case "5":
229
230
231             System.out.println("Enter new account number:");
232             final int account_number = sc.nextInt();
233
234
235             System.out.println("Enter the Date Established:");
236             sc.nextLine();
237             final String date_established = sc.nextLine();
238
239
240             System.out.println("Enter the account type (process, assembly, department) case-sensitive:");
241             final String account_type = sc.nextLine();
242
243
244
245
246             int process_id3 = 0;
247             int assembly_id5 = 0;
248             int department_number2 = 0;
249             if(account_type.equals("process")) {
250
251                 System.out.println("Enter existing process id:");
252                 process_id3 = sc.nextInt();
253             }
254
255             else if (account_type.equals("assembly")) {
256                 System.out.println("Enter existing assembly id:");
257                 assembly_id5 = sc.nextInt();
258             }
259
260
261             else if (account_type.equals("department")) {
262
263                 System.out.println("Enter existing department number:");
264                 department_number2 = sc.nextInt();
265             }
266
267             System.out.println("cost details:");
268             final float cost_details = sc.nextFloat();
269
270             System.out.println("Connecting to the database...");
271
272             try (final Connection connection = DriverManager.getConnection(URL)) {
273                 try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_5)) {
274                     statement.setString(1, account_type);
275                     statement.setInt(2, account_number);
276                     statement.setFloat(3, cost_details);
277                     statement.setString(4, date_established);
278                     statement.setInt(5, process_id3);
279                     statement.setInt(6, assembly_id5);
280                     statement.setInt(7, department_number2);
281
282                     statement.execute();
283                 }
284             }
285             break;
286
287             case "6":
288                 System.out.println("Enter new job number:");
289                 final int job_number = sc.nextInt();
290                 System.out.println("Enter job commenced date:");
291                 sc.nextLine();
292                 final String job_commenced_date = sc.nextLine();
293
294
295                 System.out.println("Enter Existing assembly id:");
296                 final int assembly_id6 = sc.nextInt();
297                 System.out.println("process id:");
298                 final int process_id6 = sc.nextInt();
299                 System.out.println("Enter job type (paint, fit, cut) case-sensitive:");
300                 final String job_type = sc.next();
301                 String machine_type_used = "";
302                 String material_used = "";
303                 String color = "";
304                 String volume = "";
305                 if(job_type.equals("cut")) {
306                     System.out.println("Enter machine type used:");
307                     sc.nextLine();
308                     machine_type_used = sc.nextLine();
309
310                     System.out.println("Enter material used:");
311                     material_used = sc.nextLine();
312                 }

```



```

312         else if(job_type.equals("paint")) {
313             System.out.println("Enter color:");
314             sc.nextLine();
315             color = sc.nextLine();
316             System.out.println("Enter volume:");
317             volume = sc.nextLine();
318         }
319         System.out.println("Connecting to the database...");
320         try (final Connection connection = DriverManager.getConnection(URL)) {
321             try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_6)) {
322                 statement.setInt(1, job_number);
323                 statement.setInt(2, assembly_id6);
324                 statement.setInt(3, process_id6);
325                 statement.setString(4, job_commenced_date);
326                 statement.setString(5, job_type);
327                 statement.setString(6, machine_type_used);
328                 statement.setString(7, material_used);
329                 statement.setString(8, color);
330                 statement.setString(9, volume);
331                 statement.execute();
332             }
333         }
334         break;
335     case "7":
336
337         System.out.println("Enter the job number");
338         final int job_number7 = sc.nextInt();
339
340         System.out.println("Enter job completion date:");
341         sc.nextLine();
342         final String job_completed_date = sc.nextLine();
343
344
345         System.out.println("Enter Labour Time:");
346         final float labor_time = sc.nextFloat();
347
348         System.out.println("Enter Time Machine Used:");
349         final float amount_of_time_machine_used = sc.nextFloat();
350
351         System.out.println("Connecting to the database...");
352
353         try (final Connection connection = DriverManager.getConnection(URL)) {
354             try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_7)) {
355                 cs.setInt(1, job_number7);
356                 cs.setString(2, job_completed_date);
357                 cs.setFloat(3, amount_of_time_machine_used);
358                 cs.setFloat(4, labor_time);
359                 // Execute the stored procedure
360                 cs.execute();
361             }
362         }
363         break;
364     case "8":
365
366         System.out.println("Enter Transaction Number:");
367         final int transaction_number = sc.nextInt();
368
369         System.out.println("Enter the sup cost:");
370         final float sup_cost = sc.nextFloat();
371
372         System.out.println("Enter the Account Number:");
373         final int account_number8 = sc.nextInt();
374
375         System.out.println("Connecting to the database...");
376
377         try (final Connection connection = DriverManager.getConnection(URL)) {
378             try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_8)) {
379                 statement.setInt(1, transaction_number);
380                 statement.setFloat(2, sup_cost);
381                 statement.setInt(3, account_number8);
382                 statement.execute();
383             }
384         }
385         break;
386     }

```

```

386         case "9":
387
388             System.out.println("Enter assembly id:");
389             final int assembly_id9 = sc.nextInt();
390
391             System.out.println("Connecting to the database...");
392
393             try (final Connection connection = DriverManager.getConnection(URL)) {
394                 try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_9)) {
395
396                     cs.setInt(1, assembly_id9);
397                     cs.registerOutParameter(2, Types.DECIMAL);
398                     cs.execute();
399
400                     BigDecimal totalCost = cs.getBigDecimal(2);
401                     System.out.println("Final Total Cost: " + totalCost);
402                 }
403             }
404             break;
405         case "10":
406
407             System.out.println("Please Enter Department Number:");
408             final int department_number10 = sc.nextInt();
409
410             System.out.println("Please Enter Job completed date:");
411             sc.nextLine();
412             final String date_job_completed = sc.nextLine();
413
414             System.out.println("Connecting to the database...");
415
416             try (final Connection connection = DriverManager.getConnection(URL)) {
417                 try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_10)) {
418
419                     try (final Connection connection = DriverManager.getConnection(URL)) {
420                         CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_10);
421                         cs.setInt(1, department_number10);
422                         cs.setString(2, date_job_completed);
423                         cs.registerOutParameter(3, Types.DECIMAL);
424                         cs.execute();
425                         BigDecimal totalCost = cs.getBigDecimal(3);
426                         System.out.println("Total cost: " + totalCost);
427                     }
428                 }
429                 break;
430             }
431         case "11":
432
433             System.out.println("Please Enter the assembly id:");
434             final int assembly_id11 = sc.nextInt();
435
436             System.out.println("Connecting to the database...");
437
438             try (final Connection connection = DriverManager.getConnection(URL)) {
439                 CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_11);
440                 cs.setInt(1, assembly_id11);
441                 System.out.println("Dispatching the query...");
442                 ResultSet resultSet = cs.executeQuery();
443                 System.out.println("Done.");
444                 System.out.println("\nProcess for assembly-id: " + assembly_id11 +
445                                     ", and its department number; Sorted by date commenced.");
446                 System.out.println("processID | deptNo");
447                 while (resultSet.next()) {
448                     System.out.println(String.format("%s | %s ",
449                                                         resultSet.getString(1),
450                                                         resultSet.getString(2)));
451                 }
452             }
453             break;
454         case "12":
455
456             System.out.println("Enter the Start Range of Category:");
457             final int range_start = sc.nextInt();
458
459             System.out.println("Enter the End Range of category:");
460             final int range_end = sc.nextInt();
461
462             System.out.println("Connecting to the database...");

```

