--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1--- Day1---

-------------------------------------------------------------------------------------------------------------------------------

Basic Select Function (For Warm-Up)

–--------------------------------------------

-- To see all account

Select \* from [AMBSZM].[dbo].[A\_Account]

-- TO see top three rows

Select top(3)\* from [AMBSZM].[dbo].[A\_Account]

-- To change the Name of a column

Select TransactionAllowed as [T Allowed] from [AMBSZM].[dbo].A\_Account

-- To change a Name of a column Part 2

SELECT TOP (1000) [Id]

,[ParentId]

,[BranchId]

,[Code]

,[Name]

,[Level]

,[TransactionAllowed] as[T Allowed]

,[AccountType]

,[BalanceType]

,[Location]

,[IsVatable]

,[IsActive]

FROM [AMBSKE].[dbo].[A\_Account]

-- see all the Name of a column part 2

SELECT TOP (1000) [Id]

,[ParentId]

,[BranchId]

,[Code]

,[Name]

,[Level]

,[TransactionAllowed]

,[AccountType]

,[BalanceType]

,[Location]

,[IsVatable]

,[IsActive]

FROM AMBSKE.dbo.A\_Account

-- To Filter a column Name

Select \* from AMBSKE.dbo.A\_Account where Name = 'Asset'

-- To Filter a column Name part 2

Select \* from AMBSKE.dbo.A\_Account where Name = 'Cash and cash equivalents'

-- Multiple filter using where

Select \* from AMBSKE.dbo.A\_Account where Level = '2' AND AccountType = '1' AND Location = '-2'

/\*Findind a registered member by me

in the Zambia country\*/

Select \* from [AMBSZM].[dbo].P\_Member WHERE FullName like 'Akash%'

-- Using of like

Select \* from [AMBSZM].[dbo].P\_Member WHERE FullName like '% Akash%'

Select \* from [AMBSZM].[dbo].P\_Member WHERE(FullName like 'Akash%') AND (FatherName != 'Mr. A')

Select \* from [AMBSZM].[dbo].P\_Member WHERE(FullName like 'Akash%') AND (FatherName <> 'Mr. A')

Select \* from [AMBSZM].[dbo].P\_Member WHERE FullName like 'Akash%' or FullName like 'Rahat%'

Select \* from [AMBSZM].[dbo].P\_Member WHERE FullName like 'Akash%' or FullName like 'Rahat%' or FullName like 'Rafat%' or FullName like 'Hafija%'

Select \* from [AMBSZM].[dbo].P\_Member WHERE P\_GroupId BETWEEN 501126 AND 501271

-- Checked to Update a value in a column (Start)

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

Update [AMBSZM].[dbo].P\_LoanAccount

SET InterestAmount = '421'

Where P\_MemberId = '90130'

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

Update [AMBSZM].[dbo].P\_LoanAccount

SET InterestAmount = '420'

Where P\_MemberId = '90130'

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

-- Checked to Update a value in a column (End Just changed the value back to previous value)

--How to use IN

Select \* FROM [AMBSZM].[dbo].P\_Member WHERE NationalIdType IN (32,1,64,256, 16258)

-- How to select unique values in a column

SELECT DISTINCT NationalIdType FROM [AMBSZM].[dbo].P\_Member

-- How to select unique values in a column Part 2

SELECT DISTINCT FullName FROM [AMBSZM].[dbo].P\_Member

-- Count unique values in a column

SELECT Count(DISTINCT FullName) as Total\_Member\_Count

FROM [AMBSZM].[dbo].P\_Member

-- Count unique values in a column Part 2 Naming a column

SELECT Count(DISTINCT NationalIdType)

AS NationalIdType

FROM [AMBSZM].[dbo].P\_Member

--Finding Date Range

Select \* from [AMBSZM].[dbo].P\_Member as DOB where DateOfBirth >=1999 or DateOfBirth <=2000

--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2--- Day2---

-------------------------------------------------------------------------------------------------------------------------------

–Use Basic select function to understand the tables

—------------------------------------------------------------------

Select \* from [AMBSZM].[dbo].[P\_Member] WHERE FullName like '% Akash%'

Select \* from [AMBSZM].[dbo].[HRM\_GradeDesignation]

Select \* from [AMBSZM].[dbo].[Module]

Select \* from [AMBSZM].[dbo].[A\_Account]

