

**Name: Kovida Mothukuri**

**ID: 113534385**

**Email address: Kovida.mothukuri-1@ou.edu**

**Assignment: Individual Project**

**Course: CS/DSA 4513, Section 001**

**Semester and Year: Fall 2022**

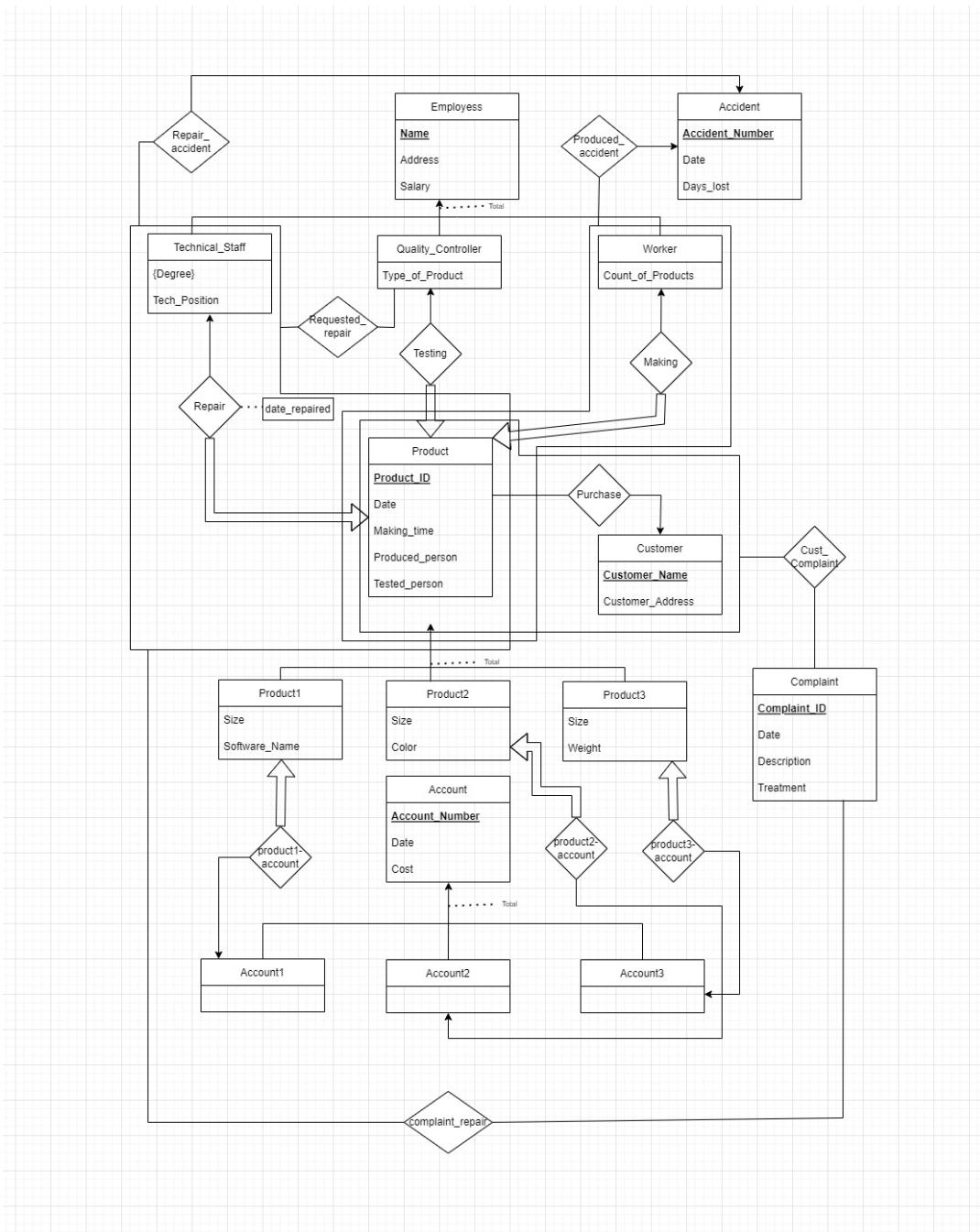
**Instructor: Dr. Le Gruenwald**

**SCORE:**

Tasks Performed	Page Number
<b>Task1.</b>	<b>3 - 4</b>
<b>1.1.ER Diagram</b>	<b>3 - 3</b>
<b>1.2. Relational Database schema:</b>	<b>4 - 4</b>
<b>Task2. Schema Diagram</b>	<b>5 - 5</b>
<b>Task3.</b>	<b>6 - 10</b>
<b>3.1. Tables Storage Structures:</b>	<b>6 - 9</b>
<b>3.2. Table Storage Structures Discussion (In Azure SQL Database):</b>	<b>10 - 10</b>
<b>Task4. Creation of all tables in azure SQL Database:</b>	<b>11 – 20</b>
<b>Task5.</b>	<b>21 – 38</b>
<b>5.1. Transact SQL stored procedures implementing all queries</b>	<b>21 – 28</b>
<b>5.2. The Java source program, screenshots showing its successful compilation</b>	<b>28 – 38</b>
<b>Task6. Java Program Execution</b>	<b>39 - 76</b>
<b>6.1. query1 testing screenshots</b>	<b>39 – 46</b>
<b>6.2. query2 testing screenshots</b>	<b>46 – 53</b>
<b>6.3. query3 testing screenshots</b>	<b>54 – 57</b>
<b>6.4. query4 testing screenshots</b>	<b>58 – 62</b>
<b>6.5. query5 testing screenshots</b>	<b>63 – 65</b>
<b>6.6. query6 testing screenshots</b>	<b>66 – 68</b>
<b>6.7. query7 testing screenshots</b>	<b>68 – 69</b>
<b>6.8. query8 testing screenshots</b>	<b>69 – 69</b>
<b>6.9. query9 testing screenshots</b>	<b>70 – 70</b>
<b>6.10. query10 testing screenshots</b>	<b>70 – 71</b>
<b>6.11. query11 testing screenshots</b>	<b>71 – 72</b>
<b>6.12. query12 testing screenshots</b>	<b>72 – 72</b>
<b>6.13. query13 testing screenshots</b>	<b>72 – 72</b>
<b>6.14. query14 testing screenshots</b>	<b>73 – 73</b>
<b>6.15. query15 testing screenshots</b>	<b>73 - 73</b>
<b>6.16. Screenshots showing the testing of the Import and Export options</b>	<b>74 – 75</b>
<b>6.17. Error Handling</b>	<b>75 – 76</b>
<b>6.18. query18 testing screenshots</b>	<b>76 - 76</b>
<b>Task 7. Web based application</b>	<b>77 - 85</b>
<b>7.1. Source code for a web application with pictures demonstrating its successful compilation</b>	<b>77 – 81</b>
<b>7.2. Screenshots demonstrating the Web database application being</b>	<b>81 - 85</b>

## Task1:

### 1.1– ER Diagram:



## **1.2– Relational Database schema:**

Tech\_Staff (tech\_name, tech\_address, tech\_salary, tech\_position)

Degree (tech\_name, degree)

Quality\_controller (qc\_name, qc\_address, qc\_salary, type\_of\_product)

Worker (w\_name, w\_address, w\_salary, count\_of\_products)

Product (product\_ID, produced\_date, making\_time, produced\_person, tested\_person)

Product1(product\_ID, size, software\_name)

Product2 (product\_ID, size, color)

Product3 (product\_ID, size, p\_weight)

Account (account\_number, created\_date, cost)

Product1\_account (product\_ID, account\_number)

Product2\_account (product\_ID, account\_number)

Product3\_account (product\_ID, account\_number)

Customer (customer\_name, customer\_address)

Purchase (product\_ID, customer\_name)

Complaint (complaint\_ID, complaint\_date, comp\_desc, treatment)

Accident (accident\_number, accident\_date, days\_lost)

Repair (product\_ID, tech\_name, date\_repaired)

Requested\_repair (product\_ID, qc\_name)

Cust\_complaint (Complaint\_ID, Product\_ID)

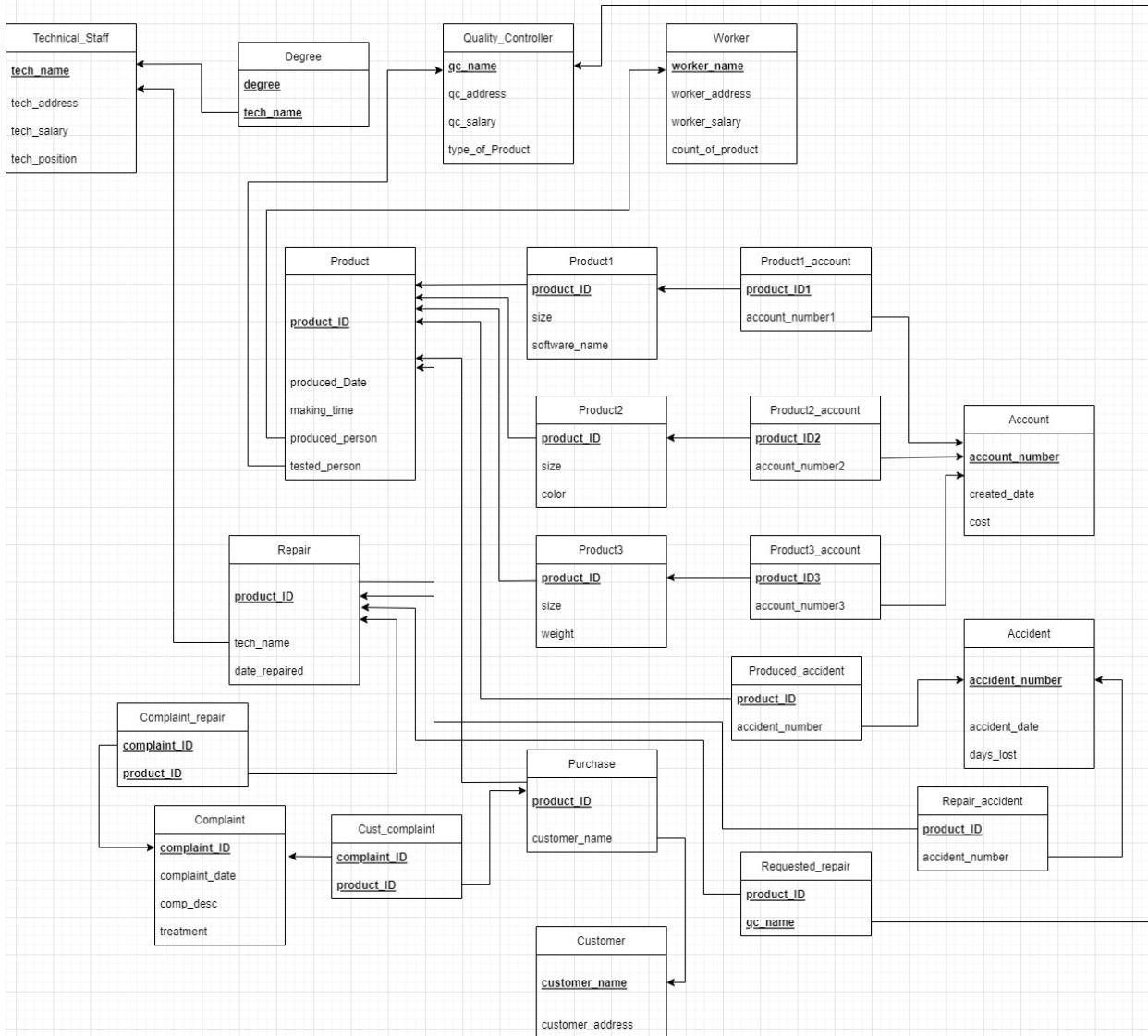
Complaint\_repair (complaint\_ID, product\_ID)

Repair\_accident (product\_ID, accident\_number)

Produced\_accident (product\_ID, accident\_number)

## Task2:

### Schema Diagram:



### Task3:

#### 3.1 – Tables Storage Structures:

Table Name	Query and Type	Search Key	Query Frequency	Selected File Organization	Justifications
Tech_Staff	1)Insert 12) Range Search	Salary	2/month 1/month	Indexed sequential file On salary	As it has range search, it is better to use indexed sequential on search key salary
Degree	1)Insert		2/month	Heap file	For Insertions, it is better to use heap file
Quality_Controller	1)Insert 12)Range Search	Salary	2/month 1/month	Indexed sequential file on salary	As it has range search, it is better to use indexed sequential on search key salary
Worker	1)Insert 12)Range Search	Salary	2/month 1/month	Indexed sequential file on salary	As it has range search, it is better to use indexed sequential on search key salary
Product	2)Insert 7)Random Search 8)Random Search 9)Random Search 14)random search	Product_ID Produced_person Tested_person Produced_date	400/day 100/day 2000/day 400/day 5/day	Dynamic Hashing on produced_person	As it has Random search and is used more often, it is better to use dynamic hashing on search key produced_person
Product1	2)Insert		400/day	Heap file	For Insertions, it is better to use heap file
Product2	2)Insert 11)Random Search	color	400/day 5/month	Dynamic hashing on color	As it has Random search and is used more often, it

					is better to use dynamic hashing on search key color
Product3	2)Insert		400/day	Heap file	For Insertions, it is better to use heap file
Account	4)Insert 10)random Search 14)random search	Account_number Account_number	40/day 40/day 5/day	Dynamic hashing on account_number	As it has Random search and is used more often, it is better to use dynamic hashing on search key account_number
Product1_account	4)Insert 14)random search	Product_ID	40/day 5/day	Dynamic hashing on Product_ID	As it has Random search and is used more often, it is better to use dynamic hashing on search key Product ID
Product2_account	4)Insert 14)random search	Product_ID	40/day 5/day	Dynamic hashing on Product_ID	As it has Random search and is used more often, it is better to use dynamic hashing on search key Product ID
Product3_account	4)Insert 10)random search 14)random search	Product_ID Product_ID	40/day 40/day 5/day	Dynamic hashing on Product_ID	As it has Random search and is used more often, it is better to use dynamic hashing on search key Product ID