```

459
460
461     try (final Connection connection = DriverManager.getConnection(URL)) {
462         try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_12)); {
463             cs.setInt(1, range_start);
464             cs.setInt(2, range_end);
465             System.out.println("Dispatching the query...");
466             ResultSet resultSet = cs.executeQuery();
467             System.out.println("Done.");
468             System.out.println("\nJobs from start date " + range_start +
469                 " completed on: " + range_end);
470             System.out.println("customer name");
471             while (resultSet.next()) {
472                 System.out.println(String.format("%s",
473                     resultSet.getString(1)));
474             }
475         }
476     }
477     break;
478 case "13":
479
480     System.out.println("Enter Job Number Start Value:");
481     final int job_number_start = sc.nextInt();
482
483     System.out.println("Enter Job Number End Value:");
484     final int job_number_end = sc.nextInt();
485
486     System.out.println("Connecting to the database...");
487
488     try (final Connection connection = DriverManager.getConnection(URL)) {
489         try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_13)) {
490             statement.setInt(1, job_number_start);
491             statement.setInt(2, job_number_end);
492             statement.executeUpdate();
493         }
494     }
495     break;
496 case "14":
497
498     System.out.println("job number:");
499     final int job_number14 = sc.nextInt();
500
501     System.out.println("color:");
502     final String color1 = sc.next();
503
504     System.out.println("Connecting to the database...");
505
506     try (final Connection connection = DriverManager.getConnection(URL)) {
507         try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_14)) {
508             statement.setInt(1, job_number14);
509             statement.setString(2, color1);
510             int rows = statement.executeUpdate();
511             System.out.println(rows);
512         }
513     }
514     break;
515 case "15":
516     System.out.println("Please enter name of CSV file with customer data");
517     String filename = sc.next();
518     String query = readCSV(filename);
519     // Database connection
520     try (final Connection connection = DriverManager.getConnection(URL)) {
521         PreparedStatement ps = connection.prepareCall(query);
522
523         try (final Connection connection = DriverManager.getConnection(URL)) {
524             PreparedStatement ps = connection.prepareCall(query);
525             System.out.println("Dispatching the query...");
526             final int inserted_rows = ps.executeUpdate();
527             System.out.println(String.format("rows inserted." + inserted_rows));
528         }
529     }
530     break;
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

527     case "16":
528         System.out.println("Please enter start range of category number");
529         int start_range = sc.nextInt();
530         System.out.println("Please enter end range of category number");
531         int end_range = sc.nextInt();
532
533         System.out.println("Enter the file name:");
534         sc.nextLine();
535         String filename16 = sc.nextLine();
536         try (final Connection connection = DriverManager.getConnection(URL)) {
537             try (CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_16)) {
538                 cs.setInt(1, start_range);
539                 cs.setInt(2, end_range);
540
541                 // Run the stored procedure and store values in resultSet
542                 System.out.println("Dispatching the query...");
543                 try (ResultSet resultSet = cs.executeQuery()) {
544                     try (FileWriter myWriter = new FileWriter(filename16 + ".csv")) {
545                         myWriter.write("customer_name,customer_address,category\n");
546
547                         // Unpack the tuples returned by the database and write them to the file
548                         while (resultSet.next()) {
549                             myWriter.write(String.format("%s,%s,%s\n",
550                                 resultSet.getString(1),
551                                 resultSet.getString(2),
552                                 resultSet.getString(3)));
553                         }
554                     } catch (IOException e) {
555                         System.out.println("File Name Error");
556                         e.printStackTrace();
557                     }
558                 }
559             }
560         }
561
562         System.out.println(filename16 + ".csv");
563         break;
564     case "17":
565         System.out.println("Quitting the Job Shop Accounting Database");
566         break;
567     default:
568         System.out.println("Wrong Option");
569         break;
570 }
571
572 }
573
574 sc.close();
575 }
576
577 // Reading CSV file
578 public static String readCSV(String filename) throws IOException, SQLException {
579     StringBuilder insertStatement = new StringBuilder("INSERT INTO customers VALUES ");
580     // Input reading
581     try (BufferedReader input = new BufferedReader(new FileReader(filename))) {
582         String line;
583         int iterCount = 0; // keep track of iterations
584         final int FIRST_ITER = 0;
585         // Iterate through each 'row' of the csv
586         while ((line = input.readLine()) != null) {
587             // First iteration
588             if (iterCount != FIRST_ITER) {
589                 insertStatement.append(", ");

```

```

        insertStatement.append(", ");
    } else {
        ++iterCount;
        // Split the line into values based on comma
        String[] values = line.split(",");
        // insertion
        insertStatement.append("(")
            .append(values[0].trim())
            .append(", ")
            .append(values[1].trim())
            .append(", ")
            .append(values[2].trim())
            .append(")");
    }
}
return insertStatement.toString();
}
}

```

Compilation of the program

```

WELCOME TO JOB-SHOP ACCOUNTING SYSTEM !

Please select one of the options below:
(1) Enter a new customer
(2) Enter a new department
(3) Enter a new process-id and its department together with its type and information
    relevant to the type
(4) Enter a new assembly with its customer-name, assembly-details, assembly-id,
    and dateordered 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 datecommenced order) and the department responsible for each process
(12) Retrieve the customers (in name order) whose category is in a given range
(13) Delete all cut-jobs whose job-no is in a given range
(14) Change the color of a given paint job
(15) Import: enter new customers from a data file until the file is empty
    ( the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
    and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit

```

Task 6: Java program Execution

6.1: Query 1 Screenshots