Select \* from [AMBSZM].[dbo].[A\_AccountOld]

Select \* from [AMBSZM].[dbo].[A\_Currency]

Select \* from [AMBSZM].[dbo].[A\_LoanInterestReceivable]

Select \* from [AMBSZM].[dbo].[A\_ReportConfig]

Select \* from [AMBSZM].[dbo].[A\_Transaction]

Select \* from [AMBSZM].[dbo].[A\_Transaction2021]

Select \* from [AMBSZM].[dbo].[A\_Transaction2022]

Select \* from [AMBSZM].[dbo].[A\_TransactionDetails]

Select \* from [AMBSZM].[dbo].[A\_TransactionDetails2021]

Select \* from [AMBSZM].[dbo].[A\_TransactionDetails2022]

Select \* from [AMBSZM].[dbo].[A\_TransactionDetailsGranular]

Select \* from [AMBSZM].[dbo].[A\_TransactionDetailsOld]

Select \* from [AMBSZM].[dbo].[A\_TransactionOld]

Select \* from [AMBSZM].[dbo].[Admissionfee\_backup\_before\_delete]

Select \* from [AMBSZM].[dbo].[AuditTrailLog]

Select \* from [AMBSZM].[dbo].[AuthorizationLog]

Select \* from [AMBSZM].[dbo].[B\_BankingReportPermission]

Select \* from [AMBSZM].[dbo].[P\_SecurityAccount]

—--------------------------------------------------------------------------------------------------------------------

–Check the ‘PermanentCountry’ in the P\_Member table.

Select \* from [AMBSZM].[dbo].P\_Member

–See the unique country present in the P\_Member

Select Count (DISTINCT PermanentCountry) as country from [AMBSZM].[dbo].P\_Member

–Lets find the join in the table.

--INNER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

--FULL OUTER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

FULL OUTER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akash%'

/\*Good Example for INNER JOIN and FULL OUTER JOIN\*/

--INNER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

--FULL OUTER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

FULL OUTER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

--LEFT JOIN

Select P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

LEFT JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

--RIGHT JOIN

Select P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

RIGHT JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

--Try to do a self join (See the below table first)

Select \* from [AMBSZM].[dbo].[HRM\_GradeDesignation]

Select HRM\_GradeId as GID1 , HRM\_GradeId as GID2

From [AMBSZM].[dbo].[HRM\_GradeDesignation]

WHERE HRM\_GradeId != HRM\_DesignationId

Select \* FROM [AMBSZM].[dbo].[P\_SecurityAccount]

--Regular SUM in the P\_SecurityAccount

SELECT SUM(P\_SecurityAccount.MonthlyIncomeAmount) as Total\_Monthly\_Income

From [AMBSZM].[dbo].[P\_SecurityAccount]

----SUM inside an INNER JOIN

SELECT SUM(P\_SecurityAccount.MonthlyIncomeAmount) as Total\_Monthly\_Income

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

----AVG inside an INNER JOIN

SELECT AVG(P\_SecurityAccount.MonthlyIncomeAmount) as Avarage\_Monthly\_Income

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

--INNER JOIN

SELECT P\_GroupId,P\_MemberId, FullName, DateOfBirth, AdmissionDate, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

Select \* FROM [AMBSZM].[dbo].[P\_Member] ORDER by PassbookNumber

--Find unique passbook number

Select Count(DISTINCT PassbookNumber)

FROM [AMBSZM].[dbo].[P\_Member]

--Finding Null value

SELECT \*

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

--Couning null value in a column

SELECT Count(Id)

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

--Max passbook Number where SpouseDateOfBirth IS NULL

SELECT MAX(PassbookNumber)

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

---Use of UNION

Select P\_GroupId FROM [AMBSZM].[dbo].[P\_Member]

UNION

SELECT

P\_MemberId FROM [AMBSZM].[dbo].[P\_SecurityAccount]

---Use of UNION with a where case

Select Name FROM [AMBSZM].[dbo].[Module] as UNION\_Name

WHERE Name like 'A%'

UNION

SELECT

Name FROM [AMBSZM].[dbo].[A\_Currency]

Where Code = 'AFN'

---Use of UNION with a where case and sort in descending order

Select Name FROM [AMBSZM].[dbo].[Module] as UNION\_Name

WHERE Name like 'A%'

UNION

SELECT

Name FROM [AMBSZM].[dbo].[A\_Currency]

