```sql
/* Scenario 1- Sort a result set by one column in ascending or descending order */

--Order the country names in ascending order from A-Z
select * from Person.CountryRegion
order by Name ASC

--Order the country names in descending order from Z-A
select * from Person.CountryRegion
order by Name DESC

/* Scenario 2- Sort a result set by an expression */

-- List the "comments with more words" to get more insight about a product
select ProductID, ReviewerName, Rating, Comments
from Production.ProductReview
order by len(Comments) DESC

/* Scenario 3- Retrieve 10% of the result set */

select TransactionID, ProductID, TransactionDate, TransactionType
from Production.TransactionHistory --113,443 rows returned * 10% = 11,345

-- Display only the first 10% rows
select Top 10 percent
TransactionID, ProductID, TransactionDate, TransactionType
from Production.TransactionHistory --11,345 rows returned

/* Scenario 4- Retrieve distinct values in a column */

select * from Sales.CreditCard

select CardType from Sales.CreditCard

--Find the total Card Types without any duplication
select distinct
CardType from Sales.CreditCard

/* Scenario 5- Return values based on one condition */

select ProductID, ReviewerName, Rating from Production.ProductReview

-- Display the output to be more understandable (1=Poor, 2=Fair, 3=Good, 4=Very Good,
5=Excellent)
select ProductID,
       ReviewerName,
       case Rating
        when 1 then 'Poor'
        when 2 then 'Fair'
        when 3 then 'Good'
        when 4 then 'Very Good'
        when 5 then 'Excellent'
       end as Rating
from Production.ProductReview

/* Scenario 6- Replace NULL values with specific values */

select BillOfMaterialsID, ProductAssemblyID, StartDate
from Production.BillOfMaterials
```

```sql
-- Instead of NULL values we need to have value 0 for 'ProductAssemblyID column "without
changing any value" in the table
select BillOfMaterialsID,
       ISNULL(ProductAssemblyID,0) AS ProductAssemblyID,
       StartDate
from Production.BillOfMaterials

/* Scenario 7- Replacing the table or column name temporarily */

select ProductModelID, ProductDescriptionID
from Production.ProductModelProductDescriptionCulture

--Name columns and table name with an Alias
select ProductModelID AS ID,
       ProductDescriptionID AS DescID
from Production.ProductModelProductDescriptionCulture AS Table1

/* Scenario 8- Filtering out information */

select * from Person.AddressType

-- Filter out rows that has the name 'Archive' from the display
select * from Person.AddressType
where NOT Name = 'Archive'

/* Scenario 9- Filtering on more than 1 condition */

select * from Purchasing.PurchaseOrderDetail

-- Find all purchase order for the ProductID = 512 that costs less than $35 unit price
select * from Purchasing.PurchaseOrderDetail
where ProductID = 512 AND UnitPrice <35

/* Scenario 10- Search within a range of values */

select Name, ProductNumber, ListPrice from Production.Product

-- Find the name of products that has a list price in the range of $10-$20
select Name, ProductNumber, ListPrice from Production.Product
where ListPrice BETWEEN 10 AND 20

/* Scenario 11- Filtering out data by comparing values */

select * from Production.WorkOrder

-- 1. Find records for Products with ProductID = 995
-- 2. Find records for Products with ProductID = 995 that has more than 500 orders
-- 3. Find records for Products with ProductID = 995 that has more than 500 orders and
received before May 3, 2013

select * from Production.WorkOrder
where ProductID = 995

select * from Production.WorkOrder
where ProductID = 995 AND OrderQty > 500

select * from Production.WorkOrder
```

```sql
where ProductID = 995 AND OrderQty > 500 AND StartDate < '2013-05-03'

/* Scenario 12- Finding rows based on a list of values */

select Name, ListPrice from Production.Product

-- Find the name of products that has these 3 ListPrice values: 106.50, 1003.91, 333.42
select Name, ListPrice from Production.Product
