

- Hunter DeVoe, 113510028, hunter.devoe@ou.edu
  - Assignment: Individual Project
  - Course: CS/DSA 4513, Section 001
  - Semester and Year: Fall 2022
  - Instructor: Dr. Le Gruenwald
  - SCORE:

# Table of Contents

---

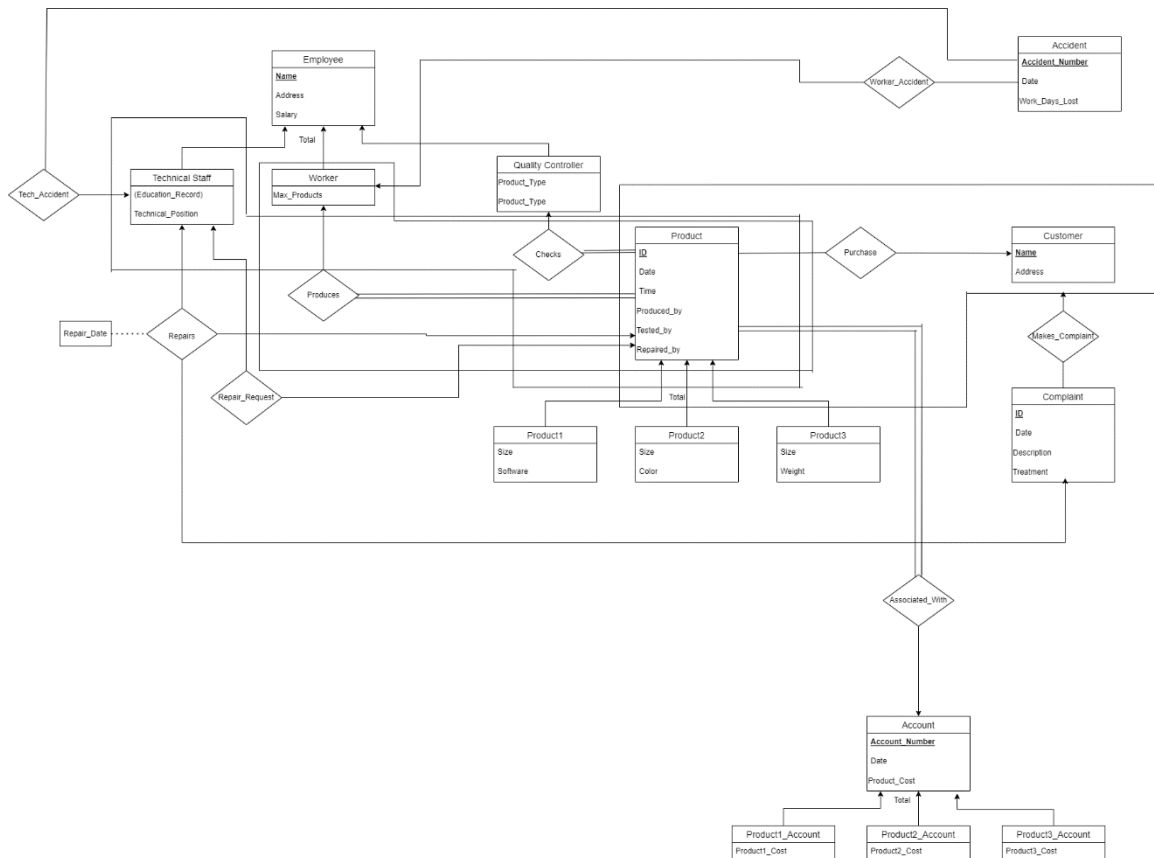
Task 1.....	4
1.1: ER Diagram .....	4
1.2: Relational Database Schema.....	5
Task 2: Schema Diagram .....	6
Task 3.....	8
3.1: Discussion of storage structures for tables.....	8
3.2: Discussion of storage structures for tables (Azure SQL Database) .....	12
Task 4: SQL statements and screenshots showing the creation of tables in Azure SQL Database .....	13
Task 5.....	23
5.1: SQL statements and Transact SQL stored procedures.....	23
5.2: The Java source program and screenshots showing it's successful .....	34
Task 6: Java Program Execution.....	65
6.1: Query 1 .....	65
6.2: Query 2 .....	77
6.3: Query 3 .....	88
6.4: Query 4 .....	99
6.5: Query 5 .....	106
6.6: Query 6 .....	109
6.7: Query 7 .....	112
6.8: Query 8 .....	115
6.9: Query 9 .....	118
6.10: Query 10 .....	121
6.11: Query 11 .....	124
6.12: Query 12 .....	127
6.13: Query 13 .....	128
6.14: Query 14 .....	129
6.15: Query 15 .....	130
6.16: Query 16 .....	131
6.17: Query 17 .....	133

6.18: Query 18 .....	134
6.19: Error Handling .....	135
Task 7.....	138
7.1 Source Code and Compilation .....	138
7.2: Testing Web-Based Application .....	147

# Task 1

## 1.1: ER Diagram

The first task of the Individual Project was to create an ER Diagram based on the given information in the Description section of the project instructions file. Using this information, we were to create multiple entity – relation sets between entities, which would help in the design of the actual database program later. Below is my ER Diagram design that I obtained based on the information in the Individual Project:



Though it is hard to tell, I have three aggregations in the diagram: One around Technical Staff and Product, one around Worker and Product, and one around Product and Customer.

## 1.2: Relational Database Schema

After creating the ER Diagram, we must convert it into a Relational Database Schema. We will build most of our SQL tables from this diagram, as it converts each entity and each relationship into its own schema. Here is my resulting Relational Database Schema based on my ER Diagram:

Employee(Name, Address, Salary)

Technical\_Staff(Employee\_Name, Address, Salary, Technical\_Position)

Technical\_Staff(Employee\_Name, Degree)

Worker(Employee\_Name, Address, Salary, Max\_Products)

Quality\_Controller(Employee\_Name, Address, Salary, Product\_Type)

Product(ID, Date, Days\_Developed, Produced\_By, Tested\_By, Repaired\_By)

Product1(ID, Date, Days\_Developed, Produced\_By, Tested\_By, Repaired\_By, Size, Software)

Product2(ID, Date, Days\_Developed, Produced\_By, Tested\_By, Repaired\_By, Size, Color)

Product3(ID, Date, Days\_Developed, Produced\_By, Tested\_By, Repaired\_By, Size, Weight)

Checks(Employee\_Name, Product\_ID)

Produces(Employee\_Name, Product\_ID)

Repair\_Request(ID, Technical\_Staff\_Name, Quality\_Controller\_Name, Product\_ID)

Repair\_Complaint(Complaint\_ID, Technical\_Staff\_Name, Date\_Repaired, Product\_ID)

Repairs(Technical\_Staff\_Name, Complaint\_ID, Date\_Repaired, Product\_ID)

Customer(Name, Address)

Purchase(Customer\_Name, Product\_ID)

Complaint(Complaint\_ID, Date, Description, Treatment, Customer\_Name, Product\_ID)

Makes\_Complaint(Customer\_Name, Complaint\_ID, Product\_ID)

Accident(Accident\_Number, Date, Work\_Days\_Lost, Product\_ID, Employee\_Name, Employee\_Type)

Tech\_Accident(Accident\_Number, Employee\_Name, Product\_ID)

Worker\_Accident(Accident\_Number, Employee\_Name, Product\_ID)

Product1\_Account(Account\_Number, Date, Product\_ID, Product\_Cost)

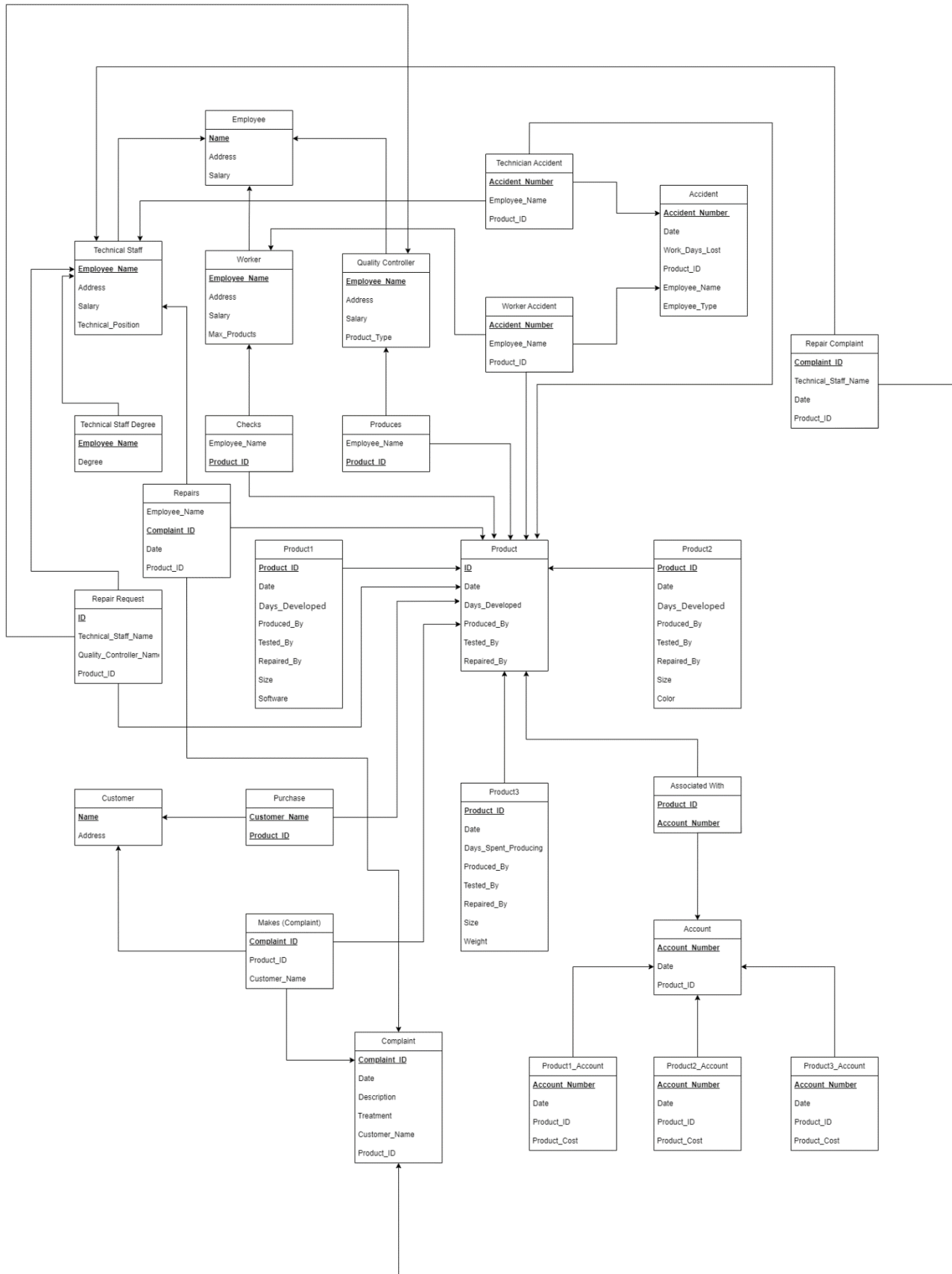
Product2\_Account(Account\_Number, Date, Product\_ID, Product\_Cost)

Product3\_Account(Account\_Number, Date, Product\_ID, Product\_Cost)

Associated\_With(Product\_ID, Account\_Number)

## Task 2: Schema Diagram

Now that we have a Relational Database Schema, we can create a Schema Diagram that shows all of the foreign key dependencies for each entity and relationship. We can model the Schema Diagram from the Relational Database Schema and easily identify which tables will have foreign key constraints, and where these foreign keys will be coming from.



## Task 3

### 3.1: Discussion of storage structures for tables

Table Name	Query Number and Type	Search key	Query Frequency	File organization	Reason
Employee	1, 16, Insertion	Employee_Name	2/month	Heap	Insertion is fast
Technical_Staff	1, 16, Insertion		2/month	Heap	Insertion is fast
	12, Range Search	Salary	1/month	Heap	Few numbers of queries
Technical_Staff_Degree	1, 16, Insertion	Name	2/month	Heap	Few numbers of queries
Quality_Controller	1, 16, Insertion		2/ month	Heap	Insertion is fast
	12, Range Search	Salary	1/month	Heap	Few numbers



					of queries
Worker	1, 16, Insertion		2 /month	Heap	Insertion is fast
	12, Range Search	Salary	1/ month	Heap	Few numbers of queries
Product	2, Insertion		400/day	Primary sequential index on ID	Execution could be faster.
	7, Random Search	ID	100/day	Primary sequential index on ID	Products are sorted on product ID
	8, Random Search	Produced_By	2000/day	Secondary index on Produced_By	A large number of queries per day

Customer	3, Insertion		50/day	Heap	Insertion is fairly easy
	11, Random Search	Name	5/month	Primary index on Name	Speed up execution by sorting on Name
Complaint	5, Insertion		30/day	Heap	Insertion is fast
Accident	15, Delete, Range Search	Date_Created	1/day	Secondary index on Date_Created	Speed up execution by sorting on Name
	6, Insertion		1/week	Heap	Insertion is fast
Product1	2, Insertion			Heap	Insertion is fast
Product2	11, Random	Color	5/month	Secondary index on Color	Speed up execution by

					sorting on Color
	2, Insertion		400/day	Heap	Insertion is fast
Product3	2, Insertion		400/day	Heap	Insertion is fast
Account	4, Insertion		40/day	Heap	Insertion is fast
	2, Insertion		400/day	Heap	Insertion is fast
Product1_Account	4, Insertion		40/day	Heap	Insertion is fast
	2, Insertion		400/day	Heap	Insertion is fast
Product2_Account	4, Insertion		40/day	Heap	Insertion is fast
	2, Insertion		400/day	Heap	Insertion is fast
Product3_Account	4, Insertion		40/day	Heap	Insertion is fast
	2, Insertion		400/day	Heap	Insertion is fast

Repair_Complaint	5, Insertion		30/day	Heap	Insertion is fast
Repair_Request	10, Random Search	Quality_Controller_Name	40/day	Sequential index file on Quality_Controller_Name	Random search efficient
Purchase	3, Insertion		50/day	Heap	Insertion is fast

### 3.2: Discussion of storage structures for tables (Azure SQL Database)

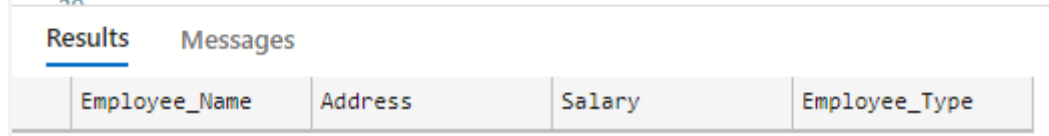
When researching storage structures for tables in Azure SQL, I came to realize that Azure was not very flexible with their options and had limited storage options to choose from. To begin with the most basic structure, Heaps in Azure SQL exist and are implemented on tables that do not have a clustered index. Any number of non-clustered indexes can be created on a table, and they will be stored as a Heap. Heaps are generally useful when doing insertions on a table, or when you do not have a plan on how to use an already existing table yet. Next, we have clustered indexes. Each table with a primary key already has a clustered index created on that primary key, and in many cases, that is all you will need for the table. If you do need more indexes though, you can create secondary indexes based on other attributes and implement them as well. As you will see later, I do this a few times. In Azure, clustered indexes are implemented as B+ Trees, which is what gives them their speedy traversal times. Finally, Azure does not support dynamic hashing, so we cannot use any form of hashing to sort our tables. For my tables in Azure, each table has a clustered index on its primary key, implementing a B+ Tree structure on the index. I also implemented a few secondary indexes on some of the tables too. First, I implement a secondary index on Produced\_By in my Product table since we do 2000 queries per day where

we look for the Product\_By column values. Next, I create a secondary index on Date\_Created in my Accident table. We only search for this once a day, but I then create a secondary index on Color in my Product2 table. Lastly, I create a sequential index file on Quality\_Controller\_Name in my Repair\_Request table. For the insertions, I let the tables be Heaps since insertions are quick on Heaps.