Where Code = 'AFN' Order BY Name DESC

---Follow the below table for above example

-------------------------------------------

Select \* from [AMBSZM].[dbo].[Module]

Select \* from [AMBSZM].[dbo].[A\_Currency]

-------------------------------------------

---Using GRUOP BY

Select Count(Id), FullName

From [AMBSZM].[dbo].[P\_Member]

Group by FullName

ORDER BY COUNT(ID) ASC

---INNER JOIN

SELECT P\_GroupId,P\_MemberId, FullName, DateOfBirth, AdmissionDate, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3--- Day3---

-----------------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------------

/\*Let's make a guideline for all\*/

**/\*The SQL SELECT Statement\*/**

/\*

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

Example:

SELECT column1, column2, ...

FROM table\_name;

\*/

--Example 1

Select \* from [AMBSZM].[dbo].[A\_Account]

--Example 2

SELECT TOP (1000) [Id]

,[ParentId]

,[BranchId]

,[Code]

,[Name]

,[Level]

,[TransactionAllowed]

,[AccountType]

,[BalanceType]

,[Location]

,[IsVatable]

,[IsActive]

FROM AMBSKE.dbo.A\_Account

/\*Aliases\*/

/\*

SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the 'AS' keyword.

\*/

--Example 1

Select Level AS Lvl\_Head, Code AS Code\_Head, Name As Name\_Head

FROM AMBSKE.dbo.A\_Account

/\*

SELECT DISTINCT Examples

SELECT DISTINCT Column\_Name FROM Table\_Name;

\*/

-- Count unique values in a column

--Example 1

Select DISTINCT NationalIdType from [AMBSZM].[dbo].P\_Member

--Example 2 (Count unique values in a column Part 2 Naming a column)

SELECT Count(DISTINCT NationalIdType)

AS NationalIdType

FROM [AMBSZM].[dbo].P\_Member

--Example 3 (Using count to count the total unique name)

SELECT Count(DISTINCT FullName) as Total\_Member\_Count

FROM [AMBSZM].[dbo].P\_Member

/\*SQL WHERE Clause\*/

-- Multiple filter using where

--Example 1

Select \* from AMBSKE.dbo.A\_Account where Level = '2'

--Example 2

Select \* from AMBSKE.dbo.A\_Account where Name = 'Asset'

/\*The SQL AND, OR and NOT Operators\*/

--AND

--Example 1

Select \* FROM AMBSKE.dbo.A\_Account

WHERE Level = '2'

AND AccountType = '1'

AND Location = '-2'

--OR

--Example 1

Select \* FROM AMBSKE.dbo.A\_Account

WHERE Level = '2'

OR AccountType = '1'

OR Location = '-2'

--NOT

--Example 1

Select \* FROM AMBSKE.dbo.A\_Account

WHERE NOT Level = '2'

/\*SQL ORDER BY Keyword\*/

/\*

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Example:

SELECT column1, column2, ...

FROM table\_name

ORDER BY column1, column2, ... ASC|DESC;

\*/

--Example 1

Select Level, Code, Name

FROM AMBSKE.dbo.A\_Account

ORDER BY Name, Code --Name will be ascending then Code will be in ascending order By default

-- Example 2

Select Level, Code, Name

FROM AMBSKE.dbo.A\_Account

ORDER BY Name ASC, Code DESC --Name will be ascending then Code will be in descending order

/\*Try different Combination

for exercise

\*/

/\*SQL NULL Values\*/

/\*

What is a NULL Value?

A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field. Then, the field will be saved with a NULL value.

How to Test for NULL Values?

It is not possible to test for NULL values with comparison operators, such as =, <, or <>.

We will have to use the IS NULL and IS NOT NULL operators instead.

