

/\*-----

## Part A

The first thing that the sales manager would like to know is the total sales revenue associated with each sales territory. Create a report showing the total sales revenue broken out by territory name. Round the revenue figures to the nearest dollar and sort the list so that the highest revenue territory appears first.

```
SELECT B.Name TerritoryName,
       FORMAT(CAST(ROUND(SUM(SubTotal),2) AS MONEY), 'C2') SalesRevenue
FROM Sales.SalesOrderHeader A
     LEFT OUTER JOIN Sales.SalesTerritory B
       On A.TerritoryID = B.TerritoryID

GROUP BY B.Name
ORDER BY SUM(SubTotal) DESC;
```

The screenshot shows the Azure Data Studio interface with a SQL query executed against the AdventureWorks2014 database. The query calculates the total sales revenue by territory, rounded to the nearest dollar, and sorted in descending order of revenue. The results are displayed in a table with 8 rows.

	TerritoryName	SalesRevenue
1	Southwest	\$24,184,609.60
2	Canada	\$16,355,770.46
3	Northwest	\$16,084,942.55
4	Australia	\$10,655,335.96
5	Central	\$7,909,009.01
6	Southeast	\$7,879,655.07
7	United Kingdom	\$7,670,721.04
8	France	\$7,251,555.65

The interface also shows the SQL query text in the editor, which includes comments about the task and the query structure. The status bar at the bottom indicates the current line and column (Ln 14, Col 1) and the number of rows returned (10 rows).

/\*-----\*/

## Part B

The sales manager would now like to drill into the 2013 sales results. Revise your query from Part A to focus only on results from 2013 and break the results down by month. Your result set should include four columns: Name (of the territory), Month, Year, and SalesRevenue. This time, sort the list first by territory name and then by month.

```
SELECT B.Name TerritoryName,
       DATEPART(yyyy, A.OrderDate) AS SalesYear,
       DATEPART(MM, A.OrderDate) AS SalesMonth,
       FORMAT(CAST(ROUND(SUM(SubTotal),2) AS money), 'C2') SalesRevenue
FROM Sales.SalesOrderHeader A
     LEFT OUTER JOIN Sales.SalesTerritory B
       ON A.TerritoryID = B.TerritoryID
WHERE DATEPART(yyyy, A.OrderDate) = 2013
GROUP BY B.Name,
         DATEPART(yyyy, A.OrderDate) ,
         DATEPART(MM, A.OrderDate)
ORDER BY 1,2,3;
```

The screenshot shows the Azure Data Studio interface. The top pane displays a SQL query that filters for the year 2013 and groups results by territory name, year, and month. The bottom pane shows the results of this query as a table with 8 rows, all for the territory 'Australia'.

	TerritoryName	SalesYear	SalesMonth	SalesRevenue
1	Australia	2013	1	\$222,513.72
2	Australia	2013	2	\$177,688.06
3	Australia	2013	3	\$215,687.85
4	Australia	2013	4	\$205,166.13
5	Australia	2013	5	\$252,418.64
6	Australia	2013	6	\$380,028.94
7	Australia	2013	7	\$456,778.32
8	Australia	2013	8	\$394,785.63

At the bottom of the interface, a status bar indicates 'Ln 23, Col 1 (484 selected)', 'Tab Size: 4', 'UTF-8', 'CRLF', 'SQL', 'MSSQL', '120 rows', and '00:00:00'. The taskbar at the very bottom shows the time as 09:50 PM on 06/18/2020.