```

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

```

Enter your option :

1

Enter customer name:

Sruthi Pedakolimi

Enter customer address:

Beaumont Drive Norman

Enter category:

3

Connecting to the database...

Enter your option :

1

Enter customer name:

Dhana Laxmi

Enter customer address:

Vijayawada Andhra

Enter category:

6

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

Enter your option :

1

Enter customer name:

Vijaya Kumar

Enter customer address:

Spring Field, California

Enter category:

9

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

(2) Enter a new department

(17) Quit

Enter your option :

1

Enter customer name:

Ratan

Enter customer address:

Hyderabad, Telengana

Enter category:

1

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

Enter your option :

1

Enter customer name:

Rithwik

Enter customer address:

Summer point, Boston

Enter category:

7

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

Customer Table after 5 insertions

Run	Cancel	Disconnect	Change	Database: cs-dsa-4513-sql-db	Estimate
1	select * from customers;				
2					
Results					
	customer_name	customer_address	category		
1	Dhana Laxmi	Vijayawada Andhra	6		
2	Ratan	Hyderabad, Telengana	1		
3	Rithwik	Summer point, Boston	7		
4	Sruthi Pedakolimi	Beaumont Drive Norman	3		
5	Vijaya Kumar	Spring Field, California	9		

6.2: Query 2 Screenshots

Enter your option :

2

Enter Department number:

1001

Enter Department data:

Data Science and Analytics

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

(2) Enter a new department

Enter your option :

2

Enter Department number:

1002

Enter Department data:

Computer Science

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

(2) Enter a new department

Enter your option :

2

Enter Department number:

1003

Enter Department data:

Aerospace and Mechanical

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

(2) Enter a new department

Enter your option :

2

Enter Department number:

1004

Enter Department data:

Civil Engineering

Connecting to the database...

Please select one of the options below:

(1) Enter a new customer

(2) Enter a new department


```

Enter your option :
2
Enter Department number:
1005
Enter Department data:
Industrial & System Engineering
Connecting to the database...

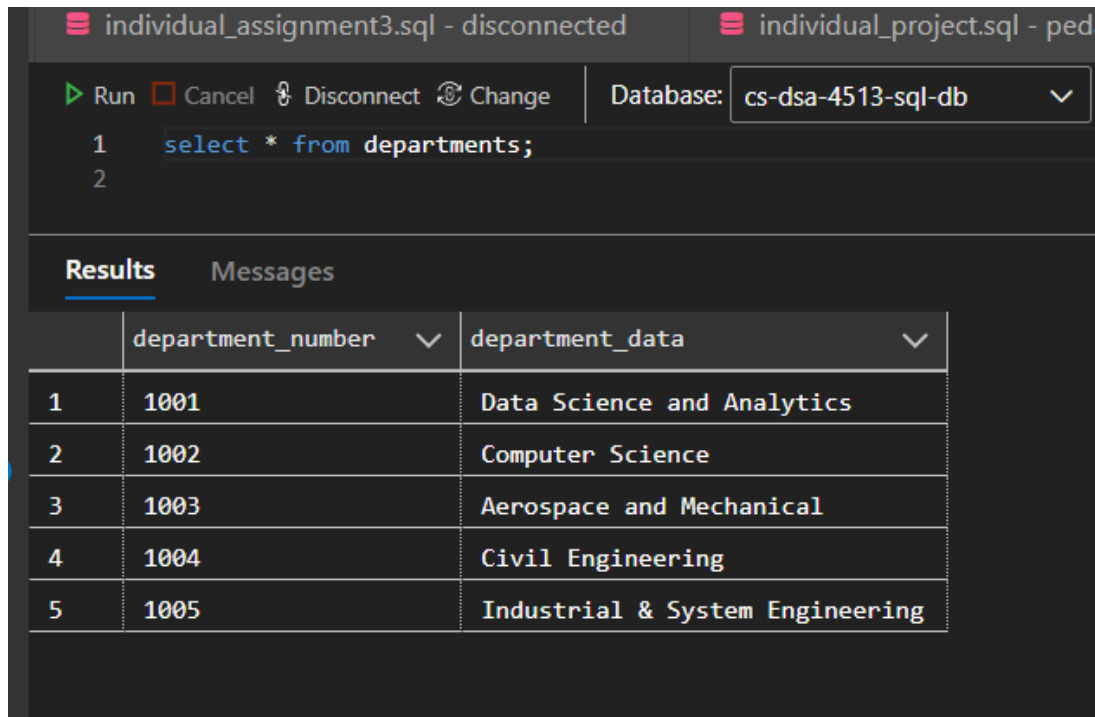
```

```

Please select one of the options below:
(1) Enter a new customer
(2) Enter a new department

```

Output of the department after insertion



individual_assignment3.sql - disconnected | individual_project.sql - ped

Run Cancel Disconnect Change Database: cs-dsa-4513-sql-db

```
1 select * from departments;
2
```

Results Messages

	department_number	department_data
1	1001	Data Science and Analytics
2	1002	Computer Science
3	1003	Aerospace and Mechanical
4	1004	Civil Engineering
5	1005	Industrial & System Engineering

6.3: Query 3 Screenshots

Enter your option :

3

Enter the Process ID:

2001

Enter the Department Number:

1001

Enter the Process Data:

process data 1

Please enter the process_type (cut/paint/fit) to insert

cut

cutting type:

type 1

Enter the machine type:

m_type1

Connecting to the database...

Dispatching the query...

Enter your option :

3

Enter the Process ID:

2002

Enter the Department Number:

1002

Enter the Process Data:

process data 2

Please enter the process_type (cut/paint/fit) to insert

paint

Enter the paint type:

ptype1

Enter the painting method:

paint method 1

Connecting to the database...

Dispatching the query...

Enter your option :

3

Enter the Process ID:

2003

Enter the Department Number:

1003

Enter the Process Data:

process data 3

Please enter the process_type (cut/paint/fit) to insert

fit

Enter the fit type:

ftype 1

Connecting to the database...

Dispatching the query...

Enter your option :

3

Enter the Process ID:

2004

Enter the Department Number:

1004

Enter the Process Data:

process data 4

Please enter the process_type (cut/paint/fit) to insert

paint

Enter the paint type:

paint type 2

Enter the painting method:

paint method 2

Connecting to the database...

Dispatching the query...

```

Enter your option :
3
Enter the Process ID:
2005
Enter the Department Number:
1005
Enter the Process Data:
process data 5
Please enter the process_type (cut/paint/fit) to insert
fit
Enter the fit type:
fit type 2
Connecting to the database...
Dispatching the query...

```

```

Enter your option :
3
Enter the Process ID:
2006
Enter the Department Number:
1002
Enter the Process Data:
process data 6
Please enter the process_type (cut/paint/fit) to insert
fit
Enter the fit type:
fit type 3
Connecting to the database...
Dispatching the query...

```

```

Enter your option :
3
Enter the Process ID:
2007
Enter the Department Number:
1001
Enter the Process Data:
process data 7
Please enter the process_type (cut/paint/fit) to insert
paint
Enter the paint type:
paint type 3
Enter the painting method:
painting method 3
Connecting to the database...
Dispatching the query...

```

