

Student's Full Name: Ajibola Olayinka Dedenuola

Course Title: Data Warehousing and Analytics in the Cloud

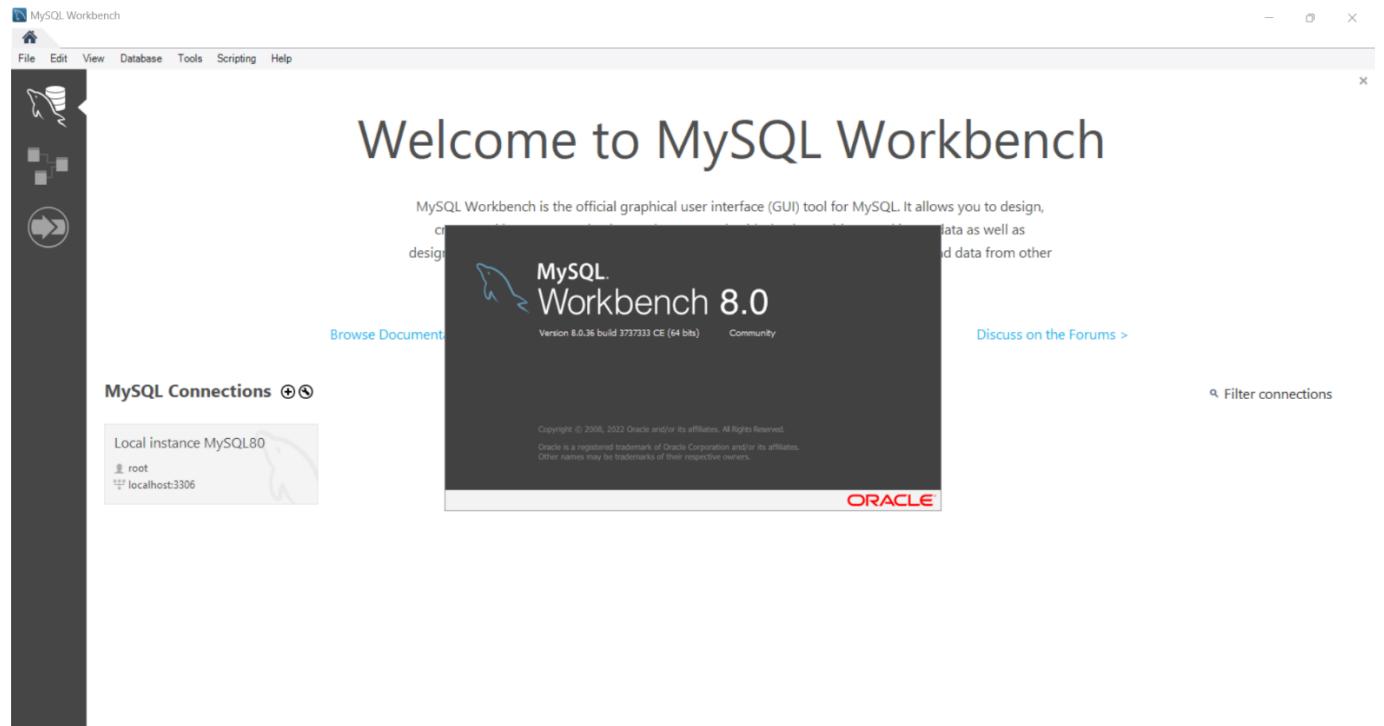
Term name and year: Fall 2023

Submission Week: Assessment (Week 3)

Instructor's Name: Dr. Nayem Rahman

Date of Submission: 2024-05-02

Q1. Show the screenshot of a successful installation of MySQL Software and MySQL Workbench with the latest version on your machine. Show the screenshot of the database "Worker" created.



The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, the 'worker' schema is selected. In the Query1 editor, the following DDL statements are run:

```

1 -- Question 1:
2 • CREATE DATABASE Worker;
3
4 • SHOW DATABASES;
5
6 -- Question 2
7 • CREATE TABLE Worker.Department (
8     DepartmentID TINYINT NOT NULL,
9     DepartmentNm CHAR(30) NOT NULL,
10    PRIMARY KEY (DepartmentID)
11 );
12

```

The Result Grid shows the output of the SHOW DATABASES command:

Database
information_schema
mysql
performance_schema
sys
worker

In the Result 1 editor, the output of the SHOW DATABASES command is displayed:

#	Time	Action
1	00:37:35	SHOW DATABASES

Message: 5 row(s) returned.