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

This section will show photos of all of my tables being created successfully, along with the creation of any secondary indexes.

```
21 CREATE TABLE Employee (  
22     Employee_Name varchar(20) PRIMARY KEY,  
23     Address varchar(50),  
24     Salary REAL,  
25     Employee_Type varchar(20)  
26 );  
27  
28 SELECT * FROM Employee
```



Employee_Name	Address	Salary	Employee_Type
---------------	---------	--------	---------------

- Creation of the Employee table

```

28 CREATE TABLE Technical_Staff (
29     Technical_Staff_Name varchar(20) PRIMARY KEY,
30     Position varchar(20),
31     FOREIGN KEY (Technical_Staff_Name) REFERENCES Employee(Employee_Name)
32 );
33
34 SELECT * FROM Technical_Staff
35

```

Results Messages

Technical_Staff_Name	Position
----------------------	----------

- Creation of the Technical\_Staff table

```

34 CREATE TABLE Technical_Staff_Degree (
35     Name varchar(20),
36     Degree varchar(5),
37     CONSTRAINT chk_Degree_Type CHECK(Degree = 'BS' OR Degree = 'MS' OR Degree = 'PHD'),
38     PRIMARY KEY (Name, Degree),
39     FOREIGN KEY (Name) REFERENCES Technical_Staff(Technical_Staff_Name)
40 );
41
42 SELECT * FROM Technical_Staff_Degree

```

Results Messages

Name	Degree
------	--------

- Creation of the Technical\_Staff\_Degree table

```

42 CREATE TABLE Worker (
43     Worker_Name varchar(20) PRIMARY KEY,
44     Max_Products INT,
45     FOREIGN KEY (Worker_Name) REFERENCES Employee(Employee_Name)
46 );
47
48 SELECT * FROM Worker

```

Results Messages

Worker_Name	Max_Products
-------------	--------------

- Creation of the Worker table

```

48 CREATE TABLE Quality_Controller (
49     Quality_Controller_Name varchar(20) PRIMARY KEY,
50     Product_Type varchar(20),
51     FOREIGN KEY (Quality_Controller_Name) REFERENCES Employee(Employee_Name)
52 );
53
54 SELECT * FROM Quality_Controller

```

Results Messages

Quality_Control...	Product_Type
--------------------	--------------

- Creation of the Quality Controller table.

```

54 CREATE TABLE Product (
55     ID INT PRIMARY KEY,
56     Date_Created Date,
57     Days_Developed INT,
58     Produced_By varchar(20),
59     Tested_By varchar(20),
60     Repaired_By varchar(20),
61     Size varchar(20),
62     Product_Type INT,
63
64     CONSTRAINT chk_Product_Size CHECK(Size = 'Small' OR Size = 'Medium' OR Size = 'Large'),
65     FOREIGN KEY (Produced_By) REFERENCES Worker(Worker_Name),
66     FOREIGN KEY (Tested_By) REFERENCES Quality_Controller(Quality_Controller_Name),
67     FOREIGN KEY (Repaired_By) REFERENCES Technical_Staff(Technical_Staff_Name)
68 );
69
70 -- Product secondary index on worker name
71 CREATE INDEX Product_Worker ON Product(Produced_By)
72
73 SELECT * FROM Product

```

Results Messages

ID	Date_Created	Days_Developed	Produced_By	Tested_By	Repaired_By	Size	Product_Type
----	--------------	----------------	-------------	-----------	-------------	------	--------------

- Creation of the Product table, as well as the index on Produced\_By

```

73 CREATE TABLE Product1 (
74     Product1_ID INT PRIMARY KEY,
75     Software varchar(30),
76     FOREIGN KEY (Product1_ID) REFERENCES Product(ID)
77 );
78
79 SELECT * FROM Product1

```

Results Messages

Product1_ID	Software
-------------	----------

- Creation of the Product1 table



```

79 CREATE TABLE Product2 (
80     Product2_ID INT PRIMARY KEY,
81     Color varchar(10),
82     FOREIGN KEY (Product2_ID) REFERENCES Product(ID)
83 );
84
85 SELECT * FROM Product2
86
87 -- Index on Color for Product2
88 CREATE INDEX Product2_Color ON Product2(Color)
89

```

Results    Messages

Product2_ID	Color
-------------	-------

- Creation of the Product2 table, as well as the index on Color

```

88 CREATE TABLE Product3 (
89     Product3_ID INT PRIMARY KEY,
90     Weight varchar(10),
91     FOREIGN KEY (Product3_ID) REFERENCES Product(ID)
92 );
93
94 SELECT * FROM Product3

```

Results    Messages

Product3_ID	Weight
-------------	--------

- Creation of the Product3 table

```

94 CREATE TABLE Customer (
95     Name varchar(20) PRIMARY KEY,
96     Address varchar(50)
97 );
98
99 SELECT * FROM Customer

```

**Results**    Messages

Name	Address
------	---------

- Creation of the Customer table

```

99 CREATE TABLE Complaint (
100     Complaint_ID INT PRIMARY KEY,
101     Date_Created DATE,
102     Description varchar(250),
103     Treatment varchar(20),
104     Customer_Name varchar(20),
105     Product_ID INT,
106     FOREIGN KEY (Customer_Name) REFERENCES Customer(Name),
107     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
108 );
109
110 SELECT * FROM Complaint
111

```

**Results**    Messages

Complaint_ID	Date_Created	Description	Treatment	Customer_Name	Product_ID
--------------	--------------	-------------	-----------	---------------	------------

- Creation of the Complaint table

```

110 CREATE TABLE Purchase (
111     Customer_Name varchar(20),
112     Product_ID INT,
113     PRIMARY KEY (Customer_Name, Product_ID),
114     FOREIGN KEY (Customer_Name) REFERENCES Customer(Name),
115     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
116 );
117
118 SELECT * FROM Purchase

```

Results Messages

Customer_Name	Product_ID
---------------	------------

- Creation of the Purchase table

```

118 CREATE TABLE Accident (
119     Accident_Number INT PRIMARY KEY,
120     Date_Created DATE,
121     Work_Days_Lost INT,
122     Product_ID INT,
123     Employee_Name varchar(20),
124     Employee_Type varchar(20),
125     CONSTRAINT chk_Employee_Type CHECK(Employee_Type != 'Quality Controller'),
126     FOREIGN KEY (Product_ID) REFERENCES PRODUCT(ID),
127     FOREIGN KEY (Employee_Name) REFERENCES Employee(Employee_Name)
128 );
129
130 -- Index on Date for Accident
131 CREATE INDEX Accident_Date ON Accident(Date_Created)
132
133 SELECT * FROM Accident

```

Results Messages

Accident_Number	Date_Created	Work_Days_Lost	Product_ID	Employee_Name	Employee_Type
-----------------	--------------	----------------	------------	---------------	---------------

- Creation of the Accident table, as well as the index on Date\_Created

```

133 CREATE TABLE Account (
134     Account_Number INT PRIMARY KEY,
135     Date_Created DATE,
136     Product_ID INT UNIQUE,
137     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
138 );
139
140
141 SELECT * FROM Account

```

Results Messages

Account_Number	Date_Created	Product_ID
----------------	--------------	------------

- Creation of the Account table

```

141 CREATE TABLE Product1_Account (
142     Account_Number INT PRIMARY KEY,
143     Product_Cost REAL,
144     Product_ID INT UNIQUE,
145     FOREIGN KEY (Account_Number) REFERENCES Account(Account_Number),
146     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
147 );
148
149
150 SELECT * FROM Product1_Account

```

Results Messages

Account_Number	Product_Cost	Product_ID
----------------	--------------	------------

- Creation of the Product1\_Account table

```

150 CREATE TABLE Product2_Account (
151     Account_Number INT PRIMARY KEY,
152     Product_Cost REAL,
153     Product_ID INT UNIQUE,
154
155     FOREIGN KEY (Account_Number) REFERENCES Account(Account_Number),
156     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
157 );
158
159 SELECT * FROM Product2_Account

```

**Results**    Messages

Account_Number	Product_Cost	Product_ID
----------------	--------------	------------

- Creation of the Product2\_Account table

```

159 CREATE TABLE Product3_Account (
160     Account_Number INT PRIMARY KEY,
161     Product_Cost REAL,
162     Product_ID INT UNIQUE,
163
164     FOREIGN KEY (Account_Number) REFERENCES Account(Account_Number),
165     FOREIGN KEY (Product_ID) REFERENCES Product(ID)
166 );
167
168 SELECT * FROM Product3_Account

```

**Results**    Messages

Account_Number	Product_Cost	Product_ID
----------------	--------------	------------

- Creation of the Product3\_Account table

```

168 CREATE TABLE Repair_Complaint (
169     ID INT PRIMARY KEY,
170     Technical_Staff_Name varchar(20),
171     Date_Created varchar(10),
172     Product_ID INT,
173     Complaint_ID INT,
174
175     FOREIGN KEY (Technical_Staff_Name) REFERENCES Technical_Staff(Technical_Staff_Name),
176     FOREIGN KEY (Product_ID) REFERENCES Product(ID),
177     FOREIGN KEY (Complaint_ID) REFERENCES Complaint(Complaint_ID)
178 );
179
180 SELECT * FROM Repair_Complaint

```

Results Messages

ID	Technical_Staff...	Date_Created	Product_ID	Complaint_ID
----	--------------------	--------------	------------	--------------

- Creation of the Repair\_Complaint table

```

180 CREATE TABLE Repair_Request (
181     ID INT PRIMARY KEY,
182     Technical_Staff_Name varchar(20),
183     Quality_Controller_Name varchar(20),
184     Product_ID INT
185 );
186
187 -- Index on Quality Controller for Repair_Request
188 CREATE INDEX Repair_Request_Quality_Controller ON Repair_Request(Quality_Controller_Name asc)
189
190 SELECT * FROM Repair_Complaint

```

Results Messages

ID	Technical_Staff...	Date_Created	Product_ID	Complaint_ID
----	--------------------	--------------	------------	--------------

- Creation of the Repair\_Request table, as well as the index on Quality\_Controller\_Name

## Task 5

### 5.1: SQL statements and Transact SQL stored procedures

This section will go in order by query and show the SQL query and TSQL Procedures, if there are any

#### 1) Enter a new employee

```
"INSERT INTO Employee(Employee_Name, Address, Salary, Employee_Type) VALUES(" +  
Employee_Name + ", " + Address + ", " + Salary + ", " + TECHNICAL_STAFF + ")"
```

For inserting a technical staff:

```
"INSERT INTO Technical_Staff(Technical_Staff_Name, Position) VALUES(" + Employee_Name + ", " +  
Position + ")"
```

```
"INSERT INTO Technical_Staff_Degree(Name, Degree) VALUES(" + Employee_Name + ", " + degree + ")"
```

For inserting a worker:

```
"INSERT INTO Worker(Worker_Name, Max_Products) VALUES(" + Employee_Name + ", " +  
Max_Products + ")"
```

For inserting a quality controller:

```
"INSERT INTO Quality_Controller(Quality_Controller_Name, Product_Type) VALUES(" + Employee_Name  
+ ", " + Product_Type + ")"
```

#### 2) Enter a new product associated with the person who made the product, repaired the product if it is repaired, or checked the product

The stored procedure to enter a product called sp\_enterProduct:

```
3) DROP PROCEDURE IF EXISTS sp_enterProduct
4)
5) GO
6)
7) CREATE PROCEDURE sp_enterProduct
8)
9) @ID INT,
10) @Date_Created Date,
11) @Days_Developed INT,
12) @Produced_By varchar(20),
13) @Tested_By varchar(20),
14) @Repaired_By varchar(20),
15) @Size varchar(20),
16) @Product_Type INT
17)
18) AS
19)
20) BEGIN
21)
22) IF @Repaired_By = 'NULL'
23) INSERT INTO Product(ID, Date_Created, Days_Developed, Produced_By, Tested_By,
    Repaired_By, Size, Product_Type)
24) VALUES(@ID, @Date_Created, @Days_Developed, @Produced_By, @Tested_By, NULL,
    @Size, @Product_Type)
25)
```



```

26) ELSE
27)   INSERT INTO Product(ID, Date_Created, Days_Developed, Produced_By, Tested_By,
        Repaired_By, Size, Product_Type)
28)   VALUES(@ID, @Date_Created, @Days_Developed, @Produced_By, @Tested_By,
        @Repaired_By, @Size, @Product_Type)
29)
30) END

```

If inserting Product 1:

```
"INSERT INTO Product1(Product1_ID, Software) VALUES(" + Product_ID + ", " + Software + ")"
```

If inserting Product 2:

```
"INSERT INTO Product2(Product2_ID, Color) VALUES(" + Product_ID + ", " + Color + ")"
```

If inserting Product 3:

```
"INSERT INTO Product3(Product3_ID, Weight) VALUES(" + Product_ID + ", " + Weight + ")"
```

3) Enter a customer associated with some products

```
"INSERT INTO Customer(Name, Address) VALUES(" + Customer_Name + ", " + Address + ")"
```

```
"INSERT INTO Purchase(Customer_Name, Product_ID) VALUES(" + Customer_Name + ", " +
Product_ID + ")"
```

4) Create a new account associated with a product

The stored procedure to enter an account called sp\_enterAccount:

```
DROP PROCEDURE IF EXISTS sp_enterAccount
```

```

GO

CREATE PROCEDURE sp_enterAccount

@Account_Number INT,
@Date_Created Date,
@Product_Cost REAL,
@Product_ID INT

AS

BEGIN

DECLARE @Product_Type INT

SET @Product_TYPE = (SELECT Product_Type FROM Product WHERE ID = @Product_ID)

INSERT INTO Account(Account_Number, Date_Created, Product_ID)
VALUES(@Account_Number, @Date_Created, @Product_ID)

IF @Product_Type = 1
    BEGIN
        INSERT INTO Product1_Account(Account_Number, Product_Cost, Product_ID)
        VALUES(@Account_Number, @Product_Cost, @Product_ID)
    END

IF @Product_Type = 2

