

1. (10%) In plain English, please state what the following query computes:

$$\pi_{Customer_Name}(\pi_{Customer_ID}(\pi_{Order_ID}(\sigma_{Product_Name='Computer_Desk'}(Order_Line_t \bowtie Product_t) \bowtie Order_t) \bowtie Customer_t))$$

This query selects the “Customer name” who ordered “Computer desk”. Using Customer_ID, Order_ID, Product_ID as keys to link between Customer_t and Order_t, Order_t and Order_line_t, Order_line_t and Product_t, respectively.

2. (15%) Using an expression that involves a division operator (see Slide #20, Chapter 4), compose a relational algebra expression that would show customer name who submitted an order that requested both of the following products in the *same* order: End Table, Coffee Table.

$$\begin{aligned} & \rho(TempA, \pi_{Order_ID, Product_ID}(Order_Line_t)) \\ & \rho(TempB, \pi_{Product_ID}(\sigma_{Product_Name='EndTable' \vee 'CoffeeTable'}(Product_t))) \\ & \rho(TempC, \pi_{Customer_ID}(TempA / TempB \bowtie Order_t)) \\ & \pi_{Customer_Name}(TempC \bowtie Customer_t) \end{aligned}$$

3. (10%) Compose a SQL statement that is equivalent to Question 1 above. Note, you might want to try and execute this SQL statement against the NewPVF database in SQL Server 2014 to see if it works as intended.

```

1  /* Question 3*/
2  SELECT Customer_name
3  FROM Customer_t
4  WHERE Customer_ID IN
5      (
6          SELECT Customer_ID
7          FROM Order_t
8          WHERE Order_ID IN
9              (
10                 SELECT Order_ID
11                 FROM Order_line_t
12                 WHERE Product_ID IN
13                     (
14                         SELECT Product_ID
15                         FROM Product_t
16                         WHERE Product_Name = 'Computer Desk'
17                     )
18             )
19      )

```

Results Messages

	Customer_name
1	Contemporary Casuals
2	Eastern Furniture
3	Impressions
4	California Classics
5	American Euro Lifestyles
6	Battle Creek Furniture

4. (17.5%) Compose a SQL statement that is equivalent to Question 2 above. Note, you might want to try and execute this SQL statement against the NewPVF database in SQL Server 2014 to see if it works as intended [hint: use nested queries with correlation, like the first query on Slide #15, Chapter 5].

```
22  /*Question 4*/
23  SELECT C.Customer_Name
24  FROM Customer_t as C INNER JOIN Order_t as O on C.Customer_ID = O.Customer_ID
25  WHERE NOT EXISTS
26  (
27      (
28          SELECT P.Product_ID
29          FROM Product_t as P
30          WHERE Product_Name IN ('End Table', 'Coffee Table')
31      )
32  EXCEPT
33      (
34          SELECT OL.Product_ID
35          FROM Order_Line_t as OL
36          WHERE OL.Order_ID = O.Order_ID
37      )
38  )
```

Results Messages

	Customer_Name
1	Contemporary Casuals
2	American Euro Lifestyles

5. (15%) Compose a query that would find the customer who had the highest total amount (in dollars) of purchases in year 2011. Show customer name and total dollars amount of the purchases in the result table.

```
41  /*Question 5*/
42  SELECT Top 1 Customer_Name, SUM(Quantity*P.Unit_Price) as Total_Purchases
43  FROM Order_t as O, Order_Line_t as OL, Customer_t as C, Product_t as P
44  WHERE C.Customer_ID = O.Customer_ID AND O.Order_ID = OL.Order_ID AND OL.Product_ID = P.Product_ID AND
45  O.Order_Date BETWEEN '1/1/2011' and '12/31/2011'
46  GROUP BY Customer_Name
47  ORDER BY Total_Purchases Desc;
48
```

Results Messages

	Customer_Name	Total_Purchases
1	Contemporary Casuals	6275.0000

6. (17.5%) Compose an SQL statement to generate a list of two least expensive vendors (suppliers) for each raw material. In the result table, show the following columns: material ID, material description, vendor ID, vendor name, and the supplier's unit price.

Sort the result table by material ID and supplier's unit price in ascending order. Note: If a raw material has only one vendor (supplier), that supplier and its unit price for the raw material should also be in the result (output) table *[hint: use nested queries with correlation]*.