Q2. Create the Department table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the Department table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, the 'worker' schema is selected. In the Query1 editor, the following DDL statements are run:

```

6 -- Question 2
7 • CREATE TABLE Worker.Department (
8     DepartmentID TINYINT NOT NULL,
9     DepartmentNm CHAR(30) NOT NULL,
10    PRIMARY KEY (DepartmentID)
11 );
12
13 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (1, 'Research & Development');
14 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (2, 'Production');
15 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (3, 'IT Support');
16 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (4, 'Operations');
17 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (5, 'Customer Service');
18 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (6, 'Purchasing');
19 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (7, 'Sales & Marketing');
20 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (8, 'Human Resource Management');
21 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (9, 'Accounting and Finance');
22 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (10, 'Legal');
23

```

The screenshot shows the SQL Server Management Studio interface. In the left pane, the Navigator displays the 'SCHEMAS' section with 'sys' and 'worker' expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' node under 'worker' is selected. The main pane, titled 'Query 1', contains the following DDL code:

```

15 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (3, 'IT Support');
16 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (4, 'Operations');
17 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (5, 'Customer Service');
18 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (6, 'Purchasing');
19 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (7, 'Sales & Marketing');
20 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (8, 'Human Resource Management');
21 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (9, 'Accounting and Finance');
22 • INSERT INTO Worker.Department (DepartmentID, DepartmentNm) VALUES (10, 'Legal');

23
24 • SELECT * FROM Worker.Department ORDER BY DepartmentID;
25
26 -- Question 3
27 • CREATE TABLE Worker.Employee (
28     EmployeeID INTEGER NOT NULL,
29     DepartmentID TINYINT NOT NULL,

```

Below the code, the 'Result Grid' shows the data inserted into the 'Worker.Department' table:

DepartmentID	DepartmentNm
1	Research & Development
2	Production
3	IT Support
4	Operations
5	Customer Service
6	Purchasing
7	Sales & Marketing
8	Human Resource Management
9	Accounting and Finance
10	Legal
*	NULL

Q3. Create the Employee table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type and length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented (not in a graphical view). (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the Employee table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

The screenshot shows the SQL Server Management Studio interface. In the left pane, the Navigator displays the 'SCHEMAS' section with 'sys' and 'worker' expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' node under 'worker' is selected. The main pane, titled 'Query 1', contains the following DDL code:

```

25
26 -- Question 3
27 • CREATE TABLE Worker.Employee (
28     EmployeeID INTEGER NOT NULL,
29     DepartmentID TINYINT NOT NULL,
30     FirstName VARCHAR(20) NOT NULL,
31     LastName VARCHAR(20) NOT NULL,
32     Address VARCHAR(50),
33     PhoneNumberTxt VARCHAR(15),
34     HireDate DATE NOT NULL,
35     PRIMARY KEY (EmployeeID),
36     FOREIGN KEY (DepartmentID) REFERENCES Worker.Department(DepartmentID)
37 );

```

The screenshot shows a database interface with a left sidebar labeled 'Navigator' containing 'SCHEMAS' and a tree view for the 'worker' schema with 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The main area is titled 'Query 1' and contains the following SQL code:

```
35     PRIMARY KEY (EmployeeID),
36     FOREIGN KEY (DepartmentID) REFERENCES Worker.Department(DepartmentID)
37   };
38
39 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
40   VALUES (1, 2, 'Andy', 'Wong', '345 South Street', '(603) 555-6880', '2001-01-15');
41
42 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
43   VALUES (3, 3, 'Vivek', 'Pandey', '15 Mineral Drive', '(603) 555-4420', '2003-11-15');
44
45 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
46   VALUES (5, 8, 'Kathy', 'Cooper', '15 Hatter Drive', '(212) 555-9630', '2011-11-18');
47
48 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
49   VALUES (2, 1, 'John', 'Wilson', '560 Broadway', '(518) 555-6690', '2017-03-19');
50
51 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
52   VALUES (4, 7, 'Nola', 'Davis', '15 Long Ave', '(478) 555-8822', '2016-03-23');
53
54 • INSERT INTO Worker.Employee (EmployeeID, DepartmentID, FirstName, LastName, Address, PhoneNumberTxt, HireDate)
55   VALUES (6, 9, 'Tom', 'Harper', '64 Highland Street', '(212) 555-7755', '2010-04-11');
```

Q4. Create the Equipment table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the Equipment table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

```

58
59      -- Question 4
60 • CREATE TABLE Worker.Equipment (
61     EquipmentID INTEGER NOT NULL,
62     EquipmentName VARCHAR(30) NOT NULL,
63     EquipmentCostAmount DECIMAL(13,2),
64     PRIMARY KEY (EquipmentID)
65 );
66
67 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
68     VALUES (1, 'Notebook Computers', NULL);
69
70 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
71     VALUES (2, 'Headsets', NULL);
72
73 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
74     VALUES (3, 'Computer Monitor', NULL);
75