Customer	3)Insert		50/day	Heap file	For Insertions, it is better to use heap file
Purchase	3)Insert 11)random search	Product_ID	50/day 5/month	Dynamic hashing on Product_ID	As it has Random search and is used more often, it is better to use dynamic hashing on search key Product ID
Complaint	5)Insert		30/day	Heap file	For Insertions, it is better to use heap file
Accident	6)Insert 13)random search 15)range search	Accident_number Accident_date	1/week 1/month 1/day	Indexed sequential file on Accident_date	As it has range search, it is better to use indexed sequential on search key salary
Repair	2)Insert 5)Insert		400/day 30/day	Heap file	For Insertions, it is better to use heap file
Requested_repair	2)insert 10)random search	qc_name	400/day 40/day	Dynamic hashing on qc_name	As it has Random search and is used more often, it is better to use dynamic hashing on search key qc_name
Cust_complaint	5)Insert		30/day	Heap file	For Insertions, it is better to use heap file
Complaint_repair	5)Insert 9)random search 13)random search	Product_ID Product_ID	30/day 400/day 1/month	Dynamic hashing on Product_ID	As it has Random search and is used more often, it is better to use dynamic hashing on

					search key Product ID
Repair_accident	6)Insert 13)random search 15)random search	Product_ID Accident_number	1/week 1/month 1/day	Dynamic hashing on accident_number	As it has Random search and is used more often, it is better to use dynamic hashing on search key accident_number
Produced_accident	6)Insert 15)random search	Accident_number	1/week 1/day	Dynamic hashing on accident_number	As it has Random search and is used more often, it is better to use dynamic hashing on search key accident_number

### 3.2 – Table Storage Structures Discussion (In Azure SQL Database):

Table	Storage Structure
Tech_Staff	Indexing on salary
Degree	No indexing
Quality_Controller	Indexing on salary
Worker	Indexing on salary
Product	Indexing on produced_person
Product1	No indexing
Product2	Indexing on color
Product3	No indexing
Account	No indexing as search key is primary key
Product1_account	No indexing as search key is primary key
Product2_account	No indexing as search key is primary key
Product3_account	No indexing as search key is primary key
Customer	No indexing
Purchase	No indexing as search key is primary key
Complaint	No indexing
Accident	Indexing on accident_date
Repair	No indexing
Requested_Repair	No indexing as search key is primary key
Cust_complaint	No indexing
Complaint_repair	No indexing as search key is primary key
Repair_accident	Indexing on accident_number
Produced_accident	Indexing on accident_number

## Task4:

### Creation of All Tables in Azure SQL Database:

#### Tech\_Staff table:

```
-- Droping tables if exists
DROP table if exists Produced_accident;
DROP table if exists Repair_accident;
DROP table if exists Accident;
drop table if exists Complaint_repair;
DROP table if exists Cust_complaint;
DROP table if exists Requested_repair;
DROP table if exists Repair;
DROP table if exists Complaint;
DROP table if exists Purchase;
DROP table if exists Customer;
DROP table if exists Product3_account;
drop table if exists Product2_account;
DROP table if exists Product1_account;
DROP table if exists Account;
DROP table if exists Product3;
DROP table if exists Product2;
DROP table if exists Product1;
DROP table if exists Product;
DROP table if exists Worker;
DROP table if exists Quality_Controller;
DROP table if exists Degree;
Drop table if exists Tech_Staff;

-- Creating Tech_Staff table
CREATE TABLE Tech_Staff (
    tech_name VARCHAR(40) NOT NULL ,
    tech_address VARCHAR(150),
    tech_salary real,
    tech_position VARCHAR(40),
    PRIMARY KEY(tech_name)
);

```

#### Messages

```
11:37:57 PM  Started executing query at Line 26
Commands completed successfully.
Total execution time: 00:00:00.074
```

#### Degree Table:

```
-- Creating Degree table
CREATE TABLE Degree(
    tech_name VARCHAR(40) NOT NULL,
    degree VARCHAR(10) ,
    PRIMARY KEY (tech_name,degree),
    FOREIGN KEY (tech_name) REFERENCES Tech_Staff(tech_name),
    CONSTRAINT CHK_degree CHECK(degree IN ('BS', 'MS', 'PhD'))
);
```

### Messages

```
11:38:52 PM  Started executing query at Line 34
              Commands completed successfully.
              Total execution time: 00:00:00.086
```

## Quality\_Controller Table:

```
-- Creating Quality_Controller table
CREATE TABLE Quality_Controller(
    qc_name VARCHAR(40) NOT NULL,
    qc_address VARCHAR(150),
    qc_salary real,
    type_of_Product VARCHAR(40) ,
    PRIMARY KEY(qc_name)
);


```

### Messages

```
11:39:24 PM  Started executing query at Line 43
              Commands completed successfully.
              Total execution time: 00:00:00.080
```

## Worker Table:

```
-- Creating Worker table
CREATE TABLE Worker(
    w_name VARCHAR(40) NOT NULL,
    w_address VARCHAR(150),
    w_salary real,
    count_of_Products INT,
    PRIMARY KEY(w_name)
);
```

### Messages

```
11:39:59 PM  Started executing query at Line 53
              Commands completed successfully.
              Total execution time: 00:00:00.075
```

## Product Table:

```
CREATE TABLE Product(
    product_ID INT NOT NULL,
    produced_date DATE,
    making_time float,
    produced_person VARCHAR(40),
    tested_person VARCHAR(40),
    PRIMARY KEY(product_ID),
    FOREIGN KEY (produced_person) REFERENCES Worker(w_name),
    FOREIGN KEY (tested_person) REFERENCES Quality_Controller(qc_name)
);


```

### Messages

```
11:40:35 PM      Started executing query at Line 62
                  Commands completed successfully.
                  Total execution time: 00:00:00.191
```

## Product1 Table:

```
-- Creating Product1 table
CREATE TABLE Product1(
    product_ID INT NOT NULL,
    size VARCHAR(40) ,
    software_name VARCHAR(40),
    PRIMARY KEY(product_ID),
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

### Messages

```
11:41:09 PM      Started executing query at Line 76
                  Commands completed successfully.
                  Total execution time: 00:00:00.076
```

## Product2 Table:

```
-- Creating Product2 table
CREATE TABLE Product2(
    product_ID INT NOT NULL,
    size VARCHAR(40),
    color VARCHAR(20),
    PRIMARY KEY(product_ID),
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

#### Messages

```
11:41:36 PM  Started executing query at Line 85
              Commands completed successfully.
              Total execution time: 00:00:00.069
```

## Product3 Table:

```
-- Creating Product3 table
CREATE TABLE product3(
    product_ID INT NOT NULL,
    size VARCHAR(40),
    p_weight FLOAT,
    PRIMARY KEY(product_ID),
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

#### Messages

```
11:42:02 PM  Started executing query at Line 94
              Commands completed successfully.
              Total execution time: 00:00:00.108
```

## Account Table:

```
-- Creating Account table
CREATE TABLE Account(
    account_number VARCHAR(40) NOT NULL,
    created_date DATE,
    cost REAL,
    PRIMARY KEY(account_number)
);
```

#### Messages

```
11:42:30 PM  Started executing query at Line 103
              Commands completed successfully.
              Total execution time: 00:00:00.075
```

## Product1\_account Table:

```
-- Creating Product1_account table
CREATE TABLE Product1_account(
    product_ID INT,
    account_number VARCHAR(40),
    PRIMARY KEY(product_ID),
    FOREIGN KEY(product_ID) REFERENCES Product1(product_ID),
    FOREIGN KEY(account_number) REFERENCES Account(account_number)
);
```

#### Messages

```
11:42:54 PM  Started executing query at Line 111
Commands completed successfully.
Total execution time: 00:00:00.067
```

### **Product2\_account Table:**

```
-- Creating Product2_account table
CREATE TABLE Product2_account(
    product_ID INT,
    account_number VARCHAR(40),
    PRIMARY KEY(product_ID),
    FOREIGN KEY(product_ID) REFERENCES Product2(product_ID),
    FOREIGN KEY(account_number) REFERENCES Account(account_number)
);
```

#### Messages

```
11:43:30 PM  Started executing query at Line 120
Commands completed successfully.
Total execution time: 00:00:00.156
```

### **Product3\_account Table:**

```
-- Creating Product3_account table
CREATE TABLE Product3_account(
    product_ID INT,
    account_number VARCHAR(40),
    PRIMARY KEY(product_ID),
    FOREIGN KEY(product_ID) REFERENCES Product3(product_ID),
    FOREIGN KEY(account_number) REFERENCES Account(account_number)
);
```

#### Messages

```
11:44:04 PM  Started executing query at Line 129
Commands completed successfully.
Total execution time: 00:00:00.070
```

### **Customer:**

```
-- Creating Customer table
CREATE TABLE Customer(
    customer_name VARCHAR(40) NOT NULL,
    customer_address VARCHAR(150) ,
    PRIMARY KEY(customer_name)
);
```

## Messages

11:44:28 PM    Started executing query at Line 138  
Commands completed successfully.  
Total execution time: 00:00:00.069

## Purchase:

```
-- Creating Purchase table
CREATE TABLE Purchase(
    Product_ID INT NOT NULL,
    customer_name VARCHAR(40),
    PRIMARY KEY(Product_ID),
    FOREIGN KEY (Product_ID) REFERENCES Product(product_ID),
    FOREIGN KEY(customer_name) REFERENCES Customer(customer_name)
);
```

## Messages

11:45:01 PM    Started executing query at Line 145  
Commands completed successfully.  
Total execution time: 00:00:00.067

## Complaint:

```
-- Creating Complaint table
CREATE TABLE Complaint(
    complaint_id VARCHAR(40) NOT NULL,
    complaint_date DATE,
    comp_desc VARCHAR(150),
    treatment VARCHAR(20),
    PRIMARY KEY (complaint_id)
);
```

## Messages

11:45:31 PM    Started executing query at Line 154  
Commands completed successfully.  
Total execution time: 00:00:00.064

## Accident:

```
-- Creating Accident table
CREATE TABLE Accident(
    accident_number VARCHAR(10) NOT NULL,
    accident_date DATE,
    days_lost INT ,
    PRIMARY KEY(accident_number)
);
```

### Messages

```
11:45:55 PM  Started executing query at Line 163
Commands completed successfully.
Total execution time: 00:00:00.070
```

## Repair:

```
-- Creating Repair table
CREATE TABLE Repair(
    product_ID INT,
    tech_name VARCHAR(40),
    date_repaired Date,
    PRIMARY KEY (product_ID),
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID),
    FOREIGN KEY (tech_name) REFERENCES Tech_Staff(tech_name)
);
```

### Messages

```
11:46:18 PM  Started executing query at Line 171
Commands completed successfully.
Total execution time: 00:00:00.065
```

## Requested\_repair:

```
-- Creating Requested_repair table
CREATE TABLE Requested_repair(
    product_ID INT,
    qc_name VARCHAR(40),
    PRIMARY KEY (product_ID,qc_name),
    FOREIGN KEY (product_ID) REFERENCES Repair(product_ID),
    FOREIGN KEY (qc_name) REFERENCES Quality_controller(qc_name)
);
```

### Messages

```
11:46:47 PM  Started executing query at Line 181
Commands completed successfully.
Total execution time: 00:00:00.067
```

## Cust\_complaint:

```
-- Creating Cust_complaint table
CREATE TABLE Cust_complaint(
    Complaint_ID VARCHAR(40) ,
    Product_ID INT NOT NULL,
    PRIMARY KEY (Complaint_ID,product_ID),
    FOREIGN KEY (Complaint_ID) REFERENCES Complaint(Complaint_ID),
    FOREIGN KEY (Product_ID) REFERENCES Purchase(Product_ID)
);
```

## Messages

```
11:47:12 PM  Started executing query at Line 190
              Commands completed successfully.
              Total execution time: 00:00:00.085
```

## **Complaint\_repair:**

```
-- Creating Complaint_repair table
CREATE TABLE Complaint_repair(
    complaint_ID VARCHAR(40),
    product_ID INT,
    PRIMARY KEY(complaint_ID,product_ID),
    FOREIGN KEY (complaint_ID) REFERENCES Complaint(complaint_ID),
    FOREIGN KEY (product_ID) REFERENCES Repair(product_ID)
);
```

## Messages

```
11:47:41 PM  Started executing query at Line 200
              Commands completed successfully.
              Total execution time: 00:00:00.072
```

## **Repair\_accident:**

```
-- Creating Repair_accident table
CREATE TABLE Repair_accident(
    Product_ID INT,
    Accident_number VARCHAR(10),
    PRIMARY KEY(Product_ID),
    FOREIGN KEY (Product_ID) REFERENCES Repair(Product_ID),
    FOREIGN KEY (Accident_number) REFERENCES Accident(Accident_number)
);
```

## Messages

```
11:48:12 PM  Started executing query at Line 209
              Commands completed successfully.
              Total execution time: 00:00:00.084
```

## **Produced\_accident:**

```
-- Creating Produced_accident table
CREATE TABLE Produced_accident(
    Product_ID INT,
    Accident_number VARCHAR(10) ,
    PRIMARY KEY (Product_ID),
    FOREIGN KEY (Accident_number) REFERENCES Accident(Accident_number),
    FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID)
);
```

## Messages

11:48:36 PM    Started executing query at Line 219  
Commands completed successfully.  
Total execution time: 00:00:00.096

The screenshot shows the SSMS interface with the following details:

- Object Explorer (Left):** Shows the database structure under "Servers" (moth0000-sql-server.database.windows.net, cs-dsa-4513-s...). The "Tables" node is expanded, listing numerous tables such as dbo.Accident, dbo.Account, etc.
- Results Pane (Right):** Displays the execution history of a query. The results are as follows:

```
22  DROP t
23  DROP t
24  Drop t
25
26
27
28  CREATE
29    te
30    te
31    te
32    te
33    PF
34 );
35
36  CREATE
37    te
38    de
39    PF
40    FC
41    CC
42 );
43
44
45  CREATE
46    qc
47    qc
48    qc
49    tq
50    PF
51
```

**Messages (Bottom):**

3:52:04 PM

## Indexing:

```
218  
219 CREATE INDEX sal1 ON Tech_Staff(tech_salary);  
220 CREATE INDEX sal2 ON Quality_Controller(qc_salary);  
221 CREATE INDEX sal3 ON Worker(w_salary);  
222 CREATE INDEX pp1 ON product(produced_person);  
223 CREATE INDEX pp2 ON product2(color);  
224 CREATE INDEX A1 ON Accident(accident_date);  
225 Create INDEX a1 ON Repair_accident(accident_number);  
226 Create INDEX a2 ON produced_accident(accident_number);  
227
```

## Messages

---

```
9:44:50 PM      Started executing query at line 219  
                  Commands completed successfully.  
                  Total execution time: 00:00:00.149
```

## Task5:

### 5.1. Transact SQL stored procedures implementing all queries:

#### Query:1)

```
drop PROCEDURE if exists q1;
GO
CREATE PROCEDURE q1
    @emptytype VARCHAR(40),
    @ename VARCHAR(40),
    @address VARCHAR(150),
    @salary real,
    @tech_position VARCHAR(40) ,
    @degree VARCHAR(10),
    @type_of_Product VARCHAR(40),
    @count_of_Products INT

AS
BEGIN
    DECLARE @dupname int = 0
    set @dupname = (select count(name1) from (select tech_name as name1 from Tech_Staff UNION
    select qc_name from Quality_Controller UNION select w_name from Worker) as temp where name1 = @ename )

