Chapter2_Part2

September 14, 2021

0.1 Chapter 2 Part 2

Akhil Devarapalli

Dr. DeGennaro

Database Design

Page 122-123 Questions to 2.40 to 2.55

[64]:
```
import pymysql.cursors
import pandas as pd
# Connecting to database
connection = pymysql.connect (host = '10.1.11.26',
                              user = 'adevarapalli',
                              password = 'password',
                              database= 'cape_codd',
                              charset='utf8mb4',
                             cursorclass=pymysql.cursors.DictCursor)
```

Question 2.40 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, WarehouseCity, and WarehouseState for all items stored in the Atlanta, Bangor, or Chicago warehouse. Do not use the IN keyword

[65]:
```
sql2_40 = 'SELECT SKU, SKU_Description, INVENTORY.WarehouseID, WarehouseCity,␣
 ↪WarehouseState FROM INVENTORY JOIN WAREHOUSE ON INVENTORY.WarehouseID =␣
 ↪WAREHOUSE.WarehouseID WHERE WarehouseCity = "Atlanta" OR WarehouseCity =␣
 ↪"Bangor" OR WarehouseCity = "Chicago";'
df = pd.read_sql_query(sql2_40, connection)
df
```

[65]:

	SKU	SKU_Description	WarehouseID	WarehouseCity	\
0	100100	Std. Scuba Tank, Yellow	100	Atlanta	
1	100200	Std. Scuba Tank, Magenta	100	Atlanta	
2	101100	Dive Mask, Small Clear	100	Atlanta	
3	101200	Dive Mask, Med Clear	100	Atlanta	
4	201000	Half-dome Tent	100	Atlanta	

5	202000	Half-dome Tent Vestibule	100	Atlanta
6	301000	Light Fly Climbing Harness	100	Atlanta
7	302000	Locking Carabiner, Oval	100	Atlanta
8	100100	Std. Scuba Tank, Yellow	200	Chicago
9	100200	Std. Scuba Tank, Magenta	200	Chicago
10	101100	Dive Mask, Small Clear	200	Chicago
11	101200	Dive Mask, Med Clear	200	Chicago
12	201000	Half-dome Tent	200	Chicago
13	202000	Half-dome Tent Vestibule	200	Chicago
14	301000	Light Fly Climbing Harness	200	Chicago
15	302000	Locking Carabiner, Oval	200	Chicago
16	100100	Std. Scuba Tank, Yellow	300	Bangor
17	100200	Std. Scuba Tank, Magenta	300	Bangor
18	101100	Dive Mask, Small Clear	300	Bangor
19	101200	Dive Mask, Med Clear	300	Bangor
20	201000	Half-dome Tent	300	Bangor
21	202000	Half-dome Tent Vestibule	300	Bangor
22	301000	Light Fly Climbing Harness	300	Bangor
23	302000	Locking Carabiner, Oval	300	Bangor

	WarehouseState
0	GA
1	GA
2	GA
3	GA
4	GA
5	GA
6	GA
7	GA
8	IL
9	IL
10	IL
11	IL
12	IL
13	IL
14	IL
15	IL
16	ME
17	ME
18	ME
19	ME
20	ME
21	ME
22	ME
23	ME

Question 2.41 Write an SQL statement to display the SKU, SKU_Description, WarehouseID,

WarehouseCity, and WarehouseState for all items stored in the Atlanta, Bangor, or Chicago warehouse. Use the IN keyword.

[66]:
```
sql2_41 = 'SELECT SKU, SKU_Description, INVENTORY.WarehouseID, WarehouseCity,␣
 ↪WarehouseState FROM INVENTORY JOIN WAREHOUSE ON INVENTORY.WarehouseID =␣
 ↪WAREHOUSE.WarehouseID WHERE WarehouseCity IN ("Atlanta", "Bangor",␣
 ↪"Chicago");'
df = pd.read_sql_query(sql2_41, connection)
df
```

[66]:

	SKU	SKU_Description	WarehouseID	WarehouseCity	WarehouseState
0	100100	Std. Scuba Tank, Yellow	100	Atlanta	GA
1	100200	Std. Scuba Tank, Magenta	100	Atlanta	GA
2	101100	Dive Mask, Small Clear	100	Atlanta	GA
3	101200	Dive Mask, Med Clear	100	Atlanta	GA
4	201000	Half-dome Tent	100	Atlanta	GA
5	202000	Half-dome Tent Vestibule	100	Atlanta	GA
6	301000	Light Fly Climbing Harness	100	Atlanta	GA
7	302000	Locking Carabiner, Oval	100	Atlanta	GA
8	100100	Std. Scuba Tank, Yellow	200	Chicago	IL
9	100200	Std. Scuba Tank, Magenta	200	Chicago	IL
10	101100	Dive Mask, Small Clear	200	Chicago	
11	101200	Dive Mask, Med Clear	200	Chicago	
12	201000	Half-dome Tent	200	Chicago	
13	202000	Half-dome Tent Vestibule	200	Chicago	
14	301000	Light Fly Climbing Harness	200	Chicago	
15	302000	Locking Carabiner, Oval	200	Chicago	
16	100100	Std. Scuba Tank, Yellow	300	Bangor	
17	100200	Std. Scuba Tank, Magenta	300	Bangor	
18	101100	Dive Mask, Small Clear	300	Bangor	
19	101200	Dive Mask, Med Clear	300	Bangor	
20	201000	Half-dome Tent	300	Bangor	
21	202000	Half-dome Tent Vestibule	300	Bangor	
22	301000	Light Fly Climbing Harness	300	Bangor	
23	302000	Locking Carabiner, Oval	300	Bangor	

```
10              IL
11              IL
12              IL
13              IL
14              IL
15              IL
16              ME
17              ME
18              ME
19              ME
20              ME
21              ME
22              ME
23              ME
```

Question 2.42 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, WarehouseCity, and WarehouseState of all items not stored in the Atlanta, Bangor, or Chicago warehouse. Do not use the NOT IN keyword.

[67]:
```
sql2_42 = 'SELECT SKU, SKU_Description, INVENTORY.WarehouseID, WarehouseCity,␣
 ↪WarehouseState FROM INVENTORY JOIN WAREHOUSE ON INVENTORY.WarehouseID =␣
 ↪WAREHOUSE.WarehouseID WHERE WarehouseCity != "Atlanta" AND WarehouseCity !=␣
 ↪"Bangor" AND WarehouseCity != "Chicago";'
df = pd.read_sql_query(sql2_42, connection)
df
```

[67]:

	SKU	SKU_Description	WarehouseID	WarehouseCity	WarehouseState
0	100100	Std. Scuba Tank, Yellow	400	Seattle	WA
1	100200	Std. Scuba Tank, Magenta	400	Seattle	WA
2	101100	Dive Mask, Small Clear	400	Seattle	WA
3	101200	Dive Mask, Med Clear	400	Seattle	WA
4	201000	Half-dome Tent	400	Seattle	WA
5	202000	Half-dome Tent Vestibule	400	Seattle	WA
6	301000	Light Fly Climbing Harness	400	Seattle	WA
7	302000	Locking Carabiner, Oval	400	Seattle	WA

Question 2.43 Write an SQL statement to display the SKU, SKU_Description, WarehouseID, WarehouseCity, and WarehouseState of all items not stored in the Atlanta, Bangor, or Chicago

warehouse. Use the NOT IN keyword.

[68]:
```
sql2_43 = 'SELECT SKU, SKU_Description, INVENTORY.WarehouseID, WarehouseCity,␣
 ↪WarehouseState FROM INVENTORY JOIN WAREHOUSE ON INVENTORY.WarehouseID =␣
 ↪WAREHOUSE.WarehouseID WHERE WarehouseCity NOT IN ("Atlanta", "Bangor",␣
 ↪"Chicago");'
df = pd.read_sql_query(sql2_43, connection)
df
```

[68]:

	SKU	SKU_Description	WarehouseID	WarehouseCity	WarehouseState
0	100100	Std. Scuba Tank, Yellow	400	Seattle	WA
1	100200	Std. Scuba Tank, Magenta	400	Seattle	WA
2	101100	Dive Mask, Small Clear	400	Seattle	WA
3	101200	Dive Mask, Med Clear	400	Seattle	WA
4	201000	Half-dome Tent	400	Seattle	WA
5	202000	Half-dome Tent Vestibule	400	Seattle	WA
6	301000	Light Fly Climbing Harness	400	Seattle	WA
7	302000	Locking Carabiner, Oval	400	Seattle	WA

Question 2.44 Write an SQL statement to produce a single column called ItemLocation that combines the SKU_Description, the phrase "is located in," and WarehouseCity. Do not be concerned with removing leading or trailing blanks.