```

Enter your option :
3
Enter the Process ID:
2008
Enter the Department Number:
1005
Enter the Process Data:
PD8
Please enter the process_type (cut/paint/fit) to insert
cut
cutting type:
CT3
Enter the machine type:
MT1
Connecting to the database...
Dispatching the query...

```

Enter your option :

3

Enter the Process ID:

2009

Enter the Department Number:

1003

Enter the Process Data:

PD5

Please enter the process_type (cut/paint/fit) to insert

fit

Enter the fit type:

F8T

Connecting to the database...

Dispatching the query...

Enter your option :

3

Enter the Process ID:

2010

Enter the Department Number:

1005

Enter the Process Data:

PD10

Please enter the process_type (cut/paint/fit) to insert

fit

Enter the fit type:

F65

Connecting to the database...

Dispatching the query...

Output of tables involved

	process_id	paint_type	painting_method
1	2002	ptype1	paint method 1
2	2004	paint type 2	paint method 2
3	2007	paint type 3	painting method 3

	process_id	cutting_type	machine_type
1	2001	type 1	m_type1
2	2008	CT3	MT1

	process_id	fit_type
1	2003	ftype 1
2	2005	fit type 2
3	2006	fit type 3
4	2009	F8T
5	2010	F65

Results		Messages
	process_id	process_data
1	2001	process data 1
2	2002	process data 2
3	2003	process data 3
4	2004	process data 4
5	2005	process data 5
6	2006	process data 6
7	2007	process data 7
8	2008	PD8
9	2009	PD5
10	2010	PD10

	process_id	department_number
1	2001	1001
2	2007	1001
3	2002	1002
4	2006	1002
5	2003	1003
6	2009	1003
7	2004	1004
8	2005	1005
9	2008	1005
10	2010	1005

6.4: Query 4 Screenshots

```

Enter your option :
4
Enter the new assembly id:
3001
Enter the date ordered (MM-dd-yyyy):
01-21-2023
Enter Assembly Details:
AD1
Enter Customer Name
Dhana Laxmi
Enter Existing process id
2001
Connecting to the database...

```

```

Enter your option :
4
Enter the new assembly id:
3003
Enter the date ordered (MM-dd-yyyy):
02-04-2023
Enter Assembly Details:
AD3
Enter Customer Name
Rithwik
Enter Existing process id
2003
Connecting to the database...

```

```

4
Enter the new assembly id:
3002
Enter the date ordered (MM-dd-yyyy):
02-23-2023
Enter Assembly Details:
AD2
Enter Customer Name
Ratan
Enter Existing process id
2002
Connecting to the database...

```

```

Enter your option :
4
Enter the new assembly id:
3004
Enter the date ordered (MM-dd-yyyy):
06-05-2023
Enter Assembly Details:
AD4
Enter Customer Name
Sruthi Pedakolimi
Enter Existing process id
2007
Connecting to the database...

```

```

4
Enter the new assembly id:
3005
Enter the date ordered (MM-dd-yyyy):
09-24-2023
Enter Assembly Details:
AD5
Enter Customer Name
Rithwik
Enter Existing process id
2007
Connecting to the database...

```

```

Enter your option :
4
Enter the new assembly id:
3007
Enter the date ordered (MM-dd-yyyy):
04-15-2023
Enter Assembly Details:
AD7
Enter Customer Name
Vijaya Kumar
Enter Existing process id
2001
Connecting to the database...

```

```

Enter your option :
4
Enter the new assembly id:
3006
Enter the date ordered (MM-dd-yyyy):
06-13-2023
Enter Assembly Details:
AD6
Enter Customer Name
Ratan
Enter Existing process id
2009
Connecting to the database...

```

```

Enter your option :
4
Enter the new assembly id:
3008
Enter the date ordered (MM-dd-yyyy):
01-29-2023
Enter Assembly Details:
AD8
Enter Customer Name
Ratan
Enter Existing process id
2010
Connecting to the database...

```

Enter your option :

4

Enter the new assembly id:

3009

Enter the date ordered (MM-dd-yyyy):

08-12-2022

Enter Assembly Details:

AD9

Enter Customer Name

Vijaya Kumar

Enter Existing process id

2005

Connecting to the database...

Enter your option :

4

Enter the new assembly id:

3010

Enter the date ordered (MM-dd-yyyy):

06-04-2022

Enter Assembly Details:

AD10

Enter Customer Name

Sruthi Pedakolimi

Enter Existing process id

2006

Connecting to the database...

Output

```

1 SELECT * from customers
2 SELECT * from assemblies
3 select * from orders
4 SELECT * from manufactures
  
```

Results		Messages	
	assembly_id	date_ordered	assembly_details
1	3001	2023-01-21	AD1
2	3002	2023-02-23	AD2
3	3003	2023-02-04	AD3
4	3004	2023-06-05	AD4
5	3005	2023-09-24	AD5
6	3006	2023-06-13	AD6
7	3007	2023-04-15	AD7
8	3008	2023-01-29	AD8
9	3009	2022-08-12	AD9
10	3010	2022-06-04	AD10

	assembly_id	customer_name
1	3001	Dhana Laxmi
2	3002	Ratan
3	3003	Rithwik
4	3004	Sruthi Pedakolimi
5	3005	Rithwik
6	3006	Ratan
7	3007	Vijaya Kumar
8	3008	Ratan
9	3009	Vijaya Kumar
10	3010	Sruthi Pedakolimi

	assembly_id	process_id
1	3001	2001
2	3002	2002
3	3003	2003
4	3004	2007
5	3005	2007
6	3006	2009
7	3007	2001
8	3008	2010
9	3009	2005
10	3010	2006

6.5: Query 5 Screenshots

```

Enter your option :
5
Enter new account number:
4001
Enter the Date Established:
03-15-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2001
cost details:
24
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4002
Enter the Date Established:
06-22-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3001
cost details:
543
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4003
Enter the Date Established:
09-08-2022
Enter the account type (process, assembly, department) case-sensitive:
department
Enter existing department number:
1001
cost details:
7647
Connecting to the database...

```



```

Enter your option :
5
Enter new account number:
4004
Enter the Date Established:
11-04-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3004
cost details:
5425
Connecting to the database...

```

```

\ / \

```

```

Enter your option :
5
Enter new account number:
4005
Enter the Date Established:
02-19-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2003
cost details:
46432
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4006
Enter the Date Established:
05-07-2022
Enter the account type (process, assembly, department) case-sensitive:
department
Enter existing department number:
1006
cost details:
8768
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4007
Enter the Date Established:
08-12-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3009
cost details:
8975674
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4008
Enter the Date Established:
12-30-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2010
cost details:
543
Connecting to the database...

```

Please select one of the options below:

```

Enter your option :
5
Enter new account number:
4009
Enter the Date Established:
04-25-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2005
cost details:
87653421
Connecting to the database...

```

```

Enter your option :
5
Enter new account number:
4010
Enter the Date Established:
07-18-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3008
cost details:
65745
Connecting to the database...

```

Tables that are changed:

Account Table

