INDIVIDUAL PROJECT
A JOB-SHOP ACCOUNTING DATABASE SYSTEM

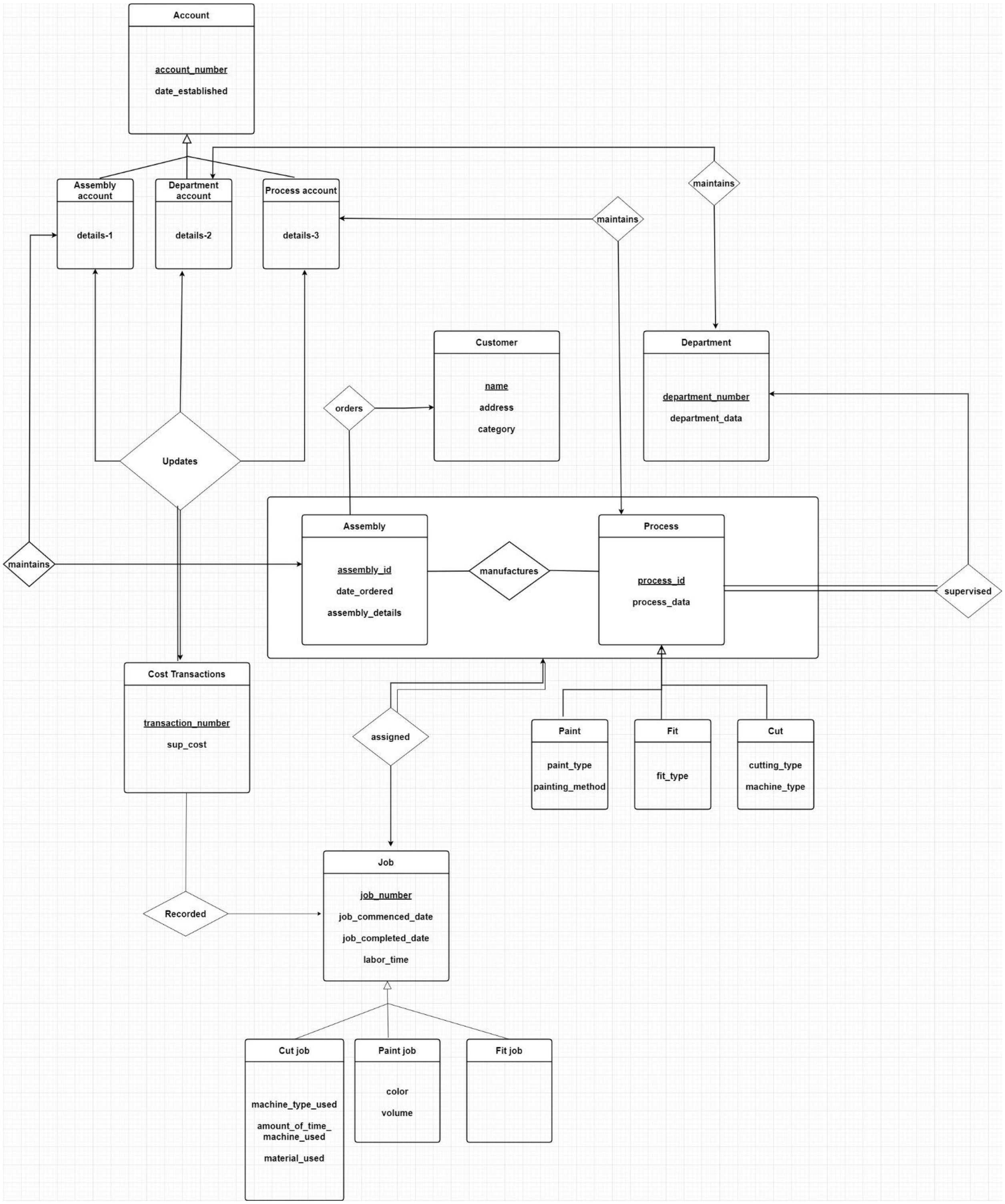
Name: Pedakolimi Ramya Sruthi
Course Name: Database Management Systems (DSA4513-995)
Semester: Fall-2023
Instructor: Dr. Le Gruenwald
Email id: sruthipedakolimi@ou.edu

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 infreaquent
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 bestr
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 bestr
Paint job	14 (update & Random search)	job_number	1/week		

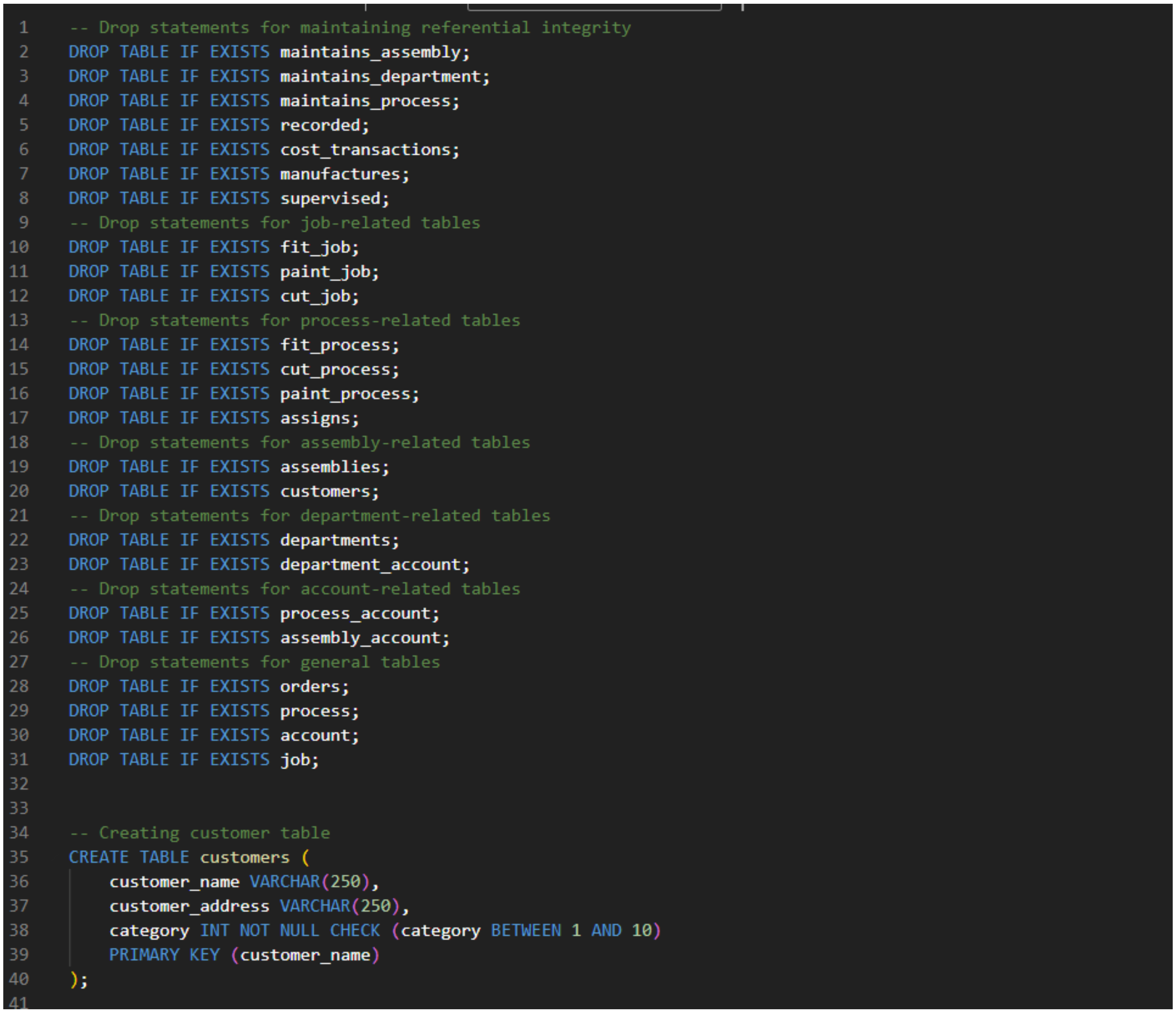
cost_transactions	8 (update & random search)	transaction_number	50/day	B+ Tree	Here we are updating the cost transactions more frequently and we also use both the range and random search so B+ Tree would be better choice
cost_transactions	9 (range search)	account_number	200/day		
recorded	9 (range search)		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/department	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/department	9 (Random search)	assembly_id	200/day		

3.2 : Storage structures for tables (Azure SQL Database)

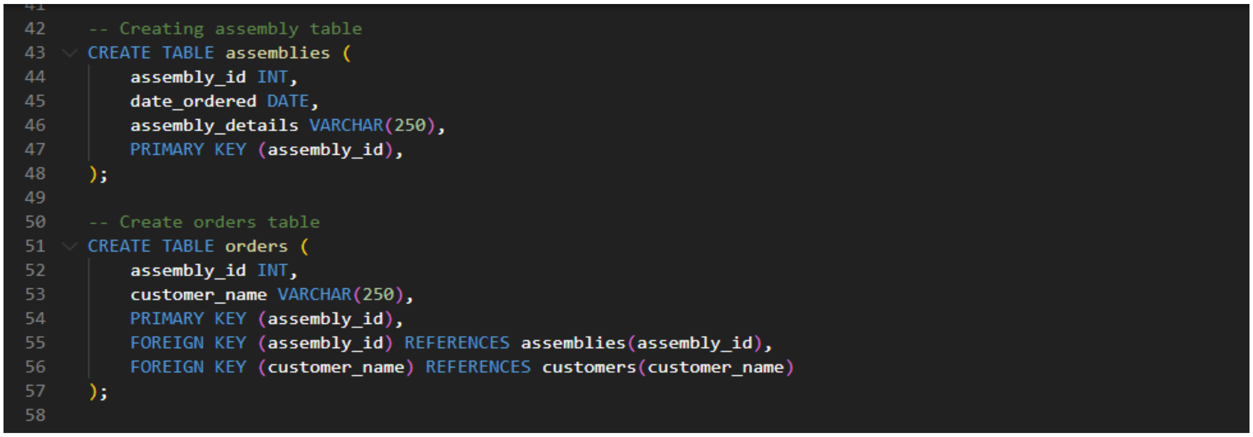
While I intend to utilize the storage structures mentioned above for my Azure SQL Database, it's worth noting that Azure automatically generates an index on the primary key, which proves beneficial for query performance. Specifically clustered Index should be used. we are using the range search for the recorded more frequently. So B+ tree would be bestGiven 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.

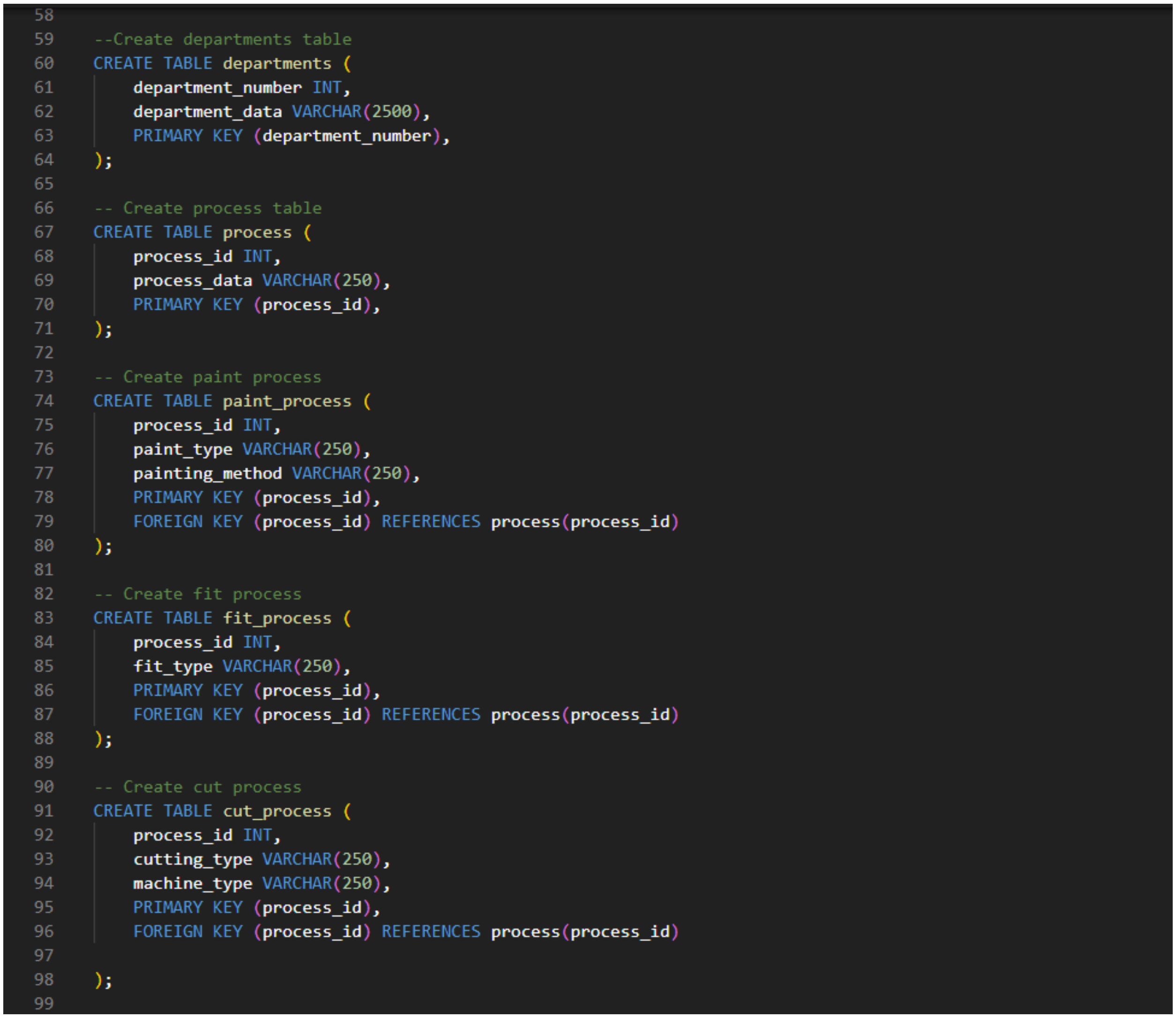
Task 4: SQL statements and screenshots



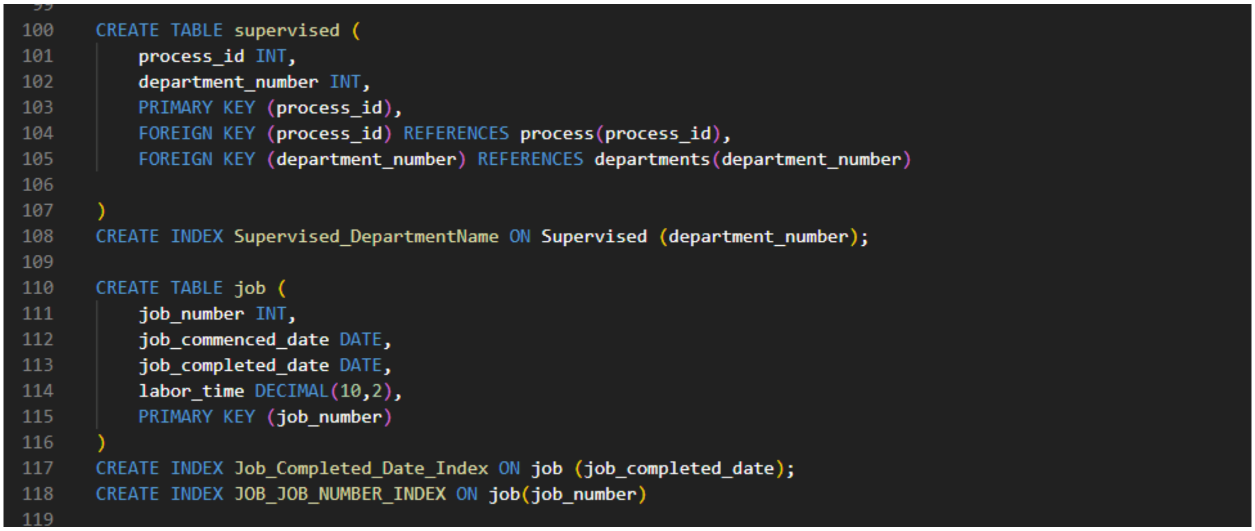
```sql
-- Drop statements for maintaining referential integrity
DROP TABLE IF EXISTS maintains_assembly;
DROP TABLE IF EXISTS maintains_department;
DROP TABLE IF EXISTS maintains_process;
DROP TABLE IF EXISTS recorded;
DROP TABLE IF EXISTS cost_transactions;
DROP TABLE IF EXISTS manufactures;
DROP TABLE IF EXISTS supervised;
-- Drop statements for job-related tables
DROP TABLE IF EXISTS fit_job;
DROP TABLE IF EXISTS paint_job;
DROP TABLE IF EXISTS cut_job;
-- Drop statements for process-related tables
DROP TABLE IF EXISTS fit_process;
DROP TABLE IF EXISTS cut_process;
DROP TABLE IF EXISTS paint_process;
DROP TABLE IF EXISTS assigns;
-- Drop statements for assembly-related tables
DROP TABLE IF EXISTS assemblies;
DROP TABLE IF EXISTS customers;
-- Drop statements for department-related tables
DROP TABLE IF EXISTS departments;
DROP TABLE IF EXISTS department_account;
-- Drop statements for account-related tables
DROP TABLE IF EXISTS process_account;
DROP TABLE IF EXISTS assembly_account;
-- Drop statements for general tables
DROP TABLE IF EXISTS orders;
DROP TABLE IF EXISTS process;
DROP TABLE IF EXISTS account;
DROP TABLE IF EXISTS job;


-- Creating customer table
CREATE TABLE customers (
    customer_name VARCHAR(250),
    customer_address VARCHAR(250),
    category INT NOT NULL CHECK (category BETWEEN 1 AND 10)
    PRIMARY KEY (customer_name)
);

```



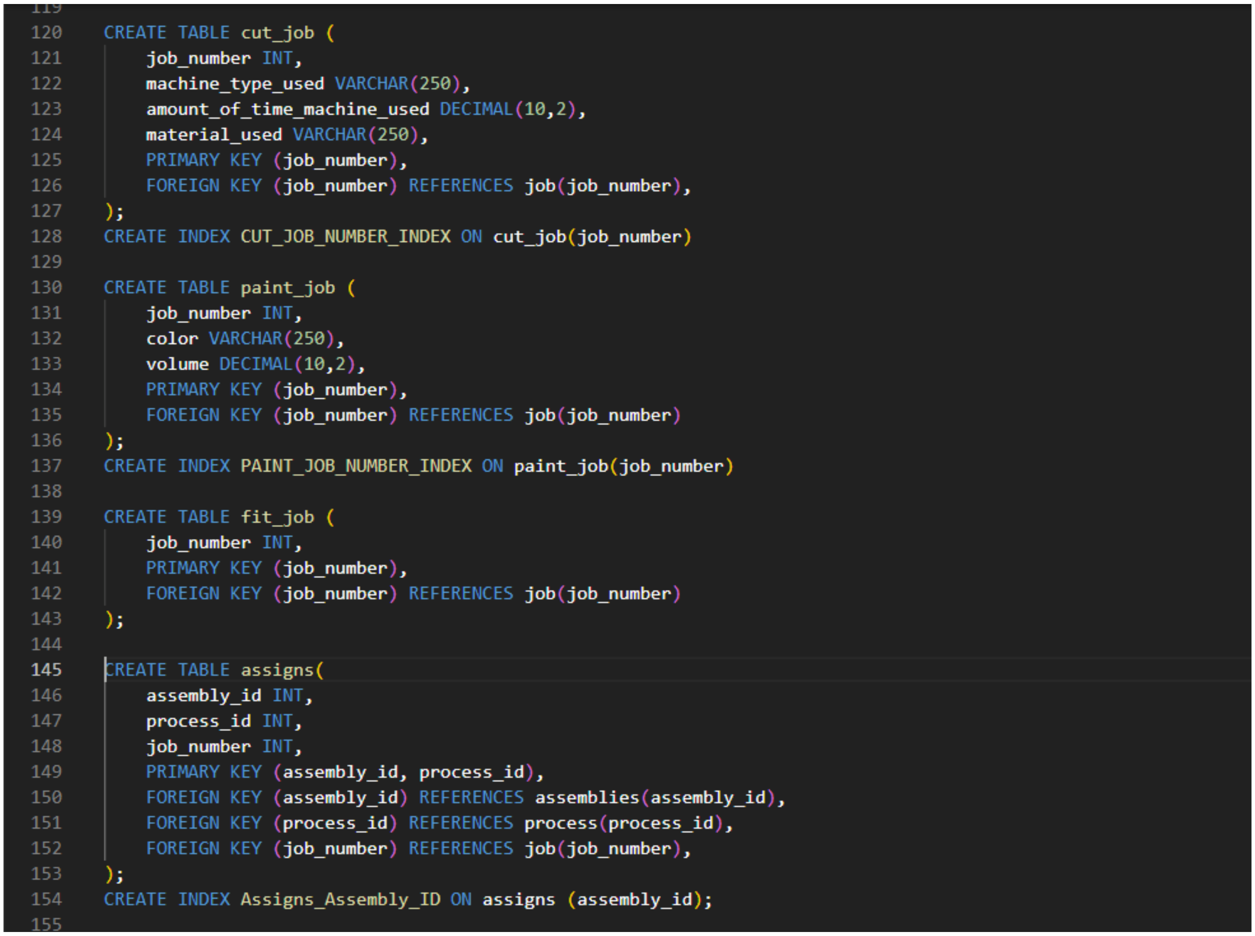
```sql
-- Creating assembly table
CREATE TABLE assemblies (
    assembly_id INT,
    date_ordered DATE,
    assembly_details VARCHAR(250),
    PRIMARY KEY (assembly_id),
);

-- Create orders table
CREATE TABLE orders (
    assembly_id INT,
    customer_name VARCHAR(250),
    PRIMARY KEY (assembly_id),
    FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
    FOREIGN KEY (customer_name) REFERENCES customers(customer_name)
);

```



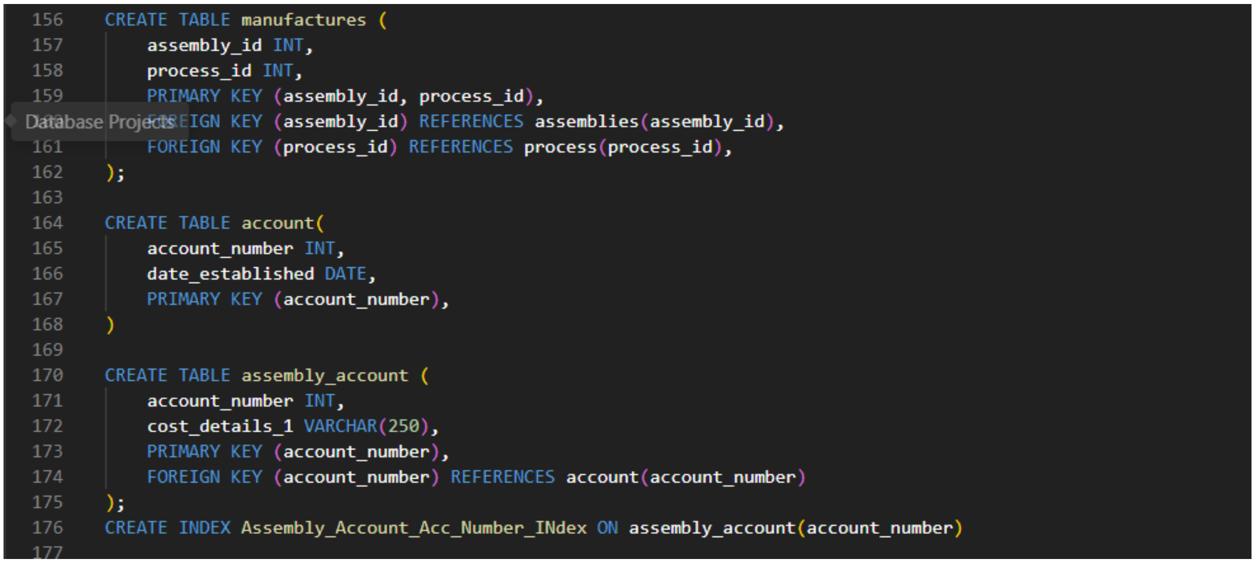
```
--Create departments table
CREATE TABLE departments (
    department_number INT,
    department_data VARCHAR(2500),
    PRIMARY KEY (department_number),
);

-- Create process table
CREATE TABLE process (
    process_id INT,
    process_data VARCHAR(250),
    PRIMARY KEY (process_id),
);

-- Create paint process
CREATE TABLE paint_process (
    process_id INT,
    paint_type VARCHAR(250),
    painting_method VARCHAR(250),
    PRIMARY KEY (process_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id)
);

-- Create fit process
CREATE TABLE fit_process (
    process_id INT,
    fit_type VARCHAR(250),
    PRIMARY KEY (process_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id)
);

-- Create cut process
CREATE TABLE cut_process (
    process_id INT,
    cutting_type VARCHAR(250),
    machine_type VARCHAR(250),
    PRIMARY KEY (process_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id)

);
```



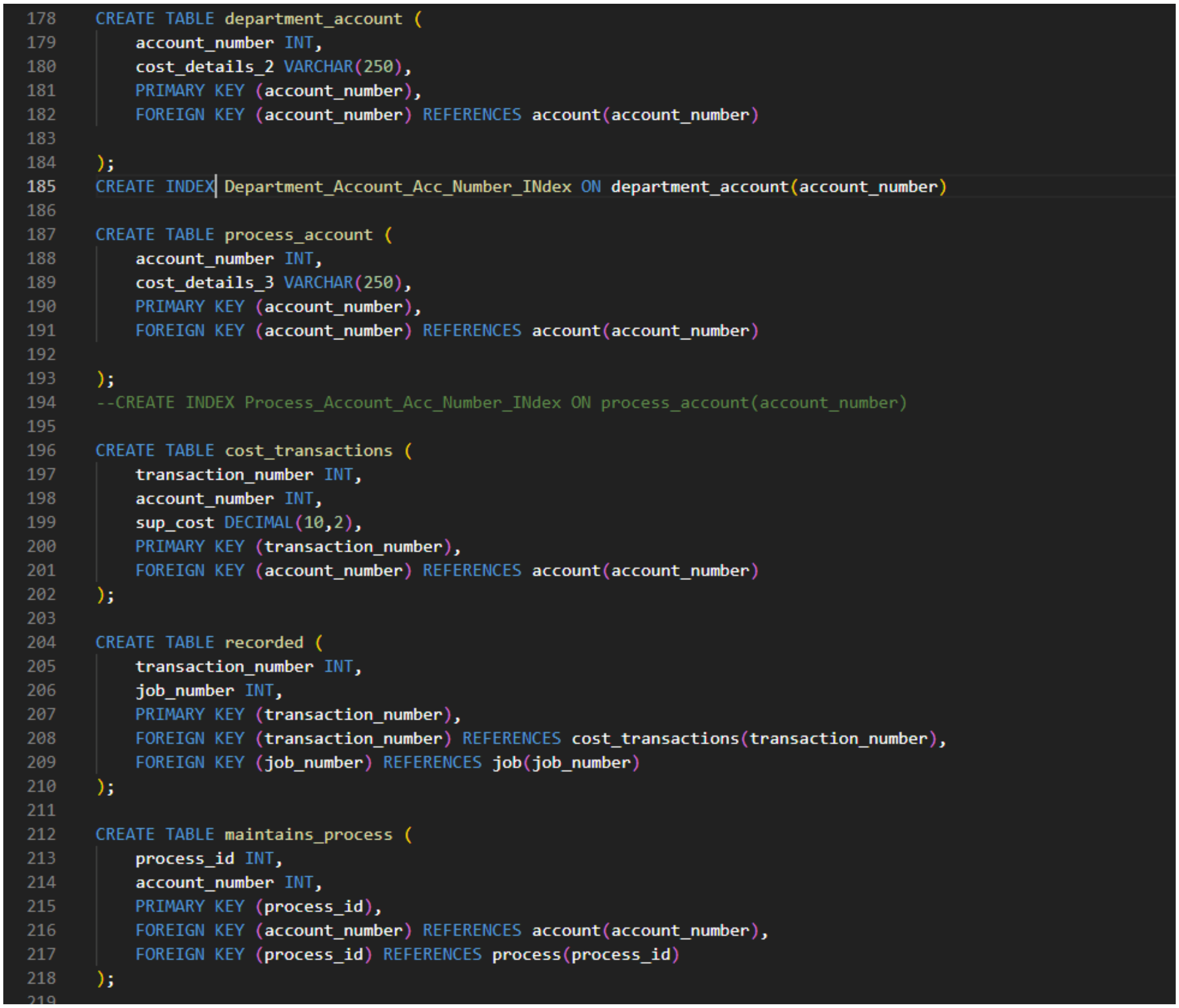
```
CREATE TABLE supervised (
    process_id INT,
    department_number INT,
    PRIMARY KEY (process_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id),
    FOREIGN KEY (department_number) REFERENCES departments(department_number)

)
CREATE INDEX Supervised_DepartmentName ON Supervised (department_number);

CREATE TABLE job (
    job_number INT,
    job_commenced_date DATE,
    job_completed_date DATE,
    labor_time DECIMAL(10,2),
    PRIMARY KEY (job_number)
)
CREATE INDEX Job_Completed_Date_Index ON job (job_completed_date);
CREATE INDEX JOB_JOB_NUMBER_INDEX ON job(job_number)
```



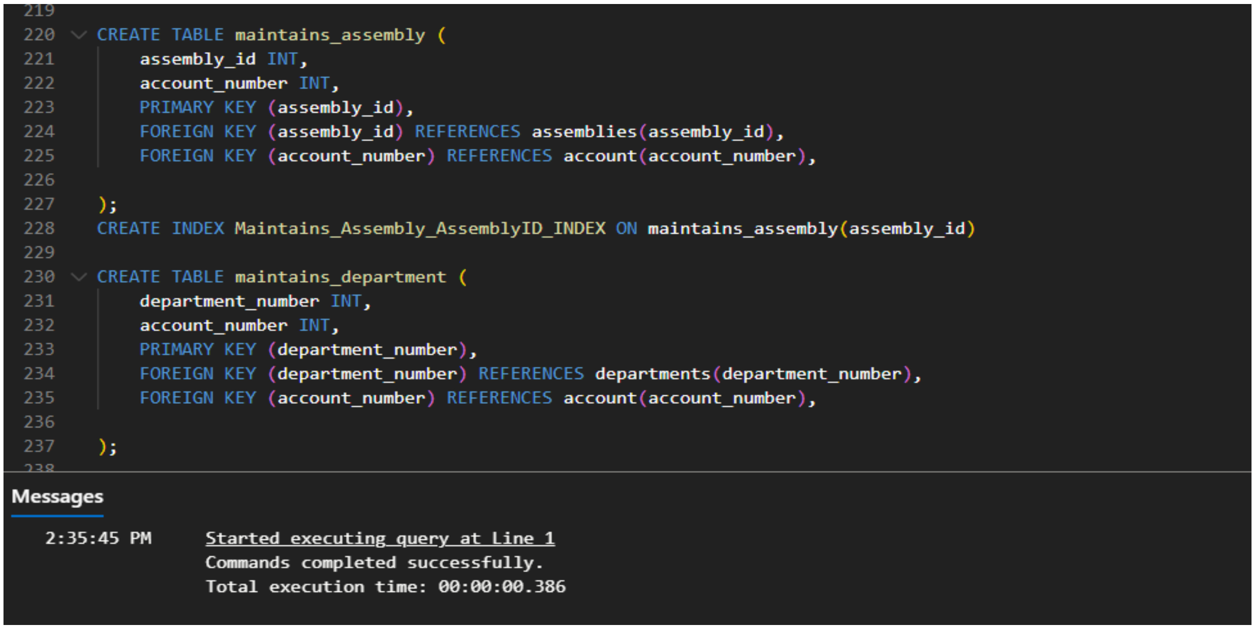
```sql
CREATE TABLE cut_job (
    job_number INT,
    machine_type_used VARCHAR(250),
    amount_of_time_machine_used DECIMAL(10,2),
    material_used VARCHAR(250),
    PRIMARY KEY (job_number),
    FOREIGN KEY (job_number) REFERENCES job(job_number),
);
CREATE INDEX CUT_JOB_NUMBER_INDEX ON cut_job(job_number)

CREATE TABLE paint_job (
    job_number INT,
    color VARCHAR(250),
    volume DECIMAL(10,2),
    PRIMARY KEY (job_number),
    FOREIGN KEY (job_number) REFERENCES job(job_number)
);
CREATE INDEX PAINT_JOB_NUMBER_INDEX ON paint_job(job_number)

CREATE TABLE fit_job (
    job_number INT,
    PRIMARY KEY (job_number),
    FOREIGN KEY (job_number) REFERENCES job(job_number)
);

CREATE TABLE assigns(
    assembly_id INT,
    process_id INT,
    job_number INT,
    PRIMARY KEY (assembly_id, process_id),
    FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id),
    FOREIGN KEY (job_number) REFERENCES job(job_number),
);
CREATE INDEX Assigns_Assembly_ID ON assigns (assembly_id);
```



```sql
CREATE TABLE manufactures (
    assembly_id INT,
    process_id INT,
    PRIMARY KEY (assembly_id, process_id),
    FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
    FOREIGN KEY (process_id) REFERENCES process(process_id),
);

CREATE TABLE account(
    account_number INT,
    date_established DATE,
    PRIMARY KEY (account_number),
)

CREATE TABLE assembly_account (
    account_number INT,
    cost_details_1 VARCHAR(250),
    PRIMARY KEY (account_number),
    FOREIGN KEY (account_number) REFERENCES account(account_number)
);
CREATE INDEX Assembly_Account_Acc_Number_INdex ON assembly_account(account_number)
```



```
CREATE TABLE department_account (
    account_number INT,
    cost_details_2 VARCHAR(250),
    PRIMARY KEY (account_number),
    FOREIGN KEY (account_number) REFERENCES account(account_number)

);
CREATE INDEX Department_Account_Acc_Number_INdex ON department_account(account_number)

CREATE TABLE process_account (
    account_number INT,
    cost_details_3 VARCHAR(250),
    PRIMARY KEY (account_number),
    FOREIGN KEY (account_number) REFERENCES account(account_number)

);
--CREATE INDEX Process_Account_Acc_Number_INdex ON process_account(account_number)

CREATE TABLE cost_transactions (
    transaction_number INT,
    account_number INT,
    sup_cost DECIMAL(10,2),
    PRIMARY KEY (transaction_number),
    FOREIGN KEY (account_number) REFERENCES account(account_number)
);

CREATE TABLE recorded (
    transaction_number INT,
    job_number INT,
    PRIMARY KEY (transaction_number),
    FOREIGN KEY (transaction_number) REFERENCES cost_transactions(transaction_number),
    FOREIGN KEY (job_number) REFERENCES job(job_number)
);

CREATE TABLE maintains_process (
    process_id INT,
    account_number INT,
    PRIMARY KEY (process_id),
    FOREIGN KEY (account_number) REFERENCES account(account_number),
    FOREIGN KEY (process_id) REFERENCES process(process_id)
);
```



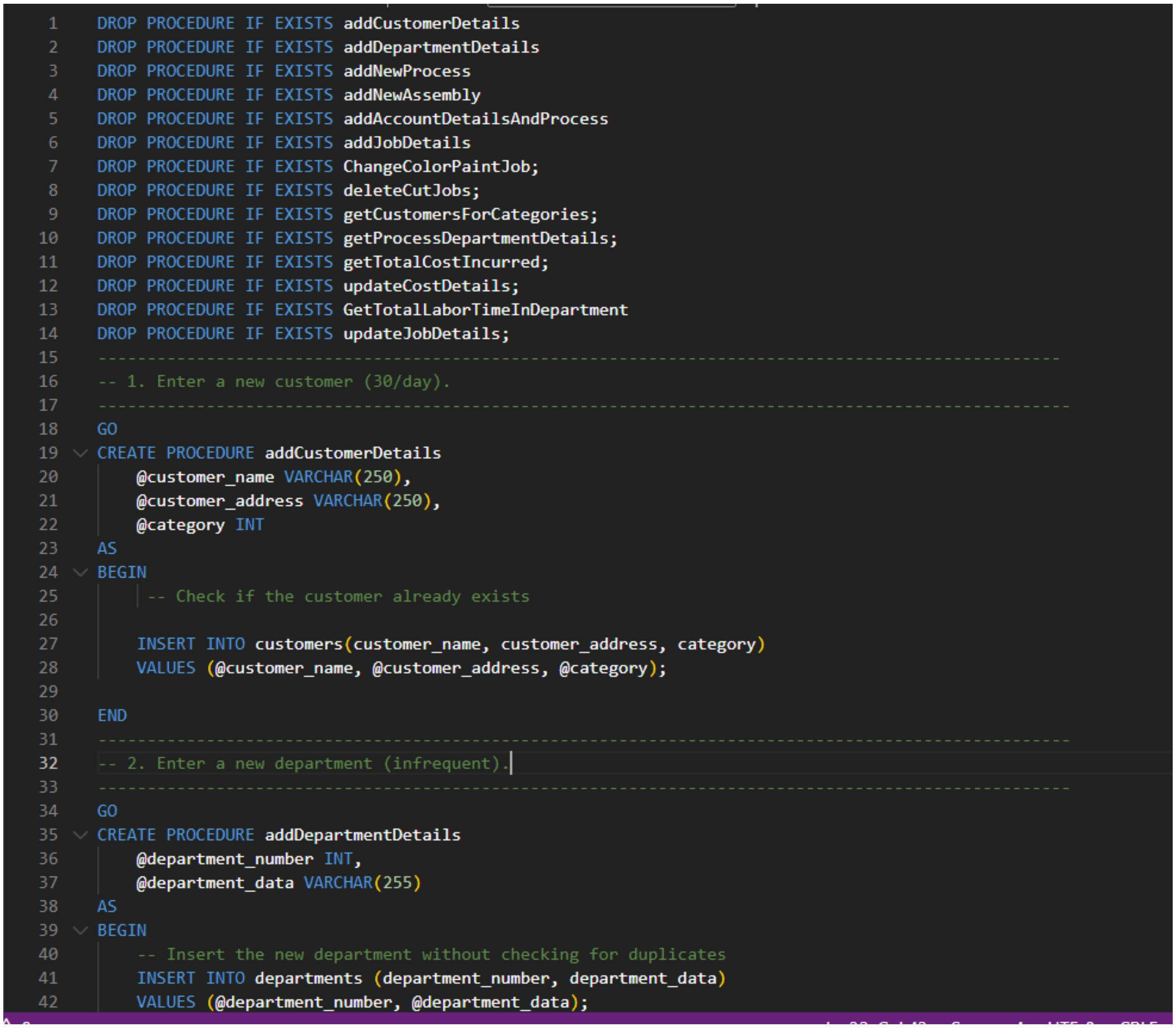
```
CREATE TABLE maintains_assembly (
    assembly_id INT,
    account_number INT,
    PRIMARY KEY (assembly_id),
    FOREIGN KEY (assembly_id) REFERENCES assemblies(assembly_id),
    FOREIGN KEY (account_number) REFERENCES account(account_number),

);
CREATE INDEX Maintains_Assembly_AssemblyID_INDEX ON maintains_assembly(assembly_id)

CREATE TABLE maintains_department (
    department_number INT,
    account_number INT,
    PRIMARY KEY (department_number),
    FOREIGN KEY (department_number) REFERENCES departments(department_number),
    FOREIGN KEY (account_number) REFERENCES account(account_number),

);
```