IS NULL Syntax

SELECT column\_names

FROM table\_name

WHERE column\_name IS NULL;

IS NOT NULL Syntax

SELECT column\_names

FROM table\_name

WHERE column\_name IS NOT NULL;

\*/

--Example 1

--Finding Null value IS NULL

SELECT P\_GroupId, DefaultP\_ProgramId, FullName, DateOfBirth, SpouseDateOfBirth

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

--Example 2

--Finding Null value IS NOT NULL

SELECT P\_GroupId, DefaultP\_ProgramId, FullName, DateOfBirth, SpouseDateOfBirth

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NOT NULL;

--Example 3

--Counting IS NULL value in a column

SELECT Count(Id)

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

--Example 4

--Counting IS NOT NULL value in a column

SELECT Count(Id)

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NOT NULL;

--Example 5

--Max passbook Number where SpouseDateOfBirth IS NULL

SELECT MAX(PassbookNumber)

FROM [AMBSZM].[dbo].[P\_Member]

WHERE SpouseDateOfBirth IS NULL;

/\*SQL UPDATE Statement\*/

/\*

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax

Example:

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

\*/

--Please see the below example. I have updated the value and changed it back again.

-- Checked to Update a value in a column (Start)

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

Update [AMBSZM].[dbo].P\_LoanAccount

SET InterestAmount = '421'

Where P\_MemberId = '90130'

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

Update [AMBSZM].[dbo].P\_LoanAccount

SET InterestAmount = '420'

Where P\_MemberId = '90130'

Select \* from [AMBSZM].[dbo].P\_LoanAccount where P\_MemberId = '90130'

-- Checked to Update a value in a column (End Just changed the value back to previous value)

/\*SQL DELETE Statement\*/

/\*

The DELETE statement is used to delete existing records in a table.

Example:

DELETE FROM table\_name WHERE condition;

\*/

/\*SQL TOP\*/

/\*

The SQL SELECT TOP Clause

The SELECT TOP clause is used to specify the number of records to return.

The SELECT TOP clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

Example:

SELECT TOP (N) column\_name(s)

FROM table\_name;

\*/

--TOP

--Example 1

SELECT TOP 3 \* FROM AMBSKE.dbo.A\_Account;

--Example 2

SELECT TOP (15) [Id]

,[ParentId]

,[BranchId]

,[Code]

,[Name]

,[Level]

,[TransactionAllowed]

FROM AMBSKE.dbo.A\_Account

--Using OFFSET

--Example 3

Select \*

from [AMBSZM].[dbo].[A\_Currency]

Order By Id ASC

OFFSET 10 ROWS

--Using Fetch and OFFSET

--Example 4

Select \*

from [AMBSZM].[dbo].[A\_Currency]

Order By Id ASC

OFFSET 10 ROWS

FETCH FIRST 2 ROWS ONLY

--Using Fetch and OFFSET

--Example 5

Select Id, FullName, ContactNumber , DateOfBirth

from [AMBSZM].[dbo]. P\_Member

order by DateOfBirth ASC

OFFSET 10 rows

FETCH First 25 rows only;

/\*SQL MIN() and MAX() Functions\*/

/\*

The MIN() function returns the smallest value of the selected column.

Example:

SELECT MIN(column\_name)

FROM table\_name

WHERE condition;

The MAX() function returns the largest value of the selected column.

Example:

SELECT MAX(column\_name)

FROM table\_name

WHERE condition;

\*/

--Max passbook Number where SpouseDateOfBirth IS NULL

SELECT MAX(PassbookNumber) as MAX

FROM [AMBSZM].[dbo].[P\_Member]

--MIN passbook Number where SpouseDateOfBirth IS NULL

SELECT MIN(PassbookNumber) AS MIN

FROM [AMBSZM].[dbo].[P\_Member]

/\*

SQL COUNT(), AVG() and SUM() Functions

COUNT() Syntax:

The COUNT() function returns the number of rows that matches a specified criterion.

Example:

SELECT COUNT(column\_name)

FROM table\_name

WHERE condition;

AVG() Syntax:

The AVG() function returns the average value of a numeric column.

Example:

SELECT AVG(column\_name)

FROM table\_name

WHERE condition;

SUM() Syntax:

The SUM() function returns the total sum of a numeric column.

Example:

SELECT SUM(column\_name)

FROM table\_name

WHERE condition;

\*/

--COUNT() Syntax: Example

SELECT Count(PassbookNumber) AS Count

FROM [AMBSZM].[dbo].[P\_Member]

--AVG() Syntax: Example

SELECT AVG(PassbookNumber) AS Agerage

FROM [AMBSZM].[dbo].[P\_Member]

--SUM() Syntax: Example

SELECT SUM(PassbookNumber) AS SUM

FROM [AMBSZM].[dbo].[P\_Member]

/\*SQL LIKE Operator\*/

/\*

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

The percent sign (%) represents zero, one, or multiple characters

The underscore sign (\_) represents one, single character

Example:

SELECT column1, column2, ...

