

SQL project – Tools for Data Science

Victoria Nguyen

-- Use your database\

-- Code here:

USE vyn1;

-- Screenshot of the output here:

Question 1.

Create a table in your database named Pilot

-- Add variables: PilotID (primary key), Name, State

-- Populate the table with fake data - at least 5 rows

-- Select everything from the table

-- Code here

-- Create a table in your database named Pilot\

-- Add variables: PilotID (primary key), Name, State\

CREATE TABLE Pilot (

PilotID INT PRIMARY KEY,

 Name VARCHAR(50),

 State VARCHAR(50)

);

-- Populate the table with fake data - at least 5 rows\

INSERT INTO Pilot (PilotID, Name, State)

VALUES (1, 'Victoria Nguyen', 'Texas');

INSERT INTO Pilot (PilotID, Name, State)

VALUES (2, 'Tom Holland', 'California');

INSERT INTO Pilot (PilotID, Name, State)

VALUES (3, 'Angelina Jolie', 'Oregon');

INSERT INTO Pilot (PilotID, Name, State)

VALUES (4, 'Jennie Kim', 'New York');

INSERT INTO Pilot (PilotID, Name, State)

VALUES (5, 'Jacob Elordi', 'Minnesota');

-- Select everything from the table\

SELECT *

FROM Pilot;

-- Screenshot of the output here:

	PilotID	Name	State
1	1	Victoria Nguyen	Texas
2	2	Tom Holland	California
3	3	Angelina Jolie	Oregon
4	4	Jennie Kim	New York
5	5	Jacob Elordi	Minnesota

Question 2.

Create a table in your database named Certification

-- Add variables: CertificateID (Primary), PilotID (foreign key references to Pilot table), Certification_Level

-- Populate the table with fake data - at least 5 rows. (Consider Certification_Level = 1 for Student Pilot and 2 for Professional Pilot)

-- Select everything from the table

-- Code here:

-- Create a table in your database named Certification\

-- Add variables: CertificateID (Primary), PilotID (foreign key references to Pilot table), Certification_Level\

```
CREATE TABLE Certification (  
    CertificateID INT PRIMARY KEY identity (1,1),  
    PilotID INT,  
    Certification_Level INT,  
    FOREIGN KEY (PilotID) REFERENCES Pilot(PilotID)  
);
```

-- Populate the table with fake data - at least 5 rows. (Consider Certification_Level = 1 for Student Pilot and 2 for Professional Pilot)\

```
INSERT INTO Certification (PilotID, Certification_Level)  
VALUES (1, 2);
```

```
INSERT INTO Certification (PilotID, Certification_Level)  
VALUES (2, 1);
```

```
INSERT INTO Certification (PilotID, Certification_Level)  
VALUES (3, 2);
```

```
INSERT INTO Certification (PilotID, Certification_Level)  
VALUES (4, 2);
```

```
INSERT INTO Certification (PilotID, Certification_Level)
```

```
VALUES (5, 1);
```

```
-- Select everything from the table\
```

```
SELECT *
```

```
FROM Certification;
```

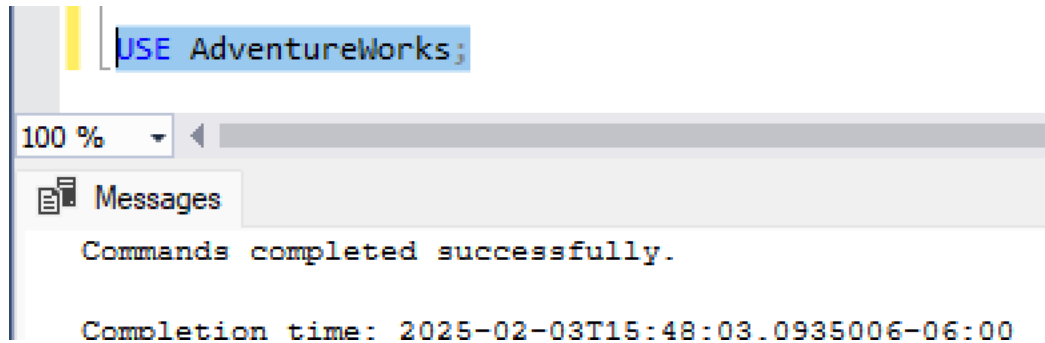
	CertificateID	PilotID	Certification_Level
1	1	1	2
2	2	2	1
3	3	3	2
4	4	4	2
5	5	5	1

--Use AdventureWorks database

-- Code here:

USE AdventureWorks;

-- Screenshot of the output here:



Question 3.

Write a query to list the SalesOrderID, SalesOrderDetailID, ProductID, OrderQty from SalesOrderDetail table

-- Code here:

-- Write a query to list the SalesOrderID, SalesOrderDetailID, ProductID, OrderQty from SalesOrderDetail table

SELECT SalesOrderID, SalesOrderDetailID, ProductID, OrderQty

FROM Sales.SalesOrderDetail;

	SalesOrderID	SalesOrderDetailID	ProductID	OrderQty
1	43659	1	776	1
2	43659	2	777	3
3	43659	3	778	1
4	43659	4	771	1
5	43659	5	772	1
6	43659	6	773	2
7	43659	7	774	1
8	43659	8	714	3
9	43659	9	716	1
10	43659	10	709	6
11	43659	11	712	2
12	43659	12	711	4
13	43660	13	762	1
14	43660	14	758	1
15	43661	15	745	1
16	43661	16	743	1
17	43661	17	747	2
18	43661	18	712	4
19	43661	19	715	4
20	43661	20	742	2
21	43661	21	775	3
22	43661	22	778	2
23	43661	23	711	2
24	43661	24	741	2
25	43661	25	776	4
26	43661	26	773	2
27	43661	27	716	2
28	43661	28	777	2
29	43661	29	708	5
30	43662	30	764	3
31	43662	31	770	5

-- Screenshot of the output here:

Question 4.

