Crime Project - Gangs, Guns and Drugs

Contents

1 Introduction

The Home Office is coming under increasing pressure to tackle the rising levels of crime in the UK. To justify expenditure and better target resources the government has enlisted Kubrick Group consultants to land, profile, cleanse, model and analyse data on crime and present analyses via an attractive and informative dashboard.

Given this brief, I set out to look at gang-related crime across England and Wales with a hope to offer new means of analysis and therefore high quality information to assist in acting upon the data available.

The main source of data used was directly from the UK police, however various pieces of external data were also utilised in order to enrich and allow further analysis for the end user.

2 Aims of the Project

The idea to look at gang crime was conceptualised after researching modern crime trends in the UK and discovering the *Amnesty report on the Met Police's gang matrix*. To condense the message of this report, the current prime method for police in England (and specifically London) to tackle gang crime is through a racially biased system that has been criticised by one of the leading human rights institutions in the world.

I wanted to offer a new solution.

To achieve such a goal, I entirely eliminated ethnicity and race from the data during the engineering phase so as to create a 'colour-blind' model to analyse. The hope was to visualise potential gang hotspots and areas derived from instances of gang-related crime; including drug use, weapon crime, theft and robbery as well as violent or sexual offences.

I also wanted to offer other related insights as a means to look at *why* these areas were hotspots for gangs, rather than just pointing them out.
Some of these ideas were:

- What type of deprivation in areas is more likely to be correlated with gang crime?
- Does Taser use by police help fight weapon crime and violent offences?
- Does a 'war on drugs' by police actually cut drug deaths and hospitalisations?
- Are the number of legal firearms linked to the number of illegal firearm offences?

Before we dig into the details of the project, a brief note on the final outcome itself:

While offering the ability to look at gang crime, the project quickly expanded and was finally realised as a wide-reaching dashboard allowing parameter-driven visualisation of crime across England and Wales.

3 Data Sources

England/Wales Police Force Data
data.police.uk

- Detailed crime data
- Geographic force boundaries

UK Government Data
gov.uk / ons.gov.uk

- LSOA population data
- Deprivation indices (England)
- Firearm and shotgun ownership
- Firearm offences
- Police use of tasers
- Police drug seizure data
- Drug misuse data
- Drug deaths data

UK Government GeoPortal Data
geoportal.statistics.gov.uk

- Area comparison table
- 2011 LSOA boundaries

UK Parliament Data
parliament.uk

- Weapon crime statistics

NHS Data
nhs.uk

- NHS drug abuse data

Government of Wales Data
statswales.gov.wales

- Deprivation indices (Wales)

4 Methodology

This project spanned across much of the data pipeline and thus can be broken down and discussed in separate parts. These three sections, titled **Engineering**, **Modelling** and **Analysis** will comprise the ETL process as well as visualisation of the modelled data.

- The **Engineering** section will cover the *extraction* and *transformation* of data, with references to the SQL code used for detailed further review; depicted in the appendices.
- The **Modelling** section, as the name implies, will frame the *loading* of data into a dimensional model as preparation for effective, efficient utilisation.
- Thirdly, the **Analysis** section will look at the design decisions taken in creating the final dashboard visualisation and offer suggestions on its use as a source of information and analysis.

4.1 Engineering

All data sources of extraction are listed in section 3, with individual specific links given in the references section at the end of this documentation.

Appendix section 8.1 and section 8.2 show the SQL and Alteryx cleansing and transformation of this data respectively. These appendices are commented throughout as form of explanation.