FROM table\_name

WHERE columnN LIKE pattern;

\*/

/\*

LIKE Operator Description

WHERE CustomerName LIKE 'a%' Finds any values that start with "a"

WHERE CustomerName LIKE '%a' Finds any values that end with "a"

WHERE CustomerName LIKE '%or%' Finds any values that have "or" in any position

WHERE CustomerName LIKE '\_r%' Finds any values that have "r" in the second position

WHERE CustomerName LIKE 'a\_%' Finds any values that start with "a" and are at least 2 characters in length

WHERE CustomerName LIKE 'a\_\_%' Finds any values that start with "a" and are at least 3 characters in length

WHERE ContactName LIKE 'a%o' Finds any values that start with "a" and ends with "o"

\*/

--Example 1

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like 'a%'

--Example 2

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like '%a'

--Example 3

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like '%Akash%'

--Example 4

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like '\_kash%'

--Example 5

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like 'a\_%'

--Example 6

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like 'a\_\_%'

--Example 7

SELECT P\_GroupId, PassbookNumber, FullName

FROM [AMBSZM].[dbo].[P\_Member]

WHERE FullName like 'A%d'

/\*SQL IN Operator\*/

/\*

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

\*/

--Example 1

--The query below can be shorted using IN Operator

Select \*

FROM [AMBSZM].[dbo].P\_Member

WHERE FullName = 'Akash Ahmed' or FullName = 'Rafat1-Aaksh' or FullName = 'Rafatul123'

--Using IN

Select \*

FROM [AMBSZM].[dbo].P\_Member

WHERE FullName IN ('Akash Ahmed','Rafat1-Aaksh','Rafatul123')

--Example For NOT IN

Select \*

FROM [AMBSZM].[dbo].P\_Member

WHERE FullName NOT IN ('Akash Ahmed','Rafat1-Aaksh','Rafatul123')

/\*SQL BETWEEN Operator\*/

/\*

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

Example:

SELECT column\_name(s)

FROM table\_name

WHERE column\_name BETWEEN value1 AND value2;

\*/

--Example 1

--BETWEEN

SELECT Id, P\_GroupId, DefaultP\_ProgramId, FullName

FROM [AMBSZM].[dbo].P\_Member

WHERE P\_GroupId BETWEEN 20015 AND 20017

--Example 2

--NOT BETWEEN

SELECT Id, P\_GroupId, DefaultP\_ProgramId, FullName

FROM [AMBSZM].[dbo].P\_Member

WHERE P\_GroupId NOT BETWEEN 20015 AND 20017

--Example 3

--BETWEEN with IN Example

SELECT Id, P\_GroupId, DefaultP\_ProgramId, FullName

FROM [AMBSZM].[dbo].P\_Member

WHERE P\_GroupId NOT BETWEEN 20015 AND 20017

AND P\_GroupId NOT IN (1,3,501275)

--Example 4

SELECT Id, P\_GroupId, DefaultP\_ProgramId, FullName

FROM [AMBSZM].[dbo].P\_Member

WHERE P\_GroupId NOT BETWEEN 20015 AND 20017

AND P\_GroupId NOT IN (1,3,10002)

ORDER BY P\_GroupId ASC

/\*SQL Joins\*/

/\*

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

\*/

/\*SQL INNER JOIN Keyword\*/

/\*The INNER JOIN keyword selects records that have matching values in both tables.

INNER JOIN Syntax:

SELECT column\_name(s)

FROM table1

INNER JOIN table2

ON table1.column\_name = table2.column\_name;

\*/

--Example 1

--INNER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akas%'

--Example 2

/\*Good Example for INNER JOIN and FULL OUTER JOIN\*/

--INNER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

/\*SQL LEFT JOIN Keyword\*/

/\*

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.

LEFT JOIN Syntax:

SELECT column\_name(s)

FROM table1

LEFT JOIN table2

ON table1.column\_name = table2.column\_name;

\*/

--Example 1

--LEFT JOIN

Select P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

LEFT JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

/\*SQL RIGHT JOIN Keyword\*/

/\*

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.

RIGHT JOIN Syntax:

SELECT column\_name(s)

FROM table1

RIGHT JOIN table2

ON table1.column\_name = table2.column\_name;

\*/

--Example 1

--RIGHT JOIN

Select P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

RIGHT JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

/\*SQL FULL OUTER JOIN Keyword\*/

/\*

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records.

