Adventure Works 2016 SQL Challenge Problems

1. Select the SalesOrderID, OrderQty, UnitPrice, LineTotal from the table [Sales.SalesOrderDetail]. Make sure that the only SalesOrderID is 43660.
   Final Result:

| | SalesOrderID | OrderQty | UnitPrice | LineTotal |
|---|---|---|---|---|
| 1 | 43660 | 1 | 419.4589 | 419.458900 |
| 2 | 43660 | 1 | 874.794 | 874.794000 |

2. Select the Name and Unit Measure Code for the Name “Cubic Foot” in table [Production.UnitMeasure].
   Final Result:

| | Name | UnitMeasureCode |
|---|---|---|
| 1 | Cubic foot | FT3 |

3. Select the BusinessEntityID, JobTitle, Gender, MaritalStatus, and VacationHours from the table [HumanResources.Employee]. Make sure the only Gender selected is “F” and VacationHours are greater than 92.
   Final Result:

| | BusinessEntityID | JobTitle | Gender | MaritalStatus | VacationHours |
|---|---|---|---|---|---|
| 1 | 88 | Production Technician - WC10 | F | S | 99 |
| 2 | 90 | Production Technician - WC10 | F | S | 97 |
| 3 | 91 | Production Technician - WC10 | F | S | 95 |
| 4 | 104 | Production Technician - WC10 | F | M | 94 |
| 5 | 109 | Production Technician - WC50 | F | M | 95 |
| 6 | 113 | Production Technician - WC50 | F | M | 96 |
| 7 | 122 | Stocker | F | S | 97 |
| 8 | 124 | Stocker | F | S | 98 |

4. Find the BusinessEntityID, PhoneNumber, and ModifiedDate for the people with phone numbers that start with “608” from the table [Person.PersonPhone]
   Final Result:

| | BusinessEntityID | PhoneNumber | ModifiedDate |
|---|---|---|---|
| 1 | 443 | 608-555-0114 | 2013-07-31 00:00:00.000 |
| 2 | 5781 | 608-555-0117 | 2013-08-14 00:00:00.000 |
| 3 | 11000 | 608-555-0117 | 2013-12-14 00:00:00.000 |
| 4 | 3665 | 608-555-0125 | 2013-07-04 00:00:00.000 |
| 5 | 3460 | 608-555-0126 | 2013-07-02 00:00:00.000 |
| 6 | 1241 | 608-555-0162 | 2012-06-30 00:00:00.000 |
| 7 | 8586 | 608-555-0170 | 2013-09-05 00:00:00.000 |

5. Select the CreditCardID, CardType, CardNumber, ExpMonth, and ExpYear from the table [Sales.CreditCard]. We only need “Vista” cards and would like to see cards with expiration dates between 2005 and 2006. Lastly, we only need cards that contain the digits “5246” within its CardNumber.
   Final Result:

| | CreditCardID | CardType | CardNumber | ExpMonth | ExpYear |
|---|---|---|---|---|---|
| 1 | 14318 | Vista | 11111957524635 | 2 | 2005 |
| 2 | 11986 | Vista | 11115246457023 | 12 | 2006 |

6. Select the SalesTaxRateID, TaxType, and TaxRate from the table [Sales.SalesTaxRate]. We are looking for records with a TaxType of 2 but if a record’s TaxRate is 7.00 select it anyways. Lastly, we prefer to not see anything with the Name “Massachusetts State Sales Tax”.
   Final Result:

| | SalesTaxRateID | TaxType | TaxRate | Name |
|---|---|---|---|---|
| 1 | 4 | 2 | 7.00 | Canadian GST |
| 2 | 5 | 2 | 7.00 | Canadian GST |
| 3 | 6 | 2 | 7.00 | Canadian GST |
| 4 | 7 | 3 | 7.00 | Canadian GST |
| 5 | 8 | 3 | 7.00 | Canadian GST |
| 6 | 9 | 3 | 7.00 | Canadian GST |
| 7 | 10 | 3 | 7.00 | Canadian GST |
| 8 | 11 | 3 | 7.00 | Canadian GST |
| 9 | 12 | 3 | 7.00 | Canadian GST |
| 10 | 13 | 3 | 7.00 | Canadian GST |
| 11 | 16 | 3 | 7.00 | Canadian GST |
| 12 | 17 | 3 | 7.00 | Canadian GST |