```

```

BEGIN

    INSERT INTO Product2_Account(Account_Number, Product_Cost, Product_ID)

    VALUES(@Account_Number, @Product_Cost, @Product_ID)

END

IF @Product_Type = 3

BEGIN

    INSERT INTO Product3_Account(Account_Number, Product_Cost, Product_ID)

    VALUES(@Account_Number, @Product_Cost, @Product_ID)

END

END

```

5) Enter a complaint associated with a customer and product

The stored procedure to enter an account called sp\_Complaint:

```

6) DROP PROCEDURE IF EXISTS sp_Complaint
7)
8) GO
9)
10) CREATE PROCEDURE sp_Complaint
11)
12) @Complaint_ID INT,
13) @Date_Created DATE,
14) @Description varchar(250),
15) @Treatment varchar(100),
16) @Customer_Name varchar(20),

```

```

17) @Product_ID INT
18)
19) AS
20)
21) BEGIN
22)
23) DECLARE @NUMBER INT -- Check if the customer has actually purchased the product before
24)
25) SET @NUMBER = (SELECT COUNT(Product_ID) FROM Purchase WHERE Customer_Name =
    @Customer_Name AND Product_ID = @Product_ID)
26)
27) IF @NUMBER = 0
28)     BEGIN
29)         RAISERROR ('%s has not bought a product with the ID of %d', 14, 1, @Customer_Name,
    @Product_ID)
30)     END
31)
32) ELSE
33)     BEGIN
34)         INSERT INTO Complaint(Complaint_ID, Date_Created, Description, Treatment,
    Customer_Name, Product_ID)
35)         VALUES(@Complaint_ID, @Date_Created, @Description, @Treatment, @Customer_Name,
    @Product_ID)
36)     END
37) END

```

The stored procedure to enter a repair on a complaint called sp\_ComplaintAndRepair:

```
DROP PROCEDURE IF EXISTS sp_ComplaintAndRepair

GO

CREATE PROCEDURE sp_ComplaintAndRepair

@ID INT,
@Technical_Staff_Name varchar(20),
@Date_Created Date,=
@Product_ID INT,
@Complaint_ID INT

AS

BEGIN

INSERT INTO Repair_Complaint(ID, Technical_Staff_Name, Date_Created, Product_ID, Complaint_ID)
VALUES(@ID, @Technical_Staff_Name, @Date_Created, @Product_ID, @Complaint_ID)

-- Update the Product's Technical Staff Name (it is now being repaired)
UPDATE Product
SET Repaired_By = @Technical_Staff_Name
WHERE @ID = ID

END
```

6) Enter an accident associated with an appropriate employee and product

The stored procedure to enter an accident called sp\_enterAccident

```
DROP PROCEDURE IF EXISTS sp_enterAccident

GO

CREATE PROCEDURE sp_enterAccident

@Accident_Number INT,

@Date_Created Date,

@Work_Days_Lost INT,

@Product_ID INT,

@Employee_Name varchar(20)

AS

BEGIN

DECLARE @Employee_Type varchar(20)

-- Get the employee type of who had the accident

SET @Employee_TYPE = (SELECT Employee_Type FROM Employee WHERE Employee_Name =

@Employee_Name)
```

```

INSERT INTO Accident(Accident_Number, Date_Created, Work_Days_Lost, Product_ID, Employee_Name,
Employee_Type)
VALUES(@Accident_Number, @Date_Created, @Work_Days_Lost, @Product_ID, @Employee_Name,
@Employee_Type)

END

```

7) Retrieve the date produced and time spent to produce a particular product

```
"SELECT Date_Created, Days_Developed FROM Product WHERE ID = " + Product_ID + """
```

8) Retrieve all products made by a particular worker

```
"SELECT ID FROM Product WHERE Produced_By = " + Employee_Name + """
```

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

```
"SELECT COUNT(ID) FROM Product WHERE Tested_By = " + Employee_Name + " AND
Repaired_By IS NOT NULL"
```

10) Retrieve the total costs of the products in the product3 category which were repaired at the request of a particular quality controller

```
"SELECT ROUND(SUM(Product_Cost),2) FROM Product3_Account JOIN Repair_Request ON
Product3_Account.Product_ID = Repair_Request.Product_ID Where
Repair_Request.Quality_Controller_Name = " + Employee_Name + """
```

11) Retrieve all customers (in name order) who purchased all products of a particular color

```
"SELECT Customer_Name FROM Purchase INNER JOIN Product2 ON Purchase.Product_ID =  
Product2.Product2_ID WHERE Color = '" + Color + "' ORDER BY Customer_Name ASC"
```

12) Retrieve all employees whose salary is above a particular salary

```
"SELECT * FROM Employee WHERE Salary > '" + Salary + "'"
```

13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints

```
"SELECT SUM(Work_Days_Lost) FROM Accident Where Employee_Type = '" + Employee_Name + "'"
```

14) Retrieve the average cost of all products made in a particular year

```
"SELECT ROUND(AVG(Product_Cost),2) AS AverageCost FROM (SELECT Product_Cost FROM  
Product1_Account JOIN Product ON Product1_Account.Product_ID = Product.ID WHERE  
Product.Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "'" + "UNION SELECT  
Product_Cost FROM Product2_Account JOIN Product ON Product2_Account.Product_ID = Product.ID  
WHERE Product.Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "'" + "UNION  
SELECT Product_Cost FROM Product3_Account JOIN Product ON Product3_Account.Product_ID =  
Product.ID WHERE Product.Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "' )  
AverageCost"
```

15) Delete all accidents whose dates are in some range

Get the accident table first:

```
"SELECT * FROM Accident"
```



Now delete:

```
"DELETE FROM Accident WHERE Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End +
''"
```

Get the accident table again to see what was deleted:

```
"SELECT * FROM Accident"
```

16) Import: Enter new employees from a data file until the file is empty

```
"INSERT INTO Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" +
Employee_Name + "', '" + Address + "', " + Salary + ", '" + TECHNICAL_STAFF + "')"
```

For inserting a technical staff:

```
"INSERT INTO Technical_Staff(Technical_Staff_Name, Position) VALUES('" + Employee_Name + "', "
+ Position + "')"
```

```
"INSERT INTO Technical_Staff_Degree(Name, Degree) VALUES('" + Employee_Name + "', " + degree
+ "')"
```

For inserting a worker:

```
"INSERT INTO Worker(Worker_Name, Max_Products) VALUES('" + Employee_Name + "', " +
Max_Products + "')"
```

For inserting a quality controller:

```
"INSERT INTO Quality_Controller(Quality_Controller_Name, Product_Type) VALUES('" +
Employee_Name + "', " + Product_Type + "')"
```

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

```
"SELECT Customer_Name FROM Purchase INNER JOIN Product2 ON Purchase.Product_ID =  
Product2.Product2_ID WHERE Color = " + Color + " ORDER BY Customer_Name ASC"
```

## 5.2: The Java source program and screenshots showing it's successful

```
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.CallableStatement;  
import java.sql.Connection;  
import java.sql.Statement;  
import java.util.Arrays;  
import java.util.Scanner;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.DriverManager;  
public class Individual_Project {  
  
    // Database credentials  
    final static String HOSTNAME = "devo0008-sql-server.database.windows.net";  
    final static String DBNAME = "cs-dsa-4513-sql-db";  
    final static String USERNAME = "devo0008";  
    final static String PASSWORD = "n0mathwords1!";  
  
    // Database connection string  
    final static String URL =  
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=t  
rue;trustServerCertificate=false;hostNameInCertificate=*.database.windows.net;log  
inTimeout=30;", HOSTNAME, DBNAME, USERNAME, PASSWORD);  
  
    // Input Variables  
    static final String TECHNICAL_STAFF = "Technical Staff";  
    static final String WORKER = "Worker";  
    static final String QUALITY_CONTROLLER = "Quality Controller";  
    static final int PRODUCT1 = 1;  
    static final int PRODUCT2 = 2;  
    static final int PRODUCT3 = 3;  
  
    // User input prompt
```

```

    final static String PROMPT = "Welcome to Hunter's Warehouse!\n" + "Please
select one of the options below:\n" + "1) Insert a new employee \n" + "2) Insert
a new product\n" +
    "3) Insert a customer that has purchased one or more products\n" + "4) Insert
a new account associated with a product\n" + "5) Enter a complaint associated
with a customer and product\n" +
    "6) Log an accident between an employee and product\n" + "7) Retrieve the
date produced and time spent to produce a particular product\n" + "8) Retrieve
all products made by a particular worker\n" +
    "9) Retrieve the total number of errors a particular quality controller
made\n" + "10) Retrieve the total costs of the products in the Product 3 category
which were repaired at the request of a particular quality controller\n" +
    "11) Retrieve all customers in name order who purchased all products of a
particular color\n" + "12) Retrieve all employees whose salary is above a
particular salary\n" +
    "13) Retrieve the total number of workdays lost due to accidents in repairing
the products which got complaints\n" + "14) Retrieve the average cost of all
products made in a particular year\n" + "15) Delete all accidents whose dates are
in some range\n" +
    "16) Import: Enter new employees from a data file until the file is empty\n"
+ "17) Export: Retrieve all customers in name order who purchased all products of
a particular color and output them to a data file\n" + "18) Exit!";

    public static void main(String[] args) throws SQLException, IOException {
        final Scanner sc = new Scanner(System.in); // Scanner is used to collect
the user input
        String option = ""; // Initialize user option selection as nothing

        while (!option.equals("18")) { // As user for options until option 4 is
selected
            System.out.println(PROMPT); // Print the available options
            option = sc.nextLine(); // Read in the user option selection
            switch (option) { // Switch between different options

                case "1": // Insert a new Employee
                    System.out.println("Choose the type of employee to insert:\n"
+ "1) Technical Staff\n" + "2) Worker\n" + "3) Quality Controller");
                    option = sc.next();
                    sc.nextLine();

                    String Employee_Name;
                    String Address;
                    Double Salary;
                    String Position;

```

```

switch(option) {

    // Insert a new Technical Staff and their degrees
    case "1":
        String degree;

        System.out.println("Insert Technician name:");
        Employee_Name = sc.nextLine();

        System.out.println("Insert Technician address:");
        Address = sc.nextLine();

        System.out.println("Insert Technician salary:");
        Salary = sc.nextDouble();
        sc.nextLine();

        System.out.println("Insert Technician position:");
        Position = sc.nextLine();

        // Insert into Employee table
        System.out.println("Inserting into Employee table. .");

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();

            statement.execute("INSERT INTO
Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" + Employee_Name
+ "', '" + Address + "', " + Salary + ", '" + TECHNICAL_STAFF + "')");
            System.out.println("Execution complete!");
        }

        System.out.println();

        // Insert into Technical Staff table
        System.out.println("Inserting into Technical Staff
table. . .");

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();

            statement.execute("INSERT INTO
Technical_Staff(Technical_Staff_Name, Position) VALUES('" + Employee_Name + "',
'" + Position + "')");

            System.out.println("Execution complete!");

```

```

    }

    System.out.println();

    System.out.println("Insert number of degrees this
Technical Staff has (BS, MS, PHD):");
    int numDegrees = sc.nextInt();
    sc.nextLine();

    // Loop based on number of degrees
    for(int i = 0; i < numDegrees; i++) {
        System.out.println("Insert degree " + (i + 1));
        degree = sc.nextLine();

        // Insert into Technical Staff Degree table
        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();
            statement.execute("INSERT INTO
Technical_Staff_Degree(Name, Degree) VALUES('\" + Employee_Name + \"', '\" + degree
+ \"')");
            System.out.println("Execution
complete!");
        }
    }

    System.out.println();
    break;

    // Insert a new Worker
    case "2":
        int Max_Products;

        System.out.println("Insert Worker name:");
        Employee_Name = sc.nextLine();

        System.out.println("Insert Worker address:");
        Address = sc.nextLine();

        System.out.println("Insert Worker salary:");
        Salary = sc.nextDouble();

        System.out.println("Insert the Worker's maximum
number of products:");

```

```

        Max_Products = sc.nextInt();
        sc.nextLine();

        // Insert into the Employee table
        System.out.println("Inserting into Employee table. .
.");
        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();
            statement.execute("INSERT INTO
Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" + Employee_Name
+ "', '" + Address + "', " + Salary + ", '" + WORKER + "')");
            System.out.println("Execution complete!");
        }

        System.out.println();

        // Insert into the Employee table
        System.out.println("Inserting into Worker table. .
.");
        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();
            statement.execute("INSERT INTO
Worker(Worker_Name, Max_Products) VALUES('" + Employee_Name + "', '" +
Max_Products + "')");
            System.out.println("Execution complete!");
        }

        System.out.println();
        break;

        // Insert a new Quality Controller
        case "3" :
            String Product_Type;

            System.out.println("Insert Quality Controller
name:");

            Employee_Name = sc.nextLine();

            System.out.println("Insert Quality Controller
address:");

```

```

        Address = sc.nextLine();

        System.out.println("Insert Quality Controller
salary:");

        Salary = sc.nextDouble();
        sc.nextLine();

        System.out.println("Insert the Quality Controller's
product type:");

        Product_Type = sc.nextLine();

        // Insert into the Employee table
        System.out.println("Inserting into Employee table. .
.");

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();

            statement.execute("INSERT INTO
Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" + Employee_Name
+ "', '" + Address + "', " + Salary + ", '" + QUALITY_CONTROLLER + "')");
            System.out.println("Execution complete!");
        }

        System.out.println();

        // Insert into the Quality Controller table
        System.out.println("Inserting into Quality Controller
table. . .");

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();

            statement.execute("INSERT INTO
Quality_Controller(Quality_Controller_Name, Product_Type) VALUES('" +
Employee_Name + "', '" + Product_Type + "')");
            System.out.println("Execution complete!");
        }

        System.out.println();
        break;

    default:
        System.out.println("Unrecognized option: " + option +
"\nPlease try again!");

```

```

        break;
    }

    break;

    // Insert a new product of type 1, 2, or 3
    case "2":
        System.out.println("Choose the type of product to insert:\n"
+ "1) Product 1\n" + "2) Product 2\n" + "3) Product 3");
        option = sc.nextLine();
        int Product_ID;
        String Date_Created;
        int Days_Developed;
        String Produced_By;
        String Tested_By;
        String Repaired_By;
        String Size;

        switch(option) {

            // Insert product 1
            case "1" :
                String Software;

                System.out.println("Insert product ID:");
                Product_ID = sc.nextInt();
                sc.nextLine();

                System.out.println("Insert product creation date:");
                Date_Created = sc.nextLine();

                System.out.println("Insert number of days it took to
develop the product:");
                Days_Developed = sc.nextInt();
                sc.nextLine();

                System.out.println("Insert the name of the worker
that produced the product:");
                Produced_By = sc.nextLine();

                System.out.println("Insert the name of the quality
controller that tested the product:");
                Tested_By = sc.nextLine();