```

```

60 • CREATE TABLE Worker.Equipment (
61     EquipmentID INTEGER NOT NULL,
62     EquipmentName VARCHAR(30) NOT NULL,
63     EquipmentCostAmount DECIMAL(13,2),
64     PRIMARY KEY (EquipmentID)
65 );
66
67 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
68     VALUES (1, 'Notebook Computers', NULL);
69
70 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
71     VALUES (2, 'Headsets', NULL);
72
73 • INSERT INTO Worker.Equipment (EquipmentID, EquipmentName, EquipmentCostAmount)
74     VALUES (3, 'Computer Monitor', NULL);
75
76 • SELECT * FROM Worker.Equipment ORDER BY EquipmentID;
77

```

EquipmentID	EquipmentName	EquipmentCostAmount
1	Notebook Computers	NULL
2	Headsets	NULL
3	Computer Monitor	NULL
NULL	NULL	NULL

Q5. Create the EmployeeEquipment table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the EmployeeEquipment table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

```

    Navigator: Schemas
    SCHEMAS
    Filter objects
    ▶ sys
    ▶ worker
        Tables
        Views
        Stored Procedures
        Functions

    Query 1
    -- Question 5
    79 • CREATE TABLE Worker.EmployeeEquipment (
        EmployeeID INTEGER NOT NULL,
        EquipmentID INTEGER NOT NULL,
        PRIMARY KEY (EmployeeID, EquipmentID),
        FOREIGN KEY (EmployeeID) REFERENCES Worker.Employee(EmployeeID),
        FOREIGN KEY (EquipmentID) REFERENCES Worker.Equipment(EquipmentID)
    );
    86
    87 • SHOW CREATE TABLE Worker.EmployeeEquipment;
    88
    89 -- Equipment for Andy Wong
    90 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (1, 1); -- Notebook Computers
    91
    92 -- Equipment for Vivek Pandey
    93 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 1); -- Notebook Computers
    94 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 2); -- Headsets
    95 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 3); -- Computer Monitor
  
```

```

    Navigator: Schemas
    SCHEMAS
    Filter objects
    ▶ sys
    ▶ worker
        Tables
        Views
        Stored Procedures
        Functions

    Administration Schemas Information
    Query 1
    88
    89 -- Equipment for Andy Wong
    90 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (1, 1); -- Notebook Computers
    91
    92 -- Equipment for Vivek Pandey
    93 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 1); -- Notebook Computers
    94 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 2); -- Headsets
    95 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 3); -- Computer Monitor
    96
    97 -- Equipment for Kathy Cooper
    98 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 1); -- Notebook Computers
    99 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 2); -- Headsets
    100 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 3); -- Computer Monitor
    101
    102 -- Equipment for John Wilson
    103 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (2, 1); -- Notebook Computers
    104 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (2, 3); -- Computer Monitor
    105
    106 -- Equipment for Nola Davis
    107 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (4, 1); -- Notebook Computers
    108 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (4, 2); -- Headsets
    109
  
```

Navigator:

SCHEMAS

- Filter objects
- sys
- worker**
 - Tables
 - Views
 - Stored Procedures
 - Functions

Administration Schemas Information

Query 1

```

94 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 2); -- Headsets
95 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (3, 3); -- Computer Monitor
96
97 -- Equipment for Kathy Cooper
98 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 1); -- Notebook Computers
99 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 2); -- Headsets
100 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (5, 3); -- Computer Monitor
101
102 -- Equipment for John Wilson
103 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (2, 1); -- Notebook Computers
104 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (2, 3); -- Computer Monitor
105
106 -- Equipment for Nola Davis
107 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (4, 1); -- Notebook Computers
108 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (4, 2); -- Headsets
109
110 -- Equipment for Tom Harper
111 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (6, 1); -- Notebook Computers
112 • INSERT INTO Worker.EmployeeEquipment (EmployeeID, EquipmentID) VALUES (6, 3); -- Computer Monitor
113
114 • SELECT * FROM Worker.EmployeeEquipment ORDER BY EmployeeID, EquipmentID;
115

```

SCHEMAS

- Filter objects
- sys
- worker**
 - Tables
 - Views
 - Stored Procedures
 - Functions

Administration Schemas Information

No object selected

Result Grid

EmployeeID	EquipmentID
1	1
2	1
2	3
3	1
3	2
3	3
4	1
4	2
5	1
5	2
5	3
6	1
6	3
*	NULL
*	NULL

Q6. Create the Training table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the Training table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected. The right pane is a query window titled 'Query 1' containing the following DDL:

```

116 -- Question 6
117 • CREATE TABLE Worker.Training (
118     TrainingID INTEGER NOT NULL,
119     TrainingName VARCHAR(50) NOT NULL,
120     PRIMARY KEY (TrainingID)
121 );
122
123 • SHOW CREATE TABLE Worker.Training;
124
125 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (1, 'COVID-19 Awareness and Protection Training');

```

The result grid shows the created table structure:

Table	Create Table
Training	CREATE TABLE 'training' ('TrainingID' int NOT NULL, 'TrainingName' varchar(50) NOT NULL, PRIMARY KEY ('TrainingID')) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4...