Tip: FULL OUTER JOIN and FULL JOIN are the same.

FULL OUTER JOIN Syntax:

SELECT column\_name(s)

FROM table1

FULL OUTER JOIN table2

ON table1.column\_name = table2.column\_name

WHERE condition;

\*/

--Example 1

--FULL OUTER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

FULL OUTER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like '%Akash%'

--Example 2

--FULL OUTER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

FULL OUTER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

/\*SQL Self Join\*/

/\*

A self join is a regular join, but the table is joined with itself.

Self Join Syntax:

SELECT column\_name(s)

FROM table1 T1, table1 T2

WHERE condition;

\*/

--Try to do a self join using the below tabe:

Select \* from [AMBSZM].[dbo].[HRM\_GradeDesignation]

--Example 1

Select HRM\_GradeId as GID1 , HRM\_GradeId as GID2

From [AMBSZM].[dbo].[HRM\_GradeDesignation]

WHERE HRM\_GradeId != HRM\_DesignationId

/\*SQL UNION Operator\*/

/\*

The SQL UNION Operator

The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION must have the same number of columns

The columns must also have similar data types

The columns in every SELECT statement must also be in the same order

UNION Syntax:

SELECT column\_name(s) FROM table1

UNION

SELECT column\_name(s) FROM table2;

\*/

---Use of UNION

Select P\_GroupId FROM [AMBSZM].[dbo].[P\_Member]

UNION

SELECT

P\_MemberId FROM [AMBSZM].[dbo].[P\_SecurityAccount]

---Use of UNION with a where case

Select Name FROM [AMBSZM].[dbo].[Module] as UNION\_Name

WHERE Name like 'A%'

UNION

SELECT

Name FROM [AMBSZM].[dbo].[A\_Currency]

Where Code = 'AFN'

---Use of UNION with a where case and sort in descending order

Select Name FROM [AMBSZM].[dbo].[Module] as UNION\_Name

WHERE Name like 'A%'

UNION

SELECT

Name FROM [AMBSZM].[dbo].[A\_Currency]

Where Code = 'AFN' Order BY Name DESC

---Follow the below table for above example

-------------------------------------------

Select \* from [AMBSZM].[dbo].[Module]

Select \* from [AMBSZM].[dbo].[A\_Currency]

-------------------------------------------

/\*

UNION ALL Syntax

The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

SELECT column\_name(s) FROM table1

UNION ALL

SELECT column\_name(s) FROM table2;

\*/

Select Id, Name, IsActive from [AMBSZM].[dbo].[Module]

UNION ALL

Select Id, Name, IsActive from [AMBSZM].[dbo].[A\_Currency]

/\*SQL GROUP BY Statement\*/

/\*

The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

GROUP BY Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

ORDER BY column\_name(s);

\*/

---Using GRUOP BY

Select Count(Id), FullName

From [AMBSZM].[dbo].[P\_Member]

Group by FullName

ORDER BY COUNT(ID) ASC

/\*SQL HAVING Clause\*/

/\*

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

HAVING Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE condition

GROUP BY column\_name(s)

HAVING condition

ORDER BY column\_name(s);

\*/

--Example 1

Select Count(Id) AS ID1, FullName

From [AMBSZM].[dbo].[P\_Member]

Group by FullName

HAVING FullName = 'Mellan Kalilo'

--Example 2

Select Count (FullName) As Number\_of\_Name, NationalIdType

From [AMBSZM].[dbo].[P\_Member]

Group By NationalIdType

Having NationalIdType > 6

--Example 3

Select Count (FullName) As Number\_of\_Name, NationalIdType

From [AMBSZM].[dbo].[P\_Member]

Group By NationalIdType

Having Count (FullName) > 6

--FULL OUTER JOIN

SELECT P\_GroupId,P\_MemberId, DefaultP\_ProgramId, PassbookNumber, MembershipType, FullName, DateOfBirth, AdmissionDate, P\_ProgramId, OpeningDate, MonthlyIncomeAmount, ClosingReason

FROM [AMBSZM].[dbo].[P\_Member]

FULL OUTER JOIN [AMBSZM].[dbo].[P\_SecurityAccount]

ON [AMBSZM].[dbo].[P\_Member].P\_GroupId = [AMBSZM].[dbo].[P\_SecurityAccount].P\_MemberId

WHERE FullName like 'Aliness%'

/\*SQL EXISTS Operator\*/

/\*

The EXISTS operator is used to test for the existence of any record in a subquery.