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

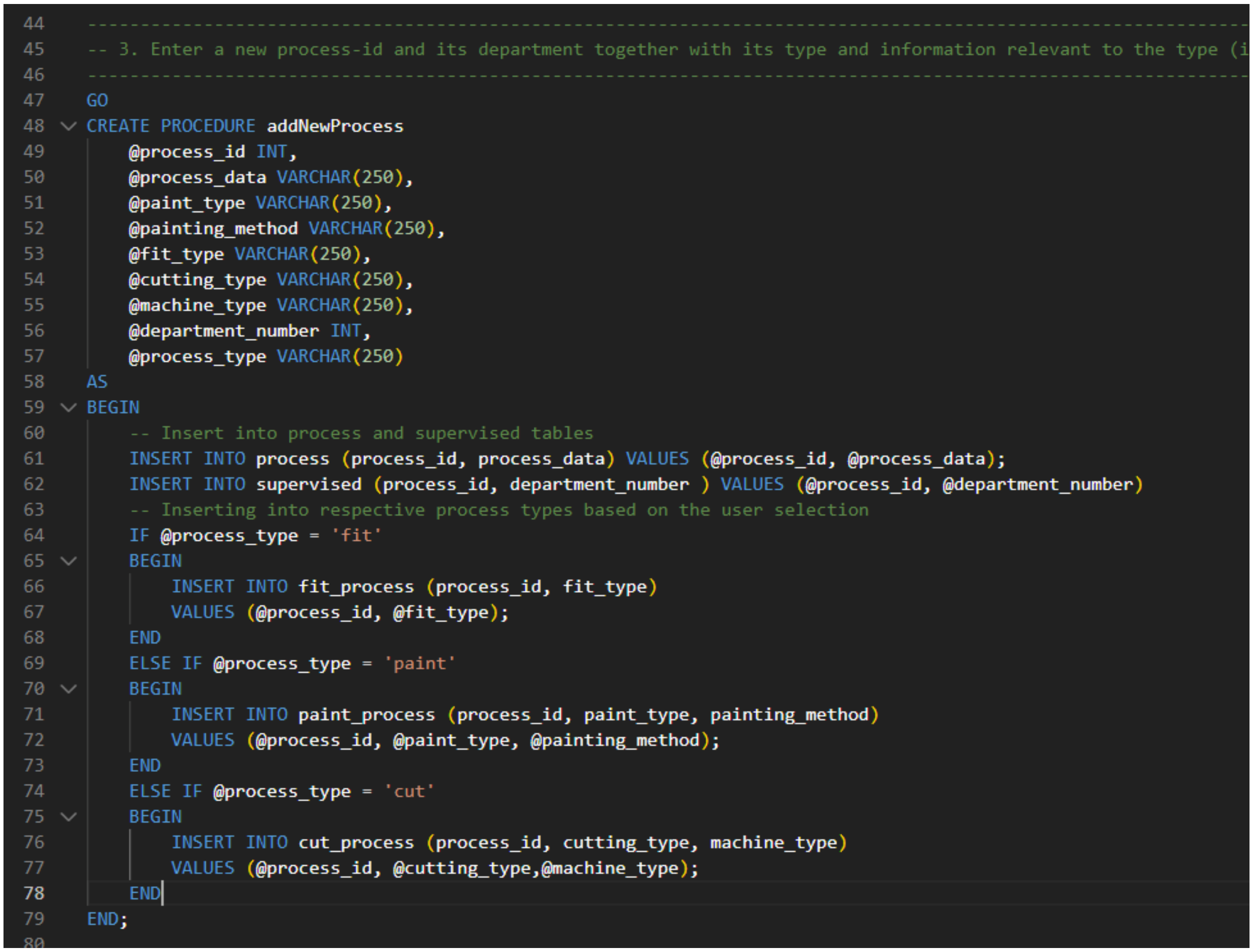
Task 5

5.1: SQL Statements (Stored Procedures) and Error Handling

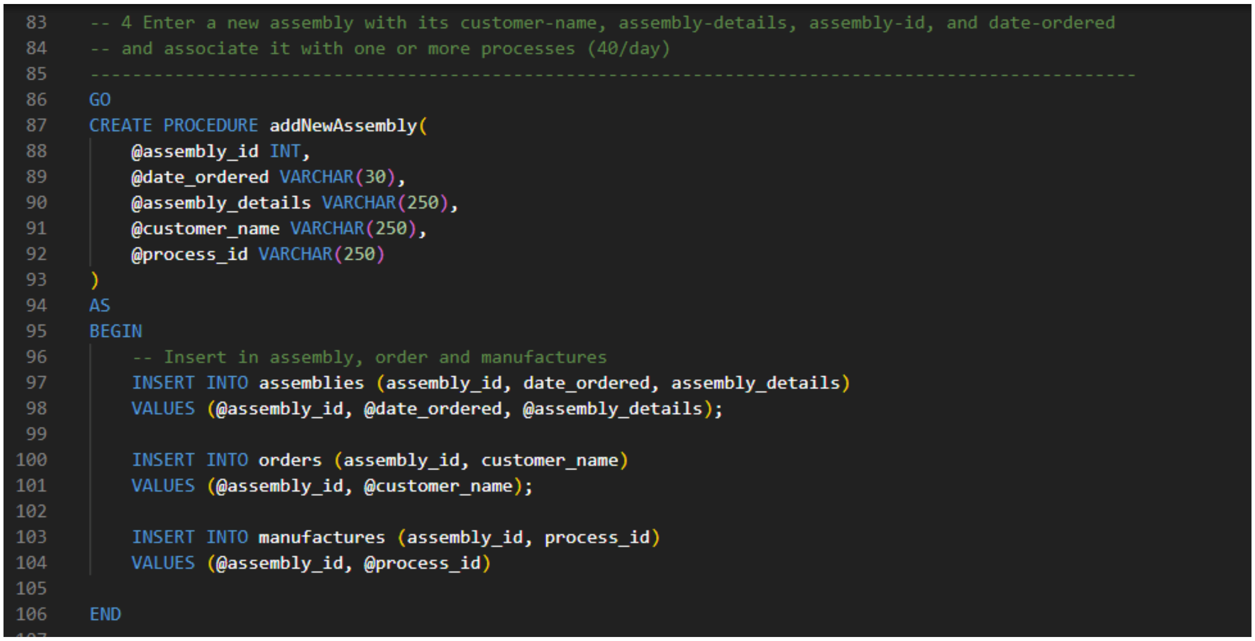
SQL Statements (Stored Procedures)



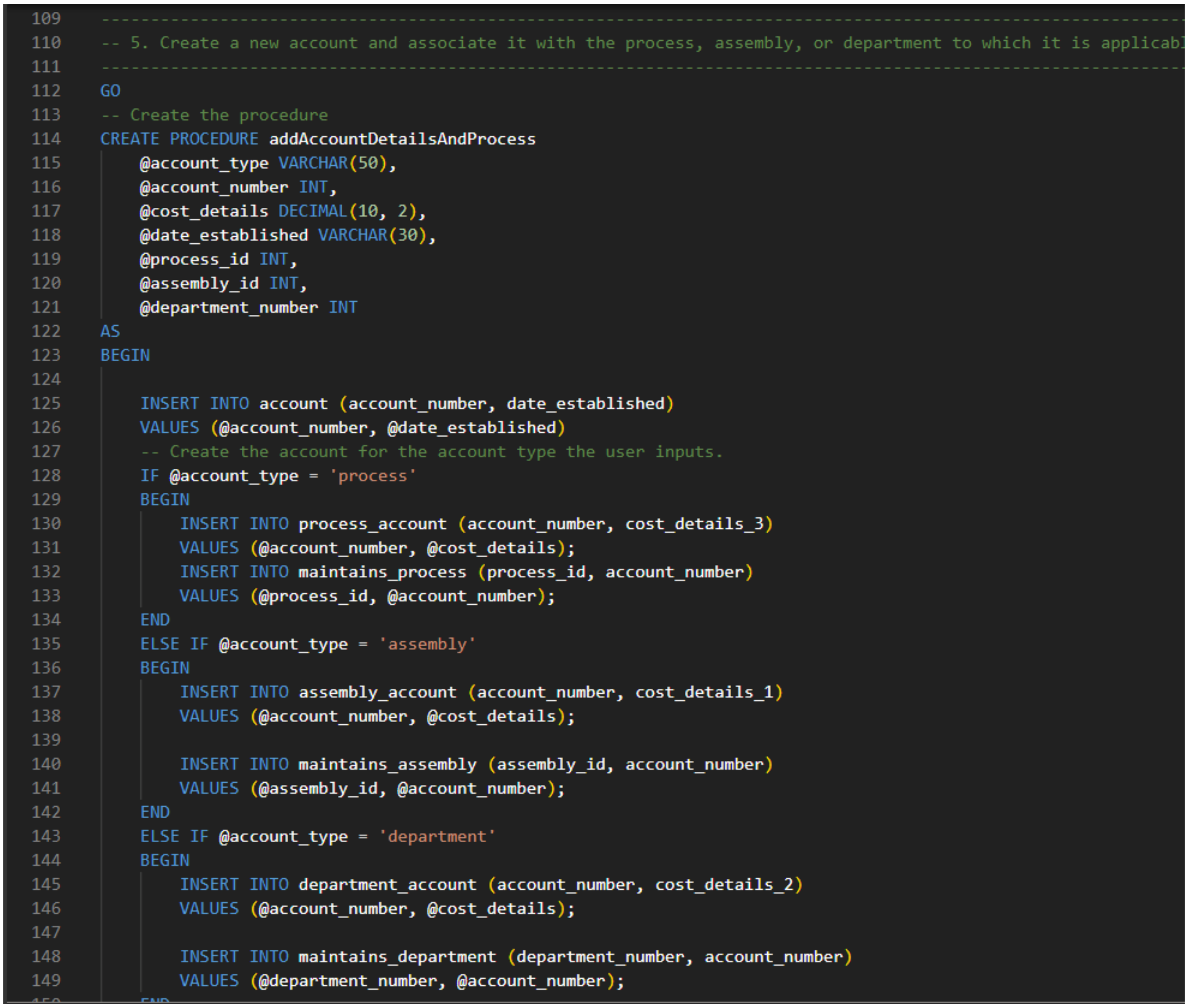
```sql
DROP PROCEDURE IF EXISTS addCustomerDetails
DROP PROCEDURE IF EXISTS addDepartmentDetails
DROP PROCEDURE IF EXISTS addNewProcess
DROP PROCEDURE IF EXISTS addNewAssembly
DROP PROCEDURE IF EXISTS addAccountDetailsAndProcess
DROP PROCEDURE IF EXISTS addJobDetails
DROP PROCEDURE IF EXISTS ChangeColorPaintJob;
DROP PROCEDURE IF EXISTS deleteCutJobs;
DROP PROCEDURE IF EXISTS getCustomersForCategories;
DROP PROCEDURE IF EXISTS getProcessDepartmentDetails;
DROP PROCEDURE IF EXISTS getTotalCostIncurred;
DROP PROCEDURE IF EXISTS updateCostDetails;
DROP PROCEDURE IF EXISTS GetTotalLaborTimeInDepartment
DROP PROCEDURE IF EXISTS updateJobDetails;
-------------------------------------------------------------------------------------------------
-- 1. Enter a new customer (30/day).
--------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE addCustomerDetails
    @customer_name VARCHAR(250),
    @customer_address VARCHAR(250),
    @category INT
AS
BEGIN
     -- Check if the customer already exists

    INSERT INTO customers(customer_name, customer_address, category)
    VALUES (@customer_name, @customer_address, @category);

END
--------------------------------------------------------------------------------------------------
-- 2. Enter a new department (infrequent).
--------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE addDepartmentDetails
    @department_number INT,
    @department_data VARCHAR(255)
AS
BEGIN
    -- Insert the new department without checking for duplicates
    INSERT INTO departments (department_number, department_data)
    VALUES (@department_number, @department_data);
```



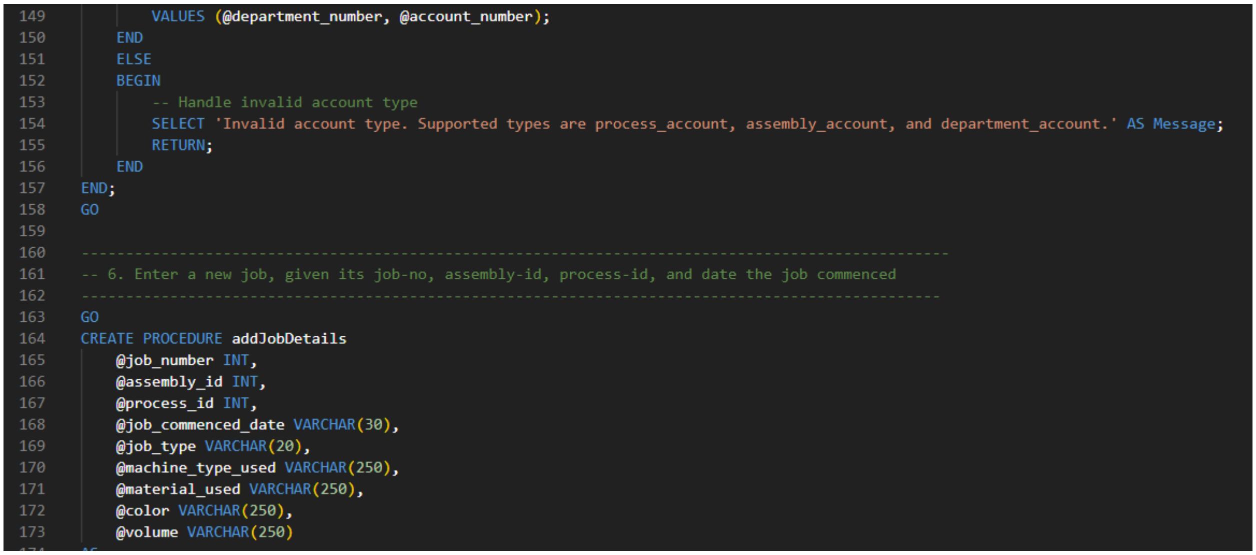
```
-----------------------------------------------------------------------------------------------------------------------
-- 3. Enter a new process-id and its department together with its type and information relevant to the type (i
-----------------------------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE addNewProcess
    @process_id INT,
    @process_data VARCHAR(250),
    @paint_type VARCHAR(250),
    @painting_method VARCHAR(250),
    @fit_type VARCHAR(250),
    @cutting_type VARCHAR(250),
    @machine_type VARCHAR(250),
    @department_number INT,
    @process_type VARCHAR(250)
AS
BEGIN
    -- Insert into process and supervised tables
    INSERT INTO process (process_id, process_data) VALUES (@process_id, @process_data);
    INSERT INTO supervised (process_id, department_number ) VALUES (@process_id, @department_number)
    -- Inserting into respective process types based on the user selection
    IF @process_type = 'fit'
    BEGIN
        INSERT INTO fit_process (process_id, fit_type)
        VALUES (@process_id, @fit_type);
    END
    ELSE IF @process_type = 'paint'
    BEGIN
        INSERT INTO paint_process (process_id, paint_type, painting_method)
        VALUES (@process_id, @paint_type, @painting_method);
    END
    ELSE IF @process_type = 'cut'
    BEGIN
        INSERT INTO cut_process (process_id, cutting_type, machine_type)
        VALUES (@process_id, @cutting_type,@machine_type);
    END
END;
```



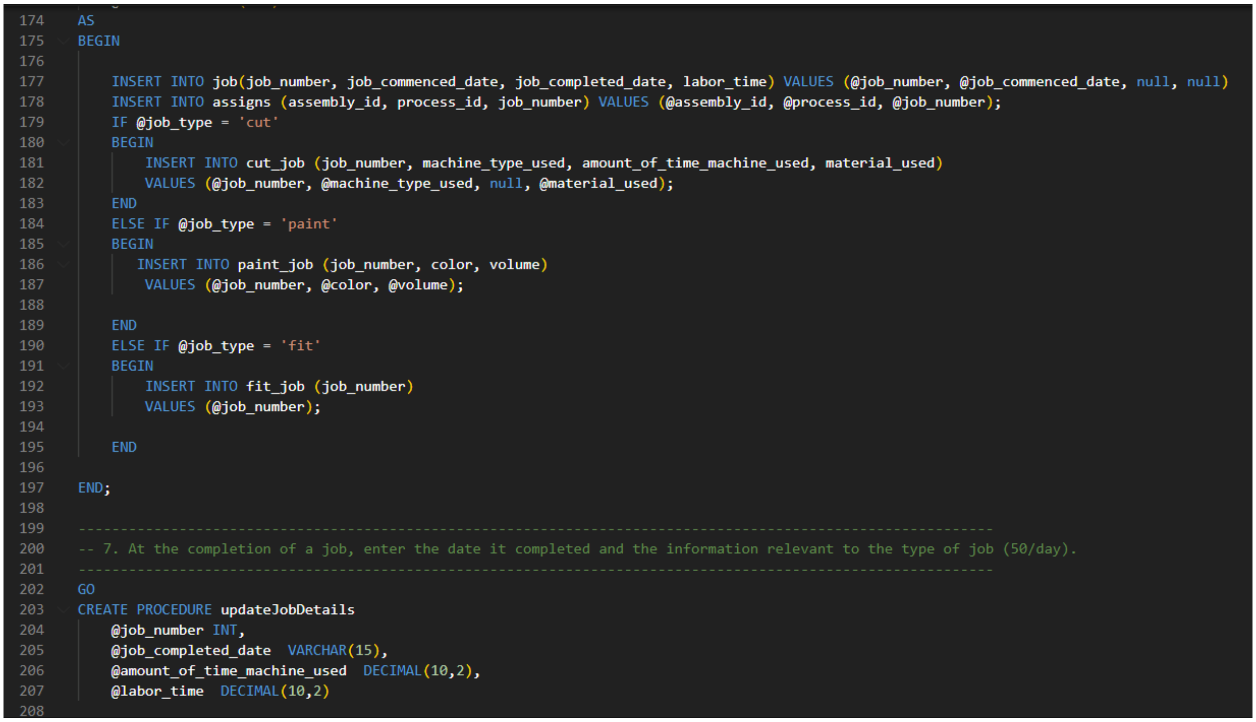
```
-- 4 Enter a new assembly with its customer-name, assembly-details, assembly-id, and date-ordered
-- and associate it with one or more processes (40/day)
----------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE addNewAssembly(
    @assembly_id INT,
    @date_ordered VARCHAR(30),
    @assembly_details VARCHAR(250),
    @customer_name VARCHAR(250),
    @process_id VARCHAR(250)
)
AS
BEGIN
    -- Insert in assembly, order and manufactures
    INSERT INTO assemblies (assembly_id, date_ordered, assembly_details)
    VALUES (@assembly_id, @date_ordered, @assembly_details);

    INSERT INTO orders (assembly_id, customer_name)
    VALUES (@assembly_id, @customer_name);

    INSERT INTO manufactures (assembly_id, process_id)
    VALUES (@assembly_id, @process_id)

END
```



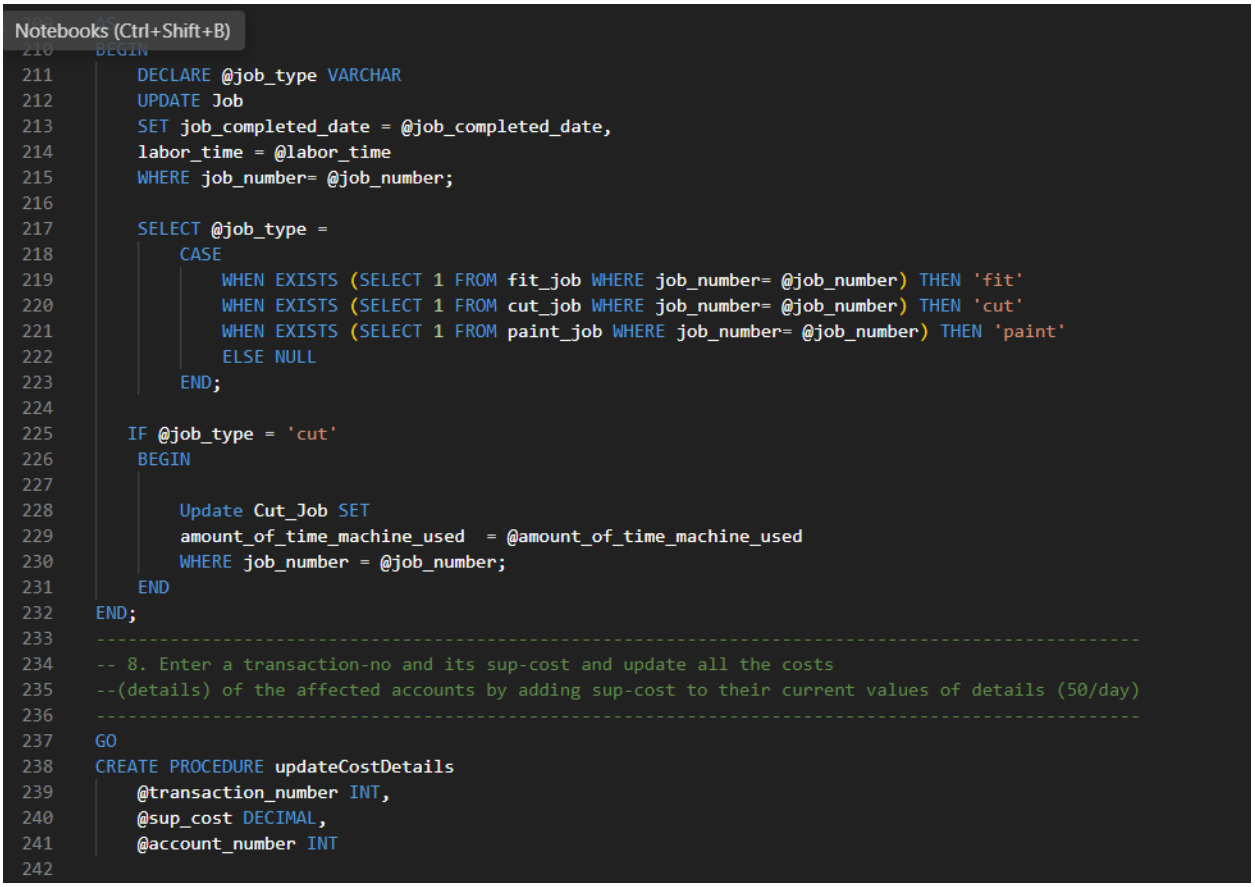
```
-----------------------------------------------------------------------------------------------------------
-- 5. Create a new account and associate it with the process, assembly, or department to which it is applicabl
-----------------------------------------------------------------------------------------------------------
GO
-- Create the procedure
CREATE PROCEDURE addAccountDetailsAndProcess
    @account_type VARCHAR(50),
    @account_number INT,
    @cost_details DECIMAL(10, 2),
    @date_established VARCHAR(30),
    @process_id INT,
    @assembly_id INT,
    @department_number INT
AS
BEGIN

    INSERT INTO account (account_number, date_established)
    VALUES (@account_number, @date_established)
    -- Create the account for the account type the user inputs.
    IF @account_type = 'process'
    BEGIN
        INSERT INTO process_account (account_number, cost_details_3)
        VALUES (@account_number, @cost_details);
        INSERT INTO maintains_process (process_id, account_number)
        VALUES (@process_id, @account_number);
    END
    ELSE IF @account_type = 'assembly'
    BEGIN
        INSERT INTO assembly_account (account_number, cost_details_1)
        VALUES (@account_number, @cost_details);

        INSERT INTO maintains_assembly (assembly_id, account_number)
        VALUES (@assembly_id, @account_number);
    END
    ELSE IF @account_type = 'department'
    BEGIN
        INSERT INTO department_account (account_number, cost_details_2)
        VALUES (@account_number, @cost_details);

        INSERT INTO maintains_department (department_number, account_number)
        VALUES (@department_number, @account_number);
```



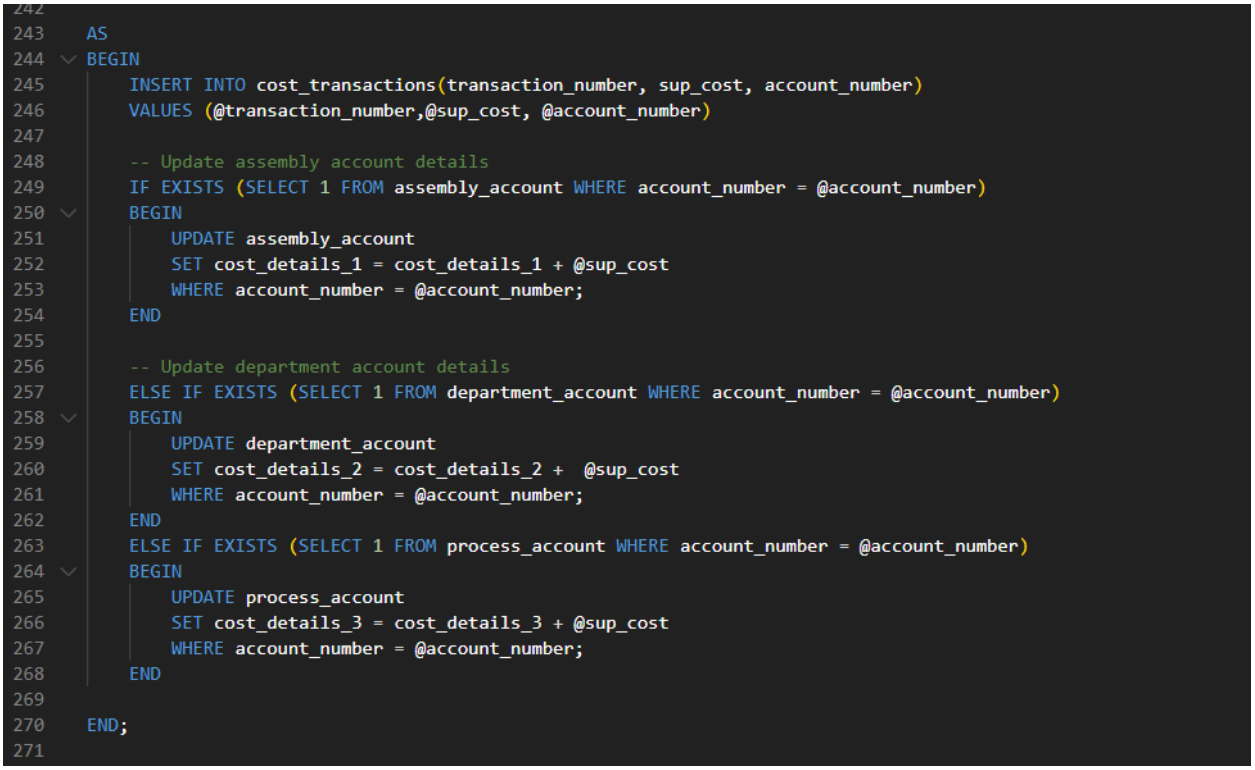
```
        VALUES (@department_number, @account_number);
    END
    ELSE
    BEGIN
        -- Handle invalid account type
        SELECT 'Invalid account type. Supported types are process_account, assembly_account, and department_account.' AS Message;
        RETURN;
    END
END;
GO

-------------------------------------------------------------------------------------------------
-- 6. Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
-------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE addJobDetails
    @job_number INT,
    @assembly_id INT,
    @process_id INT,
    @job_commenced_date VARCHAR(30),
    @job_type VARCHAR(20),
    @machine_type_used VARCHAR(250),
    @material_used VARCHAR(250),
    @color VARCHAR(250),
    @volume VARCHAR(250)
```



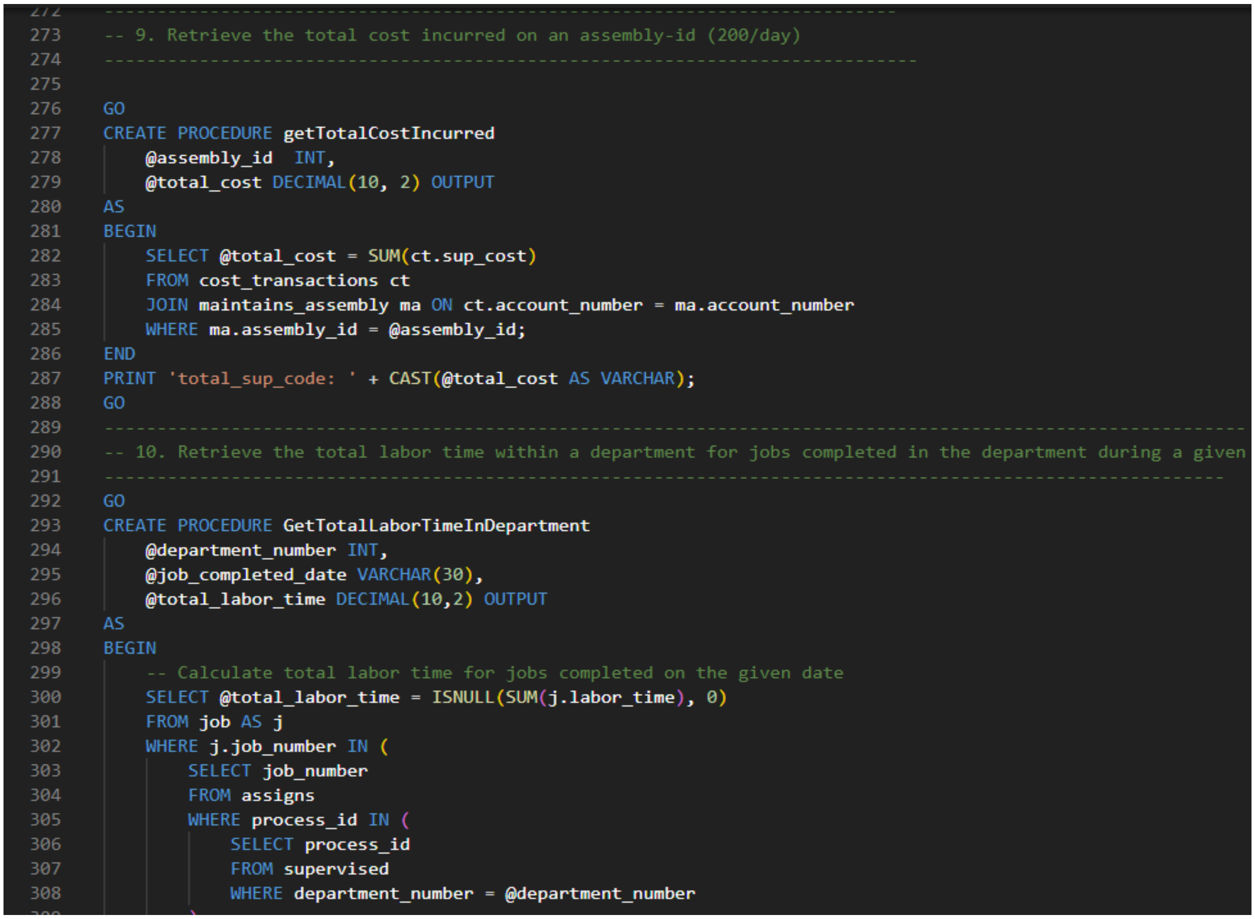
```
AS
BEGIN

    INSERT INTO job(job_number, job_commenced_date, job_completed_date, labor_time) VALUES (@job_number, @job_commenced_date, null, null)
    INSERT INTO assigns (assembly_id, process_id, job_number) VALUES (@assembly_id, @process_id, @job_number);
    IF @job_type = 'cut'
    BEGIN
        INSERT INTO cut_job (job_number, machine_type_used, amount_of_time_machine_used, material_used)
        VALUES (@job_number, @machine_type_used, null, @material_used);
    END
    ELSE IF @job_type = 'paint'
    BEGIN
       INSERT INTO paint_job (job_number, color, volume)
        VALUES (@job_number, @color, @volume);

    END
    ELSE IF @job_type = 'fit'
    BEGIN
        INSERT INTO fit_job (job_number)
        VALUES (@job_number);

    END

END;

---------------------------------------------------------------------------------------------------------------
-- 7. At the completion of a job, enter the date it completed and the information relevant to the type of job (50/day).
---------------------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE updateJobDetails
    @job_number INT,
    @job_completed_date  VARCHAR(15),
    @amount_of_time_machine_used  DECIMAL(10,2),
    @labor_time  DECIMAL(10,2)

```



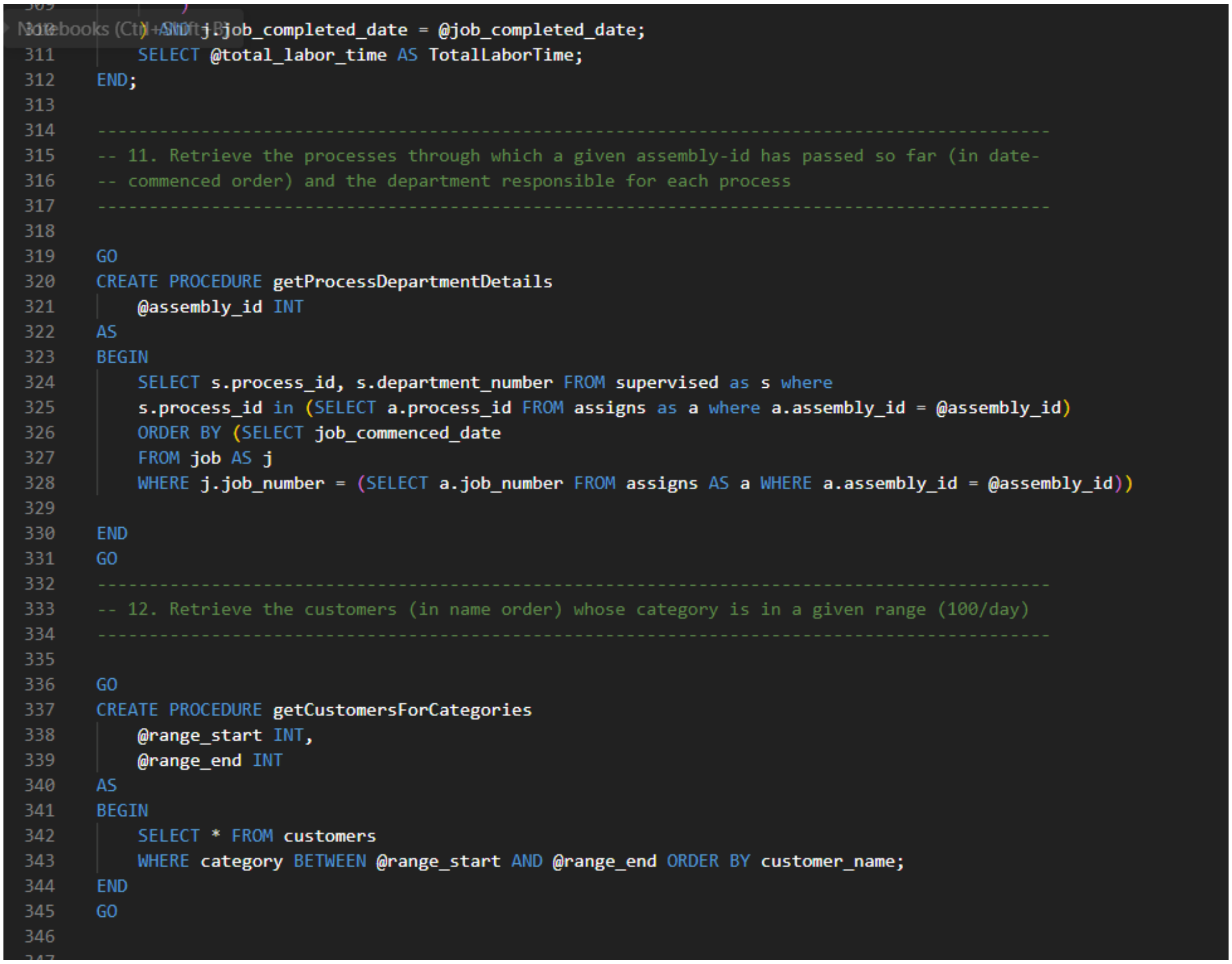
```
BEGIN
    DECLARE @job_type VARCHAR
    UPDATE Job
    SET job_completed_date = @job_completed_date,
    labor_time = @labor_time
    WHERE job_number= @job_number;

    SELECT @job_type =
        CASE
            WHEN EXISTS (SELECT 1 FROM fit_job WHERE job_number= @job_number) THEN 'fit'
            WHEN EXISTS (SELECT 1 FROM cut_job WHERE job_number= @job_number) THEN 'cut'
            WHEN EXISTS (SELECT 1 FROM paint_job WHERE job_number= @job_number) THEN 'paint'
            ELSE NULL
        END;

   IF @job_type = 'cut'
    BEGIN

        Update Cut_Job SET
        amount_of_time_machine_used  = @amount_of_time_machine_used
        WHERE job_number = @job_number;
    END
END;
----------------------------------------------------------------------------------------------------
-- 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 (50/day)
----------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE updateCostDetails
    @transaction_number INT,
    @sup_cost DECIMAL,
    @account_number INT

```



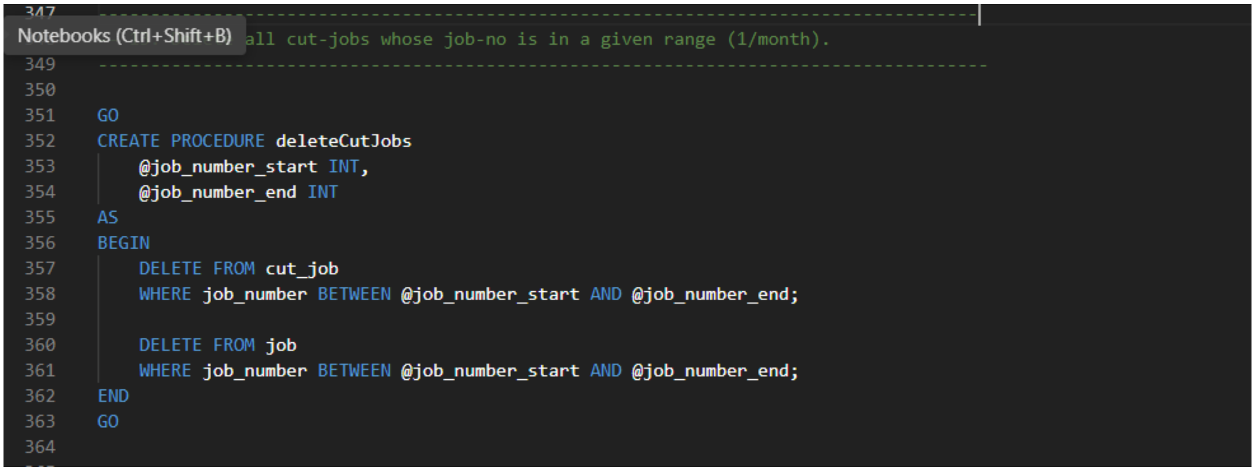
```sql
AS
BEGIN
    INSERT INTO cost_transactions(transaction_number, sup_cost, account_number)
    VALUES (@transaction_number,@sup_cost, @account_number)

    -- Update assembly account details
    IF EXISTS (SELECT 1 FROM assembly_account WHERE account_number = @account_number)
    BEGIN
        UPDATE assembly_account
        SET cost_details_1 = cost_details_1 + @sup_cost
        WHERE account_number = @account_number;
    END

    -- Update department account details
    ELSE IF EXISTS (SELECT 1 FROM department_account WHERE account_number = @account_number)
    BEGIN
        UPDATE department_account
        SET cost_details_2 = cost_details_2 +  @sup_cost
        WHERE account_number = @account_number;
    END
    ELSE IF EXISTS (SELECT 1 FROM process_account WHERE account_number = @account_number)
    BEGIN
        UPDATE process_account
        SET cost_details_3 = cost_details_3 + @sup_cost
        WHERE account_number = @account_number;
    END

END;

```



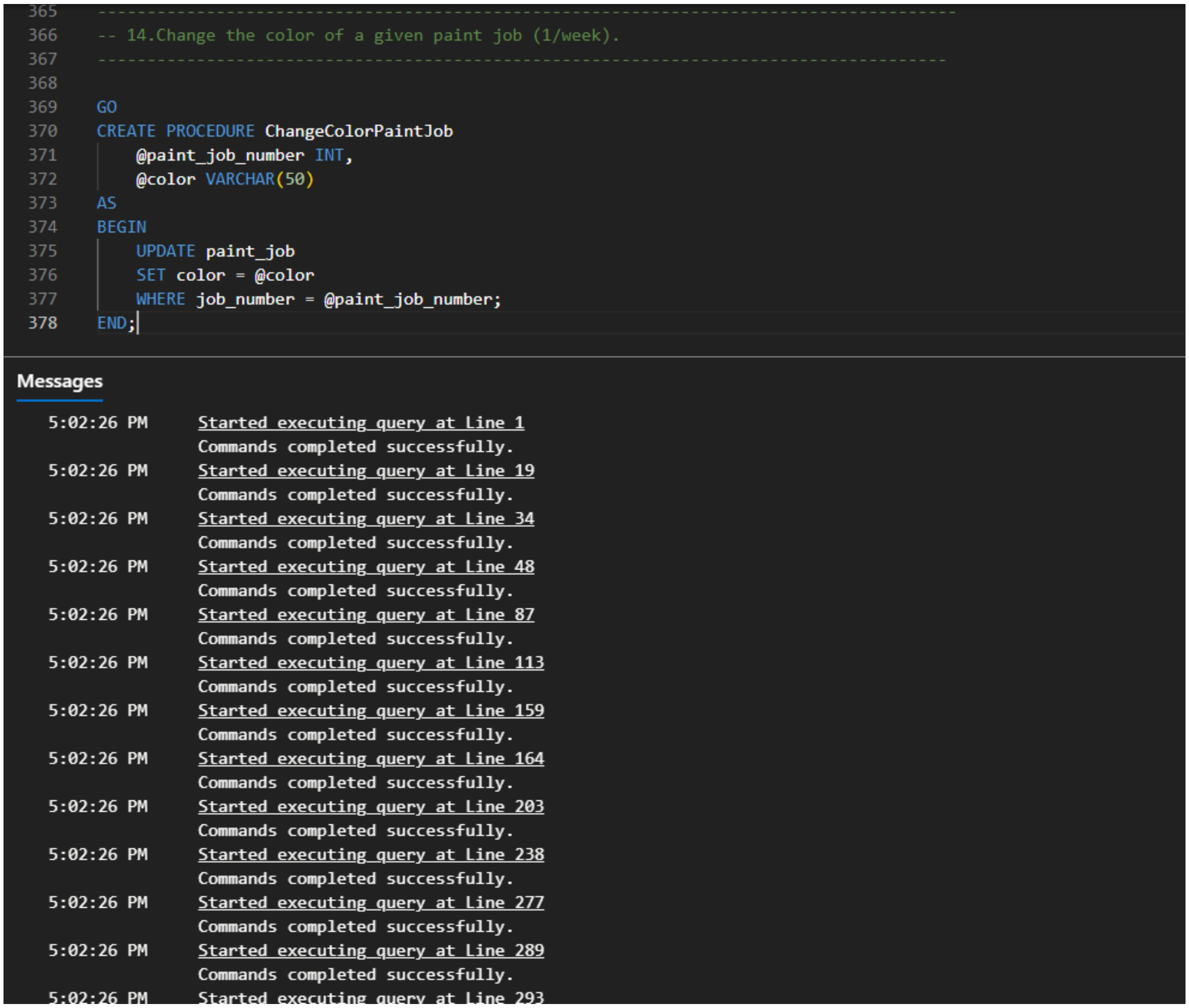
```sql
---------------------------------------------------------------------------
-- 9. Retrieve the total cost incurred on an assembly-id (200/day)
----------------------------------------------------------------------------

GO
CREATE PROCEDURE getTotalCostIncurred
    @assembly_id  INT,
    @total_cost DECIMAL(10, 2) OUTPUT
AS
BEGIN
    SELECT @total_cost = SUM(ct.sup_cost)
    FROM cost_transactions ct
    JOIN maintains_assembly ma ON ct.account_number = ma.account_number
    WHERE ma.assembly_id = @assembly_id;
END
PRINT 'total_sup_code: ' + CAST(@total_cost AS VARCHAR);
GO
-----------------------------------------------------------------------------------------------------------
-- 10. Retrieve the total labor time within a department for jobs completed in the department during a given
---------------------------------------------------------------------------------------------------------
GO
CREATE PROCEDURE GetTotalLaborTimeInDepartment
    @department_number INT,
    @job_completed_date VARCHAR(30),
    @total_labor_time DECIMAL(10,2) OUTPUT
AS
BEGIN
    -- Calculate total labor time for jobs completed on the given date
    SELECT @total_labor_time = ISNULL(SUM(j.labor_time), 0)
    FROM job AS j
    WHERE j.job_number IN (
        SELECT job_number
        FROM assigns
        WHERE process_id IN (
            SELECT process_id
            FROM supervised
            WHERE department_number = @department_number
```



```
    ) AND j.job_completed_date = @job_completed_date;
    SELECT @total_labor_time AS TotalLaborTime;
END;

-------------------------------------------------------------------------------------------
-- 11. Retrieve the processes through which a given assembly-id has passed so far (in date-
-- commenced order) and the department responsible for each process
-------------------------------------------------------------------------------------------

GO
CREATE PROCEDURE getProcessDepartmentDetails
    @assembly_id INT
AS
BEGIN
    SELECT s.process_id, s.department_number FROM supervised as s where
    s.process_id in (SELECT a.process_id FROM assigns as a where a.assembly_id = @assembly_id)
    ORDER BY (SELECT job_commenced_date
    FROM job AS j
    WHERE j.job_number = (SELECT a.job_number FROM assigns AS a WHERE a.assembly_id = @assembly_id))

END
GO
-------------------------------------------------------------------------------------------
-- 12. Retrieve the customers (in name order) whose category is in a given range (100/day)
-------------------------------------------------------------------------------------------

GO
CREATE PROCEDURE getCustomersForCategories
    @range_start INT,
    @range_end INT
AS
BEGIN
    SELECT * FROM customers
    WHERE category BETWEEN @range_start AND @range_end ORDER BY customer_name;
END
GO

