STORED PROCEDURES, TRIGGERS, VIEWS, INDEXES AND SSIS PACKAGE

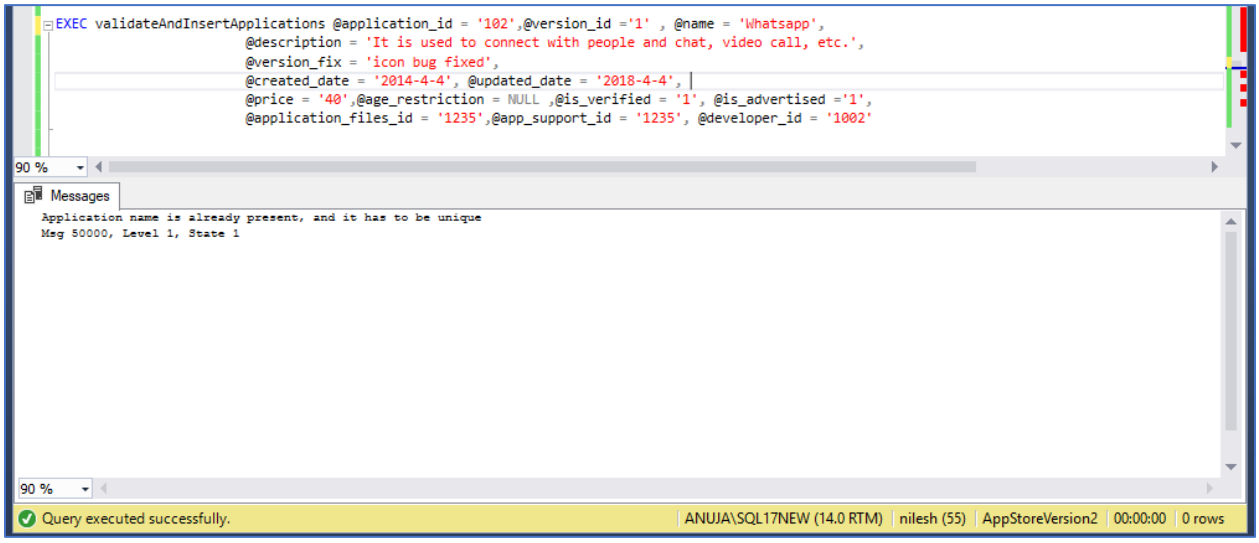
Stored Procedure to insert and validate the application name.
Since my model does not have unique constraint on the application name I had to create a procedure to check the uniqueness of the name with respect to the application_id.