```



```

        System.out.println("Insert the name of the technician
that repaired the product, if applicable (if not, insert 'NULL'):");
        Repaired_By = sc.nextLine();

        System.out.println("Enter the product size:");
        Size = sc.nextLine();

        System.out.println("Enter the product software:");
        Software = sc.nextLine();

        // Execute TSQL Statement to insert a product into
the Product table (needed because we cannot directly pass a NULL value
recognizable by Azure)
        try(final Connection connection =
DriverManager.getConnection(URL)) {
            CallableStatement myStmt =
connection.prepareCall("{call sp_enterProduct(?, ?, ?, ?, ?, ?, ?, ?)}");

            myStmt.setInt(1, Product_ID);
            myStmt.setString(2, Date_Created);
            myStmt.setInt(3, Days_Developed);
            myStmt.setString(4, Produced_By);
            myStmt.setString(5, Tested_By);
            myStmt.setString(6, Repaired_By);
            myStmt.setString(7, Size);
            myStmt.setInt(8, PRODUCT1);
            myStmt.execute();
            System.out.println("Execution Complete!");
        }

        System.out.println();

        // Insert into the Product 1 table
        System.out.println("Inserting into Product 1 table. .
.");

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();
            statement.execute("INSERT INTO
Product1(Product1_ID, Software) VALUES(" + Product_ID + ", '" + Software + "'");
            System.out.println("Execution complete!");
        }

        System.out.println();

```

```

        break;

        // Insert product 2
        case "2":

            String Color;

            System.out.println("Insert product ID:");
            Product_ID = sc.nextInt();
            sc.nextLine();

            System.out.println("Insert product creation date:");
            Date_Created = sc.nextLine();

            System.out.println("Insert number of days it took to
develop the product:");
            Days_Developed = sc.nextInt();
            sc.nextLine();

            System.out.println("Insert the name of the worker
that produced the product:");
            Produced_By = sc.nextLine();

            System.out.println("Insert the name of the quality
controller that tested the product:");
            Tested_By = sc.nextLine();

            System.out.println("Insert the name of the technician
that repaired the product, if applicable:");
            Repaired_By = sc.nextLine();

            System.out.println("Enter the product size:");
            Size = sc.nextLine();

            System.out.println("Enter the product color:");
            Color = sc.nextLine();

            // Execute TSQL Statement to insert a product into
            the Product table (needed because we cannot directly pass a NULL value
            recognizable by Azure)
            try(final Connection connection =
DriverManager.getConnection(URL)) {
                CallableStatement myStmt =
connection.prepareCall("{call sp_enterProduct(?, ?, ?, ?, ?, ?, ?, ?)}");

```

```

        myStmt.setInt(1, Product_ID);
        myStmt.setString(2, Date_Created);
        myStmt.setInt(3, Days_Developed);
        myStmt.setString(4, Produced_By);
        myStmt.setString(5, Tested_By);
        myStmt.setString(6, Repaired_By);
        myStmt.setString(7, Size);
        myStmt.setInt(8, PRODUCT2);
        myStmt.execute();
        System.out.println("Statement executed!");
    }

    System.out.println();

    // Insert into the Product 2 table
    System.out.println("Inserting into Product 2 table. .
.");
    try (final Connection connection =
DriverManager.getConnection(URL)) {
        final Statement statement =
connection.createStatement();
        statement.execute("INSERT INTO
Product2(Product2_ID, Color) VALUES(" + Product_ID + ", '" + Color + "'");
        System.out.println("Execution complete!");
    }

    System.out.println();
    break;

    // Insert product 3
    case "3":
        String Weight;

        System.out.println("Insert product ID:");
        Product_ID = sc.nextInt();
        sc.nextLine();

        System.out.println("Insert product creation date:");
        Date_Created = sc.nextLine();

        System.out.println("Insert number of days it took to
develop the product:");
        Days_Developed = sc.nextInt();
        sc.nextLine();

```

```

        System.out.println("Insert the name of the worker
that produced the product:");
        Produced_By = sc.nextLine();

        System.out.println("Insert the name of the quality
controller that tested the product:");
        Tested_By = sc.nextLine();

        System.out.println("Insert the name of the technician
that repaired the product, if applicable:");
        Repaired_By = sc.nextLine();

        System.out.println("Enter the product size:");
        Size = sc.nextLine();

        System.out.println("Enter the product weight:");
        Weight = sc.nextLine();

        // Execute TSQL Statement to insert a product into
the Product table (needed because we cannot directly pass a NULL value
recognizable by Azure)
        try(final Connection connection =
DriverManager.getConnection(URL)) {
            CallableStatement myStmt =
connection.prepareCall("{call sp_enterProduct(?, ?, ?, ?, ?, ?, ?, ?)}");

            myStmt.setInt(1, Product_ID);
            myStmt.setString(2, Date_Created);
            myStmt.setInt(3, Days_Developed);
            myStmt.setString(4, Produced_By);
            myStmt.setString(5, Tested_By);
            myStmt.setString(6, Repaired_By);
            myStmt.setString(7, Size);
            myStmt.setInt(8, PRODUCT3);
            myStmt.execute();
            System.out.println("Statement executed!");
        }

        System.out.println();

        // Insert into the Product 3 table
        System.out.println("Inserting into Product 3 table. .
.");

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

```

```

        final Statement statement =
connection.createStatement();
        statement.execute("INSERT INTO
Product3(Product3_ID, Weight) VALUES(" + Product_ID + ", '" + Weight + "')");
        System.out.println("Execution complete!");
    }

    System.out.println();
    break;

    default:
        System.out.println("Unrecognized option: " + option +
"\nPlease try again!");
        break;
    }

    break;

    // Associate a customer with one or more products they have
bought
    case "3":
        String Answer;
        String Customer_Name;
        int numProducts;

        // Ask if this is a new customer, so we are not re-inserting
a customer that already exists
        System.out.println("Is this a new customer? (Yes / No)");
        Answer = sc.nextLine();

        if(Answer.equals("Yes") || Answer.equals("yes")) {
            System.out.println("Insert customer name:");
            Customer_Name = sc.nextLine();

            System.out.println("Insert customer address:");
            Address = sc.nextLine();

            System.out.println("How many products has " +
Customer_Name + " bought?");
            numProducts = sc.nextInt();
            sc.nextLine();

            // Since this is a new customer, insert them into the
Customer table
            System.out.println("Inserting into Customer table. . .");

```

```

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement =
connection.createStatement();
            statement.execute("INSERT INTO Customer(Name,
Address) VALUES('\" + Customer_Name + '\", '\" + Address + '\")");
            System.out.println("Execution complete!");
        }

        System.out.println();

        // Loop based on the number of products the customer
bought

        for(int i = 0; i < numProducts; i++) {
            System.out.println("Insert product ID:");
            Product_ID = sc.nextInt();
            sc.nextLine();

            // Insert the customer, product pair into the
Purchase table

            System.out.println("Inserting into Purchase table. .
.");

            try (final Connection connection =
DriverManager.getConnection(URL)) {
                final Statement statement =
connection.createStatement();
                statement.execute("INSERT INTO
Purchase(Customer_Name, Product_ID) VALUES('\" + Customer_Name + '\", '\" +
Product_ID + '\")");
                System.out.println("Execution complete!");
                System.out.println();
            }
        }

        // Answer is no, so there is no need to insert them into the
Customer table

        else if(Answer.equals("No") || Answer.equals("no")) {
            System.out.println("Insert customer name:");
            Customer_Name = sc.nextLine();

            System.out.println("How many products has \" +
Customer_Name + \" bought?");
            numProducts = sc.nextInt();
            sc.nextLine();

```

```

        // Loop based on the number of products the customer
bought
        for(int i = 0; i < numProducts; i++) {
            System.out.println("Insert product ID:");
            Product_ID = sc.nextInt();
            sc.nextLine();

            // Insert the customer, product pair into the
Purchase table

            System.out.println("Inserting into Purchase table. .
.");

            try (final Connection connection =
DriverManager.getConnection(URL)) {
                final Statement statement =
connection.createStatement();
                statement.execute("INSERT INTO
Purchase(Customer_Name, Product_ID) VALUES('\" + Customer_Name + '\", '\" +
Product_ID + '\"')");

                System.out.println("Execution complete!");
                System.out.println();
            }
        }

        // Unrecognized answer
        else {
            System.out.println("Answer not given\n");
            break;
        }

        break;

// Insert a new account associated with a product
case"4":
    int Account_Number;
    double Product_Cost;

    System.out.println("Insert account number:");
    Account_Number = sc.nextInt();
    sc.nextLine();

    System.out.println("Insert account creation date");
    Date_Created = sc.nextLine();

```

```

        System.out.println("Insert product cost:");
        Product_Cost = sc.nextDouble();
        sc.nextLine();

        System.out.println("Insert product ID");
        Product_ID = sc.nextInt();
        sc.nextLine();

        // Execute TSQL Statement to insert an account into the
        Account table (needed to determine whether to insert into the Product1 Account,
        Product2 Account, or Product3 Account tables based on what type of product the
        Product ID is)
        try(final Connection connection =
        DriverManager.getConnection(URL)) {
            CallableStatement myStmt = connection.prepareCall("{call
        sp_enterAccount(?, ?, ?, ?)}");

            myStmt.setInt(1, Account_Number);
            myStmt.setString(2, Date_Created);
            myStmt.setDouble(3, Product_Cost);
            myStmt.setInt(4, Product_ID);
            myStmt.execute();
            System.out.println("Statement executed!");
        }

        System.out.println();
        break;

        // Insert a complaint associated with a customer and a product
        case"5":
            int Complaint_ID;
            String Description;
            String Treatment;
            int Repair_Complaint_ID;

            System.out.println("Insert Complaint ID:");
            Complaint_ID = sc.nextInt();
            sc.nextLine();

            System.out.println("Insert date the complaint was created:");
            Date_Created = sc.nextLine();

            System.out.println("Insert description of the complaint (250
        chars or less):");
            Description = sc.nextLine();

```



```

        System.out.println("Insert treatment of complaint:");
        Treatment = sc.nextLine();

        System.out.println("Insert customer name:");
        Customer_Name = sc.nextLine();

        System.out.println("Insert product ID:");
        Product_ID = sc.nextInt();
        sc.nextLine();

        // Execute TSQL Statement to insert a complaint into the
        Complaint table (needed to determine whether a customer has actually purchased
        the product they are complaining about)
        try(final Connection connection =
DriverManager.getConnection(URL)) {
            CallableStatement myStmt = connection.prepareCall("{call
sp_Complaint(?, ?, ?, ?, ?, ?)}");

            myStmt.setInt(1, Complaint_ID);
            myStmt.setString(2, Date_Created);
            myStmt.setString(3, Description);
            myStmt.setString(4, Treatment);
            myStmt.setString(5, Customer_Name);
            myStmt.setInt(6, Product_ID);
            myStmt.execute();
            System.out.println("Statement executed!");
        }

        System.out.println();
        System.out.println("Since we have a complaint, we need to
repair it!\n");

        System.out.println("Insert repair ID:");
        Repair_Complaint_ID = sc.nextInt();
        sc.nextLine();

        System.out.println("Insert Technician that will repair the
complaint:");
        Employee_Name = sc.nextLine();

        System.out.println("Insert date that the repair was
requested:");
        Date_Created = sc.nextLine();

```

```

        // Execute TSQL Statement to insert a repair for a complaint
        into the Repair Complaint table (needed to update the product's associated
        Technical Staff, since it is now being repaired)
        try(final Connection connection =
DriverManager.getConnection(URL)) {
            CallableStatement myStmt = connection.prepareCall("{call
sp_ComplaintAndRepair(?, ?, ?, ?, ?)}");

            myStmt.setInt(1, Repair_Complaint_ID);
            myStmt.setString(2, Employee_Name);
            myStmt.setString(3, Date_Created);
            myStmt.setInt(4, Product_ID);
            myStmt.setInt(5, Complaint_ID);
            myStmt.execute();
            System.out.println("Statement executed!");
        }

        System.out.println();
        break;

// Insert an accident between an employee and product
case "6":
    int Accident_Number;
    int Work_Days_Lost;

    System.out.println("Insert accident number");
    Accident_Number = sc.nextInt();
    sc.nextLine();

    System.out.println("Insert date of accident:");
    Date_Created = sc.nextLine();

    System.out.println("Insert number of work days lost:");
    Work_Days_Lost = sc.nextInt();
    sc.nextLine();

    System.out.println("Insert product ID:");
    Product_ID = sc.nextInt();
    sc.nextLine();

    System.out.println("Insert the Technical Staff or Worker that
had the accident:");
    Employee_Name = sc.nextLine();

```

```

        // Execute TSQL Statement to insert a repair for a product
        into the Repair Accident table (needed to determine if a Quality Controller or
        Technical Staff damaged the product)
        try(final Connection connection =
DriverManager.getConnection(URL)) {
            CallableStatement myStmt = connection.prepareCall("{call
sp_enterAccident(?, ?, ?, ?, ?)}");

            myStmt.setInt(1, Accident_Number);
            myStmt.setString(2, Date_Created);
            myStmt.setInt(3, Work_Days_Lost);
            myStmt.setInt(4, Product_ID);
            myStmt.setString(5, Employee_Name);
            myStmt.execute();
            System.out.println("Statement executed!");
        }

        System.out.println();
        break;

// Get the date a product was produced and how many days it took
to create it
case "7":

    System.out.println("Insert product ID");
    Product_ID = sc.nextInt();
    sc.nextLine();

    try (final Connection connection =
DriverManager.getConnection(URL)) {
        try (
            final Statement statement =
connection.createStatement();
            final ResultSet resultSet =
statement.executeQuery("SELECT Date_Created, Days_Developed FROM Product WHERE ID
= '" + Product_ID + "'")) {
            System.out.println("Contents of the Product
table:");

            System.out.println("Date Created | Days Developed
");

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

```

```

        }
    }

    System.out.println();
    break;

    // Get all of the product ID's made by a particular Worker
    case "8":
        System.out.println("Insert Worker name:");
        Employee_Name = sc.nextLine();

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            try (
                final Statement statement =
connection.createStatement();
                final ResultSet resultSet =
statement.executeQuery("SELECT ID FROM Product WHERE Produced_By = '" +
Employee_Name + "'")) {
                System.out.println("Contents of the Product
table:");

                System.out.println("| Product ID |");
                while (resultSet.next()) {
                    System.out.println(String.format("| %s |",
resultSet.getString(1)));
                }
            }
        }

        System.out.println();
        break;

    // Get the total number of errors a particular Quality Controller
has made (i.e., how many products were certified and have now been repaired by a
Technical Staff due to a complaint)
    case "9":

        System.out.println("Insert quality controller name:");
        Employee_Name = sc.nextLine();

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            try (
                final Statement statement =
connection.createStatement();