```

1 select * from account
2 select * from department_account
3 select * from assembly_account
4 select * from process_account
5 select * from maintains_assembly
6 select * from maintains_department
7 select * from maintains_process

```

Results		Messages
	account_number ▾	date_established ▾
1	4001	2022-03-15
2	4002	2022-06-22
3	4003	2022-09-08
4	4004	2022-11-04
5	4005	2022-02-19
6	4006	2022-05-07
7	4007	2022-08-12
8	4008	2022-12-30
9	4009	2022-04-25
1...	4010	2022-07-18

Department Account, Assembly Account, Process Account

	account_number	cost_details_2
1	4003	7647.00
2	4006	8768.00

	account_number	cost_details_1
1	4004	5425.00
2	4007	8975674.00
3	4010	65745.00

	account_number	cost_details_3
1	4001	24.00
2	4005	46432.00
3	4008	543.00
4	4009	87653424.00

Maintain Assembly, Maintain Department & Maintain Process Tables

	assembly_id	account_number
1	3001	4002
2	3004	4004
3	3008	4010
4	3009	4007

	department_number	account_number
1	1001	4003

	process_id	account_number
1	2001	4001
2	2003	4005
3	2005	4009
4	2010	4008

6.6: Query 6 Screenshots

```

Enter your option :
6
Enter new job number:
5001
Enter job commenced date:
04-03-2023
Enter Existing assembly id:
3001
process id:
2001
Enter job type (paint, fit, cut) case-sensitive:
paint
Enter color:
blue
Enter volume:
23
Connecting to the database...

```

```

Enter your option :
6
Enter new job number:
5002
Enter job commenced date:
07-19-2023
Enter Existing assembly id:
3002
process id:
2002
Enter job type (paint, fit, cut) case-sensitive:
fit
Connecting to the database...

```

```

Enter your option :
6
Enter new job number:
5003
Enter job commenced date:
10-06-2023
Enter Existing assembly id:
3003
process id:
2003
Enter job type (paint, fit, cut) case-sensitive:
cut
Enter machine type used:
cut machine 1
Enter material used:
iron
Connecting to the database...

```

```

Enter your option :
6
Enter new job number:
5004
Enter job commenced date:
02-28-2023
Enter Existing assembly id:
3004
process id:
2004
Enter job type (paint, fit, cut) case-sensitive:
paint
Enter color:
red
Enter volume:
5
Connecting to the database...

```

```

1 select * from job
2 select * from assigns
3 select * from cut_job
4 select * from paint_job
5 select * from fit_job
6
7

```

Results

Messages

	job_number	job_commenced_date	job_completed_date	labor_time
1	5001	2023-04-03	NULL	NULL
2	5002	2023-07-19	NULL	NULL
3	5003	2023-10-06	NULL	NULL
4	5004	2023-02-28	NULL	NULL
5	5005	2023-05-15	NULL	NULL
6	5006	2023-08-22	NULL	NULL
7	5007	2023-11-11	NULL	NULL
8	5008	2023-03-04	NULL	NULL
9	5009	2023-06-27	NULL	NULL
10	5010	2023-07-19	NULL	NULL

	assembly_id	process_id	job_number
1	3001	2001	5001
2	3002	2002	5002
3	3003	2003	5003
4	3004	2004	5004
5	3005	2005	5005
6	3006	2006	5006
7	3007	2007	5007
8	3008	2008	5008
9	3009	2009	5009
10	3010	2010	5010

	job_number	machine_type_used	amount_of_time_machine_used	material_used
1	5003	cut machine 1	NULL	iron
2	5006	cut M3	NULL	MA5
3	5010	ML3	NULL	Iron

	job_number	color	volume
1	5001	blue	23.00
2	5004	red	5.00
3	5007	green	67.00
4	5009	Yellow	2.00

	job_number
1	5002
2	5005
3	5008

6.7: Query 7 Screenshots

```

Enter your option :
7
Enter the job number
5001
Enter job completion date:
02-15-2023
Enter Labour Time:
50
Enter Process Type:
Connecting to the database...

```

```
Enter your option :  
7  
Enter the job number  
5002  
Enter job completion date:  
04-08-2023  
Enter Labour Time:  
345  
Enter Process Type:  
Connecting to the database...
```

```
Enter your option :  
7  
Enter the job number  
5003  
Enter job completion date:  
06-21-2023  
Enter Labour Time:  
86  
Enter Process Type:  
Connecting to the database...
```

```
Enter your option :  
7  
Enter the job number  
5004  
Enter job completion date:  
08-11-2023  
Enter Labour Time:  
342  
Enter Process Type:  
Connecting to the database...
```

Output Tables:

Job Table

```

1  select * from job
2  select * from paint_job
3  select * from fit_job
4  select * from cut_job
5
6

```

	job_number	job_commenced_date	job_completed_date	labor_time
1	5001	2023-04-03	2023-02-15	50.00
2	5002	2023-07-19	2023-04-08	345.00
3	5003	2023-10-06	2023-06-21	86.00
4	5004	2023-02-28	2023-08-11	342.00
5	5005	2023-05-15	2023-10-05	6784.00
6	5006	2023-08-22	2023-01-28	4335.00
7	5007	2023-11-11	2023-03-19	97.00
8	5008	2023-03-04	2023-05-14	31351.00
9	5009	2023-06-27	2023-07-09	87654.00
10	5010	2023-07-19	2023-09-27	345.00

Paint job, fit job and cut job tables

	job_number	color	volume
1	5001	blue	23.00
2	5004	red	5.00
3	5007	green	67.00
4	5009	Yellow	2.00

	job_number
1	5002
2	5005
3	5008

	job_number	machine_type_used	amount_of_time_machine_used	material_used
1	5003	cut machine 1	0.00	iron
2	5006	cut M3	0.00	MA5
3	5010	ML3	0.00	Iron

6.8: Query 8 Screenshots

Enter your option :

8

Enter Transaction Number:

6001

Enter the sup cost:

35

Enter the Account Number:

4001

Connecting to the database...

Enter your option :

8

Enter Transaction Number:

6002

Enter the sup cost:

35

Enter the Account Number:

4002

Connecting to the database...

Enter your option :

8

Enter Transaction Number:

6003

Enter the sup cost:

3426

Enter the Account Number:

4003

Connecting to the database...