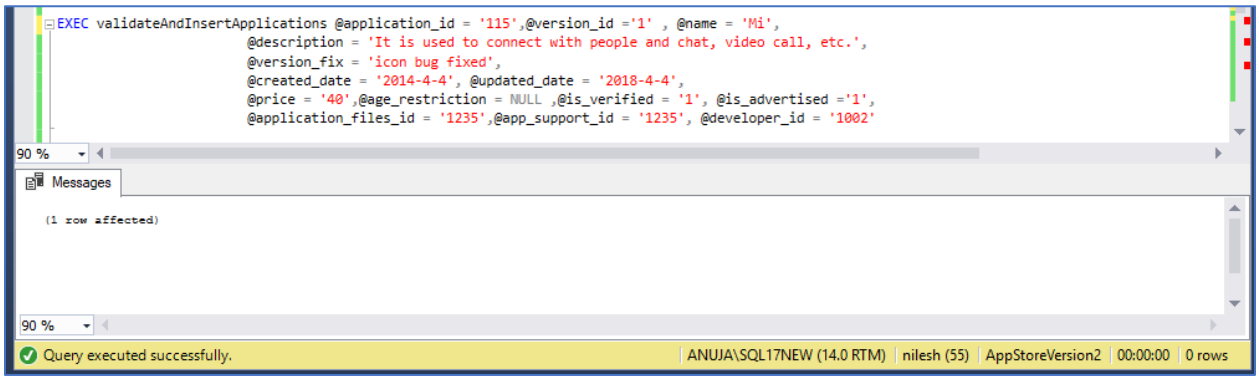
```
GO
ALTER PROCEDURE dbo.validateAndInsertApplications (@application_id INT,@version_id FLOAT,
@name VARCHAR(20),
                        @description VARCHAR(20),@version_fix
VARCHAR(50),@created_date DATE ,@updated_date DATE, @price FLOAT,@age_restriction
CHAR(4),
                        @is_verified BIT, @is_advertised BIT, @application_files_id
INT,@app_support_id INT, @developer_id INT)
AS
BEGIN
Declare @nameInserted varchar(20)
            (SELECT top 1 @nameInserted =  name FROM Applications WHERE application_id
= @application_id)
Declare @appID INT
             SELECT top 1 @appID =  application_id FROM Applications WHERE name = @name

            IF
            @name != @nameInserted

            BEGIN
                        RAISERROR('Application name is already present, and it has to
be unique',1,1)
            END
            ELSE
            BEGIN
                  INSERT INTO Applications
(application_id,version_id,name,description,version_fix,created_date,updated_date,price,a
ge_restriction,

      is_verified,is_advertised,application_files_id,app_support_id,developer_id)
                  VALUES (@application_id,@version_id,
@name,@description,@version_fix,@created_date,@updated_date, @price,@age_restriction,
                                          @is_verified, @is_advertised,
@application_files_id,@app_support_id, @developer_id)
            END
END
GO
```

```
GO
ALTER PROCEDURE dbo.validateAndInsertApplications (@application_id INT,@version_id FLOAT, @name VARCHAR(20),
                @description VARCHAR(20),@version_fix VARCHAR(50),@created_date DATE ,@updated_date DATE, @price FLOAT,@age_restriction CHAR(4),
                @is_verified BIT, @is_advertised BIT, @application_files_id INT,@app_support_id INT, @developer_id INT)
AS
BEGIN
Declare @nameInserted varchar(20)
        (SELECT top 1 @nameInserted =  name FROM Applications WHERE application_id = @application_id)
Declare @appID INT
         SELECT top 1 @appID =  application_id FROM Applications WHERE name = @name

        IF
        @name != @nameInserted

        BEGIN
                RAISERROR('Application name is already present, and it has to be unique',1,1)
        END
        ELSE
        BEGIN
            INSERT INTO Applications (application_id,version_id,name,description,version_fix,created_date,updated_date,price,age_restriction,
                        is_verified,is_advertised,application_files_id,app_support_id,developer_id)
            VALUES (@application_id,@version_id, @name,@description,@version_fix,@created_date,@updated_date, @price,@age_restriction,
                        @is_verified, @is_advertised, @application_files_id,@app_support_id, @developer_id)
        END
END
GO
```

Execution of the SP for insertion of values.

The snippet shows an incorrect entry and a correct entry of the application name.





SP for inserting password and encrypt it.

The password is inserted and stored in binary form.

Before inserting a hash is created using HASHBYTES function.

It adds the inserted password with a new uniqueidentifier value.

```
GO
CREATE PROCEDURE InsertUserCredentials
       @user_name VARCHAR(12),
       @password VARCHAR(64),
       @email_address VARCHAR(18),
       @is_verified BIT,
       @is_active BIT,
       @user_id INT,
    @responseMessage NVARCHAR(250) OUTPUT
AS
BEGIN
    SET NOCOUNT ON

    DECLARE @id uniqueidentifier
       SET @id = NEWID()
    BEGIN TRY

              INSERT INTO User_Account_Details (account_id,user_name, password,
[password_hash], email_address, is_verified, is_active, user_id)
              VALUES (NEXT VALUE FOR UserAccountDetailsSeq, @user_name,
HASHBYTES('SHA2_512', @password + CAST(@id AS VARCHAR(64))),
               @id,@email_address,@is_verified,@is_active,@user_id);

        SET @responseMessage='Success'

    END TRY
    BEGIN CATCH
        SET @responseMessage=ERROR_MESSAGE()
    END CATCH
END

DECLARE @responseMessage NVARCHAR(250)

EXEC InsertUserCredentials
       @user_name = N'karan',
       @password = N'123',
       @email_address = N'karan@gmail.com',
       @is_verified = N'1',
       @is_active = N'1',
       @user_id = N'10002',
    @responseMessage = @responseMessage OUTPUT

SELECT @responseMessage as N'@responseMessage'
```

```
GO
CREATE PROCEDURE InsertUserCredentials
    @user_name VARCHAR(12),
    @password VARCHAR(64),
    @email_address VARCHAR(18),
    @is_verified BIT,
    @is_active BIT,
    @user_id INT,
    @responseMessage NVARCHAR(250) OUTPUT
AS
BEGIN
    SET NOCOUNT ON
    DECLARE @id uniqueidentifier
    SET @id = NEWID()
    BEGIN TRY

        INSERT INTO User_Account_Details (account_id,user_name, password, [password_hash], email_address, is_verified, is_active, user_id)
        VALUES (NEXT VALUE FOR UserAccountDetailsSeq, @user_name, HASHBYTES('SHA2_512', @password + CAST(@id AS VARCHAR(64))),
         @id,@email_address,@is_verified,@is_active,@user_id);

        SET @responseMessage='Success'

    END TRY
    BEGIN CATCH
        SET @responseMessage=ERROR_MESSAGE()
    END CATCH
END
```

```
DECLARE @responseMessage NVARCHAR(250)

EXEC InsertUserCredentials
    @user_name = N'nilesh',
    @password = N'123',
    @email_address = N'karan@gmail.com',
    @is_verified = N'1',
    @is_active = N'1',
    @user_id = N'10002',
    @responseMessage = @responseMessage OUTPUT

SELECT  @responseMessage as N'@responseMessage'
```