```

```

        final ResultSet resultSet =
statement.executeQuery("SELECT COUNT(ID) FROM Product WHERE Tested_By = '" +
Employee_Name + "' AND Repaired_By IS NOT NULL")) {
            System.out.println("Contents of the Product
table:");

            System.out.println("| Number of Errors |");
            while (resultSet.next()) {
                System.out.println(String.format("| %s |",
resultSet.getString(1)));
            }
        }

        System.out.println();
        break;

        // Get the total cost of every product in the Product 3 category
which has been repaired at the request of a particular Quality Controller
        case "10":
            System.out.println("Insert Quality Controller:");
            Employee_Name = sc.nextLine();

            try (final Connection connection =
DriverManager.getConnection(URL)) {
                try (
                    final Statement statement =
connection.createStatement();
                    final ResultSet resultSet =
statement.executeQuery("SELECT ROUND(SUM(Product_Cost),2) FROM Product3_Account
JOIN Repair_Request ON Product3_Account.Product_ID = Repair_Request.Product_ID
Where Repair_Request.Quality_Controller_Name = '" + Employee_Name + "'")) {
                    System.out.println("Contents of the table:");
                    System.out.println("| Total costs |");
                    while (resultSet.next()) {
                        System.out.println(String.format("| %s |",
resultSet.getString(1)));
                    }
                }
            }

            System.out.println();
            break;

        // Get every customer in named order that has bought a product of
a particular color

```

```

        case "11":
            String Color;

            System.out.println("Insert color:");
            Color = sc.nextLine();

            try (final Connection connection =
DriverManager.getConnection(URL)) {
                try (
                    final Statement statement =
connection.createStatement();
                    final ResultSet resultSet =
statement.executeQuery("SELECT Customer_Name FROM Purchase INNER JOIN Product2 ON
Purchase.Product_ID = Product2.Product2_ID WHERE Color = '" + Color + "' ORDER BY
Customer_Name ASC")) {
                    System.out.println("Contents of the Customer
table:");

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

            System.out.println();
            break;

// Get all employees whose salary is above a particular salary
case "12":
            System.out.println("Enter salary:");
            Salary = sc.nextDouble();
            sc.nextLine();

            try (final Connection connection =
DriverManager.getConnection(URL)) {
                try (
                    final Statement statement =
connection.createStatement();
                    final ResultSet resultSet =
statement.executeQuery("SELECT * FROM Employee WHERE Salary > '" + Salary + "'"))
            {
                System.out.println("Contents of the Employee
table:");

                System.out.println("| Work Days Lost |");

```

```

        while (resultSet.next()) {
            System.out.println(String.format("| %s | %s | %s | %s |",
                resultSet.getString(1),
                resultSet.getString(2),
                resultSet.getString(3),
                resultSet.getString(4)));
        }
    }

    System.out.println();
    break;

    // Get the total number of work days lost due to accidents in
    // repairing products that got complaints
    case "13":
        System.out.println("Enter Employee name:");
        Employee_Name = sc.nextLine();

        try (final Connection connection =
            DriverManager.getConnection(URL)) {
            try (
                final Statement statement =
                    connection.createStatement();
                final ResultSet resultSet =
                    statement.executeQuery("SELECT SUM(Work_Days_Lost) FROM Accident Where
                    Employee_Name = '" + Employee_Name + "'")) {
                System.out.println("Contents of the Employee
                table:");

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

        System.out.println();
        break;

    // Get the average cost of all products produced in a particular
    // year
    case "14":
        String Date_Start;

```

```

        String Date_End;
        String Year;

        System.out.println("Insert year:");
        Year = sc.nextLine();

        // Append the first day and month to the date
        Date_Start = Year + "/01/01";

        // Append the last day and month to the date
        Date_End = Year + "/12/31";

        try (final Connection connection =
DriverManager.getConnection(URL)) {
            try (
                final Statement statement =
connection.createStatement();
                final ResultSet resultSet =
statement.executeQuery("SELECT ROUND(AVG(Product_Cost),2) AS AverageCost FROM
(SELECT Product_Cost FROM Product1_Account JOIN Product ON
Product1_Account.Product_ID = Product.ID WHERE Product.Date_Created BETWEEN '" +
Date_Start + "' AND '" + Date_End + "'"
                    + "UNION SELECT Product_Cost FROM
Product2_Account JOIN Product ON Product2_Account.Product_ID = Product.ID WHERE
Product.Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "'"
                    + "UNION SELECT Product_Cost FROM
Product3_Account JOIN Product ON Product3_Account.Product_ID = Product.ID WHERE
Product.Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "' )
AverageCost")) {

                System.out.println("Contents of the Total Cost
table:");

                System.out.println("| Average Cost |");
                while (resultSet.next()) {
                    System.out.println(String.format("| %s |",
resultSet.getString(1)));
                }
            }
        }

        System.out.println();
        break;

        // Delete all accidents whose dates are between a particular
range of dates
        case "15":

```



```

        System.out.println("Insert starting date:");
        Date_Start = sc.nextLine();

        System.out.println("Insert ending date:");
        Date_End = sc.nextLine();

        System.out.println();
        System.out.println("Initial Accident table:");

        // We print out the initial Accident table first to visualize
        any changes after deletion
        try (final Connection connection =
DriverManager.getConnection(URL)) {
            try (
                final Statement statement =
connection.createStatement();
                final ResultSet resultSet =
statement.executeQuery("SELECT * FROM Accident")) {
                System.out.println("Contents of the Accident
table before deletion:");

                System.out.println("| Accident Number | Date
Created | Work Days Lost | Product ID | Employee Name | Employee Type |");
                while (resultSet.next()) {
                    System.out.println(String.format("| %s | %s |
%s | %s | %s | %s |",
                        resultSet.getString(1),
                        resultSet.getString(2),
                        resultSet.getString(3),
                        resultSet.getString(4),
                        resultSet.getString(5),
                        resultSet.getString(6)));
                }
            }
        }

        System.out.println();

        // Now delete any accidents that fall into the inserted date
        range, and then get the table again
        try (final Connection connection =
DriverManager.getConnection(URL)) {
            final Statement statement = connection.createStatement();
            statement.executeUpdate("DELETE FROM Accident WHERE
Date_Created BETWEEN '" + Date_Start + "' AND '" + Date_End + "'");

```

```

        final ResultSet resultSet =
statement.executeQuery("SELECT * FROM Accident"); {
        System.out.println("Contents of the Accident
table after deletion:");

        System.out.println("| Accident Number | Date
Created | Work Days Lost | Product ID | Employee Name | Employee Type |");
        while (resultSet.next()) {
            System.out.println(String.format("| %s | %s |
%s | %s | %s | %s |",

            resultSet.getString(1),
            resultSet.getString(2),
            resultSet.getString(3),
            resultSet.getString(4),
            resultSet.getString(5),
            resultSet.getString(6)));
        }
    }

    System.out.println();
    break;

    // The import function that enters new employees from a user file
until it is empty. These employees are inserted into the Employee table,
    // and are then inserted into their respective Technical Staff
and Technical Staff Degree Tables, Worker Tables, or Quality Controller Tables
    case "16":
        String File_Name;
        String[] Temp = new String[8];
        String[] Array = new String[8];
        String Employee_Type;
        String Degree;
        int Max_Products;
        String Product_Type;

        // Get the user inserted file name
        System.out.println("Insert file name:");
        File_Name = sc.nextLine();

        // Create a new buffered reader and line that holds each line
in the file
        BufferedReader br = new BufferedReader(new
FileReader(File_Name));
        String line;
        line = br.readLine();

```

```

        // Loop until line is null
        while(line != null) {

            // Split each line up to 8 times
            Temp = line.split(",", 8);
            System.arraycopy(Temp, 0, Array, 0, Temp.length);

            Employee_Name = Array[0];
            Address = Array[1];
            Salary = Double.parseDouble(Array[2]);
            Employee_Type = Array[3];

            // The line is a Technical Staff employee
            if(Employee_Type.equals("Technical Staff")) {
                Position = Array[4];

                // Insert into the Employee table
                System.out.println("Inserting into Employee table. .
.");

                try (final Connection connection =
DriverManager.getConnection(URL)) {
                    final Statement statement =
connection.createStatement();
                    statement.execute("INSERT INTO
Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" + Employee_Name
+ "', '" + Address + "', " + Salary + ", '" + Employee_Type + "')");
                    System.out.println("Execution complete!");
                }

                System.out.println();

                // Insert into the Technical Staff table
                System.out.println("Inserting into Technical Staff
table. . .");

                try (final Connection connection =
DriverManager.getConnection(URL)) {
                    final Statement statement =
connection.createStatement();
                    statement.execute("INSERT INTO
Technical_Staff(Technical_Staff_Name, Position) VALUES('" + Employee_Name + "',
'" + Position + "')");
                    System.out.println("Execution complete!");
                }
            }

```

```

        System.out.println();

        // The maximum number of degrees a Technical Staff
        can have is 3 (BS, MS, PHD)
        for(int i = 0; i < 3; i++) {

            // If we are inserting less than 3 degrees, we
            need to exit the loop when the array is empty
            if(Array[i + 5].equals("0")) {
                break;
            }

            // Get the degree from the array
            Degree = Array[i + 5];

            // Insert the degree into the Technical Staff
            Degree table

            try (final Connection connection =
            DriverManager.getConnection(URL)) {
                final Statement statement =
            connection.createStatement();

                statement.execute("INSERT INTO
            Technical_Staff_Degree(Name, Degree) VALUES('" + Employee_Name + "', '" + Degree
            + "')");

                System.out.println("Execution
            complete!");
            }
        }

        // The line is a Worker employee
        else if(Employee_Type.equals("Worker")) {
            Max_Products = Integer.parseInt(Array[4]);

            // Insert into the Employee table
            System.out.println("Inserting into Employee table. .
            .");

            try (final Connection connection =
            DriverManager.getConnection(URL)) {
                final Statement statement =
            connection.createStatement();

                statement.execute("INSERT INTO
            Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('" + Employee_Name
            + "', '" + Address + "', " + Salary + ", '" + Employee_Type + "')");

                System.out.println("Execution complete!");
            }
        }
    }
}

```

```

    }

    System.out.println();

    // Insert into the Worker table
    System.out.println("Inserting into Worker table. .
.");
    try (final Connection connection =
DriverManager.getConnection(URL)) {
        final Statement statement =
connection.createStatement();
        statement.execute("INSERT INTO
Worker(Worker_Name, Max_Products) VALUES('\" + Employee_Name + \"', '\" +
Max_Products + \"')");
        System.out.println("Execution complete!");
    }
}

// The line is a Quality Controller employee
else if(Employee_Type.equals("Quality Controller")) {
    Product_Type = Array[4];

    // Insert into the Employee table
    System.out.println("Inserting into Employee table. .
.");
    try (final Connection connection =
DriverManager.getConnection(URL)) {
        final Statement statement =
connection.createStatement();
        statement.execute("INSERT INTO
Employee(Employee_Name, Address, Salary, Employee_Type) VALUES('\" + Employee_Name
+ \"', '\" + Address + \"', '\" + Salary + \"', '\" + Employee_Type + \"')");
        System.out.println("Execution complete!");
    }

    System.out.println();

    // Insert into the Quality Controller table
    System.out.println("Inserting into Quality Controller
table. . .");
    try (final Connection connection =
DriverManager.getConnection(URL)) {
        final Statement statement =
connection.createStatement();

```

```

        statement.execute("INSERT INTO
Quality_Controller(Quality_Controller_Name, Product_Type) VALUES(' +
Employee_Name + "', '" + Product_Type + "')");
        System.out.println("Execution complete!");
    }
}

// If the employee type is not defined, we cannot insert
else {
    System.out.println("Employee type not defined!");
}

// Remove existing information in preparation for a new
line
Arrays.fill(Array, "0");
line = br.readLine();
}

// Close the buffered reader and exit
br.close();
System.out.println();
break;

// The export function gets every customer in named order that
has bought a product of a particular color. These customers are placed into a
file at the user's specification
case "17":

    // Get the user inserted name of the file that will be
created
    System.out.println("Insert file name:");
    File_Name = sc.nextLine();

    System.out.println("Insert color:");
    Color = sc.nextLine();

    // Create a new printwriter
    PrintWriter writer = new PrintWriter(File_Name, "UTF-8");

    // Get all of the customer names. Each name will be written
to a line in the file
    try (final Connection connection =
DriverManager.getConnection(URL)) {
        try (

```

```

        final Statement statement =
connection.createStatement();
        final ResultSet resultSet =
statement.executeQuery("SELECT Customer_Name FROM Purchase INNER JOIN Product2 ON
Purchase.Product_ID = Product2.Product2_ID WHERE Color = '" + Color + "' ORDER BY
Customer_Name ASC")) {
            while (resultSet.next()) {
                Customer_Name = resultSet.getString(1);
                writer.println(Customer_Name);
            }
        }

        // Close the printwriter and exit
        writer.close();
        System.out.println();
        break;

    case "18": // Do nothing, the while loop will terminate upon the
next iteration

        System.out.println("Exiting! Goodbye!");
        break;

    default: // Unrecognized option, re-prompt the user for the
correct one

        System.out.println(String.format("Unrecognized option: %s\n"
+ "Please try again!", option));
        break;
    }
}

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

```

```
Console X
<terminated> Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 7:50:54 PM – 7:51:05 PM) [pid: 38796]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
18
Exiting! Goodbye!
```

Here we can see my program compile and execute in the terminal. I chose option 18 just to terminate it. Note - I wrote the code in Eclipse but copied it from VSCode so that it formats better in the Word document.



# Task 6: Java Program Execution

## 6.1: Query 1

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:13:39 PM) [pid: 34324]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
1
Insert Technician name:
Hunter
Insert Technician address:
2325 Louise Ln
Insert Technician salary:
10000
Insert Technician position:
IT
Inserting into Employee table. . .
Execution complete!

Inserting into Technical Staff table. . .
Execution complete!

Insert number of degrees this Technical Staff has (BS, MS, PHD):
2
Insert degree 1
BS
Execution complete!
Insert degree 2
MS
Execution complete!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:15:26 PM) [pid: 32824]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
2
Insert Worker name:
Jackson
Insert Worker address:
1410 Wisteria Ave
Insert Worker salary:
20000
Insert the Worker's maximum number of products:
23
Inserting into Employee table. . .
Execution complete!

