InsightConfig Table migration Instructions:

MSSQL to Postgres Table Migration Checklist.

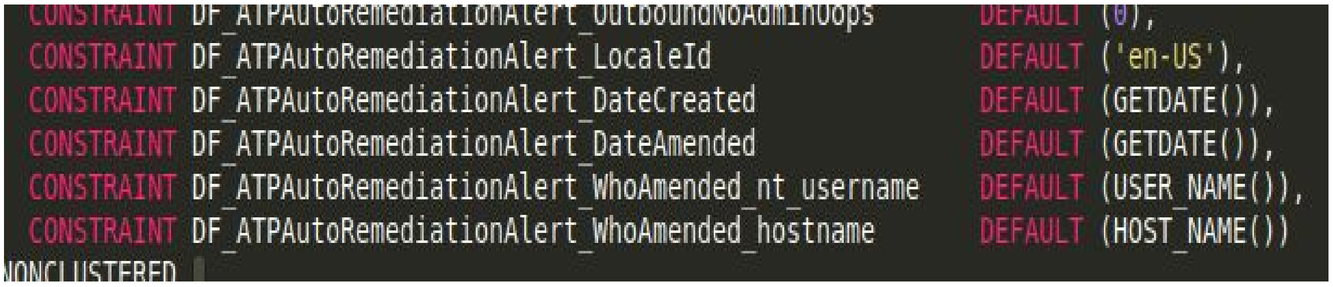
1) File Name should be TableName.sql for individual tables.
2) Table Name and Column Names should be similar as per MsSQl source and with camel case with Double Quotation.
3) Order of Column names should be the same as source for **Table, Constraints and Indexes.**
4) Column Data Type should be equivalent for each column as per given document.
   eg) MSSQL - timestamp to Postgres - BYTEA
5) Alignment should be in the approved format shared already.
6) Once it is done just compare with MsSQl source file and check it line by line.

1)Datatype Mapping

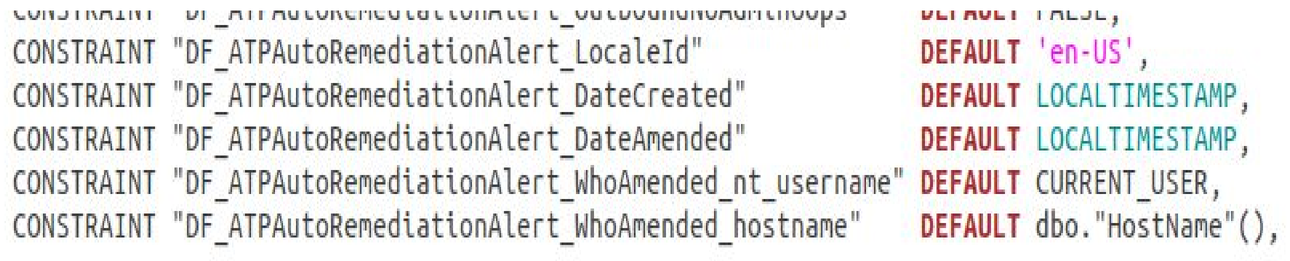
Datatype Mapping			
SQL Server		**Postgres**	
BIGINT	64-bit integer	**BIGINT**	
BINARY(n)	Fixed length byte string	**BYTEA**	
BIT	1, 0 or NULL	**BOOLEAN**	
CHAR(n)	Fixed length char string, 1 <= n <= 8000	**CHAR(n)**	
VARCHAR(n)	Variable length char string, 1 <= n <= 8000	**VARCHAR(n)**	
VARCHAR(max)	Variable length char string, <= 2GB	**TEXT**	
VARBINARY(n)	Variable length byte string , 1 <= n <= 8000	**BYTEA**	
VARBINARY(max)	Variable length byte string , <= 2GB	**BYTEA**	
NVARCHAR(n)	Variable length Unicode UCS-2 string	**VARCHAR(n)**	
NVARCHAR(max)	Variable length Unicode UCS-2 data, <= 2GB	**TEXT**	
TEXT	Variable length character data, <= 2GB	**TEXT**	