[69]:
```
sql2_44 = 'SELECT CONCAT(INVENTORY.SKU_Description, " located in ", WAREHOUSE.
 ↪WarehouseCity) AS ItemLocation FROM INVENTORY JOIN WAREHOUSE ON INVENTORY.
 ↪WarehouseID = WAREHOUSE.WarehouseID;'
df = pd.read_sql_query(sql2_44, connection)
df
```

[69]:

	ItemLocation
0	Std. Scuba Tank, Yellow located in Atlanta
1	Std. Scuba Tank, Magenta located in Atlanta
2	Dive Mask, Small Clear located in Atlanta
3	Dive Mask, Med Clear located in Atlanta
4	Half-dome Tent located in Atlanta
5	Half-dome Tent Vestibule located in Atlanta
6	Light Fly Climbing Harness located in Atlanta
7	Locking Carabiner, Oval located in Atlanta

```
8      Std. Scuba Tank, Yellow located in Chicago
9     Std. Scuba Tank, Magenta located in Chicago
10      Dive Mask, Small Clear located in Chicago
11        Dive Mask, Med Clear located in Chicago
12              Half-dome Tent located in Chicago
13   Half-dome Tent Vestibule located in Chicago
14  Light Fly Climbing Harness located in Chicago
15    Locking Carabiner, Oval located in Chicago
16     Std. Scuba Tank, Yellow located in Bangor
17    Std. Scuba Tank, Magenta located in Bangor
18      Dive Mask, Small Clear located in Bangor
19        Dive Mask, Med Clear located in Bangor
20              Half-dome Tent located in Bangor
21    Half-dome Tent Vestibule located in Bangor
22   Light Fly Climbing Harness located in Bangor
23     Locking Carabiner, Oval located in Bangor
24    Std. Scuba Tank, Yellow located in Seattle
25   Std. Scuba Tank, Magenta located in Seattle
26     Dive Mask, Small Clear located in Seattle
27       Dive Mask, Med Clear located in Seattle
28              Half-dome Tent located in Seattle
29   Half-dome Tent Vestibule located in Seattle
30  Light Fly Climbing Harness located in Seattle
31    Locking Carabiner, Oval located in Seattle
```

Question 2.45 Write an SQL statement to show the SKU, SKU_Description, and WarehouseID for all items stored in a warehouse managed by 'Lucille Smith'. Use a subquery.

[70]:
```
sql2_45 = 'SELECT SKU, SKU_Description, WarehouseID FROM INVENTORY WHERE␣
 ↪WarehouseID IN (SELECT WarehouseID FROM WAREHOUSE WHERE Manager = "Lucille␣
 ↪Smith");'
df = pd.read_sql_query(sql2_45, connection)
df
```

[70]:

	SKU	SKU_Description	WarehouseID
0	100100	Std. Scuba Tank, Yellow	200
1	100200	Std. Scuba Tank, Magenta	200
2	101100	Dive Mask, Small Clear	200
3	101200	Dive Mask, Med Clear	200
4	201000	Half-dome Tent	200
5	202000	Half-dome Tent Vestibule	200
6	301000	Light Fly Climbing Harness	200
7	302000	Locking Carabiner, Oval	200

Question 2.46 Write an SQL statement to show the SKU, SKU_Description, and WarehouseID for all items stored in a warehouse managed by 'Lucille Smith'. Use a join, but do not use JOIN ON syntax.