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected. The right pane is a query window titled 'Query 1' containing the following DML statements:

```

122
123 • SHOW CREATE TABLE Worker.Training;
124
125 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (1, 'COVID-19 Awareness and Protection Training');
126 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (2, 'Code of Conduct Training');
127 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (3, 'Safety Training');
128 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (4, 'Intro to Python');
129 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (5, 'Machine Learning');
130 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (6, 'Microsoft Certifications');
131 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (7, 'Security and Privacy');
132 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (8, 'Product Knowledge');
133 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (9, 'Sales Skills');
134 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (10, 'Employee Relations');
135 • INSERT INTO Worker.Training (TrainingID, TrainingName) VALUES (11, 'Travel and Expense Management');
136
137 • SELECT * FROM Worker.Training ORDER BY TrainingID;

```

The result grid shows the data inserted into the table:

TrainingID	TrainingName
1	COVID-19 Awareness and Protection Training
2	Code of Conduct Training
3	Safety Training
4	Intro to Python
5	Machine Learning
6	Microsoft Certifications
7	Security and Privacy
8	Product Knowledge
9	Sales Skills
10	Employee Relations
11	Travel and Expense Management
12	HOLE
13	HOLE

Q7. Create the EmployeeTraining table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the EmployeeTraining table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected, showing tables, views, stored procedures, and functions. The main pane is titled 'Query 1' and contains the following DDL and data:

```

139 -- Question 7
140 • CREATE TABLE Worker.EmployeeTraining (
141     EmployeeID INTEGER NOT NULL,
142     TrainingID INTEGER NOT NULL,
143     PRIMARY KEY (EmployeeID, TrainingID),
144     FOREIGN KEY (EmployeeID) REFERENCES Worker.Employee(EmployeeID),
145     FOREIGN KEY (TrainingID) REFERENCES Worker.Training(TrainingID)
146 );
147
148 • SHOW CREATE TABLE Worker.EmployeeTraining;
149
150 -- Training for Andy Wong
151 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (1, 2); -- Code of Conduct Training
152 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (1, 3); -- Safety Training
153
154 -- Training for Vivek Pandey

```

The results grid shows the table 'EmployeeTraining' was created with the specified constraints. The data inserted includes rows for Andy Wong and Vivek Pandey.

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected. The main pane is titled 'Query 1' and contains the following DML statements:

```

160 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (5, 2); -- Code of Conduct Training
161 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (5, 10); -- Employee Relations
162
163 -- Training for John Wilson
164 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (2, 2); -- Code of Conduct Training
165 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (2, 4); -- Intro to Python
166 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (2, 5); -- Machine Learning
167
168 -- Training for Nola Davis
169 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (4, 2); -- Code of Conduct Training
170 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (4, 8); -- Product Knowledge
171 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (4, 9); -- Sales Skills
172
173 -- Training for Tom Harper
174 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (6, 2); -- Code of Conduct Training
175 • INSERT INTO Worker.EmployeeTraining (EmployeeID, TrainingID) VALUES (6, 11); -- Travel and Expense Management
176
177 • SELECT * FROM Worker.EmployeeTraining ORDER BY EmployeeID, TrainingID;

```

The results grid shows the data inserted into the 'EmployeeTraining' table, ordered by EmployeeID and TrainingID. The data includes rows for John Wilson, Nola Davis, and Tom Harper.