```



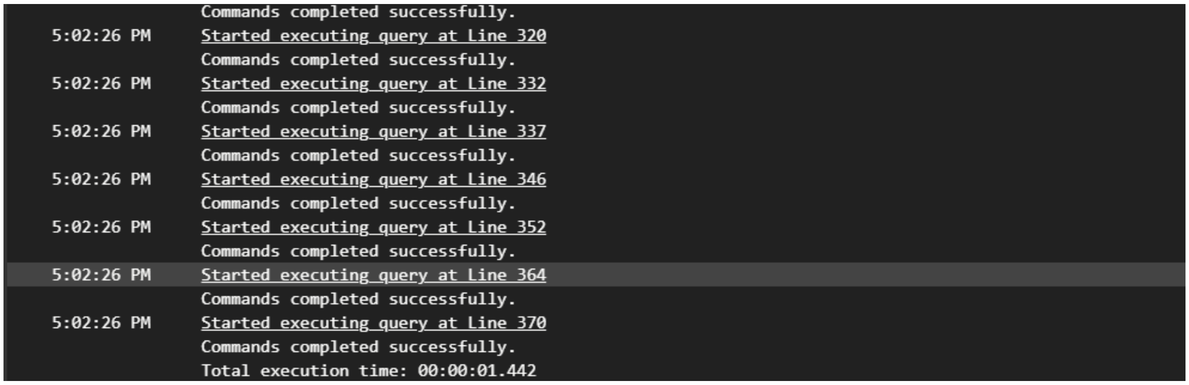
```
------------------------------------------------------------------------------------
                  all cut-jobs whose job-no is in a given range (1/month).
------------------------------------------------------------------------------------

GO
CREATE PROCEDURE deleteCutJobs
    @job_number_start INT,
    @job_number_end INT
AS
BEGIN
    DELETE FROM cut_job
    WHERE job_number BETWEEN @job_number_start AND @job_number_end;

    DELETE FROM job
    WHERE job_number BETWEEN @job_number_start AND @job_number_end;
END
GO

```



```
---------------------------------------------------------------------------------------
-- 14.Change the color of a given paint job (1/week).
---------------------------------------------------------------------------------------

GO
CREATE PROCEDURE ChangeColorPaintJob
    @paint_job_number INT,
    @color VARCHAR(50)
AS
BEGIN
    UPDATE paint_job
    SET color = @color
    WHERE job_number = @paint_job_number;
END;
```

Messages

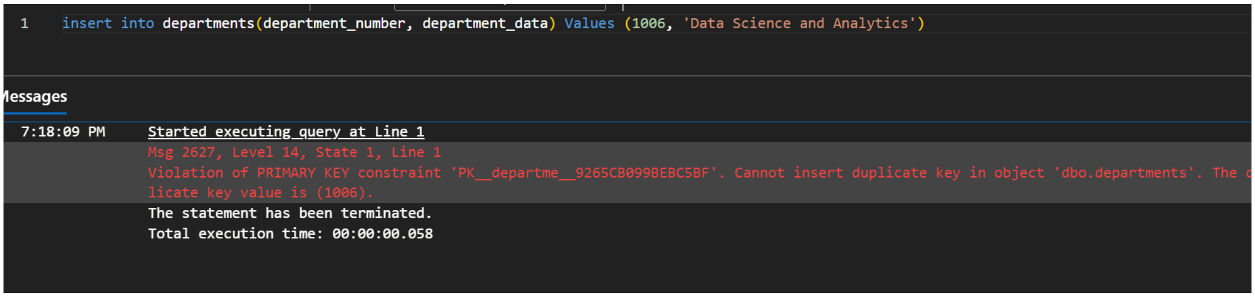
```
5:02:26 PM    Started executing query at Line 1
              Commands completed successfully.
5:02:26 PM    Started executing query at Line 19
              Commands completed successfully.
5:02:26 PM    Started executing query at Line 34
              Commands completed successfully.
5:02:26 PM    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

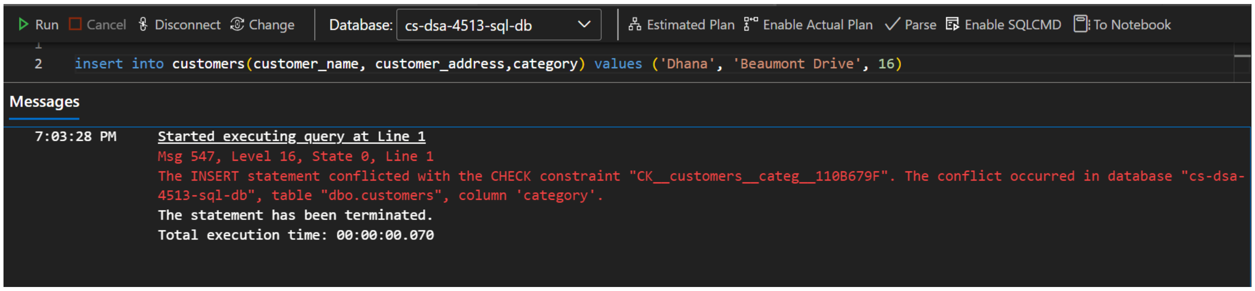
1. Primary Key Constraint



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

```
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__9265CB099BEBC5BF'. Cannot insert duplicate key in object 'dbo.departments'. The d
                licate key value is (1006).
                The statement has been terminated.
                Total execution time: 00:00:00.058
```

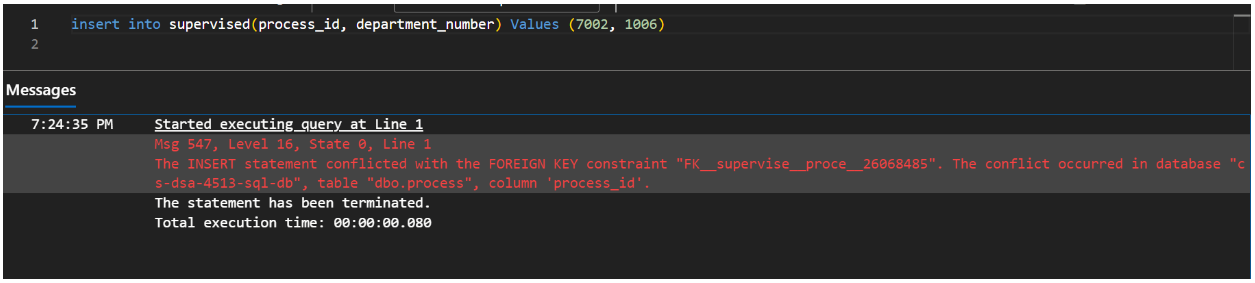
2. Category Constranit Error Handling



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

```
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
```

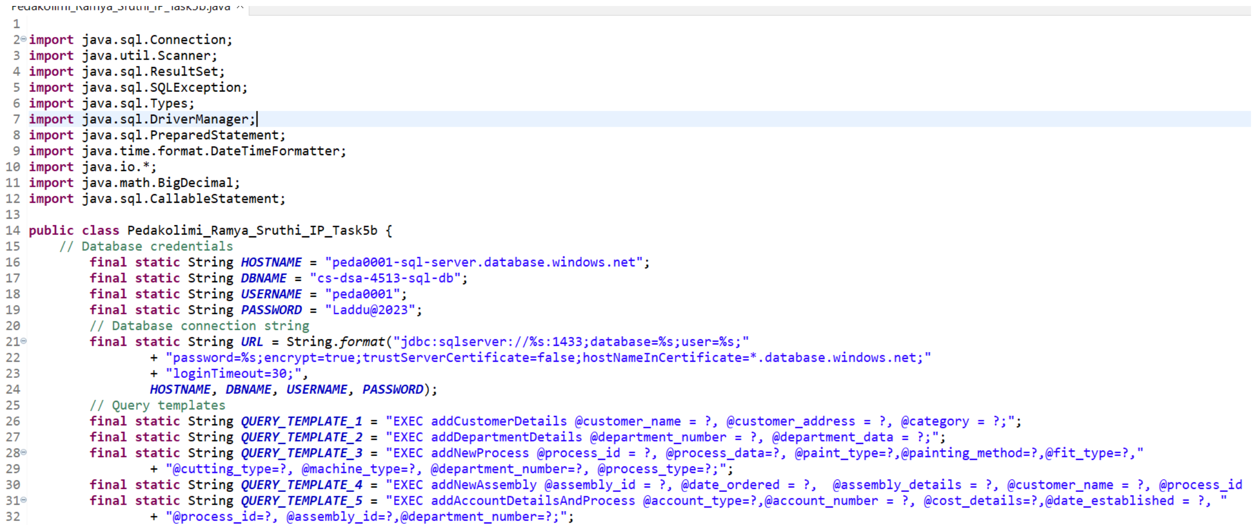
3. Foreign Key Constraint



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

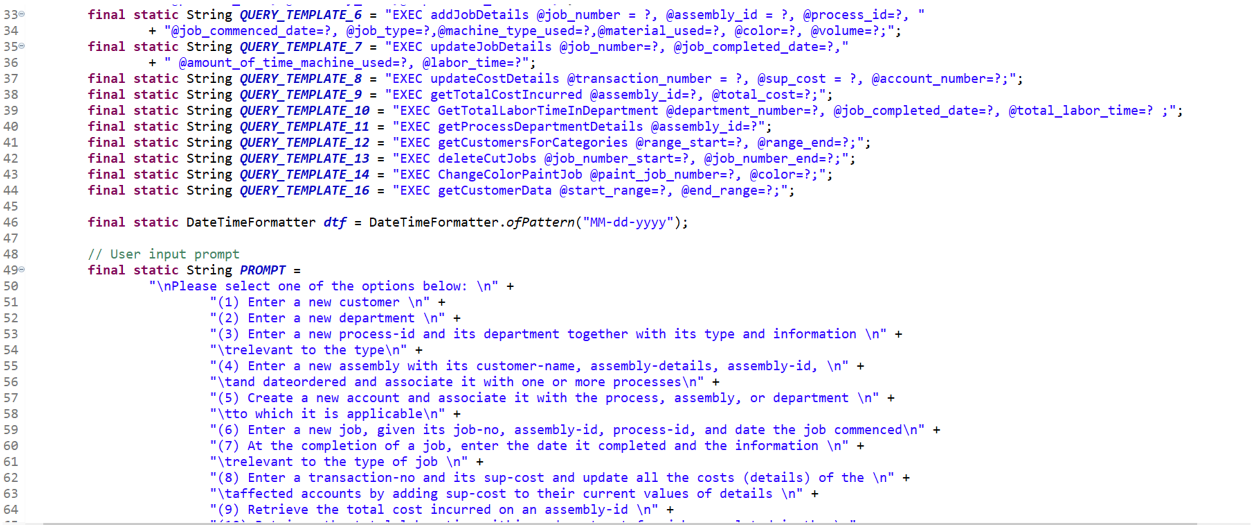
```
7:24:35 PM      Started executing query at Line 1
                Msg 547, Level 16, State 0, Line 1
                The INSERT statement conflicted with the FOREIGN KEY constraint "FK__supervise__proce__26068485". The conflict occurred in database "c
                s-dsa-4513-sql-db", table "dbo.process",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

5.2 Java Source Program

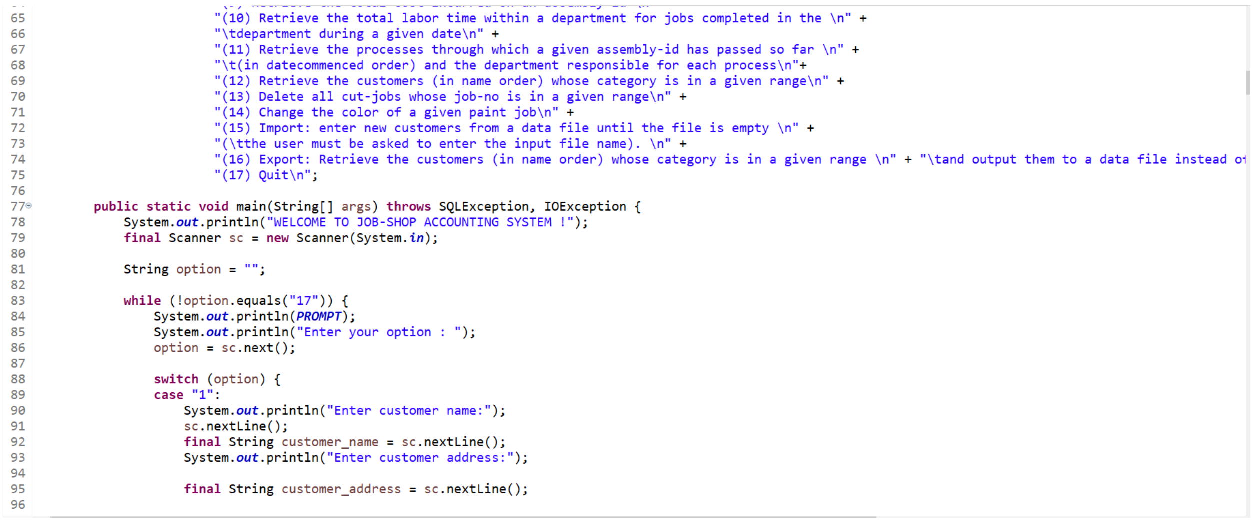


Pedakolimi_Ramya_Sruthi_IP_Task5b.java

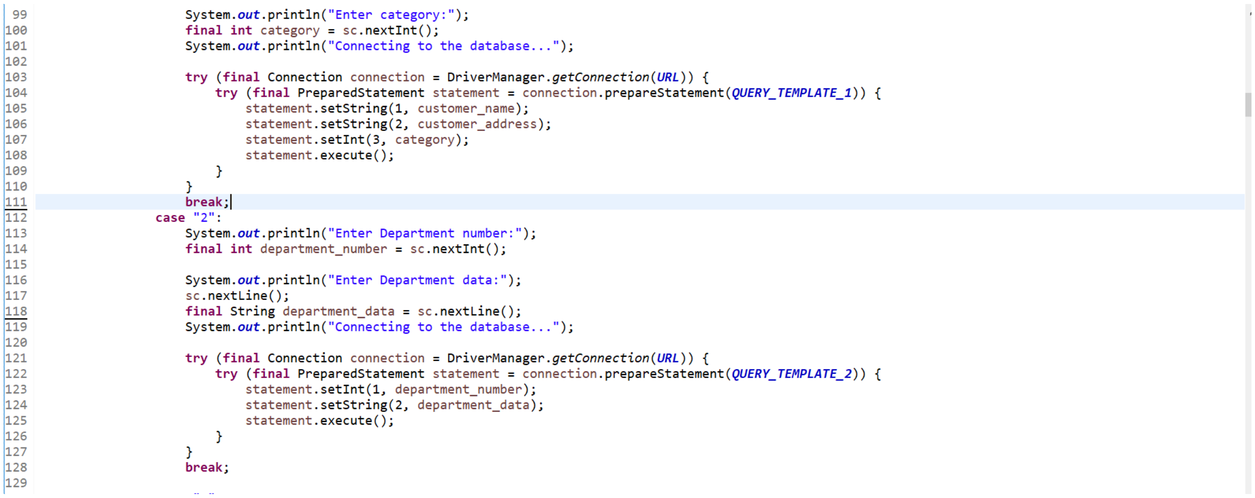
```java

import java.sql.Connection;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Types;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.time.format.DateTimeFormatter;
import java.io.*;
import java.math.BigDecimal;
import java.sql.CallableStatement;

public class Pedakolimi_Ramya_Sruthi_IP_Task5b {
    // Database credentials
        final static String HOSTNAME = "peda0001-sql-server.database.windows.net";
        final static String DBNAME = "cs-dsa-4513-sql-db";
        final static String USERNAME = "peda0001";
        final static String PASSWORD = "Laddu@2023";
        // Database connection string
        final static String URL = String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;"
                + "password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;"
                + "loginTimeout=30;",
                HOSTNAME, DBNAME, USERNAME, PASSWORD);
        // Query templates
        final static String QUERY_TEMPLATE_1 = "EXEC addCustomerDetails @customer_name = ?, @customer_address = ?, @category = ?;";
        final static String QUERY_TEMPLATE_2 = "EXEC addDepartmentDetails @department_number = ?, @department_data = ?;";
        final static String QUERY_TEMPLATE_3 = "EXEC addNewProcess @process_id = ?, @process_data=?, @paint_type=?,@painting_method=?,@fit_type=?,"
                + "@cutting_type=?, @machine_type=?, @department_number=?, @process_type=?;";
        final static String QUERY_TEMPLATE_4 = "EXEC addNewAssembly @assembly_id = ?, @date_ordered = ?,  @assembly_details = ?, @customer_name = ?, @process_id
        final static String QUERY_TEMPLATE_5 = "EXEC addAccountDetailsAndProcess @account_type=?,@account_number = ?, @cost_details=?,@date_established = ?, "
                + "@process_id=?, @assembly_id=?,@department_number=?;";
```



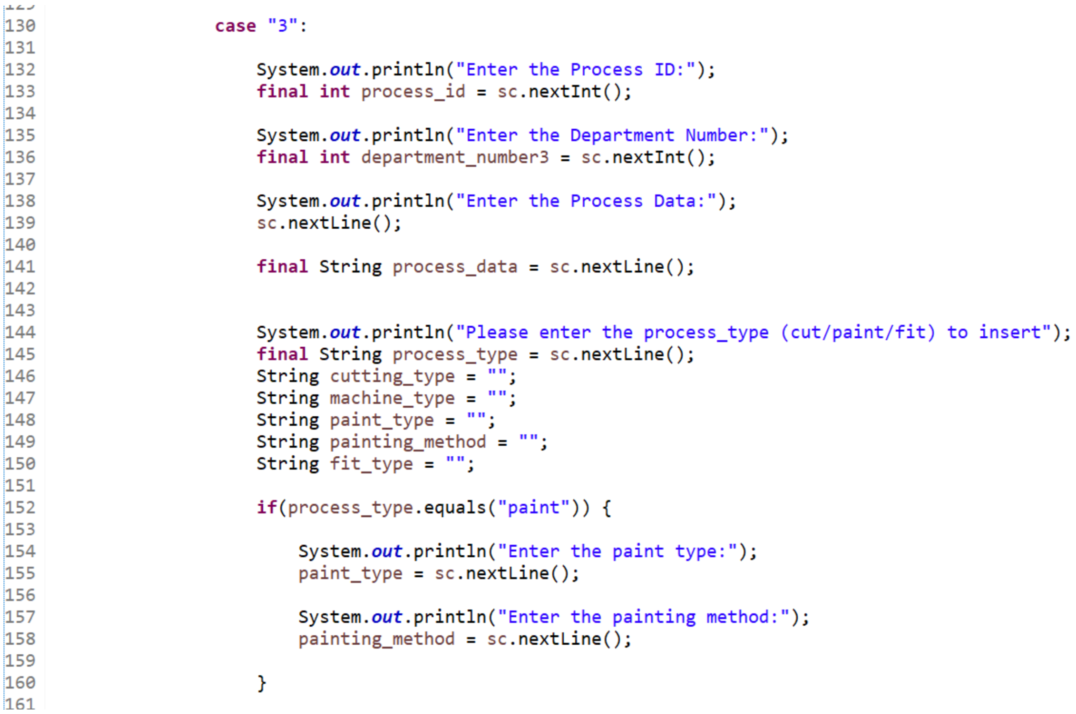
```java
        final static String QUERY_TEMPLATE_6 = "EXEC addJobDetails @job_number = ?, @assembly_id = ?, @process_id=?, "
                + "@job_commenced_date=?, @job_type=?,@machine_type_used=?,@material_used=?, @color=?, @volume=?;";
        final static String QUERY_TEMPLATE_7 = "EXEC updateJobDetails @job_number=?, @job_completed_date=?,"
                + " @amount_of_time_machine_used=?, @labor_time=?";
        final static String QUERY_TEMPLATE_8 = "EXEC updateCostDetails @transaction_number = ?, @sup_cost = ?, @account_number=?;";
        final static String QUERY_TEMPLATE_9 = "EXEC getTotalCostIncurred @assembly_id=?, @total_cost=?;";
        final static String QUERY_TEMPLATE_10 = "EXEC GetTotalLaborTimeInDepartment @department_number=?, @job_completed_date=?, @total_labor_time=? ;";
        final static String QUERY_TEMPLATE_11 = "EXEC getProcessDepartmentDetails @assembly_id=?";
        final static String QUERY_TEMPLATE_12 = "EXEC getCustomersForCategories @range_start=?, @range_end=?;";
        final static String QUERY_TEMPLATE_13 = "EXEC deleteCutJobs @job_number_start=?, @job_number_end=?;";
        final static String QUERY_TEMPLATE_14 = "EXEC ChangeColorPaintJob @paint_job_number=?, @color=?;";
        final static String QUERY_TEMPLATE_16 = "EXEC getCustomerData @start_range=?, @end_range=?;";

        final static DateTimeFormatter dtf = DateTimeFormatter.ofPattern("MM-dd-yyyy");

        // User input prompt
        final static String PROMPT =
                "\nPlease select one of the options below: \n" +
                        "(1) Enter a new customer \n" +
                        "(2) Enter a new department \n" +
                        "(3) Enter a new process-id and its department together with its type and information \n" +
                        "\trelevant to the type\n" +
                        "(4) Enter a new assembly with its customer-name, assembly-details, assembly-id, \n" +
                        "\tand dateordered and associate it with one or more processes\n" +
                        "(5) Create a new account and associate it with the process, assembly, or department \n" +
                        "\tto which it is applicable\n" +
                        "(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced\n" +
                        "(7) At the completion of a job, enter the date it completed and the information \n" +
                        "\trelevant to the type of job \n" +
                        "(8) Enter a transaction-no and its sup-cost and update all the costs (details) of the \n" +
                        "\taffected accounts by adding sup-cost to their current values of details \n" +
                        "(9) Retrieve the total cost incurred on an assembly-id \n" +
```



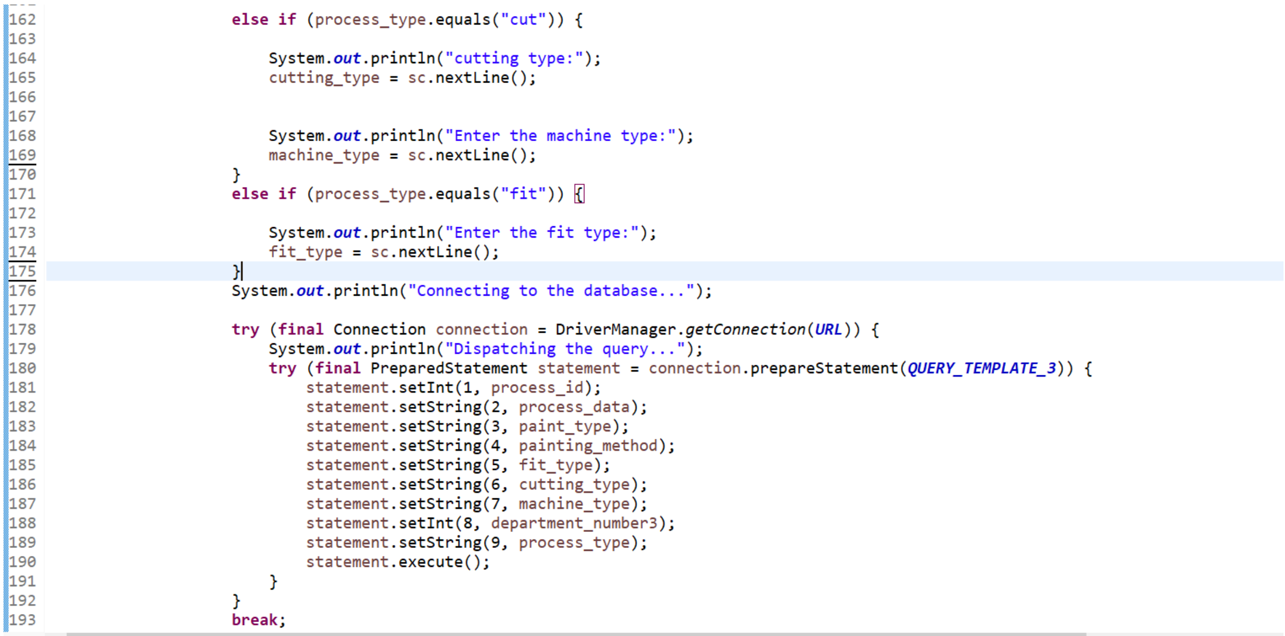
```java
                        "(10) Retrieve the total labor time within a department for jobs completed in the \n" +
                        "\tdepartment during a given date\n" +
                        "(11) Retrieve the processes through which a given assembly-id has passed so far \n" +
                        "\t(in datecommenced order) and the department responsible for each process\n"+
                        "(12) Retrieve the customers (in name order) whose category is in a given range\n" +
                        "(13) Delete all cut-jobs whose job-no is in a given range\n" +
                        "(14) Change the color of a given paint job\n" +
                        "(15) Import: enter new customers from a data file until the file is empty \n" +
                        "(\tthe user must be asked to enter the input file name). \n" +
                        "(16) Export: Retrieve the customers (in name order) whose category is in a given range \n" + "\tand output them to a data file instead o
                        "(17) Quit\n";

        public static void main(String[] args) throws SQLException, IOException {
            System.out.println("WELCOME TO JOB-SHOP ACCOUNTING SYSTEM !");
            final Scanner sc = new Scanner(System.in);

            String option = "";

            while (!option.equals("17")) {
                System.out.println(PROMPT);
                System.out.println("Enter your option : ");
                option = sc.next();

                switch (option) {
                case "1":
                    System.out.println("Enter customer name:");
                    sc.nextLine();
                    final String customer_name = sc.nextLine();
                    System.out.println("Enter customer address:");

                    final String customer_address = sc.nextLine();

```



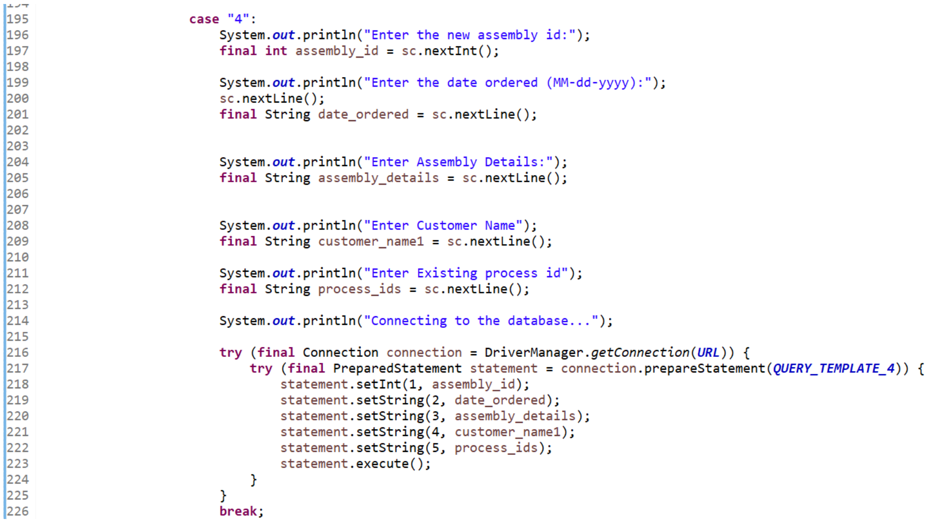
```
                System.out.println("Enter category:");
                final int category = sc.nextInt();
                System.out.println("Connecting to the database...");

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_1)) {
                        statement.setString(1, customer_name);
                        statement.setString(2, customer_address);
                        statement.setInt(3, category);
                        statement.execute();
                    }
                }
                break;
            case "2":
                System.out.println("Enter Department number:");
                final int department_number = sc.nextInt();

                System.out.println("Enter Department data:");
                sc.nextLine();
                final String department_data = sc.nextLine();
                System.out.println("Connecting to the database...");

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_2)) {
                        statement.setInt(1, department_number);
                        statement.setString(2, department_data);
                        statement.execute();
                    }
                }
                break;
```



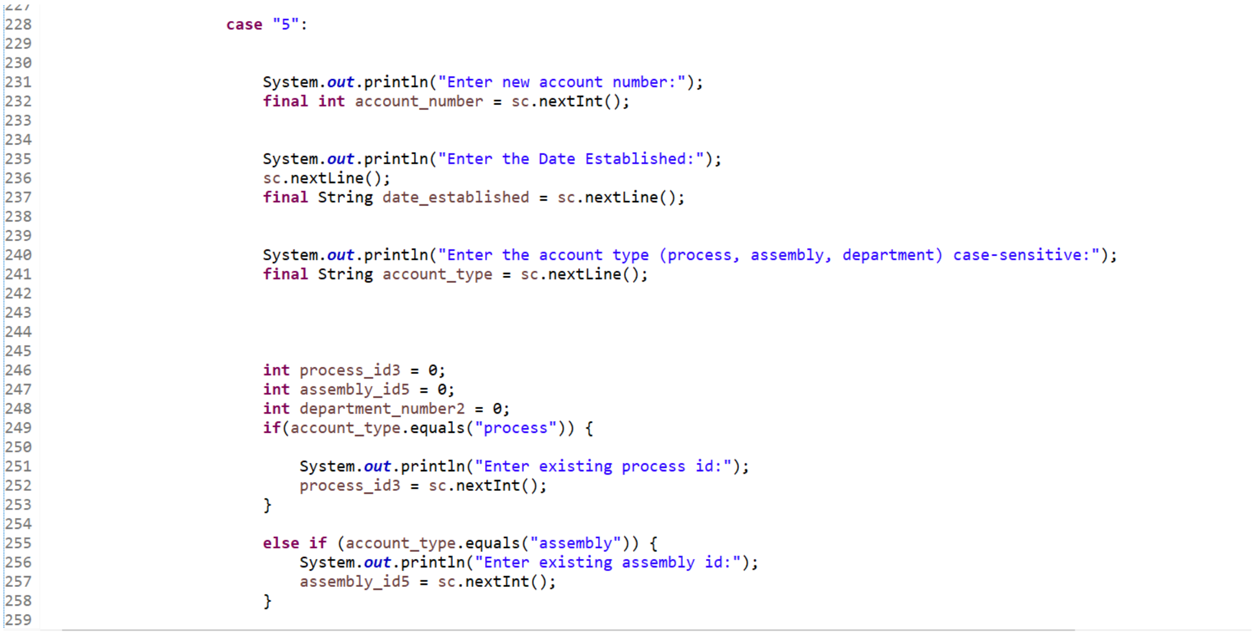
```
            case "3":

                System.out.println("Enter the Process ID:");
                final int process_id = sc.nextInt();

                System.out.println("Enter the Department Number:");
                final int department_number3 = sc.nextInt();

                System.out.println("Enter the Process Data:");
                sc.nextLine();

                final String process_data = sc.nextLine();


                System.out.println("Please enter the process_type (cut/paint/fit) to insert");
                final String process_type = sc.nextLine();
                String cutting_type = "";
                String machine_type = "";
                String paint_type = "";
                String painting_method = "";
                String fit_type = "";

                if(process_type.equals("paint")) {

                    System.out.println("Enter the paint type:");
                    paint_type = sc.nextLine();

                    System.out.println("Enter the painting method:");
                    painting_method = sc.nextLine();

                }
```



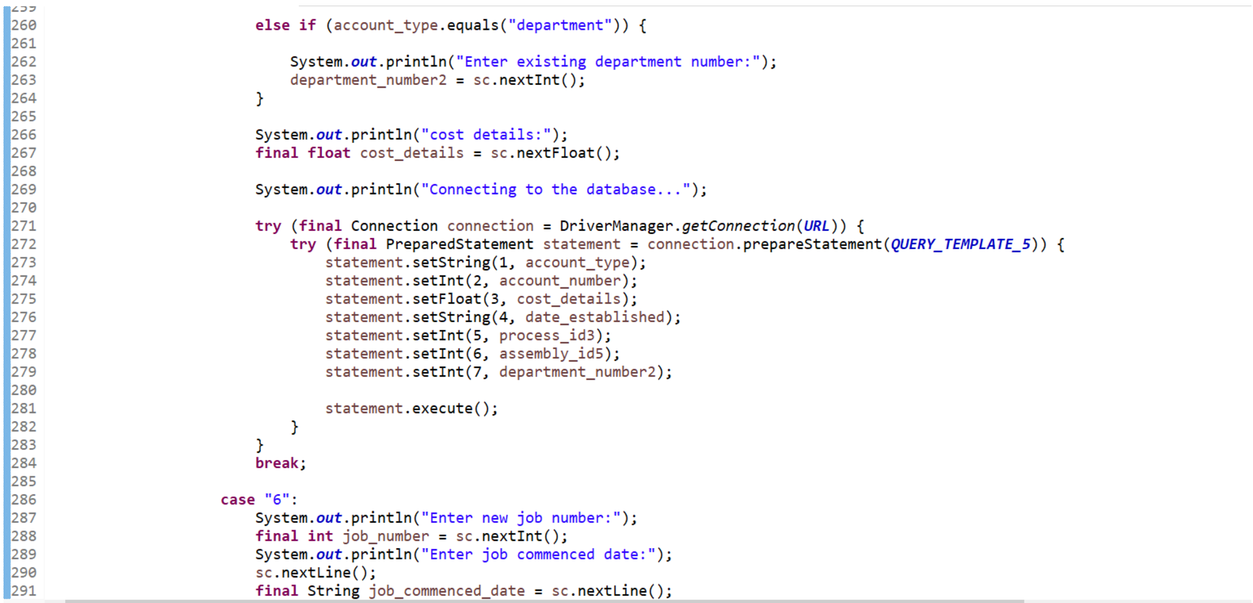
```
else if (process_type.equals("cut")) {

    System.out.println("cutting type:");
    cutting_type = sc.nextLine();


    System.out.println("Enter the machine type:");
    machine_type = sc.nextLine();
}
else if (process_type.equals("fit")) {

    System.out.println("Enter the fit type:");
    fit_type = sc.nextLine();
}
System.out.println("Connecting to the database...");

try (final Connection connection = DriverManager.getConnection(URL)) {
    System.out.println("Dispatching the query...");
    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_3)) {
        statement.setInt(1, process_id);
        statement.setString(2, process_data);
        statement.setString(3, paint_type);
        statement.setString(4, painting_method);
        statement.setString(5, fit_type);
        statement.setString(6, cutting_type);
        statement.setString(7, machine_type);
        statement.setInt(8, department_number3);
        statement.setString(9, process_type);
        statement.execute();
    }
}
break;
```



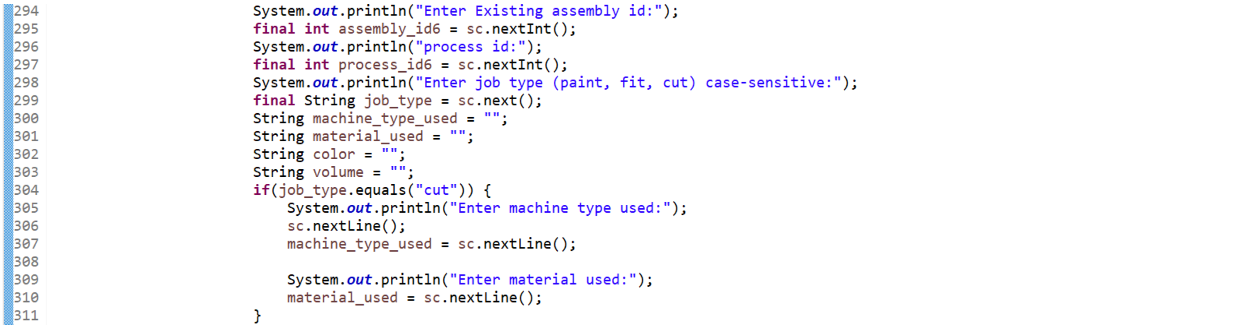
```
case "4":
    System.out.println("Enter the new assembly id:");
    final int assembly_id = sc.nextInt();

    System.out.println("Enter the date ordered (MM-dd-yyyy):");
    sc.nextLine();
    final String date_ordered = sc.nextLine();


    System.out.println("Enter Assembly Details:");
    final String assembly_details = sc.nextLine();


    System.out.println("Enter Customer Name");
    final String customer_name1 = sc.nextLine();

    System.out.println("Enter Existing process id");
    final String process_ids = sc.nextLine();

    System.out.println("Connecting to the database...");

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_4)) {
            statement.setInt(1, assembly_id);
            statement.setString(2, date_ordered);
            statement.setString(3, assembly_details);
            statement.setString(4, customer_name1);
            statement.setString(5, process_ids);
            statement.execute();
        }
    }
    break;
```



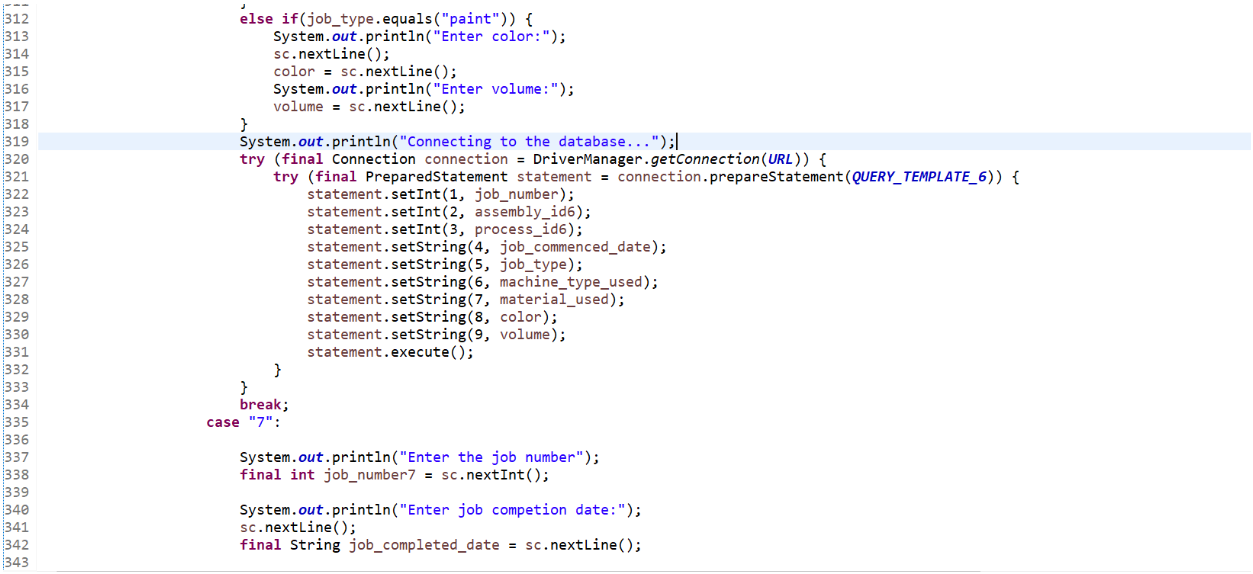
```
case "5":


    System.out.println("Enter new account number:");
    final int account_number = sc.nextInt();


    System.out.println("Enter the Date Established:");
    sc.nextLine();
    final String date_established = sc.nextLine();


    System.out.println("Enter the account type (process, assembly, department) case-sensitive:");
    final String account_type = sc.nextLine();




    int process_id3 = 0;
    int assembly_id5 = 0;
    int department_number2 = 0;
    if(account_type.equals("process")) {

        System.out.println("Enter existing process id:");
        process_id3 = sc.nextInt();
    }

    else if (account_type.equals("assembly")) {
        System.out.println("Enter existing assembly id:");
        assembly_id5 = sc.nextInt();
    }