/\*-----\*/

### Part C

The sales manager would like to recognize territories that had at least one month in 2013 with monthly sales over \$750,000. Create a report listing these territories. It should contain only a single column named AwardWinners and should be sorted alphabetically. Territories can only win this award once, regardless of the number of months where they exceeded the threshold, so don't include duplicate values in your list.

```
SELECT DISTINCT TerritoryName
FROM (
    SELECT B.Name TerritoryName,
           DATEPART(yyyy, A.OrderDate) AS SalesYear,
           DATEPART(MM, A.OrderDate) AS SalesMonth,
           FORMAT(CAST(ROUND(SUM(SubTotal),2) AS money), 'C') SalesRevenue
    FROM Sales.SalesOrderHeader A
         LEFT OUTER JOIN Sales.SalesTerritory B
           ON A.TerritoryID = B.TerritoryID
   WHERE DATEPART(yyyy, A.OrderDate) = 2013
   GROUP BY B.Name,
            DATEPART(yyyy, A.OrderDate) ,
            DATEPART(MM, A.OrderDate)
   HAVING ROUND(SUM(A.SubTotal),2) >= 750000
) X;
```

The screenshot shows the Azure Data Studio interface with a SQL query executed in the 'SQLQuery\_3 - sql.da...tudent)' window. The query is a T-SQL statement that filters for territories in 2013 with total monthly sales of at least \$750,000. The results pane shows a table with four rows of territory names.

	TerritoryName
1	Canada
2	France
3	Northwest
4	Southwest

The status bar at the bottom indicates the cursor is at line 47, column 1, with 564 characters selected. The window title is 'sql.datascience.nd.edu : AdventureWorks2014'.

/\*-----\*/

#### Part D

The sales manager would like to offer additional training to those territories that did not achieve the \$750,000 at least once in 2013. Create a list of territories that did not achieve their target, sorted alphabetically. Do this by using set theory operators, adding a few lines to your query from Part C.

-----\*/

```
SELECT DISTINCT
    Name TerritoryName
FROM Sales.SalesTerritory
EXCEPT
SELECT DISTINCT TerritoryName
FROM
    (
        SELECT B.Name TerritoryName,
            DATEPART(yyyy, A.OrderDate) AS SalesYear,
            DATEPART(MM, A.OrderDate) AS SalesMonth,
            FORMAT(CAST(ROUND(SUM(SubTotal),2) AS money), 'C2') SalesRevenue
        FROM Sales.SalesOrderHeader A
        LEFT OUTER JOIN Sales.SalesTerritory B
            ON A.TerritoryID = B.TerritoryID
        WHERE DATEPART(yyyy, A.OrderDate) = 2013
        GROUP BY B.Name,
            DATEPART(yyyy, A.OrderDate),
            DATEPART(MM, A.OrderDate)
        HAVING ROUND(SUM(A.SubTotal),2) >= 750000
    ) X
ORDER BY 1;
```

The screenshot shows the Azure Data Studio interface with a SQL query executed against the AdventureWorks2014 database. The query uses an EXCEPT operator to find territories that did not achieve their target revenue in 2013. The results pane shows a table with 6 rows, listing the territories: Australia, Central, Germany, Northeast, Southeast, and United Kingdom.

	TerritoryName
1	Australia
2	Central
3	Germany
4	Northeast
5	Southeast
6	United Kingdom

/\*-----

## Problem 2

Your next assignment is to work with the production manager on a series of questions that he has about your product catalog.

/\*-----

### Part A

The production manager is considering dropping some products and would like to generate a report that lists finished goods products where the total sales volume is under 50 units.

Create a report showing the product name and the total number of units ordered.

```
SELECT B.Name          AS ProductName,
       SUM(A.OrderQty)  AS SoldUnits
FROM   Sales.SalesOrderDetail A
       LEFT OUTER JOIN Production.Product B ON A.ProductID = B.ProductID
GROUP BY B.Name
HAVING SUM(A.OrderQty) < 50
ORDER BY B.Name;
```

The screenshot shows the Azure Data Studio interface with a SQL query executed against the AdventureWorks2014 database. The query is displayed in the editor, and the results are shown in the Results pane below. The query is a SELECT statement that joins Sales.SalesOrderDetail (A) with Production.Product (B) on ProductID, groups by Product Name, and filters for total sales volume (SUM of OrderQty) less than 50 units. The results pane shows 10 rows of data, including product names and the total units sold.