```

Enter your option :
8
Enter Transaction Number:
6004
Enter the sup cost:
42315
Enter the Account Number:
4004
Connecting to the database...

```

Output

Cost transactions

```

2  select * from cost_transactions
3  select * from assembly_account
4
5  select * from department_account
6  select * from process_account
7

```

	transaction_number	account_number	sup_cost
1	6001	4001	35.00
2	6002	4002	35.00
3	6003	4003	3426.00
4	6004	4004	42315.00
5	6005	4005	341.00
6	6006	4006	43332.00
7	6007	4007	3423.00
8	6008	4008	3453.00
9	6009	4009	3243.00
10	6010	4010	547.00

Assembly, Department and Process Accounts respectively

	account_number ▼	cost_details_1 ▼
1	4002	578
2	4004	42749
3	4007	28777
4	4010	889

	account_number ▼	cost_details_2 ▼
1	4003	72104
2	4006	46794

	account_number ▼	cost_details_3 ▼
1	4001	383
2	4005	3803
3	4008	58945
4	4009	434786

6.9: Query 9 Screenshots

Enter your option :

9

Enter assembly id:

3001

Connecting to the database...

Final Total Cost: 35.0000

(17) Quit

Enter your option :

9

Enter assembly id:

3004

Connecting to the database...

Final Total Cost: 42315.0000

Enter your option :

9

Enter assembly id:

3008

Connecting to the database...

Final Total Cost: 547.0000

6.10: Query 10 Screenshots

Enter your option :

10

Please Enter Department Number:

1002

Please Enter Job completed date:

05-06-2023

Connecting to the database...

Total cost: 0.0000

(17) Quit

Enter your option :

10

Please Enter Department Number:

1005

Please Enter Job completed date:

2023-06-07

Connecting to the database...

Total cost: 42343.0000

Enter your option :

10

Please Enter Department Number:

1007

Please Enter Job completed date:

2023-05-06

Connecting to the database...

Total cost: 0.0000

6.11: Query 11 Screenshots

Enter your option :

11

Please Enter the assembly id:

3004

Connecting to the database...

Dispatching the query...

Done.

Process for assembly-id: 3004, and its departement number; Sorted by date commenced.

processID	deptNo
2004	1004

```

Enter your option :
11
Please Enter the assembly id:
3007
Connecting to the database...
Dispatching the query...
Done.

Process for assembly-id: 3007, and its departement number; Sorted by date commenced.
processID | deptNo
2007 | 1001

```

```

Enter your option :
11
Please Enter the assembly id:
3001
Connecting to the database...
Dispatching the query...
Done.

Process for assembly-id: 3001, and its departement number; Sorted by date commenced.
processID | deptNo
2001 | 1001

```

6.12: Query 12 Screenshots

```

Enter your option :
12
Enter the Start Range of Category:
2
Enter the End Range of category:
5
Connecting to the database...
Dispatching the query...
Done.

Jobs from start date 2 completed on: 5
customer name
Sruthi Pedakolimi

```

Enter your option :

12

Enter the Start Range of Category:

1

Enter the End Range of category:

9

Connecting to the database...

Dispatching the query...

Done.

Jobs from start date 1 completed on: 9

customer name

Dhana Laxmi

Ratan

Rithwik

Sruthi Pedakolimi

Vijaya Kumar

Enter your option :

12

Enter the Start Range of Category:

7

Enter the End Range of category:

9

Connecting to the database...

Dispatching the query...

Done.

Jobs from start date 7 completed on: 9

customer name

Rithwik

Vijaya Kumar

6.13: Query 13 Screenshots

```
Enter your option :  
13  
Enter Job Number Start Value:  
0  
Enter Job Number End Value:  
5005  
Connecting to the database...
```

```
Enter your option :  
13  
Enter Job Number Start Value:  
5001  
Enter Job Number End Value:  
5005  
Connecting to the database...
```

```
(1/) Quit
```

```
Enter your option :  
13  
Enter Job Number Start Value:  
5006  
Enter Job Number End Value:  
5009  
Connecting to the database...
```

Here we have deleted all the cut jobs

```

1
2  select * from cut_job
3

```

Results Messages

job_number	machine_type_us...	amount_of_time_...	material_used
------------	--------------------	--------------------	---------------

6.14: Query 14 Screenshots

Before changing the color

```

2  select * from paint_job
3

```

Results Messages

	job_number	color	volume
1	5001	blue	23.00
2	5004	fsd	22.00
3	5006	edf	4.00

```

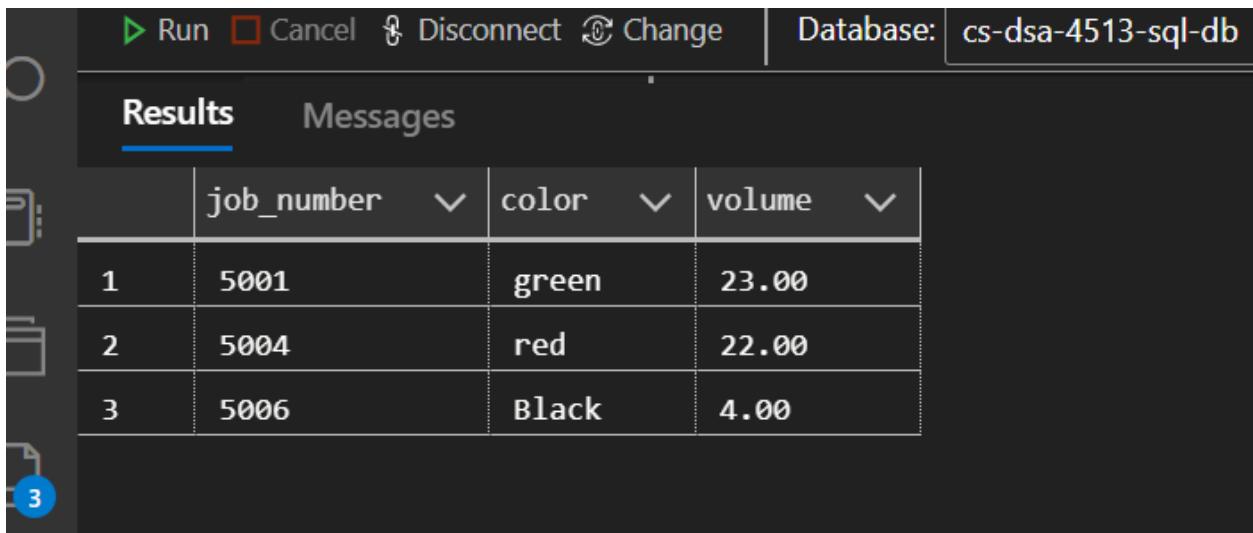
Enter your option :
14
job number:
5004
color:
red
Connecting to the database...
1

```

```
Enter your option :  
14  
job number:  
5001  
color:  
green  
Connecting to the database...  
1
```