```



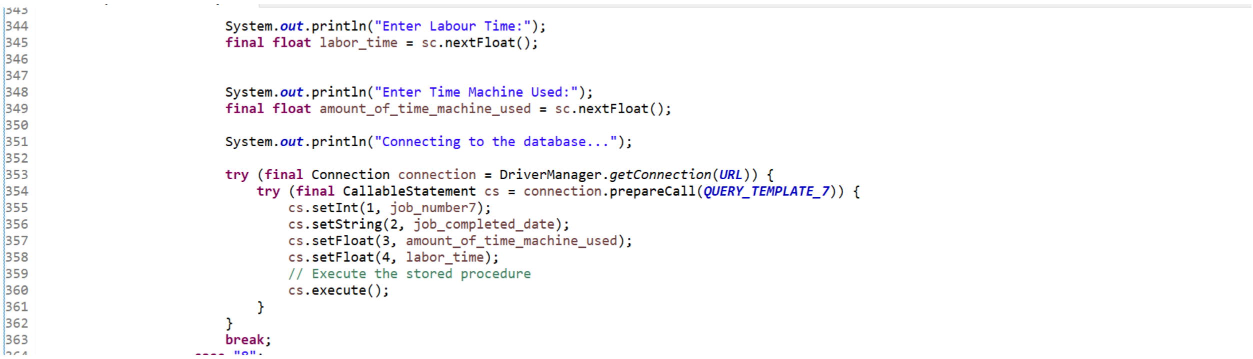
```
    else if (account_type.equals("department")) {

        System.out.println("Enter existing department number:");
        department_number2 = sc.nextInt();
    }

    System.out.println("cost details:");
    final float cost_details = sc.nextFloat();

    System.out.println("Connecting to the database...");

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_5)) {
            statement.setString(1, account_type);
            statement.setInt(2, account_number);
            statement.setFloat(3, cost_details);
            statement.setString(4, date_established);
            statement.setInt(5, process_id3);
            statement.setInt(6, assembly_id5);
            statement.setInt(7, department_number2);

            statement.execute();
        }
    }
    break;

case "6":
    System.out.println("Enter new job number:");
    final int job_number = sc.nextInt();
    System.out.println("Enter job commenced date:");
    sc.nextLine();
    final String job_commenced_date = sc.nextLine();
```



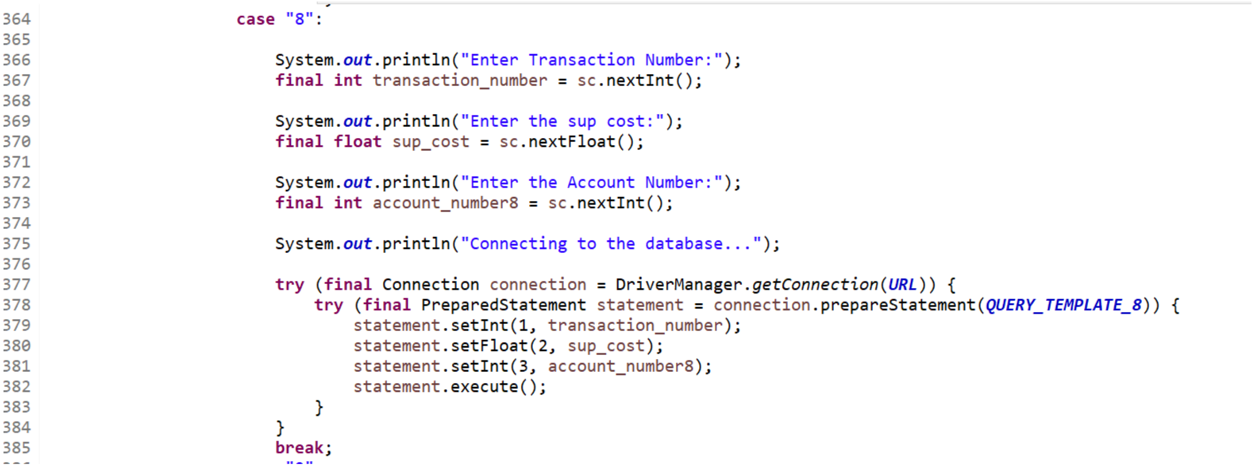
```
    System.out.println("Enter Existing assembly id:");
    final int assembly_id6 = sc.nextInt();
    System.out.println("process id:");
    final int process_id6 = sc.nextInt();
    System.out.println("Enter job type (paint, fit, cut) case-sensitive:");
    final String job_type = sc.next();
    String machine_type_used = "";
    String material_used = "";
    String color = "";
    String volume = "";
    if(job_type.equals("cut")) {
        System.out.println("Enter machine type used:");
        sc.nextLine();
        machine_type_used = sc.nextLine();

        System.out.println("Enter material used:");
        material_used = sc.nextLine();
    }
```



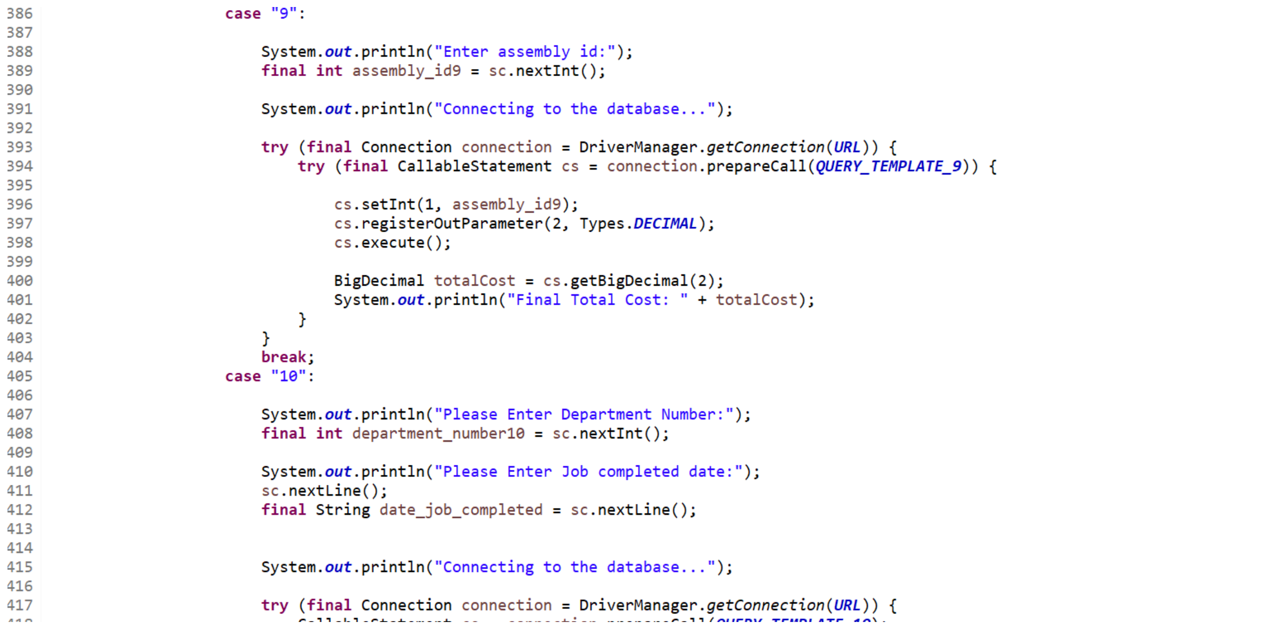
```
                else if(job_type.equals("paint")) {
                    System.out.println("Enter color:");
                    sc.nextLine();
                    color = sc.nextLine();
                    System.out.println("Enter volume:");
                    volume = sc.nextLine();
                }
                System.out.println("Connecting to the database...");
                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_6)) {
                        statement.setInt(1, job_number);
                        statement.setInt(2, assembly_id6);
                        statement.setInt(3, process_id6);
                        statement.setString(4, job_commenced_date);
                        statement.setString(5, job_type);
                        statement.setString(6, machine_type_used);
                        statement.setString(7, material_used);
                        statement.setString(8, color);
                        statement.setString(9, volume);
                        statement.execute();
                    }
                }
                break;
            case "7":

                System.out.println("Enter the job number");
                final int job_number7 = sc.nextInt();

                System.out.println("Enter job competion date:");
                sc.nextLine();
                final String job_completed_date = sc.nextLine();

```



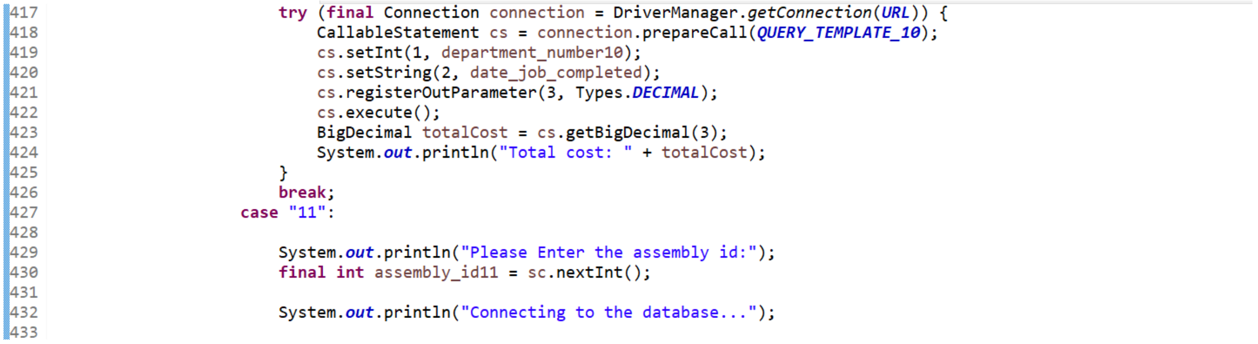
```
                System.out.println("Enter Labour Time:");
                final float labor_time = sc.nextFloat();


                System.out.println("Enter Time Machine Used:");
                final float amount_of_time_machine_used = sc.nextFloat();

                System.out.println("Connecting to the database...");

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_7)) {
                        cs.setInt(1, job_number7);
                        cs.setString(2, job_completed_date);
                        cs.setFloat(3, amount_of_time_machine_used);
                        cs.setFloat(4, labor_time);
                        // Execute the stored procedure
                        cs.execute();
                    }
                }
                break;
```



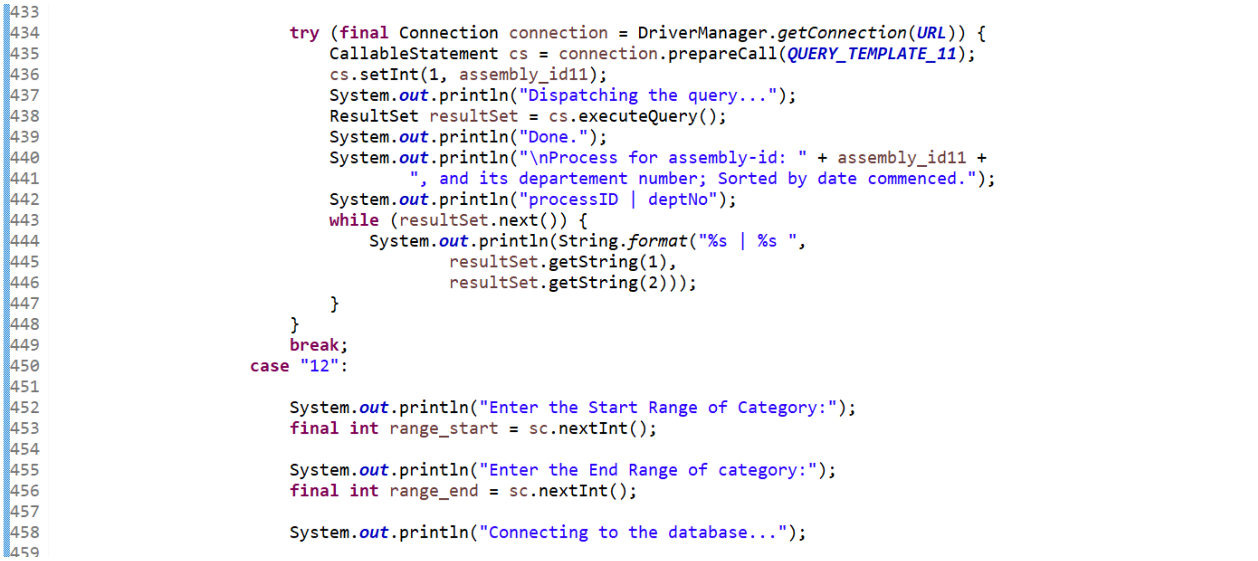
```
            case "8":

                System.out.println("Enter Transaction Number:");
                final int transaction_number = sc.nextInt();

                System.out.println("Enter the sup cost:");
                final float sup_cost = sc.nextFloat();

                System.out.println("Enter the Account Number:");
                final int account_number8 = sc.nextInt();

                System.out.println("Connecting to the database...");

                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_8)) {
                        statement.setInt(1, transaction_number);
                        statement.setFloat(2, sup_cost);
                        statement.setInt(3, account_number8);
                        statement.execute();
                    }
                }
                break;
```



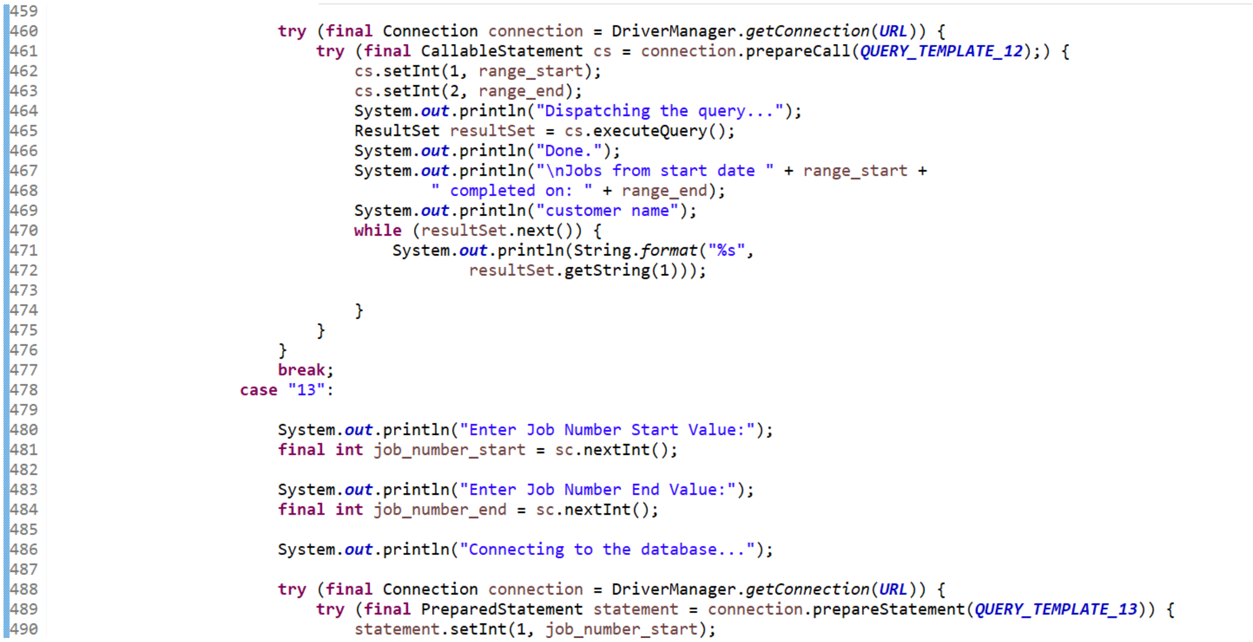
```
case "9":

    System.out.println("Enter assembly id:");
    final int assembly_id9 = sc.nextInt();

    System.out.println("Connecting to the database...");

    try (final Connection connection = DriverManager.getConnection(URL)) {
        try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_9)) {

            cs.setInt(1, assembly_id9);
            cs.registerOutParameter(2, Types.DECIMAL);
            cs.execute();

            BigDecimal totalCost = cs.getBigDecimal(2);
            System.out.println("Final Total Cost: " + totalCost);
        }
    }
    break;
case "10":

    System.out.println("Please Enter Department Number:");
    final int department_number10 = sc.nextInt();

    System.out.println("Please Enter Job completed date:");
    sc.nextLine();
    final String date_job_completed = sc.nextLine();


    System.out.println("Connecting to the database...");

    try (final Connection connection = DriverManager.getConnection(URL)) {
```



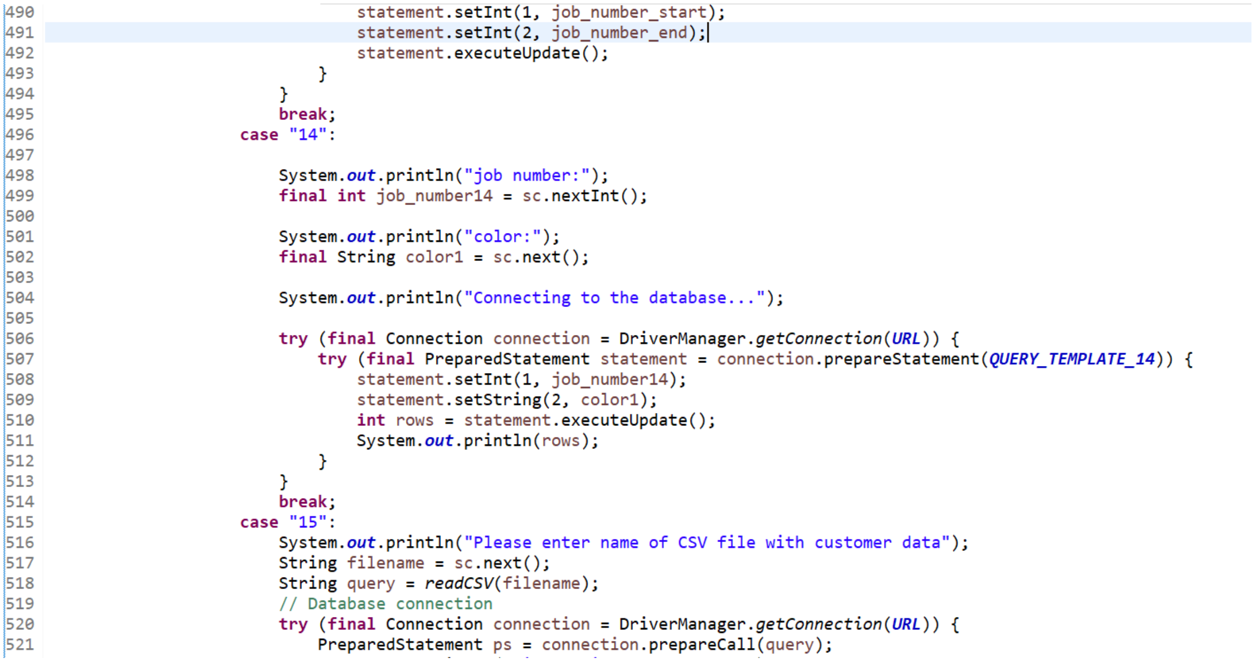
```
    try (final Connection connection = DriverManager.getConnection(URL)) {
        CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_10);
        cs.setInt(1, department_number10);
        cs.setString(2, date_job_completed);
        cs.registerOutParameter(3, Types.DECIMAL);
        cs.execute();
        BigDecimal totalCost = cs.getBigDecimal(3);
        System.out.println("Total cost: " + totalCost);
    }
    break;
case "11":

    System.out.println("Please Enter the assembly id:");
    final int assembly_id11 = sc.nextInt();

    System.out.println("Connecting to the database...");
```



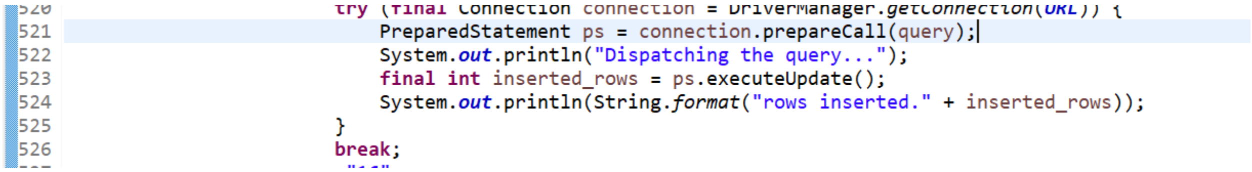
```
    try (final Connection connection = DriverManager.getConnection(URL)) {
        CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_11);
        cs.setInt(1, assembly_id11);
        System.out.println("Dispatching the query...");
        ResultSet resultSet = cs.executeQuery();
        System.out.println("Done.");
        System.out.println("\nProcess for assembly-id: " + assembly_id11 +
                ", and its departement number; Sorted by date commenced.");
        System.out.println("processID | deptNo");
        while (resultSet.next()) {
            System.out.println(String.format("%s | %s ",
                    resultSet.getString(1),
                    resultSet.getString(2)));
        }
    }
    break;
case "12":

    System.out.println("Enter the Start Range of Category:");
    final int range_start = sc.nextInt();

    System.out.println("Enter the End Range of category:");
    final int range_end = sc.nextInt();

    System.out.println("Connecting to the database...");
```



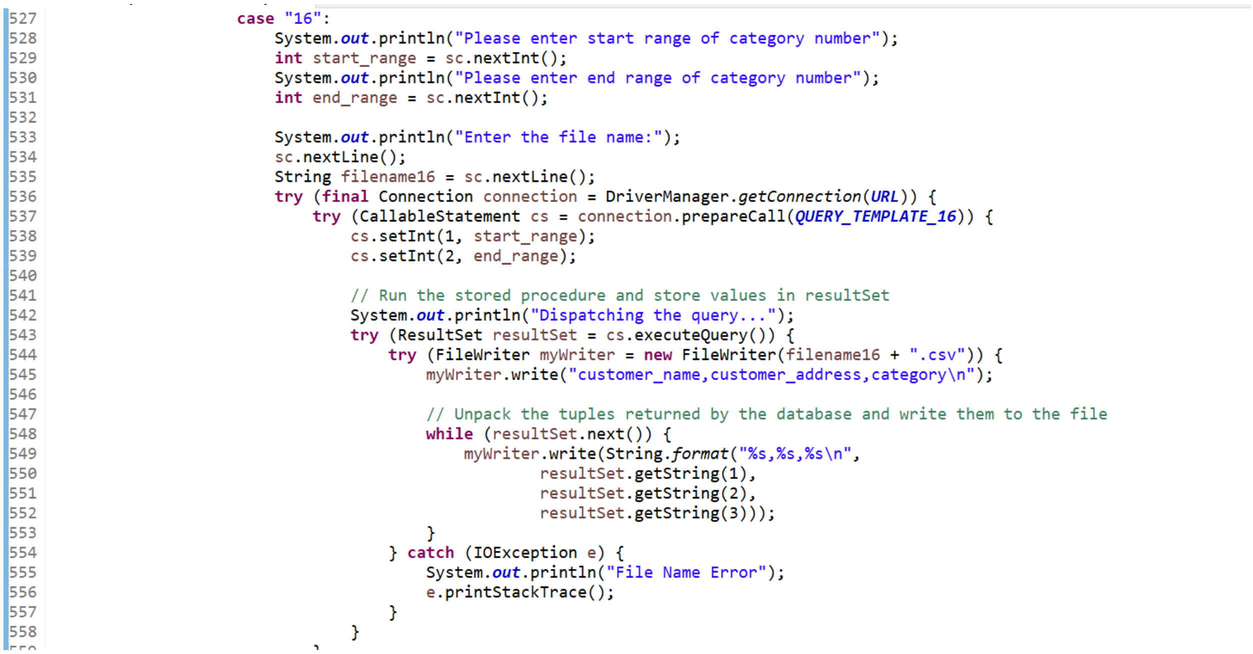
```

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try (final CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_12);) {
                    cs.setInt(1, range_start);
                    cs.setInt(2, range_end);
                    System.out.println("Dispatching the query...");
                    ResultSet resultSet = cs.executeQuery();
                    System.out.println("Done.");
                    System.out.println("\nJobs from start date " + range_start +
                            " completed on: " + range_end);
                    System.out.println("customer name");
                    while (resultSet.next()) {
                        System.out.println(String.format("%s",
                                resultSet.getString(1)));

                    }
                }
            }
            break;
        case "13":

            System.out.println("Enter Job Number Start Value:");
            final int job_number_start = sc.nextInt();

            System.out.println("Enter Job Number End Value:");
            final int job_number_end = sc.nextInt();

            System.out.println("Connecting to the database...");

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_13)) {
                    statement.setInt(1, job_number_start);
```



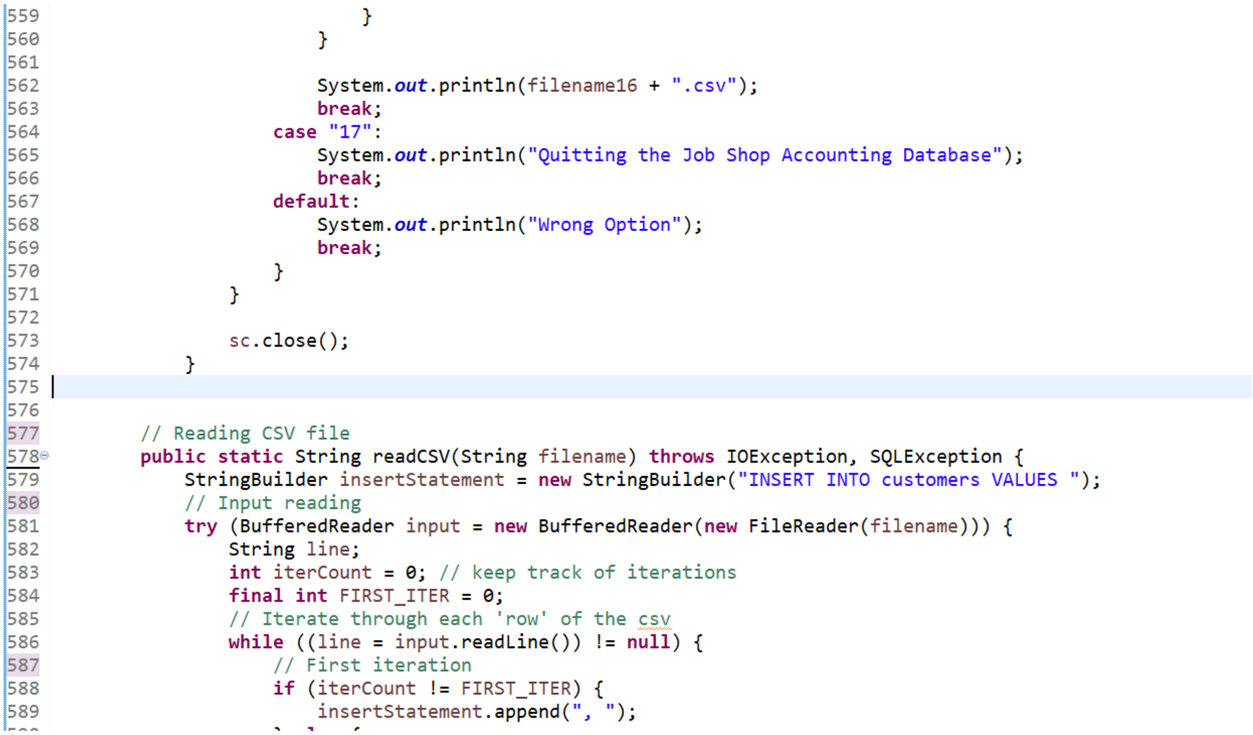
```
                    statement.setInt(1, job_number_start);
                    statement.setInt(2, job_number_end);
                    statement.executeUpdate();
                }
            }
            break;
        case "14":

            System.out.println("job number:");
            final int job_number14 = sc.nextInt();

            System.out.println("color:");
            final String color1 = sc.next();

            System.out.println("Connecting to the database...");

            try (final Connection connection = DriverManager.getConnection(URL)) {
                try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_14)) {
                    statement.setInt(1, job_number14);
                    statement.setString(2, color1);
                    int rows = statement.executeUpdate();
                    System.out.println(rows);
                }
            }
            break;
        case "15":
            System.out.println("Please enter name of CSV file with customer data");
            String filename = sc.next();
            String query = readCSV(filename);
            // Database connection
            try (final Connection connection = DriverManager.getConnection(URL)) {
                PreparedStatement ps = connection.prepareCall(query);
```



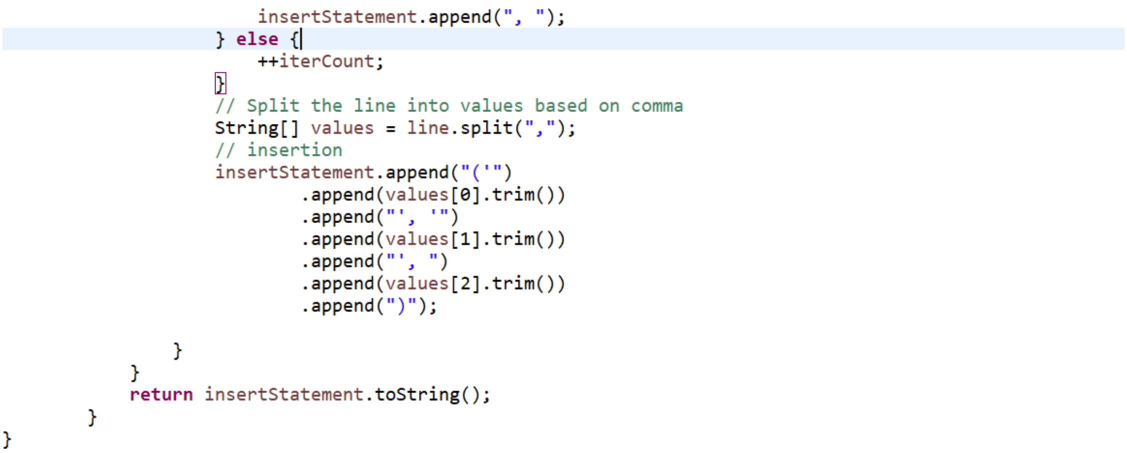
```
            try (final Connection connection = DriverManager.getConnection(URL)) {
                PreparedStatement ps = connection.prepareCall(query);
                System.out.println("Dispatching the query...");
                final int inserted_rows = ps.executeUpdate();
                System.out.println(String.format("rows inserted." + inserted_rows));
            }
            break;
```



```
                case "16":
                    System.out.println("Please enter start range of category number");
                    int start_range = sc.nextInt();
                    System.out.println("Please enter end range of category number");
                    int end_range = sc.nextInt();

                    System.out.println("Enter the file name:");
                    sc.nextLine();
                    String filename16 = sc.nextLine();
                    try (final Connection connection = DriverManager.getConnection(URL)) {
                        try (CallableStatement cs = connection.prepareCall(QUERY_TEMPLATE_16)) {
                            cs.setInt(1, start_range);
                            cs.setInt(2, end_range);

                            // Run the stored procedure and store values in resultSet
                            System.out.println("Dispatching the query...");
                            try (ResultSet resultSet = cs.executeQuery()) {
                                try (FileWriter myWriter = new FileWriter(filename16 + ".csv")) {
                                    myWriter.write("customer_name,customer_address,category\n");

                                    // Unpack the tuples returned by the database and write them to the file
                                    while (resultSet.next()) {
                                        myWriter.write(String.format("%s,%s,%s\n",
                                                resultSet.getString(1),
                                                resultSet.getString(2),
                                                resultSet.getString(3)));
                                    }
                                } catch (IOException e) {
                                    System.out.println("File Name Error");
                                    e.printStackTrace();
                                }
                            }
```

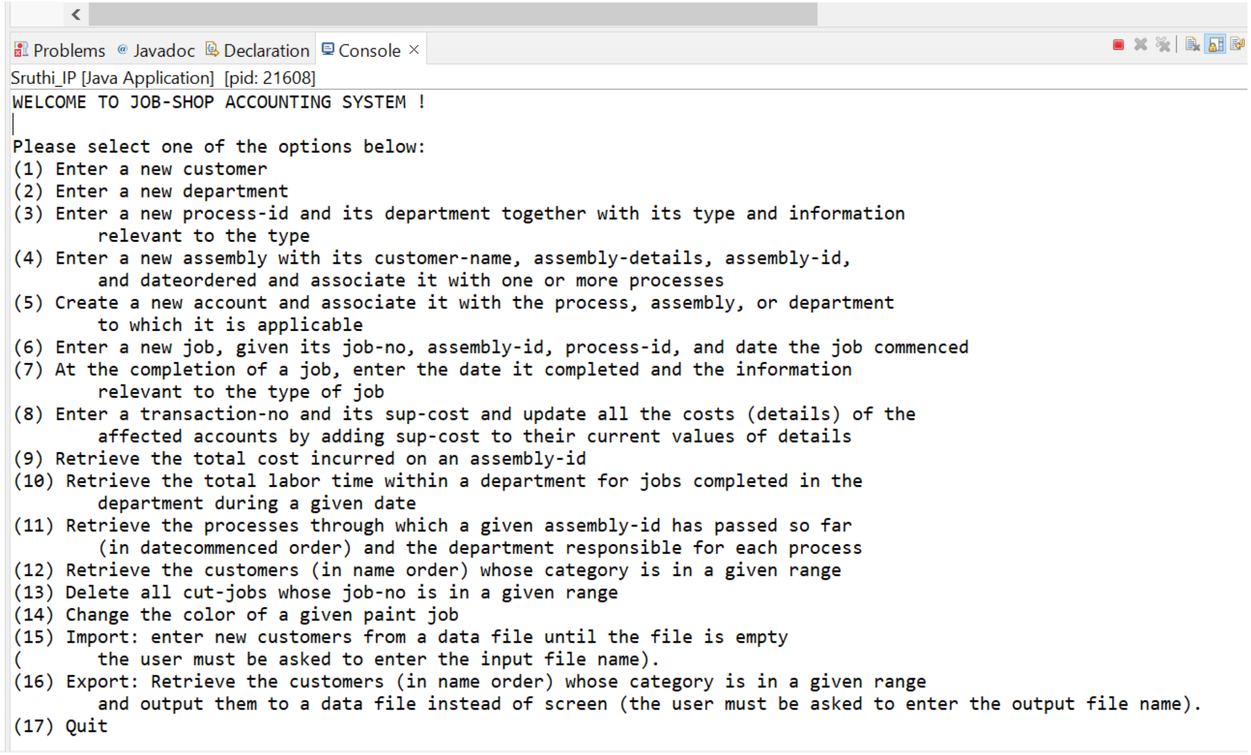


```
                        }
                    }

                    System.out.println(filename16 + ".csv");
                    break;
                case "17":
                    System.out.println("Quitting the Job Shop Accounting Database");
                    break;
                default:
                    System.out.println("Wrong Option");
                    break;
                }
            }

            sc.close();
        }


    // Reading CSV file
    public static String readCSV(String filename) throws IOException, SQLException {
        StringBuilder insertStatement = new StringBuilder("INSERT INTO customers VALUES ");
        // Input reading
        try (BufferedReader input = new BufferedReader(new FileReader(filename))) {
            String line;
            int iterCount = 0; // keep track of iterations
            final int FIRST_ITER = 0;
            // Iterate through each 'row' of the csv
            while ((line = input.readLine()) != null) {
                // First iteration
                if (iterCount != FIRST_ITER) {
                    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();
        }
}
```

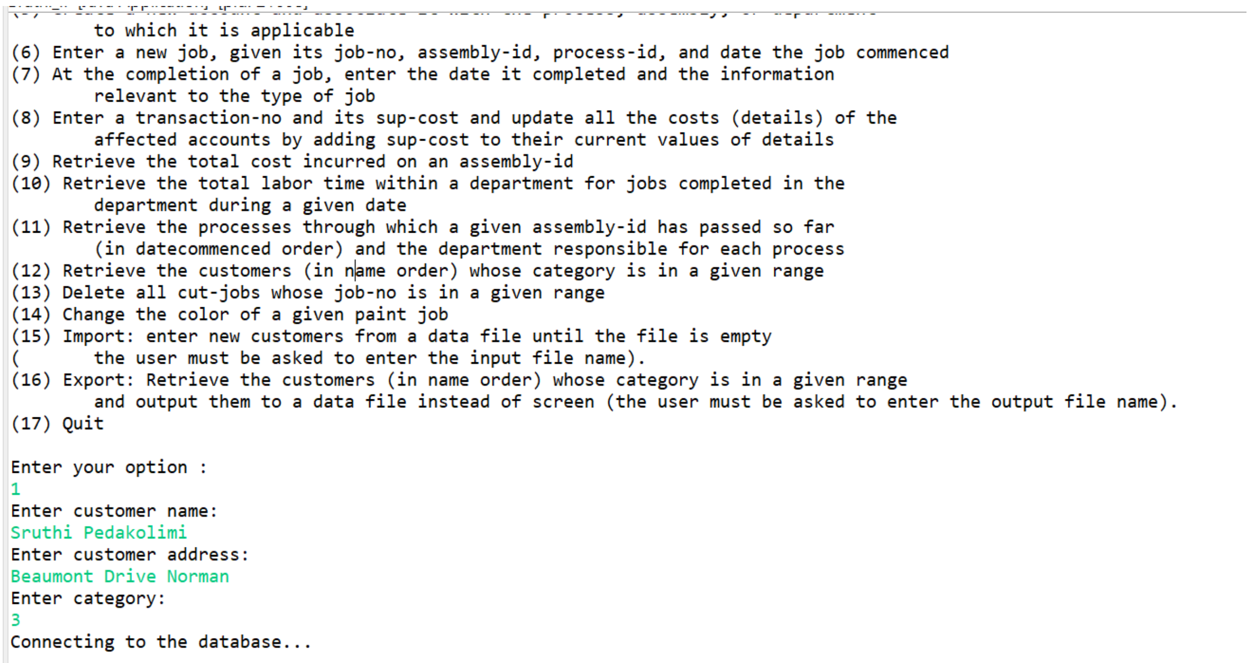
Compilation of the program



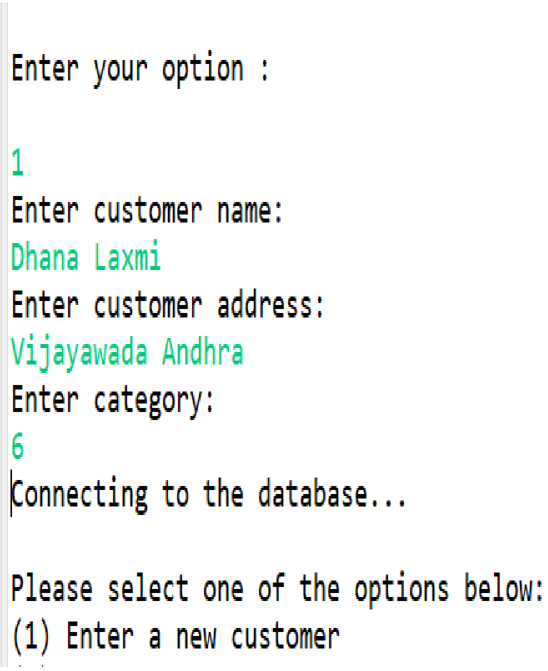
```
WELCOME TO JOB-SHOP ACCOUNTING SYSTEM !

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

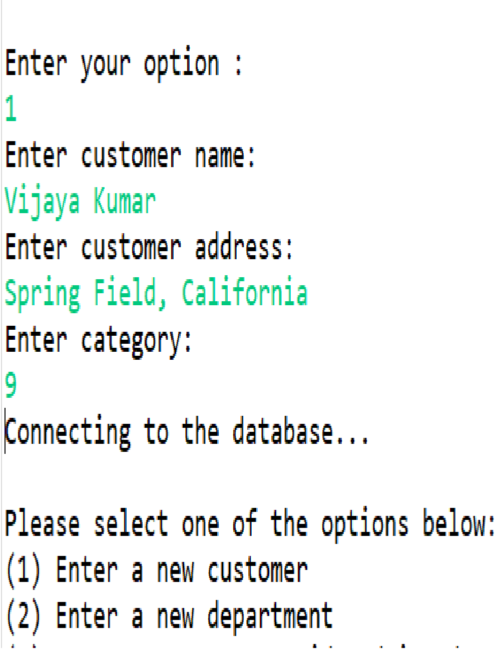
6.1: Query 1 Screenshots



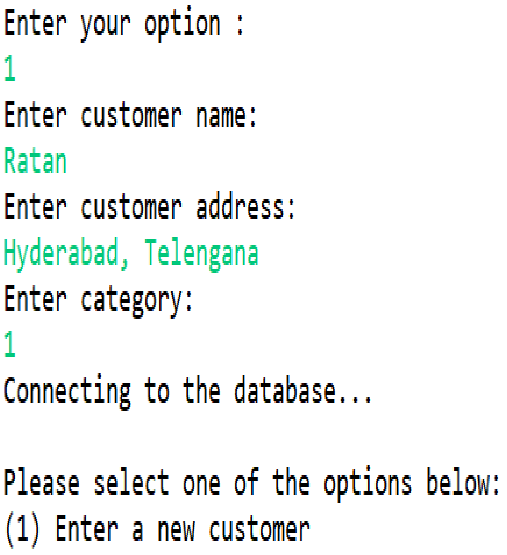
```
        to which it is applicable
(6) Enter a new job, given its job-no, assembly-id, process-id, and date the job commenced
(7) At the completion of a job, enter the date it completed and the information
        relevant to the type of job
(8) Enter a transaction-no and its sup-cost and update all 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...
```



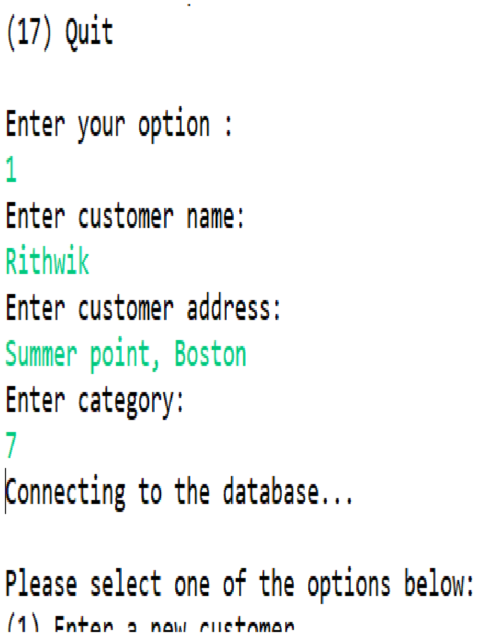
```
Enter your option :

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
```

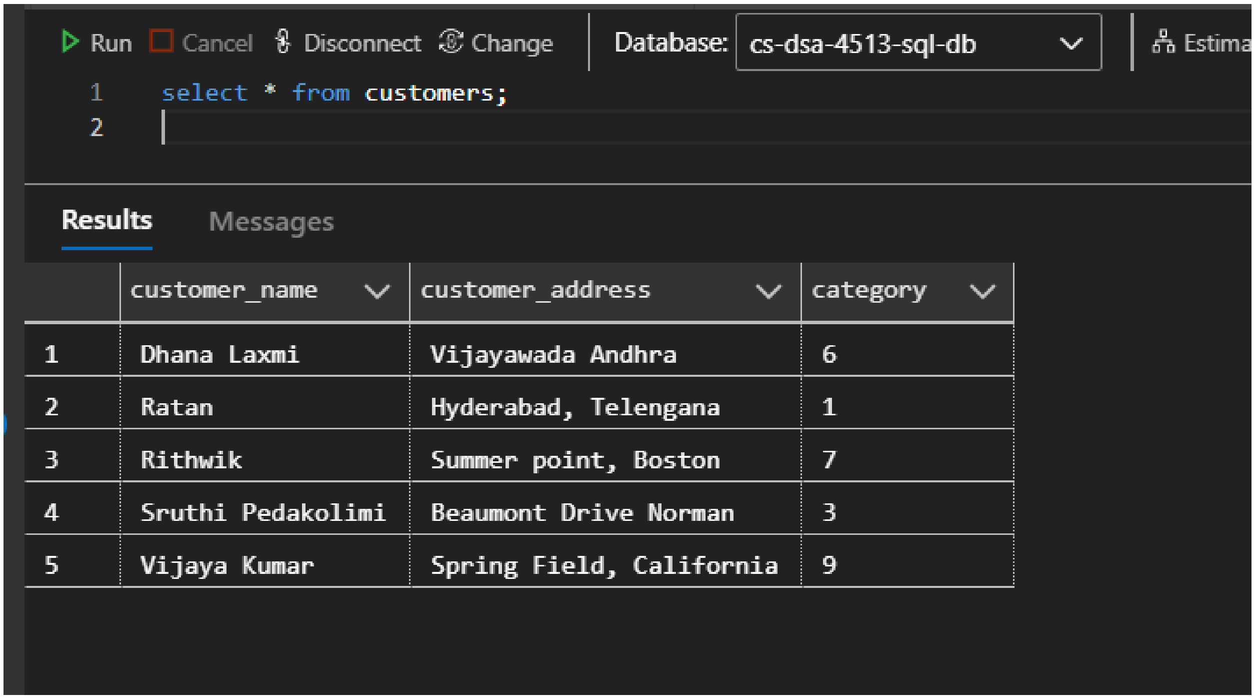


```
Enter your option :
1
Enter customer name:
Ratan
Enter customer address:
Hyderabad, Telengana
Enter category:
1
Connecting to the database...