NTEXT	Variable length Unicode UCS-2 data, <= 2GB	**TEXT**	
DOUBLE PRECISION	Double precision floating point number	**DOUBLE PRECISION**	
FLOAT(p)	Floating point number	**DOUBLE PRECISION**	
INTEGER	32 bit integer	**INTEGER**	
NUMERIC(p,s)	Fixed point number	**NUMERIC(p,s)**	
DATE	Date includes year, month and day	**DATE**	
DATETIME	Date and Time with fraction	**TIMESTAMP(3)**	
DATETIME2(p)	Date and Time with fraction	**TIMESTAMP(n)**	
DATETIMEOFFSET(p)	Date and Time with fraction and time zone	**TIMESTAMP(p) WITH TIME ZONE**	
SMALLDATETIME	Date and Time	**TIMESTAMP(0)**	
TINYINT	8 bit unsigned integer, 0 to 255	**SMALLINT**	
UNIQUEIDENTIFIER	16 byte GUID(UUID) data	**UUID**	
ROWVERSION	Automatically updated binary data	**BYTEA**	
SMALLMONEY	32 bit currency amount	**MONEY**	
IMAGE	Variable length binary data, <= 2GB	**BYTEA**	
BIT(32)		BYTEA	
BIT VARYING(16)		BYTEA	

2)Default constraints



```
 CONSTRAINT DF_ATPAutoRemediationAlert_OutboundNoAdminOops       DEFAULT (0),
 CONSTRAINT DF_ATPAutoRemediationAlert_LocaleId                  DEFAULT ('en-US'),
 CONSTRAINT DF_ATPAutoRemediationAlert_DateCreated               DEFAULT (GETDATE()),
 CONSTRAINT DF_ATPAutoRemediationAlert_DateAmended               DEFAULT (GETDATE()),
 CONSTRAINT DF_ATPAutoRemediationAlert_WhoAmended_nt_username    DEFAULT (USER_NAME()),
 CONSTRAINT DF_ATPAutoRemediationAlert_WhoAmended_hostname       DEFAULT (HOST_NAME())
NONCLUSTERED
```

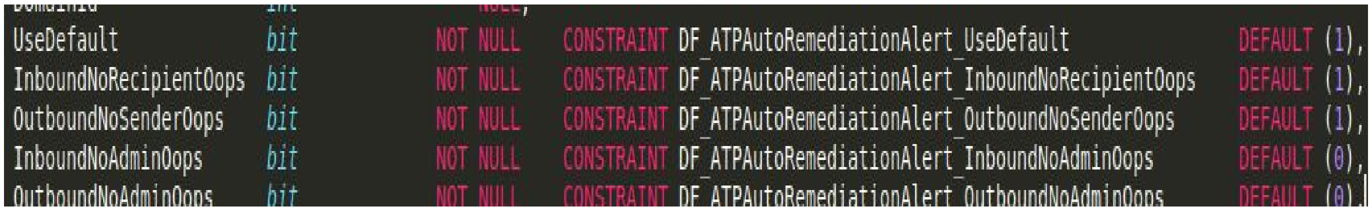


```
 CONSTRAINT "DF_ATPAutoRemediationAlert_OutboundNoAdminOops"     DEFAULT FALSE,
 CONSTRAINT "DF_ATPAutoRemediationAlert_LocaleId"                DEFAULT 'en-US',
 CONSTRAINT "DF_ATPAutoRemediationAlert_DateCreated"             DEFAULT LOCALTIMESTAMP,
 CONSTRAINT "DF_ATPAutoRemediationAlert_DateAmended"             DEFAULT LOCALTIMESTAMP,
 CONSTRAINT "DF_ATPAutoRemediationAlert_WhoAmended_nt_username" DEFAULT CURRENT_USER,
 CONSTRAINT "DF_ATPAutoRemediationAlert_WhoAmended_hostname"     DEFAULT dbo."HostName"(),
```