Write a query to list the SalesOrderID, TerritoryID from SalesOrderHeader table

-- Code here

SELECT SalesOrderID, TerritoryID

FROM Sales.SalesOrderHeader;

-- Screenshot of the output here:

	SalesOrderID	TerritoryID
1	43659	5
2	43660	5
3	43661	6
4	43662	6
5	43663	4
6	43664	1
7	43665	1
8	43666	4
9	43667	3
10	43668	6
11	43669	1
12	43670	3
13	43671	1
14	43672	6
15	43673	2
16	43674	6
17	43675	3
18	43676	5
19	43677	6
20	43678	4
21	43679	6
22	43680	4
23	43681	5
24	43682	2
25	43683	1
26	43684	5
27	43685	5
28	43686	1
29	43687	2
30	43688	2

Question 5.

Write a query to list the name of the Product, ProductID, OrderQty
-- and SalesOrderID by joining Product and SalesOrderDetail Tables

-- Code here:

```
SELECT P.Name, P.ProductID, SOD.OrderQty, SOD.SalesOrderID
```

```
FROM Production.Product AS P
```

```
JOIN Sales.SalesOrderDetail AS SOD ON P.ProductID = SOD.ProductID;
```

-- Screenshot of the output here:

	Name	ProductID	OrderQty	SalesOrderID
1	Mountain-100 Black, 42	776	1	43659
2	Mountain-100 Black, 44	777	3	43659
3	Mountain-100 Black, 48	778	1	43659
4	Mountain-100 Silver, 38	771	1	43659
5	Mountain-100 Silver, 42	772	1	43659
6	Mountain-100 Silver, 44	773	2	43659
7	Mountain-100 Silver, 48	774	1	43659
8	Long-Sleeve Logo Jersey, M	714	3	43659
9	Long-Sleeve Logo Jersey, XL	716	1	43659
10	Mountain Bike Socks, M	709	6	43659
11	AWC Logo Cap	712	2	43659
12	Sport-100 Helmet, Blue	711	4	43659
13	Road-650 Red, 44	762	1	43660
14	Road-450 Red, 52	758	1	43660
15	HL Mountain Frame - Black,...	745	1	43661
16	HL Mountain Frame - Black,...	743	1	43661
17	HL Mountain Frame - Black,...	747	2	43661
18	AWC Logo Cap	712	4	43661
19	Long-Sleeve Logo Jersey, L	715	4	43661
20	HL Mountain Frame - Silver,...	742	2	43661
21	Mountain-100 Black, 38	775	3	43661
22	Mountain-100 Black, 48	778	2	43661
23	Sport-100 Helmet, Blue	711	2	43661
24	HL Mountain Frame - Silver,...	741	2	43661
25	Mountain-100 Black, 42	776	4	43661
26	Mountain-100 Silver, 44	773	2	43661
27	Long-Sleeve Logo Jersey, XL	716	2	43661
28	Mountain-100 Black, 44	777	2	43661
29	Sport-100 Helmet, Black	708	5	43661
30	Road-650 Red, 52	764	3	43662
31	Road-650 Black, 52	770	5	43662

Question 6.

Write a query to list the name of the Product, ProductID, Total OrderQty by ProductID

-- by joining Product and SalesOrderDetail Tables

-- Code here:

```
SELECT P.Name, P.ProductID, SUM(SOD.OrderQty) AS TotalOrderQty
```

```
FROM Production.Product AS P
```

```
JOIN Sales.SalesOrderDetail AS SOD ON P.ProductID = SOD.ProductID
```

```
GROUP BY P.Name, P.ProductID;
```

-- Screenshot of the output here:

	Name	ProductID	TotalOrderQty
1	All-Purpose Bike Stand	879	249
2	AWC Logo Cap	712	8311
3	Bike Wash - Dissolver	877	3319
4	Cable Lock	843	1087
5	Chain	952	774
6	Classic Vest, L	866	207
7	Classic Vest, M	865	2284
8	Classic Vest, S	864	4247
9	Fender Set - Mountain	878	2121
10	Front Brakes	948	789
11	Front Derailleur	945	813
12	Full-Finger Gloves, L	863	3378
13	Full-Finger Gloves, M	862	2206
14	Full-Finger Gloves, S	861	500
15	Half-Finger Gloves, L	860	1276
16	Half-Finger Gloves, M	859	3464
17	Half-Finger Gloves, S	858	2188
18	Hitch Rack - 4-Bike	876	3166
19	HL Bottom Bracket	996	543
20	HL Crankset	951	613
21	HL Fork	804	444
22	HL Headset	807	255
23	HL Mountain Frame -...	747	659
24	HL Mountain Frame -...	743	1181
25	HL Mountain Frame -...	744	17
26	HL Mountain Frame -...	745	131
27	HL Mountain Frame -...	748	1206
28	HL Mountain Frame -...	739	347
29	HL Mountain Frame -...	742	649
30	HL Mountain Frame -...	741	173
31	HL Mountain Front	817	222

Question 7.

Pick a SalesOrderID from SalesOrderDetail table and find the total OrderQty and Average UnitPrice

-- Code here:

```
SELECT SUM(OrderQty) AS TotalOrderQty, AVG(UnitPrice) AS AverageUnitPrice
```

```
FROM Sales.SalesOrderDetail
```

```
WHERE SalesOrderID = 43665
```

-- Screenshot of the output here:

	TotalOrderQty	AverageUnitPrice
1	20	1022.0069