Q8. Create the Trainer table in the Worker database (table must be based on Physical Model Provided in the Assignment folder). (a) Columns, Primary Key (PK), Data Type & length, and NULL/NOT NULL need to be implemented, as provided in the Physical Model. (b) Show the table definition (DDL) that you implemented. (c) Insert the complete set of data provided in the Excel file (uploaded in the Assignment folder) and show the insert statements used. (d) Retrieve the data from the Trainer table by using the SELECT * statement and order by PK column(s). Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

```

Navigator: Schemas
SCHEMAS
  Filter objects
  ▶ sys
  ▶ worker
    ▶ Tables
    ▶ Views
    ▶ Stored Procedures
    ▶ Functions

Query 1 ×
File Edit View Insert Tools Help
179 -- Question 8
180 • CREATE TABLE Worker.Trainer (
181   TrainerID INTEGER NOT NULL,
182   TrainingID INTEGER NOT NULL,
183   TrainerFirstName VARCHAR(20) NOT NULL,
184   TrainerLastName VARCHAR(20) NOT NULL,
185   PRIMARY KEY (TrainerID),
186   FOREIGN KEY (TrainingID) REFERENCES Worker.Training(TrainingID)
187 );
188
189 • SHOW CREATE TABLE Worker.Trainer;
190
191 -- COVID-19 Awareness and Protection Training
192 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (1, 1, 'James', 'Smith');
193 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (2, 1, 'Johnny', 'Khor');
194
195 -- Code of Conduct Training
196 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (3, 2, 'Michael', 'Smith');

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ⓘ |
Table Create Table
▶ Trainer CREATE TABLE `trainer` ( `TrainerID` int NOT NULL, `TrainingID` int NOT NULL, `TrainerFirstName` varchar(20) NOT NULL, `TrainerLastName` varchar(20) NOT NULL, PRIMARY KEY (`TrainerID`))

Administration Schemas Information

SCHEMAS
  Filter objects
  ▶ sys
  ▶ worker
    ▶ Tables
    ▶ Views
    ▶ Stored Procedures
    ▶ Functions

190
191 -- COVID-19 Awareness and Protection Training
192 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (1, 1, 'James', 'Smith');
193 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (2, 1, 'Johnny', 'Khor');
194
195 -- Code of Conduct Training
196 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (3, 2, 'Michael', 'Smith');
197
198 -- Safety Training
199 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (4, 3, 'Maria', 'Garcia');
200
201 -- Intro to Python
202 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (5, 4, 'John', 'Placeholder');
203 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (6, 4, 'Paul', 'Deitel');
204
205 -- Machine Learning
206 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (7, 5, 'Mike', 'Taylor');
207 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (8, 5, 'Avinash', 'Navlani');
208
209 -- Microsoft Certifications
210 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (9, 6, 'Robert', 'Smith');

Administration Schemas Information

No object selected
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: ⓘ |
TrainerID TrainingID TrainerFirstName TrainerLastName
▶ 1 1 James Smith
  2 1 Johnny Khor
  3 2 Michael Smith
  4 3 Maria Garcia
  5 4 John Placeholder

```

Navigator Schemas Administration Information

No object selected

Query 1

```

209 -- Microsoft Certifications
210 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (9, 6, 'Robert', 'Smith');
211
212 -- Security and Privacy
213 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (10, 7, 'Maria', 'Rodriguez');
214
215 -- Product Knowledge
216 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (11, 8, 'Mike', 'Donlon');
217
218 -- Sales Skills
219 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (12, 9, 'Kathy', 'Corby');
220
221 -- Employee Relations
222 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (13, 10, 'Mary', 'Garcia');
223 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (14, 10, 'Vanessa', 'Placeholder');
224
225 -- Travel and Expense Management
226 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (15, 11, 'Jordan', 'Placeholder');
227 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (16, 11, 'Maria', 'Hernandez');
228
229 • SELECT * FROM Worker.Trainer ORDER BY TrainerID;
230

```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: □

TrainerID	TrainingID	TrainerFirstName	TrainerLastName
1	1	James	Smith
2	1	Johnny	Knor
3	2	Michael	Smith
4	3	Maria	Garcia
5	4	John	Placeholder

Navigator Schemas Administration Information

No object selected

Query 1

```

211
212 -- Security and Privacy
213 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (10, 7, 'Maria', 'Rodriguez');
214
215 -- Product Knowledge
216 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (11, 8, 'Mike', 'Donlon');
217
218 -- Sales Skills
219 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (12, 9, 'Kathy', 'Corby');
220
221 -- Employee Relations
222 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (13, 10, 'Mary', 'Garcia');
223 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (14, 10, 'Vanessa', 'Placeholder');
224
225 -- Travel and Expense Management
226 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (15, 11, 'Jordan', 'Placeholder');
227 • INSERT INTO Worker.Trainer (TrainerID, TrainingID, TrainerFirstName, TrainerLastName) VALUES (16, 11, 'Maria', 'Hernandez');
228
229 • SELECT * FROM Worker.Trainer ORDER BY TrainerID;
230

```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: □

TrainerID	TrainingID	TrainerFirstName	TrainerLastName
9	6	Robert	Smith
10	7	Maria	Rodriguez
11	8	Mike	Donlon
12	9	Kathy	Corby
13	10	Mary	Garcia
14	10	Vanessa	Placeholder
15	11	Jordan	Placeholder
16	11	Maria	Hernandez
NULL	NULL	NULL	NULL

Q9. Retrieve the data from the Trainer table by using the SELECT * statement with filter, WHERE TrainerLastName IS NULL. Show the output. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

```

231 -- Question 9
232 •   SELECT * FROM Worker.Trainer WHERE TrainerLastName IS NULL;
233
234 -- Question 10
235 •   SHOW TABLES FROM Worker;
236
237 -- Question 11
238 •   SELECT EmployeeID, FirstName, LastName, HireDate
239     FROM Worker.Employee
240   WHERE HireDate > (SELECT HireDate FROM Worker.Employee WHERE FirstName = 'Vivek' AND LastName = 'Panney')
241   ORDER BY EmployeeID;
242
243 -- Question 12
244 •   SELECT e.FirstName, e.LastName, t.TrainingName
245     FROM Worker.Employee e, Worker.EmployeeTraining et, Worker.Training t
246   WHERE e.EmployeeID = et.EmployeeID
247     AND et.TrainingID = t.TrainingID
248     AND e.FirstName = 'Tom'
249     AND e.LastName = 'Harper'