where ListPrice IN (106.50, 1003.91, 333.42)

/* Scenario 13- Finding rows having a specific string */

select * from Person.CountryRegion

-- Need to find the name of the countries that start with the letter 'V'
select * from Person.CountryRegion
where Name like 'V%'

-- Need to find the name of the countries that start with the letter 'Vi'
select * from Person.CountryRegion
where Name like 'Vi%'

/* Scenario 14- Filtering rows having no data value in the column */

select * from Production.WorkOrder

-- Find Work Orders that has a Scrap Reason
select * from Production.WorkOrder
where ScrapReasonID IS NOT NULL

/* Scenario 15- Filtering rows based on some values in a sub-query (look-up method) */

select * from Production.WorkOrder

-- Find the Name of Products having more than 20,000 Order Quantity
select ProductID from Production.WorkOrder
where OrderQty > 20000                          --Query 1

--Name of Products from another Table
select * from Production.Product

select ProductID, Name from Production.Product  --Query 2
where ProductID = ANY(
   select ProductID from Production.WorkOrder
where OrderQty > 20000
)                                               --Query 1 inside Query 2

/* Scenario 16- Return values by converting them into Upper or Lower case */

select * from Production.Product

select Name, ProductNumber from Production.Product

-- The "ProductNumber" has Characters in uppercase.
-- So, Covert characters in "Name" column also in uppercase

select UPPER(Name) AS Name, ProductNumber from Production.Product
```

```sql
--Converting to "ProductNumber" lowercase characters
select Name, ProductNumber from Production.Product

select Name, LOWER(ProductNumber) AS ProductNumber from Production.Product

/* Scenario 17- Return values by extracting specific characters */

select * from Production.Product

select Name, ProductNumber from Production.Product

-- Get(extract) only the 2 characters from the "ProductName" for each "Name" column

-- Left function
select Name, LEFT(ProductNumber,2) AS ProductNumber from Production.Product

-- Get(extract) only the 4 numbers from the "ProductName" for each "Name" column

-- Right function
select Name, RIGHT(ProductNumber,4) AS ProductNumber from Production.Product

/* Scenario 18- Select records that has matching values in two tables */

select * from Production.WorkOrder

select WorkOrderID, ProductID from Production.WorkOrder

-- Find the "Product Name" of each ProductID along the with the WorkOrderID

-- Getting the Product Name
select * from Production.Product

select ProductID, Name from Production.Product

-- Inner Join
select A.WorkOrderID, A.ProductID, B.Name from Production.WorkOrder AS A
INNER JOIN Production.Product AS B
ON A.ProductID = B.ProductID

/* Scenario 19- Select all records from first table and only the matching records from
second table */

select ProductID, Name from Production.Product

-- Find the Sales Orders for all ProductID's along with the ProductID and Name
-- Sales Order details
select * from Sales.SalesOrderDetail

select ProductID, SalesOrderID from Sales.SalesOrderDetail

--Left Join
select A.ProductID, A.Name, B.SalesOrderID from Production.Product AS A
LEFT JOIN Sales.SalesOrderDetail AS B
ON A.ProductID = B.ProductID

/* Scenario 20- Select all records from second table and only the matching records from
first table */
```

```sql
-- First Table
select * from Production.Product

select ProductID, Name from Production.Product

-- Second Table
select * from Production.ProductReview

select ProductID, Comments from Production.ProductReview

-- Find the reviews of products along with the product name (Right Join)
select B.ProductID, B.Comments, A.Name from Production.Product AS A
RIGHT JOIN Production.ProductReview AS B
ON A.ProductID = B.ProductID

/* Scenario 21- Select all records from two tables when there is a match between them or
not */

select * from Production.Product

select ProductID, Name, ProductSubcategoryID from Production.Product -- Product Table