    IF @dupname < 1
    | BEGIN
    IF @emptytype = 'Tech_staff'
    BEGIN
        INSERT into Tech_Staff VALUES (@ename,@address,@salary,@tech_position)
        INSERT into Degree VALUES (@ename,@degree)
    END

    IF @emptytype = 'QC'
    BEGIN
        INSERT into Quality_Controller VALUES (@ename,@address,@salary,@type_of_Product)
    END

    IF @emptytype = 'QC'
    BEGIN
        INSERT into Quality_Controller VALUES (@ename,@address,@salary,@type_of_Product)
    END

    IF @emptytype = 'Worker'
    BEGIN
        INSERT into Worker VALUES (@ename,@address,@salary,@count_of_Products)
    END
    END
END
```

#### Messages

6:39:44 AM      Started executing query at Line 233  
Commands completed successfully.  
Total execution time: 00:00:00.095

## Query:2)

```
DROP PROCEDURE if EXISTS q2;
GO
CREATE PROCEDURE q2
    @prototype int,
    @product_ID INT,
    @produced_date VARCHAR(40),
    @making_time float,
    @produced_person VARCHAR(40),
    @tested_person VARCHAR(40),
    @repaired_person VARCHAR(40),
    @date_repaired VARCHAR(40),
    @size VARCHAR(40) ,
    @software_name VARCHAR(40),
    @color VARCHAR(20),
    @p_weight FLOAT,
    @Req int

AS
BEGIN

INSERT INTO Product VALUES (@product_ID,cast(@produced_date as date),@making_time,@produced_person,@tested_person)

IF @prototype = 1
BEGIN
    insert into Product1 values (@product_ID,@size,@software_name)
END
IF @prototype = 2
BEGIN
    insert into Product2 values (@product_ID,@size,@color)
END
IF @prototype = 3
BEGIN
    insert into Product3 values (@product_ID,@size,@p_weight)
END

IF @Req = 1
BEGIN

    IF @Req = 1
    BEGIN
        insert into Repair values (@product_ID,@repaired_person,@date_repaired)
        insert into Requested_repair values (@product_ID,@tested_person)
    END
END

```

## Messages

6:48:38 AM      Started executing query at Line 258  
Commands completed successfully.  
Total execution time: 00:00:00.376

### Query:3)

```
299  DROP PROCEDURE if EXISTS q3;
300  GO
301  CREATE PROCEDURE q3
302      @customer_name VARCHAR(40),
303      @customer_address VARCHAR(150) ,
304      @Product_ID INT
305
306  AS
307  BEGIN
308      DECLARE @dupcustname int = 0
309      set @dupcustname = (select count(customer_name) from Customer where customer_name=@customer_name)
310
311      if @dupcustname <1
312          BEGIN
313              INSERT INTO customer VALUES(@customer_name,@customer_address);
314              INSERT INTO Purchase VALUES(@Product_ID,@customer_name);
315          END
316      IF @dupcustname >=1
317          BEGIN
318              INSERT INTO Purchase VALUES(@Product_ID,@customer_name);
319          END
320  END
321
322
```

#### Messages

```
6:50:53 AM      Started executing query at Line 301
Commands completed successfully.
Total execution time: 00:00:00.072
```

### Query:4)

```
323  DROP PROCEDURE if EXISTS q4;
324  GO
325  CREATE PROCEDURE q4
326      @account_number VARCHAR(40),
327      @created_date VARCHAR(40),
328      @cost REAL,
329      @product_ID INT,
330      @ptype INT
331
332  AS
333  BEGIN
334
335  IF @ptype=1
336      BEGIN
337          INSERT INTO Account VALUES(@account_number,cast(@created_date as date),@cost);
338          INSERT INTO Product1_account VALUES(@product_ID, @account_number);
339      END
340
341  IF @ptype=2
342      BEGIN
343          INSERT INTO Account VALUES(@account_number,cast(@created_date as date),@cost);
344          INSERT INTO Product2_account VALUES(@product_ID, @account_number);
345      END
346  IF @ptype=3
347      BEGIN
348          INSERT INTO Account VALUES(@account_number,cast(@created_date as date),@cost);
349          INSERT INTO Product3_account VALUES(@product_ID, @account_number);
350      END
351  END
352
```

#### Messages

```
6:51:56 AM      Started executing query at Line 325
Commands completed successfully.
Total execution time: 00:00:00.063
```

## Query:5)

```
354  DROP PROCEDURE if EXISTS q5;
355  GO
356  CREATE PROCEDURE q5
357  @complaint_id VARCHAR(40),
358  @complaint_date VARCHAR(40),
359  @comp_desc VARCHAR(150),
360  @treatment VARCHAR(20),
361  @Product_ID INT ,
362  @req INT,
363  @techname VARCHAR(40),
364  @date_repaired VARCHAR(40)
365  AS
366  BEGIN
367
368  INSERT INTO Complaint VALUES(@complaint_id,cast(@complaint_date as date),@comp_desc,@treatment);
369  INSERT INTO Cust_complaint VALUES(@complaint_id,@Product_ID);
370  IF @req =1
371  BEGIN
372    INSERT INTO Repair VALUES(@Product_ID,@techname,cast(@date_repaired as date));
373    INSERT INTO Complaint_repair VALUES (@complaint_id,@Product_ID);
374  END
375 END
376
```

### Messages

```
6:52:48 AM      Started executing query at Line 356
Commands completed successfully.
Total execution time: 00:00:00.080
```

## Query:6)

```
377  DROP PROCEDURE if EXISTS q6;
378  GO
379  CREATE PROCEDURE q6
380  @accident_number VARCHAR(10) ,
381  @accident_date VARCHAR(40),
382  @days_lost INT ,
383  @Product_ID INT,
384  @accitype VARCHAR(10)
385  AS
386  BEGIN
387
388  IF @accitype = 'type1'
389  BEGIN
390    INSERT INTO Accident VALUES (@accident_number,cast(@accident_date as date),@days_lost);
391    INSERT INTO Produced_accident VALUES(@Product_ID,@accident_number);
392  END
393  IF @accitype = 'type2'
394  BEGIN
395    INSERT INTO Accident VALUES (@accident_number,cast(@accident_date as date),@days_lost);
396    INSERT INTO Repair_accident VALUES(@Product_ID,@accident_number);
397  END
398 END
399
```

### Messages

```
6:53:54 AM      Started executing query at Line 379
Commands completed successfully.
Total execution time: 00:00:00.065
```

## Query:7)

```
399  DROP PROCEDURE if EXISTS q7;
400
401
402  GO
403  CREATE PROCEDURE q7
404  |    @product_id INT
405  AS
406  BEGIN
407
408  select produced_date,making_time from Product
409  where product_ID = @product_id;
410
411  END
```

---

### Messages

```
6:54:35 AM      Started executing query at Line 403
Commands completed successfully.
Total execution time: 00:00:00.061
```

## Query:8)

```
414  DROP PROCEDURE if EXISTS q8;
415  GO
416  CREATE PROCEDURE q8
417  |    @produced_person VARCHAR(40)
418  AS
419  BEGIN
420  |    SELECT product_ID from Product where produced_person=@produced_person;
421  END
```

---

### Messages

```
6:55:25 AM      Started executing query at Line 416
Commands completed successfully.
Total execution time: 00:00:00.063
```

## Query:9)

```
425  DROP PROCEDURE if EXISTS q9;
426  GO
427  CREATE PROCEDURE q9
428  |    @tested_person VARCHAR(40)
429
430  AS
431  BEGIN
432  |    select count(tested_person) as errors from (SELECT tested_person from Product p
433  INNER JOIN Complaint_repair cr on p.product_ID=cr.product_ID) as temp where tested_person = @tested_person;
434
435  END
```

---

### Messages

```
6:56:08 AM      Started executing query at Line 427
Commands completed successfully.
Total execution time: 00:00:00.066
```

## Query:10)

```
438  DROP PROCEDURE if EXISTS q10;
439  GO
440  CREATE PROCEDURE q10
441  | @qc_name VARCHAR(40)
442
443  AS
444  BEGIN
445
446  | select sum(a.cost) as total_cost from Requested_repair rr
447  | INNER JOIN Product3_account p3 on rr.product_ID=p3.product_ID
448  | INNER JOIN Account a on p3.account_number=a.account_number
449  | WHERE rr.qc_name = @qc_name;
450
451
452  END
453
```

### Messages

6:56:53 AM    Started executing query at Line 440  
Commands completed successfully.  
Total execution time: 00:00:00.386

## Query:11)

```
469  DROP PROCEDURE if EXISTS q11;
470  GO
471  CREATE PROCEDURE q11
472  | @color VARCHAR(20)
473  AS
474  BEGIN
475  | SELECT p.customer_name from Purchase p , Product2 p2
476  | WHERE p.product_ID = p2.product_ID AND p2.color=@color
477  | ORDER BY p.customer_name;
478  END
```

### Messages

6:57:41 AM    Started executing query at Line 456  
Commands completed successfully.  
Total execution time: 00:00:00.066

## Query:12)

```
465  DROP PROCEDURE if EXISTS q12;
466  GO
467  CREATE PROCEDURE q12
468  | @salary real
469
470  AS
471  BEGIN
472  | SELECT tech_name as employee_name,tech_salary as salary,tech_address as address from Tech_Staff where tech_salary>@salary UNION
473  | SELECT qc_name as employee_name,qc_salary as salary ,qc_address as address from Quality_Controller where qc_salary>@salary UNION
474  | SELECT w_name as employee_name, w_salary as salary,w_address as address from Worker WHERE w_salary > @salary;
475
476  END
```

### Messages

6:59:52 AM    Started executing query at Line 467  
Commands completed successfully.  
Total execution time: 00:00:00.070

### Query:13)

```
478  DROP PROCEDURE if EXISTS q13;
479  GO
480  CREATE PROCEDURE q13
481
482  AS
483  BEGIN
484
485      SELECT sum(a.days_lost) as total_days_lost from Accident a , Repair_accident ra , Complaint_repair cr
486      WHERE a. accident_number=ra.Accident_number AND ra.Product_ID=cr.product_ID;
487  END
488
```

---

#### Messages

7:02:30 AM    [Started executing query at Line 480](#)  
Commands completed successfully.  
Total execution time: 00:00:00.095

### Query:14)

```
488
489  DROP PROCEDURE if EXISTS q14;
490  GO
491  CREATE PROCEDURE q14
492  |   @year INT
493  AS
494  BEGIN
495
496      SELECT avg(temp.cost) as avg_cost from
497      (SELECT a.cost,p.produced_date from Product1_account p1
498      INNER JOIN Account a on p1.account_number=a.account_number
499      INNER JOIN Product as p on p1.product_ID=p.product_ID
500      WHERE YEAR(p.produced_date)=@year UNION
501      SELECT a.cost,p.produced_date from Product2_account as p2
502      INNER JOIN Account a on p2.account_number= a.account_number
503      INNER JOIN Product as p on p2.product_ID=p.product_ID
504      WHERE YEAR(p.produced_date)=@year UNION
505      SELECT a.cost,p.produced_date from Product3_account as p3
506      INNER JOIN Account a on p3.account_number= a.account_number
507      INNER JOIN Product as p on p3.product_ID=p.product_ID
508      WHERE YEAR(p.produced_date)=@year) as temp
509  END
```

---

#### Messages

12:11:20 PM    [Started executing query at Line 491](#)  
Commands completed successfully.  
Total execution time: 00:00:00.061

## Query:15)

```
511  DROP PROCEDURE if EXISTS q15;
512  GO
513  CREATE PROCEDURE q15
514      @date1 DATE,
515      @date2 DATE
516  AS
517  BEGIN
518
519      DELETE produced_accident FROM (SELECT accident_number from Accident WHERE accident_date BETWEEN @date1 AND @date2) as temp
520      WHERE Produced_accident.Accident_number=temp.accident_number;
521      DELETE Repair_accident FROM (SELECT accident_number from Accident WHERE accident_date BETWEEN @date1 AND @date2) as temp
522      WHERE Repair_accident.Accident_number=temp.accident_number;
523      DELETE from Accident WHERE accident_date BETWEEN @date1 AND @date2;
524
525  END
526
527
```

### Messages

```
12:12:08 PM Started executing query at Line 513
Commands completed successfully.
Total execution time: 00:00:00.069
```

## 5.2-The Java source program, screenshots showing its successful compilation



```
1 package Mothukuri;
2
3 import java.io.File;
4 import java.util.Date;
5 import java.util.Scanner;
6 import java.io.FileWriter;
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.Statement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.sql.DriverManager;
13
14 public class Mothukuri_Kovida_IP_Task5 {
15     final static String HOSTNAME = "moth0000-sql-server.database.windows.net";
16     final static String DBNAME = "cs-dsa-4513-sql-db";
17     final static String USERNAME = "moth0000";
18     final static String PASSWORD = "Srinuani@9398";
19
20     // Database connection string
21     final static String URL =
22     String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;" +
23         + "trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
24         HOSTNAME, DBNAME, USERNAME, PASSWORD);
25
26     // User input prompt//
27     final static String PROMPT =
28         "\nPlease select any of the options below: \n" +
29         "1) Enter a new employee (option 1) \n" +
30         "2) Enter a new product associated with the person who made the product, repaired the product if it is repaired, or checked the product (option 2) \n" +
31         "3) Enter a customer associated with some products (option 3) \n" +
32         "4) Create a new account associated with a product (option 4) \n" +
33         "5) Enter a complaint associated with a customer and product (option 5) \n" +
34         "6) Enter an accident associated with an appropriate employee and product (option 6) \n" +
35         "7) Retrieve the date produced and time spent to produce a particular product (option 7) \n" +
36         "8) Retrieve all products made by a particular worker (option 8) \n"+
```