90 %

Results | Messages

	@responseMessage
1	Success

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion3 | 00:00:00 | 1 rows

SP for decrypting the password and validate the user.
The reverse is done in this procedure to decrypt the password.

```
GO
ALTER PROCEDURE LoginUSer

    @user_name VARCHAR(12),
    @password VARCHAR(64),
    @responseMessage NVARCHAR(250)='' OUTPUT
AS
BEGIN
    SET NOCOUNT ON
       DECLARE @account_id INT

    IF EXISTS (SELECT TOP 1 account_id FROM dbo.User_Account_Details WHERE
user_name=@user_name)
    BEGIN
        SET @account_id = (SELECT account_id FROM dbo.User_Account_Details WHERE
user_name = @user_name
            AND password = HASHBYTES('SHA2_512', @password + CAST(password_hash AS
VARCHAR(64))))

       IF(@account_id IS NULL)
           SET @responseMessage='Incorrect password'
       ELSE
           SET @responseMessage='User successfully logged in'
    END
    ELSE
       SET @responseMessage='Invalid login'
END

DECLARE       @responseMessage nvarchar(250)

--Correct login and password
EXEC   LoginUSer
              @user_name = N'sid77',
              @password = N'sid77',
              @responseMessage = @responseMessage OUTPUT

SELECT @responseMessage as N'@responseMessage'
```

```
GO
ALTER PROCEDURE LoginUSer

    @user_name VARCHAR(12),
    @password VARCHAR(64),
    @responseMessage NVARCHAR(250)='' OUTPUT
AS
BEGIN
    SET NOCOUNT ON
    DECLARE @account_id INT

    IF EXISTS (SELECT TOP 1 account_id FROM dbo.User_Account_Details WHERE user_name=@user_name)
    BEGIN
        SET @account_id = (SELECT account_id FROM dbo.User_Account_Details WHERE user_name = @user_name
        AND password = HASHBYTES('SHA2_512', @password + CAST(password_hash AS VARCHAR(64))))

       IF(@account_id IS NULL)
           SET @responseMessage='Incorrect password'
       ELSE
           SET @responseMessage='User successfully logged in'
    END
    ELSE
       SET @responseMessage='Invalid login'
END
GO
```

```
DECLARE @responseMessage nvarchar(250)

EXEC    LoginUSer
        @user_name = N'nilesh',
        @password = N'123',
        @responseMessage = @responseMessage OUTPUT

SELECT  @responseMessage as N'@responseMessage'
```

90 %

Results | Messages

	@responseMessage
1	User successfully logged in

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion3 | 00:00:00 | 1 rows

When the credentials are incorrect.

```
DECLARE @responseMessage nvarchar(250)

EXEC    LoginUSer
        @user_name = N'nilesh1',
        @password = N'123',
        @responseMessage = @responseMessage OUTPUT

SELECT  @responseMessage as N'@responseMessage'
```

90 %

Results | Messages

	@responseMessage
1	Invalid login

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion3 | 00:00:00 | 1 rows

```
DECLARE @responseMessage nvarchar(250)

EXEC    LoginUSer
        @user_name = N'nilesh',
        @password = N'122',
        @responseMessage = @responseMessage OUTPUT

SELECT  @responseMessage as N'@responseMessage'
```

90 %

Results | Messages

	@responseMessage
1	Incorrect password

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion3 | 00:00:00 | 1 rows

Trigger to backup applications after the 6th version is entered in the application for a app.