-- Find the Sub-category name to which each Product belongs,
-- and also find if any Sub-category name is not assigned to a Product name

-- Find Sub-category name
select * from Production.ProductSubcategory -- Sub-category Table

select ProductSubcategoryID, Name from Production.ProductSubcategory

-- Full Join
select A.ProductID, A.Name, A.ProductSubcategoryID, B.Name from Production.Product AS A
FULL JOIN Production.ProductSubcategory AS B
ON A.ProductSubcategoryID = B.ProductSubcategoryID

/* Scenario 22- Return the number of items found in a result set */

select * from Production.Product

select COUNT(ProductNumber) from Production.Product

-- Check how many product Numbers are there without any duplication
select distinct COUNT(ProductNumber) from Production.Product

/* Scenario 23- Compute the total amount */

select * from Sales.SalesOrderDetail

select SalesOrderID, ProductID, LineTotal, ModifiedDate from Sales.SalesOrderDetail

-- Find the Total revenue from the Product 777 sold in the year 2011
select SalesOrderID, ProductID, LineTotal, ModifiedDate from Sales.SalesOrderDetail
where ProductID = 777
and ModifiedDate BETWEEN '2011-01-01' AND '2011-12-31'

-- Total Revenue
select SUM(LineTotal) from Sales.SalesOrderDetail
where ProductID = 777
```

```sql
and ModifiedDate BETWEEN '2011-01-01' AND '2011-12-31'

/* Scenario 24- Compute the average value */

select * from Sales.SalesOrderDetail

select SalesOrderID, ProductID, LineTotal, ModifiedDate from Sales.SalesOrderDetail

-- Find the Average Price on which the Product 777 got sold in 2011
select SalesOrderID, ProductID, LineTotal, ModifiedDate from Sales.SalesOrderDetail
where ProductID = 777
and ModifiedDate BETWEEN '2011-01-01' AND '2011-12-31'

--Average
select AVG(LineTotal) from Sales.SalesOrderDetail
where ProductID = 777
and ModifiedDate BETWEEN '2011-01-01' AND '2011-12-31'

/* Scenario 25- Compute the lowest value */

-- Find the lowest quantity in stock for the ProductID 944
select * from Production.ProductInventory
where ProductID = 944

select MIN(Quantity) from Production.ProductInventory
where ProductID = 944

/* Scenario 26- Compute the largest value */

-- Find the largest quantity in stock for the ProductID 747
select * from Production.ProductInventory
where ProductID = 747

select MAX(Quantity) from Production.ProductInventory
where ProductID = 747

/* Scenario 27- Combine values from two columns into one column */

select StateProvinceID, StateProvinceCode, Name from Person.StateProvince

-- Display State Code and State Name in this format: State Code-State Name
-- Example: for State Code-AK and State Name- Alaska, need to display as 'Ak-Alaska'

select StateProvinceID, CONCAT(StateProvinceCode, '-', Name) from Person.StateProvince

-- Using 'State' as the column name
select StateProvinceID, CONCAT(StateProvinceCode, '-', Name) AS State from
Person.StateProvince

/* Scenario 28- Create a calculated field */

select ProductID, UnitPrice, OrderQty, LineTotal, RejectedQty
from Purchasing.PurchaseOrderDetail

-- Find the Amount lost due to the rejected quantity
select ProductID,
       UnitPrice,
       OrderQty,
```

```
       LineTotal,
       RejectedQty,
       (UnitPrice * RejectedQty) AS LossAmount
from Purchasing.PurchaseOrderDetail

/* Scenario 29- Arrange rows in groups */

select ProductID, Quantity from Production.ProductInventory

-- Find the lowest Quantity for each ProductID in the Inventory
select ProductID, MIN(Quantity) AS MinQuantity from Production.ProductInventory
GROUP BY ProductID

/* Scenario 30- Filter Groups based on condition */

select ProductID, Quantity, LocationID from Production.ProductInventory

-- If the location 'LocationID' is less than 3 for a ProductID,
-- then we need to find the lowest stock quantity 'Quantity' for only that ProductID

select ProductID,
       MIN(Quantity) AS MinCount,
       COUNT(LocationID) AS Locations
from Production.ProductInventory
GROUP BY ProductID
HAVING COUNT(LocationID) < 3
```