[71]:
```
sql2_46 = 'SELECT SKU, SKU_Description, INVENTORY.WarehouseID FROM INVENTORY␣
 ↪JOIN WAREHOUSE WHERE INVENTORY.WarehouseID = WAREHOUSE.WarehouseID AND␣
 ↪WAREHOUSE.Manager = "Lucille Smith";'
df = pd.read_sql_query(sql2_46, connection)
df
```

[71]:

	SKU	SKU_Description	WarehouseID
0	100100	Std. Scuba Tank, Yellow	200
1	100200	Std. Scuba Tank, Magenta	200
2	101100	Dive Mask, Small Clear	200
3	101200	Dive Mask, Med Clear	200
4	201000	Half-dome Tent	200
5	202000	Half-dome Tent Vestibule	200
6	301000	Light Fly Climbing Harness	200
7	302000	Locking Carabiner, Oval	200

Question 2.47 Write an SQL statement to show the SKU, SKU_Description, and WarehouseID for all items stored in a warehouse managed by 'Lucille Smith'. Use a join using JOIN ON syntax.

[72]:
```
sql2_47 = "SELECT SKU, SKU_Description, INVENTORY.WarehouseID FROM INVENTORY␣
 ↪JOIN WAREHOUSE ON INVENTORY.WarehouseID = WAREHOUSE.WarehouseID WHERE␣
 ↪WAREHOUSE.Manager = 'Lucille Smith';"
df = pd.read_sql_query(sql2_47, connection)
df
```

[72]:

	SKU	SKU_Description	WarehouseID
0	100100	Std. Scuba Tank, Yellow	200
1	100200	Std. Scuba Tank, Magenta	200
2	101100	Dive Mask, Small Clear	200
3	101200	Dive Mask, Med Clear	200
4	201000	Half-dome Tent	200
5	202000	Half-dome Tent Vestibule	200
6	301000	Light Fly Climbing Harness	200
7	302000	Locking Carabiner, Oval	200

Question 2.48 Write an SQL statement to show the WarehouseID and average QuantityOnHand of all items stored in a warehouse managed by 'Lucille Smith'. Use a subquery.

[73]:
```
sql2_48 = 'SELECT AVG(QuantityOnHand), INVENTORY.WarehouseID FROM INVENTORY␣
 ↪WHERE WarehouseID IN (SELECT WarehouseID FROM WAREHOUSE WHERE Manager =␣
 ↪"Lucille Smith")GROUP BY WarehouseID;'
df = pd.read_sql_query(sql2_48, connection)
df
```

[73]:

	AVG(QuantityOnHand)	WarehouseID
0	217.0	200

Question 2.49 Write an SQL statement to show the WarehouseID and average QuantityOnHand

of all items stored in a warehouse managed by 'Lucille Smith'. Use a join, but do not use JOIN ON syntax.

[74]:
```
sql2_49 = 'SELECT INVENTORY.WarehouseID, AVG(QuantityOnHand)FROM INVENTORY JOIN␣
 ↪WAREHOUSE WHERE INVENTORY.WarehouseID = WAREHOUSE.WarehouseID AND Manager =␣
 ↪"Lucille Smith" GROUP BY WarehouseID;'
df = pd.read_sql_query(sql2_49, connection)
df
```

[74]:

	WarehouseID	AVG(QuantityOnHand)
0	200	217.0

Question 2.50 Write an SQL statement to show the WarehouseID and average QuantityOnHand of all items stored in warehouse managed by 'Lucille Smith'. Use a join using JOIN ON syntax.

[75]:
```
sql2_50 = 'SELECT INVENTORY.WarehouseID, AVG(QuantityOnHand) FROM INVENTORY␣
 ↪JOIN WAREHOUSE ON INVENTORY.WarehouseID = WAREHOUSE.WarehouseID WHERE␣
 ↪Manager = "Lucille Smith" GROUP BY WarehouseID;'
df = pd.read_sql_query(sql2_50, connection)
df
```

[75]:

	WarehouseID	AVG(QuantityOnHand)
0	200	217.0

Question 2.51 Write an SQL statement to show the WarehouseID, WarehouseCity, WarehouseState, Manager, SKU, SKU_Description, and QuantityOnHand of all items stored in a warehouse managed by 'Lucille Smith'. Use a join using JOIN ON syntax.