The EXISTS operator returns TRUE if the subquery returns one or more records.

EXISTS Syntax:

SELECT column\_name(s)

FROM table\_name

WHERE EXISTS

(SELECT column\_name FROM table\_name WHERE condition);

\*/

--Example 1

Select P\_GroupId, PassbookNumber, FullName

From [AMBSZM].[dbo].[P\_Member]

Where EXISTS (

Select P\_MemberId

From [AMBSZM].[dbo].[P\_SecurityAccount])

/\*SQL ANY Examples\*/

/\*

The ANY operator:

returns a boolean value as a result

returns TRUE if ANY of the subquery values meet the condition

ANY means that the condition will be true if the operation is true for any of the values in the range.

ANY Syntax

SELECT column\_name(s)

FROM table\_name

WHERE column\_name operator ANY

(SELECT column\_name

FROM table\_name

WHERE condition);

\*/

--Example 1

Select \* from [AMBSZM].[dbo].[Module] Where Id = ANY(

Select Id from [AMBSZM].[dbo].[A\_Currency])

--Example 2

Select P\_GroupId, PassbookNumber, FullName

From [AMBSZM].[dbo].[P\_Member]

Where P\_GroupId = ANY(

Select P\_MemberId

From [AMBSZM].[dbo].[P\_SecurityAccount])

/\*The SQL ALL Operator\*/

/\*

The ALL operator:

returns a boolean value as a result

returns TRUE if ALL of the subquery values meet the condition

is used with SELECT, WHERE and HAVING statements

ALL means that the condition will be true only if the operation is true for all values in the range.

ALL Syntax With SELECT:

SELECT ALL column\_name(s)

FROM table\_name

WHERE condition;

\*/

--Example 1

Select \* from [AMBSZM].[dbo].[P\_Member] Where P\_GroupId = ALL(

Select P\_MemberId from [AMBSZM].[dbo].[P\_SecurityAccount])

--Example 2

Select P\_GroupId, PassbookNumber, FullName

From [AMBSZM].[dbo].[P\_Member]

Where P\_GroupId = ALL(

Select P\_MemberId

From [AMBSZM].[dbo].[P\_SecurityAccount] Where Status = 1)

/\*Time to find some relation between the tables in AMBS\*/

-------------------------------------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------------------------------------

Let's find all the relation of primary key and foreign key in the tables.

/\*ALL THE Primary Key\*/

SELECT

OBJECT\_NAME(parent\_object\_id) AS 'Table',

name

FROM sys.objects

WHERE type = 'PK';

/\*ALL THE Foreign Key\*/

SELECT obj.name AS FK\_NAME,

sch.name AS [schema\_name],

tab1.name AS [table],

col1.name AS [column],

tab2.name AS [referenced\_table],

col2.name AS [referenced\_column]

FROM sys.foreign\_key\_columns fkc

INNER JOIN sys.objects obj

ON obj.object\_id = fkc.constraint\_object\_id

INNER JOIN sys.tables tab1

ON tab1.object\_id = fkc.parent\_object\_id

INNER JOIN sys.schemas sch

ON tab1.schema\_id = sch.schema\_id

INNER JOIN sys.columns col1

ON col1.column\_id = parent\_column\_id AND col1.object\_id = tab1.object\_id

INNER JOIN sys.tables tab2

ON tab2.object\_id = fkc.referenced\_object\_id

INNER JOIN sys.columns col2

ON col2.column\_id = referenced\_column\_id AND col2.object\_id = tab2.object\_id

/\*ALL foreign key and primary key\*/

SELECT PKTABLE\_QUALIFIER = CONVERT(SYSNAME,DB\_NAME()),

PKTABLE\_OWNER = CONVERT(SYSNAME,SCHEMA\_NAME(O1.SCHEMA\_ID)),

PKTABLE\_NAME = CONVERT(SYSNAME,O1.NAME),

PKCOLUMN\_NAME = CONVERT(SYSNAME,C1.NAME),

FKTABLE\_QUALIFIER = CONVERT(SYSNAME,DB\_NAME()),

FKTABLE\_OWNER = CONVERT(SYSNAME,SCHEMA\_NAME(O2.SCHEMA\_ID)),

FKTABLE\_NAME = CONVERT(SYSNAME,O2.NAME),

FKCOLUMN\_NAME = CONVERT(SYSNAME,C2.NAME),

-- Force the column to be non-nullable (see SQL BU 325751)