*/*Question 6*/*

```
SELECT R.Material_ID, R.Material_Description, V.Vendor_ID, V.Vendor_name, S.Unit_Price
FROM (Vendor_t as V inner join Supplies_t as S ON V.Vendor_ID = S.Vendor_ID)
INNER JOIN Raw_Materials_t as R ON S.Material_ID = R.Material_ID
WHERE S.Unit_Price IN
(
    Select Top 2 S2.Unit_Price
    From Supplies_t as S2
    Where S2.Material_ID = R.Material_ID
    Order By S2.Unit_Price
)
Order By R.Material_ID, S.Unit_Price
```

	Material_ID	Material_Description	Vendor_ID	Vendor_name	Unit_Price
1	1	Red Oak	1	L & M Hardwoods	7.8900
2	1	Red Oak	3	McKnight Hardware	8.0000
3	2	Red Oak	1	L & M Hardwoods	12.0500
4	2	Red Oak	6	McEwen Lumber Co.	12.5000
5	3	Red Oak	1	L & M Hardwoods	13.6700
6	3	Red Oak	6	McEwen Lumber Co.	15.0000
7	4	Red Oak	3	McKnight Hardware	7.5000
8	4	Red Oak	1	L & M Hardwoods	7.6600
9	5	White Oak	4	Seattle Forest Products	7.2300
10	6	White Oak	4	Seattle Forest Products	15.1900
11	7	Walnut	4	Seattle Forest Products	13.0200
12	8	Walnut	4	Seattle Forest Products	15.4500
13	9	Walnut	4	Seattle Forest Products	10.8800
14	10	Natural Ash	4	Seattle Forest Products	15.5500
15	10	Natural Ash	5	Axley Lumber Co.	16.0000
16	11	White Ash	4	Seattle Forest Products	8.7900
17	12	White Ash	L & M Hardwoods	Axley Lumber Co.	14.2600
18	13	Cherry	5	Axley Lumber Co.	15.8200
19	14	Cherry	5	Axley Lumber Co.	13.7500
20	15	Cherry	5	Axley Lumber Co.	16.7200
21	16	Natural Maple	2	Building Blocks	5.7000
22	17	Natural Maple	2	Building Blocks	7.9500
23	18	Natural Maple	2	Building Blocks	11.1300
24	19	Natural Maple	2	Building Blocks	10.4700
25	20	Natural Maple	2	Building Blocks	9.1800

7. (15%) In the previous assignment (Assignment #1), you were asked to design a data model for a database that would store patient's information, HCC codes, patient assigned HCC codes, and HCC code business rules. Based on that database design, I have created a physical database (named HccDB) with some sample data for each table. After Assignment #1 due date, I plan to upload a backup file of HccDB to the Blackboard; and you should be able to download it from the Blackboard and restore it on your computer. This database includes the following (hypothetical) short list of HCC codes: 8, 9, 10, 11, 12, 17, 18, 19, 50, and 51. Moreover, the following business rules for HCC codes are also included in the database:

- a. Code 8 dominates codes 9, 10, 11, and 12. Code 9 dominates codes 10, 11, and 12. Code 10 dominates codes 11 and 12. Code 11 dominates code 12.
- b. Code 17 dominates codes 18 and 19. Code 18 dominates code 19.
- c. Code 50 dominates code 51.

For this question, using the restored HccDB database, your task is to compose a query so that if a patient is assigned one or more HCC codes, the resulting query shows only the patient's relevant dominant HCC code(s) plus any of the patient's assigned HCC codes that has no corresponding dominant code. For examples:

- a. Assume that based on John Doe's medical conditions, he is assigned 8, 9, and 11 HCC codes. The resulting query for John Doe should display his patient id, his name, and his relevant HCC code, which is 8 in this simple example.
- b. Assume that based on Jane Public's medical conditions, she is assigned 9, 11, 12, 17, 18, and 19 HCC codes. The resulting query for Jane Public should display her patient id, her name, and her relevant HCC codes, which are 9 and 17.
- c. Assume that based on Dave Plumber's medical conditions, he is assigned 12, 18, 19, and 51 HCC codes. The resulting query for Dave Plumber should display his patient id, his name, and his relevant HCC codes, which are 12, 18, and 51.
- d. Assume that Ms. Healthie has no medical conditions; thus, she is NOT assigned any HCC code. The resulting query should NOT display Ms. Healthie.
- e. Putting the four examples above together, your query should display a result like this table:

Patient_ID	Patient_Name	Relevant_HccCode
1	John Doe	8
2	Jane Public	9
2	Jane Public	17
3	Dave Plumber	12
3	Dave Plumber	18
3	Dave Plumber	51

```

64  /*Question 7*/
65  SELECT PH.PatientID, P.Name, PH.HccCode
66  FROM Patient as P
67  JOIN PatientHcc as PH ON P.ID = PH.PatientID
68  EXCEPT
69      (
70      SELECT PH.PatientID, P.Name, BR.HccCode
71      FROM Patient as P
72      JOIN PatientHcc as PH ON P.ID = PH.PatientID
73      JOIN BizRule as BR ON PH.HccCode = BR.MainHccCode
74      )
75

```

Results

Messages

	PatientID	Name	HccCode
1	1	John Doe	8
2	2	Jane Public	9
3	2	Jane Public	17
4	3	Dave Plumber	12
5	3	Dave Plumber	18
6	3	Dave Plumber	51