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



```
(17) Quit

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
```

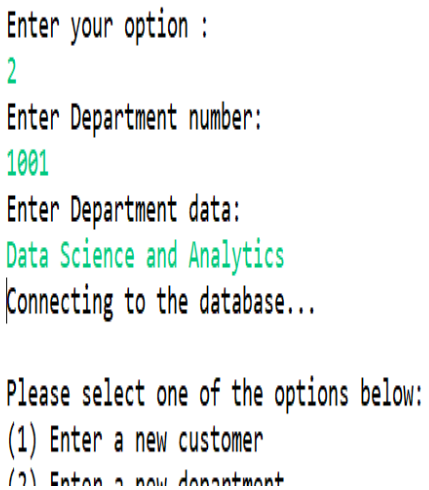
Customer Table after 5 insertions



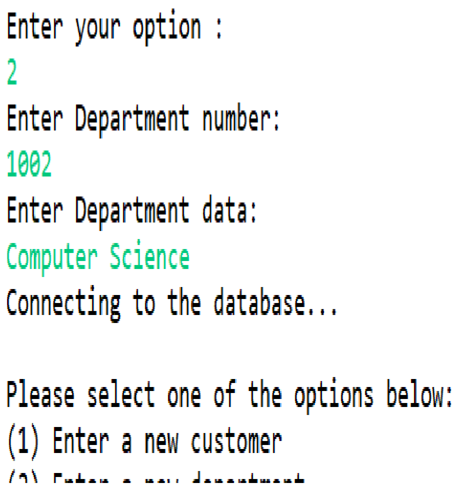
```
select * from customers;
```

	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

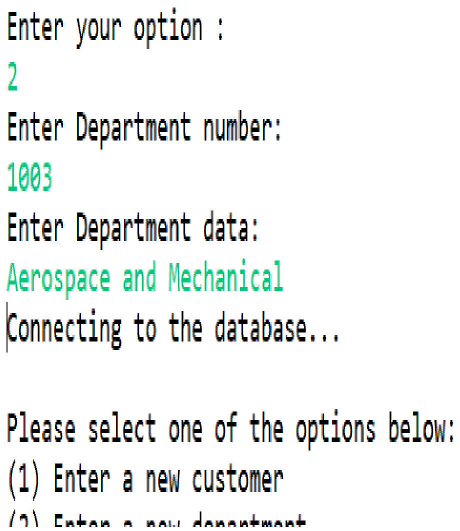
6.2: Query 2 Screenshots



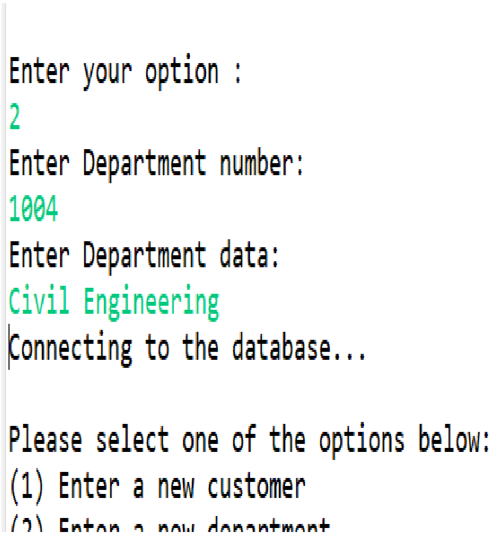
```
Enter your option :
2
Enter Department number:
1001
Enter Department data:
Data Science and Analytics
Connecting to the database...

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



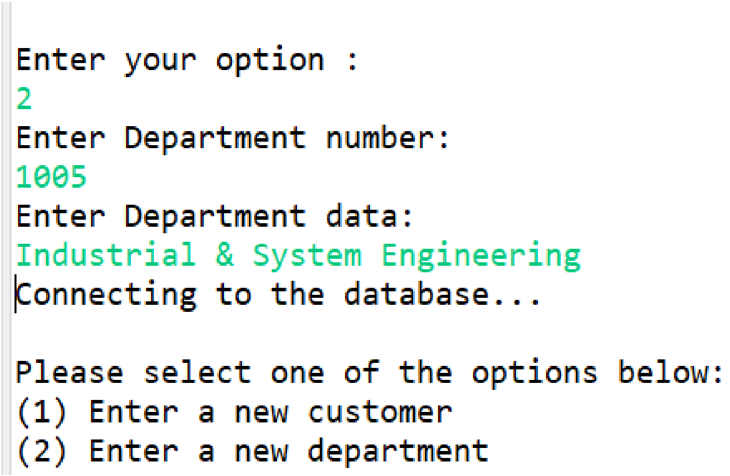
```
Enter your option :
2
Enter Department number:
1002
Enter Department data:
Computer Science
Connecting to the database...

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
```

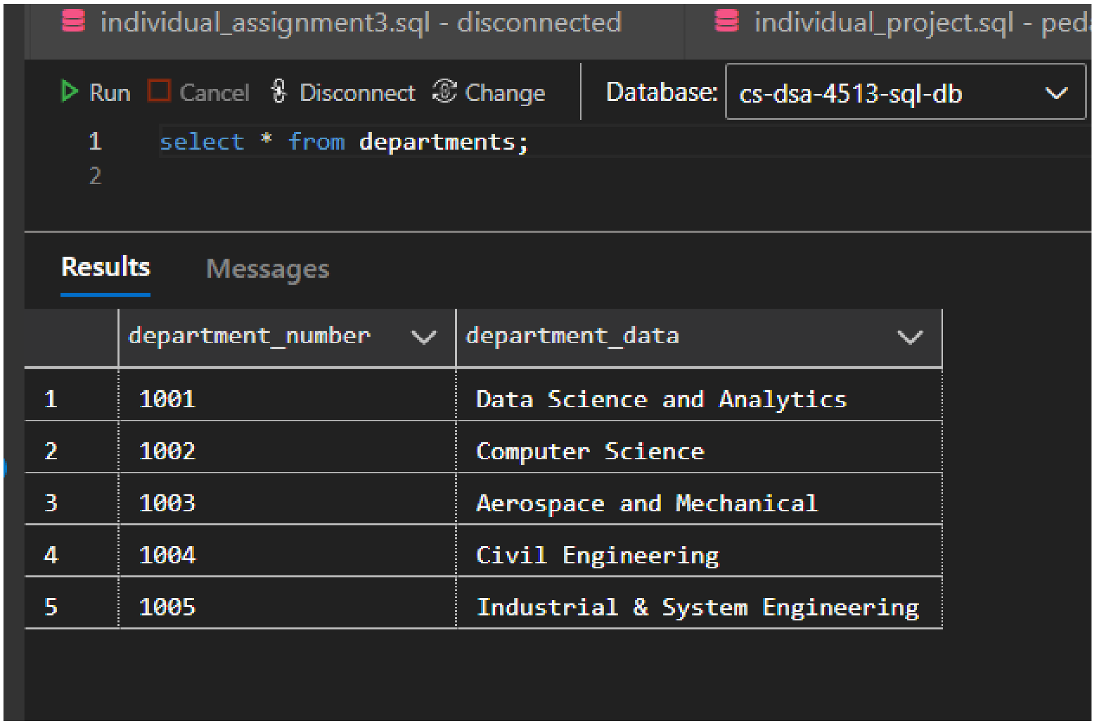


```
Enter your option :
2
Enter Department number:
1004
Enter Department data:
Civil Engineering
Connecting to the database...

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



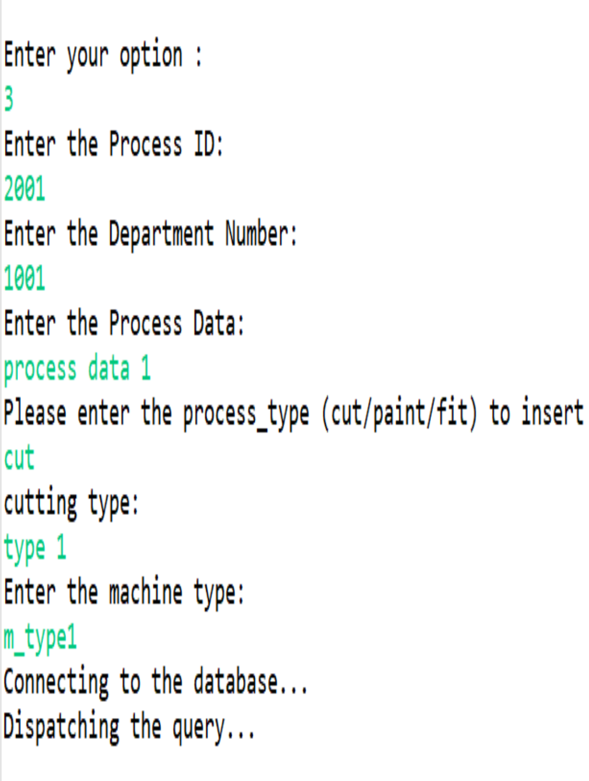
individual_assignment3.sql - disconnected
individual_project.sql - ped
Run
Cancel
Disconnect
Change
Results
Messages

Database: cs-dsa-4513-sql-db

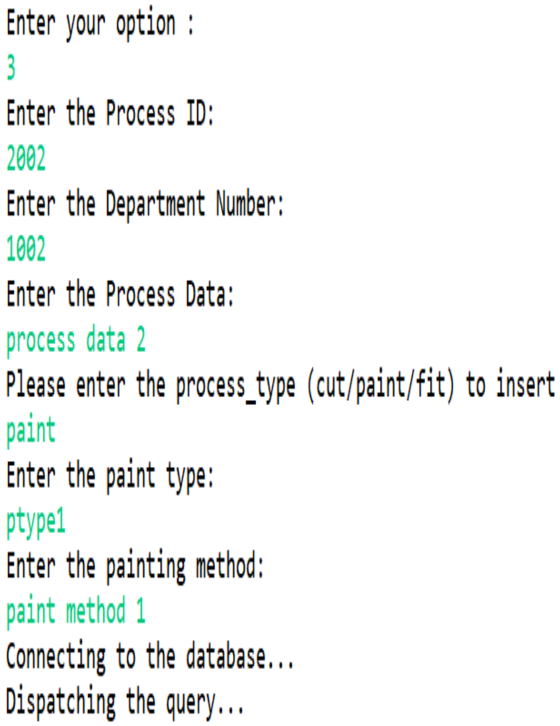
```
select * from departments;
```

	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

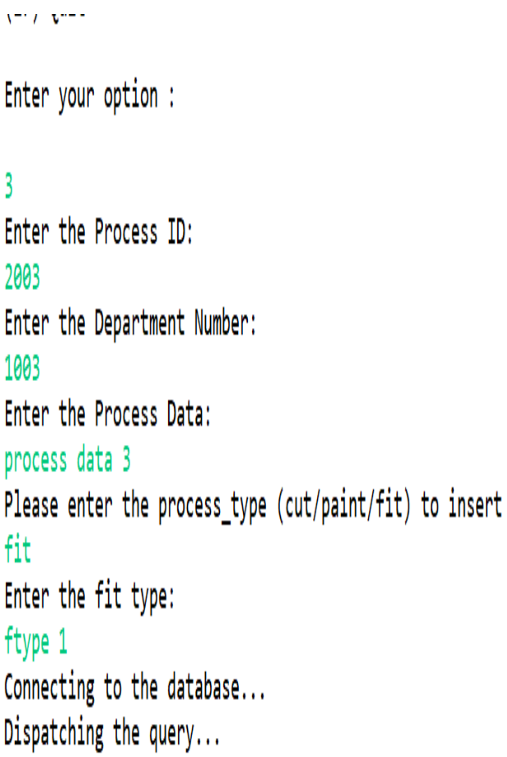
6.3: Query 3 Screenshots



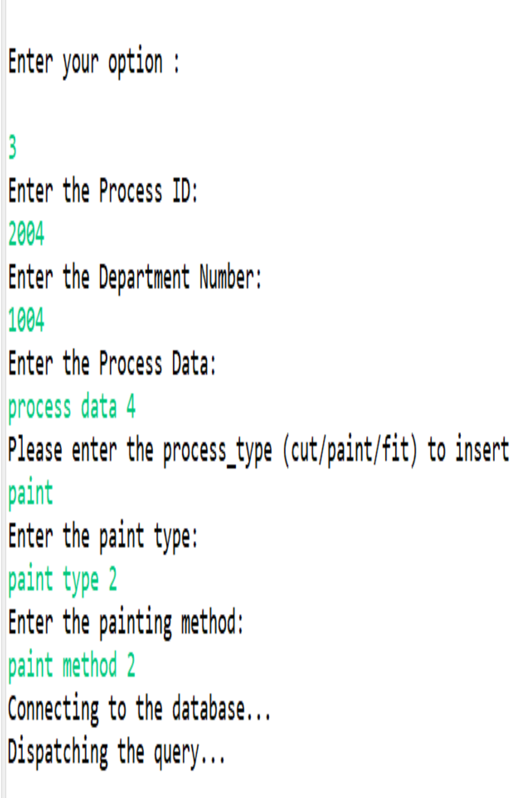
```
Enter your option :
3
Enter the Process ID:
2001
Enter the Department Number:
1001
Enter the Process Data:
process data 1
Please enter the process_type (cut/paint/fit) to insert
cut
cutting type:
type 1
Enter the machine type:
m_type1
Connecting to the database...
Dispatching the query...
```



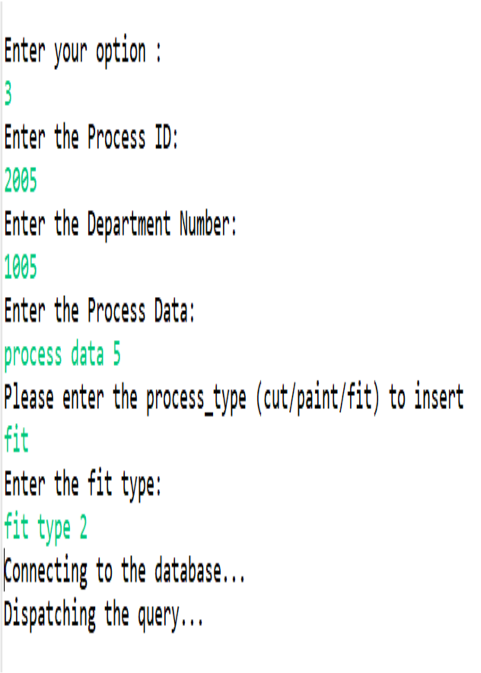
```
Enter your option :
3
Enter the Process ID:
2002
Enter the Department Number:
1002
Enter the Process Data:
process data 2
Please enter the process_type (cut/paint/fit) to insert
paint
Enter the paint type:
ptype1
Enter the painting method:
paint method 1
Connecting to the database...
Dispatching the query...
```



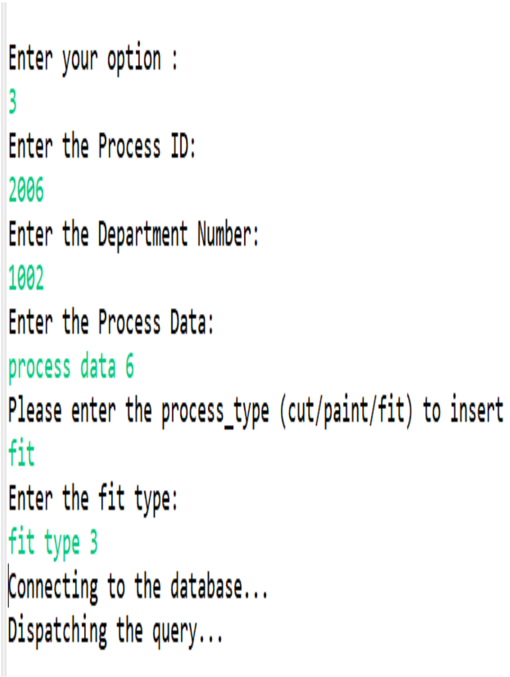
```
Enter your option :

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...
```



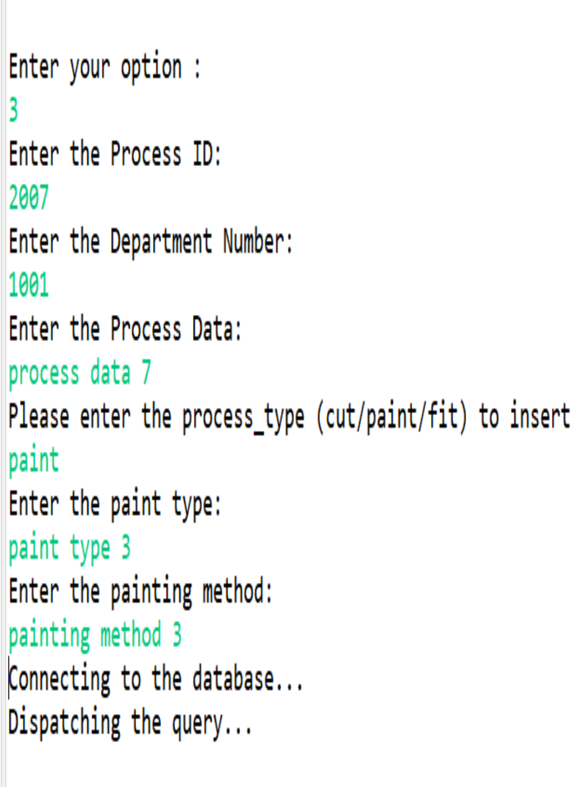
```
Enter your option :

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...
```



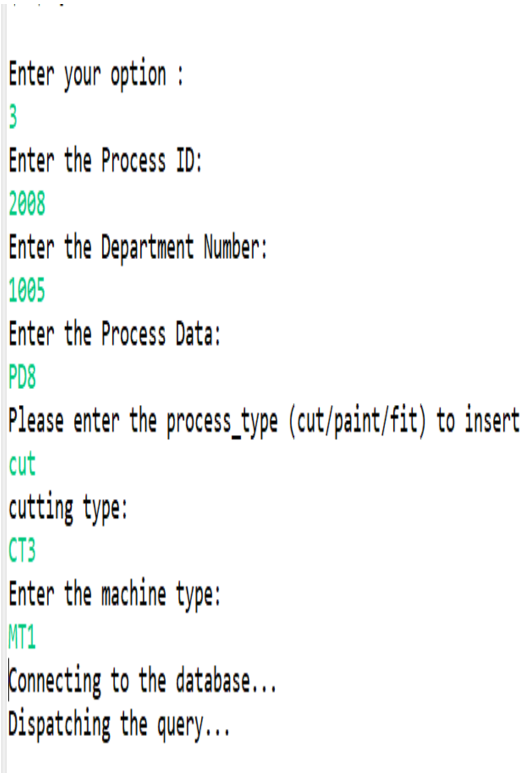
```
Enter your option :
3
Enter the Process ID:
2005
Enter the Department Number:
1005
Enter the Process Data:
process data 5
Please enter the process_type (cut/paint/fit) to insert
fit
Enter the fit type:
fit type 2
Connecting to the database...
Dispatching the query...
```



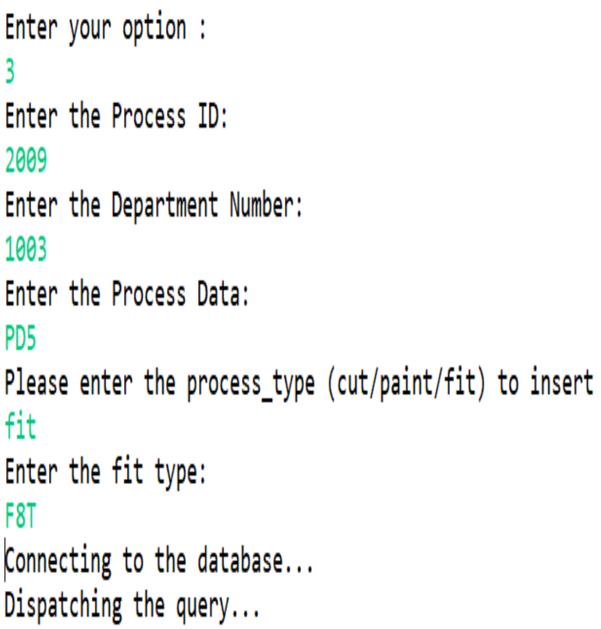
```
Enter your option :
3
Enter the Process ID:
2006
Enter the Department Number:
1002
Enter the Process Data:
process data 6
Please enter the process_type (cut/paint/fit) to insert
fit
Enter the fit type:
fit type 3
Connecting to the database...
Dispatching the query...
```



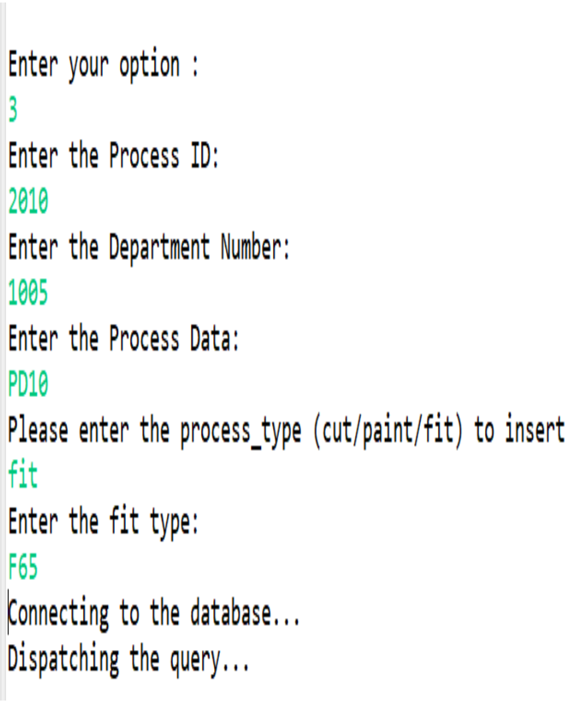
```
Enter your option :
3
Enter the Process ID:
2007
Enter the Department Number:
1001
Enter the Process Data:
process data 7
Please enter the process_type (cut/paint/fit) to insert
paint
Enter the paint type:
paint type 3
Enter the painting method:
painting method 3
Connecting to the database...
Dispatching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

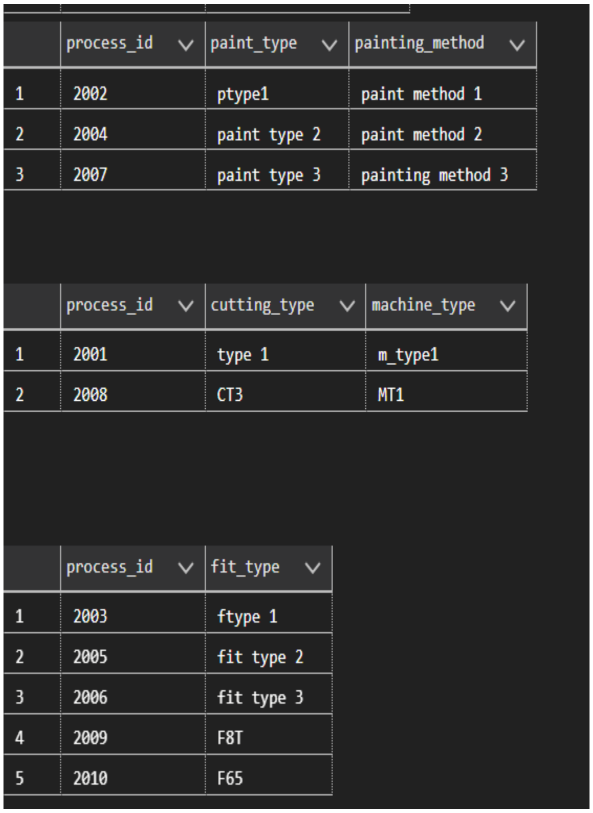


```
Enter your option :
3
Enter the Process ID:
2009
Enter the Department Number:
1003
Enter the Process Data:
PD5
Please enter the process_type (cut/paint/fit) to insert
fit
Enter the fit type:
F8T
Connecting to the database...
Dispatching the query...
```



```
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...
```

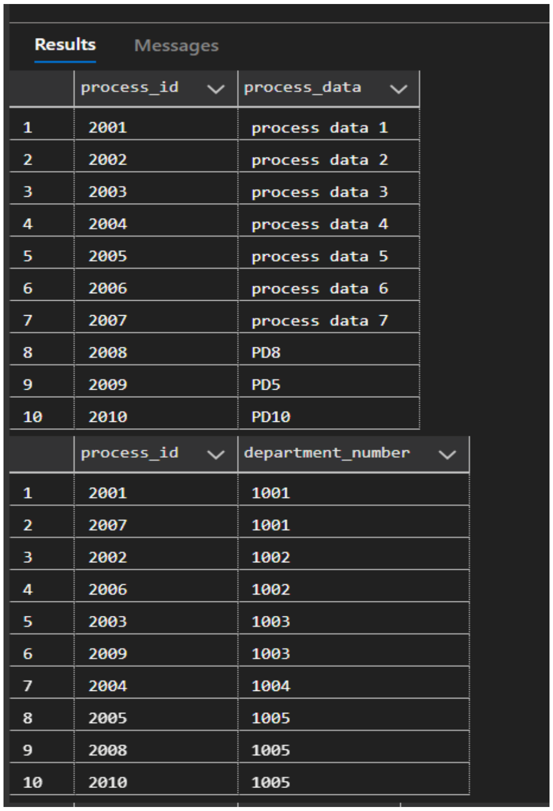
Output of tables involved



	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

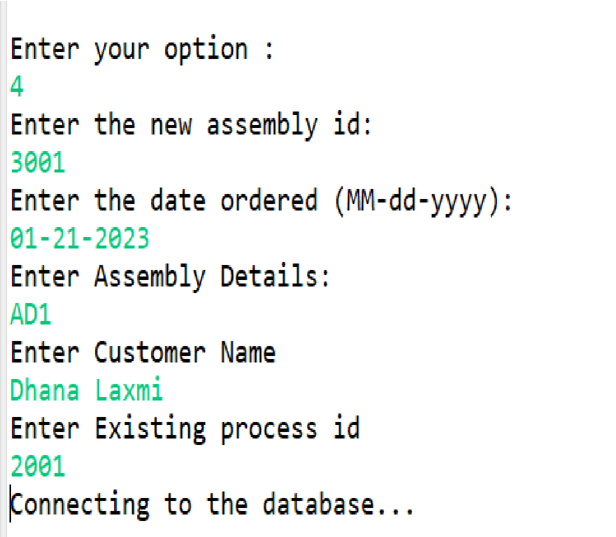
	process_id	fit_type
1	2003	ftype 1
2	2005	fit type 2
3	2006	fit type 3
4	2009	F8T
5	2010	F65



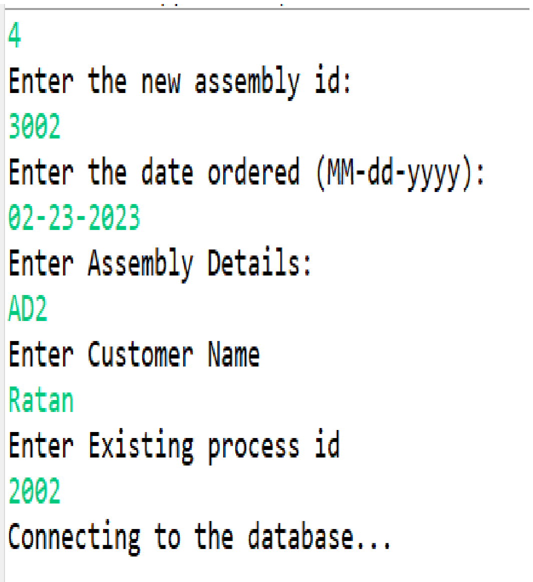
	process_id	process_data
1	2001	process data 1
2	2002	process data 2
3	2003	process data 3
4	2004	process data 4
5	2005	process data 5
6	2006	process data 6
7	2007	process data 7
8	2008	PD8
9	2009	PD5
10	2010	PD10

	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

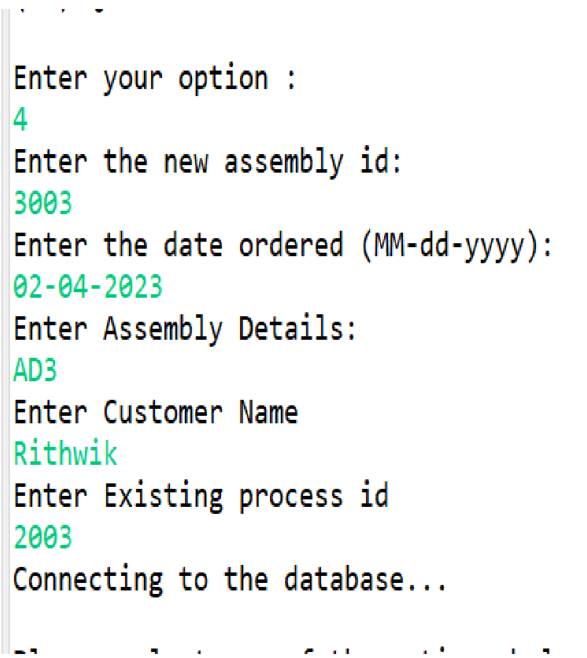
6.4: Query 4 Screenshots



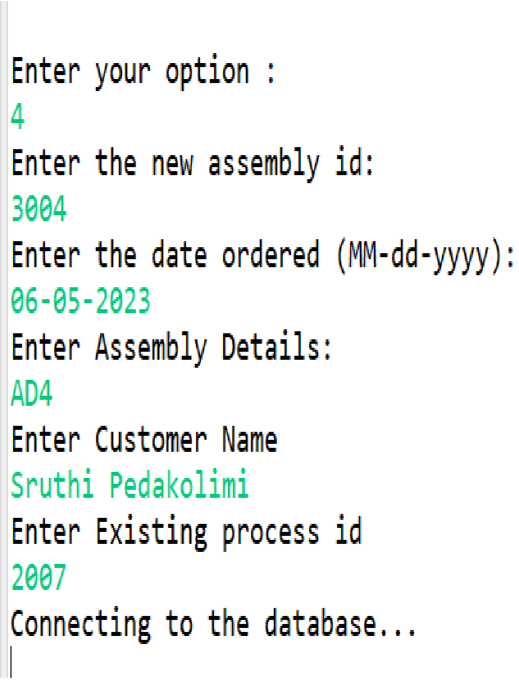
```
Enter your option :
4
Enter the new assembly id:
3001
Enter the date ordered (MM-dd-yyyy):
01-21-2023
Enter Assembly Details:
AD1
Enter Customer Name
Dhana Laxmi
Enter Existing process id
2001
Connecting to the database...
```



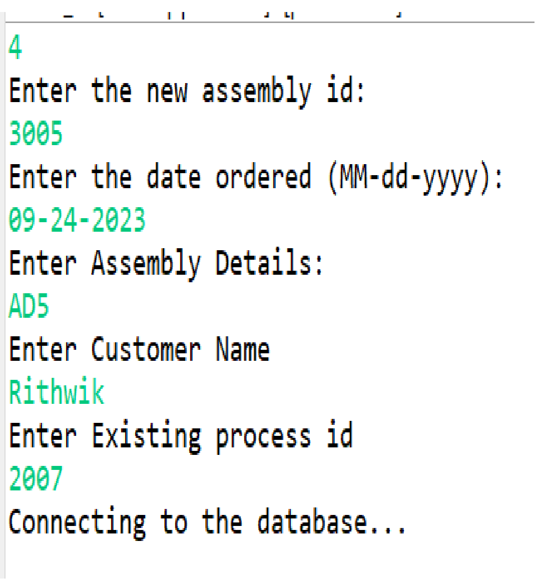
```
4
Enter the new assembly id:
3002
Enter the date ordered (MM-dd-yyyy):
02-23-2023
Enter Assembly Details:
AD2
Enter Customer Name
Ratan
Enter Existing process id
2002
Connecting to the database...
```



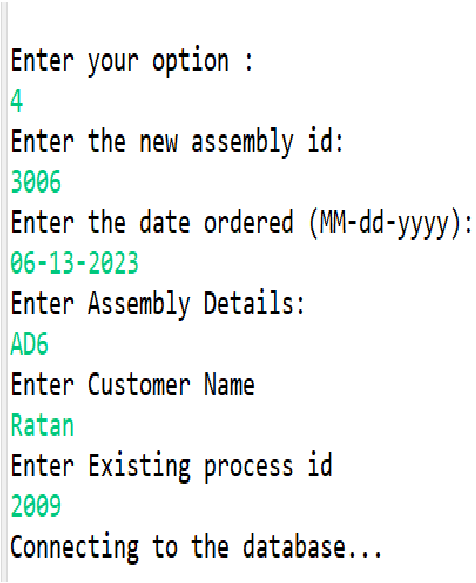
```
Enter your option :
4
Enter the new assembly id:
3003
Enter the date ordered (MM-dd-yyyy):
02-04-2023
Enter Assembly Details:
AD3
Enter Customer Name
Rithwik
Enter Existing process id
2003
Connecting to the database...
```



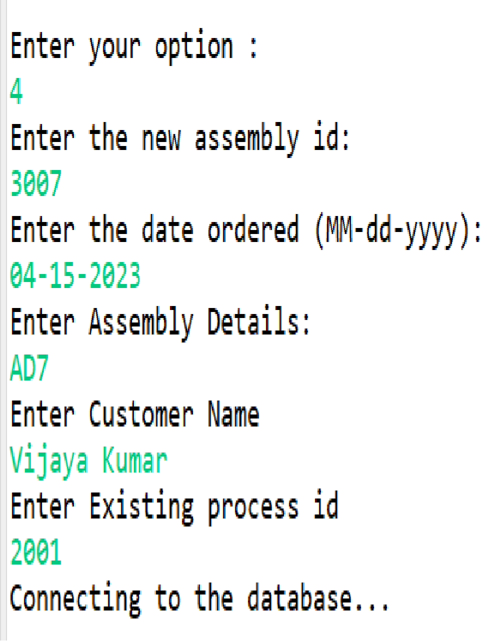
```
Enter your option :
4
Enter the new assembly id:
3004
Enter the date ordered (MM-dd-yyyy):
06-05-2023
Enter Assembly Details:
AD4
Enter Customer Name
Sruthi Pedakolimi
Enter Existing process id
2007
Connecting to the database...
```



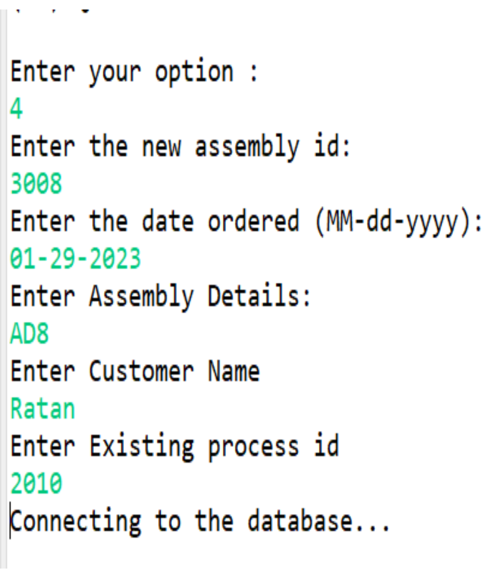
```
4
Enter the new assembly id:
3005
Enter the date ordered (MM-dd-yyyy):
09-24-2023
Enter Assembly Details:
AD5
Enter Customer Name
Rithwik
Enter Existing process id
2007
Connecting to the database...
```



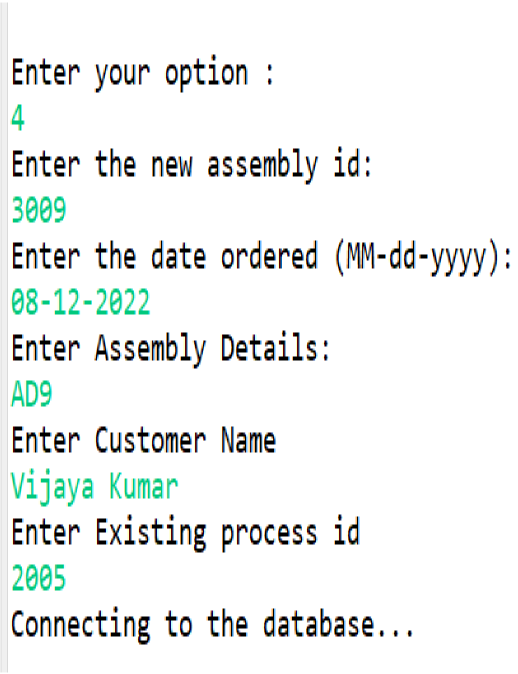
```
Enter your option :
4
Enter the new assembly id:
3006
Enter the date ordered (MM-dd-yyyy):
06-13-2023
Enter Assembly Details:
AD6
Enter Customer Name
Ratan
Enter Existing process id
2009
Connecting to the database...
```



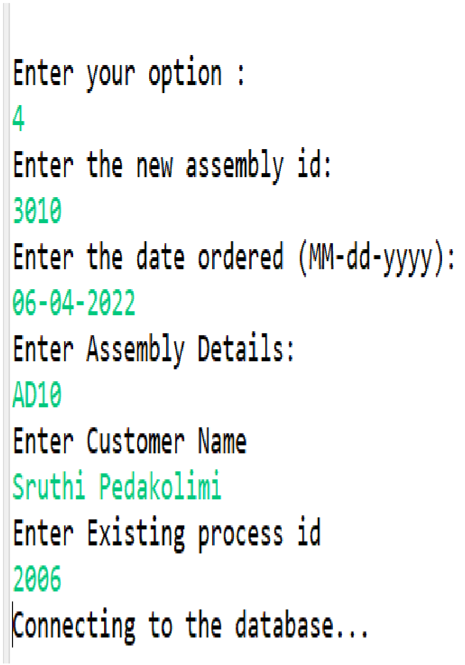
```
Enter your option :
4
Enter the new assembly id:
3007
Enter the date ordered (MM-dd-yyyy):
04-15-2023
Enter Assembly Details:
AD7
Enter Customer Name
Vijaya Kumar
Enter Existing process id
2001
Connecting to the database...
```