```

TrainerID	TrainingID	TrainerFirstName	TrainerLastName
HULL	HULL	HULL	HULL

Q10. By using the SHOW tables statements, show the list of tables you have created in the Worker database. Show the screenshot of the execution of the above statements and results. Make sure you show the print screen of the complete set of the rows and columns. The rows must be ordered by PK column(s).

```

234 -- Question 10
235 •   SHOW TABLES FROM Worker;
236
237
238

```

Tables_in_worker
department
employee
equipment
employeeequipment
employeetraining
equipment
trainer
training

Q11. Write a single-row subquery to display EmployeeID, FirstName, LastName, and HireDate of employees hired after employee Vivek Pandey. Sort the results by EmployeeID. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected. The right pane contains a query window titled 'Query 1' with the following code:

```

237 -- Question 11
238 • SELECT EmployeeID, FirstName, LastName, HireDate
239 FROM Worker.Employee
240 WHERE HireDate > (SELECT HireDate FROM Worker.Employee WHERE FirstName = 'Vivek' AND LastName = 'Pandey')
241 ORDER BY EmployeeID;
242

```

The results grid shows the following data:

EmployeeID	FirstName	LastName	HireDate
2	John	Wilson	2017-03-19
4	Nola	Davis	2016-03-23
5	Kathy	Cooper	2011-11-18
6	Tom	Harper	2010-04-11
*	NULL	NULL	NULL

Q12. Write a query to display FirstName, LastName, and TrainingName for employee Tom Harper. Sort the results by TrainingName. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with the schema 'worker' selected. The right pane contains a query window titled 'Query 1' with the following code:

```

243 -- Question 12
244 • SELECT e.FirstName, e.LastName, t.TrainingName
245 FROM Worker.Employee e, Worker.EmployeeTraining et, Worker.Training t
246 WHERE e.EmployeeID = et.EmployeeID
247 AND et.TrainingID = t.TrainingID
248 AND e.FirstName = 'Tom'
249 AND e.LastName = 'Harper'
250 ORDER BY t.TrainingName;

```

The results grid shows the following data:

FirstName	LastName	TrainingName
Tom	Harper	Code of Conduct Training
Tom	Harper	Travel and Expense Management

Q13. Write a query to display the complete list of Trainings, and trainers (first and last name) available for each training. Sort the output by TrainingName and Trainers' first and last name. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface with the following details:

- Navigator:** Shows the schema structure under the "worker" schema, including Tables, Views, Stored Procedures, and Functions.
- Query Editor:** Labeled "Query 1" with the following T-SQL code:


```

252 -- Question 13
253 • SELECT t.TrainingName, tr.TrainerFirstName, tr.TrainerLastName
254 FROM Worker.Training t
255 JOIN Worker.Trainer tr ON t.TrainingID = tr.TrainingID
256 ORDER BY t.TrainingName, tr.TrainerFirstName, tr.TrainerLastName;
      
```
- Result Grid:** Displays the results of the query in a tabular format with three columns: TrainingName, TrainerFirstName, and TrainerLastName. The data includes various training names like "Code of Conduct Training" and "Machine Learning" along with their corresponding trainer details.

Q14. Write a multiple-row subquery to display EmployeeID, FirstName, LastName, and HireDate of employees who work for the following departments: Accounting and Finance, IT Support, and Production. Sort the results by EmployeeID. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface with the following details:

- Navigator:** Shows the schema structure under the "worker" schema, including Tables, Views, Stored Procedures, and Functions.
- Query Editor:** Labeled "Query 1" with the following T-SQL code:


```

258 -- Question 14
259 • SELECT EmployeeID, FirstName, LastName, HireDate
260 FROM Worker.Employee
261 WHERE DepartmentID IN (
262   SELECT DepartmentID
263   FROM Worker.Department
264   WHERE DepartmentName IN ('Accounting and Finance', 'IT Support', 'Production')
265 )
266 ORDER BY EmployeeID;
267
      
```
- Result Grid:** Displays the results of the query in a tabular format with four columns: EmployeeID, FirstName, LastName, and HireDate. The data shows employees from the specified departments: Andy Wong, Vivek Pandey, Tom Harper, and a row for NULL values.