```
Enter your option :  
14  
job number:  
5006  
color:  
Black  
Connecting to the database...  
1
```

After Execution

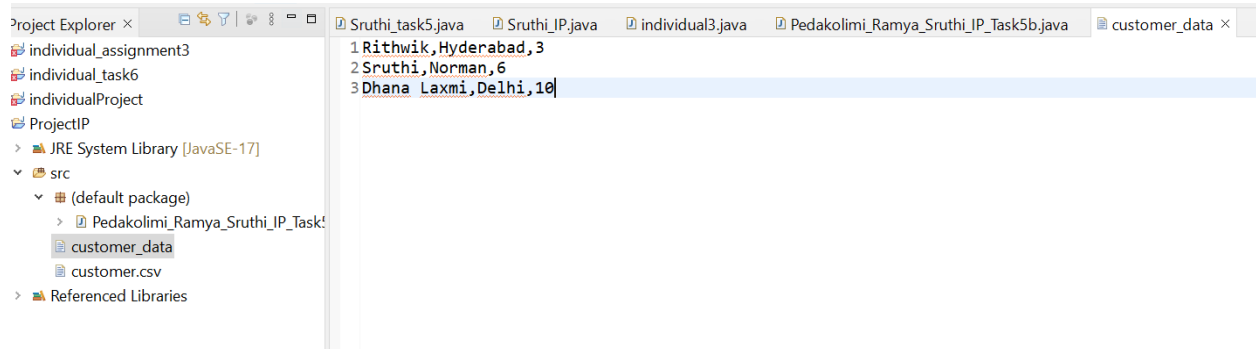


The screenshot shows a database client interface with a dark theme. At the top, there is a toolbar with buttons: 'Run' (green play icon), 'Cancel' (red square icon), 'Disconnect' (plug icon), and 'Change' (refresh icon). To the right of the toolbar, it says 'Database: cs-dsa-4513-sql-db'. Below the toolbar, there are two tabs: 'Results' (selected, underlined) and 'Messages'. The 'Results' tab displays a table with four columns: an index column, 'job_number', 'color', and 'volume'. The table contains three rows of data. On the left side of the interface, there is a sidebar with icons for a database, a folder, and a document with a blue circle containing the number '3'.

	job_number	color	volume
1	5001	green	23.00
2	5004	red	22.00
3	5006	Black	4.00

6.15: Screenshots showing the testing of the import and export options

Importing



Enter your option :

15

Please enter name of CSV file with customer data

\Users\peda001\eclipse-workspace\ProjectIP\src\customer_data

Entered csv

Going into try.

Inside while

Done with while

ROW 0Rithwik

ROW 1Hyderabad

ROW 23

Inside while

Done with while

ROW 0Sruthi

ROW 1Norman

ROW 26

Inside while

Done with while





ROW 0Dhana Laxmi

ROW 1Delhi




ROW 210

Dispatching the query...

Done. 3 rows inserted.

AutoSave ☐ Off    customers • Saved  Search

File Home Insert Page Layout Formulas Data Review View

G10   

	A	B	C	D	E	F	G	H
1	Sujatha	California	3					
2	Harika	Hyderabad	2					
3	Charan	California	8					
4	Harini	Bostom	4					
5	Lokesh	Chicago	7					
6								
7								
8								
9								
10								
11								

Exporting

Enter your option :

16

Please enter start range of category number

3

Please enter end range of category number

8

Enter the file name:

exp_customer

Dispatching the query...

Done. File Location here:

exp_customer.csv

Output file

	A	B	C	D	E
1	customer_na	customer_addres	category		
2	Lokesh	Hyderabad	8		
3	Rithwik	Hyderabad	3		
4	Sruthi	Norman	6		
5					
6					
7					
8					

6.16: Screenshots showing the testing of three types of errors

Error Handling

Category Constraint

```
(
    the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
    and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit

Enter your option :
1
Enter customer name:
Vandana
Enter customer address:
Spring Field
Enter category:
20
Connecting to the database...
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint "CK_cus
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:165
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLSer
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLSer
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.TDSCCommand.execute(IOBuffer.java:7675)
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:27
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:
at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement
at Pedakolimi_Ramya_Sruthi_IP_Task5b.main(Pedakolimi_Ramya_Sruthi_IP_Task5b.java:112)
```

Primary Key Constraint

(16) Export: Retrieve the customers (in name order) whose category is in a given range and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit

Enter your option :

1

Enter customer name:

Sruthi

Enter customer address:

Norman

Enter category:

5

Connecting to the database...

```
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_customer__5B894ACA055B294'
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7675)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:27)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:27)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:169)
    at Pedakolimi_Ramya_Sruthi_IP_Task5b.main(Pedakolimi_Ramya_Sruthi_IP_Task5b.java:112)
```

Date Conversion Error Handling

Enter your option :

4

Enter the new assembly id:

6002

Enter the date ordered (MM-dd-yyyy):

23-02-2032

Enter Assembly Details:

Testing

Enter Customer Name

Many

Enter Existing process id

3154

Connecting to the database...

```
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Conversion failed when converting date and/or time from character
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7675)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:169)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:27)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:27)
    at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:169)
```

6.17: Screenshots showing the testing of the quit option

Enter your option :

17

Quitting the Job Shop Accounting Database

Task 7: Web database application and its execution

Pedakolimi_Ramya_Sruthi_IP_Task7_data_handler

```
package Pedakolimi_Ramya_Sruthi_IP_Task7;

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

public class DataHandler {
    private Connection conn;
    final static String HOSTNAME = "peda0001.database.windows.net";
    final static String DBNAME = "cs-dsa-4513-sql-db";
    final static String USERNAME = "peda0001";
    final static String PASSWORD = "Laddu@2021";
    // Database connection string
    final static String URL =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.w
indows.net;loginTimeout=30;",
        HOSTNAME, DBNAME, USERNAME, PASSWORD);

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

    // Return the result of selecting everything from the customer table
    public ResultSet getAllCustomers() throws SQLException {
        getDBConnection();
        final String sqlQuery = "SELECT * FROM customers;";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }
}
```



```

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

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

        // Return the result of selecting everything from the customer table
in the given range
        public ResultSet retrieveCustomers(int start_range, int end_range)
throws SQLException {
            getDBConnection();
            final String sqlQuery = "SELECT * FROM customers WHERE category
BETWEEN '" + start_range + "' and '" + end_range + "'";
            final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
            return stmt.executeQuery();
        }
    }
}

```

Pedakolimi_Ramya_Sruthi_IP_Task7_get_all_customers

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Customers Data</title>
</head>
<body>
<%@page
import="Pedakolimi_Ramya_Sruthi_IP_Task7_data_handler.DataHandle"%>
<%@page import="java.sql.ResultSet"%>
<%
// We instantiate the data handler1 here,and get all the customers from
the database
final DataHandler1 handler = new DataHandler();
final ResultSet customers = handler.getAllCustomers();
%>
<!-- The table for displaying all the movie records -->
<table cellpadding="2" cellspacing="2" border="1">
<tr> <!-- The table headers row -->
<td align="center">
<h4>Customer Name</h4>
</td>
<td align="center">
<h4>Customer Address</h4>
</td>
<td align="center">
<h4>Category</h4>
</td>
</tr>
<%
while(customers.next()) { // For each customer record returned...
// Extract the attribute values for every row returned
final String customer_name = customers.getString("customer_name");
final String customer_address = customers.getString("customer_address");
final String category = customers.getString("category");
out.println("<tr>");
out.println(

```

```

"<td align=\"center\">" + customer_name +
"</td><td align=\"center\"> " + customer_address +
"</td><td align=\"center\"> " + customer_category +
"</td>");
out.println("</tr>");
}
%>
</table>
</body>
</html>