7. Select the BusinessEntityID, JobTitle, Gender, MaritalStatus, and VacationHours from the table [HumanResources.Employee]. We only want to see JobTitles that are “Marketing” and the VacationHours go from low to high. We also only want the top 5 records.
   Final Result:

| | BusinessEntityID | JobTitle | Gender | MaritalStatus | VacationHours |
|---|---|---|---|---|---|
| 1 | 16 | Marketing Manager | M | S | 40 |
| 2 | 20 | Marketing Assistant | F | M | 41 |
| 3 | 17 | Marketing Assistant | M | S | 42 |
| 4 | 19 | Marketing Assistant | F | S | 43 |
| 5 | 21 | Marketing Specialist | M | M | 44 |

8. Pull the Name , AccountNumber, CreditRating, and PurchasingWebServiceURL from the table [Purchasing.Vendor]. We only want those with a CreditRating of 2 and to have any PurchasingWebServiceURLs that are not null.
   Final Result:

| | Name | AccountNumber | CreditRating | PurchasingWebServiceURL |
|---|---|---|---|---|
| 1 | Wide World Importers | WIDEWOR0001 | 2 | www.wideworldimporters.com/ |

9. Select the TransactionType and make 2 new columns called “AverageQuantity” and AverageActualCost pulled from the table [Production.TransactionHistoryArchive]. We would like this to be aggregate calculations for each TransactionType.
   Final Result:

| | TransactionType | AverageQuantity | AverageActualCost |
|---|---|---|---|
| 1 | W | 64 | 0.00 |
| 2 | S | 2 | 756.6514 |
| 3 | P | 246 | 35.2535 |

10. Select the BusinessEntityID and PurchasingWebServiceURL from the table [Purchasing.Vendor]. In addition, select the AccountNumber and call it “UpdatedAN” however, we do not want the follow “0001” within each value but want it to be instead “#1”. Lastly, order it by BusinessEntityID ascending.
    Final Result:

| | BusinessEntityID | UpdatedAN | PurchasingWebServiceURL |
|---|---|---|---|
| 1 | 1580 | LITWARE#1 | www.litwareinc.com/ |
| 2 | 1584 | TREYRE#1 | www.treyresearch.net/ |
| 3 | 1596 | ADATUM#1 | www.adatum.com/ |
| 4 | 1606 | NORTHW#1 | www.northwindtraders.com/ |
| 5 | 1648 | WIDEWOR#1 | www.wideworldimporters.com/ |
| 6 | 1678 | PROSE#1 | www.proseware.com/ |

11. Build a Scalar Function called “CalculateAge” that calculates the age of any given employee in a query. Let’s assume it’s possible that an employee can be under the age of 1. If so, make sure the function rounds to 1 no matter if they are above or below a month old. After, query the BusinessEntityID, JobTitle, Gender, BirthDate, dbo.CalculateAge(BirthDate) as “AGE” from the table [HumanResources.Employee].
Final Result: (snippet of full result)

| | BusinessEntityID | JobTitle | Gender | BirthDate | AGE |
|---|---|---|---|---|---|
| 1 | 1 | Chief Executive Officer | M | 1969-01-29 | 52 |
| 2 | 2 | Vice President of Engineering | F | 1971-08-01 | 50 |
| 3 | 3 | Engineering Manager | M | 1974-11-12 | 46 |
| 4 | 4 | Senior Tool Designer | M | 1974-12-23 | 46 |
| 5 | 5 | Design Engineer | F | 1952-09-27 | 68 |
| 6 | 6 | Design Engineer | M | 1959-03-11 | 62 |
| 7 | 7 | Research and Development Manager | M | 1987-02-24 | 34 |
| 8 | 8 | Research and Development Engineer | F | 1986-06-05 | 35 |
| 9 | 9 | Research and Development Engineer | F | 1979-01-21 | 42 |
| 10 | 10 | Research and Development Manager | M | 1984-11-30 | 36 |
| 11 | 11 | Senior Tool Designer | M | 1978-01-17 | 43 |
| 12 | 12 | Tool Designer | M | 1959-07-29 | 62 |
| 13 | 13 | Tool Designer | F | 1989-05-28 | 32 |
| 14 | 14 | Senior Design Engineer | M | 1979-06-16 | 42 |