--KEY\_SEQ = isnull(convert(smallint,k.constraint\_column\_id), sysconv(smallint,0)),

UPDATE\_RULE = CONVERT(SMALLINT,CASE OBJECTPROPERTY(F.OBJECT\_ID,'CnstIsUpdateCascade')

WHEN 1 THEN 0

ELSE 1

END),

DELETE\_RULE = CONVERT(SMALLINT,CASE OBJECTPROPERTY(F.OBJECT\_ID,'CnstIsDeleteCascade')

WHEN 1 THEN 0

ELSE 1

END),

FK\_NAME = CONVERT(SYSNAME,OBJECT\_NAME(F.OBJECT\_ID)),

PK\_NAME = CONVERT(SYSNAME,I.NAME),

DEFERRABILITY = CONVERT(SMALLINT,7) -- SQL\_NOT\_DEFERRABLE

FROM SYS.ALL\_OBJECTS O1,

SYS.ALL\_OBJECTS O2,

SYS.ALL\_COLUMNS C1,

SYS.ALL\_COLUMNS C2,

SYS.FOREIGN\_KEYS F

INNER JOIN SYS.FOREIGN\_KEY\_COLUMNS K

ON (K.CONSTRAINT\_OBJECT\_ID = F.OBJECT\_ID)

INNER JOIN SYS.INDEXES I

ON (F.REFERENCED\_OBJECT\_ID = I.OBJECT\_ID

AND F.KEY\_INDEX\_ID = I.INDEX\_ID)

WHERE O1.OBJECT\_ID = F.REFERENCED\_OBJECT\_ID

AND O2.OBJECT\_ID = F.PARENT\_OBJECT\_ID

AND C1.OBJECT\_ID = F.REFERENCED\_OBJECT\_ID

AND C2.OBJECT\_ID = F.PARENT\_OBJECT\_ID

AND C1.COLUMN\_ID = K.REFERENCED\_COLUMN\_ID

AND C2.COLUMN\_ID = K.PARENT\_COLUMN\_ID

/\*ALL foreign key and primary key (END)\*/

--Relationship between Member and Security Account

**P\_Member &P\_LoanAccount**

--Relation ship between **P\_Member** and **P\_LoanApplication**

----------------------------------------------------

SELECT \* from [AMBSZM].[dbo].[P\_Member]

SELECT \* from [AMBSZM].[dbo].[P\_LoanApplication]

--INNNER Join between P\_Member and P\_LoanApplication

SELECT FullName, P\_GroupId, P\_MemberId, P\_LoanAccountId

From [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_LoanApplication]

On [AMBSZM].[dbo].[P\_Member].Id = [AMBSZM].[dbo].[P\_LoanApplication].P\_MemberId

---Prove of my solution ----

--Look at the ProposedDuration, PassbookNumber they should be 6,9 for below queries

SELECT \* from [AMBSZM].[dbo].[P\_LoanApplication] where P\_MemberId = '1266885'

SELECT \* from [AMBSZM].[dbo].[P\_Member] where Id = '1266885'

SELECT FullName, P\_GroupId, P\_MemberId, P\_LoanAccountId, ProposedDuration, PassbookNumber

From [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_LoanApplication]

On [AMBSZM].[dbo].[P\_Member].Id = [AMBSZM].[dbo].[P\_LoanApplication].P\_MemberId where P\_MemberId = '1266885'

-------------------------------------------------------------------------------------------

**P\_Member &P\_MemberStatus**

--P\_Member &P\_MemberStatus

Select \* FROM [AMBSZM].[dbo].[P\_Member]

Select \* FROM [AMBSZM].[dbo].[P\_MemberStatus]

--Inner join

Select P\_Member.Id,P\_MemberId, P\_GroupId, FullName

From [AMBSZM].[dbo].[P\_Member]

INNER JOIN [AMBSZM].[dbo].[P\_MemberStatus]

On [AMBSZM].[dbo].[P\_Member].Id = [AMBSZM].[dbo].[P\_MemberStatus].P\_MemberId