DATA MIGRATION CONTEST - Question & Answer

Question:

The company is moving its customer and personal details into a new consolidated system. The target system requires a new table **CustomerMigration** to store cleaned and transformed customer data.

Create a new table named **CustomerMigration** in the AdventureWorks db. The table should contain the following columns:

- CustomerID
- AccountNumber
- FullName
- LoginName
- Email
- CustomerJoinedDate
- YearsOfExperience
- CreatedDate

Note: Use Appropriate DataType for each Column.

Constraints:

The `CustomerID` column should be the Primary Key and must not allow null values. Both `LoginName` and `Email` should not contain any duplicates - ensure no two customers share the same login or email. The `CreatedDate` column should have a default value of the current date.

Index:

- Create a **non-clustered index** on `LoginName`.

Calculations & Transformations

- **FullName** = `FirstName + ' ' + LastName`
- **CustomerJoinedDate** = `ModifiedDate` from `Person.Person`

- **YearsOfExperience** = difference in years between `CustomerJoinedDate` and today
- **LoginName** = extract the portion before `'@'` from `EmailAddress`

Rules:

Only migrate customers whose **CustomerJoinedDate (ModifiedDate)** is **2008 or later**.

Optional: Use a **Transaction** so that if any error occurs, the migration is **rolled back**. - Extra points if implemented

Hint: Tables and Columns

Migrate data from the following source tables into `CustomerMigration`:

- `Sales.Customer` → `CustomerID`, `PersonID`, `AccountNumber`
- `Person.Person` → `FirstName`, `LastName`, `ModifiedDate`
- `Person.EmailAddress` → `EmailAddress`

Answer:

```
-- Create or Alter Stored Procedure for Customer Migration
CREATE OR ALTER PROC usp_MigrateCustomerData
AS BEGIN
BEGIN TRY
   BEGIN TRANSACTION;

   -- Drop existing table if it exists
   DROP TABLE IF EXISTS CustomerMigration;

   -- Create the CustomerMigration table
   CREATE TABLE CustomerMigration (
      CustomerID INT NOT NULL PRIMARY KEY,
```

```
    AccountNumber NVARCHAR(20) NOT NULL,
    FullName NVARCHAR(200) NOT NULL,
    LoginName NVARCHAR(100) UNIQUE NOT NULL,
    Email NVARCHAR(200) UNIQUE NOT NULL,
    CustomerJoinedDate DATE NOT NULL,
    YearsOfExperience INT NOT NULL,
    CreatedDate DATETIME NOT NULL DEFAULT GETDATE()
);

-- Create Non-clustered Index on LoginName
CREATE NONCLUSTERED INDEX IX_CustomerMigration_LoginName
ON CustomerMigration(LoginName);

-- Insert data into CustomerMigration with transformations
INSERT INTO CustomerMigration (
    CustomerID,
    AccountNumber,
    FullName,
    LoginName,
    Email,
    CustomerJoinedDate,
    YearsOfExperience,
    CreatedDate
)
SELECT
    c.CustomerID,
    c.AccountNumber,
    p.FirstName + ' ' + p.LastName AS FullName,
    SUBSTRING(e.EmailAddress, 1, CHARINDEX('@', e.EmailAddress) - 1) AS
LoginName,
    e.EmailAddress AS Email,
    p.ModifiedDate AS CustomerJoinedDate,
    DATEDIFF(YEAR, p.ModifiedDate, GETDATE()) AS YearsOfExperience,
    GETDATE() AS CreatedDate
FROM Sales.Customer c
INNER JOIN Person.Person p
    ON c.PersonID = p.BusinessEntityID
INNER JOIN Person.EmailAddress e
    ON p.BusinessEntityID = e.BusinessEntityID
WHERE YEAR(p.ModifiedDate) >= 2008;
```

```
    COMMIT TRANSACTION;
    PRINT 'Data Migration Completed Successfully!';

END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Error occurred during migration. Transaction rolled back.';
    PRINT ERROR_MESSAGE();
END CATCH;
END;
GO

-- Execute the procedure
EXEC usp_MigrateCustomerData;

-- Check the migrated data
SELECT * FROM CustomerMigration;
```