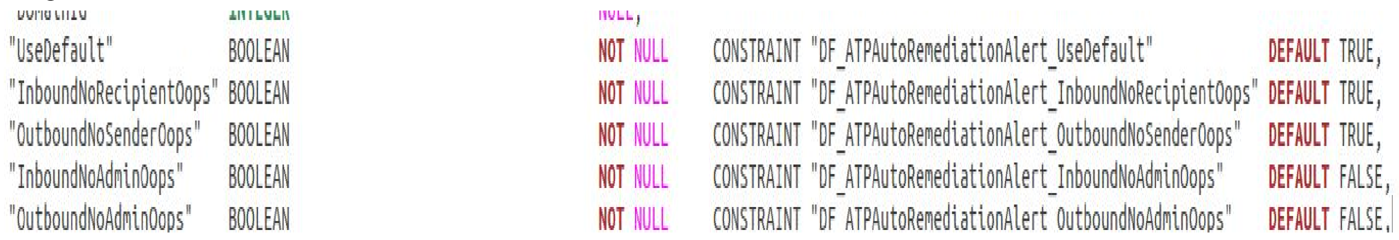
Default functions mapping

	MsSQL	Postgres
1)	DEFAULT ('en-US')	DEFAULT 'en-US'
2)	DEFAULT (GETDATE())	DEFAULT LOCALTIMESTAMP
3)	DEFAULT (USER_NAME())	DEFAULT CURRENT_USER
4)	DEFAULT (HOST_NAME())	DEFAULT dbo."HostName"()
5)	DEFAULT NEWID()	DEFAULT dbo.UUID_GENERATE_V4()
	DEFAULT (GETUTCDATE())	DEFAULT timezone('UTC'::text, now())
	IDENTITY (1,1)	GENERATED ALWAYS AS IDENTITY
	IDENTITY(N, 1)	GENERATED ALWAYS AS IDENTITY (START WITH N)

If there is a default value for COLUMN_NAME **bit(data type)** columns(**DEFAULT 0 or DEFAULT 1**) in MsSQL,
convert it like to postgres
COLUMN_NAME **BOOLEAN(data type)** columns **(DEFAULT FALSE OR TRUE)**

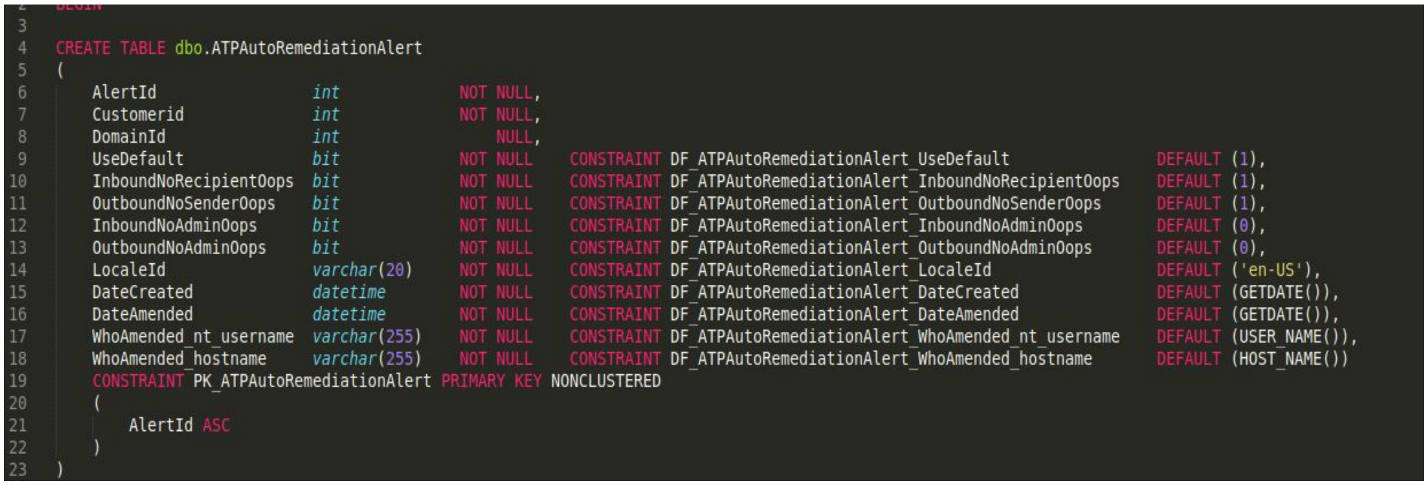
Eg:
MsSQL


```
UseDefault             bit          NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_UseDefault              DEFAULT (1),
InboundNoRecipientOops bit          NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_InboundNoRecipientOops  DEFAULT (1),
OutboundNoSenderOops   bit          NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_OutboundNoSenderOops    DEFAULT (1),
InboundNoAdminOops     bit          NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_InboundNoAdminOops      DEFAULT (0),
OutboundNoAdminOops    bit          NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_OutboundNoAdminOops     DEFAULT (0),
```

Postgres:


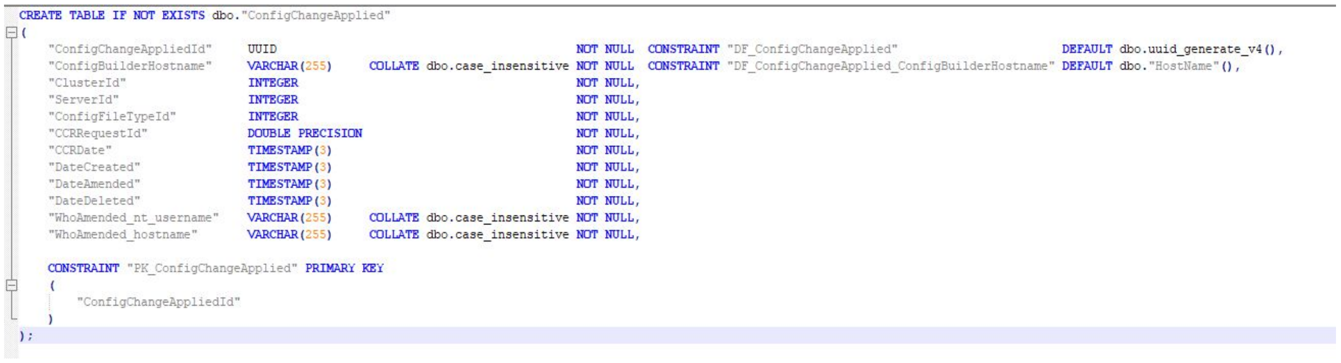
```
"UseDefault"             BOOLEAN          NOT NULL    CONSTRAINT "DF_ATPAutoRemediationAlert_UseDefault"              DEFAULT TRUE,
"InboundNoRecipientOops" BOOLEAN          NOT NULL    CONSTRAINT "DF_ATPAutoRemediationAlert_InboundNoRecipientOops" DEFAULT TRUE,
"OutboundNoSenderOops"   BOOLEAN          NOT NULL    CONSTRAINT "DF_ATPAutoRemediationAlert_OutboundNoSenderOops"   DEFAULT TRUE,
"InboundNoAdminOops"     BOOLEAN          NOT NULL    CONSTRAINT "DF_ATPAutoRemediationAlert_InboundNoAdminOops"     DEFAULT FALSE,
"OutboundNoAdminOops"    BOOLEAN          NOT NULL    CONSTRAINT "DF_ATPAutoRemediationAlert_OutboundNoAdminOops"    DEFAULT FALSE,
```

3)Collation

Add collation(**COLLATE dbo.case_insensitive)** for string data type columns(**CHAR, VARCHAR, TEXT**)
Eg:
MsSQL:


```

CREATE TABLE dbo.ATPAutoRemediationAlert
(
    AlertId                int           NOT NULL,
    Customerid             int           NOT NULL,
    DomainId               int               NULL,
    UseDefault             bit           NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_UseDefault              DEFAULT (1),
    InboundNoRecipientOops bit           NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_InboundNoRecipientOops  DEFAULT (1),
    OutboundNoSenderOops   bit           NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_OutboundNoSenderOops    DEFAULT (1),
    InboundNoAdminOops     bit           NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_InboundNoAdminOops      DEFAULT (0),
    OutboundNoAdminOops    bit           NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_OutboundNoAdminOops     DEFAULT (0),
    LocaleId               varchar(20)   NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_LocaleId                DEFAULT ('en-US'),
    DateCreated            datetime      NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_DateCreated             DEFAULT (GETDATE()),
    DateAmended            datetime      NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_DateAmended             DEFAULT (GETDATE()),
    WhoAmended_nt_username varchar(255)  NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_WhoAmended_nt_username  DEFAULT (USER_NAME()),
    WhoAmended_hostname    varchar(255)  NOT NULL    CONSTRAINT DF_ATPAutoRemediationAlert_WhoAmended_hostname     DEFAULT (HOST_NAME())
    CONSTRAINT PK_ATPAutoRemediationAlert PRIMARY KEY NONCLUSTERED
    (
        AlertId ASC
    )
)
```

Postgres:



```
CREATE TABLE IF NOT EXISTS dbo."ConfigChangeApplied"
(
    "ConfigChangeAppliedId"     UUID                                             NOT NULL  CONSTRAINT "DF_ConfigChangeApplied"                         DEFAULT dbo.uuid_generate_v4(),
    "ConfigBuilderHostname"     VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL  CONSTRAINT "DF_ConfigChangeApplied_ConfigBuilderHostname" DEFAULT dbo."HostName"(),
    "ClusterId"                 INTEGER                                          NOT NULL,
    "ServerId"                  INTEGER                                          NOT NULL,
    "ConfigFileTypeId"          INTEGER                                          NOT NULL,
    "CCRRequestId"              DOUBLE PRECISION                                 NOT NULL,
    "CCRDate"                   TIMESTAMP(3)                                     NOT NULL,
    "DateCreated"               TIMESTAMP(3)                                     NOT NULL,
    "DateAmended"               TIMESTAMP(3)                                     NOT NULL,
    "DateDeleted"               TIMESTAMP(3)                                     NOT NULL,
    "WhoAmended_nt_username"    VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL,
    "WhoAmended_hostname"       VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL,

    CONSTRAINT "PK_ConfigChangeApplied" PRIMARY KEY
    (
        "ConfigChangeAppliedId"
    )
);
```