The backup is taken into Applications Archive Table.

```
GO
ALTER TRIGGER trgArchiveApplications
ON Applications
AFTER INSERT AS
BEGIN

DECLARE @count INT
DECLARE @inserted_application_id INT
        SELECT @inserted_application_id = (SELECT application_id FROM INSERTED)
        (SELECT  @count =  count(application_id) FROM Applications WHERE application_id = @inserted_application_id)

        IF (@count >= 6)
            BEGIN
            INSERT INTO Applications_Archive SELECT TOP (1) application_id,version_id,name,description,
            version_fix,created_date,updated_date,price,age_restriction,application_files_id,app_support_id,developer_id
            FROM Applications WHERE application_id = @inserted_application_id

                RAISERROR('Data insertion to applications archive successfull',1,1)
            END
END
GO
```

```
INSERT INTO Applications VALUES ('102','4.00','Spotify','professional','',
'7/8/2014','4/21/2018','40','','1','1','1232','1234','1005');
```

90 %

Messages

```
(1 row affected)
Data insertion to applications archive successfull
Msg 50000, Level 1, State 1

(1 row affected)
```

90 %

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion2 | 00:00:00 | 0 rows

Created a VIEW as we need to the RATING of application to be the average of all. Instead of querying many table, created a view which is used to in the application using PHP.

```
ALTER VIEW Category_Applications
AS SELECT DISTINCT A.name, ROUND(AVG(AR.rating),2) AS Ratings,C.category_name AS Category
FROM Categories C
INNER JOIN Application_Categories AC
ON (C.category_id = AC.category_id)
INNER JOIN Applications A
ON (AC.application_id = A.application_id)
AND (AC.version_id = A.version_id)
INNER JOIN Application_Reviews AR
ON (AR.application_id = A.application_id)
AND (AR.version_id = A.version_id)
GROUP BY A.name,C.category_name
```

```
GO
CREATE VIEW Category_Applications
AS
SELECT DISTINCT A.name,ROUND(AVG(AR.rating),2) AS Ratings,C.category_name AS Category FROM Categories C
INNER JOIN Application_Categories AC
ON (C.category_id = AC.category_id)
INNER JOIN Applications A
ON (AC.application_id = A.application_id)
AND (AC.version_id = A.version_id)
INNER JOIN Application_Reviews AR
ON (AR.application_id = A.application_id)
AND (AR.version_id = A.version_id)
GROUP BY A.name, C.category_name
GO
```

```
SELECT * FROM Category_Applications
```

90 %

Results | Messages

	name	Ratings	Category
1	Whatsapp	3.9	gaming
2	Spotify	4.9	music
3	Gmail	4	social
4	Spotify	4.9	social
5	Whatsapp	4.3	social

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (55) | AppStoreVersion2 | 00:00:00 | 5 rows

```
insert into Applications values ('101','2.00','Whatsapp','Chat app','',
'7/8/2014','4/18/2018','40','','1','1','1232','1234','1001');
```

90 %

Messages

```
No insertion to applications archive
Msg 50000, Level 1, State 1

(1 row affected)
```

90 %

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (54) | AppStoreVersion2 | 00:00:00 | 0 rows

```
SELECT * FROM Applications
```

90 %

Results | Messages

	application_id	version_id	name	description	version_fix	created_date	updated_date	price	age_restriction	is_verified	is_advertised	application_files_id
1	101	1	Whatsapp	Chat		2014-07-08	2018-06-18	40		1	1	1232
2	101	2	Whatsapp	Chat app		2014-07-08	2018-04-18	40		1	1	1232
3	101	5	WhatsApp	It is used to connect with people and chat, vide...	contact details bug fixed	2014-07-08	2015-07-18	40		1	1	1232
4	101	6	WhatsApp	It is used to connect with people and chat, vide...	contact details bug fixed	2014-07-08	2018-07-18	40		1	1	1232
5	101	7	WhatsApp	It is used to connect with people and chat, vide...	location bug fixed	2014-02-18	2016-07-18	20	NULL	1	1	1234

```
insert into Applications values ('101','8.00','Whatsapp','It is used to connect with people and chat, video call, etc.','',
'7/8/2014','9/18/2018','40','','1','1','1232','1234','1001');
```

90 %

Messages

```
(1 row affected)

(1 row affected)
```

90 %

Query executed successfully. | ANUJA\SQL17NEW (14.0 RTM) | nilesh (54) | AppStoreVersion2 | 00:00:00 | 0 rows

Created sequences on many tables to auto increment the primary keys after INSERT.

UserAccountDetailsSeq sequence created is used for inserting values through SP.