[76]:
```
sql2_51 = 'SELECT WAREHOUSE.WarehouseID, WAREHOUSE.WarehouseCity, WAREHOUSE.
 ↪WarehouseState, WAREHOUSE.Manager, INVENTORY.SKU, INVENTORY.SKU_Description,␣
 ↪INVENTORY.QuantityOnHand FROM WAREHOUSE JOIN INVENTORY ON WAREHOUSE.
 ↪WarehouseID = INVENTORY.WarehouseID WHERE Manager = "Lucille Smith";'
df = pd.read_sql_query(sql2_51, connection)
df
```

[76]:

	WarehouseID	WarehouseCity	WarehouseState	Manager	SKU	SKU_Description	QuantityOnHand
0	200	Chicago	IL	Lucille Smith	100100	Std. Scuba Tank, Yellow	100
1	200	Chicago	IL	Lucille Smith	100200	Std. Scuba Tank, Magenta	75
2	200	Chicago	IL	Lucille Smith	101100		
3	200	Chicago	IL	Lucille Smith	101200		
4	200	Chicago	IL	Lucille Smith	201000		
5	200	Chicago	IL	Lucille Smith	202000		
6	200	Chicago	IL	Lucille Smith	301000		
7	200	Chicago	IL	Lucille Smith	302000		

2	Dive Mask, Small Clear	0
3	Dive Mask, Med Clear	50
4	Half-dome Tent	10
5	Half-dome Tent Vestibule	1
6	Light Fly Climbing Harness	250
7	Locking Carabiner, Oval	1250

Question 2.52 Write an SQL statement to display the WarehouseID, the sum of QuantityOnOrder, and the sum of QuantityOnHand, grouped by WarehouseID and QuantityOnOrder. Name the sum of QuantityOnOrder as TotalItemsOnOrder and the sum of Quantity OnHand as TotalItemsOnHand. Use only the INVENTORY table in your SQL statement.

[77]:
```
sql2_52 = 'SELECT WarehouseID, SUM(QuantityOnOrder) AS TotalItemsOnOrder,␣
↪SUM(QuantityOnHand) AS TotalItemsOnHand FROM INVENTORY GROUP BY WarehouseID,␣
↪QuantityOnOrder;'
df = pd.read_sql_query(sql2_52, connection)
df
```

[77]:

	WarehouseID	TotalItemsOnOrder	TotalItemsOnHand
0	100	0.0	1250.0
1	100	30.0	200.0
2	100	100.0	2.0
3	100	500.0	310.0
4	100	1000.0	100.0
5	200	0.0	1250.0
6	200	50.0	100.0
7	200	75.0	75.0
8	200	750.0	261.0
9	200	1000.0	50.0
10	300	0.0	925.0
11	300	100.0	100.0
12	300	200.0	300.0
13	300	250.0	0.0
14	300	500.0	500.0
15	400	0.0	900.0
16	400	200.0	0.0
17	400	750.0	250.0
18	400	1000.0	0.0

Question 2.53 Explain why you cannot use a subquery in your answer to Review Question 2.52 **Answer -->** A subquery needs you to query another table. In 2.52 you could only query the INVENTORY table

Question 2.54 Explain how subqueries and joins differ. **Answer -->** A subquery can only be used to retreive data from the top table, whereas a join can be used to obtain data from any number of tables.

Question 2.55 Write an SQL statement to join WAREHOUSE and INVENTORY and include all

rows of WAREHOUSE in your answer, regardless of whether they have any INVENTORY. Include all columns of both tables, but do not repeat the join column.

[78]:
```
sql2_55 = 'SELECT WAREHOUSE.WarehouseID, WAREHOUSE.WarehouseCity, WAREHOUSE.
 ↪WarehouseCity, WAREHOUSE.SquareFeet, WAREHOUSE.Manager, INVENTORY.SKU,␣
 ↪INVENTORY.SKU_Description, INVENTORY.QuantityOnHand, INVENTORY.
 ↪QuantityOnOrder FROM WAREHOUSE LEFT OUTER JOIN INVENTORY ON WAREHOUSE.
 ↪WarehouseID = INVENTORY.WarehouseID;'
df = pd.read_sql_query(sql2_55, connection)
df
```

[78]:

	WarehouseID	WarehouseCity	WarehouseCity	SquareFeet	Manager	\
0	100	Atlanta	Atlanta	125000	Dave Jones	
1	100	Atlanta	Atlanta	125000	Dave Jones	
2	100	Atlanta	Atlanta	125000	Dave Jones	
3	100	Atlanta	Atlanta	125000	Dave Jones	
4	100	Atlanta	Atlanta	125000	Dave Jones	
5	100	Atlanta	Atlanta	125000	Dave Jones	
6	100	Atlanta	Atlanta	125000	Dave Jones	
7	100	Atlanta	Atlanta	125000	Dave Jones	
8	200	Chicago	Chicago	100000	Lucille Smith	
9	200	Chicago	Chicago	100000	Lucille Smith	
10	200	Chicago	Chicago	100000	Lucille Smith	
11	200	Chicago	Chicago	100000	Lucille Smith	
12	200	Chicago	Chicago	100000	Lucille Smith	
13	200	Chicago	Chicago	100000	Lucille Smith	
14	200	Chicago	Chicago	100000	Lucille Smith	
15	200	Chicago	Chicago	100000	Lucille Smith	
16	300	Bangor	Bangor	150000	Bart Evans	
17	300	Bangor	Bangor	150000	Bart Evans	
18	300	Bangor	Bangor	150000	Bart Evans	
19	300	Bangor	Bangor	150000	Bart Evans	
20	300	Bangor	Bangor	150000	Bart Evans	
21	300	Bangor	Bangor	150000	Bart Evans	
22	300	Bangor	Bangor	150000	Bart Evans	
23	300	Bangor	Bangor	150000	Bart Evans	
24	400	Seattle	Seattle	130000	Dale Rogers	
25	400	Seattle	Seattle	130000	Dale Rogers	
26	400	Seattle	Seattle	130000	Dale Rogers	
27	400	Seattle	Seattle	130000	Dale Rogers	
28	400	Seattle	Seattle	130000	Dale Rogers	
29	400	Seattle	Seattle	130000	Dale Rogers	
30	400	Seattle	Seattle	130000	Dale Rogers	
31	400	Seattle	Seattle	130000	Dale Rogers	
32	500	San Francisco	San Francisco	200000	Grace Jefferson	

SKU	SKU_Description	QuantityOnHand	QuantityOnOrder

0	100100.0	Std. Scuba Tank, Yellow	250.0	0.0
1	100200.0	Std. Scuba Tank, Magenta	200.0	30.0
2	101100.0	Dive Mask, Small Clear	0.0	500.0
3	101200.0	Dive Mask, Med Clear	100.0	500.0
4	201000.0	Half-dome Tent	2.0	100.0
5	202000.0	Half-dome Tent Vestibule	10.0	250.0
6	301000.0	Light Fly Climbing Harness	300.0	250.0
7	302000.0	Locking Carabiner, Oval	1000.0	0.0
8	100100.0	Std. Scuba Tank, Yellow	100.0	50.0
9	100200.0	Std. Scuba Tank, Magenta	75.0	75.0
10	101100.0	Dive Mask, Small Clear	0.0	500.0
11	101200.0	Dive Mask, Med Clear	50.0	500.0
12	201000.0	Half-dome Tent	10.0	250.0
13	202000.0	Half-dome Tent Vestibule	1.0	250.0
14	301000.0	Light Fly Climbing Harness	250.0	250.0
15	302000.0	Locking Carabiner, Oval	1250.0	0.0
16	100100.0	Std. Scuba Tank, Yellow	100.0	0.0
17	100200.0	Std. Scuba Tank, Magenta	100.0	100.0
18	101100.0	Dive Mask, Small Clear	300.0	200.0
19	101200.0	Dive Mask, Med Clear	475.0	0.0
20	201000.0	Half-dome Tent	250.0	0.0
21	202000.0	Half-dome Tent Vestibule	100.0	0.0
22	301000.0	Light Fly Climbing Harness	0.0	250.0
23	302000.0	Locking Carabiner, Oval	500.0	500.0
24	100100.0	Std. Scuba Tank, Yellow	200.0	0.0
25	100200.0	Std. Scuba Tank, Magenta	250.0	0.0
26	101100.0	Dive Mask, Small Clear	450.0	0.0
27	101200.0	Dive Mask, Med Clear	250.0	250.0
28	201000.0	Half-dome Tent	0.0	250.0
29	202000.0	Half-dome Tent Vestibule	0.0	200.0
30	301000.0	Light Fly Climbing Harness	0.0	250.0
31	302000.0	Locking Carabiner, Oval	0.0	1000.0
32	NaN	None	NaN	NaN