12. Select the SpecialOfferID and the Type from table [Sales.SpecialOffer]. Add an additional column called “MaxDP” that calculates a running total sum for DiscountPct for each Type and is ordered by SpecialOfferID ascending.
Final Result:

| | SpecialOfferID | Type | MaxDP |
|---|---|---|---|
| 1 | 7 | Discontinued Product | 0.35 |
| 2 | 16 | Discontinued Product | 0.75 |
| 3 | 9 | Excess Inventory | 0.30 |
| 4 | 10 | Excess Inventory | 0.80 |
| 5 | 12 | Excess Inventory | 1.15 |
| 6 | 13 | New Product | 0.15 |
| 7 | 14 | New Product | 0.35 |
| 8 | 1 | No Discount | 0.00 |
| 9 | 8 | Seasonal Discount | 0.10 |
| 10 | 11 | Seasonal Discount | 0.25 |
| 11 | 15 | Seasonal Discount | 0.75 |
| 12 | 2 | Volume Discount | 0.02 |
| 13 | 3 | Volume Discount | 0.07 |
| 14 | 4 | Volume Discount | 0.17 |
| 15 | 5 | Volume Discount | 0.32 |
| 16 | 6 | Volume Discount | 0.52 |

13. Create a Scalar Function called “CPN” that conditionally replaces the values in column ProductNumber from the table [Production.Product]. If a value in ProductNumber starts with “HN” replace with “HN”, if “LI” replace with “LI”, if “LN” replace with “LN”, and everything else should be replaced with “IR”.

    Once completed, query a table that includes the CPN of the ProductNumber column with a count row called “CountPN” and a row that calculates the average of SafetyStockLevel
    Final Result:

| | CPN | CountPN | AVGSSL |
|---|---|---|---|
| 1 | HN | 23 | 1000 |
| 2 | IR | 448 | 477 |
| 3 | LI | 10 | 1000 |
| 4 | LN | 23 | 1000 |

14. Create a Common Table Expression that pulls the City, AddressLine1, and type of Location (Name) from the tables [Person.BusinessEntityAddress], [Person.AddressType], and [Person.Address]. After, query the type of Location titled ‘Location’ then combine the AddressLine1 and the City with a separation of ‘, ‘ into a column titled ‘Address’. Filter the data so that only the Location with ‘Shipping’ and the City named ‘Renton’ is shown.
    Final Result:

| | Location | Address |
|---|---|---|
| 1 | Shipping | 7943 Walnut Ave, Renton |
| 2 | Shipping | 3770 Viewpoint Ct, Renton |
| 3 | Shipping | 5297 Algiers Drive, Renton |

15. Create a Common Table Expression that pulls in every product starting with “Road-650”, the location of where the part is assembled, and the unit price for each part at each of its assembly locations. This information can be found in the following tables: [Production.Product], [Production.ProductInventory], [Production.ProductModelProductDescriptionCulture], [Production.Location], [Sales.SalesOrderDetail]

    Once the Common Table Expression has been made, calculate the Average Unit Price or [AVG MSRP] for each product name in each location and only show any AVG MSRP greater than or equal to 500. Lastly, have it be in order of the AVG MSRP descending.
    Final Result:

| | Product Name | Location | AVG MSRP |
|---|---|---|---|
| 1 | Road-650 Red, 58 | Finished Goods Storage | 532.9409 |
| 2 | Road-650 Red, 58 | Final Assembly | 532.9409 |
| 3 | Road-650 Black, 62 | Finished Goods Storage | 517.6857 |
| 4 | Road-650 Black, 62 | Final Assembly | 517.6857 |
| 5 | Road-650 Black, 48 | Finished Goods Storage | 513.2344 |
| 6 | Road-650 Black, 48 | Final Assembly | 513.2344 |

ANSWERS:

1. SELECT SalesOrderID, OrderQty, UnitPrice, LineTotal
FROM Sales.SalesOrderDetail
WHERE SalesOrderID = 43660;

2. SELECT Name, UnitMeasureCode
FROM Production.UnitMeasure
WHERE Name = 'Cubic Foot';

3. SELECT BusinessEntityID, JobTitle, Gender, MaritalStatus, VacationHours
FROM HumanResources.Employee
WHERE Gender = 'F' AND VacationHours > 92

4. SELECT BusinessEntityID, PhoneNumber, ModifiedDate
FROM Person.PersonPhone
WHERE PhoneNumber LIKE '608%';

5. SELECT CreditCardID, CardType, CardNumber, ExpMonth, ExpYear
FROM Sales.CreditCard
WHERE CardType = 'Vista' AND
CardNumber LIKE '%5246%' AND
ExpYear BETWEEN 2005 AND 2006;