	ProductName	SoldUnits
1	HL Mountain Frame - Black, 44	17
2	LL Mountain Frame - Black, 40	8
3	LL Mountain Frame - Black, 52	15
4	LL Mountain Frame - Silver, ...	44
5	LL Road Seat/Saddle	10
6	LL Touring Frame - Blue, 44	25
7	LL Touring Frame - Blue, 58	4
8	LL Touring Frame - Blue, 62	15

/\*-----\*/

## Part B

The production manager is working on profit projections for the coming year. She needs to make some assumptions about the sales tax rates in various countries and wants to use a conservatively selected single rate for each country.

Create a report showing the maximum sales tax rate for each country served by AdventureWorks. Sort the list by tax rate with the highest tax rate first.

-----\*/

```
SELECT C.Name          AS CountryRegionName,
       MAX(TaxRate) AS MaxTaxRate
FROM Sales.SalesTaxRate A
     LEFT OUTER JOIN Person.StateProvince B
       ON A.StateProvinceID = B.StateProvinceID
     LEFT OUTER JOIN Person.CountryRegion C
       ON B.CountryRegionCode = C.CountryRegionCode
GROUP BY C.NAME
ORDER BY 2 DESC;
```

The screenshot shows the Azure Data Studio interface with a SQL query executed against the AdventureWorks2014 database. The query is designed to find the maximum sales tax rate for each country region. The results pane displays a table with 6 rows, sorted by the maximum tax rate in descending order.

	CountryRegionName	MaxTaxRate
1	France	19.6000
2	United Kingdom	17.5000
3	Germany	16.0000
4	Canada	14.2500
5	Australia	10.0000
6	United States	8.8000

The bottom status bar indicates the current position is at line 125, column 1, with 310 characters selected. The interface also shows various toolbars and a taskbar at the bottom with system icons and the date/time (09:51 PM, 06/18/2020).

/\*-----\*/

## Part C

The quality assurance department is issuing a recall for all helmets shipped to stores during the first five days of February 2014 due to a safety issue. They are already handling the notification of individual consumers where records are available. The sales department needs your help to follow up with stores that received the recalled helmets. Produce a listing of stores that received helmets shipped during the recall period. Your result should have two columns:

StoreName  
TerritoryName

Sort the list in alphabetical order, first by territory name and then by store name. Each store should only appear once in the list, regardless of the number of shipments they received.

-----\*/

```
SELECT DISTINCT
    C.Name TerritoryName,
    D.Name StoreName
FROM Sales.SalesOrderHeader A
    INNER JOIN Sales.Customer      B ON A.CustomerID = B.CustomerID
    INNER JOIN Sales.SalesTerritory C ON B.TerritoryID = C.TerritoryID
    INNER JOIN Sales.Store         D ON B.StoreID     = D.BusinessEntityID
    INNER JOIN Sales.SalesOrderDetail E ON A.SalesOrderID = E.SalesOrderID
    INNER JOIN Production.Product   F ON E.ProductID  = F.ProductID
WHERE A.ShipDate BETWEEN '2014-02-01' AND '2014-02-05'
    AND UPPER(F.Name) LIKE '%HELMET%'
ORDER BY 1, 2;
```

The screenshot shows the SQL Data Studio interface. The query editor at the top displays the SQL code from the previous block. Below the editor, the 'Results' pane shows the output of the query. The results are displayed in a table with two columns: 'TerritoryName' and 'StoreName'. The table contains 8 rows of data, sorted alphabetically by territory name and then by store name. The status bar at the bottom indicates that the query executed successfully, returning 36 rows.

	TerritoryName	StoreName
1	Australia	Bike Part Wholesalers
2	Australia	Nationwide Supply
3	Australia	Popular Bike Lines
4	Canada	Corner Bicycle Supply
5	Canada	Odometers and Accessories Co.
6	Canada	Petroleum Products Distribut...
7	Canada	Rapid Bikes
8	Canada	Vigorous Exercise Company