```
/*
    Working with NULL values
    Top and Distinct functions
    WHERE Clause


*/

-- Working with NULL VALUES
-- This doesn't work. Anytime the MiddleName is NULL the entire expression becomes NULL
SELECT FirstName + ' ' + MiddleName + ' ' + LastName AS FullName,
    FirstName, MiddleName, LastName
FROM Person.Person;

--Use the ISNULL function to replace the NULL value
SELECT FirstName + ' ' + ISNULL(MiddleName,'') + ' ' + LastName AS FullName,
    FirstName, MiddleName, LastName
FROM Person.Person;

-- *** Top and Distinct Functions
--Returns 121,317 rows
SELECT * FROM Sales.SalesOrderDetail;

--Returns 10 rows
SELECT TOP(10) *
FROM Sales.SalesOrderDetail;

--Returns 12,132 rows
SELECT TOP(10) PERCENT *
FROM Sales.SalesOrderDetail;

--Use with ORDER BY
SELECT TOP(10) SalesOrderID, SalesOrderDetailID
FROM Sales.SalesOrderDetail
ORDER BY SalesOrderID;

--Use DISTINCT to get a unique set of rows
```

```sql
SELECT DISTINCT Color
FROM Production.Product;

-- Still returns all the rows because each row is unique
SELECT *
FROM Production.Product;

SELECT DISTINCT *
FROM Production.Product;

-- WHERE  CLAUSE
SELECT CustomerID, SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
WHERE CustomerID = 29825;

SELECT CustomerID, SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
WHERE CustomerID = CustomerID;

SELECT CustomerID, SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
WHERE 1 = 2;

-- Using a function in the where clause
-- This is not always a good idea.
-- If there was an index on OrderDate, SQL would not be able to use it effectively
SELECT CustomerID, SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
WHERE YEAR(OrderDate) = 2013;

SELECT FirstName, LastName
FROM Person.Person
WHERE LEFT(LastName ,1) = 'S';

-- OPERATORS
SELECT CustomerID
FROM Sales.SalesOrderHeader
```

```sql
WHERE CustomerID = 11000;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID <> 11000;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID != 11000;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID > 11000;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID < 11005;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID <= 11005;

SELECT CustomerID
FROM Sales.SalesOrderHeader
WHERE CustomerID BETWEEN 11000 AND 11005;

SELECT FirstName, LastName
FROM Person.Person
WHERE LastName BETWEEN 'A' and 'C';

-- LIKE operator
-- Use LIKE when you know at least the first letter
-- Return all the LastName values that start with S
SELECT LastName
FROM Person.Person
WHERE LastName LIKE 'S%';
```

```sql
--Can use % anywhere in the value
--Return all the LastName values that have S in them
SELECT LastName
FROM Person.Person
WHERE LastName LIKE '%s%';

--Use _ to replace one character
SELECT LastName
FROM Person.Person
WHERE LastName LIKE 'Anders_n';

--Can use a list of possible values to replace one character
SELECT LastName
FROM Person.Person
WHERE LastName LIKE 'Anders[eo]n';

-- IN OPERATOR
SELECT FirstName, LastName
FROM Person.Person
WHERE LastName IN ('Smith','Anderson');

SELECT OrderDate, SalesOrderID
FROM Sales.SalesOrderHeader
WHERE OrderDate IN ('2012-08-01','2013-08-01');

SELECT CustomerID, OrderDate, SalesOrderID
FROM Sales.SalesOrderHeader
WHERE CustomerID IN (11000,11001);

-- Multiple Predicates
-- Use AND when both predicates must be true
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE FirstName = 'Hailey' AND LastName = 'Barnes';

-- USE OR When either can be true
SELECT BusinessEntityID, FirstName, LastName
```

```sql
FROM Person.Person
WHERE FirstName = 'Hailey' OR LastName = 'Barnes';

-- Can use more than two conditions
-- and any type of predicate
SELECT SalesOrderID, CustomerID, OrderDate
FROM Sales.SalesOrderHeader
WHERE CustomerID BETWEEN 11000 AND 12000
    AND OrderDate >= '2012-01-01' AND OrderDate < '2013-01-01'

ORDER BY CustomerID

--Combine OR and AND
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE FirstName = 'Hailey' OR FirstName = 'Haley';

--Find Hailey or Haley Barnes
--This one finds Haley Barnes plus any Hailey
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE FirstName = 'Hailey' OR FirstName = 'Haley'
    AND LastName = 'Barnes';

--To solve this, always include parentheses to enforce logic
--Here, the first name can be Hailey or Haley, and the last name must be Barnes
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE (FirstName = 'Hailey' OR FirstName = 'Haley')
    AND LastName = 'Barnes';

-- NOT Predicate
-- Use NOT to negate a predicate
SELECT FirstName, LastName
FROM Person.Person
WHERE NOT LastName = 'Smith'; -- same thing as LastName <> 'Smith'
```

```sql
--Retuns any records with Haley OR Barnes
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE FirstName = 'Hailey' OR LastName = 'Barnes';

--Returns all the rest
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE NOT(FirstName = 'Hailey' OR LastName = 'Barnes');

--All rows with 11000
SELECT CustomerID, SalesOrderID
FROM Sales.SalesOrderHeader
WHERE CustomerID IN(11000);

--all the other rows
SELECT CustomerID, SalesOrderID
FROM Sales.SalesOrderHeader
WHERE CustomerID NOT IN(11000);

--the rows that start with S
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE LastName LIKE 'S%';

--the rows that do not start with S
SELECT BusinessEntityID, FirstName, LastName
FROM Person.Person
WHERE LastName NOT LIKE 'S%';

-- Working with NULL values
-- Total Rows
-- 19,972 rows
SELECT FirstName, MiddleName, LastName
FROM Person.Person;

--Find the rows with MiddleName = B
```

```sql
--291 rows
SELECT FirstName, MiddleName, LastName
FROM Person.Person
WHERE MiddleName = 'B';

--Find the rows where MiddleName <> B
-- 19972 - 291 = 19681?
-- No, only 11,182 rows returned
-- We do not know if the NULL rows are B or not!
SELECT FirstName, MiddleName, LastName
FROM Person.Person
WHERE MiddleName <> 'B';

--Use the IS NULL operator
SELECT FirstName, MiddleName, LastName
FROM Person.Person
WHERE MiddleName <> 'B' OR MiddleName IS NULL;
```