Q15. Write a query to display the EmployeeID, FirstName, LastName, EquipmentName, and EquipmentCostAmount for one of the employees. Sort the results by EmployeeID. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface with a query window titled "Query 1". The code is as follows:

```

-- Question 15
SELECT e.EmployeeID, e.FirstName, e.LastName, eq.EquipmentName, eq.EquipmentCostAmount
FROM Worker.Employee e
JOIN Worker.EmployeeEquipment ee ON e.EmployeeID = ee.EmployeeID
JOIN Worker.Equipment eq ON ee.EquipmentID = eq.EquipmentID
WHERE e.EmployeeID = 4
ORDER BY e.EmployeeID;

```

The result grid displays the following data:

EmployeeID	FirstName	LastName	EquipmentName	EquipmentCostAmount
4	Nola	Davis	Notebook Computers	NULL
4	Nola	Davis	Headsets	NULL

Q16. Write a query to display the TrainingID, TrainingName, TrainerID, TrainerFirstName, and TrainerLastName with the trainers who did not provide their last name. Sort the results by TrainingID and TrainerID. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface with a query window titled "Query 1". The code is as follows:

```

-- Question 16
SELECT TrainerID, TrainingID, TrainerFirstName, TrainerLastName
FROM Worker.Trainer
WHERE TrainerLastName IS NULL OR TrainerLastName = ''
ORDER BY TrainingID, TrainerID;

```

The result grid displays the following data:

TrainerID	TrainingID	TrainerFirstName	TrainerLastName
NULL	NULL	NULL	NULL

Q17. Write a query to display the distinct list of equipments used by the current employees. Sort the output by EquipmentName. Make sure you show the print screen of the complete set of the rows, and columns as specified.

The screenshot shows the SQL Server Management Studio interface with a query window titled "Query 1". The code is as follows:

```

-- Question 17
SELECT DISTINCT EquipmentName
FROM Worker.Equipment
ORDER BY EquipmentName;

```

The result grid displays the following data:

EquipmentName
Computer Monitor
Headsets
Notebook Computers

Q18. Write a query to display the FirstName, LastName, TrainingName, and trainer(s) (with first and last name in two separate columns) for one of the employees. Sort the results by TrainingName and TrainerFirstName. Make sure you show the print screen of the complete set of the rows, and columns as specified.

```

287 -- Question 18
288 • SELECT e.FirstName, e.LastName, t.TrainingName, tr.TrainerFirstName, tr.TrainerLastName
289 FROM Worker.Employee e
290 JOIN Worker.EmployeeTraining et ON e.EmployeeID = et.EmployeeID
291 JOIN Worker.Training t ON et.TrainingID = t.TrainingID
292 JOIN Worker.Trainer tr ON t.TrainingID = tr.TrainingID
293 WHERE e.EmployeeID = 3
294 ORDER BY t.TrainingName, tr.TrainerFirstName
295 LIMIT 1;
296

```

FirstName	LastName	TrainingName	TrainerFirstName	TrainerLastName
Vivek	Pandey	Code of Conduct Training	Michael	Smith

Q19. Write a query to display the EmployeeID, FirstName, LastName, DepartmentID, DepartmentName, EquipmentID, EquipmentName for all employees. Sort the results by EmployeeID, DepartmentID, and EquipmentID. Make sure you show the print screen of the complete set of the rows, and columns as specified.

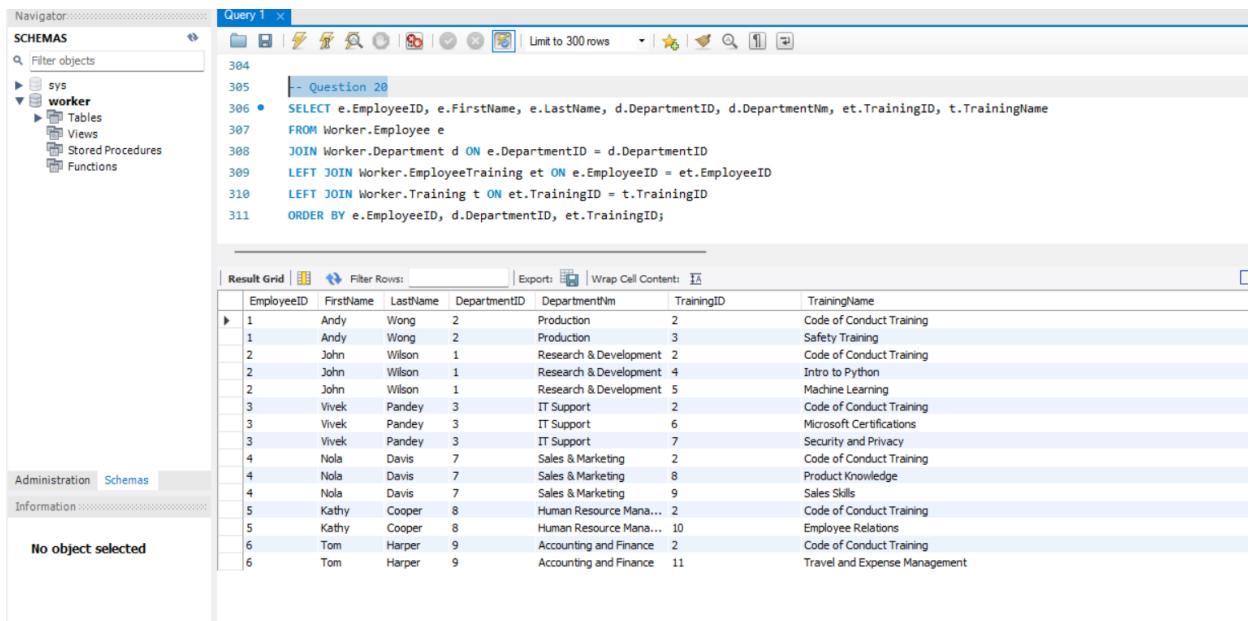
```

297 -- Question 19
298 • SELECT e.EmployeeID, e.FirstName, e.LastName, d.DepartmentID, d.DepartmentNm, ee.EquipmentID, eq.EquipmentName
299 FROM Worker.Employee e
300 JOIN Worker.Department d ON e.DepartmentID = d.DepartmentID
301 LEFT JOIN Worker.EmployeeEquipment ee ON e.EmployeeID = ee.EmployeeID
302 LEFT JOIN Worker.Equipment eq ON ee.EquipmentID = eq.EquipmentID
303 ORDER BY e.EmployeeID, d.DepartmentID, ee.EquipmentID;
304

