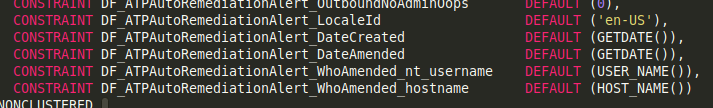
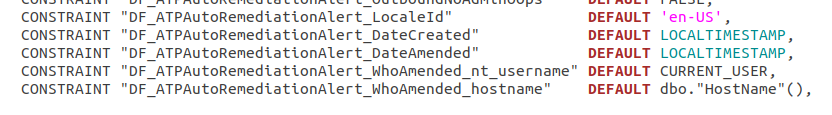
InsightConfig Table migration Instructions:

### **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**





#### **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) |

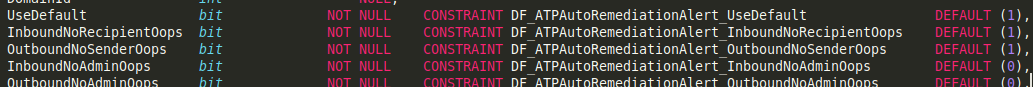
If there is a default value for COLUMN\_NAME **bit(data type)** columns(**DEFAULT 0 or DEFAULT 1**) in MsSQL,

convert it like to postgres

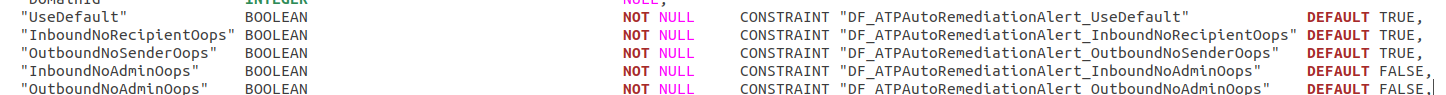
COLUMN\_NAME **BOOLEAN(data type)** columns **(DEFAULT FALSE OR TRUE)**

Eg:

MsSQL



Postgres:

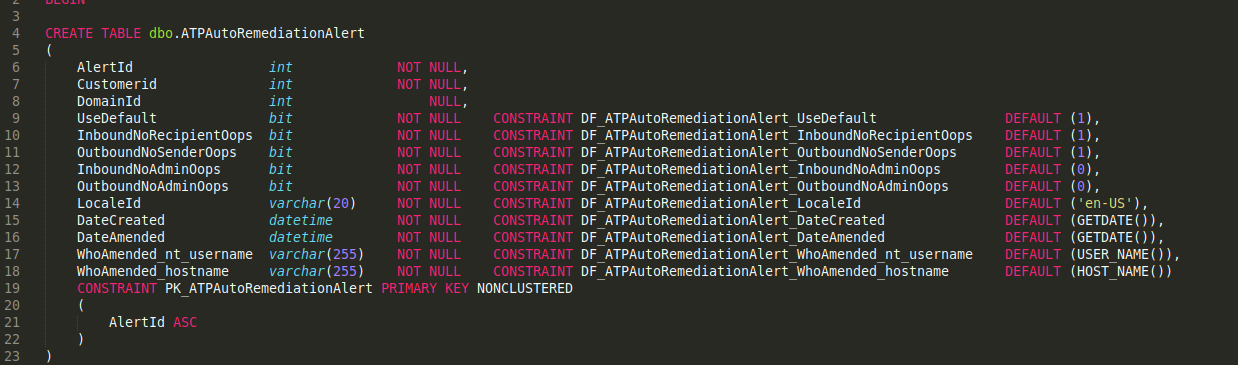


### 3)Collation

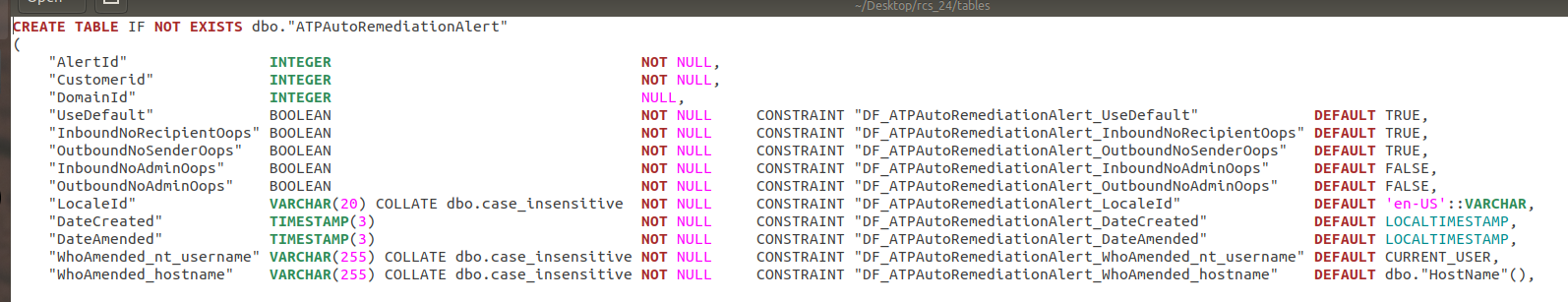
Add collation(**COLLATE dbo.case\_insensitive)** for string data type columns(**CHAR, VARCHAR, TEXT**)

Eg:

**MsSQL:**

****

**Postgres:**

****

### **4)Table and Index creating syntax**

Tables and Indexes are re executable

Eg:

Tables:

MsSQL:

CREATE TABLE dbo."AllDomains"

Postgres:

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

### Indexes:

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;

Add collations and extensions:

* 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.

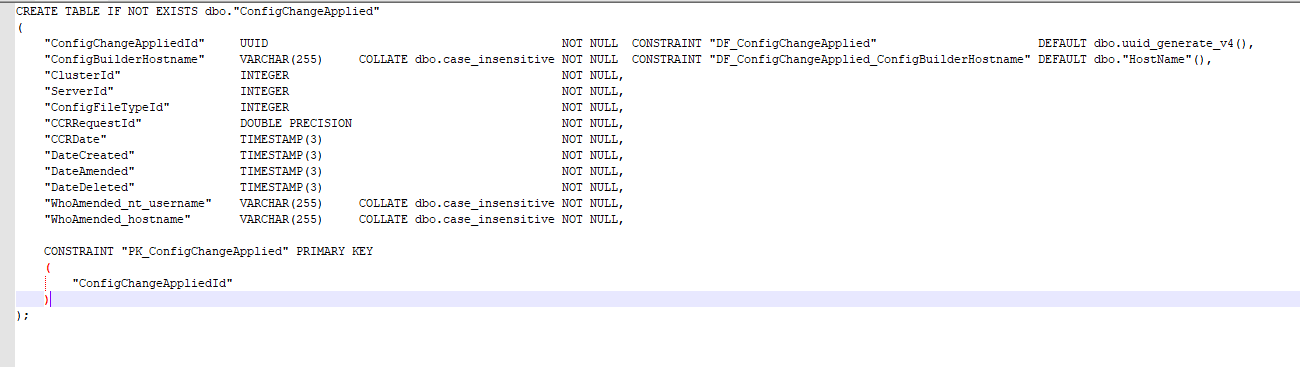
4)Make sure placed the commas(,)

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)**



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**");