```
Mothukuri_Kovida_IP_Task5b.java x
37
38     "9) Retrieve the total number of errors a particular quality controller made.This is the total number of products certified"
39     + " by this controller and got some complaints (option 9)\n"+
40     "10) Retrieve the total costs of the products in the product3 category which were repaired at the \r\n"
41     + "request of a particular quality controller(option 10) \n"+
42     "11) Retrieve all customers (in name order) who purchased all products of a particular color (option 11) \n"+
43     "12) Retrieve all employees whose salary is above a particular salary (option 12)\n"+
44     "13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints (option 13) \n"+
45     "14) Retrieve the average cost of all products made in a particular year (option 14)\n"+
46     "15) Delete all accidents whose dates are in some range (option 15)\n"+
47     "16) Import: enter new employees from a data file until the file is empty (the user must enter the input file name) (option 16)\n"+
48     "17) Export: Retrieve all customers (in name order) who purchased all products of a particular color \n"
49     + "and output them to a data file instead of screen (the user must enter the output file name) (option 17)\n"+
50     "18) Exit [option 18] ";
51
52 public static void main(String[] args) throws SQLException {
53
54     System.out.println("Now you can start using the application");
55     final Scanner input = new Scanner(System.in); // Scanner is used to collect the user input
56     String option = ""; // Initialize user option selection as nothing
57     while (!option.equals("18")) { // As user for options until option 18 is selected
58         System.out.println(PROMPT); // Print the available options
59         option = input.nextLine(); // Read in the user option selection
60
61         final Connection connection = DriverManager.getConnection(URL);
62         switch (option) { // Switch between different options
63             // employee insertion
64             case "1":
65                 try {
66
67                     String emptype,empname,empaddress,tech_position,degree,type_of_product;
68                     int count_of_products;
69                     Float empsalary;
70
71                     System.out.println("enter employee type");
72                     input.nextLine();
73                     emptype= input.nextLine(); // input of employee type
74
75                     System.out.println("enter employee name");
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1390
1391
1392
1393
1394
1395
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1490
1491
1492
1493
1494
1495
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2190
2191
2192
2193
2194
2195
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2288
2289
2290
2291
2292
2293
2294
2295
2295
2296
2297
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2388
2389
2390
2391
2392
2393
2394
2395
2395
2396
239
```

```

Mothukuri_Kovida_IP_Task5b.java X
1   import java.util.Scanner;
2   class Employee {
3       String empname, address, degree;
4       int salary, type_of_product;
5       float p_weight;
6       String protype;
7       int product_id, prod_date, prod_per, test_per, rep_per, date_rep, size;
8       String soft_name, color;
9       float req;
10      Scanner input = new Scanner(System.in);
11      System.out.println("Enter employee name");
12      empname = input.nextLine();
13      System.out.println("Enter employee address");
14      address = input.nextLine();
15      System.out.println("Enter employee degree");
16      degree = input.nextLine();
17      System.out.println("Enter employee salary");
18      salary = input.nextInt();
19      System.out.println("Enter employee type of product");
20      type_of_product = input.nextInt();
21      System.out.println("Enter employee soft_name");
22      soft_name = input.nextLine();
23      System.out.println("Enter employee color");
24      color = input.nextLine();
25      System.out.println("Enter employee req");
26      req = input.nextFloat();
27      System.out.println("Employee record inserted successfully");
28      System.out.println("Employee record inserted successfully");
29      System.out.println("Employee record inserted successfully");
30      System.out.println("Employee record inserted successfully");
31      System.out.println("Employee record inserted successfully");
32      System.out.println("Employee record inserted successfully");
33      System.out.println("Employee record inserted successfully");
34      System.out.println("Employee record inserted successfully");
35      System.out.println("Employee record inserted successfully");
36      System.out.println("Employee record inserted successfully");
37      System.out.println("Employee record inserted successfully");
38      System.out.println("Employee record inserted successfully");
39      System.out.println("Employee record inserted successfully");
40      System.out.println("Employee record inserted successfully");
41      System.out.println("Employee record inserted successfully");
42      System.out.println("Employee record inserted successfully");
43      System.out.println("Employee record inserted successfully");
44      System.out.println("Employee record inserted successfully");
45      System.out.println("Employee record inserted successfully");
46      System.out.println("Employee record inserted successfully");
47      System.out.println("Employee record inserted successfully");
48      System.out.println("Employee record inserted successfully");
49      System.out.println("Employee record inserted successfully");
50      System.out.println("Employee record inserted successfully");
51      System.out.println("Employee record inserted successfully");
52      System.out.println("Employee record inserted successfully");
53      System.out.println("Employee record inserted successfully");
54      System.out.println("Employee record inserted successfully");
55      System.out.println("Employee record inserted successfully");
56      System.out.println("Employee record inserted successfully");
57      System.out.println("Employee record inserted successfully");
58      System.out.println("Employee record inserted successfully");
59      System.out.println("Employee record inserted successfully");
60      System.out.println("Employee record inserted successfully");
61      System.out.println("Employee record inserted successfully");
62      System.out.println("Employee record inserted successfully");
63      System.out.println("Employee record inserted successfully");
64      System.out.println("Employee record inserted successfully");
65      System.out.println("Employee record inserted successfully");
66      System.out.println("Employee record inserted successfully");
67      System.out.println("Employee record inserted successfully");
68      System.out.println("Employee record inserted successfully");
69      System.out.println("Employee record inserted successfully");
70      System.out.println("Employee record inserted successfully");
71      System.out.println("Employee record inserted successfully");
72      System.out.println("Employee record inserted successfully");
73      System.out.println("Employee record inserted successfully");
74      System.out.println("Employee record inserted successfully");
75      System.out.println("Employee record inserted successfully");
76      System.out.println("Employee record inserted successfully");
77      System.out.println("Employee record inserted successfully");
78      System.out.println("Employee record inserted successfully");
79      System.out.println("Employee record inserted successfully");
80      System.out.println("Employee record inserted successfully");
81      System.out.println("Employee record inserted successfully");
82      System.out.println("Employee record inserted successfully");
83      System.out.println("Employee record inserted successfully");
84      System.out.println("Employee record inserted successfully");
85      System.out.println("Employee record inserted successfully");
86      System.out.println("Employee record inserted successfully");
87      System.out.println("Employee record inserted successfully");
88      System.out.println("Employee record inserted successfully");
89      System.out.println("Employee record inserted successfully");
90      System.out.println("Employee record inserted successfully");
91      System.out.println("Employee record inserted successfully");
92      System.out.println("Employee record inserted successfully");
93      System.out.println("Employee record inserted successfully");
94      System.out.println("Employee record inserted successfully");
95      System.out.println("Employee record inserted successfully");
96      System.out.println("Employee record inserted successfully");
97      System.out.println("Employee record inserted successfully");
98      System.out.println("Employee record inserted successfully");
99      System.out.println("Employee record inserted successfully");
100     System.out.println("Employee record inserted successfully");
101    System.out.println("Employee record inserted successfully");
102   System.out.println("Employee record inserted successfully");
103  System.out.println("Employee record inserted successfully");
104 System.out.println("Employee record inserted successfully");
105 Statement statement1 = connection.createStatement();
106 final int rows_inserted = statement1.executeUpdate(option1);
107 System.out.println(String.format("employee record inserted Successfully. %d rows inserted.", rows_inserted));
108 }
109 catch(Exception e) {
110     System.out.println("you got an error,exception in option 1" +e);
111 }
112 break;
113 //product insertion
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153

```

```

Mothukuri_Kovida_IP_Task5b.java X
116 case "2":
117     try {
118
119         String prototype,prod_date,prod_per,test_per,rep_per,date_rep,size,soft_name,color;
120         int product_id,making_time,req;
121         Float p_Weight;
122
123         System.out.println("Enter the type of product");
124         input.nextLine();
125         prototype = input.nextLine(); // type of product
126
127         System.out.println("Enter Product id");
128         product_id = input.nextInt(); // product id
129
130         System.out.println("Enter Produced date");
131         input.nextLine();
132         prod_date = input.nextLine(); // produced date
133
134         System.out.println("Enter making time");
135         making_time = input.nextInt(); //making time
136
137
138         System.out.println("Enter Produced person name");
139         input.nextLine();
140         prod_per= input.nextLine(); // produced person
141
142         System.out.println("Enter tested person name");
143         test_per = input.nextLine(); // tested person
144
145
146         System.out.println("Enter Repaired person name ");
147         rep_per= input.nextLine(); // repaired person
148
149
150         System.out.println("Enter Repaired date");
151         date_rep= input.nextLine(); // repaired date
152
153         System.out.println("Enter the size of product");

```

```

Mothukuri_Kovida_IP_Task5bjava × 
153     System.out.println("Enter the size of product");
154     size= input.nextLine(); // size of product
155
156     System.out.println("Enter software Name");
157     soft_name= input.nextLine(); // software name
158
159     System.out.println("Enter Color of product");
160     color= input.nextLine(); // color of product
161
162     System.out.println("Enter Weight of product");
163     p_Weight= input.nextFloat(); // weight of product3
164
165
166     System.out.println("if product repaired enter 1 or -1");
167     req= input.nextInt(); // repaired 1 or -1
168
169
170     final String option2 = " EXEC q2 @prototype = '"+prototype+"', @product_ID = '"+product_id+"', "
171         + " @produced_date = '"+prod_date+"', @making_time = '"+making_time+"', "
172         + " @produced_person = '"+prod_per+"', @tested_person= '"+test_per+"', "
173         + " @repaired_person = '"+rep_per+"', @date_repaired = '"+date_rep+"', "
174         + " @size = '"+size+"', @software_name = '"+soft_name+"', "
175         + " @color = '"+color+"', @p_weight= '"+p_Weight+"', @req = '"+req+"'"; 
176
177
178     System.out.println("Connecting to the database...");
179
180     final Statement statement2 = connection.createStatement();
181
182     final int rows_inserted = statement2.executeUpdate(option2);
183     System.out.println(String.format("product record inserted Successfully. %d rows inserted.", rows_inserted));
184 }
185
186
187     catch(Exception e ) {
188         System.out.println("you got an error,exception in option 2" +e);
189     }
190     break;

```

---

```

Mothukuri_Kovida_IP_Task5bjava × 
191     case "3":
192         try {
193
194             String customer_name,customer_address;
195             int Product_ID;
196
197             System.out.println("Enter customer name");
198             input.nextLine();
199             customer_name= input.nextLine(); //customer name
200
201             System.out.println("Enter customer address");
202             customer_address= input.nextLine(); // customer address
203
204
205             System.out.println("Enter customer purchased product id ");
206             Product_ID= input.nextInt(); // product id
207
208             final String option3 = "EXEC q3 @customer_name = '"+customer_name+"', "
209                 + " @customer_address = '"+ customer_address+"', @product_ID = '"+Product_ID+"'"; 
210
211
212             System.out.println("Connecting to the database...");
213             final Statement statement3 = connection.createStatement();
214             final int rows_inserted = statement3.executeUpdate(option3);
215             System.out.println(String.format("Customer record inserted Successfully. %d rows inserted.", rows_inserted));
216
217         } catch(Exception e ) {
218             System.out.println("you got an error,exception in option 3" +e);
219         }
220
221         break;

```

```

223         break;
224
225     case "4":
226         try {
227
228         int acc_product_id,ptype;
229         Float Cost;
230         String created_date,account_number;
231
232         System.out.println("enter Account number");
233         input.nextLine();
234         account_number= input.nextLine(); // account number
235
236         System.out.println("enter date created");
237         created_date= input.nextLine(); // created date
238
239         System.out.println("enter product cost");
240         Cost= input.nextFloat(); // cost
241
242         System.out.println("enter product id");
243         acc_product_id= input.nextInt(); // product id
244
245         System.out.println("enter account type");
246         input.nextLine();
247         ptype= input.nextInt(); // account type
248
249
250
251         final String option4 = "EXEC q4  @account_number = '"+account_number+"', "
252             + "@created_date = '"+created_date+"', @cost = '"+Cost+"', "
253             + "@product_ID = '"+acc_product_id+"',@ptype = '"+ptype+"'";
254
255         System.out.println("Connecting to the database...");
256         final Statement statement4 = connection.createStatement();

```

```

254
255         System.out.println("Connecting to the database...");
256         final Statement statement4 = connection.createStatement();
257
258
259
260         final int rows_inserted = statement4.executeUpdate(option4);
261         System.out.println(String.format("account record inserted Successfully. %d rows inserted.", rows_inserted));
262     }
263     catch(Exception e ) {
264         System.out.println("you got an error,exception in option 4" +e);
265     }
266     break;

```

```

268     case "5":
269         try {
270
271         int Complaint_Product_id,repreg;
272         String complaint_ID,complaint_date,comp_desc,treatment,techname,date_repaired;
273
274         System.out.println("enter Complaint id");
275         input.nextLine();
276         complaint_ID= input.nextLine(); // complaint id
277
278         System.out.println("enter complaint date");
279         complaint_date= input.nextLine(); //complaint date
280
281
282         System.out.println("enter description");
283         comp_desc= input.nextLine(); // description
284
285         System.out.println("treatment expected");
286         treatment= input.nextLine(); // treatment expected
287
288         System.out.println("enter product id");
289         Complaint_Product_id= input.nextInt(); // product id
290
291         System.out.println("enter 1 if possible to repair or 0");
292         input.nextLine();
293         repreg= input.nextInt();
294
295         System.out.println("enter technical staff name");
296         input.nextLine();
297         techname= input.nextLine(); //technical staff name
298
299         System.out.println("enter date of repair");
300         date_repaired= input.nextLine(); // repair date
301
302         final String option5 = "EXEC g5  @complaint_ID = '"+complaint_ID+"', "
303             + "@complaint_date = '"+complaint_date+"', @comp_desc = '"+comp_desc+"', "
304             + "@treatment = '"+treatment+"',@product_ID = '"+Complaint_Product_id+"',"
305             + " @repreg = '"+repreg+"', @techname =  '"+techname+"',"

```

```

Mothukuri_Kovida_IP_Task5b.java ×
298     System.out.println("enter date of repair");
299     date_repaired= input.nextLine(); // repair date
300
301     final String option5 = "EXEC q5 @complaint_ID = '"+complaint_ID+"', "
302         + "@complaint_date = '"+complaint_date+"', @comp_desc = '"+comp_desc+"', "
303         + "@treatment = '"+treatment+"', @product_ID = '"+Complaint_Product_id+"', "
304         + " @req = '"+req+"', @techname = '"+techname+"', "
305         + " @date_repaired = '"+date_repaired+"'";
306
307     System.out.println("Connecting to the database...");
308     final Statement statement5 = connection.createStatement();
309
310
311
312     final int rows_inserted = statement5.executeUpdate(option5);
313     System.out.println(String.format("complaint record inserted Successfully. %d rows inserted.", rows_inserted));
314 }
315     catch(Exception e ) {
316         System.out.println("you got an error,exception in option 5" +e);
317     }
318
319 break;
320

```

```

Mothukuri_Kovida_IP_Task5b.java ×
322     case"6":
323         try {
324
325             String accident_number,accident_date,accitype;
326             int days_lost,Product_id;
327
328             System.out.println("enter Accident number");
329             input.nextLine();
330             accident_number= input.nextLine(); // Accident number
331
332             System.out.println("enter accident date");
333             accident_date= input.nextLine(); // accident date
334
335             System.out.println("enter days lost");
336             days_lost= input.nextInt(); // days lost due to accident
337
338             System.out.println("enter accident product id:");
339             Product_id= input.nextInt(); // product id
340
341             System.out.println("Enter accident type(type1 or type2)");
342             input.nextLine();
343             accitype= input.nextLine(); // accident type
344
345             final String option6 = "EXEC q6 @accident_number = '"+accident_number+"', "
346                 + " @accident_date = '"+accident_date+"', @days_lost = '"+days_lost+"', "
347                 + " @product_ID = '"+Product_id+"',@accitype = '"+accitype+"'";
348
349
350             System.out.println("Connecting to the database...");
351             final Statement statement6 = connection.createStatement();
352
353             final int rows_inserted = statement6.executeUpdate(option6);
354             System.out.println(String.format("complaint record inserted Successfully. %d rows inserted.", rows_inserted));
355 }
356     catch(Exception e ) {
357         System.out.println("you got an error,exception in option 6" +e);
358     }
359
break;

```