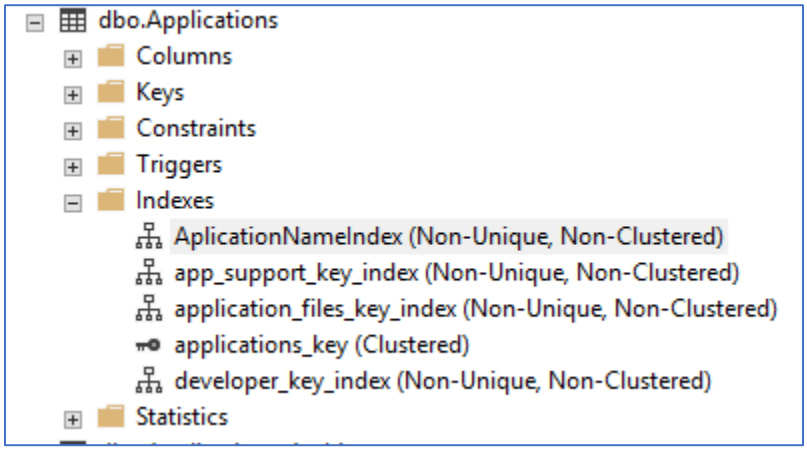
```
CREATE SEQUENCE UserAccountDetailsSeq
    AS int
    START WITH 20001
    INCREMENT BY 1 ;
```

```
GO
CREATE PROCEDURE InsertUserCredentials
    @user_name VARCHAR(12),
    @password VARCHAR(64),
    @email_address VARCHAR(18),
    @is_verified BIT,
    @is_active BIT,
    @user_id INT,
    @responseMessage NVARCHAR(250) OUTPUT
AS
BEGIN
    SET NOCOUNT ON
    DECLARE @id uniqueidentifier
    SET @id = NEWID()
    BEGIN TRY

        INSERT INTO User_Account_Details (account_id,user_name, password, [password_hash], email_address, is_verified, is_active, user_id)
    VALUES (NEXT VALUE FOR UserAccountDetailsSeq, @user_name, HASHBYTES('SHA2_512', @password + CAST(@id AS VARCHAR(64))),
     @id,@email_address,@is_verified,@is_active,@user_id);

        SET @responseMessage='Success'

    END TRY
    BEGIN CATCH
        SET @responseMessage=ERROR_MESSAGE()
    END CATCH
END
```

Using Toad Modeler Indexes are created easily; however you can create an index manually.

Have created many non-clustered indexes on column which are used often.

The retrieval of the data is faster and query performance increases by using indexes.