4.2 Modelling

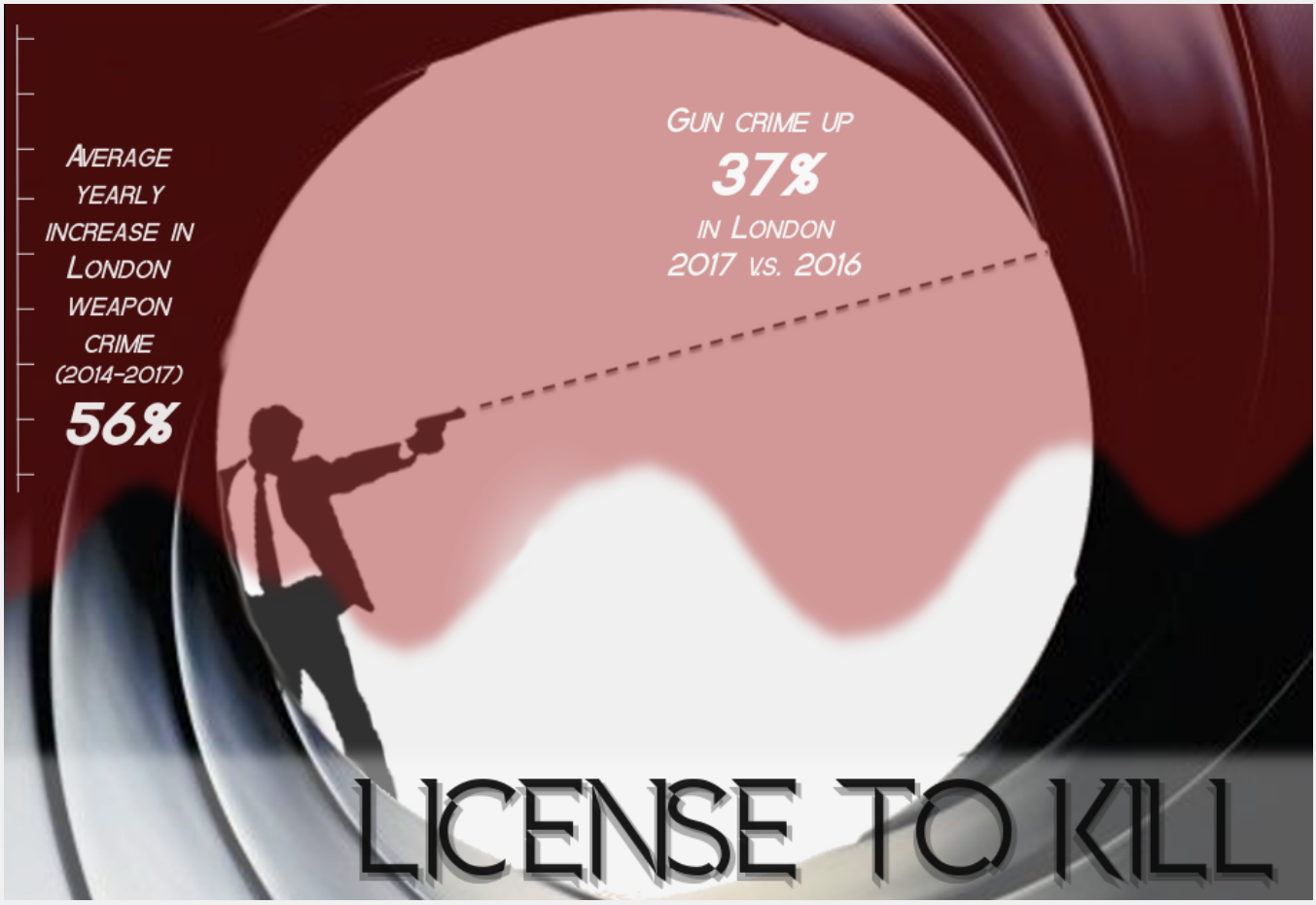
Appendix section 8.3 shows the dimensional modelling of the data in SQL and section 8.4 shows a representation of this model visually. These appendices are commented throughout or are self explanatory.

4.3 Analysis

The final product of this project is a crime dashboard that allows the user to explore many facets of crime in England and Wales and draw conclusions based on their specific requirements. The dashboard can be accessed here.

The dashboard contains detailed dynamic instructions in order to assist in data exploration and analysis.

On the following page is an example of portraying analyses made using the dashboard as an infographic.



5 Assumptions and Limitations

Most assumptions are discussed throughout the code as and when they were made during the course of the project, via comments shown in the appendices.

In terms of limitations, main points to note include:

- Disregarding crime types that are not gang-related.
- Limiting the scope in terms of time period to homogenise many sources of data.
- Large amount of data slowing down visualisation.
- Exploratory tool rather than final analysis.