Inserting into Worker table. . .
Execution complete!
```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:16:12 PM) [pid: 37432]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

1

Choose the type of employee to insert:

- 1) Technical Staff
- 2) Worker
- 3) Quality Controller

3

Insert Quality Controller name:

Ansley

Insert Quality Controller address:

1410 Wisteria Ave

Insert Quality Controller salary:

30000

Insert the Quality Controller's product type:

Toys

Inserting into Employee table. . .

Execution complete!

Inserting into Quality Controller table. . .

Execution complete!

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:18:53 PM) [pid: 29324]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
1
Insert Technician name:
Steven
Insert Technician address:
1321 E 24th St
Insert Technician salary:
50000
Insert Technician position:
Manager
Inserting into Employee table. . .
Execution complete!

Inserting into Technical Staff table. . .
Execution complete!

Insert number of degrees this Technical Staff has (BS, MS, PHD):
1
Insert degree 1
BS
Execution complete!
```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:23:42 PM) [pid: 41276]

```
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
```

1

Choose the type of employee to insert:

- 1) Technical Staff
- 2) Worker
- 3) Quality Controller

2

Insert Worker name:

Dieko

Insert Worker address:

1309 Traditions Way

Insert Worker salary:

28000

Insert the Worker's maximum number of products:

20

Inserting into Employee table. . .

Execution complete!

Inserting into Worker table. . .

Execution complete!

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:24:44 PM) [pid: 32380]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
3
Insert Quality Controller name:
Donald
Insert Quality Controller address:
1100 S Ocean Blvd, Palm Beach
Insert Quality Controller salary:
100000
Insert the Quality Controller's product type:
Flags
Inserting into Employee table. . .
Execution complete!

Inserting into Quality Controller table. . .
Execution complete!

```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:25:51 PM) [pid: 42680]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
1
Insert Technician name:
Caleb
Insert Technician address:
918 E Ninth St
Insert Technician salary:
42000
Insert Technician position:
Machine Worker
Inserting into Employee table. . .
Execution complete!

Inserting into Technical Staff table. . .
Execution complete!

Insert number of degrees this Technical Staff has (BS, MS, PHD):
3
Insert degree 1
BS
Execution complete!
Insert degree 2
MS
Execution complete!
Insert degree 3
PHD
Execution complete!

```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:28:02 PM) [pid: 37624]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
2
Insert Worker name:
Dylan
Insert Worker address:
602 Classen Blvd
Insert Worker salary:
53000
Insert the Worker's maximum number of products:
37
Inserting into Employee table. . .
Execution complete!

Inserting into Worker table. . .
Execution complete!
```



```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:29:51 PM) [pid: 37016]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
3
Insert Quality Controller name:
Ethan
Insert Quality Controller address:
602 Classen Blvd
Insert Quality Controller salary:
60000
Insert the Quality Controller's product type:
Computers
Inserting into Employee table. . .
Execution complete!

Inserting into Quality Controller table. . .
Execution complete!

```

Console X

Individual\_Project [Java Application] [pid: 31048]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

1

Choose the type of employee to insert:

- 1) Technical Staff
- 2) Worker
- 3) Quality Controller

2

Insert Worker name:

Hala

Insert Worker address:

1524 Asp Ave

Insert Worker salary:

33500

Insert the Worker's maximum number of products:

30

Inserting into Employee table. . .

Execution complete!

Inserting into Worker table. . .

Execution complete!

Results Messages				
	Employee_Name ▼	Address ▼	Salary ▼	Employee_Type ▼
1	Ansley	1410 Wisteria Ave	30000	Quality Controller
2	Caleb	918 E Ninth St	42000	Technical Staff
3	Dieko	1309 Traditions Way	28000	Worker
4	Donald	1100 S Ocean Blvd, Palm Beach	100000	Quality Controller
5	Dylan	602 Classen Blvd	53000	Worker
6	Ethan	602 Classen Blvd	60000	Quality Controller
7	Hala	1524 Asp Ave	33500	Worker
8	Hunter	2325 Louise Ln	10000	Technical Staff
9	Jackson	1410 Wisteria Ave	20000	Worker
10	Steven	1321 E 24th St	50000	Technical Staff

- Employee Table

Results Messages		
	Technical_Staff_Name ▼	Position ▼
1	Caleb	Machine Worker
2	Hunter	IT
3	Steven	Manager

- Technical Staff Table

Results Messages		
	Name ▼	Degree ▼
1	Caleb	BS
2	Caleb	MS
3	Caleb	PHD
4	Hunter	BS
5	Hunter	MS
6	Steven	BS

Table

- Technical Staff Degree

Results Messages		
	Worker_Name ▼	Max_Products ▼
1	Dieko	20
2	Dylan	37
3	Hala	30
4	Jackson	23

- Worker Table

Results Messages		
	Quality_Controller_Name ▼	Product_Type ▼
1	Ansley	Toys
2	Donald	Flags
3	Ethan	Computers

- Quality Controller Table
-

## 6.2: Query 2

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:54:26 PM) [pid: 15740]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
1
Insert product ID:
1
Insert product creation date:
11/13/2022
Insert number of days it took to develop the product:
2
Insert the name of the worker that produced the product:
Jackson
Insert the name of the quality controller that tested the product:
Ansley
Insert the name of the technician that repaired the product, if applicable (if not, insert 'NULL'):
NULL
Enter the product size:
Medium
Enter the product software:
Java
Execution Complete!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:57:19 PM) [pid: 40232]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
1
Insert product ID:
2
Insert product creation date:
08/25/2022
Insert number of days it took to develop the product:
22
Insert the name of the worker that produced the product:
Dieko
Insert the name of the quality controller that tested the product:
Donald
Insert the name of the technician that repaired the product, if applicable (if not, insert 'NULL'):
Caleb
Enter the product size:
Small
Enter the product software:
C
Execution Complete!

```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:58:10 PM) [pid: 37644]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
1
Insert product ID:
3
Insert product creation date:
04/11/2019
Insert number of days it took to develop the product:
3
Insert the name of the worker that produced the product:
Dieko
Insert the name of the quality controller that tested the product:
Ethan
Insert the name of the technician that repaired the product, if applicable (if not, insert 'NULL'):
Steven
Enter the product size:
Medium
Enter the product software:
Ruby
Execution Complete!
```

```
Console X
Individual Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 8:59:14 PM) [pid: 38236]
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
2
Insert product ID:
4
Insert product creation date:
09/02/2018
Insert number of days it took to develop the product:
13
Insert the name of the worker that produced the product:
Dylan
Insert the name of the quality controller that tested the product:
Ansley
Insert the name of the technician that repaired the product, if applicable:
NULL
Enter the product size:
Large
Enter the product color:
Black
Statement executed!

Inserting into Product 2 table. . .
Execution complete!
```



```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:00:37 PM) [pid: 11784]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
2
Insert product ID:
5
Insert product creation date:
03/21/2020
Insert number of days it took to develop the product:
45
Insert the name of the worker that produced the product:
Dylan
Insert the name of the quality controller that tested the product:
Ansley
Insert the name of the technician that repaired the product, if applicable:
NULL
Enter the product size:
Medium
Enter the product color:
Green
Statement executed!

Inserting into Product 2 table. . .
Execution complete!
```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:02:02 PM) [pid: 38764]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

2

Choose the type of product to insert:

- 1) Product 1
- 2) Product 2
- 3) Product 3

2

Insert product ID:

6

Insert product creation date:

06/09/2018

Insert number of days it took to develop the product:

2

Insert the name of the worker that produced the product:

Hala

Insert the name of the quality controller that tested the product:

Donald

Insert the name of the technician that repaired the product, if applicable:

NULL

Enter the product size:

Small

Enter the product color:

Black

Statement executed!

Inserting into Product 2 table. . .

Execution complete!

```
Console X
Individual Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:03:01 PM) [pid: 36484]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
2
Insert product ID:
7
Insert product creation date:
03/31/2022
Insert number of days it took to develop the product:
54
Insert the name of the worker that produced the product:
Jackson
Insert the name of the quality controller that tested the product:
Ethan
Insert the name of the technician that repaired the product, if applicable:
NULL
Enter the product size:
Large
Enter the product color:
Grey
Statement executed!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:04:04 PM) [pid: 26872]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
3
Insert product ID:
8
Insert product creation date:
02/25/2020
Insert number of days it took to develop the product:
21
Insert the name of the worker that produced the product:
Hala
Insert the name of the quality controller that tested the product:
Ansley
Insert the name of the technician that repaired the product, if applicable:
NULL
Enter the product size:
Medium
Enter the product weight:
20
Statement executed!

```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:05:03 PM) [pid: 16220]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
3
Insert product ID:
9
Insert product creation date:
05/13/2019
Insert number of days it took to develop the product:
3
Insert the name of the worker that produced the product:
Dylan
Insert the name of the quality controller that tested the product:
Donald
Insert the name of the technician that repaired the product, if applicable:
Hunter
Enter the product size:
Large
Enter the product weight:
107.45
Statement executed!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:07:20 PM) [pid: 28164]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
2
Choose the type of product to insert:
1) Product 1
2) Product 2
3) Product 3
3
Insert product ID:
10
Insert product creation date:
07/06/2018
Insert number of days it took to develop the product:
134
Insert the name of the worker that produced the product:
Jackson
Insert the name of the quality controller that tested the product:
Ethan
Insert the name of the technician that repaired the product, if applicable:
Steven
Enter the product size:
Large
Enter the product weight:
1067.45
Statement executed!

Inserting into Product 3 table. . .
Execution complete!

```

Results		Messages							
	ID	Date_Created	Days_Developed	Produced_By	Tested_By	Repaired_By	Size	Product_Type	
1	1	2022-11-13	2	Jackson	Ansley	NULL	Medium	1	
2	2	2022-08-25	22	Dieko	Donald	Caleb	Small	1	
3	3	2019-04-11	3	Dieko	Ethan	Steven	Medium	1	
4	4	2018-09-02	13	Dylan	Ansley	NULL	Large	2	
5	5	2020-03-21	45	Dylan	Ansley	NULL	Medium	2	
6	6	2018-06-09	2	Hala	Donald	NULL	Small	2	
7	7	2022-03-31	54	Jackson	Ethan	NULL	Large	2	
8	8	2020-02-25	21	Hala	Ansley	NULL	Medium	3	
9	9	2019-05-13	3	Dylan	Donald	Hunter	Large	3	
10	10	2018-07-06	134	Jackson	Ethan	Steven	Large	3	

- The Product Table

Results		Messages
	Product1_ID ▼	Software ▼
1	1	Java
2	2	C
3	3	Ruby

- The Product1 Table

Results		Messages
	Product2_ID ▼	Color ▼
1	4	Black
2	6	Black
3	5	Green
4	7	Grey

- The Product2 Table

Results		Messages
	Product3_ID ▼	Weight ▼
1	8	20
2	9	107.45
3	10	1067.45

- The Product3 Table

## 6.3: Query 3

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:22:24 PM) [pid: 25148]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Gene
Insert customer address:
404 Sarasota Drive
How many products has Gene bought?
1
Inserting into Customer table. . .
Execution complete!

Insert product ID:
5
Inserting into Purchase table. . .
Execution complete!
```



```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:23:24 PM) [pid: 38436]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Rebecca
Insert customer address:
2209 Briarcrest Rd
How many products has Rebecca bought?
2
Inserting into Customer table. . .
Execution complete!

Insert product ID:
1
Inserting into Purchase table. . .
Execution complete!

Insert product ID:
7
Inserting into Purchase table. . .
Execution complete!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:24:49 PM) [pid: 7944]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)|
Yes
Insert customer name:
Marcus
Insert customer address:
1416 Crescent Dr
How many products has Marcus bought?
1
Inserting into Customer table. . .
Execution complete!

Insert product ID:
8
Inserting into Purchase table. . .
Execution complete!

```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:26:30 PM) [pid: 42960]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

3

Is this a new customer? (Yes / No)

Yes

Insert customer name:

Leigh

Insert customer address:

1416 Crescent Dr

How many products has Leigh bought?

1

Inserting into Customer table. . .

Execution complete!

Insert product ID:

7

Inserting into Purchase table. . .

Execution complete!

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:28:45 PM) [pid: 31920]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

3

Is this a new customer? (Yes / No)

Yes

Insert customer name:

Tana

Insert customer address:

3108 13th St

How many products has Tana bought?

1

Inserting into Customer table. . .

Execution complete!

Insert product ID:

3

Inserting into Purchase table. . .

Execution complete!

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:29:25 PM) [pid: 37072]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Tyler
Insert customer address:
405 E Elm St
How many products has Tyler bought?
3
Inserting into Customer table. . .
Execution complete!

Insert product ID:
2
Inserting into Purchase table. . .
Execution complete!

Insert product ID:
6
Inserting into Purchase table. . .
Execution complete!

Insert product ID:
9
Inserting into Purchase table. . .
Execution complete!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:30:15 PM) [pid: 39012]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Josh
Insert customer address:
239 Classen St
How many products has Josh bought?
1
Inserting into Customer table. . .
Execution complete!

Insert product ID:
6
Inserting into Purchase table. . .
Execution complete!

```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:31:44 PM) [pid: 35712]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Bethany
Insert customer address:
569 Terrace Heights
How many products has Bethany bought?
2
Inserting into Customer table. . .
Execution complete!

Insert product ID:
4
Inserting into Purchase table. . .
Execution complete!

Insert product ID:
8
Inserting into Purchase table. . .
Execution complete!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:35:34 PM) [pid: 45824]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
Yes
Insert customer name:
Mike
Insert customer address:
2398 W Sapulpa St
How many products has Mike bought?
1
Inserting into Customer table. . .
Execution complete!

Insert product ID:
6
Inserting into Purchase table. . .
Execution complete!
```



```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 9:34:01 PM) [pid: 396]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
3
Is this a new customer? (Yes / No)
No
Insert customer name:
Gene
How many products has Gene bought?
1
Insert product ID:
10
Inserting into Purchase table. . .
Execution complete!
```

Results		Messages
	Name ▾	Address ▾
1	Bethany	569 Terrace Heights
2	Gene	404 Sarrasota Drive
3	Josh	239 Classen St
4	Leigh	1416 Crescent Dr
5	Marcus	1416 Crescent Dr
6	Mike	2398 W Sapulpa St
7	Rebecca	2209 Briarcrest Rd
8	Tana	3108 13th St
9	Tyler	405 E Elm St

- The Customer Table

Results			Messages	
	Customer_Name	Product_ID		
1	Bethany	4		
2	Bethany	8		
3	Gene	5		
4	Gene	10		
5	Josh	6		
6	Leigh	7		
7	Marcus	8		
8	Mike	6		
9	Rebecca	1		
10	Rebecca	7		
11	Tana	3		
12	Tyler	2		
13	Tyler	6		
14	Tyler	9		

Table

- The Purchase

## 6.4: Query 4

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:11:52 PM) [pid: 43360]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
1
Insert account creation date
11/13/2022
Insert product cost:
32.00
Insert product ID
1
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:12:48 PM) [pid: 35864]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
2
Insert account creation date
08/25/2022
Insert product cost:
74.99
Insert product ID
2
Statement executed!
```

```
Console X
Individual_Project [Java Application] [pid: 40496]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
3
Insert account creation date
04/11/2022
Insert product cost:
49.99
Insert product ID
3
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:16:53 PM) [pid: 38864]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
4
Insert account creation date
09/02/2018
Insert product cost:
20
Insert product ID
4
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:17:50 PM) [pid: 33684]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
5
Insert account creation date
03/21/2020
Insert product cost:
14.99
Insert product ID
5
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:18:27 PM) [pid: 35084]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
6
Insert account creation date
06/09/2018
Insert product cost:
9.99
Insert product ID
6
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:19:12 PM) [pid: 13128]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
7
Insert account creation date
03/31/2022
Insert product cost:
24.75
Insert product ID
7
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:19:50 PM) [pid: 37948]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
8
Insert account creation date
02/25/2020
Insert product cost:
250
Insert product ID
8
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:20:33 PM) [pid: 43732]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
9
Insert account creation date
05/13/2019
Insert product cost:
115
Insert product ID
9
Statement executed!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:22:08 PM) [pid: 38020]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
4
Insert account number:
10
Insert account creation date
07/06/2018
Insert product cost:
62.65
Insert product ID
10
Statement executed!