6. SELECT SalesTaxRateID, TaxType, TaxRate, Name
FROM Sales.SalesTaxRate
WHERE TaxType = 2 OR
TaxRate = 7.00 AND
Name != 'Massachusetts State Sales Tax';

7. SELECT TOP 5 BusinessEntityID, JobTitle, Gender, MaritalStatus, VacationHours
FROM HumanResources.Employee
WHERE JobTitle LIKE 'Marketing%'
ORDER BY VacationHours ASC ;

8. SELECT [Name], AccountNumber, CreditRating, PurchasingWebServiceURL
FROM Purchasing.Vendor
WHERE CreditRating = 2 AND
PurchasingWebServiceURL IS NOT NULL

9. SELECT TransactionType, AVG(Quantity) AS AverageQuantity,
AVG(ActualCost) AS AverageActualCost
FROM Production.TransactionHistoryArchive
GROUP BY TransactionType ;

10.SELECT BusinessEntityID,
REPLACE(AccountNumber, '0001', '#1') AS UpdatedAN, PurchasingWebServiceURL,
FROM Purchasing.Vendor
WHERE PurchasingWebServiceURL IS NOT NULL
ORDER BY BusinessEntityID

11. CREATE FUNCTION CalculateAge (@DOB DATE)
RETURNS INT
AS
BEGIN
DECLARE @Age INT

SET @Age = DATEDIFF(Year, @DOB, GETDATE()) -
CASE
WHEN (MONTH(@DOB) > MONTH(GETDATE())) OR
(MONTH(@DOB) = MONTH(GETDATE()) AND
DAY(@DOB) > DAY(GETDATE()))
THEN 1
ELSE 0
END
RETURN @Age
END

SELECT BusinessEntityID, JobTitle, Gender, BirthDate, dbo.CalculateAge(BirthDate) AS AGE
FROM HumanResources.Employee

12. SELECT SpecialOfferID, Type,
SUM(DiscountPct) OVER (PARTITION BY Type ORDER BY SpecialOfferID ROWS
BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS MaxDP
FROM Sales.SpecialOffer

13.
```
CREATE FUNCTION CPN (@PN NVARCHAR(25))
        RETURNS NVARCHAR(25)
        AS
        BEGIN
        DECLARE @NPN NVARCHAR(25)

        SET @NPN = CASE
                        WHEN @PN LIKE 'HN%' THEN 'HN'
                        WHEN @PN LIKE 'LI%' THEN 'LI'
                        WHEN @PN LIKE 'LN%' THEN 'LN'
                        ELSE 'IR'
                END
        RETURN @NPN
        END

SELECT dbo.CPN(ProductNumber) AS CPN, COUNT(ProductNumber) AS CountPN,
AVG(SafetyStockLevel) AS AVGSSL
FROM Production.Product
GROUP BY dbo.CPN(ProductNumber)
```

14.
```
WITH CTE AS (
        SELECT c.City, c.AddressLine1 AS Address, b.Name AS Location
        FROM Person.BusinessEntityAddress a
        JOIN Person.AddressType b
        ON a.AddressTypeID = b.AddressTypeID
        JOIN Person.Address c
        ON a.AddressID = c.AddressID)

SELECT Location, CONCAT(Address, ', ', City) AS Address
FROM CTE
WHERE Location = 'Shipping' AND City = 'Renton'
```

15. WITH CTE AS

```
(SELECT prod.Name [Product Name], loc.Name [Location],
SOD.UnitPrice AS  [MSRP]
FROM Production.Product prod
JOIN Production.ProductInventory PI
ON prod.ProductID = PI.ProductID
JOIN Production.ProductModelProductDescriptionCulture Cul
ON prod.ProductModelID = Cul.ProductModelID
JOIN Production.Location loc
ON PI.LocationID = loc.LocationID
JOIN Sales.SalesOrderDetail SOD
ON prod.ProductID = SOD.ProductID
WHERE prod.Name LIKE 'Road-650%')

SELECT [Product Name], Location, AVG(MSRP) [AVG MSRP] FROM CTE
GROUP BY [Product Name], Location
HAVING AVG(MSRP) >= 500
ORDER BY [AVG MSRP] DESC
```