```
Mothukuri_Kovida_IP_Task5b.java X
360
361
362     case"7":
363         try {
364             int product_id;
365             System.out.println("Please enter Product id");
366             input.nextLine();
367             product_id= input.nextInt() // product id
368             final String option7 = "EXEC q7 @product_id = '"+product_id+"'"; 
369             try (final Statement statement7 = connection.createStatement());
370                 final ResultSet resultSet1 = statement7.executeQuery(option7));
371                 System.out.println("Results");
372                 input.nextLine();
373                 System.out.println(" Produced_date | making_time ");
374
375                 while (resultSet1.next()) {
376                     System.out.println(String.format(" %s | %s ",
377                         resultSet1.getString(1),
378                         resultSet1.getFloat(2)));
379                 }
380             }
381         } catch(Exception e ) {
382             System.out.println("you got an error,exception in option 7" +e);
383         }
384
385
386     break;
```

```
Mothukuri_Kovida_IP_Task5b.java X
386
387     case"8":
388         try {
389             String produced_per;
390             System.out.println("enter produced person name");
391             input.nextLine();
392             produced_per= input.nextLine() // produced person
393
394             final String option8 = "EXEC q8 @produced_person = '"+produced_per+"'"; 
395
396             try (final Statement statement8 = connection.createStatement());
397                 final ResultSet resultSet2 = statement8.executeQuery(option8));
398                 System.out.println(" product_ID ");
399                 while (resultSet2.next()) {
400                     System.out.println(String.format(" %s ",
401                         resultSet2.getInt(1)));
402                 }
403             }
404         } catch(Exception e ) {
405             System.out.println("you got an error,exception in option 8" +e);
406         }
407
408     break;
```

```

Mothukuri_Kovida_IP_Task5b.java ×
410
411
412     case"9":
413
414         try {
415             String tested_per;
416             System.out.println("enter tested person name");
417             input.nextLine();
418             tested_per= input.nextLine(); // tested person
419
420             final String option9 = "EXEC q9 @tested_person = '"+tested_per+"'";
421
422             try (final Statement statement9 = connection.createStatement();
423                  final ResultSet resultSet3 = statement9.executeQuery(option9)){
424                 System.out.println(" total errors ");
425                 while (resultSet3.next()) {
426                     System.out.println(String.format(" %s ", resultSet3.getInt(1)));
427                 }
428             }
429         } catch(Exception e ) {
430             System.out.println("you got an error,exception in option 9" +e);
431         }
432     break;
433
434

```

```

Mothukuri_Kovida_IP_Task5b.java ×
434
435     case"10":
436         try {
437             String qc_name;
438             System.out.println("enter tested person name");
439             input.nextLine();
440             qc_name= input.nextLine(); // tested person
441             final String option10 = "EXEC q10 @qc_name = '"+qc_name+"'"; 
442             try (final Statement statement10 = connection.createStatement();
443                  final ResultSet resultSet4 = statement10.executeQuery(option10)){
444                 System.out.println(" total cost ");
445                 while (resultSet4.next()) {
446                     System.out.println(String.format(" %s ", resultSet4.getFloat(1)));
447                 }
448             }
449         } catch(Exception e ) {
450             System.out.println("you got an error,exception in option 10" +e);
451         }
452     break;
453
454
455
456

```

```

Mothukuri_Kovida_IP_Task5b.java ×
456
457     case"11":
458
459         try {
460             String color;
461             System.out.println("enter color");
462             input.nextLine();
463             color= input.nextLine(); // tested person
464             final String option11 = "EXEC q11 @color = '"+color+"'"; 
465
466             try (final Statement statement11 = connection.createStatement();
467                  final ResultSet resultSet5 = statement11.executeQuery(option11)){
468                 System.out.println(" Customer names ");
469                 while (resultSet5.next()) {
470                     System.out.println(String.format(" %s ", resultSet5.getString(1)));
471                 }
472             }
473         } catch(Exception e ) {
474             System.out.println("you got an error,exception in option 11" +e);
475         }
476     break;
477
478
479

```

```

Mothukuri_Kovida_IP_Task5b.java ×
479
480     case"12":
481         try {
482             Float salary;
483             System.out.println("enter salary");
484             input.nextLine();
485             salary= input.nextFloat(); // salary
486             final String option12 = "EXEC q12 @salary = '"+salary+"'"; 
487             try (final Statement statement12 = connection.createStatement());
488                 final ResultSet resultSet6 = statement12.executeQuery(option12));
489                 System.out.println(" Employee name | salary | address ");
490                 while (resultSet6.next()) {
491                     System.out.println(String.format(" %s | %s | %s ", 
492                         resultSet6.getString(1),
493                         resultSet6.getFloat(2),
494                         resultSet6.getString(3)));
495                 }
496             }
497         }
498         catch(Exception e ) {
499             System.out.println("you got an error,exception in option 12" +e);
500         }
501
502     break;

```

```

Mothukuri_Kovida_IP_Task5b.java ×
503
504     case"13":
505         try {
506             final String option13 = "EXEC q13;";
507             try (final Statement statement13 = connection.createStatement());
508                 final ResultSet resultSet7 = statement13.executeQuery(option13));
509                 System.out.println(" total days lost");
510                 while (resultSet7.next()) {
511                     System.out.println(String.format(" %s ", resultSet7.getInt(1)));
512                 }
513             }
514         }
515         catch(Exception e ) {
516             System.out.println("you got an error,exception in option 13" +e);
517         }
518     break;

```

```

Mothukuri_Kovida_IP_Task5b.java ×
520
521     case"14":
522
523         try {
524
525             int year;
526             System.out.println("enter year");
527             input.nextLine();
528             year= input.nextInt(); // year
529             final String option14 = "EXEC q14 @year = '"+year+"'"; 
530
531             try (final Statement statement14 = connection.createStatement());
532                 final ResultSet resultSet8 = statement14.executeQuery(option14));
533                 System.out.println(" average cost");
534                 while (resultSet8.next()) {
535                     System.out.println(String.format(" %s ", resultSet8.getFloat(1)));
536                 }
537             }
538             catch(Exception e ) {
539                 System.out.println("you got an error,exception in option 14" +e);
540             }
541
542         break;

```

```

Mothukuri_Kovida_IP_Task5b.java X
556     case "15":
557
558         try {
559             String date1,date2;
560             System.out.println("enter date1");
561             input.nextLine();
562             date1= input.nextLine(); // date1
563
564             System.out.println("enter date2");
565             date2= input.nextLine(); //date2
566             final String option15 = "EXEC q15 @date1='"+date1+"',@date2='"+date2+"'"; 
567
568             final Statement statement15 = connection.createStatement();
569                 statement15.executeUpdate(option15);
570                 System.out.println("deleted succesfully");
571             }
572             catch(Exception e ) {
573                 System.out.println("you got an error,exception in option 15" +e);
574             }
575         break;

```

```

Mothukuri_Kovida_IP_Task5b.java X
577     case "16":
578         try {
579             String path, eachline,emptytype,empname,empaddress,tech_position,degree,type_of_product;
580             int empsalary,count_of_products;
581             System.out.println("enter the file name for import"); // read in the file name
582             path = input.nextLine();
583
584             File csv_file = new File(path);
585             Scanner input1= new Scanner(csv_file);
586             while(input1.hasNextLine()) {
587                 eachline = input1.nextLine(); // read in the line
588
589                 String[] strings = eachline.split(",");
590
591                 emptytype = strings[0];
592                 empname = strings[1];
593                 empaddress = strings[2];
594                 empsalary = Integer.parseInt(strings[3]);
595                 tech_position = strings[4];
596                 degree = strings[5];
597                 type_of_product = strings[6];
598                 count_of_products = Integer.parseInt(strings[7]);
599
600                 final String option16 = "EXEC q1 @emptytype = '"+emptytype+"', @ename = '"+empname+"', "
601                     + "@address = '"+empaddress+"',@salary = '"+empsalary+"',@tech_position = '"+tech_position+"',"
602                     + " @degree = '"+degree+"', @type_of_product = '"+type_of_product+"', @count_of_Products = '"+count_of_products+"'"; 
603
604                 final Statement statement16 = connection.createStatement();
605                 statement16.executeUpdate(option16);
606
607                 System.out.println("inserted successfully");
608
609             } catch(Exception e ) {
610                 System.out.println("you got an error,exception in option 16" +e);
611             }
612         break;

```

```

J *Mothukuri_Kovida_IP_Task5b.java ×

614
615     case "17":
616         try {
617             System.out.println("Enter the path with file name");
618             String path = input.next();
619
620             System.out.println("Enter the color of product");
621             String color = input.next();
622
623             FileWriter fw = new FileWriter(path);
624
625             final String option17 = "EXEC q11 @color = '" + color + "';";
626
627             try (final Statement statement17 = connection.createStatement();
628                  final ResultSet resultSet10 = statement17.executeQuery(option17)) {
629
630                 while (resultSet10.next()) {
631                     fw.write(resultSet10.getString(1) + "\n");
632                 }
633                 fw.close();
634
635             } catch (Exception e) {
636                 System.out.println("you got an error,exception in option 17"+e);
637             }
638
639         } catch (Exception e) {
640             System.out.println("you got an error,exception in option 17"+e);
641         }
642
643         break;
644
645
646 }

647     case "18":
648         System.out.println("You choose to Quit! bye");
649
650     }
651
652
653 }
654
655
656
657 }

```

## Output of java code:

Console ×

Mothukuri\_Kovida\_IP\_Task5b [Java Application] C:\Users\motha\Downloads\eclipse-jee-2022-09-R-win32-x86\_64\eclipse\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.4.v20220903-1038\jre\bin\javaw.exe (Nov 20, 2022, 5:05:12 PM)

Now you can start using the application

Please select any of the options below:

- 1) Enter a new employee (option 1)
- 2) Enter a new product associated with the person who made the product, repaired the product if it is repaired, or checked the product (option 2)
- 3) Enter a customer associated with some products (option 3)
- 4) Create a new account associated with a product (option 4)
- 5) Enter a complaint associated with a customer and product (option 5)
- 6) Enter an accident associated with an appropriate employee and product (option 6)
- 7) Retrieve the date produced and time spent to produce a particular product (option 7)
- 8) Retrieve all products made by a particular worker (option 8)
- 9) Retrieve the total number of errors a particular quality controller made. This is the total number of products certified by this controller and got some complaints
- 10) Retrieve the total costs of the products in the product3 category which were repaired at the request of a particular quality controller(option 10)
- 11) Retrieve all customers (in name order) who purchased all products of a particular color (option 11)
- 12) Retrieve all employees whose salary is above a particular salary (option 12)
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints (option 13)
- 14) Retrieve the average cost of all products made in a particular year (option 14)
- 15) Delete all accidents whose dates are in some range (option 15)
- 16) Import: enter new employees from a data file until the file is empty (the user must enter the input file name) (option 16)
- 17) Export: Retrieve all customers (in name order) who purchased all products of a particular color and output them to a data file instead of screen (the user must enter the output file name) (option 17)
- 18) Exit (option 18)

## Task6. Java Program Execution

### 6.1. query1 testing screenshots:

For inserting values into the table, following data has been used:

#### Tech\_Staff:

	A	B	C	D	E	F	G	H
1	emptytype	ename	address	salary	tech_position	degree	type_of_product	count_of_products
2	Tech_staff	Ela	ajo	1000	lead	BS	NULL	-1
3	Tech_staff	anna	avondale	2000	manager	MS	NULL	-1
4	Tech_staff	olaf	bisbee	3000	engineer	PhD	NULL	-1
5	Tech_staff	kristoff	casa grande	40000	lead	BS	NULL	-1
6	Tech_staff	hans	chandler	50000	engineer	MS	NULL	-1
7	Tech_staff	oaken	clifton	10000	siftware developer	PhD	NULL	-1
8	Tech_staff	iduna	florence	500000	sales engineer	BS	NULL	-1
9	Tech_staff	duke	gila bend	900	IT manager	MS	NULL	-1
10	Tech_staff	pabble	glendale	300000	data scientist	PhD	NULL	-1
11	Tech_staff	Bulda	globe	500000	web developer	BS	NULL	-1
12	Tech_staff	agnarr	kingman	5876000	sales engineer	MS	NULL	-1
13	Tech_staff	dignitary	lake havasu city	234900	lead	PhD	NULL	-1
14	Tech_staff	gerda	mesa	123600	engineer	BS	NULL	-1
15	Tech_staff	sven	nogales	1234	manager	MS	NULL	-1
16	Tech_staff	marshmallow	oraibi	5678	web developer	PhD	NULL	-1
17								

#### Quality\_Controller:

	A	B	C	D	E	F	G	H
17								
18	qc	rapunzel	phoenix	9876	NULL	NULL	AC	-1
19	qc	flynn	prescott	5432	NULL	NULL	TV	-1
20	qc	rider	temple	1982	NULL	NULL	fridge	-1
21	qc	gothel	tucson	198300	NULL	NULL	AC	-1
22	qc	thug	winslow	18945	NULL	NULL	TV	-1
23	qc	rufian	yuma	356782	NULL	NULL	fridge	-1
24	qc	timothy	california	2000	NULL	NULL	AC	-1
25	qc	rajtur	huntington beach	4000	NULL	NULL	TV	-1
26	qc	vlad	los angeles	6000	NULL	NULL	fridge	-1
27	qc	bill	mountain view	9000	NULL	NULL	AC	-1
28	qc	mcknight	oroville	8000	NULL	NULL	TV	-1
29	qc	simba	roseville	5000	NULL	NULL	fridge	-1
30	qc	mufasa	san francisco	3000	NULL	NULL	AC	-1
31	qc	scar	colorado	73620	NULL	NULL	TV	-1
32	qc	nala	canon city	22400	NULL	NULL	fridge	-1
33	qc	timon	brighton	57900	NULL	NULL	AC	-1
34								

## Worker:

	A	B	C	D	E	F	G	H
35	Worker	zazu	claremore	1100	NULL	NULL	NULL	8
36	Worker	rafiki	edmond	1200	NULL	NULL	NULL	6
37	Worker	pumba	el reno	1300	NULL	NULL	NULL	15
38	Worker	sarabi	chickasha	1400	NULL	NULL	NULL	13
39	Worker	shenzi	alva	1500	NULL	NULL	NULL	10
40	Worker	ed	enid	1600	NULL	NULL	NULL	5
41	Worker	banzai	elk city	1700	NULL	NULL	NULL	3
42	Worker	sarafina	hobart	1800	NULL	NULL	NULL	9
43	Worker	hyena	midwest city	1900	NULL	NULL	NULL	0
44	Worker	belle	Moore	2000	NULL	NULL	NULL	2
45	Worker	adam	norman	2100	NULL	NULL	NULL	4
46	Worker	gaston	oklahoma city	2200	NULL	NULL	NULL	7
47	Worker	lefou	pauls valley	2300	NULL	NULL	NULL	11
48	Worker	cadenza	miami	2400	NULL	NULL	NULL	10
49	Worker	maurice	lawton	2500	NULL	NULL	NULL	19
50	Worker	jam	ponca city	2600	NULL	NULL	NULL	2

## Insertions for Tech\_Staff table:

### Record 1:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.
1
enter employee type
Tech_staff
enter employee name
Ela
enter employee address
ajo
enter employee salary
1000
enter position
lead
enter degree
BS
enter type of product
NULL
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

### Record 2:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v202204
1
enter employee type
Tech_staff
enter employee name
anna
enter employee address
avondale
enter employee salary
2000
enter position
manager
enter degree
MS
enter type of product
NULL
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v2
1
enter employee type
Tech_staff
enter employee name
olaf
enter employee address
bisbee
enter employee salary
3000
enter position
engineer
enter degree
PhD
enter type of product
NULL
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

### Results:

Results    Messages

	tech_name	tech_address	tech_salary	tech_position
1	anna	avondale	2000	manager
2	Ela	ajo	1000	lead
3	olaf	bisbee	3000	engineer

	tech_name	degree
1	anna	MS
2	Ela	BS
3	olaf	PhD

### Insertions for Quality\_controller table:

### Record 4:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
1
enter employee type
QC
enter employee name
rapunzel
enter employee address
phoenix
enter employee salary

9876
enter position
NULL
enter degree
NULL
enter type of product
AC
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Record 5:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_1
1
enter employee type
QC
enter employee name
flynn
enter employee address
prescott
enter employee salary
5432
enter position
NULL
enter degree
NULL
enter type of product
TV
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Record 6:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot
1
enter employee type
QC
enter employee name
rider
enter employee address
temple
enter employee salary
1982
enter position
NULL
enter degree
NULL
enter type of product
Fridge
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Record 7:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v202209
1
enter employee type
QC
enter employee name
gothel
enter employee address
tucson
enter employee salary
198300
enter position
NULL
enter degree
NULL
enter type of product
AC
enter count of products per day
-1
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Results:

	qc_name	qc_address	qc_salary	type_of_Product
1	flynn	prescott	5432	TV
2	gothel	tucson	198300	AC
3	rapunzel	phoenix	9876	AC
4	rider	temple	1982	Fridge

## Insertions for worker table:

### Record 8:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\hotspot\jre\full\win32\x86_64_17.0.4.v20210914-1400
1
enter employee type
Worker
enter employee name
zazu
enter employee address
claremore
enter employee salary
1100
enter position
NULL
enter degree
NULL
enter type of product
NULL
enter count of products per day
8
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

### Record 9:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\hotspot\jre\full\win32\x86_64_17.0.4.v20210914-1400
1
enter employee type
Worker
enter employee name
rafiki
enter employee address
edmond
enter employee salary
1200
enter position
NULL
enter degree
NULL
enter type of product
NULL
enter count of products per day
6
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Record 10:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.openjdk.h
1
enter employee type
Worker
enter employee name
pumba
enter employee address
el reno
enter employee salary
1300
enter position
NULL
enter degree
NULL
enter type of product
NULL
enter count of products per day
15
Connecting to the database...
employee record inserted Successfully. 1 rows inserted.
```