```

Results		Messages	
	Account_Number ▾	Date_Created ▾	Product_ID ▾
1	1	2022-11-13	1
2	2	2022-08-25	2
3	3	2022-04-11	3
4	4	2018-09-02	4
5	5	2020-03-21	5
6	6	2018-06-09	6
7	7	2022-03-31	7
8	8	2020-02-25	8
9	9	2019-05-13	9
10	10	2018-07-06	10

- The Account Table



Results		Messages	
	Account_Number ▼	Product_Cost ▼	Product_ID ▼
1	1	32	1
2	2	74.99	2
3	3	49.99	3

- The Product1 Account Table

Results		Messages	
	Account_Number ▼	Product_Cost ▼	Product_ID ▼
1	4	20	4
2	5	14.99	5
3	6	9.99	6
4	7	24.75	7

- The Product2 Account Table

Results		Messages	
	Account_Number ▼	Product_Cost ▼	Product_ID ▼
1	8	250	8
2	9	115	9
3	10	62.65	10

- The Product3 Account Table

## 6.5: Query 5

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:30:49 PM) [pid: 32228]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
5
Insert Complaint ID:
1
Insert date the complaint was created:
11/13/2022
Insert description of the complaint (250 chars or less):
Faded screen
Insert treatment of complaint:
Screen replacement
Insert customer name:
Rebecca
Insert product ID:
1
Statement executed!

Since we have a complaint, we need to repair it!

Insert repair ID:
1
Insert Technician that will repair the complaint:
Hunter
Insert date that the repair was requested:
11/13/2022
Statement executed!
```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:32:05 PM) [pid: 44916]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
5
Insert Complaint ID:
2
Insert date the complaint was created:
08/25/2022
Insert description of the complaint (250 chars or less):
Broken wheel
Insert treatment of complaint:
Refund
Insert customer name:
Gene
Insert product ID:
10
Statement executed!

Since we have a complaint, we need to repair it!

Insert repair ID:
2
Insert Technician that will repair the complaint:
Steven
Insert date that the repair was requested:
08/25/2022
Statement executed!

```

```

Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (Nov 20, 2022, 10:33:10 PM) [pid: 39612]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
5
Insert Complaint ID:
3
Insert date the complaint was created:
10/20/2019
Insert description of the complaint (250 chars or less):
Faulty wiring
Insert treatment of complaint:
Replacement
Insert customer name:
Bethany
Insert product ID:
4
Statement executed!

Since we have a complaint, we need to repair it!

Insert repair ID:
3
Insert Technician that will repair the complaint:
Caleb
Insert date that the repair was requested:
10/20/2019
Statement executed!

```

Results Messages

	Complaint_ID	Date_Created	Description	Treatment	Customer_Name	Product_ID
1	1	2022-11-13	Faded screen	Screen replacement	Rebecca	1
2	2	2022-08-25	Broken wheel	Refund	Gene	10
3	3	2019-10-20	Faulty wiring	Replacement	Bethany	4

- The Complaint Table

	ID	Technical_Staff_Name	Date_Created	Product_ID	Complaint_ID
1	1	Hunter	2022-11-13	1	1
2	2	Steven	2022-08-25	10	2
3	3	Caleb	2019-10-20	4	3

- The Repair Complaint Table

## 6.6: Query 6

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:37:34 PM) [pid: 27364]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
6
Insert accident number
1
Insert date of accident:
11/13/2022
Insert number of work days lost:
2
Insert product ID:
2
Insert the Technical Staff or Quality Controller that had the accident:
Hunter
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:39:45 PM) [pid: 32992]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
6
Insert accident number
2
Insert date of accident:
02/03/2022
Insert number of work days lost:
10
Insert product ID:
5
Insert the Technical Staff or Worker that had the accident:
Jackson
Statement executed!
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:40:28 PM) [pid: 22808]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
6
Insert accident number
3
Insert date of accident:
04/22/2019
Insert number of work days lost:
3
Insert product ID:
10
Insert the Technical Staff or Worker that had the accident:
Dylan
Statement executed!
```

Results		Messages				
	Accident_Number	Date_Created	Work_Days_Lost	Product_ID	Employee_Name	Employee_Type
1	1	2022-11-13	2	2	Hunter	Technical Staff
2	2	2022-02-03	10	5	Jackson	Worker
3	3	2019-04-22	3	10	Dylan	Worker

- The Accident Table

## 6.7: Query 7

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:43:02 PM) [pid: 37100]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
7
Insert product ID
3
Contents of the Product table:
Date Created | Days Developed
2019-04-11 | 3
```



Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:43:47 PM) [pid: 46736]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

7

Insert product ID

10

Contents of the Product table:

Date Created | Days Developed

2018-07-06 | 134

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:44:14 PM) [pid: 2608]

```
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
```

7

Insert product ID

1

Contents of the Product table:

Date Created | Days Developed

2022-11-13 | 2

## 6.8: Query 8

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:46:09 PM) [pid: 38212]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
8
Insert Worker name:
Jackson
Contents of the Product table:
| Product ID |
| 1 |
| 7 |
| 10 |
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:46:45 PM) [pid: 41092]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
8
Insert Worker name:
Dylan
Contents of the Product table:
| Product ID |
| 4 |
| 5 |
| 9 |
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:47:05 PM) [pid: 24392]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
8
Insert Worker name:
Hala
Contents of the Product table:
| Product ID |
| 6 |
| 8 |
```

## 6.9: Query 9

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (Nov 20, 2022, 10:49:20 PM) [pid: 37488]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
9
Insert quality controller name:
Ansley
Contents of the Product table:
| Number of Errors |
| 1 |
```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:50:11 PM) [pid: 40788]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

9

Insert quality controller name:

Donald

Contents of the Product table:

| Number of Errors |

| 2 |

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

9

Insert quality controller name:

Ethan

Contents of the Product table:

Number of Errors
2

| 2 |



## 6.10: Query 10

```
Console X
Individual Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:56:03 PM) [pid: 38724]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
10
Insert Quality Controller:
Ansley
Contents of the table:
| Total costs |
| 250.0 |
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:56:29 PM) [pid: 33184]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
10
Insert Quality Controller:
Donald
Contents of the table:
| Total costs |
| 115.0 |
```

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:56:48 PM) [pid: 18412]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

10

Insert Quality Controller:

Ethan

Contents of the table:

Total costs
62.65

## 6.11: Query 11

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:58:49 PM) [pid: 17412]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
11
Insert color:
Green
Contents of the Customer table:
| Customer |
| Gene |
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:59:08 PM) [pid: 41200]
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
11
Insert color:
Black
Contents of the Customer table:
| Customer |
| Bethany |
| Josh |
| Mike |
| Tyler |
```

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 10:59:29 PM) [pid: 45696]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
11
Insert color:
Grey
Contents of the Customer table:
| Customer |
| Leigh |
| Rebecca |
```

## 6.12: Query 12

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:01:34 PM) [pid: 21532]
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
12
Enter salary:
40000
Contents of the Employee table:
| Employees |
| Caleb | 918 E Ninth St | 42000.0 | Technical Staff |
| Donald | 1100 S Ocean Blvd, Palm Beach | 100000.0 | Quality Controller |
| Dylan | 602 Classen Blvd | 53000.0 | Worker |
| Ethan | 602 Classen Blvd | 60000.0 | Quality Controller |
| Steven | 1321 E 24th St | 50000.0 | Technical Staff |
```

## 6.13: Query 13

Console X

Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:04:59 PM) [pid: 33404]

Welcome to Hunter's Warehouse!

Please select one of the options below:

- 1) Insert a new employee
- 2) Insert a new product
- 3) Insert a customer that has purchased one or more products
- 4) Insert a new account associated with a product
- 5) Enter a complaint associated with a customer and product
- 6) Log an accident between an employee and product
- 7) Retrieve the date produced and time spent to produce a particular product
- 8) Retrieve all products made by a particular worker
- 9) Retrieve the total number of errors a particular quality controller made
- 10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

13

Enter Employee name:

Hunter

Contents of the Employee table:

Employees
2



## 6.14: Query 14

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:06:17 PM) [pid: 26536]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
14
Insert year:
2022
Contents of the Total Cost table:
| Average Cost |
| 43.91 |
```

## 6.15: Query 15

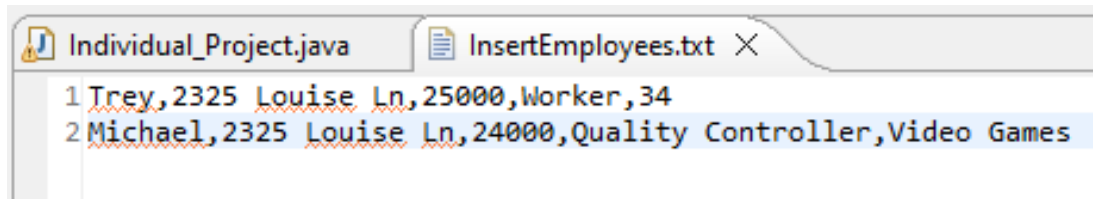
```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:07:47 PM) [pid: 5420]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
15
Insert starting date:
01/01/2022
Insert ending date:
12/31/2022
Initial Accident table:
Contents of the Accident table before deletion:
| Accident Number | Date Created | Work Days Lost | Product ID | Employee Name | Employee Type |
| 1 | 2022-11-13 | 2 | 2 | Hunter | Technical Staff |
| 2 | 2022-02-03 | 10 | 5 | Jackson | Worker |
| 3 | 2019-04-22 | 3 | 10 | Dylan | Worker |
Contents of the Accident table after deletion:
| Accident Number | Date Created | Work Days Lost | Product ID | Employee Name | Employee Type |
| 3 | 2019-04-22 | 3 | 10 | Dylan | Worker |
```

## 6.16: Query 16

```
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:11:37 PM) [pid: 42144]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
16
Insert file name:
InsertEmployees.txt
Inserting into Employee table. . .
Execution complete!

Inserting into Worker table. . .
Execution complete!
Inserting into Employee table. . .
Execution complete!

Inserting into Quality Controller table. . .
Execution complete!
```



```
Individual_Project.java InsertEmployees.txt X
1 Trey, 2325 Louise Ln, 25000, Worker, 34
2 Michael, 2325 Louise Ln, 24000, Quality Controller, Video Games
```

**Results**    Messages

	Employee_Name ▾	Address ▾	Salary ▾	Employee_Type ▾
1	Ansley	1410 Wisteria Ave	30000	Quality Controller
2	Caleb	918 E Ninth St	42000	Technical Staff
3	Dieko	1309 Traditions Way	28000	Worker
4	Donald	1100 S Ocean Blvd, Palm Beach	100000	Quality Controller
5	Dylan	602 Classen Blvd	53000	Worker
6	Ethan	602 Classen Blvd	60000	Quality Controller
7	Hala	1524 Asp Ave	33500	Worker
8	Hunter	2325 Louise Ln	10000	Technical Staff
9	Jackson	1410 Wisteria Ave	20000	Worker
10	Michael	2325 Louise Ln	24000	Quality Controller
11	Steven	1321 E 24th St	50000	Technical Staff
12	Trey	2325 Louise Ln	25000	Worker

## 6.17: Query 17

```
Console X
Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:14:25 PM) [pid: 33016]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
17
Insert file name:
GetCustomers.txt
Insert color:
Black
```

```
Individual_Project.java  InsertEmployees.txt  GetCustomers.txt X
1 Bethany
2 Josh
3 Mike
4 Tyler
5 |
```

## 6.18: Query 18

```
Console X
<terminated> Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (Nov 20, 2022, 11:16:41 PM – 11:16:43 PM) [pid: 5684]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
18
Exiting! Goodbye!
```

## 6.19: Error Handling

```
Console X
<terminated> Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (Nov 20, 2022, 11:18:35 PM - 11:18:52 PM) [pid: 39348]
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
1
Choose the type of employee to insert:
1) Technical Staff
2) Worker
3) Quality Controller
1
Insert Technician name:
Hunter
Insert Technician address:
2325 Louise Ln
Insert Technician salary:
23903
Insert Technician position:
IT
Inserting into Employee table. . .
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Employee_C089098D46138015'. Cannot insert duplicate key in object 'dbo.Employee'. The duplicate key value is (Hunter).
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.TDSTokenHandler.onEOF(tdsparser.java:300)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.TDSParser.parse(tdsparser.java:133)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1669)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.doExecuteStatement(SQLServerStatement.java:987)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement$StatExecCmd.doExecute(SQLServerStatement.java:802)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7627)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3912)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:268)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:242)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.execute(SQLServerStatement.java:775)
    at Individual_Project.main(Individual_Project.java:85)
```

<terminated> Individual\_Project [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (Nov 20, 2022, 11:19:28 PM – 11:20:19 PM) [pid: 37332]

- 11) Retrieve all customers in name order who purchased all products of a particular color
- 12) Retrieve all employees whose salary is above a particular salary
- 13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
- 14) Retrieve the average cost of all products made in a particular year
- 15) Delete all accidents whose dates are in some range
- 16) Import: Enter new employees from a data file until the file is empty
- 17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
- 18) Exit!