4)Table and Index creating syntax

Tables and Indexes are re executable

Eg:

<u>Tables:</u>

MsSQL:

    CREATE TABLE dbo."AllDomains"

Postgres:

    CREATE TABLE **IF NOT EXISTS** dbo."AllDomains"

<u>Indexes:</u>

MsSQL:

    CREATE INDEX "IX_AllDomains_CL01"

Postgres:

    CREATE INDEX **IF NOT EXISTS** "IX_AllDomains_CL01"

MsSQL:

    CREATE UNIQUE INDEX "IX_AllDomains_NU02"

Postgres:

    CREATE UNIQUE INDEX **IF NOT EXISTS** "IX_AllDomains_NU02"

5)DB setup

- Create database db_name;
- Create schema dbo;

<u>Add collations and extensions:</u>

- CREATE COLLATION IF NOT EXISTS dbo.case_insensitive (provider = icu, locale = 'und-u-ks-level2', deterministic = false);

- CREATE EXTENSION IF NOT EXISTS "uuid-ossp" schema dbo;

Once the table migration is done just execute and test the tables in the database. It is executable or not.

6)TO ENSURE

- Ensure all the **Table name**, **Column name, Constraint name, Indexes name** in Postgres as per **InsightConfig(MsSQL)** tables with double-quotes.
- The data type must be in **UPPER CASES** and follow the above data type mapping
- Add **NULL** if there is no **NOT NULL** on the column name and please ensure **NULL** or **NOT NULL** is there for all columns.
- If creating Index for the **NULL** columns please follow the below

```
CREATE INDEX IF NOT EXISTS "IDX_Customers_NC01" ON "dbo"."Customers"
("CustomerEmail" ASC NULLS FIRST);
```

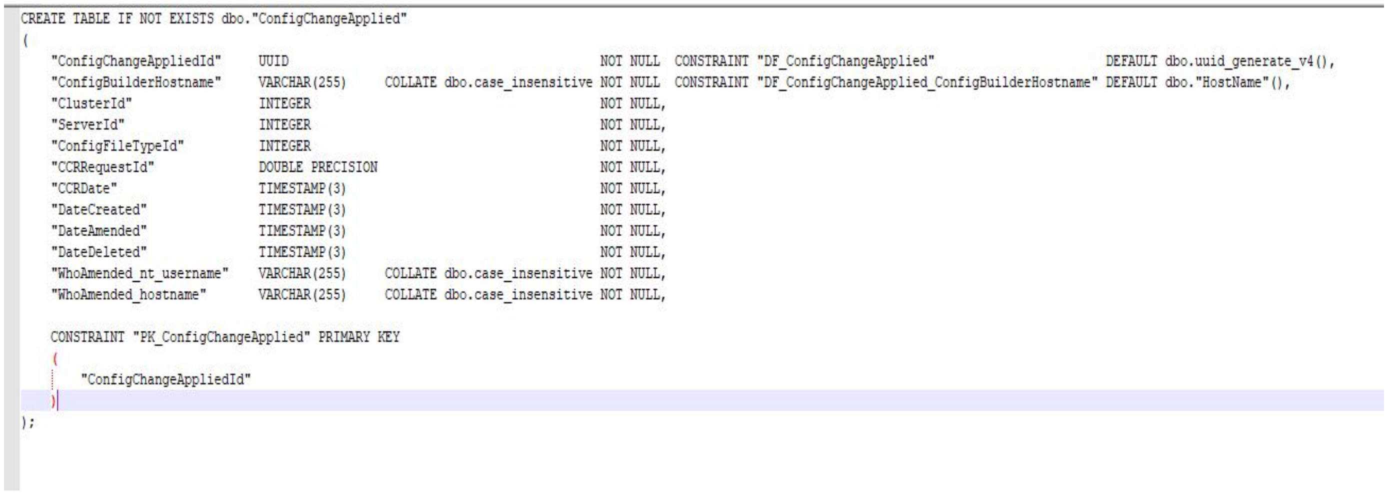
7)Table Review Points

```
1)Make sure the table script is executable.

2)Alignment correction.

3)Remove unwanted contents except for Tables and indexes in the file.

4)Make sure placed the commas(,)

5)Make sure that used UPPER CASES for keywords.
```

For Example:

Alignment should be like below in screen shot.