```
Enter your option :
4
Enter the new assembly id:
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...
```

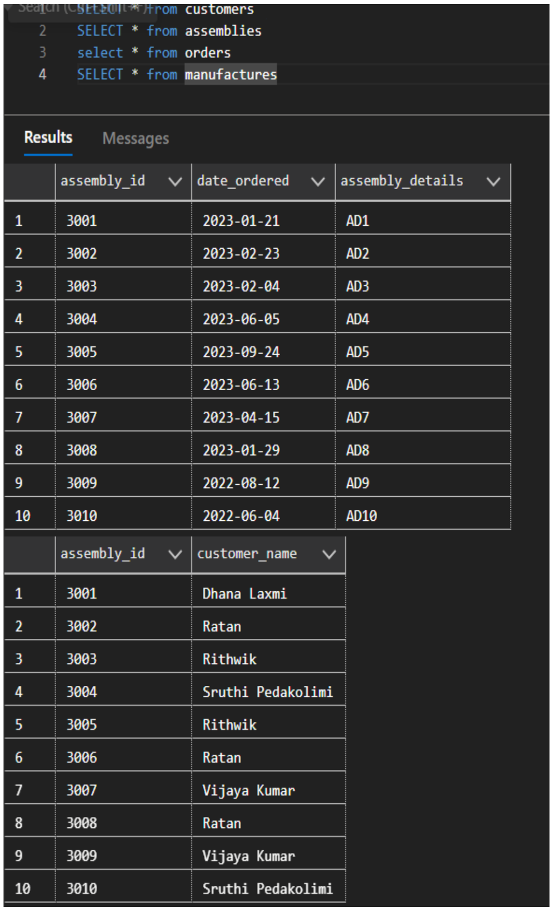


```
Enter your option :
4
Enter the new assembly id:
3009
Enter the date ordered (MM-dd-yyyy):
08-12-2022
Enter Assembly Details:
AD9
Enter Customer Name
Vijaya Kumar
Enter Existing process id
2005
Connecting to the database...
```



```
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...
```

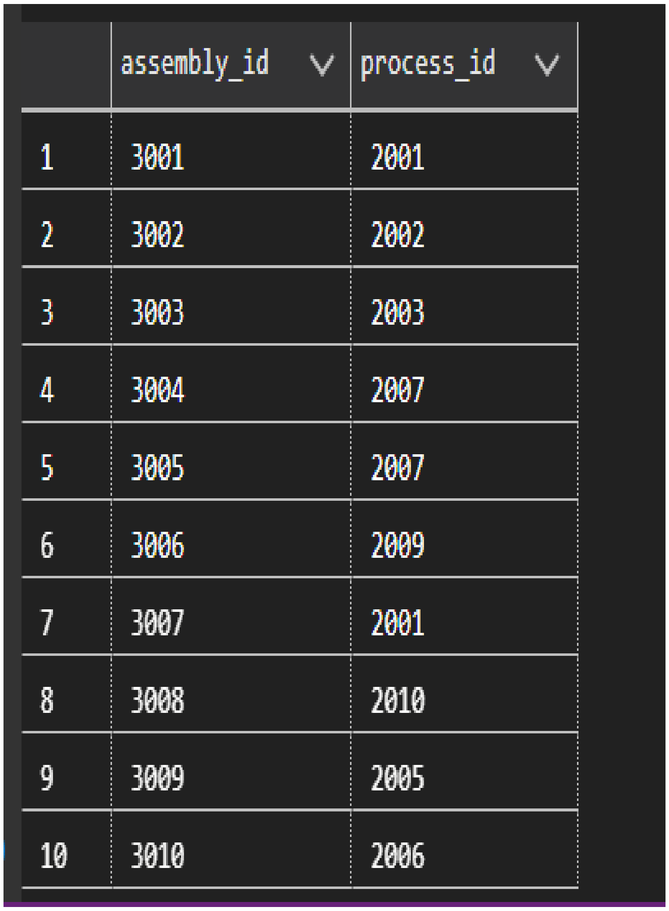
Output



```
SELECT * from customers
SELECT * from assemblies
select * from orders
SELECT * from manufactures
```

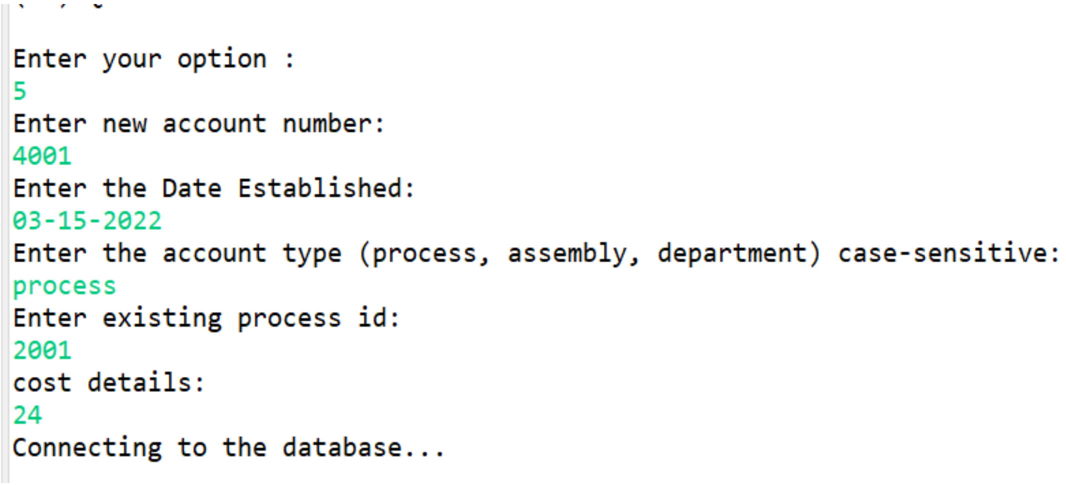
	assembly_id	date_ordered	assembly_details
1	3001	2023-01-21	AD1
2	3002	2023-02-23	AD2
3	3003	2023-02-04	AD3
4	3004	2023-06-05	AD4
5	3005	2023-09-24	AD5
6	3006	2023-06-13	AD6
7	3007	2023-04-15	AD7
8	3008	2023-01-29	AD8
9	3009	2022-08-12	AD9
10	3010	2022-06-04	AD10

	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

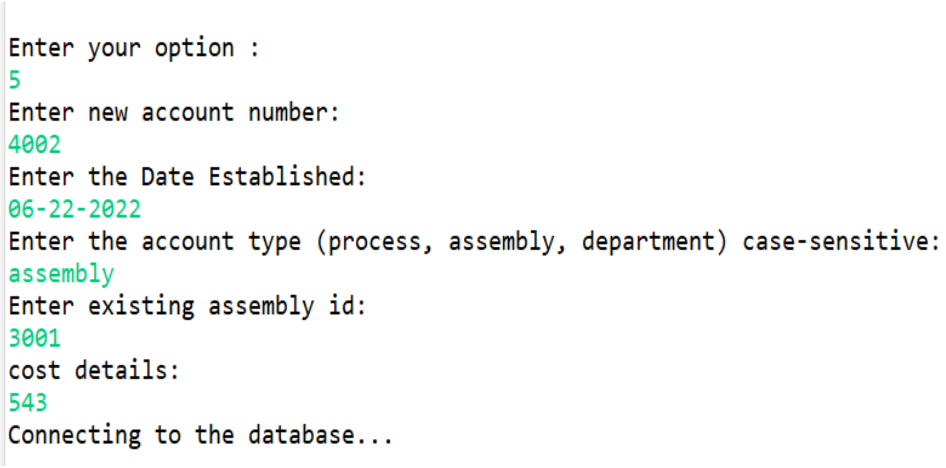


	assembly_id	process_id
1	3001	2001
2	3002	2002
3	3003	2003
4	3004	2007
5	3005	2007
6	3006	2009
7	3007	2001
8	3008	2010
9	3009	2005
10	3010	2006

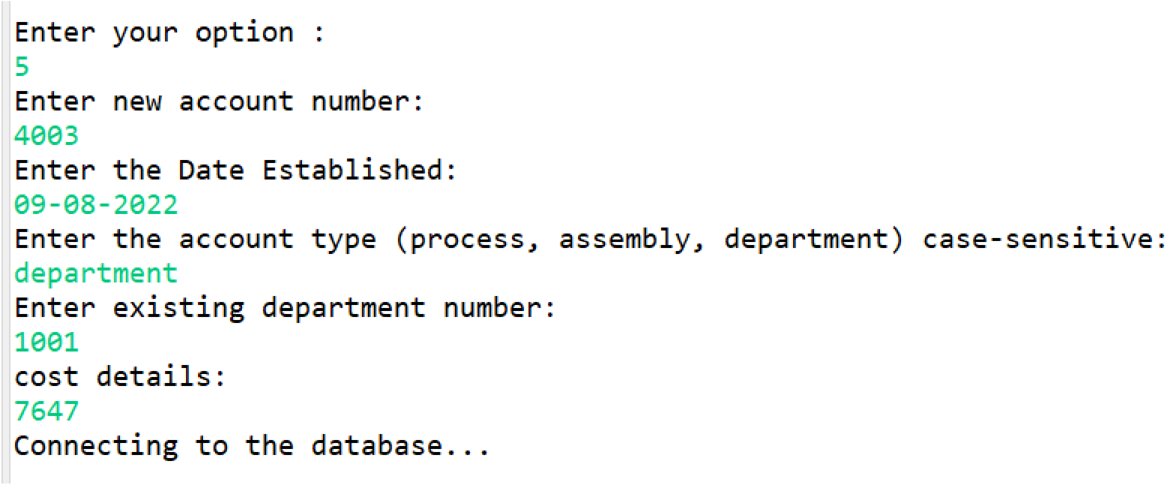
6.5: Query 5 Screenshots



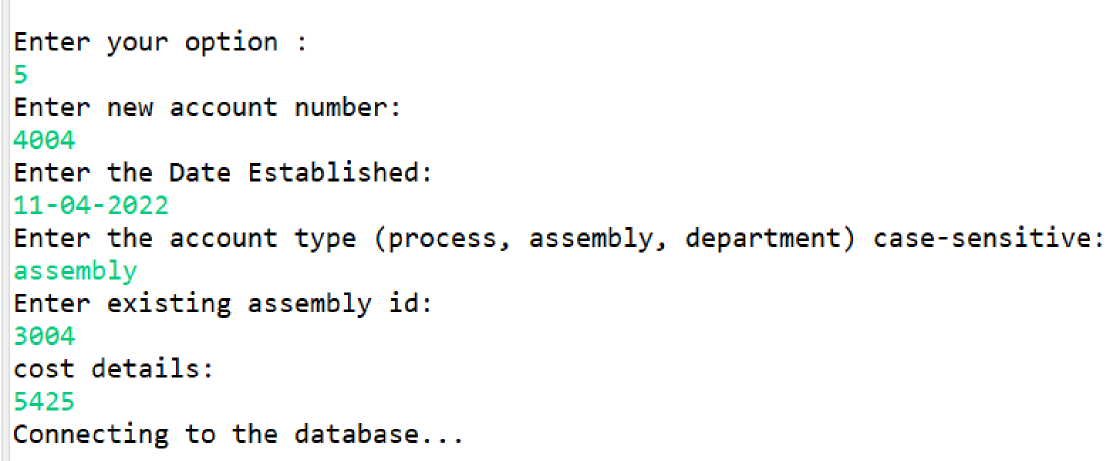
```
Enter your option :
5
Enter new account number:
4001
Enter the Date Established:
03-15-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2001
cost details:
24
Connecting to the database...
```



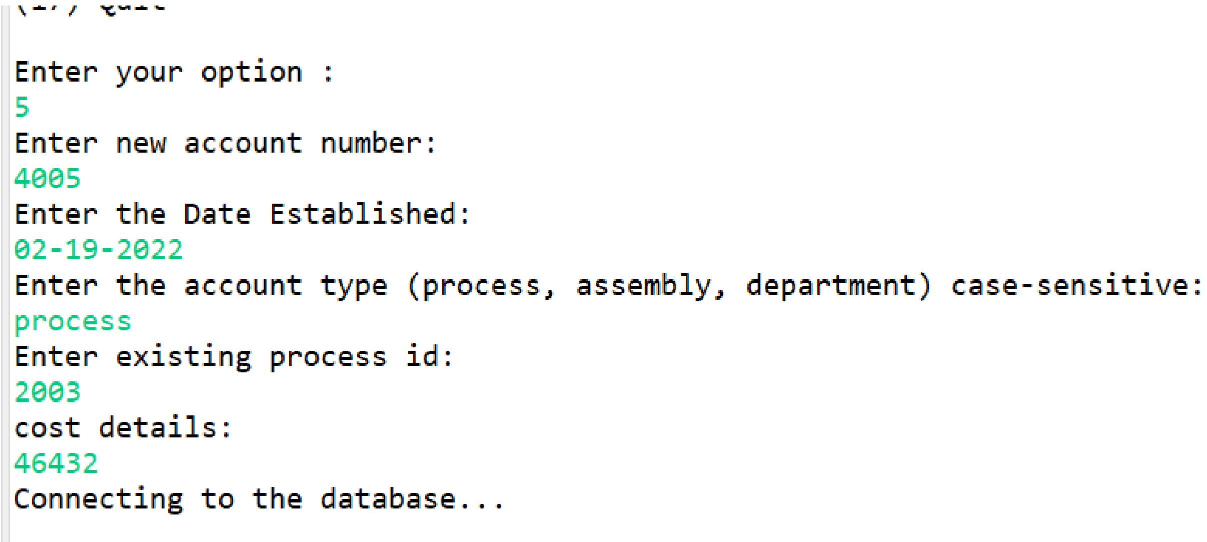
```
Enter your option :
5
Enter new account number:
4002
Enter the Date Established:
06-22-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3001
cost details:
543
Connecting to the database...
```



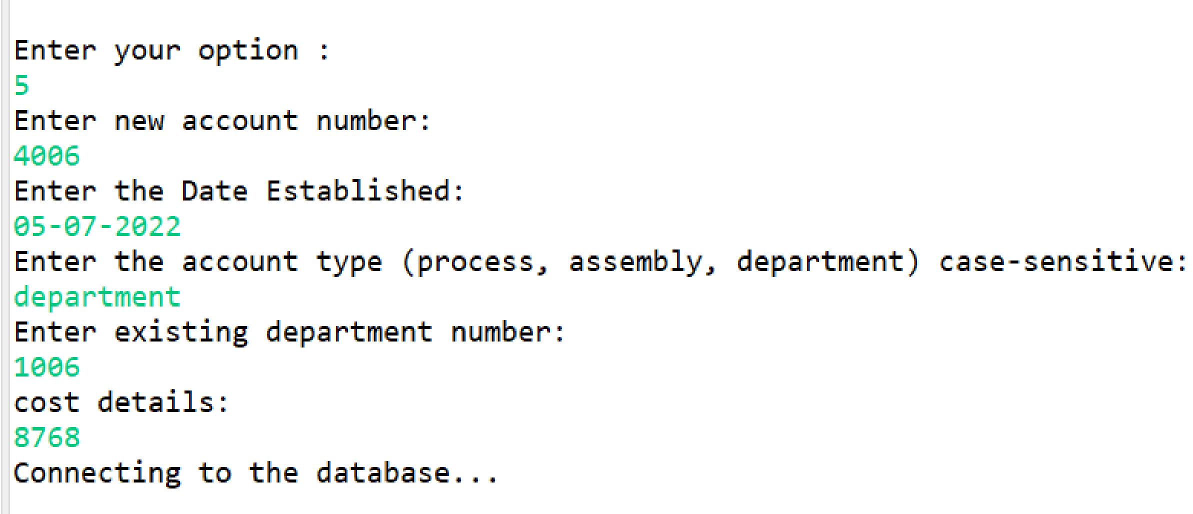
```
Enter your option :
5
Enter new account number:
4003
Enter the Date Established:
09-08-2022
Enter the account type (process, assembly, department) case-sensitive:
department
Enter existing department number:
1001
cost details:
7647
Connecting to the database...
```



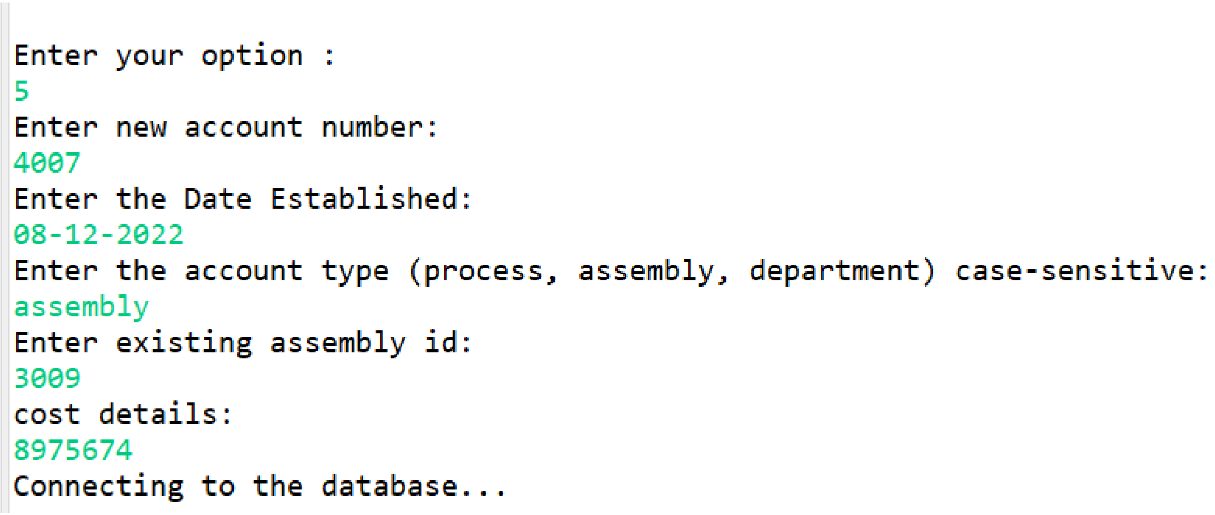
```
Enter your option :
5
Enter new account number:
4004
Enter the Date Established:
11-04-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3004
cost details:
5425
Connecting to the database...
```



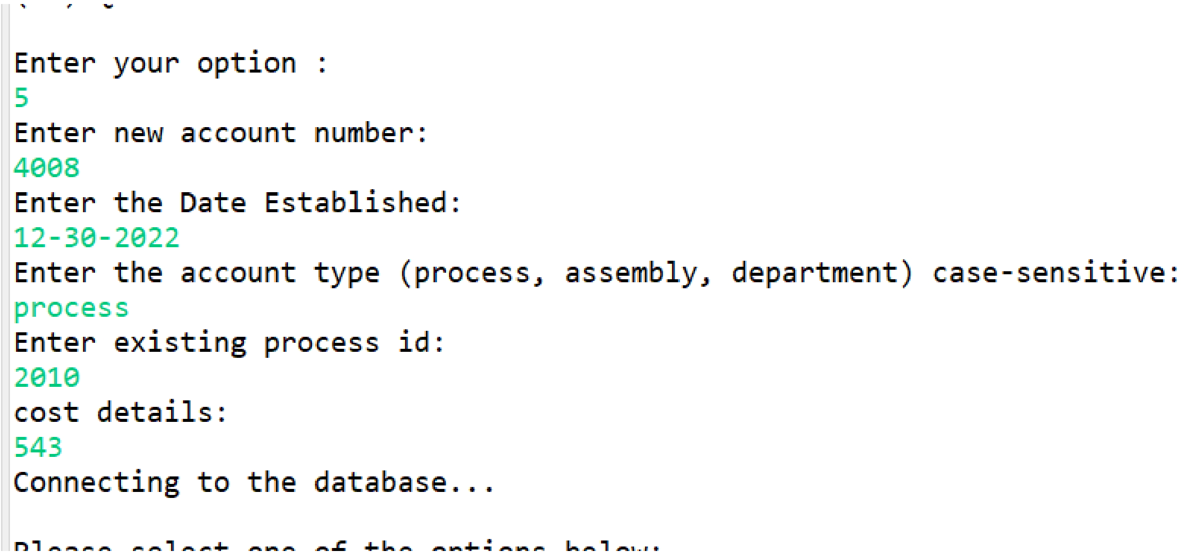
```
(17) Quit

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...
```



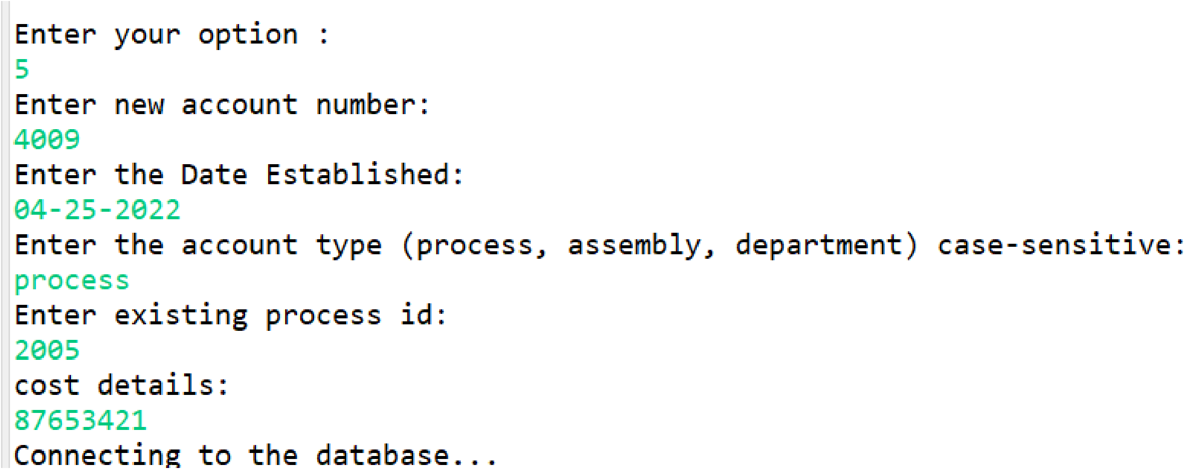
```
Enter your option :
5
Enter new account number:
4006
Enter the Date Established:
05-07-2022
Enter the account type (process, assembly, department) case-sensitive:
department
Enter existing department number:
1006
cost details:
8768
Connecting to the database...
```



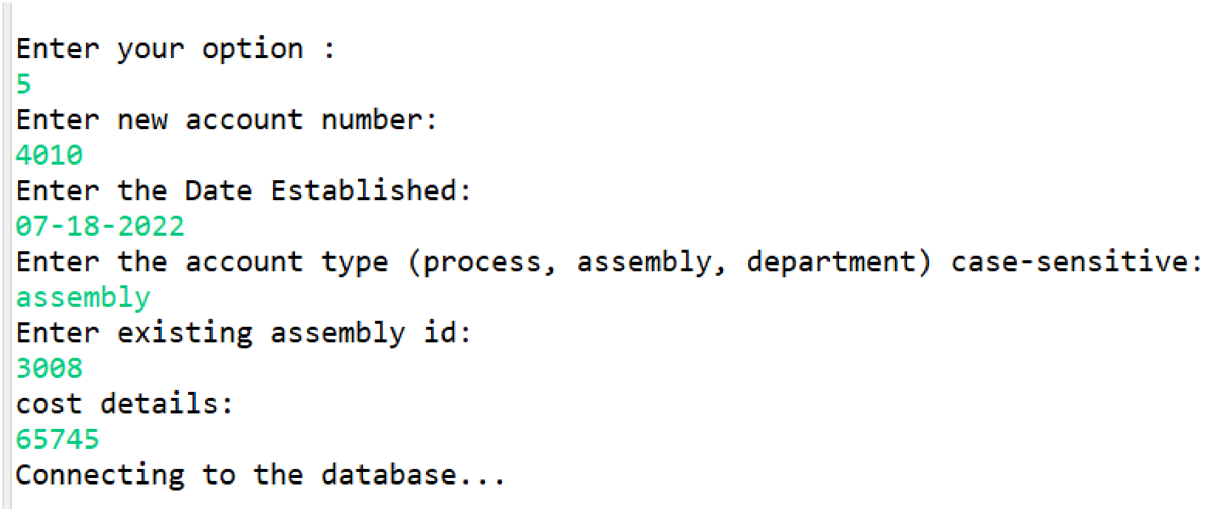
```
Enter your option :
5
Enter new account number:
4007
Enter the Date Established:
08-12-2022
Enter the account type (process, assembly, department) case-sensitive:
assembly
Enter existing assembly id:
3009
cost details:
8975674
Connecting to the database...
```



```
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:
```

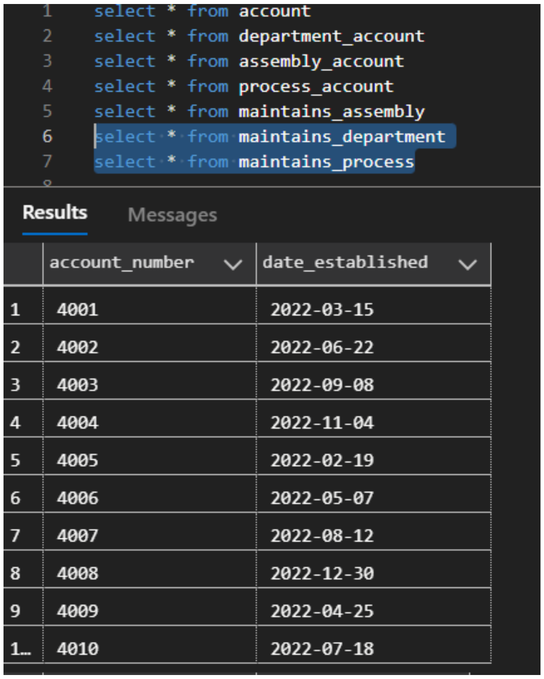


```
Enter your option :
5
Enter new account number:
4009
Enter the Date Established:
04-25-2022
Enter the account type (process, assembly, department) case-sensitive:
process
Enter existing process id:
2005
cost details:
87653421
Connecting to the database...
```



```
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...
```

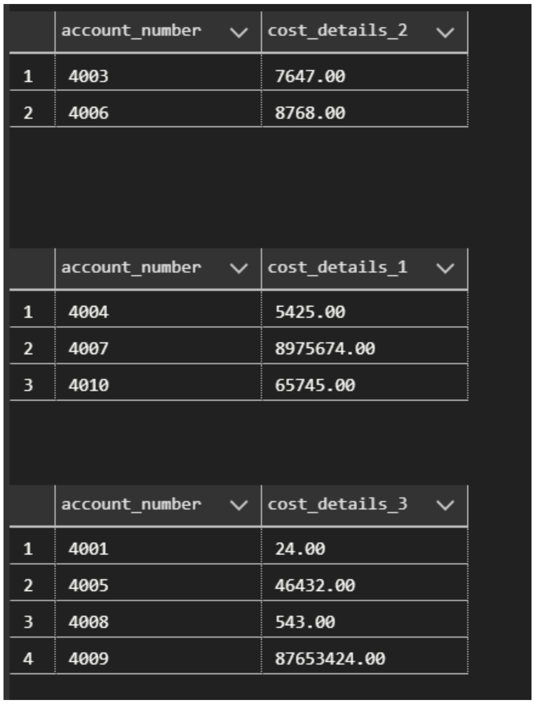
Tables that are changed:
Account Table



```
select * from account
select * from department_account
select * from assembly_account
select * from process_account
select * from maintains_assembly
select * from maintains_department
select * from maintains_process
```

	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

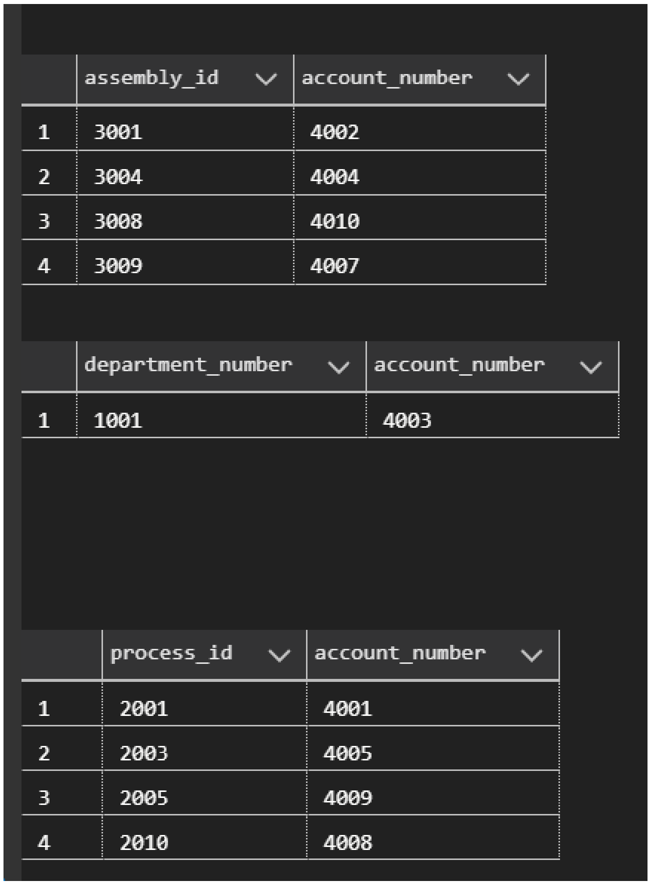


	account_number	cost_details_2
1	4003	7647.00
2	4006	8768.00

	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

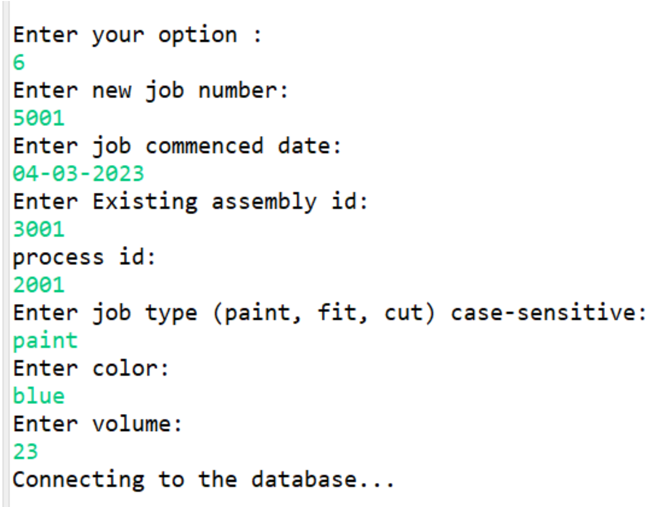


	assembly_id	account_number
1	3001	4002
2	3004	4004
3	3008	4010
4	3009	4007

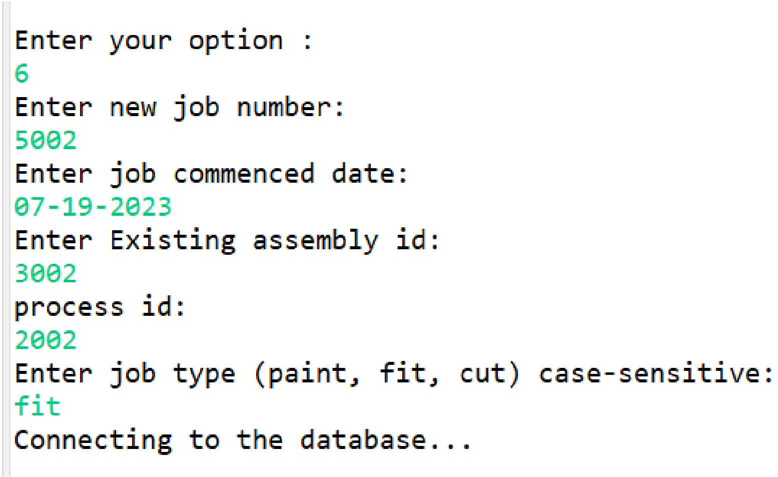
	department_number	account_number
1	1001	4003

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

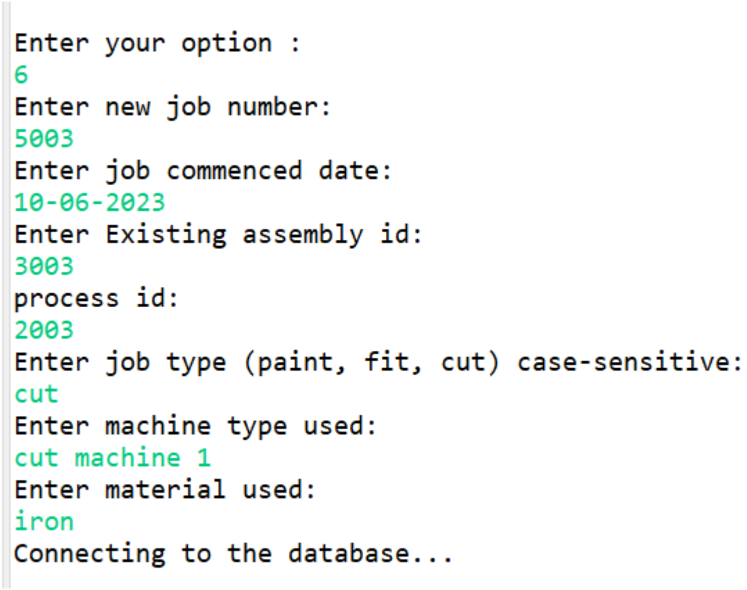
6.6: Query 6 Screenshots



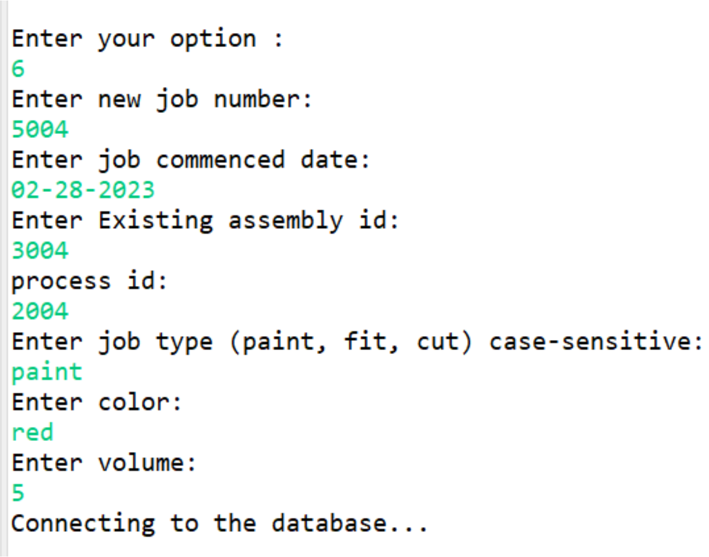
```
Enter your option :
6
Enter new job number:
5001
Enter job commenced date:
04-03-2023
Enter Existing assembly id:
3001
process id:
2001
Enter job type (paint, fit, cut) case-sensitive:
paint
Enter color:
blue
Enter volume:
23
Connecting to the database...
```



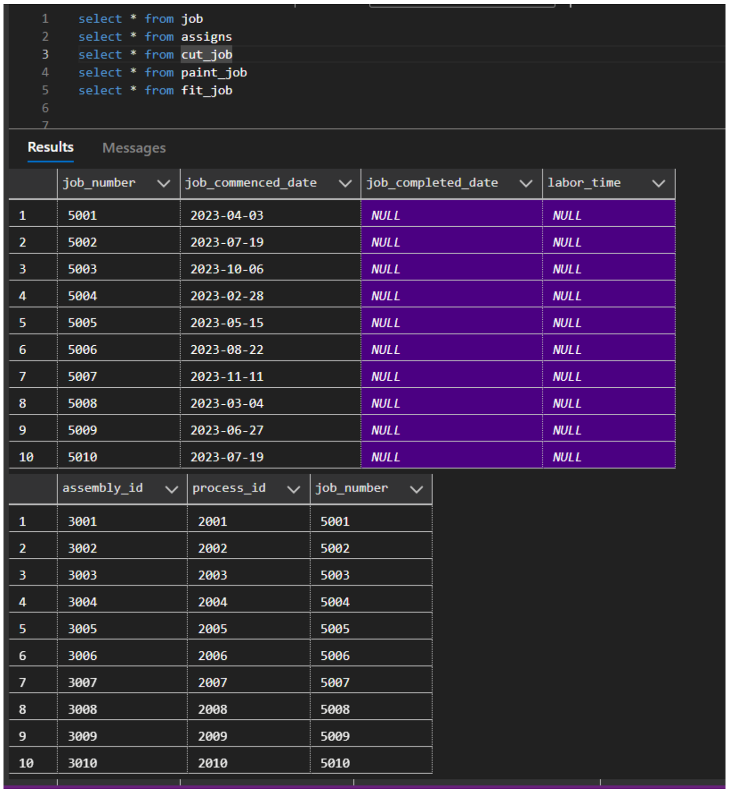
```
Enter your option :
6
Enter new job number:
5002
Enter job commenced date:
07-19-2023
Enter Existing assembly id:
3002
process id:
2002
Enter job type (paint, fit, cut) case-sensitive:
fit
Connecting to the database...
```



```
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...
```



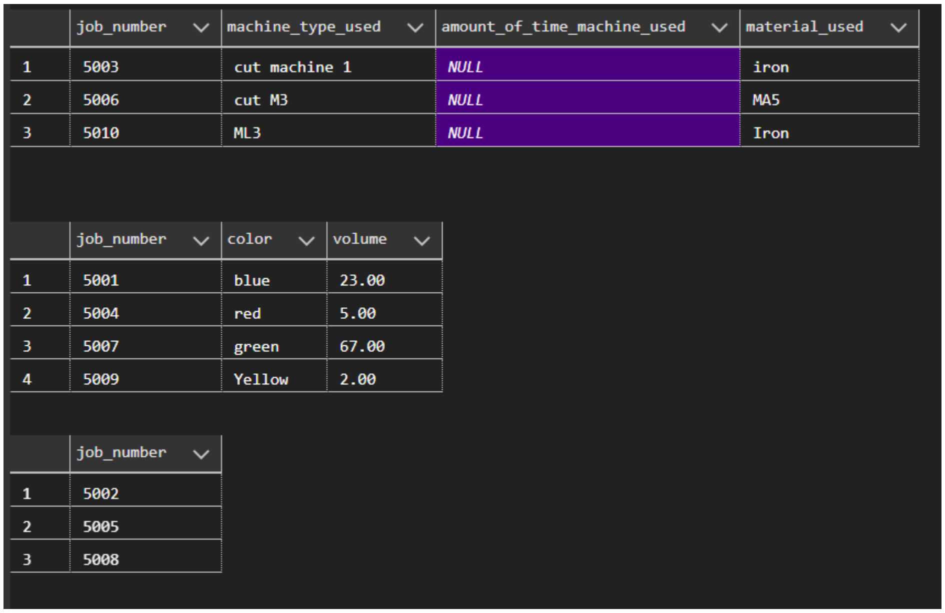
```
Enter your option :
6
Enter new job number:
5004
Enter job commenced date:
02-28-2023
Enter Existing assembly id:
3004
process id:
2004
Enter job type (paint, fit, cut) case-sensitive:
paint
Enter color:
red
Enter volume:
5
Connecting to the database...
```