5  
Insert Complaint ID:  
4  
Insert date the complaint was created:  
11/20/2022  
Insert description of the complaint (250 chars or less):

Bad connection

Insert treatment of complaint:

Refund

Insert customer name:

Rebecca

Insert product ID:

10

Exception in thread "main" [com.microsoft.sqlserver.jdbc.SQLServerException](#): Rebecca has not bought a product with the ID of 10  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError([SQLServerException.java:265](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult([SQLServerStatement.java:1673](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement([SQLServerPreparedStatement.java:620](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement\$PrepStmtExecCmd.doExecute([SQLServerPreparedStatement.java:540](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.TDSCommand.execute([IOBuffer.java:7627](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand([SQLServerConnection.java:3912](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand([SQLServerStatement.java:268](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement([SQLServerStatement.java:242](#))  
at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute([SQLServerPreparedStatement.java:518](#))  
at Individual\_Project.main([Individual\\_Project.java:558](#))



```

Console X
<terminated> Individual_Project [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe (Nov 20, 2022, 11:20:59 PM - 11:21:51 PM) [pid: 40032]
Welcome to Hunter's Warehouse!
Please select one of the options below:
1) Insert a new employee
2) Insert a new product
3) Insert a customer that has purchased one or more products
4) Insert a new account associated with a product
5) Enter a complaint associated with a customer and product
6) Log an accident between an employee and product
7) Retrieve the date produced and time spent to produce a particular product
8) Retrieve all products made by a particular worker
9) Retrieve the total number of errors a particular quality controller made
10) Retrieve the total costs of the products in the Product 3 category which were repaired at the request of a particular quality controller
11) Retrieve all customers in name order who purchased all products of a particular color
12) Retrieve all employees whose salary is above a particular salary
13) Retrieve the total number of workdays lost due to accidents in repairing the products which got complaints
14) Retrieve the average cost of all products made in a particular year
15) Delete all accidents whose dates are in some range
16) Import: Enter new employees from a data file until the file is empty
17) Export: Retrieve all customers in name order who purchased all products of a particular color and output them to a data file
18) Exit!
6
Insert accident number
4
Insert date of accident:
11/20/2022
Insert number of work days lost:
10
Insert product ID:
2
Insert the Technical Staff or Worker that had the accident:
Ethan
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT statement conflicted with the CHECK constraint "chk_Employee_Type". The conflict occurred in database "cs-dsa-4513-sql-db", table "dbo.Accident", column 'Employee_Type'.
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:265)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1673)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:620)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPreparedStatement.java:540)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.TDSCCommand.execute(IOBuffer.java:7627)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3912)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:268)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:242)
    at com.microsoft.sqlserver.jdbc@11.2.0.jre17/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:518)
    at Individual_Project.main(Individual_Project.java:622)

```

## Task 7

### 7.1 Source Code and Compilation

DataHandler.java

```
package jsp_azure_test;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class DataHandler {
    private Connection conn;
    // Azure SQL connection credentials
    private String server = "devo0008-sql-server.database.windows.net";
    private String database = "cs-dsa-4513-sql-db";
    private String username = "devo0008";
    private String password = "n0mathwords1!";
    // Resulting connection string
    final private String url =
        String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=
%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.database.win
dows.net;loginTimeout=30;", server, database, username, password);
    // Initialize and save the database connection
    private void getDBConnection() throws SQLException {
        if (conn != null) {
            return;
        }
        this.conn = DriverManager.getConnection(url);
    }
    // Return the result of selecting everything from the Employee table
    public ResultSet getAllEmployees() throws SQLException {
        getDBConnection();

        final String sqlQuery = "SELECT * FROM Employee;";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }
    // Inserts a record into the movie_night table with the given attribute
    values
    public boolean add_employee(String Employee_Name, String Address, int Salary,
String Employee_Type) throws SQLException {
        getDBConnection(); // Prepare the database connection
```

```

        // Prepare the SQL statement
        final String sqlQuery = "INSERT INTO EMPLOYEE(Employee_Name, Address,
Salary, Employee_Type) VALUES(?, ?, ?, ?)";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        // Replace the '?' in the above statement with the given attribute values
        stmt.setString(1, Employee_Name);
        stmt.setString(2, Address);
        stmt.setInt(3, Salary);
        stmt.setString(4, Employee_Type);

        // Execute the query, if only one record is updated, then we indicate
        success by returning true
        return stmt.executeUpdate() == 1;
    }

    public ResultSet getAllEmployeesBasedOnSalary(int Salary) throws SQLException
    {
        getDBConnection();

        final String sqlQuery = "SELECT * FROM Employee WHERE Salary > '" +
Salary + "'";
        final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
        return stmt.executeQuery();
    }
}

```

add\_employee\_form

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Add Employee</title>
    </head>
    <body>
        <h2>Add Employee</h2>
        <!--
            Form for collecting user input for the new movie_night record.
            Upon form submission, add_movie.jsp file will be invoked.
        -->
        <form action="add_employee.jsp">
            <!-- The form organized in an HTML table for better clarity. -->
            <table border=1>

```

```

        <tr>
            <th colspan="2">Enter the Employee Data:</th>
        </tr>
        <tr>
            <td>Employee Name:</td>
            <td><div style="text-align: center;">
                <input type="text" name=Employee_Name>
            </div></td>
        </tr>
        <tr>
            <td>Address:</td>
            <td><div style="text-align: center;">
                <input type="text" name=Address>
            </div></td>
        </tr>
        <tr>
            <td>Salary:</td>
            <td><div style="text-align: center;">
                <input type="text" name=Salary>
            </div></td>
        </tr>
        <tr>
            <td>Employee Type:</td>
            <td><div style="text-align: center;">
                <input type="text" name=Employee_Type>
            </div></td>
        </tr>
        <tr>
            <td><div style="text-align: center;">
                <input type="reset" value=Clear>
            </div></td>
            <td><div style="text-align: center;">
                <input type="submit" value=Insert>
            </div></td>
        </tr>
    </table>
</form>
</body>
</html>

```

add\_employee

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

// Get the attribute values passed from the input form.
String Employee_Name = request.getParameter("Employee_Name");
String Address = request.getParameter("Address");
String Salary_Temp = request.getParameter("Salary");
String Employee_Type = request.getParameter("Employee_Type");

if(Employee_Name.equals("") || Address.equals("") || Salary_Temp.equals("")
|| Employee_Type.equals("")) {
    response.sendRedirect("add_employee_form.jsp");
}

else {

    int Salary = Integer.parseInt(Salary_Temp);

    // Now perform the query with the data from the form.
    boolean success = handler.add_employee(Employee_Name, Address, Salary,
Employee_Type);
    if (!success) { // Something went wrong
        %>
        <h2>There was a problem inserting the course</h2>
        <%
    }

    else { // Confirm success to the user
        %>
```

```

        <h2>Employee:</h2>

        <ul>
            <li>Employee Name: <%=Employee_Name%></li>
            <li>Address: <%=Address%></li>
            <li>Salary: <%=Salary%></li>
            <li>Employee Type: <%=Employee_Type%></li>
        </ul>

        <h2>Was successfully inserted.</h2>

        <a href="get_all_employees.jsp"> See all Employees</a>
        <%
    }
}
%>
</body>
</html>

```

get\_all\_employees

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
    <title>Employees</title>
</head>
<body>
    <%@page import="jsp_azure_test.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
        // We instantiate the data handler here, and get all the movies from the
        database
        final DataHandler handler = new DataHandler();
        final ResultSet Employees = handler.getAllEmployees();
    %>
    <!-- The table for displaying all the movie records -->
    <table cellpadding="2" cellspacing="2" border="1">
        <tr> <!-- The table headers row -->
            <td align="center">
                <h4>Employee Name</h4>
            </td>

```

```

        <td align="center">
            <h4>Address</h4>
        </td>
        <td align="center">
            <h4>Salary</h4>
        </td>
        <td align="center">
            <h4>Employee Type</h4>
        </td>
    </tr>
    <%
        while(Employees.next()) { // For each movie_night record returned...
            // Extract the attribute values for every row returned
            final String Employee_Name = Employees.getString("Employee_Name");
            final String Address = Employees.getString("Address");
            final String Salary = Employees.getString("Salary");
            final String Employee_Type = Employees.getString("Employee_Type");

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

```

get\_all\_employees\_based\_on\_salary\_form

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Add Movie Night</title>
    </head>
    <body>
        <h2>Get All Employees Above a Salary</h2>
        <!--
            Form for collecting user input for the new movie_night record.
            Upon form submission, add_movie.jsp file will be invoked.
        -->
    </body>
</html>

```

```

-->
<form action="get_all_employees_based_on_salary.jsp">
  <!-- The form organized in an HTML table for better clarity. -->
  <table border=1>
    <tr>
      <th colspan="2">Enter a Salary:</th>
    </tr>
    <tr>
      <td>Salary:</td>
      <td><div style="text-align: center;">
        <input type=text name=Salary>
      </div></td>
    </tr>
    <tr>
      <td><div style="text-align: center;">
        <input type=reset value=Clear>
      </div></td>
      <td><div style="text-align: center;">
        <input type=submit value=Insert>
      </div></td>
    </tr>
  </table>
</form>
</body>
</html>

```

get\_all\_employees\_based\_on\_salary

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Employees</title>
  </head>
  <body>
    <%@page import="jsp_azure_test.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
      String Salary_Temp = request.getParameter("Salary");

      if(Salary_Temp.equals("")) {

```



```

        response.sendRedirect("get_all_employees_based_on_salary_form.jsp");
    }

    int SalaryInteger = Integer.parseInt(Salary_Temp);
    // We instantiate the data handler here, and get all the movies from
the database
    final DataHandler handler = new DataHandler();
    final ResultSet Employees =
handler.getAllEmployeesBasedOnSalary(SalaryInteger);
    %>
    <!-- The table for displaying all the movie records -->
    <table cellpadding="2" cellspacing="2" border="1">
        <tr> <!-- The table headers row -->
            <td align="center">
                <h4>Employee Name</h4>
            </td>
            <td align="center">
                <h4>Address</h4>
            </td>
            <td align="center">
                <h4>Salary</h4>
            </td>
            <td align="center">
                <h4>Employee Type</h4>
            </td>
        </tr>
    <%
        while(Employees.next()) { // For each movie_night record
returned...
            // Extract the attribute values for every row returned
            final String Employee_Name =
Employees.getString("Employee_Name");
            final String Address = Employees.getString("Address");
            final String Salary = Employees.getString("Salary");
            final String Employee_Type =
Employees.getString("Employee_Type");

            out.println("<tr>"); // Start printing out the new table row
            out.println( // Print each attribute value
                "<td align=\"center\">" + Employee_Name +
                "</td><td align=\"center\"> " + Address +
                "</td><td align=\"center\"> " + Salary +
                "</td><td align=\"center\"> " + Employee_Type +
"</td>");

```

```

        out.println("</tr>");
    }
    %>
</table>
</body>
</html>

```

The screenshot shows an IDE with two main panels. The left panel displays the project structure for 'DeVoe\_Hunter\_IP\_Task7'. The right panel shows a web browser at the URL 'http://localhost:8585/DeVoe\_Hunter\_IP\_Task7/add\_employee\_form.jsp' displaying an 'Add Employee' form.

**Project Structure (Left Panel):**

- DeVoe\_Hunter\_IP\_Task7
  - Deployment Descriptor: DeVoe\_Hunter\_IP\_Task7
  - JAX-WS Web Services
  - src/main/java
  - JRE System Library [JavaSE-11]
  - Apache Tomcat v10.0 [Apache Tomcat v10.0]
  - Referenced Libraries
  - build
  - src
    - main
      - java
      - webapp
        - META-INF
        - WEB-INF
          - add\_employee\_form.jsp
          - add\_employee.jsp
          - get\_all\_employees\_based\_on\_salary\_form.jsp
          - get\_all\_employees\_based\_on\_salary.jsp
          - get\_all\_employees.jsp
- Servers
  - Tomcat v10.0 Server at localhost-config
    - catalina.policy
    - catalina.properties
    - context.xml
    - server.xml
    - tomcat-users.xml
    - web.xml

http://localhost:8585/DeVoe\_Hunter\_IP\_Task7/add\_employee\_form.jsp

### Add Employee

Enter the Employee Data:	
Employee Name:	<input type="text"/>
Address:	<input type="text"/>
Salary:	<input type="text"/>
Employee Type:	<input type="text"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

Markers Properties Servers Data Source Explorer Snippets Console

> Tomcat v10.0 Server at localhost [Started, Synchronized]

## 7.2: Testing Web-Based Application

DataHandler.java get\_all\_employees.jsp get\_all\_e

http://localhost:8585/DeVoe\_Hunter\_IP\_Task7/ge

### Get All Employees Above a Salary

Enter a Salary:	
Salary:	<input type="text" value="25000"/>
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

DataHandler.java get\_all\_employees.jsp get\_all\_employees\_based\_on\_salary.jsp

http://localhost:8585/DeVoe\_Hunter\_IP\_Task7/get\_all\_employees\_based\_on\_salary.jsp?S

Employee Name	Address	Salary	Employee Type
Ansley	1410 Wisteria Ave	30000.0	Quality Controller
Caleb	918 E Ninth St	42000.0	Technical Staff
Dieko	1309 Traditions Way	28000.0	Worker
Donald	1100 S Ocean Blvd, Palm Beach	100000.0	Quality Controller
Dylan	602 Classen Blvd	53000.0	Worker
Ethan	602 Classen Blvd	60000.0	Quality Controller
Hala	1524 Asp Ave	33500.0	Worker
Steven	1321 E 24th St	50000.0	Technical Staff

## Add Employee

Enter the Employee Data:	
Employee Name:	<input type="text" value="Kyle"/>
Address:	<input type="text" value="1410 Wisteria Ave"/>
Salary:	<input type="text" value="30000"/>
Employee Type:	<input type="text" value="Worker"/> ×
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

⏪ ⏩ 🛑 💰 http://localhost:8585/DeVoe\_Hunter\_IP\_Ta

### Employee:

- Employee Name: Kyle
- Address: 1410 Wisteria Ave
- Salary: 30000
- Employee Type: Worker

**Was successfully inserted.**

[See all Employees](#)

Employee Name	Address	Salary	Employee Type
Ansley	1410 Wisteria Ave	30000.0	Quality Controller
Caleb	918 E Ninth St	42000.0	Technical Staff
Dieko	1309 Traditions Way	28000.0	Worker
Donald	1100 S Ocean Blvd, Palm Beach	100000.0	Quality Controller
Dylan	602 Classen Blvd	53000.0	Worker
Ethan	602 Classen Blvd	60000.0	Quality Controller
Hala	1524 Asp Ave	33500.0	Worker
Hunter	2325 Louise Ln	10000.0	Technical Staff
Jackson	1410 Wisteria Ave	20000.0	Worker
Kyle	1410 Wisteria Ave	30000.0	Worker
Michael	2325 Louise Ln	24000.0	Quality Controller
Steven	1321 E 24th St	50000.0	Technical Staff
Trey	2325 Louise Ln	25000.0	Worker

## Get All Employees Above a Salary

<b>Enter a Salary:</b>	
Salary:	<input type="text" value="25000"/> ×
<input type="button" value="Clear"/>	<input type="button" value="Insert"/>

<b>Employee Name</b>	<b>Address</b>	<b>Salary</b>	<b>Employee Type</b>
Ansley	1410 Wisteria Ave	30000.0	Quality Controller
Caleb	918 E Ninth St	42000.0	Technical Staff
Dieko	1309 Traditions Way	28000.0	Worker
Donald	1100 S Ocean Blvd, Palm Beach	100000.0	Quality Controller
Dylan	602 Classen Blvd	53000.0	Worker
Ethan	602 Classen Blvd	60000.0	Quality Controller
Hala	1524 Asp Ave	33500.0	Worker
Kyle	1410 Wisteria Ave	30000.0	Worker
Steven	1321 E 24th St	50000.0	Technical Staff