## Results:

	w_name	w_address	w_salary	count_of_Products
1	pumba	el reno	1300	15
2	rafiki	edmond	1200	6
3	zazu	claremore	1100	8

In similar way, I have inserted remaining records of employees into Tech\_Staff, Quality\_Controller, Worker tables.(from the excel screenshot).

	tech_name	tech_address	tech_salary	tech_position
1	agnarr	kingman	5876000	sales engineer
2	anna	avondale	2000	manager
3	Bulda	globe	500000	web developer
4	dignitary	lake havasu city	234900	lead
5	duke	gila bend	900	IT manager
6	Ela	ajo	1000	lead
7	gerda	mesa	123600	engineer
8	hans	chandler	50000	engineer
9	iduna	florence	500000	sales engineer
10	kristoff	casa grande	40000	lead
11	marshmallow	oraibi	5678	web developer
12	oaken	clifton	10000	software devel...
13	olaf	bisbee	3000	engineer
14	pabble	glendale	300000	data scientist
15	sven	nogales	1234	manager

	tech_name	degree
1	agnarr	MS
2	anna	MS
3	Bulda	BS
4	dignitary	PhD
5	duke	MS
6	Ela	BS
7	gerda	BS
8	hans	MS
9	iduna	BS
10	kristoff	BS
11	marshmallow	PhD
12	oaken	PhD
13	olaf	PhD
14	pabble	PhD
15	sven	MS

	qc_name	qc_address	qc_salary	type_of_Product
1	bill	mountain view	9000	AC
2	flynn	prescott	5432	TV
3	gothel	tucson	198300	AC
4	mcknight	oroville	8000	TV
5	mufasa	san francisco	3000	AC
6	nala	canon city	22400	fridge
7	rajtur	huntington beach	4000	TV
8	rapunzel	phoenix	9876	AC
9	rider	temple	1982	Fridge
10	rufian	yuma	356782	fridge
11	scar	colorado	73620	TV
12	simba	roseville	5000	fridge
13	thug	winslow	18945	TV
14	timon	brighton	57900	AC
15	timothy	california	2000	AC
16	vlad	los angeles	6000	fridge

Results    Messages

	w_name	w_address	w_salary	count_of_Products
1	adam	norman	2100	4
2	banzai	elk city	1700	3
3	belle	Moore	2000	2
4	cadenza	miami	2400	10
5	ed	enid	1600	5
6	gaston	oklahoma city	2200	7
7	hyena	midwest city	1900	0
8	jam	ponca city	2600	2
9	lefou	pauls valley	2300	11
10	maurice	lawton	2500	19
11	pumba	el reno	1300	15
12	rafiki	edmond	1200	6
13	sarabi	chickasha	1400	13
14	sarafina	hobart	1800	9
15	shenzi	alva	1500	10
16	zazu	claremore	1100	8

## 6.2. query2 testing screenshots:

For inserting values into the table, following data has been used(Excel sheet screenshot)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	prototype	product_id	prod_date	making_time	prod_per	test_per	rep_per	date_rep	size	soft_name	color	p_weight	req	
2	1	9	2021-05-01		2	gaston	simba	agnarr	2021-05-30	small	adobe	NULL	-1	1
3	1	8	2021-06-02		3	lefou	mufasa			large	cubase	NULL	-1	-1
4	1	7	2021-07-03		4	cadenza	scar	gerda	2021-07-30	medium	diablo	NULL	-1	1
5	1	6	2021-08-04		5	maurice	nala			small	excel	NULL	-1	-1
6	2	5	2021-09-05		6	jam	timon	marshmallow	2021-09-30	medium	NULL	red	-1	1
7	2	4	2021-10-06		7	ed	rufian	Ela	2021-10-30	large	NULL	red	-1	1
8	2	3	2021-11-07		8	banzai	timothy			meduim	NULL	black	-1	-1
9	2	2	2021-12-08		9	sarafina	rajtur	olaf	2021-12-30	small	NULL	white	-1	1
10	2	1	2021-01-09		1	hyena	vlad			medium	NULL	red	-1	-1
11	2	10	2021-02-10		4	belle	bill	anna	2021-02-28	small	NULL	white	-1	1
12	2	11	2021-03-11		6	adam	mcknight	kristoff	2021-03-30	large	NULL	black	-1	1
13	3	12	2022-01-01		7	rafiki	flynn			small	NULL	NULL	10	-1
14	3	13	2021-09-02		2	pumba	rider	dignitary	2021-09-30	medium	NULL	NULL	34	1
15	3	14	2022-02-01		4	sarabi	gothel	sven	2022-02-28	large	NULL	NULL	56	1
16	3	15	2022-03-04		9	shenzi	thug	agnarr	2022-03-30	medium	NULL	NULL	12	1
17														

## Record 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v2022
2
Enter the type of product
1
Enter Product id
9
Enter Produced date
2021-05-01
Enter making time
2
Enter Produced person name
gaston
Enter tested person name
simba
Enter Repaired person name
agnarr
Enter Repaired date
2021-05-30
Enter the size of product
small
Enter software Name
adobe
Enter Color of product
NULL
Enter Weight of product
-1
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.ope
2
Enter the type of product
1
Enter Product id
8
Enter Produced date
2021-06-02
Enter making time
3
Enter Produced person name
lefou
Enter tested person name
mufasa
Enter Repaired person name
NULL
Enter Repaired date
NULL
Enter the size of product
large
Enter software Name
cubase
Enter Color of product
NULL
Enter Weight of product
-1
if product repaired enter 1 or -1
-1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v2022090:
2
Enter the type of product
1
Enter Product id
7
Enter Produced date
2021-07-03
Enter making time
4
Enter Produced person name
cadenza
Enter tested person name
scar
Enter Repaired person name
gerda
Enter Repaired date
2021-07-30
Enter the size of product
medium
Enter software Name
diablo
Enter Color of product
NULL
Enter Weight of product
-1
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

### Record 4:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.jdt.openjdk.hots
2
Enter the type of product
1
Enter Product id
6
Enter Produced date
2021-08-04
Enter making time
5
Enter Produced person name
maurice
Enter tested person name
nala
Enter Repaired person name
NULL
Enter Repaired date
NULL
Enter the size of product
small
Enter software Name
excel
Enter Color of product
NULL
Enter Weight of product
-1
if product repaired enter 1 or -1
-1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 5:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspc
2
Enter the type of product
2
Enter Product id
5
Enter Produced date
2021-09-05
Enter making time
6
Enter Produced person name
jam
Enter tested person name
timon
Enter Repaired person name
marshmallow
Enter Repaired date
2021-09-30
Enter the size of product
medium
Enter software Name
NULL
Enter Color of product
red
Enter Weight of product
-1
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 6:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspc
2
Enter the type of product
2
Enter Product id
4
Enter Produced date
2021-10-06
Enter making time
7
Enter Produced person name
ed
Enter tested person name
rufian
Enter Repaired person name
Ela
Enter Repaired date
2021-10-30
Enter the size of product
large
Enter software Name
NULL
Enter Color of product
red
Enter Weight of product
-1
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 7:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17
2
Enter the type of product
2
Enter Product id
3
Enter Produced date
2021-11-07
Enter making time
8
Enter Produced person name
banzai
Enter tested person name
timothy
Enter Repaired person name
NULL
Enter Repaired date
NULL
Enter the size of product
medium
Enter software Name
NULL
Enter Color of product
black
Enter Weight of product
-1
if product repaired enter 1 or -1
-1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 8:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspo
2
Enter the type of product
3
Enter Product id
12
Enter Produced date
2022-01-01
Enter making time
7
Enter Produced person name
rafiki
Enter tested person name
flynn
Enter Repaired person name
NULL
Enter Repaired date
NULL
Enter the size of product
small
Enter software Name
NULL
Enter Color of product
NULL
Enter Weight of product
10
if product repaired enter 1 or -1
-1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 9:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32>
2
Enter the type of product
3
Enter Product id
13
Enter Produced date
2021-09-02
Enter making time
2
Enter Produced person name
pumba
Enter tested person name
rider
Enter Repaired person name
dignitary
Enter Repaired date
2021-09-30
Enter the size of product
medium
Enter software Name
NULL
Enter Color of product
NULL
Enter Weight of product
34
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Record 10:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.o
2
Enter the type of product
3
Enter Product id
14
Enter Produced date
2022-02-01
Enter making time

4
Enter Produced person name
sarabi
Enter tested person name
gothel
Enter Repaired person name
sven
Enter Repaired date
2022-02-28
Enter the size of product
large
Enter software Name
NULL
Enter Color of product
NULL
Enter Weight of product
56
if product repaired enter 1 or -1
1
Connecting to the database...
product record inserted Successfully. 1 rows inserted.
```

## Results:

### Product Table:

---

	product_ID	produced_date	making_time	produced_person	tested_person
1	3	2021-11-07	8	banzai	timothy
2	4	2021-10-06	7	ed	rufian
3	5	2021-09-05	6	jam	timon
4	6	2021-08-04	5	maurice	nala
5	7	2021-07-03	4	cadenza	scar
6	8	2021-06-02	3	lefou	mufasa
7	9	2021-05-01	2	gaston	simba
8	12	2022-01-01	7	rafiki	flynn
9	13	2021-09-02	2	pumba	rider
10	14	2022-02-01	4	sarabi	gothel

### Product1 table:

---

	product_ID	size	software_name
1	6	small	excel
2	7	medium	diablo
3	8	large	cubase
4	9	small	adobe

### Product2 table:

---

	product_ID	size	color
1	3	meduim	black
2	4	large	red
3	5	medium	red

**Product3 table:**

	product_ID	size	p_weight
1	12	small	10
2	13	medium	34
3	14	large	56

**Repair table:**

	product_ID	tech_name	date_repaired
1	4	Ela	2021-10-30
2	5	marshmallow	2021-09-30
3	7	gerda	2021-07-30
4	9	agnarr	2021-05-30
5	13	dignitary	2021-09-30
6	14	sven	2022-02-28

**Requested\_repair table:**

	product_ID	qc_name
1	4	rufian
2	5	timon
3	7	scar
4	9	simba
5	13	rider
6	14	gothel

In the similar way, I have added remaining records which are mentioned in the excel sheet screenshot.

### 6.3. query3 testing screenshots:

For inserting values into the table, following data has been used (Excel sheet screenshot)

	A	B	C	D
1	customer_name	customer_address	product_ID	
2	king	lindsey street	9	
3	kai	12th ave	8	
4	raj	beaumont drive	7	
5	avi	south jenkins	6	
6	lyle	california	5	
7	emmett	texas	4	
8	tristian	arizona	3	
9	zayd	georgia	2	
10	wells	virginia	1	
11	lakshmi	newyork	10	
12	rupa	los angeles	11	
13	devi	singapore	12	
14	janu	nepal	13	
15	harsha	china	14	
16	goutham	india	15	
17				

#### Record 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32
3
Enter customer name
king
Enter customer address
lindsey street
Enter customer purchased product id
9
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

#### Record 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-w
3
Enter customer name
kai
Enter customer address
12th ave
Enter customer purchased product id
8
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\pli
3
Enter customer name
raj
Enter customer address
beaumont drive
Enter customer purchased product id
7
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 4:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\pli
3
Enter customer name
avi
Enter customer address
south jenkins
Enter customer purchased product id
6
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 5:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\src\org\gnu\getopt\Getopt.java
3
Enter customer name
lyle
Enter customer address
california
Enter customer purchased product id
5
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 6:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\ecli
3
Enter customer name
emmett
Enter customer address
texas
Enter customer purchased product id
4
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 7:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\e
3
Enter customer name
tristian
Enter customer address
arizona
Enter customer purchased product id
3
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 8:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win3
3
Enter customer name
zayd
Enter customer address
georgia
Enter customer purchased product id
2
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 9:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse
3
Enter customer name
wells
Enter customer address
virginia
Enter customer purchased product id
1
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

### Record 10:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugin
3
Enter customer name
lakshmi
Enter customer address
newyork
Enter customer purchased product id
10
Connecting to the database...
Customer record inserted Successfully. 1 rows inserted.
```

## Results:

	customer_name	customer_address
1	avi	south jenkins
2	emmett	texas
3	kai	12th ave
4	king	lindsey street
5	lakshmi	newyork
6	lyle	california
7	raj	beaumont drive
8	tristian	arizona
9	wells	virginia
10	zayd	georgia

	Product_ID	customer_name
1	1	wells
2	2	zayd
3	3	tristian
4	4	emmett
5	5	lyle
6	6	avi
7	7	raj
8	8	kai
9	9	king
10	10	lakshmi

In the similar way, I have added remaining records which are mentioned in the excel sheet screenshot.