```sql
select * from job
select * from assigns
select * from cut_job
select * from paint_job
select * from fit_job
```

	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

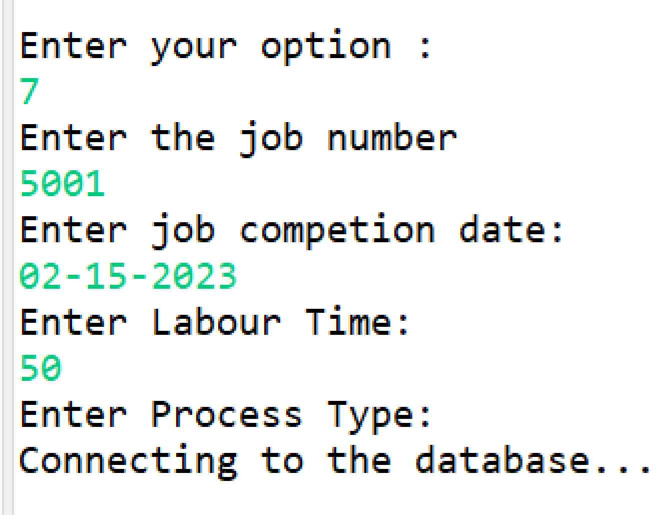


	job_number	machine_type_used	amount_of_time_machine_used	material_used
1	5003	cut machine 1	NULL	iron
2	5006	cut M3	NULL	MA5
3	5010	ML3	NULL	Iron

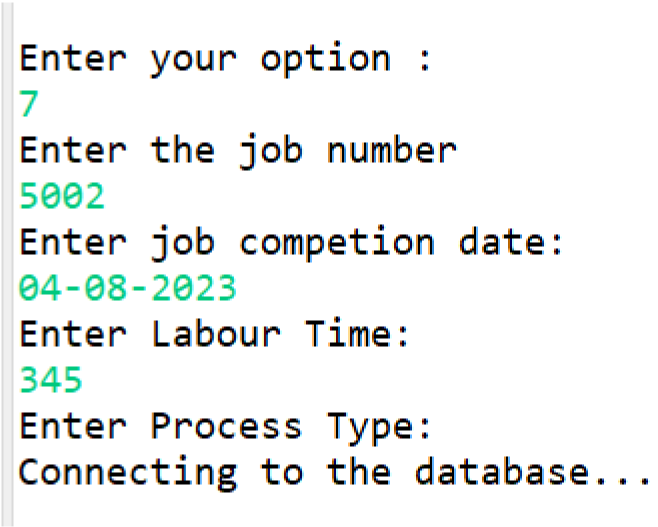
	job_number	color	volume
1	5001	blue	23.00
2	5004	red	5.00
3	5007	green	67.00
4	5009	Yellow	2.00

	job_number
1	5002
2	5005
3	5008

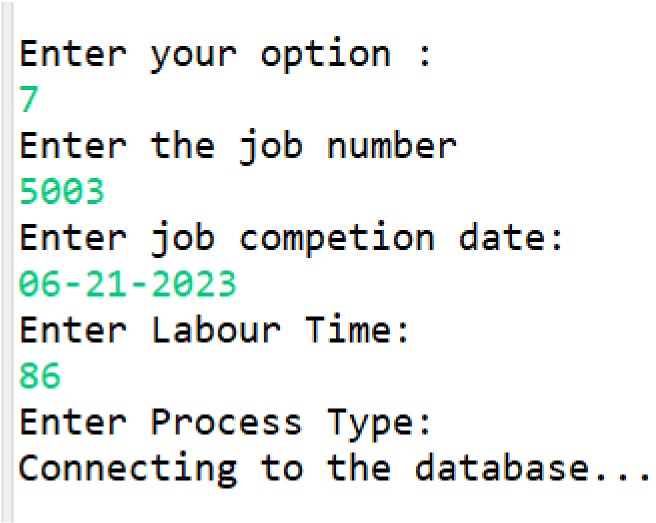
6.7: Query 7 Screenshots



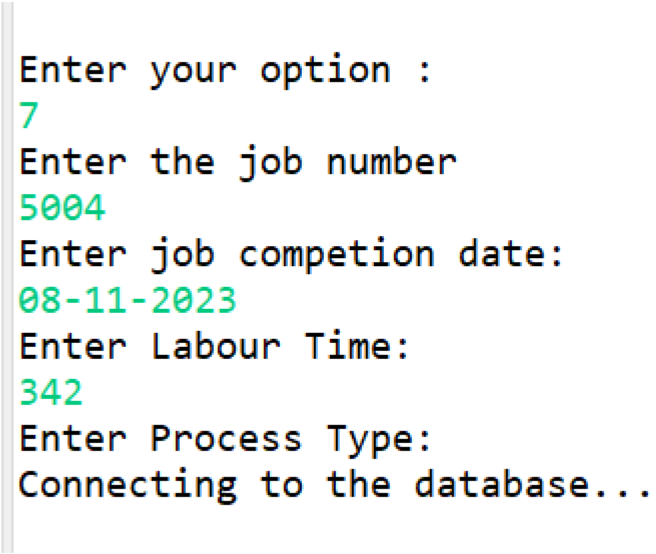
```
Enter your option :
7
Enter the job number
5001
Enter job competion date:
02-15-2023
Enter Labour Time:
50
Enter Process Type:
Connecting to the database...
```



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



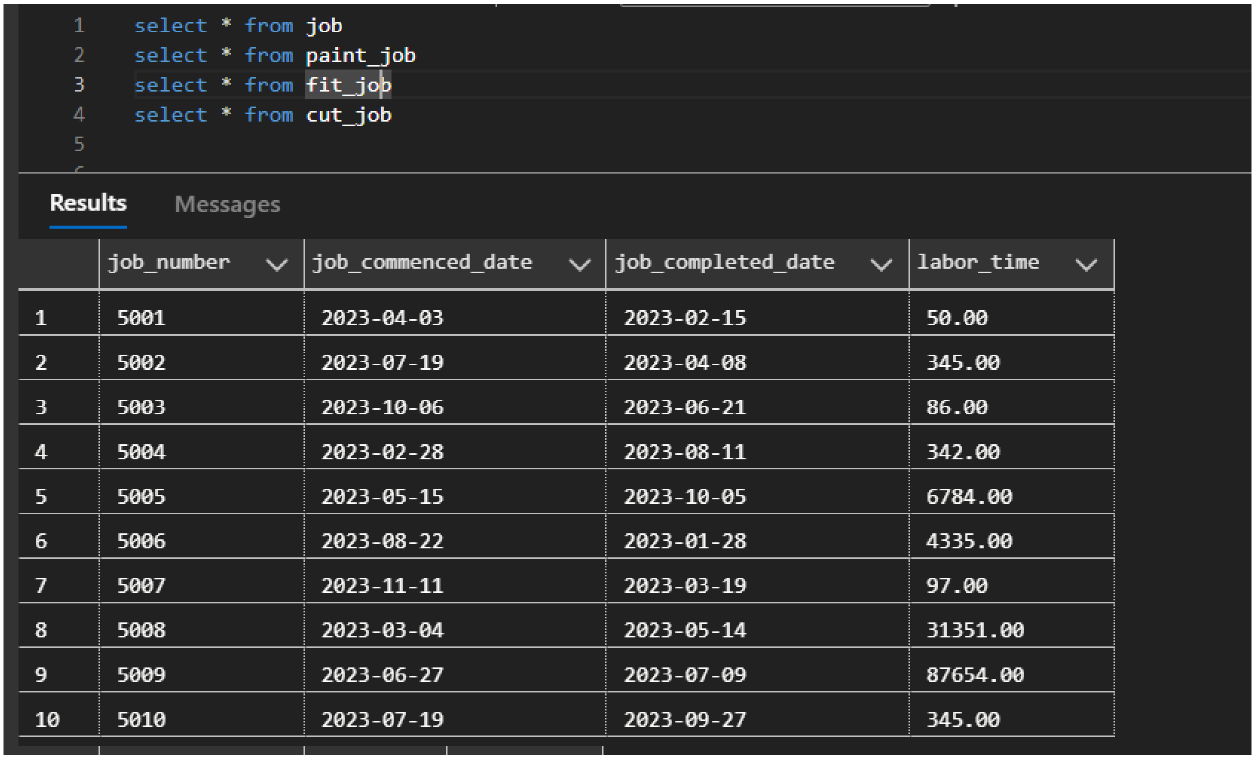
```
Enter your option :
7
Enter the job number
5003
Enter job competion date:
06-21-2023
Enter Labour Time:
86
Enter Process Type:
Connecting to the database...
```



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

Output Tables:

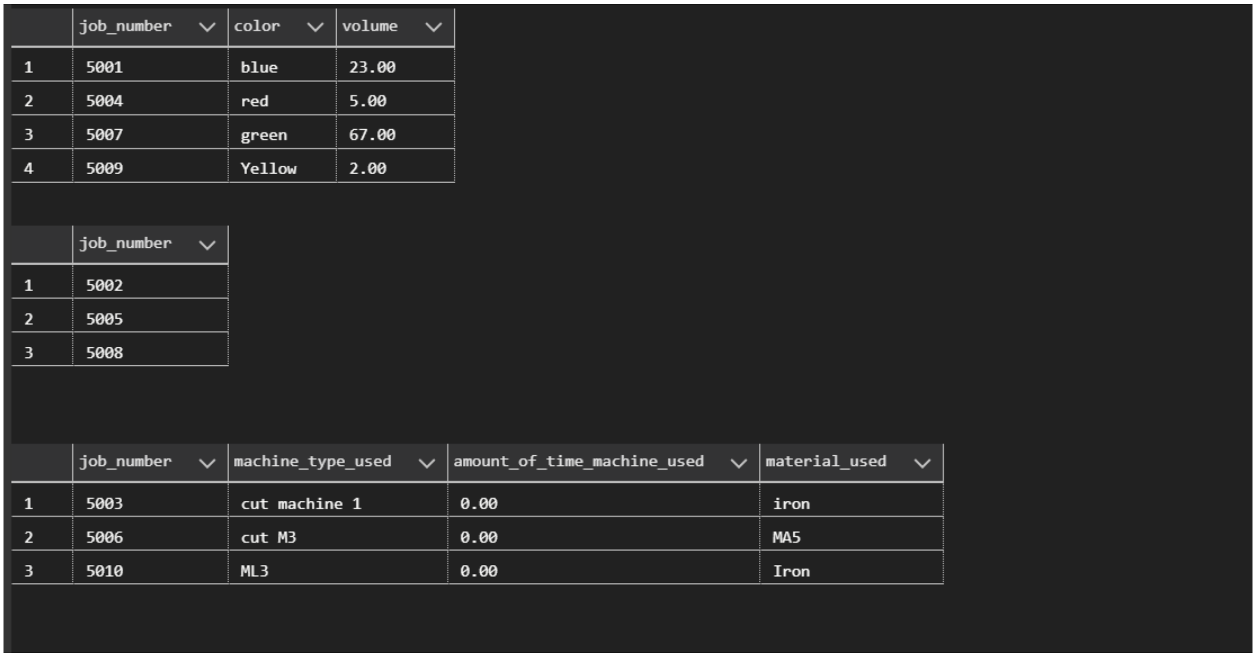
Job Table



```
select * from job
select * from paint_job
select * from fit_job
select * from cut_job
```

	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

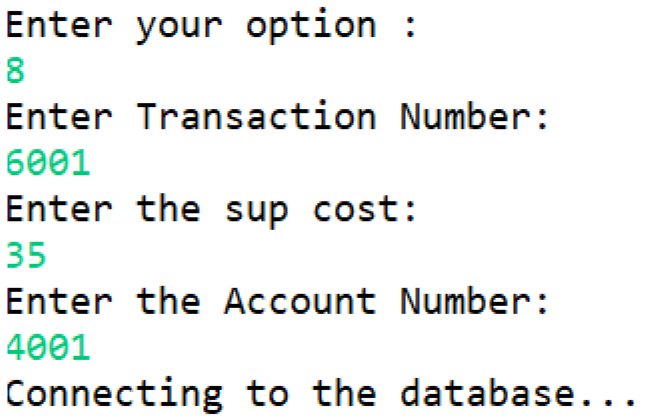


	job_number	color	volume
1	5001	blue	23.00
2	5004	red	5.00
3	5007	green	67.00
4	5009	Yellow	2.00

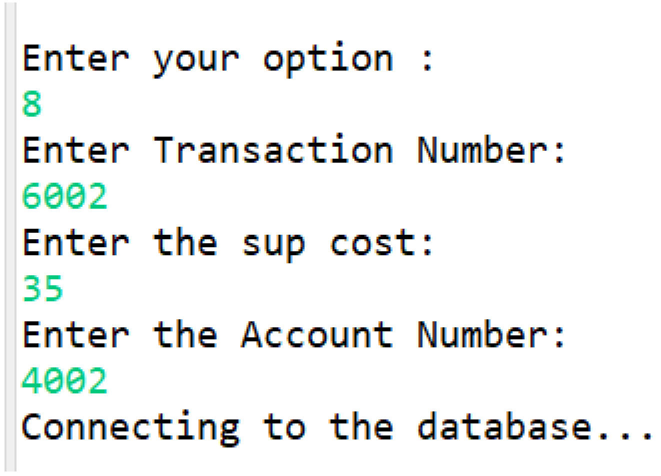
	job_number
1	5002
2	5005
3	5008

	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

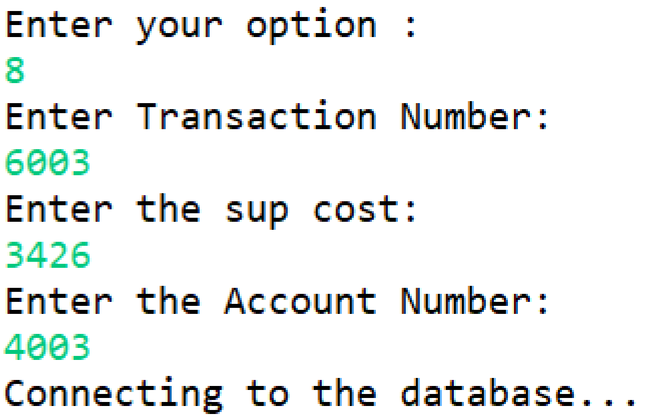
6.8: Query 8 Screenshots



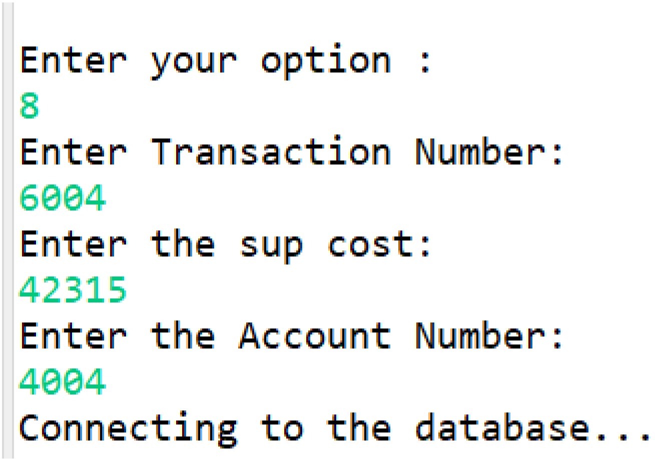
```
Enter your option :
8
Enter Transaction Number:
6001
Enter the sup cost:
35
Enter the Account Number:
4001
Connecting to the database...
```



```
Enter your option :
8
Enter Transaction Number:
6002
Enter the sup cost:
35
Enter the Account Number:
4002
Connecting to the database...
```

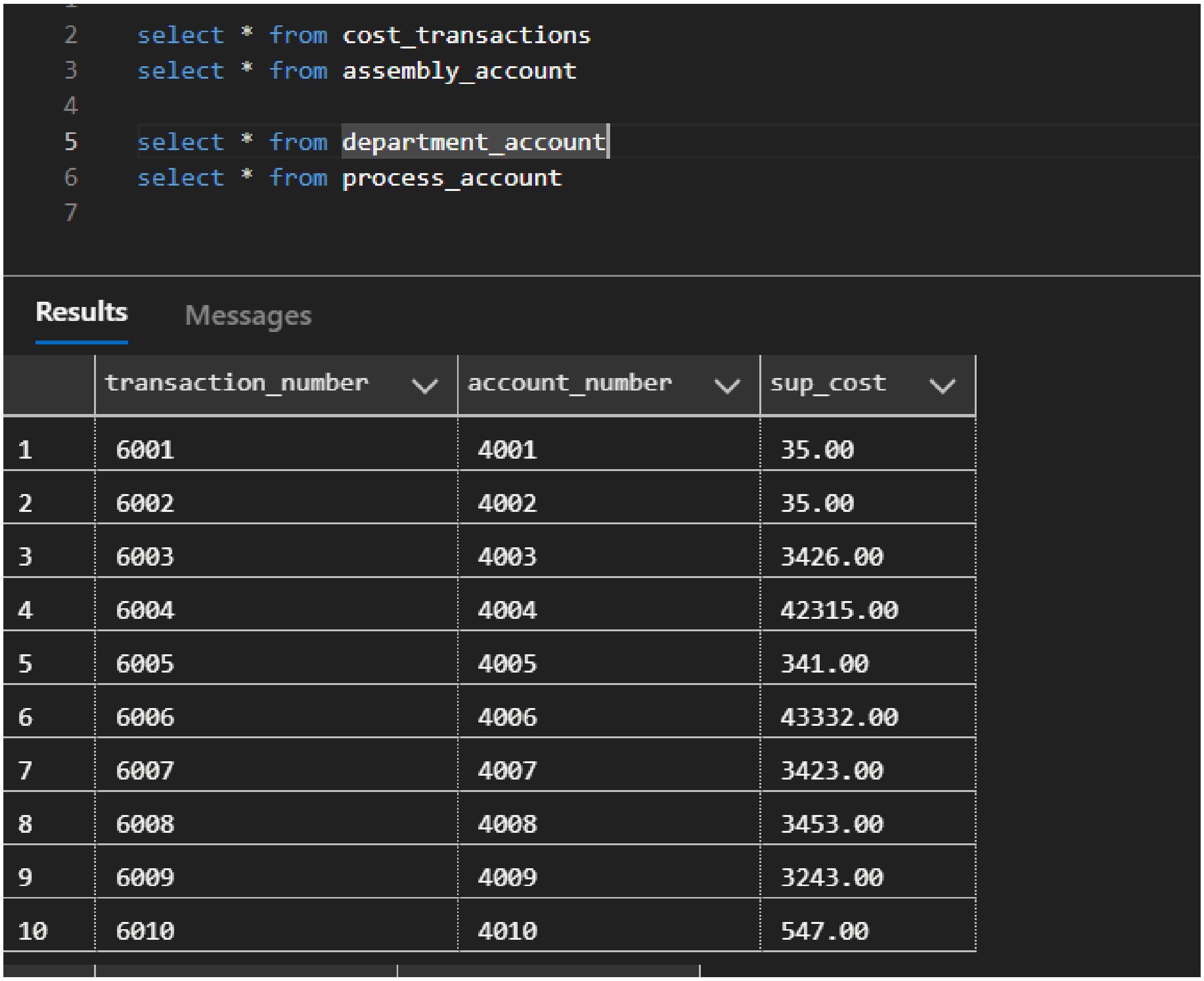


```
Enter your option :
8
Enter Transaction Number:
6003
Enter the sup cost:
3426
Enter the Account Number:
4003
Connecting to the database...
```



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

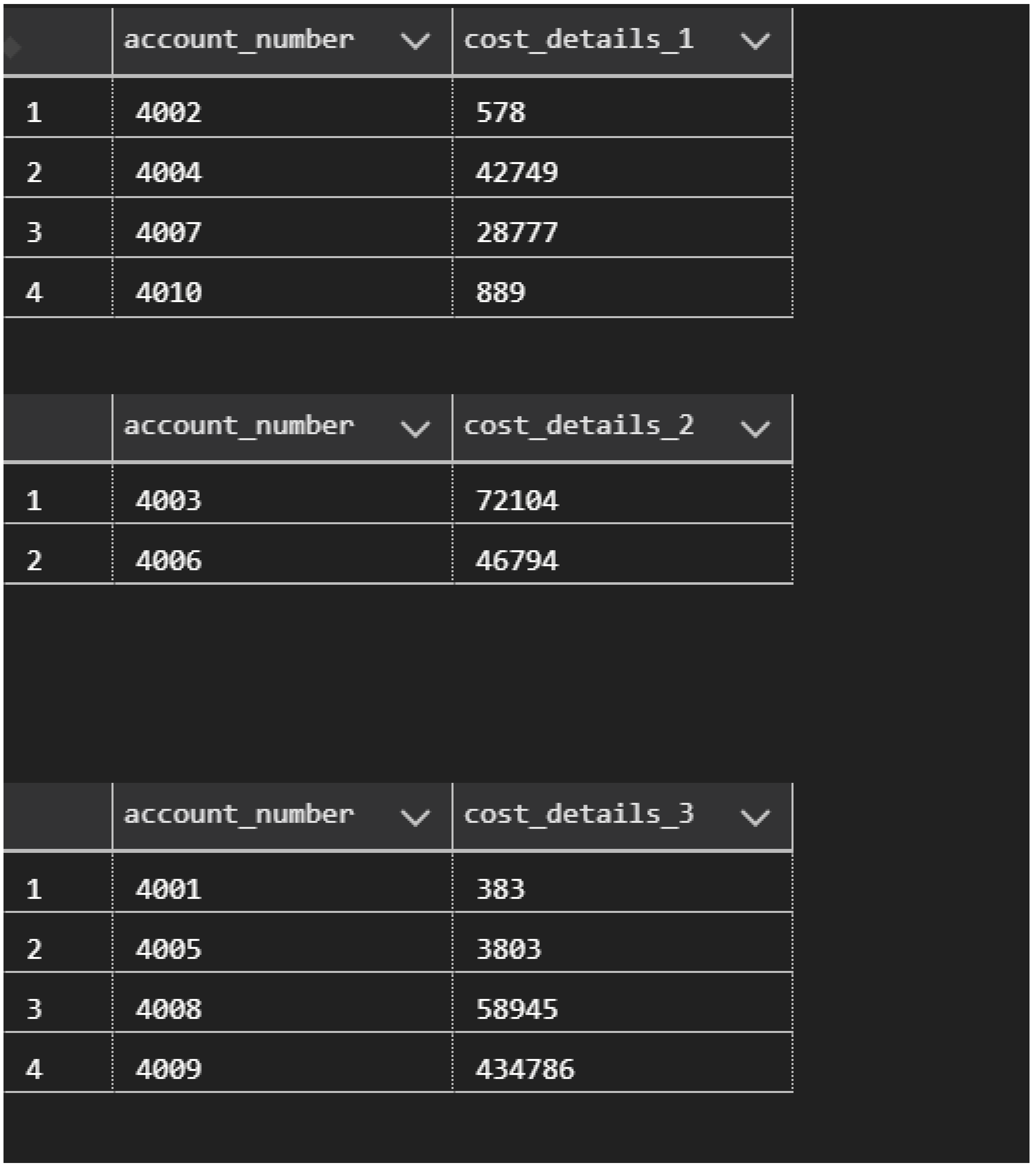
Output
Cost transactions



```
select * from cost_transactions
select * from assembly_account

select * from department_account
select * from process_account

```

	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

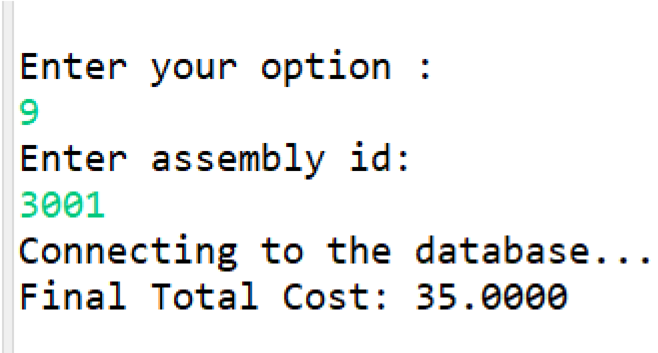


	account_number	cost_details_1
1	4002	578
2	4004	42749
3	4007	28777
4	4010	889

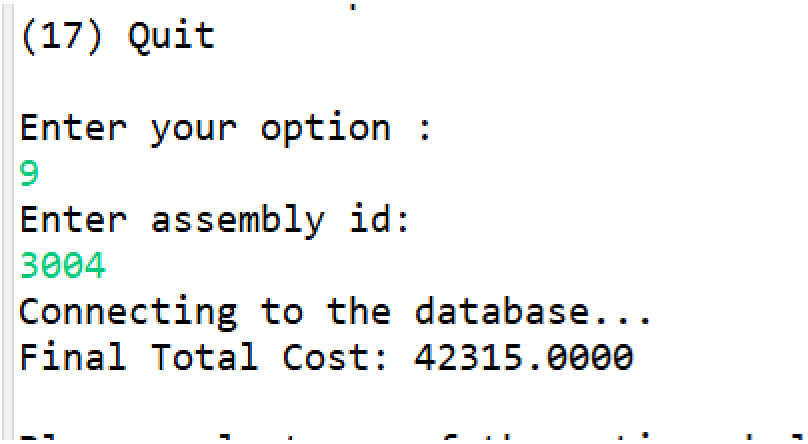
	account_number	cost_details_2
1	4003	72104
2	4006	46794

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

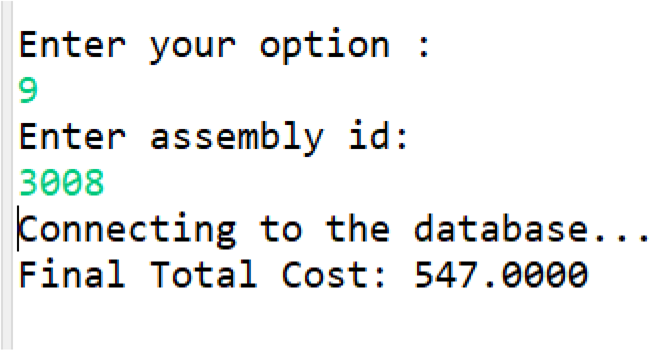
6.9: Query 9 Screenshots



```
Enter your option :
9
Enter assembly id:
3001
Connecting to the database...
Final Total Cost: 35.0000
```

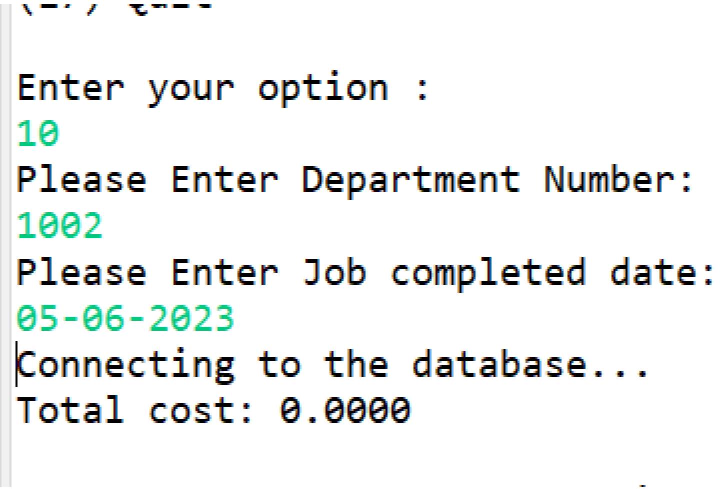


```
(17) Quit

Enter your option :
9
Enter assembly id:
3004
Connecting to the database...
Final Total Cost: 42315.0000
```

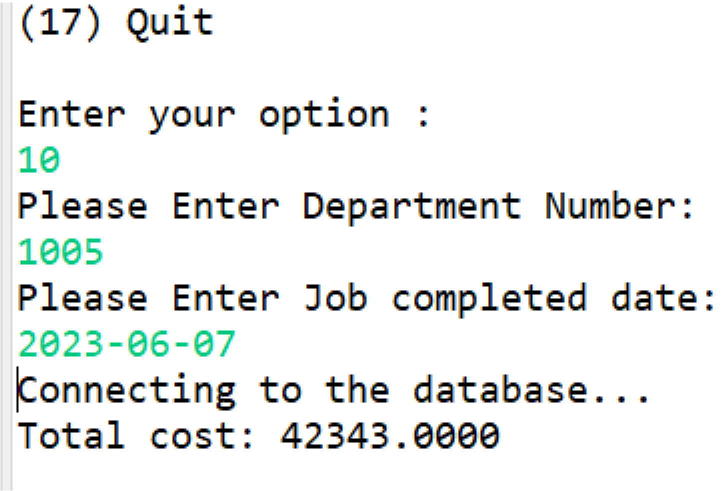


```
Enter your option :
9
Enter assembly id:
3008
Connecting to the database...
Final Total Cost: 547.0000
```

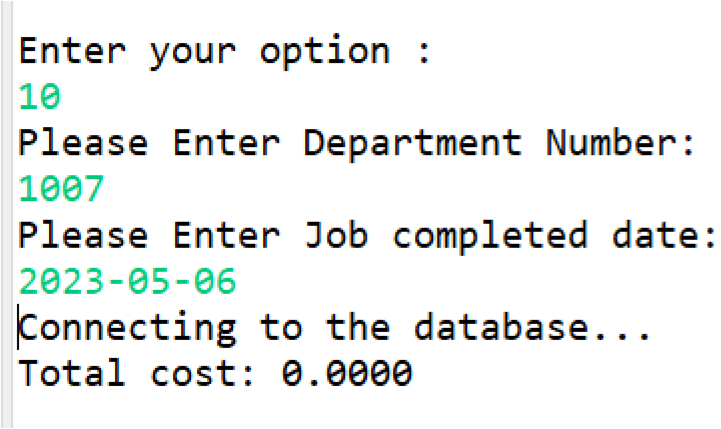
6.10: Query 10 Screenshots



```
(17) Quit

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
```

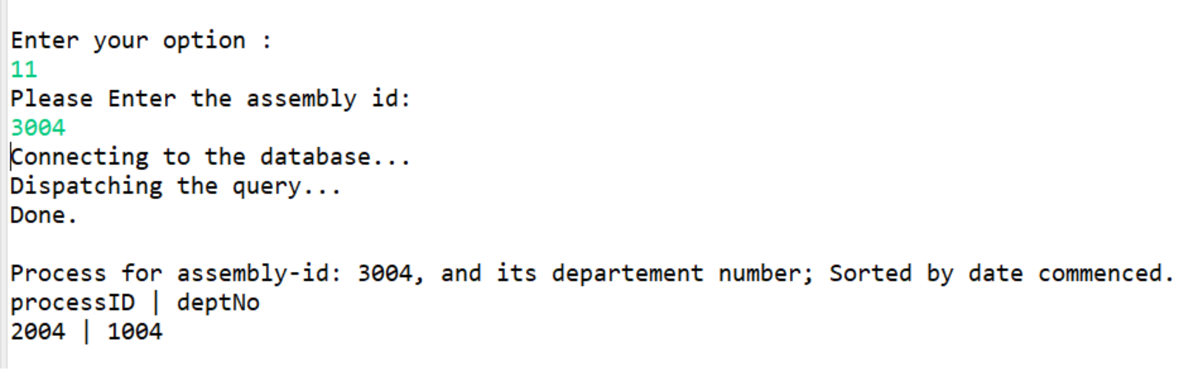


```
(17) Quit

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
```

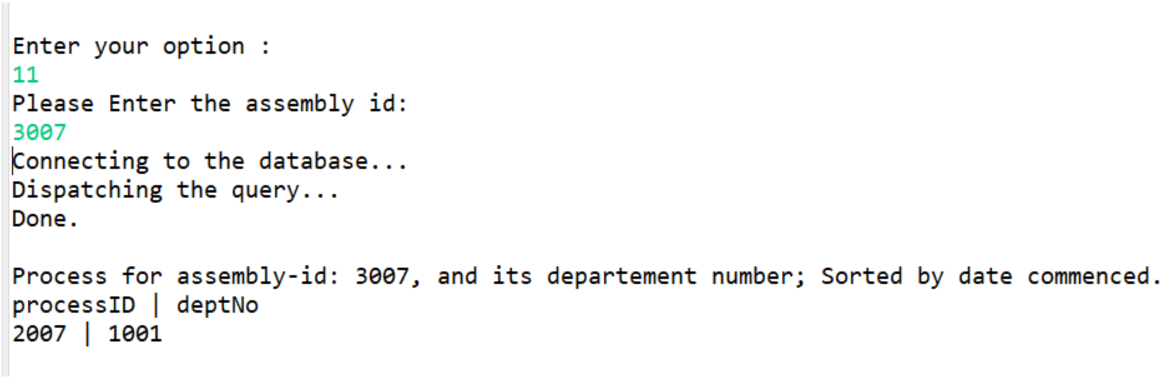


```
Enter your option :
10
Please Enter Department Number:
1007
Please Enter Job completed date:
2023-05-06
Connecting to the database...
Total cost: 0.0000
```

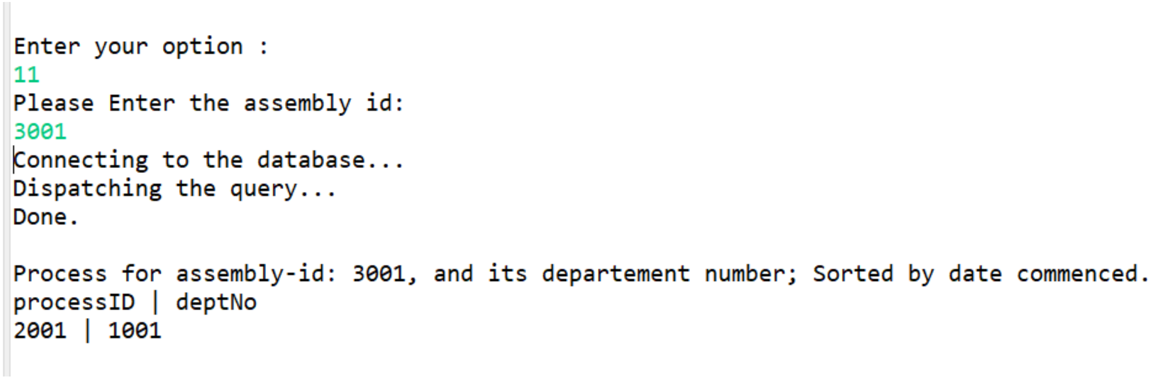
6.11: Query 11 Screenshots



```
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
```

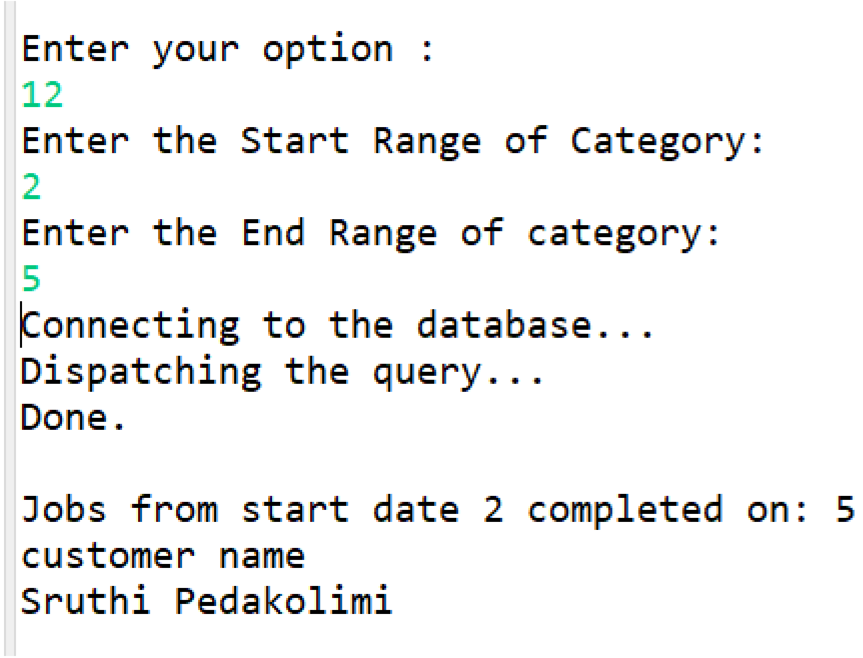


```
Enter your option :
11
Please Enter the assembly id:
3007
Connecting to the database...
Dispatching the query...
Done.

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

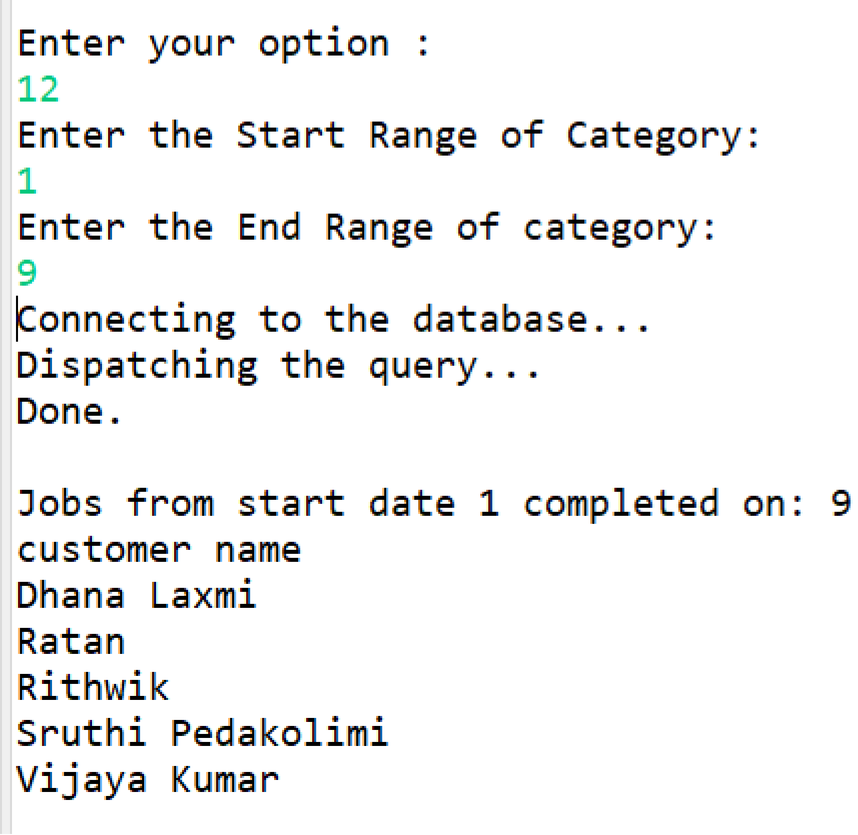


```
Enter your option :
11
Please Enter the assembly id:
3001
Connecting to the database...
Dispatching the query...
Done.

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

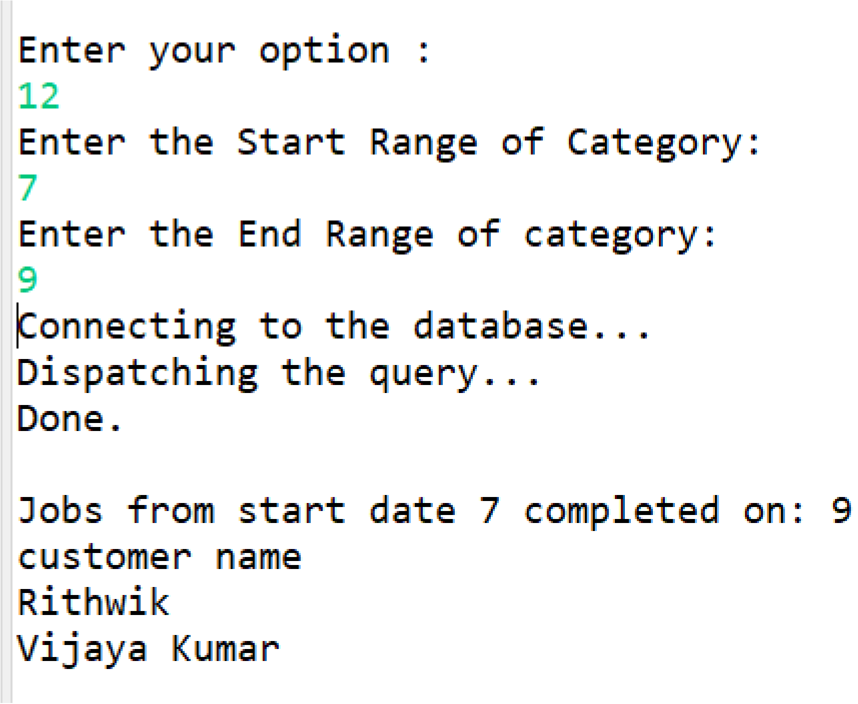
6.12: Query 12 Screenshots



```
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
```

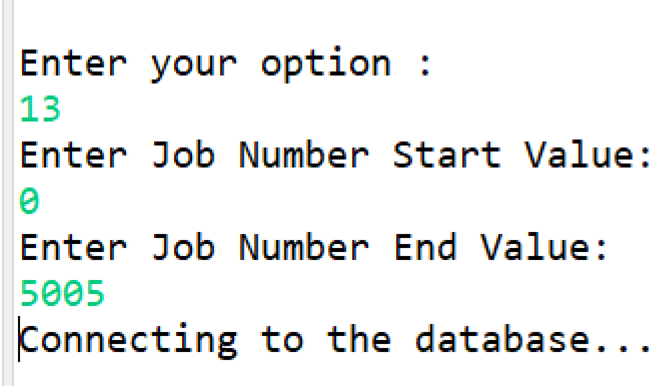


```
Enter your option :
12
Enter the Start Range of Category:
1
Enter the End Range of category:
9
Connecting to the database...
Dispatching the query...
Done.

Jobs from start date 1 completed on: 9
customer name
Dhana Laxmi
Ratan
Rithwik
Sruthi Pedakolimi
Vijaya Kumar
```

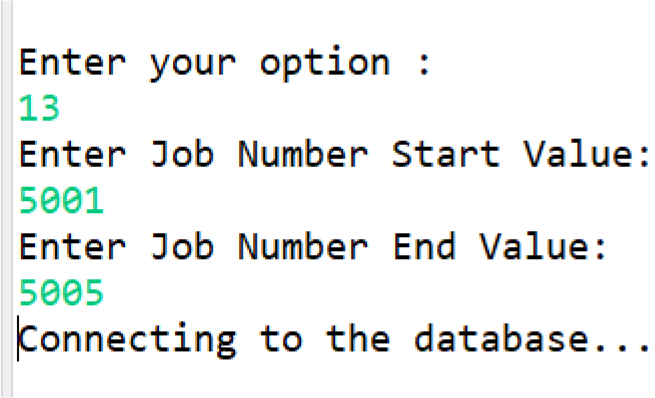


```
Enter your option :
12
Enter the Start Range of Category:
7
Enter the End Range of category:
9
Connecting to the database...
Dispatching the query...
Done.