```

Pedakolimi_Ramya_Sruthi_IP_Task7_add_customer_form

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Register New Customer</title>
</head>
<body>
    <h2>New Customer Form</h2>
    <!--
Form for collecting user input for the new customer record.
Upon form submission, add_customer.jsp file will be invoked.
-->
    <form action="Pedakolimi_Ramya_Sruthi_IP_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>Name of the Customer:</td>
                <td><div style="text-align: center;">
                    <input type="text" name="name">
                </div></td>
            </tr>
            <tr>
                <td>Address:</td>
                <td><div style="text-align: center;">
                    <input type="text" name="address">

```

```

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

```

Pedakolimi_Ramya_Sruthi_IP_Task7_add_customer

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
    <%@page import="Pedakolimi_Ramya_Sruthi_IP_Task7.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>
    <%
        // The handler is the one in charge of establishing the connection.
        DataHandler handler = new DataHandler();
        // Get the attribute values passed from the input form.
        String customer_name = request.getParameter("customer_name");
    %>

```

```

String customer_address = request.getParameter("customer_address");
String categoryString = request.getParameter("category");

if (customer_name.equals("") || customer_address.equals("") ||
categoryString.equals("")) {
    response.sendRedirect("add_customer_form.jsp");
} else {
    int category = Integer.parseInt(categoryString);
    // Now perform the query with the data from the form.
    boolean success = handler.addCustomer(customer_name,
customer_address, category);
    if (!success) { // Something went wrong
        %>
        <h2>There was a problem inserting the course</h2>
        <%
    } else { // Confirm success to the user
        %>
        <h2>The Customer Data Entered:</h2>
        <ul>
            <li>Customer Name: <%=customer_name%></li>
            <li>Customer Address: <%=customer_address%></li>
            <li>Category: <%=categoryString%></li>
        </ul>
        <h2>Was successfully inserted.</h2>
        <a href="Pedakolimi_Ramya_Sruthi_IP_Task7_get_all_customers.jsp">See
all Customers.</a>
        <%
    }
}
%>
</body>
</html>

```

Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers_form

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Retrieval of Customers</title>

```

```

</head>
<body>
    <h2>Given the Customer Range get all the Customers</h2>
    <!--
Form for collecting user input for the new movie_night record.
Upon form submission, add_customer.jsp file will be invoked.
-->
    <form
action="Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers.jsp">
        <!-- The form organized in an HTML table for better clarity. -->
        <table border=1>
            <tr>
                <th colspan="2">Enter the Details of Category Range:</th>
            </tr>
            <tr>
                <td>Ranges From:</td>
                <td><div style="text-align: center;">
                    <input type=text name=min_cat>
                </div></td>
            </tr>
            <tr>
                <td>Ranges To:</td>
                <td><div style="text-align: center;">
                    <input type=text name=max_cat>
                </div></td>
            </tr>
            <tr>
                <td><div style="text-align: center;">
                    <input type=reset value=Clear>
                </div></td>
                <td><div style="text-align: center;">
                    <input type=submit value=search>
                </div></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
    <%@page import="Pedakolimi_Ramya_Sruthi_IP_Task7.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>
    <%
        // The handler is the one in charge of establishing the connection.
        DataHandler handler = new DataHandler();
        // Get the attribute values passed from the input form.
        String rangeFrom = request.getParameter("start_range");
        String rangeTo = request.getParameter("end_range");

        if (start_range.equals("") || end_range.equals("")) {
response.sendRedirect("Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers
_form.jsp");
        } else {
            int duration_from = Integer.parseInt(rangeFrom);
            int duration_to = Integer.parseInt(rangeTo); // Now perform the
query with the data from the form.
            final ResultSet customers =
handler.retrieveCustomers(duration_from, duration_to);
        }
    %>
    <!-- The table for displaying all the movie records -->
    <table cellpadding="2" cellspacing="2" border="1">
        <tr>
            <!-- The table headers row -->
            <td align="center">
                <h4>Customer Name</h4>
            </td>
            <td align="center">
                <h4>Customer Address</h4>
            </td>
        </tr>
    </table>

```

```

        <td align="center">
            <h4>Category</h4>
        </td>
    </tr>
<%
    while (customers.next()) { // For each customerrecord returned...
        // Extract the attribute values for every row
returned
        final String name = customers.getString("customer_name");
        final String address =
customers.getString("customer_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\">" + customer_name + "<td
align=\"center\">" + customer_address + "<td align=\"center\">" + category
            + "</td>");
        out.println("</tr>");
    }
}
%>

</body>
</html>

```

Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
<body>
    <%@page import="Pedakolimi_Ramya_Sruthi_IP_Task7.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>

```



```

<%
// The handler is the one in charge of establishing the connection.
DataHandler handler = new DataHandler();
// Get the attribute values passed from the input form.
String rangeFrom = request.getParameter("start_range");
String rangeTo = request.getParameter("end_range");

if (start_range.equals("") || end_range.equals("")) {

response.sendRedirect("Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers
_form.jsp");
} else {
    int duration_from = Integer.parseInt(rangeFrom);
    int duration_to = Integer.parseInt(rangeTo); // Now perform the
query with the data from the form.
    final ResultSet customers =
handler.retrieveCustomers(duration_from, duration_to);
    %>
    <!-- The table for displaying all the movie records -->
    <table cellpadding="2" cellspacing="2" border="1">
        <tr>
            <!-- The table headers row -->
            <td align="center">
                <h4>Customer Name</h4>
            </td>
            <td align="center">
                <h4>Customer Address</h4>
            </td>
            <td align="center">
                <h4>Category</h4>
            </td>
        </tr>
    <%
    while (customers.next()) {
        final String customer_name =
customers.getString("customer_name");
        final String customer_address =
customers.getString("customer_address");
        final String category = customers.getString("category");
        out.println("<tr>"); // Start printing out the new table row

```

```

        out.println(
            "<td align=\"center\">" + customer_name + "<td
align=\"center\">" + customer_address + "<td align=\"center\">" + category
            + "</td>");
        out.println("</tr>");
    }
}
%>

</body>
</html>

```

Output:

Given the Customer Range get all the Customers

Enter the Details of Category Range	
Range From:	<input type="text"/>
Range To:	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="search"/>

Register New Customer

New Customer Form	
Name of the Customer:	<input type="text" value="Sruthi Pedakolimi"/>
Address:	<input type="text" value="Beaumont Drive"/>
Category:	<input type="text" value="4"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

Customer Name	Customer Address	Category
Ramya	Beaumont Drive	4
Ratan	Newyork	10
Rithwik	California	8
Sruthi Pedakolimi	Beaumont Drive	4

Given the Customer Range get all the Customers

Enter the Details of Category Range	
Range From:	<input type="text" value="1"/>
Range To:	<input type="text" value="6"/>
<input type="button" value="Clear"/>	<input type="button" value="search"/>

Customer Name	Customer Address	Category
Ramya	Beaumont Drive	4
Sruthi Pedakolimi	Beaumont Drive	4