Table column name and data type should have 4 spaces and all others should have single space.(Don't use Tabs for alignment)



```
CREATE TABLE IF NOT EXISTS dbo."ConfigChangeApplied"
(
    "ConfigChangeAppliedId"     UUID                                       NOT NULL  CONSTRAINT "DF_ConfigChangeApplied"                        DEFAULT dbo.uuid_generate_v4(),
    "ConfigBuilderHostname"     VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL  CONSTRAINT "DF_ConfigChangeApplied_ConfigBuilderHostname" DEFAULT dbo."HostName"(),
    "ClusterId"                 INTEGER                                    NOT NULL,
    "ServerId"                  INTEGER                                    NOT NULL,
    "ConfigFileTypeId"          INTEGER                                    NOT NULL,
    "CCRRequestId"              DOUBLE PRECISION                           NOT NULL,
    "CCRDate"                   TIMESTAMP(3)                               NOT NULL,
    "DateCreated"               TIMESTAMP(3)                               NOT NULL,
    "DateAmended"               TIMESTAMP(3)                               NOT NULL,
    "DateDeleted"               TIMESTAMP(3)                               NOT NULL,
    "WhoAmended_nt_username"    VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL,
    "WhoAmended_hostname"       VARCHAR(255)     COLLATE dbo.case_insensitive NOT NULL,

    CONSTRAINT "PK_ConfigChangeApplied" PRIMARY KEY
    (
        "ConfigChangeAppliedId"
    )
);
```

Functions,Collations,Extensions,etc for execution :

Dummy dbo."HostName"() Function :

```
create or replace function dbo."HostName"()

returns varchar

as

$$

begin

return 'system';

end;

$$

language plpgsql;
```

Import this Extension to generate 'dbo.UUID_GENERATE_V4()' Function :

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp" SCHEMA dbo;
```

For dbo.case_insensitive Collations:

```
CREATE COLLATION IF NOT EXISTS dbo.case_insensitive (provider = icu, locale =
'und-u-ks-level2', deterministic = false);
```

Foreign Key Reference.

MSSQL Foreign Key creation Script:

```
IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_ACSCustomer_ACSKitchen]') AND parent_object_id =
OBJECT_ID(N'[dbo].[ACSCustomer]'))
      ALTER TABLE [dbo].ACSCustomer DROP CONSTRAINT FK_ACSCustomer_ACSKitchen
GO

ALTER TABLE [dbo].ACSCustomer WITH CHECK ADD CONSTRAINT
FK_ACSCustomer_ACSKitchen FOREIGN KEY(PrimaryKitchenId)
      REFERENCES [dbo].ACSKitchen (KitchenId)
GO

ALTER TABLE [dbo].ACSCustomer CHECK CONSTRAINT FK_ACSCustomer_ACSKitchen
GO
```

Postgres Foreign Key conversion Script:

```
DO

$$

BEGIN

   IF EXISTS (SELECT * FROM information_schema.table_constraints WHERE CONSTRAINT_NAME
= 'FK_ACSCustomer_ACSKitchen' AND TABLE_NAME = 'ACSCustomer') THEN

       ALTER TABLE dbo."ACSCustomer" DROP CONSTRAINT "FK_ACSCustomer_ACSKitchen";

   END IF;

END;

$$;

ALTER TABLE dbo."ACSCustomer" ADD CONSTRAINT "FK_ACSCustomer_ACSKitchen" FOREIGN
KEY("PrimaryKitchenId")

       REFERENCES dbo."ACSKitchen"("KitchenId");
```

Postgres Common Foreign Key creation Script:

```
DO

$$

BEGIN

   IF EXISTS (SELECT * FROM information_schema.table_constraints WHERE CONSTRAINT_NAME
= 'ConstraintName' AND TABLE_NAME = 'TableName') THEN

       ALTER TABLE dbo."TableName" DROP CONSTRAINT "ConstraintName";

   END IF;

END;

$$;

ALTER TABLE dbo."TableName" ADD CONSTRAINT "ConstraintName" FOREIGN KEY("ColumnName")

       REFERENCES dbo."ReferenceTableName"("ReferenceColumnName");
```