Jobs from start date 7 completed on: 9
customer name
Rithwik
Vijaya Kumar
```

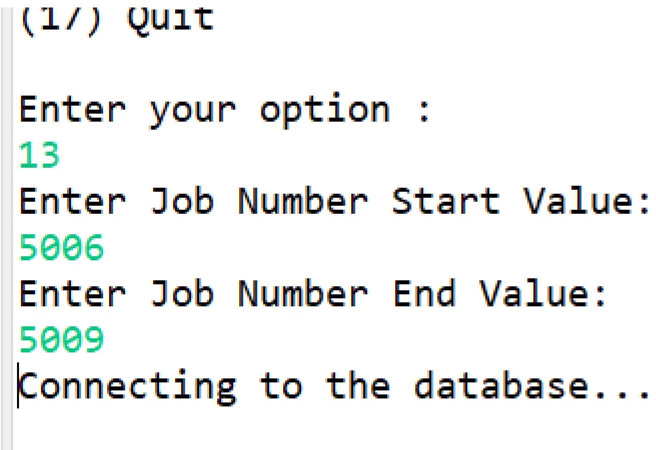
6.13: Query 13 Screenshots



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

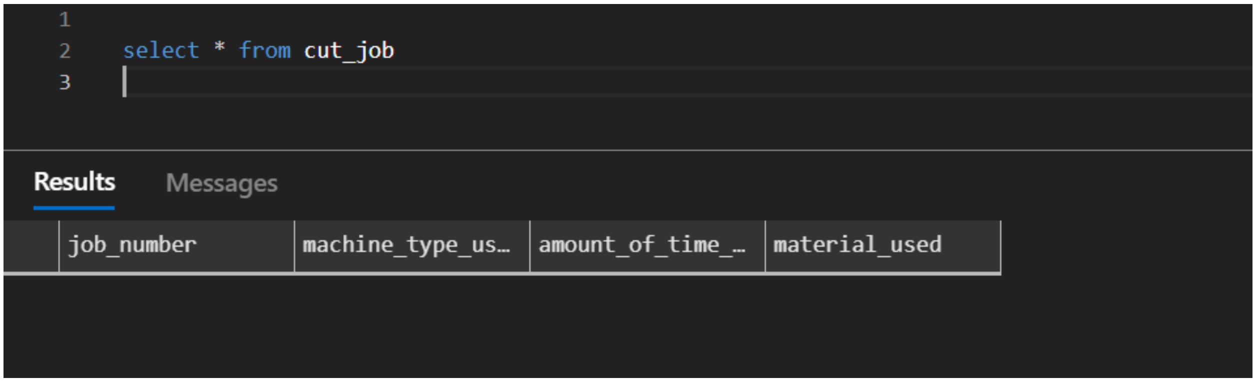


```
Enter your option :
13
Enter Job Number Start Value:
5001
Enter Job Number End Value:
5005
Connecting to the database...
```



```
(17) Quit

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

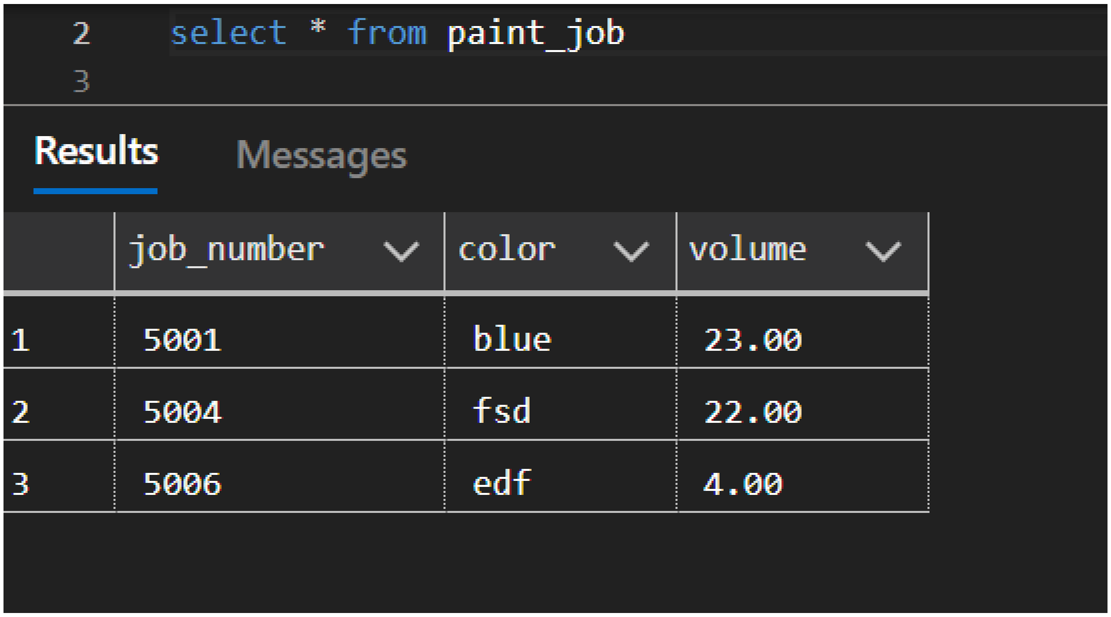
Here we have deleted all the cut jobs



```sql
select * from cut_job
```

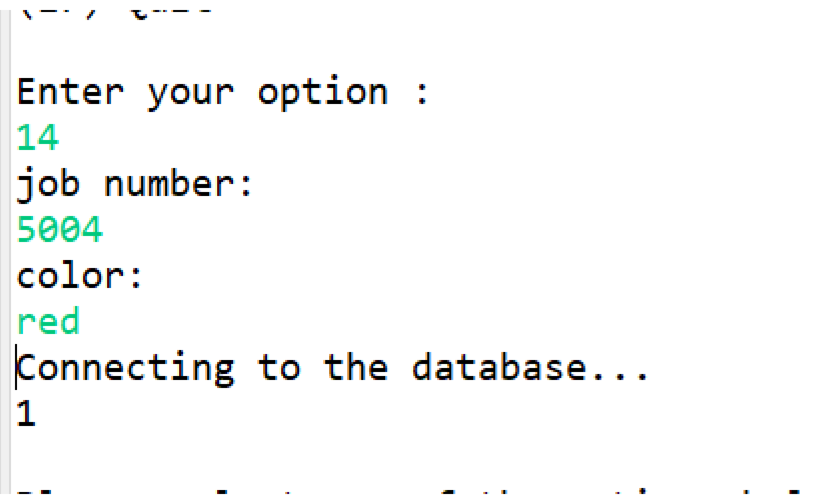
job_number	machine_type_us...	amount_of_time_...	material_used

6.14: Query 14 Screenshots
Before changing the color

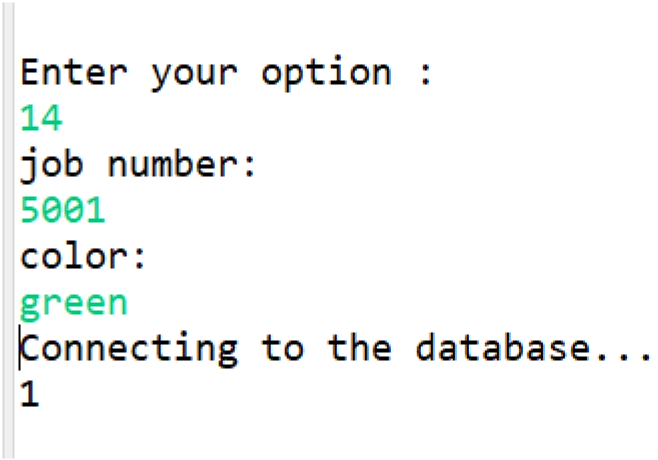


```sql
select * from paint_job
```

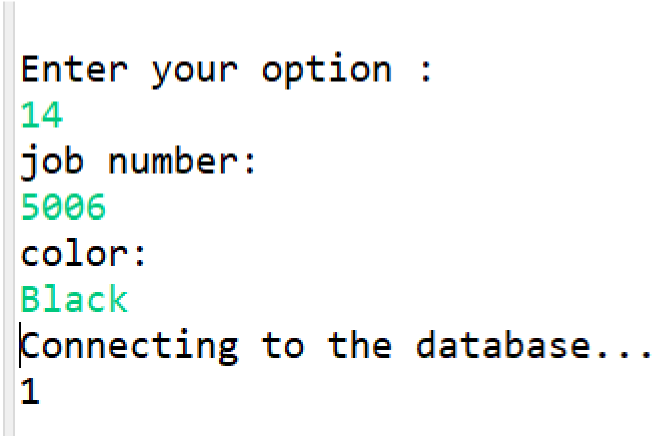
	job_number	color	volume
1	5001	blue	23.00
2	5004	fsd	22.00
3	5006	edf	4.00



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

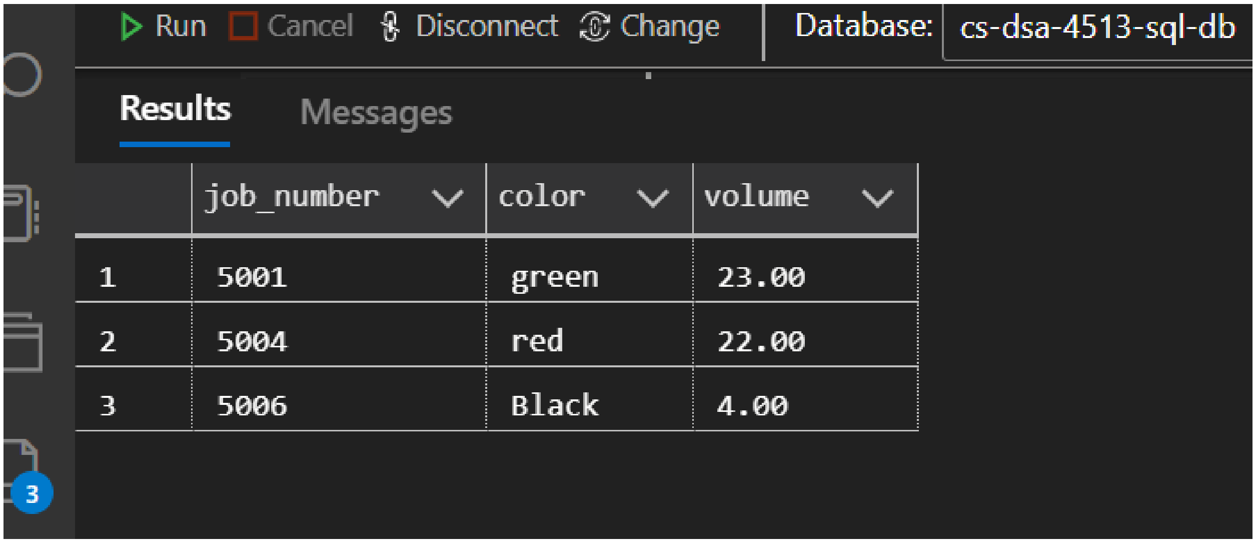


```
Enter your option :
14
job number:
5001
color:
green
Connecting to the database...
1
```



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

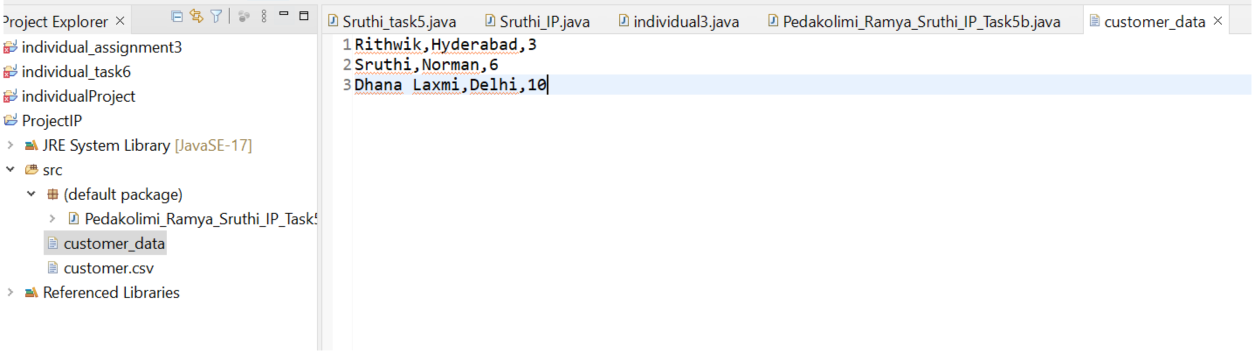
After Execution



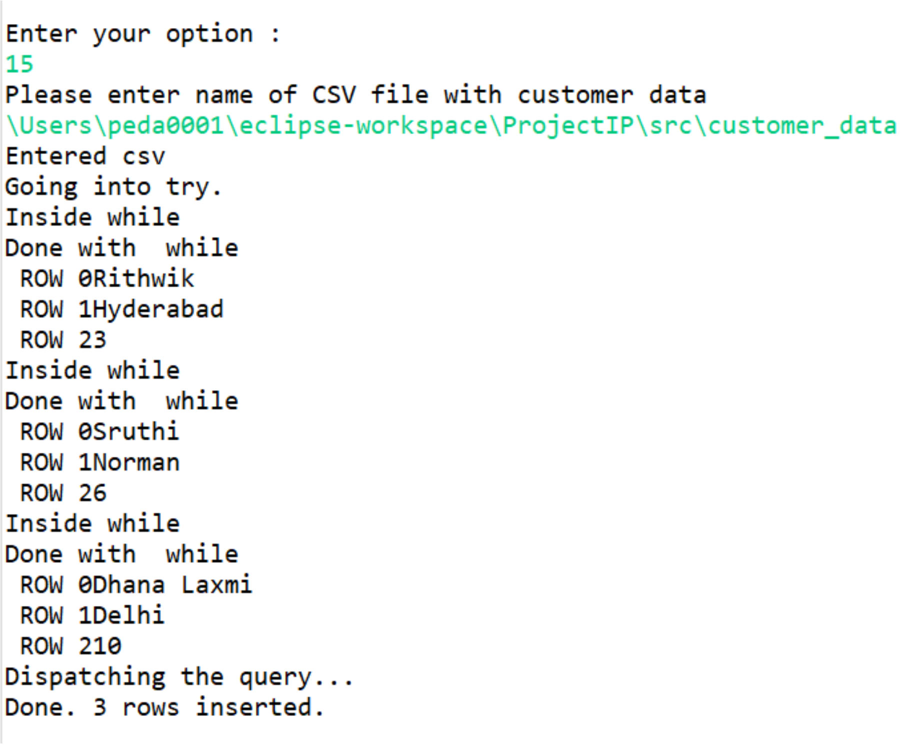
	job_number	color	volume
1	5001	green	23.00
2	5004	red	22.00
3	5006	Black	4.00

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

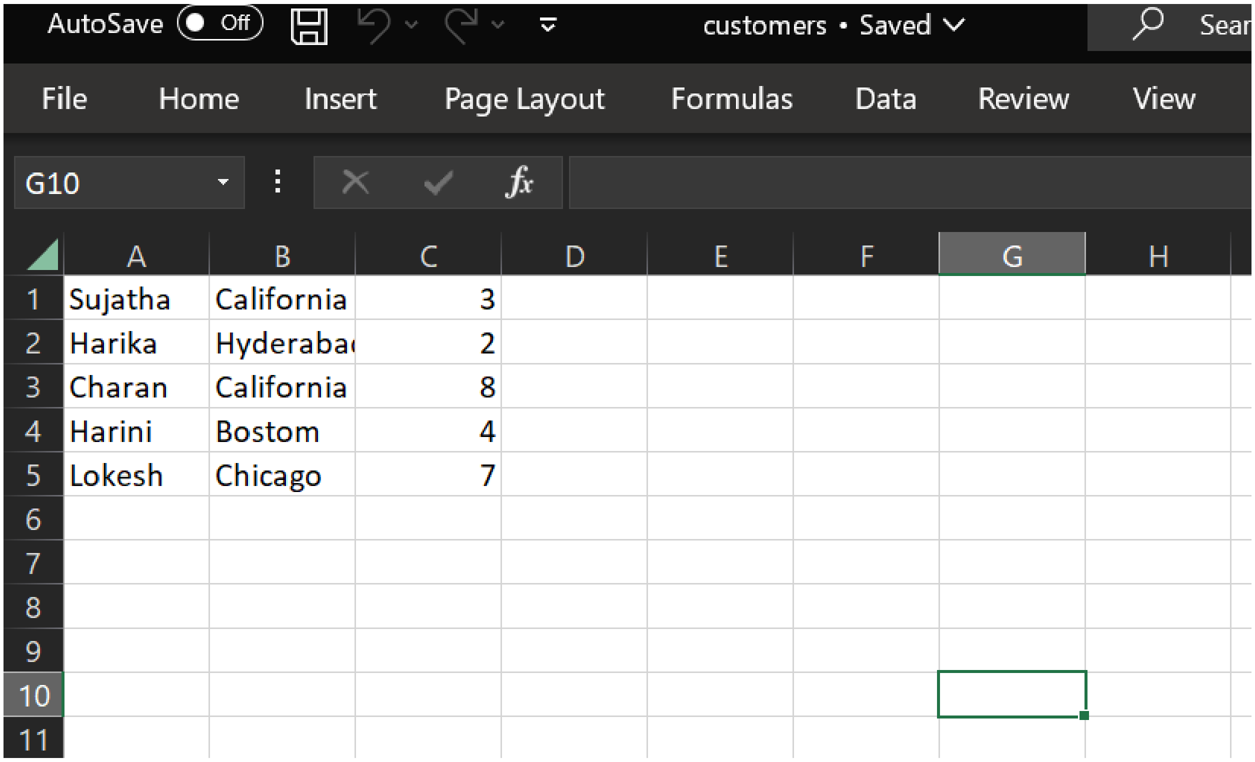
Importing



```
Rithwik,Hyderabad,3
Sruthi,Norman,6
Dhana Laxmi,Delhi,10
```

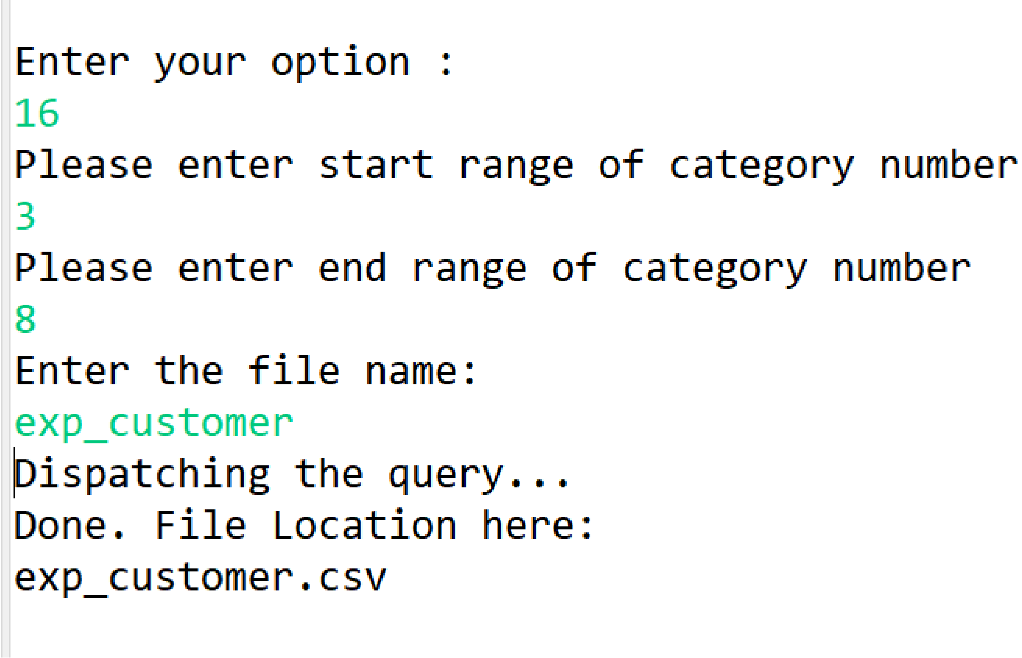


```
Enter your option :
15
Please enter name of CSV file with customer data
\Users\peda0001\eclipse-workspace\ProjectIP\src\customer_data
Entered csv
Going into try.
Inside while
Done with  while
 ROW 0Rithwik
 ROW 1Hyderabad
 ROW 23
Inside while
Done with  while
 ROW 0Sruthi
 ROW 1Norman
 ROW 26
Inside while
Done with  while
 ROW 0Dhana Laxmi
 ROW 1Delhi
 ROW 210
Dispatching the query...
Done. 3 rows inserted.
```



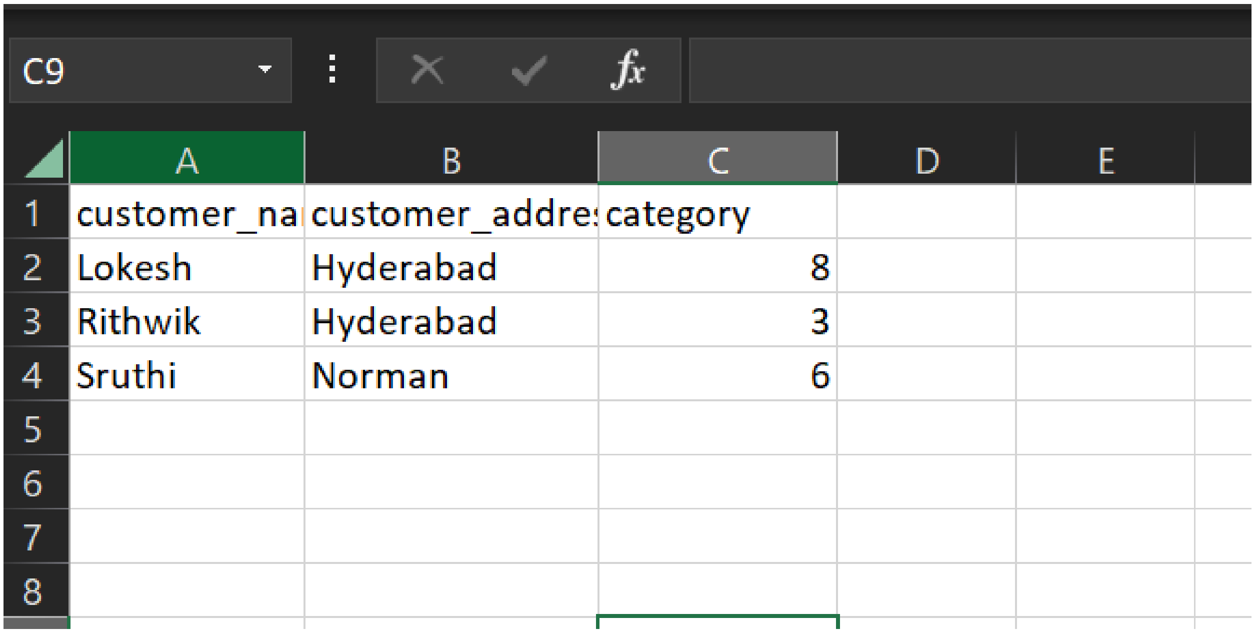
	A	B	C	D	E	F	G	H
1	Sujatha	California	3					
2	Harika	Hyderaba	2					
3	Charan	California	8					
4	Harini	Bostom	4					
5	Lokesh	Chicago	7					
6								
7								
8								
9								
10								
11								

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

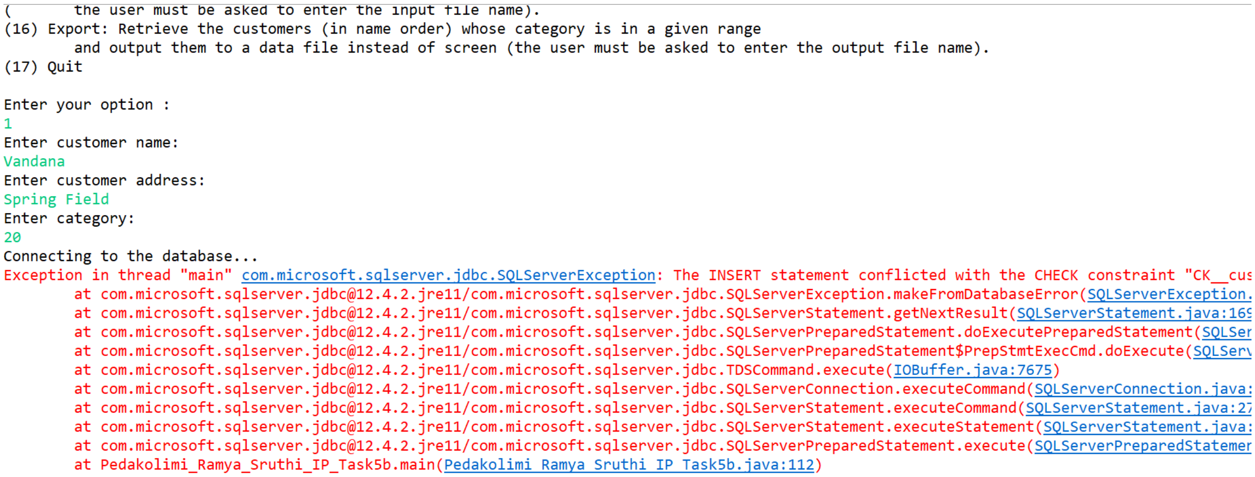


	A	B	C	D	E
1	customer_na	customer_addre	category		
2	Lokesh	Hyderabad	8		
3	Rithwik	Hyderabad	3		
4	Sruthi	Norman	6		
5					
6					
7					
8					

6.16: Screenshots showing the testing of three types of errors

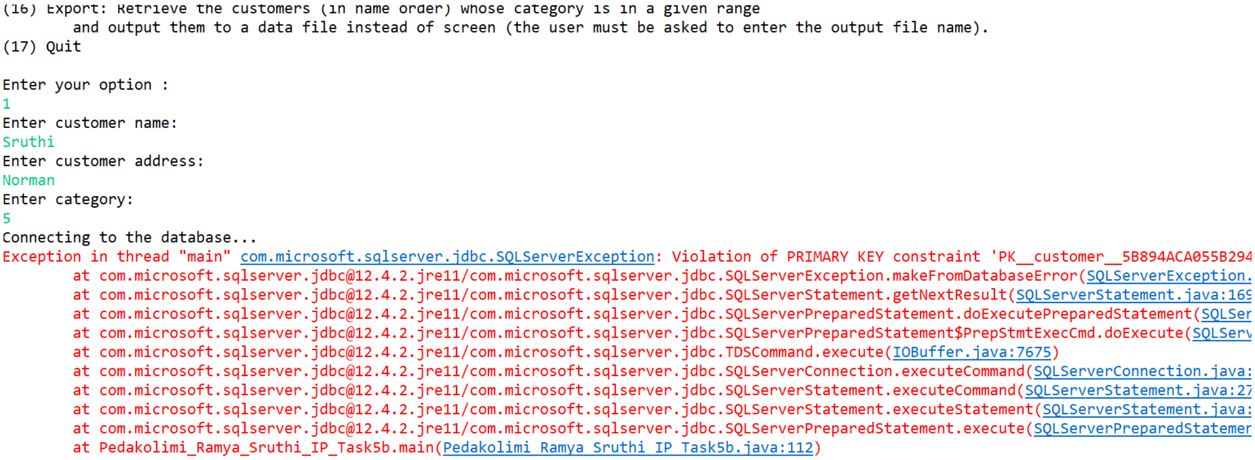
Error Handling

Category Constraint



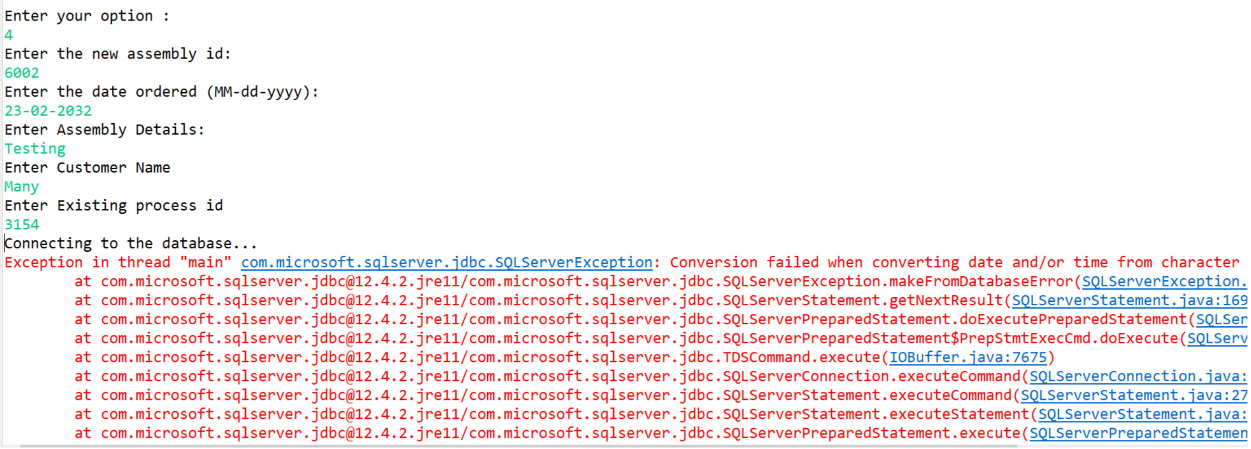
```
(       the user must be asked to enter the input file name).
(16) Export: Retrieve the customers (in name order) whose category is in a given range
        and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit

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:16
        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(SQLServ
        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:
        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(SQLServerPreparedStatemer
        at Pedakolimi_Ramya_Sruthi_IP_Task5b.main(Pedakolimi_Ramya_Sruthi_IP_Task5b.java:112)
```

Primary Key Constraint



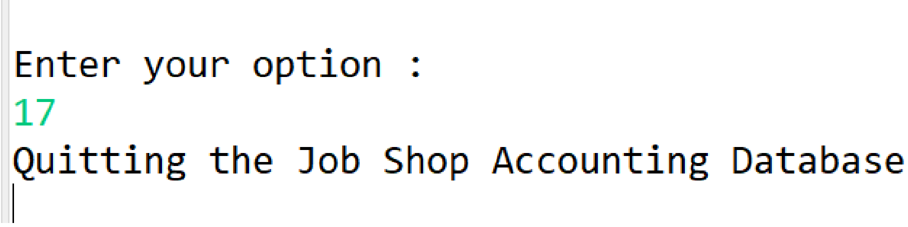
```
(16) Export: Retrieve the customers (in name order) whose category is in a given range
        and output them to a data file instead of screen (the user must be asked to enter the output file name).
(17) Quit

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.
        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(SQLSer
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServ
        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:
        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(SQLServerPreparedStatemen
        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.
        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(SQLSer
        at com.microsoft.sqlserver.jdbc@12.4.2.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServ
        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:
        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(SQLServerPreparedStatemen
```

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;en
crypt=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 cellspacing="2" cellpadding="2" border="1">
<tr> <!-- The table headers row -->
<td align="center">
<h4>Customer Name</h4>
</td>
<td align="center">
<h4>Customer Address</h4>
</td>
<td align="center">
<h4>Category</h4>
</td>
</tr>
<%
while(customers.next()) { // For each 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 customet 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

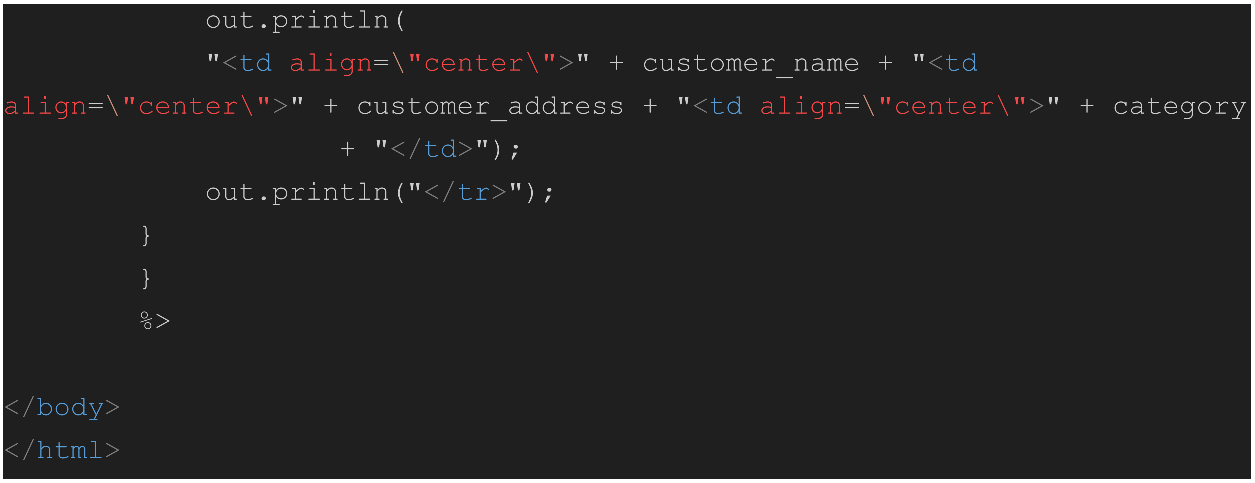
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Retrieval of  Customers</title>
```

```
</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 cellspacing="2" cellpadding="2" border="1">
        <tr>
            <!-- The table headers row -->
            <td align="center">
                <h4>Customer Name</h4>
            </td>
            <td align="center">
                <h4>Customer Address</h4>
            </td>
            <td align="center">
                <h4>Category</h4>
            </td>
        </tr>
        <%
        while (customers.next()) { // For each 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>
```

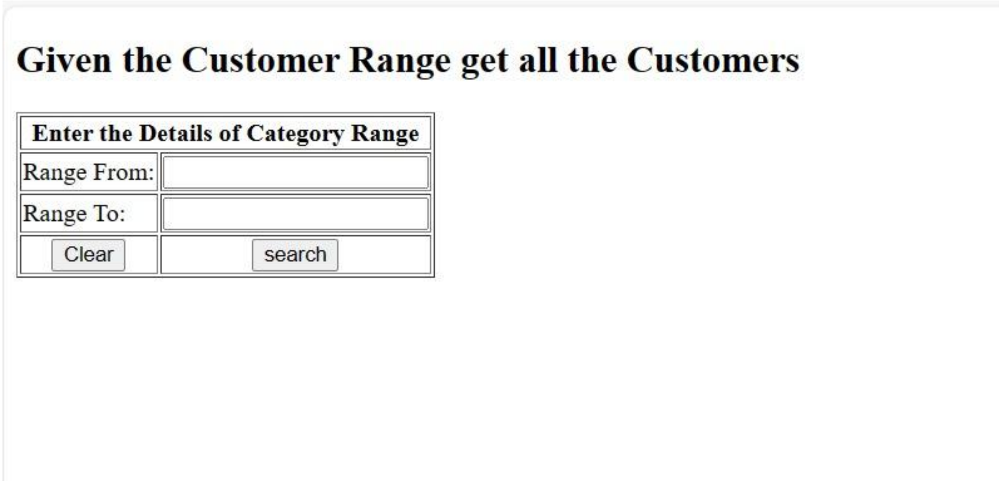
Pedakolimi_Ramya_Sruthi_IP_Task7_retrieve_customers

```
<%@ 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 cellspacing="2" cellpadding="2" border="1">
        <tr>
            <!-- The table headers row -->
            <td align="center">
                <h4>Customer Name</h4>
            </td>
            <td align="center">
                <h4>Customer Address</h4>
            </td>
            <td align="center">
                <h4>Category</h4>
            </td>
        </tr>
        <%
        while (customers.next()) {
            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
```



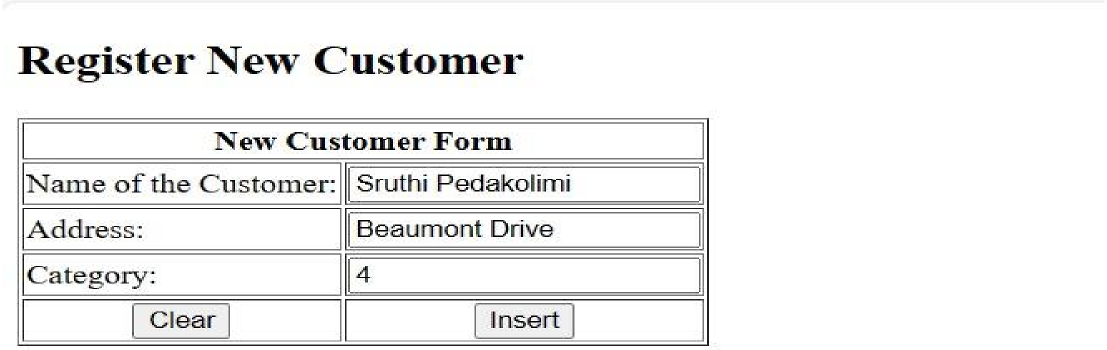
```
            out.println(
            "<td align=\"center\">" + customer_name + "<td 
align=\"center\">" + customer_address + "<td align=\"center\">" + category 
                    + "</td>");
            out.println("</tr>");
        }
        }
        %>

</body>
</html>
```

Output:



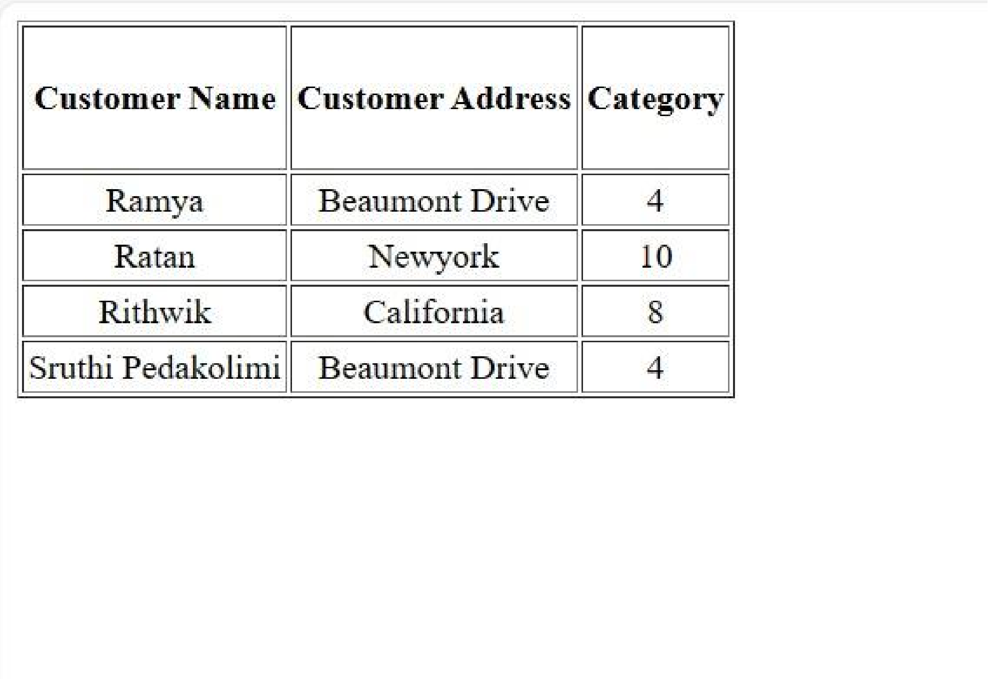
Given the Customer Range get all the Customers

Enter the Details of Category Range	
Range From:	
Range To:	
Clear	search

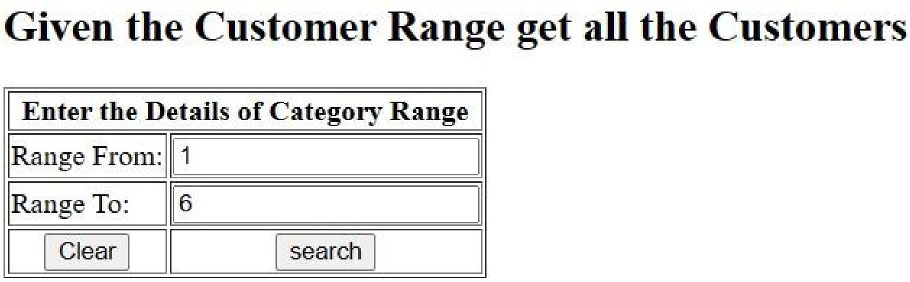


Register New Customer

New Customer Form	
Name of the Customer:	Sruthi Pedakolimi
Address:	Beaumont Drive
Category:	4
Clear	Insert

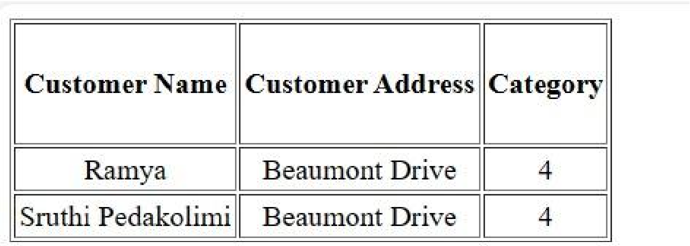


Customer Name	Customer Address	Category
Ramya	Beaumont Drive	4
Ratan	Newyork	10
Rithwik	California	8
Sruthi Pedakolimi	Beaumont Drive	4



Given the Customer Range get all the Customers

Enter the Details of Category Range	
Range From:	1
Range To:	6



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