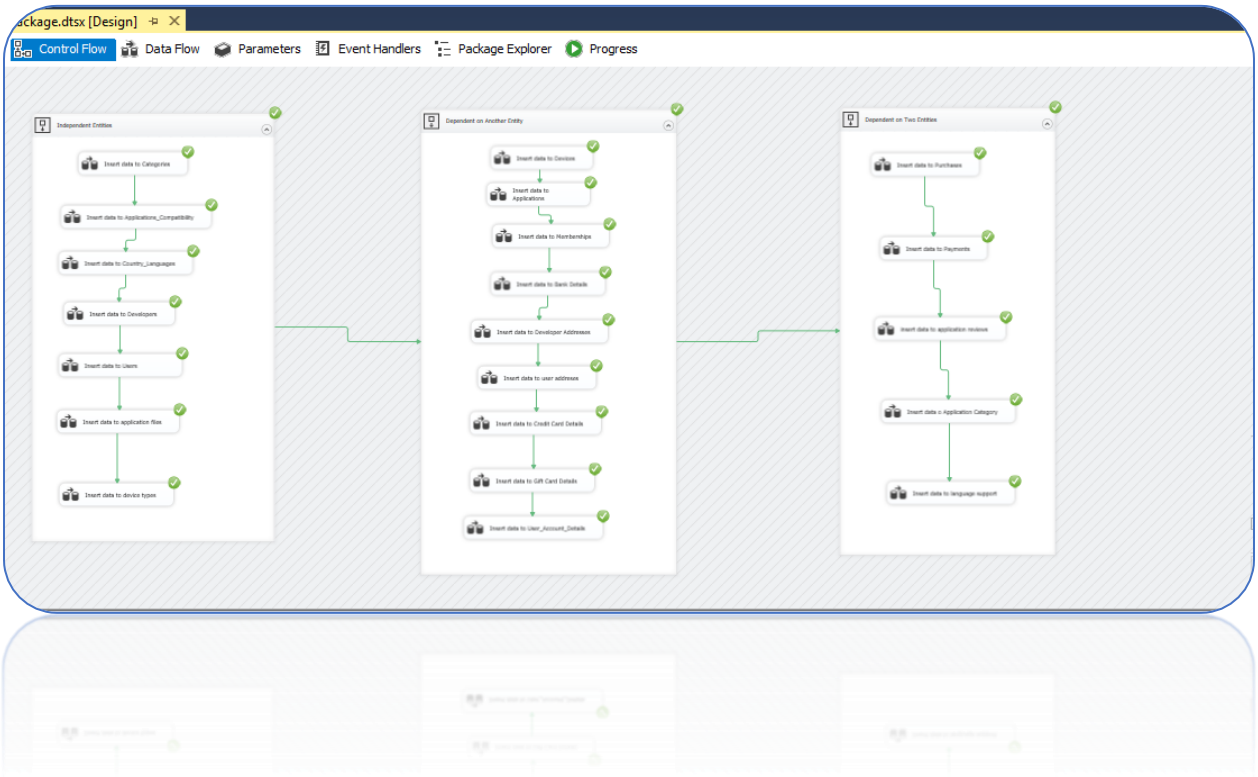


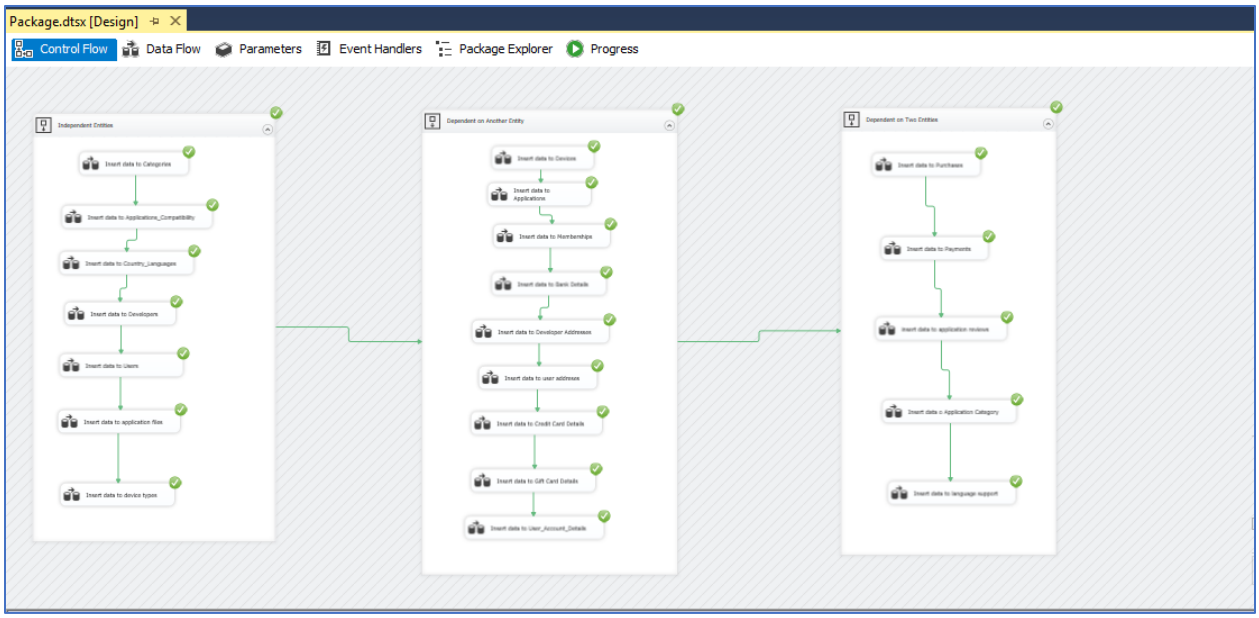
Created a SSIS package which loads data from excel to the data base using sequence container.

The first container loads independent tables .

The second loads dependent tables.

The third container loads remaining tables and bridge tables.




Package.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Independent Entities
Insert data to Categories
Insert data to Applications_Compatibility
Insert data to Country_Languages
Insert data to Developers
Insert data to Users
Insert data to application files
Insert data to device types
Dependent on Another Entity
Insert data to Devices
Insert data to Applications
Insert data to Memberships
Insert data to Bank Details
Insert data to Developer Addresses
Insert data to user addresses
Insert data to Credit Card Details
Insert data to Gift Card Details
Insert data to User_Account_Details
Dependent on Two Entities
Insert data to Purchases
Insert data to Payments
Insert data to application reviews
Insert data o Application Category
Insert data to language support