#### 6.4. query4 testing screenshots:

For inserting values into the table, following data has been used (Excel sheet screenshot)

	A	B	C	D	E	F
1	account_number	created_date	cost	product_ID	ptype	
2	A2001	2021-05-01	2000	9	1	
3	A2002	2021-06-02	2100	8	1	
4	A2003	2021-07-03	2200	7	1	
5	A2004	2021-08-04	4000	6	1	
6	A2005	2021-09-05	4500	5	2	
7	A2006	2021-10-06	2000	4	2	
8	A2007	2021-11-07	1500	3	2	
9	A2008	2021-12-08	700	2	2	
10	A2009	2021-01-09	4400	1	2	
11	A2010	2021-02-10	9888	10	2	
12	A2011	2021-03-11	2300	11	2	
13	A2012	2022-01-01	3500	12	3	
14	A2013	2021-09-02	4550	13	3	
15	A2014	2022-02-01	7689	14	3	
16	A2015	2022-03-04	1098	15	3	
17						

#### Record 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64
4
enter Account number
A2001
enter date created
2021-05-01
enter product cost
2000
enter product id
9
enter account type
1
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

#### Record 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64
4
enter Account number
A2002
enter date created
2021-06-02
enter product cost
2100
enter product id
8
enter account type
1
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\or
4
enter Account number
A2003
enter date created
2021-07-03
enter product cost
2200
enter product id
7
enter account type
1
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 4:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\pluc
4
enter Account number
A2005
enter date created
2021-09-05
enter product cost
4500
enter product id
5
enter account type
2
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 5:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\
4
enter Account number
A2006
enter date created
2021-10-06
enter product cost
2000
enter product id
4
enter account type
2
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 6:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.jdt.core\src\org\gnu\getopt\Getopt.java:115: error: cannot find symbol
        enter Account number
               ^
  symbol:   class Account
  location: class org.gnu.getopt.Getopt
A2007
enter date created
2021-11-07
enter product cost
1500
enter product id
3
enter account type
2
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 7:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.jdt.core\src\org\gnu\getopt\Getopt.java:115: error: cannot find symbol
        enter Account number
               ^
  symbol:   class Account
  location: class org.gnu.getopt.Getopt
A2008
enter date created
2021-12-08
enter product cost
700
enter product id
2
enter account type
2
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 8:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\plugins\org.eclipse.jdt.core\src\org\gnu\getopt\Getopt.java:115: error: cannot find symbol
        enter Account number
               ^
  symbol:   class Account
  location: class org.gnu.getopt.Getopt
A2012
enter date created
2022-01-01
enter product cost
3500
enter product id
12
enter account type
3
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 9:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.ec
4
enter Account number
A2013
enter date created
2021-09-02
enter product cost
4550
enter product id
13
enter account type
3
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Record 10:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse
4
enter Account number
A2014
enter date created
2022-02-01
enter product cost
7689
enter product id
14
enter account type
3
Connecting to the database...
account record inserted Successfully. 1 rows inserted.
```

### Results:

#### Product Table:

	account_number	created_date	cost
1	A2001	2021-05-01	2000
2	A2002	2021-06-02	2100
3	A2003	2021-07-03	2200
4	A2005	2021-09-05	4500
5	A2006	2021-10-06	2000
6	A2007	2021-11-07	1500
7	A2008	2021-12-08	700
8	A2012	2022-01-01	3500
9	A2013	2021-09-02	4550
10	A2014	2022-02-01	7689

### **Product1\_account Table:**

	product_ID	account_number
1	7	A2003
2	8	A2002
3	9	A2001

### **Product2\_account Table:**

	product_ID	account_number
1	2	A2008
2	3	A2007
3	4	A2006
4	5	A2005

### **Product3\_account Table:**

	product_ID	account_number
1	12	A2012
2	13	A2013
3	14	A2014

In the similar way, I have added remaining records which are mentioned in the excel sheet screenshot.

## 6.5. query5 testing screenshots:

### Record 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plu
5
enter Complaint id
C1000
enter complaint date
2021-05-28
enter description
not working
treatment expected
need repair
enter product id
8
enter 1 if possible to repair or 0
1
enter technical staff name
hans
enter date of repair
2021-05-30
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

### Record 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org
5
enter Complaint id
C3000
enter complaint date
2022-10-03
enter description
display not working
treatment expected
exchange
enter product id
3
enter 1 if possible to repair or 0
1
enter technical staff name
oaken
enter date of repair
2022-11-01
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plug
5
enter Complaint id
C4000
enter complaint date
2021-02-14
enter description
servicing
treatment expected
cleaning
enter product id
6
enter 1 if possible to repair or 0
1
enter technical staff name
iduna
enter date of repair
2021-02-28
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

### Results:

#### Complaint Table:

	complaint_id	complaint_date	comp_desc	treatment
1	C1000	2021-05-28	not working	need repair
2	C3000	2022-10-03	display not working	exchange
3	C4000	2021-02-14	servicing	cleaning

#### Cust\_complaint table:

	Complaint_ID	Product_ID
1	C1000	8
2	C3000	3
3	C4000	6

### **Repair table:**

	product_ID	tech_name	date_repaired
1	2	olaf	2021-12-30
2	3	oaken	2022-11-01
3	4	Ela	2021-10-30
4	5	marshmallow	2021-09-30
5	6	iduna	2021-02-28
6	7	gerda	2021-07-30
7	8	hans	2021-05-30
8	9	agnarr	2021-05-30
9	10	anna	2021-02-28
10	11	kristoff	2021-03-30
11	13	dignitary	2021-09-30
12	14	sven	2022-02-28
13	15	agnarr	2022-03-30

### **Complaint\_repair table:**

	complaint_ID	product_ID
1	C1000	8
2	C3000	3
3	C4000	6

## 6.6. query6 testing screenshots:

### Record 1:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\src\com\javatpoint\ip\Task5b.java:6
6
enter Accident number
ACC100
enter accident date
2021-05-30
enter days lost
2
enter accident product id:
9
Enter accident type(type1 or type2)
type2
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

### Record 2:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\src\com\javatpoint\ip\Task5b.java:6
6
enter Accident number
ACC110
enter accident date
2022-02-01
enter days lost
3
enter accident product id:
14
Enter accident type(type1 or type2)
type1
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

### Record 3:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse.jdt.core\src\com\javatpoint\ip\Task5b.java:6
6
enter Accident number
ACC120
enter accident date
2021-01-09
enter days lost
5
enter accident product id:
1
Enter accident type(type1 or type2)
type1
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

#### Record 4:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\pli
6
enter Accident number
ACC130
enter accident date
2021-03-30
enter days lost
9
enter accident product id:
11
Enter accident type(type1 or type2)
type1
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

#### Record 5:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse
6
enter Accident number
ACC140
enter accident date
2021-10-30
enter days lost
4
enter accident product id:
4
Enter accident type(type1 or type2)
type2
Connecting to the database...
complaint record inserted Successfully. 1 rows inserted.
```

#### Results:

##### Accident table:

	accident_number	accident_date	days_lost
1	ACC100	2021-05-30	2
2	ACC110	2022-02-01	3
3	ACC120	2021-01-09	5
4	ACC130	2021-03-30	9
5	ACC140	2021-10-30	4

### **Produced\_accident table:**

Results      Messages		
	Product_ID	Accident_number
1	1	ACC120
2	11	ACC130
3	14	ACC110

### **Repair\_accident table:**

Results      Messages		
	Product_ID	Accident_number
1	4	ACC140
2	9	ACC100

### **6.7. query7 testing screenshots:**

#### **Run 1:**

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2
7
Please enter Product id
14
Results
Produced_date | making_time
2022-02-01      | 4.0
```

#### **Run 2:**

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-
7
Please enter Product id
9
Results
Produced_date | making_time
2021-05-01      | 2.0
```

**Run 3:**

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\8
7
Please enter Product id
12
Results
Produced_date | making_time
2022-01-01      | 7.0
```

**6.8. query8 testing screenshots:****Run 1:**

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\8
8
enter produced person name
belle
product_ID
10
```

**Run 2:**

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\8
8
enter produced person name
jam
product_ID
5
```

**Run 3:**

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\8
8
enter produced person name
adam
product_ID
11
```

## 6.9. query9 testing screenshots:

### Run 1:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\9  
enter tested person name  
mufasa  
total errors  
1
```

### Run 2:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\9  
enter tested person name  
timothy  
total errors  
1
```

### Run 3:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\9  
enter tested person name  
nala  
total errors  
1
```

## 6.10. query10 testing screenshots:

### Run 1:

```
Console × Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mot...  
10  
enter tested person name  
rider  
total cost  
4550.0
```

### Run 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\
10
enter tested person name
gothel
total cost
7689.0
```

### Run 3:

```
10
enter tested person name
thug
total cost
1098.0
```

## 6.11. query11 testing screenshots:

### Run 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\
11
enter color
red
Customer names
emmett
lyle
wells
```

### Run 2:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\clipse
11
enter color
black
Customer names
rupa
tristian
```

### Run 3:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads
11
enter color
white
Customer names
lakshmi
zayd
```

### 6.12. query12 testing screenshots:

#### Run 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\e
12
enter salary
30000
Employee name           | salary          |      address
agnarr                 | 5876000.0       |      kingman
Bulda                  | 500000.0        |      globe
dignitary               | 234900.0        |      lake havasu city
gerda                  | 123600.0        |      mesa
gothel                 | 198300.0        |      tucson
hans                   | 50000.0          |      chandler
iduna                  | 500000.0        |      florence
kristoff                | 40000.0          |      casa grande
pabble                  | 300000.0         |      glendale
rufian                 | 356782.0         |      yuma
scar                   | 73620.0          |      colorado
timon                  | 57900.0          |      brighton
```

### 6.13. query13 testing screenshots:

#### Run 1:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mo
13
total days lost
0
```

#### 6.14. query14 testing screenshots:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-
14
enter year
2022
average cost
4095.6667
```

#### 6.15. query15 testing screenshots:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-20
15
enter date1
2021-05-30
enter date2
2022-02-01
deleted successfully
```

#### Results:

	accident_number	accident_date	days_lost
1	ACC120	2021-01-09	5
2	ACC130	2021-03-30	9

	Product_ID	Accident_number
1	1	ACC120

	Product_ID	Accident_number
1	11	ACC130

## 6.16. query16 testing screenshots:

The screenshot shows an Excel spreadsheet titled "test.csv - Excel". The data is organized into columns A through K. Row 1 contains column headers: "Tech\_staff", "Ela", "ajo", "1000", "lead", "BS", "NULL", "-1", "", "", "", "" (empty cells I, J, K). Row 2 contains: "QC", "rapunzel", "phoenix", "9876", "NULL", "NULL", "AC", "-1", "", "", "", "" (empty cells I, J, K). Row 3 contains: "Worker", "zazu", "claremore", "1100", "NULL", "NULL", "NULL", "8", "", "", "", "" (empty cells I, J, K). Row 4 is empty.

	A	B	C	D	E	F	G	H	I	J	K
1	Tech_staff	Ela	ajo	1000	lead	BS	NULL	-1			
2	QC	rapunzel	phoenix	9876	NULL	NULL	AC	-1			
3	Worker	zazu	claremore	1100	NULL	NULL	NULL	8			
4											

```
17) Export: Retrieve all customers (in name order) who pur  
and output them to a data file instead of screen (the user must e  
18) Exit (Option 18)  
16  
enter the file name for import  
D:\\\\mothukuri\\\\test.csv  
inserted successfully  
inserted successfully  
inserted successfully
```

The screenshot shows the SSMS interface with two result sets. The top result set is titled "Results" and displays data from the "tech" table. The bottom result set is titled "Messages" and displays data from the "degree" table.

**Results**

	tech_name	tech_address	tech_salary	tech_position
1	Ela	ajo	1000	lead

**Messages**

	tech_name	degree
1	Ela	BS

Results    Messages

	qc_name	qc_address	qc_salary	type_of_Product
1	rapunzel	phoenix	9876	AC

Results    Messages

	w_name	w_address	w_salary	count_of_Products
1	zazu	claremore	1100	8

### 6.16. query17 testing screenshots:

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\mothu\Downloads\eclipse-jee-2022-09-R-win32-x86_64\eclipse\plugins\org.eclipse
17
Enter the path with file name
D:\\export_file.csv
Enter the color of product
red
Exported Successfully
Exported Successfully
Exported Successfully
```

#### Results:

	A	B	C
1	emmett		
2	lyle		
3	wells		
4			

### 6.17. Handling Errors:

Error 1: Trying to add duplicate value of Account number into the account table.

```
Console X
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\m
4
enter Account number
A2001
enter date created
2021-05-01
enter product cost
2000
enter product id
9
enter account type
1
Connecting to the database...
you got an error,exception in option 4
```

### Error 2: Entered degree as btech other than 'BS', 'MS', 'PhD'.

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\r
1
enter employee type
Tech_staff
enter employee name
kovida
enter employee address
khammam
enter employee salary
300000
enter position
lead
enter degree
Btech
enter type of product
NULL
enter count of products per day
-1
Connecting to the database...
you got an error,exception in option 1
```

### Error 3: Foreign key error handling in product table:

```
Console ×
Mothukuri_Kovida_IP_Task5b [Java Application] C:\Users\m
2
Enter the type of product
1
Enter Product id
30
Enter Produced date
2021-07-28
Enter making time
5
Enter Produced person name
anitha
Enter tested person name
ravali
Enter Repaired person name
NULL
Enter Repaired date
NULL
Enter the size of product
small
Enter software Name
word
Enter Color of product
NULL
Enter Weight of product
NULL
you got an error,exception in option 2:
```

### 6.18. query18 testing screenshots:

```
18
You choose to Quit! bye
```

## Task 7. Web based application

### 7.1. Source code for a web application with pictures demonstrating its successful compilation

```
1 Mothukuri_Kovida_IP_Task7_DataHandler.java ×
2
3 package Mothukuri;
4
5 import java.sql.Connection;
6 import java.sql.ResultSet;
7 import java.sql.Statement;
8 import java.sql.DriverManager;
9 import java.sql.PreparedStatement;
10 import java.sql.SQLException;
11
12 public class Mothukuri_Kovida_IP_Task7_DataHandler {
13     private Connection conn;
14
15     private String server = "moth0000-sql-server.database.windows.net";
16     private String database = "cs-dsa-4513-sql-db";
17     private String username = "moth0000";
18     private String password = "Srinuani@9398";
19
20     final private String url =
21         String.format("jdbc:sqlserver:// %s:1433;database=%s;user=%s;password=%s;encrypt=true;" +
22                     "trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
23                     server, database, username, password);
24
25     public void getDBConnection() throws SQLException {
26         if (conn != null) {
27             return;
28         }
29
30         this.conn = DriverManager.getConnection(url);
31     }
32
33     public ResultSet FetchEmployees() throws SQLException {
34         getDBConnection();
35
36         final String option1 = "select tech_name as name ,tech_salary as salary ,tech_address as address from Tech_Staff UNION\r\n" +
37             "+ \"SELECT qc_name,qc_salary, qc_address from Quality_Controller UNION\r\n\" +
38             "+ \"select w_name,w_salary,w_address from Worker;\";
39         final PreparedStatement s = conn.prepareStatement(option1);
40         return s.executeQuery();
41     }
42
43     public ResultSet EmpSalary(
44         float Employee_salary) throws SQLException {
45
46         getDBConnection();
47         final String option12 = "EXEC q12 @salary = "+ Employee_salary + ";";
48         Statement s = conn.createStatement();
49         return s.executeQuery(option12);
50     }
51
52     public boolean addEmployee(
53         String emptytype, String empname, String empaddress, float empsalary, String tech_position,
54         String degree, String type_of_product, int count_of_products) throws SQLException {
55
56         getDBConnection(); // Prepare the database connection
57         final String option1 = "EXEC q1 @emptytype = '"+emptytype+"', @ename = '"+empname+"', "
58             + "@address = '"+empaddress+"', @salary = '"+empsalary+"', @tech_position = '"+tech_position+"', "
59             + " @degree = '"+degree+"', @type_of_product = '"+type_of_product+"', @count_of_Products = '" +count_of_products+"'";
60
61         final PreparedStatement s = conn.prepareStatement(option1);
62         return s.executeUpdate() == 1;
63     }
64 }
```

```

Motheukuri_Kovida_IP_Task7_add_employee_form.jsp X
1 <!DOCTYPE html>
2<html>
3    <head>
4        <meta charset="UTF-8">
5        <title>Add employee form</title>
6    </head>
7    <body>
8        <h2>Add Employee data in the fields below</h2>
9
10       <form action="Motheukuri_Kovida_IP_Task7_add_employee.jsp">
11
12           <table border=1>
13               <tr>
14                   <th colspan="2">Enter the employee Data:</th>
15               </tr>
16               <tr>
17                   <td>Employee type:</td>
18                   <td><div style="text-align: center;">
19                       <input type=text name=emptytype>
20                   </div></td>
21               </tr>
22               <tr>
23                   <td>Employee name:</td>
24                   <td><div style="text-align: center;">
25                       <input type=text name=empname>
26                   </div></td>
27               </tr>
28               <tr>
29                   <td>Employee address:</td>
30                   <td><div style="text-align: center;">
31                       <input type=text name=empaddress>
32                   </div></td>
33               </tr>
34               <tr>
35                   <td>salary:</td>
36                   <td><div style="text-align: center;">
37                       <input type=text name=empsalary>
38                   </div></td>
39               </tr>
40
41               <tr>
42                   <td>technical position:</td>
43                   <td><div style="text-align: center;">
44                       <input type=text name=tech_position>
45                   </div></td>
46               </tr>
47               <tr>
48                   <td>degree:</td>
49                   <td><div style="text-align: center;">
50                       <input type=text name=degree>
51                   </div></td>
52               </tr>
53               <tr>
54                   <td>product type:</td>
55                   <td><div style="text-align: center;">
56                       <input type=text name=type_of_product>
57                   </div></td>
58               </tr>
59               <tr>
60                   <td>max count of products:</td>
61                   <td><div style="text-align: center;">
62                       <input type=text name=count_of_products>
63                   </div></td>
64               </tr>
65               <tr>
66                   <td><div style="text-align: center;">
67                       <input type=reset value=Clear>
68                   </div></td>
69                   <td><div style="text-align: center;">
70                       <input type=submit value=Insert>
71                   </div></td>
72           </tr>
73       </table>
74   </form>
75 </body>
76 </html>

```

```

Mothukuri_Kovida_IP_Task7_add_employee.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2 pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4 "http://www.w3.org/TR/html4/loose.dtd">
5<html>
6<head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8 <title>Query Result</title>
9 </head>
10<body>
11 <%@page import="Mothukuri.Kovida_IP_Task7_DataHandler"%>
12 <%@page import="java.sql.ResultSet"%>
13 <%@page import="java.sql.Array"%>
14<%
15 Mothukuri_Kovida_IP_Task7_DataHandler handler = new Mothukuri_Kovida_IP_Task7_DataHandler();
16
17 String emptytype,empname,empaddress,tech_position,degree,type_of_product;
18 float empsalary;
19 int count_of_products;
20 emptytype = request.getParameter("emptytype");
21 empname = request.getParameter("empname");
22 empaddress = request.getParameter("empaddress");
23 empsalary = Float.parseFloat(request.getParameter("empsalary"));
24 tech_position = request.getParameter("tech_position");
25 degree = request.getParameter("degree");
26 type_of_product = request.getParameter("type_of_product");
27 count_of_products = Integer.parseInt(request.getParameter("count_of_products"));
28
29 if (empname.equals("") || empsalary == 0 || empaddress.equals("")) {
30     response.sendRedirect("Mothukuri_Kovida_IP_Task7_add_employee_form.jsp");
31 } else {
32
33     boolean success = handler.addEmployee(emptytype, empname, empaddress, empsalary, tech_position, degree, type_of_product, count_of_products);
34
35     if (!success) {
36         %>
37         <h2>There was a problem inserting the Employee data</h2>
38     } else {
39         %>
40         <h2>Inserted Employee data:</h2>
41         <ul>
42             <li>Employee type: <%=emptytype%></li>
43             <li>Employee name: <%=empname%></li>
44             <li>Employee address: <%=empaddress%></li>
45             <li>Employee salary: <%=empsalary%></li>
46             <li>tech position: <%=tech_position%></li>
47             <li>degree: <%=degree%></li>
48             <li>product type: <%=type_of_product%></li>
49             <li>number of products produced: <%=count_of_products%></li>
50         </ul>
51
52         <h2>Was successfully inserted.</h2>
53
54         <a href="Mothukuri_Kovida_IP_Task7_get_all_employee.jsp">See all employee data.</a>
55     }
56 }
57 }
58 %>
59
60 </body>
61 </html>

```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4<html>
5   <head>
6     <meta charset="UTF-8">
7     <title>Employee data</title>
8   </head>
9   <body>
10    <%@page import="Mothukuri.Kovida_IP_Task7_DataHandler"%>
11    <%@page import="java.sql.ResultSet"%>
12    <%
13      final Mothukuri_Kovida_IP_Task7_DataHandler handler = new Mothukuri_Kovida_IP_Task7_DataHandler();
14      final ResultSet getEmployee = handler.FetchEmployees();
15    %>
16    <table cellspacing="2" cellpadding="2" border="1">
17      <tr> <!-- The table headers row -->
18        <td align="center">
19          <h4>employee name</h4>
20        </td>
21        <td align="center">
22          <h4>employee address</h4>
23        </td>
24        <td align="center">
25          <h4>employee salary</h4>
26        </td>
27      </tr>
28    <%
29      while(getEmployee.next()) {
30        final String name = getEmployee.getString("name");
31        final String address = getEmployee.getString("address");
32        final String salary = getEmployee.getString("salary");
33        out.println("<tr>");
34        out.println(
35          "  <td align=\"center\">" + name +
36          "  </td><td align=\"center\"> " + address +
37          "  </td><td align=\"center\"> " + salary + "</td>");
38        out.println("</tr>");
39      }
40    %>
41  </table>
42 </body>
43 </html>

```

```

1 <!DOCTYPE html>
2<html>
3   <head>
4     <meta charset="UTF-8">
5     <title>employee data</title>
6   </head>
7   <body>
8     <h2>employee data </h2>
9     <form action="Mothukuri_Kovida_IP_Task7_salary.jsp">
10       <!-- The form organized in an HTML table for better clarity. -->
11       <table border=1>
12         <tr>
13           <th colspan="2">Enter salary:</th>
14         </tr>
15         <tr>
16           <td>salary</td>
17           <td><div style="text-align: center;">
18             <input type="text" name="Employee_salary">
19           </div></td>
20         </tr>
21         <tr>
22           <td><div style="text-align: center;">
23             <input type=reset value=Clear>
24           </div></td>
25           <td><div style="text-align: center;">
26             <input type=submit value=Go>
27           </div></td>
28         </tr>
29       </table>
30     </form>
31   </body>
32 </html>
33

```

```

Mothukuri_Kovida_IP_Task7_salary.jsp ×
4@<html>
5@    <head>
6@      <meta charset="UTF-8">
7@      <title>Employee data</title>
8@    </head>
9@    <body>
10@      <%@page import="Mothukuri.Kovida_IP_Task7_DataHandler"%>
11@      <%@page import="java.sql.ResultSet"%>
12@      <%
13@          float Employee_salary = Float.parseFloat(request.getParameter("Employee_salary"));
14@          Mothukuri_Kovida_IP_Task7_DataHandler handler =new Mothukuri_Kovida_IP_Task7_DataHandler();
15@          final ResultSet getsalary = handler.EmpSalary(Employee_salary);
16@          %>
17@      <table cellspacing="2" cellpadding="2" border="1">
18@          <tr> <!-- The table headers row -->
19@              <td align="center">
20@                  <h4>Employee name</h4>
21@              </td>
22@              <td align="center">
23@                  <h4>Employee address</h4>
24@              </td>
25@              <td align="center">
26@                  <h4>Employee salary</h4>
27@              </td>
28@          </tr>
29@          <%
30@          while(getsalary.next())
31@          {
32@              final String empname = getsalary.getString("employee_name");
33@              final String empaddress = getsalary.getString("address");
34@              final String empsalary = getsalary.getString("salary");
35@              out.println("<tr>");
36@              out.println(
37@                  "<td align=\"center\">" + empname +
38@                  "</td><td align=\"center\"> " + empaddress +
39@                  "</td><td align=\"center\"> " + empsalary + "</td>");
40@              out.println("</tr>");
41@          }
42@          %>
43@      </table>
44@    </body>
45@ </html>

```

## 7.2. Screenshots demonstrating the Web database application being

For q12:

### employee data

<b>Enter salary:</b>	
salary	35000
<input type="button" value="Clear"/>	<input type="button" value="Go"/>

Employee name	Employee address	Employee salary
agnarr	kingman	5876000.0
Bulda	globe	500000.0
dignitary	lake havasu city	234900.0
gerda	mesa	123600.0
gothel	tucson	198300.0
hans	chandler	50000.0
iduna	florence	500000.0
kristoff	casa grande	40000.0
pabble	glendale	300000.0
rufian	yuma	356782.0
scar	colorado	73620.0
timon	brighton	57900.0

For q1:



## Add Employee data in the fields below

Enter the employee Data:	
Employee type:	tech_staff
Employee name:	srinivas
Employee address:	hyderabad
salary:	45000
technical position:	lead
degree:	phd
product type:	null
max count of products:	-1
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

## Inserted Employee data:

- Employee type: tech\_staff
- Employee name: srinivas
- Employee address: hyderabad
- Employee salary: 45000.0
- tech position: lead
- degree: phd
- product type: null
- number of products produced: -1

**Was successfully inserted.**

[See all employee data.](#)

employee name	employee address	employee salary
adam	norman	2100.0
agnarr	kingman	5876000.0
anna	avondale	2000.0
banzai	elk city	1700.0
belle	Moore	2000.0
bill	mountain view	9000.0
Bulda	globe	500000.0
cadenza	miami	2400.0
dignitary	lake havasu city	234900.0
duke	gila bend	900.0
ed	enid	1600.0
Ela	ajo	1000.0
flynn	prescott	5432.0
gaston	oklahoma city	2200.0
gerda	mesa	123600.0
gothel	tucson	198300.0
hans	chandler	50000.0
hyena	midwest city	1900.0
iduna	florence	500000.0
jam	ponce city	2600.0
kristoff	casa grande	40000.0
lefou	pauls valley	2300.0
marshmallow	oraibi	5678.0
maurice	lawton	2500.0
meknight	oroville	8000.0

mufasa	san francisco	3000.0
nala	canon city	22400.0
oaken	clifton	10000.0
olaf	bisbee	3000.0
pabble	glendale	300000.0
pumba	el reno	1300.0
rafiki	edmond	1200.0
rajtur	huntington beach	4000.0
rapunzel	phoenix	9876.0
rider	temple	1982.0
rufian	yuma	356782.0
sarabi	chickasha	1400.0
sarafina	hobart	1800.0
scar	colorado	73620.0
shenzi	alva	1500.0
simba	roseville	5000.0
srinivas	hyderabad	45000.0
sven	nogales	1234.0
thug	winslow	18945.0
timon	brighton	57900.0
timothy	california	2000.0
vlad	los angeles	6000.0
zazu	claremore	1100.0

For q12:



## employee data

Enter salary:	
salary	35000
Clear	Go

Employee name	Employee address	Employee salary
agnarr	kingman	5876000.0
Bulda	globe	500000.0
dignitary	lake havasu city	234900.0
gerda	mesa	123600.0
gothel	tucson	198300.0
hans	chandler	50000.0
iduna	florence	500000.0
kristoff	casa grande	40000.0
pabble	glendale	300000.0
rufian	yuma	356782.0
scar	colorado	73620.0
srinivas	hyderabad	45000.0
timon	brighton	57900.0