```

EmployeeID	FirstName	LastName	DepartmentID	DepartmentNm	EquipmentID	EquipmentName
1	Andy	Wong	2	Production	1	Notebook Computers
2	John	Wilson	1	Research & Development	1	Notebook Computers
2	John	Wilson	1	Research & Development	3	Computer Monitor
3	Vivek	Pandey	3	IT Support	1	Notebook Computers
3	Vivek	Pandey	3	IT Support	2	Headsets
3	Vivek	Pandey	3	IT Support	3	Computer Monitor
4	Nola	Davis	7	Sales & Marketing	1	Notebook Computers
4	Nola	Davis	7	Sales & Marketing	2	Headsets
5	Kathy	Cooper	8	Human Resource Management	1	Notebook Computers
5	Kathy	Cooper	8	Human Resource Management	2	Headsets
5	Kathy	Cooper	8	Human Resource Management	3	Computer Monitor
6	Tom	Harper	9	Accounting and Finance	1	Notebook Computers
6	Tom	Harper	9	Accounting and Finance	3	Computer Monitor

Q20. Write a query to display the EmployeeID, FirstName, LastName, DepartmentID, DepartmentName, TrainingID, TrainingName for all employees. Sort the results by EmployeeID, DepartmentID, and TrainingID. Make sure you show the print screen of the complete set of the rows, and columns as specified.



The screenshot shows the SSMS interface with a query window titled "Query 1". The code in the query window is:

```

304
305      -- Question 20
306  •      SELECT e.EmployeeID, e.FirstName, e.LastName, d.DepartmentID, d.DepartmentNm, et.TrainingID, t.TrainingName
307  FROM Worker.Employee e
308  JOIN Worker.Department d ON e.DepartmentID = d.DepartmentID
309  LEFT JOIN Worker.EmployeeTraining et ON e.EmployeeID = et.EmployeeID
310  LEFT JOIN Worker.Training t ON et.TrainingID = t.TrainingID
311  ORDER BY e.EmployeeID, d.DepartmentID, et.TrainingID;

```

The results grid displays the following data:

EmployeeID	FirstName	LastName	DepartmentID	DepartmentNm	TrainingID	TrainingName
1	Andy	Wong	2	Production	2	Code of Conduct Training
1	Andy	Wong	2	Production	3	Safety Training
2	John	Wilson	1	Research & Development	2	Code of Conduct Training
2	John	Wilson	1	Research & Development	4	Intro to Python
2	John	Wilson	1	Research & Development	5	Machine Learning
3	Vivek	Pandey	3	IT Support	2	Code of Conduct Training
3	Vivek	Pandey	3	IT Support	6	Microsoft Certifications
3	Vivek	Pandey	3	IT Support	7	Security and Privacy
4	Nola	Davis	7	Sales & Marketing	2	Code of Conduct Training
4	Nola	Davis	7	Sales & Marketing	8	Product Knowledge
4	Nola	Davis	7	Sales & Marketing	9	Sales Skills
5	Kathy	Cooper	8	Human Resource Mana...	2	Code of Conduct Training
5	Kathy	Cooper	8	Human Resource Mana...	10	Employee Relations
6	Tom	Harper	9	Accounting and Finance	2	Code of Conduct Training
6	Tom	Harper	9	Accounting and Finance	11	Travel and Expense Management