6 Appendices

6.1 SQL Cleanse

```
USE CrimeProject
GO

/* Cleansing to-dos

        Spatial Data Tables
                LSOAs and Police Force Areas
                QGIS to convert .shp files on British system to .csv in World co-ordinates
                        Import to SQL
                SQL Cleansing
                        Convert Well-Known-Text into SQL geog polygons
                        Remove unwanted columns (e.g. Welsh variants of area names)
                                Also rename columns to fit conventions applied in other tables
                We can then use these to insert a correct Force Area and LSOA column into police. tables

        AreaCompare table
                Just keep LSOA and Region columns as this is what it's needed for
                Adding in relationship between force area and regions obtained from FirearmDealersForceArea

        PoliceCrimeData Tables
                StopSearch
                        Unnecessary columns:        Policing operation,
                                                                Ethnicities (purposely avoiding this for my project),
                                                                Outcome linked to object of search,
                                                                Removal of more than just outer clothing
                        Cleansing:        Create Geom points from Long/Lat
                                                Make [Date] into proper datetime
                                                Alter [Age Range] values to mimic style from elsewhere
                                                Group up some of the [Object of search] for easier comparison
                Street
                        Unnecessary columns:        [Reported by] and [Falls within] are always equal so just use either AS 'Area'
                                                                Location (just describes which road, we already have precise co-ords)
                                                                Context
                        Cleansing:        Fix shifted columns (thanks to erroneous commas in [Location])
                                                        In a CTE, shift these columns back in the right cases (WHERE [LSOA code] like '%"')
                                                        And then build the clean table from that fixed CTE
```

```
                                                    Remove unwanted [Crime Type] that don't relate to weapons/drugs/serious gangs:
                                                            Anti-social behaviour, Bicycle theft, Burglary, Criminal damage and arson,
                                                            Other crime, Other theft, Public order, Shoplifting, Vehicle crime
                                                            (THIS MAKES THE TABLE SMALL ENOUGH TO MANAGE)
                                                    Create Geom points from Long/Lat

        Spatialisation of PoliceCrimeData Tables
                Using the GeoPoints created for every instance of crime
                        intersect on LSOA and Force polys to create new columns within the Police. tables
                                Street already has Area but with different strings (e.g. 'Cumbria Constabulary' rather than 'Cumbria')
                                        Going to recreate this by intersecting on the GeoPoint
                                Street also already has LSOA with correct data values
                                        Could run some small intersect queries (TOP) to check that my polys line up to this data
                                StopSearch only has long/lat and thus GeoPoints
                                        Need to run intersects to create two new columns
                                                One for LSOA and one for ForceArea
                ((Might need to batch these as STIntersects is rather CPU intensive)))
                        (((Once this is done, however, there is no more need for Geog operations and we will have lovely data to analyse)))
                TURNS OUT... spatial index was the answer, uses B-Trees to index Geog points and Intersect more efficiently


        Firearm/Shotgun Tables [!= FIREARM OFFENCE TABLES]
                These all have summed rows for areas (e.g. England/England and Wales)
                        These should be removed
                Firearm/Shotgun Certificates and FirearmDealers has 2 or 3 Granted/Refused rows.
                        In firearm, these correlate to New Applications/Renewal Applications/Variation of Certificate
                        In shotgun, these correlate to New Applications/Renewal Applications
                        In Dealers, these correlate to Previously unregistered/Previously registered
                        Columns should be renamed to reflect this
                FirearmsGenderForceArea has 2 sets of triple duplicated rows
                        Total/Females/Males/Gender not known = Firearm Certs/Shotgun Certs/Firearm and-or shotgun certs
                        Total/[Age Brackets] = Firearm Certs/Shotgun Certs/Firearm and-or shotgun certs
                        Columns should be renamed to reflect this

        Firearm Offence Tables
                These tables are absolutely cancerous, deep chemo required!!!
                        Should be relatively obvious by looking at table / Excel docs

        Bladed Offence Tables
                Not quite as bad as Firearm but still need some cleaning, should be obvious by looking at tables
                Years on force area table:
                        08/09, 09/10, 10/11, 11/12, 12/13, 13/14, 14/15, 15/16, 16/17

        Taser Tables
                Imported as 09-13, then individual table for each year, need to collate into one big table with date column
```

```
82                      In 14 only final 3 columns are necessary (others are breaking down by how
    ↪ taser was used, unneccesary)
83                      In 15 and 16, data is broken up by discharge and non-discharge, use both
    ↪ TOTAL columns
84                      Also, as with most force area groupings there are "total rows" for wider
    ↪ areas so watch out!
85              NOTE: these numbers are incidents involving a taser (not necessarily discharge)
86              NOTE: the year in the table name is ending year of data (may be more than 1 years
    ↪ worth)
87
88          DrugDeathByArea Table
89              Alteryx, not much to do here
90
91          DrugSeizures Tables
92              Snapshot table is based on just year 2016/17
93                  includes a total column for each drug class
94                  also includes final unnamed column which is 'Unknown' (i.e. unknown drug)
95
96          DrugAdmissionsNHS Tables
97              In column groups of 6 for each year
98                  Order: 16/17, 15/16, 14/15, 13/14
99                  Repeated 3 columns, first is the actual admissions for All
    ↪ persons/Male/Female
100                         Then the same per 100,000 population
101
102         DrugSurveyData Table
103             Values are "Proportion of 16 to 59 year olds reporting use of drugs in the specified
    ↪ year"
104                 These are therefore percentages of total participants surveyed
105             Broken down by different drugs
106                 These are currently listed as their own row above the area data so need to
    ↪ fix this
107
108         DeprivationIndices Table
109             Ignore 'Score' columns unless there is a specific index we really want to explore
110                 then look up the meaning of the score online/in spreadsheet
111             Ranking goes in order of 1 = Most deprived/Worst, i.e. higher is 'better'
112
113         PopulationByLSOA Tables
114             Imported by seperate years (data is from mid-specified year)
115                  Need to insert column for date and collate into single table
116             In the earliest data (11), actual population numbers need to be derived
117                 Data given is 'Area(sq km)' and 'pop per sq km' so this is very easy
118
119         CHECK EVERY ORIGINAL DOCUMENT IN CASE VALUES ARE NOT EXACT
120         e.g. Column title = 'Deaths', but it's actually deaths per 100,000 or something
121
122         Check everything for (unwanted) duplicates
123         ALSO check everything for area total rows, such as the different Yorkshires grouped as
    ↪ Yorkshire and the Humber
124             These totals should be removed
125
126         Sort into Schemas
127
128         THERE ARE 43 FORCE AREAS SO ANYTHING WITH THESE SHOULD HAVE THIS MANY ROWS (at least by year
    ↪ or whatever)
129
130         Add functionality to repeatedly run this sql file by using if statements to drop tables
    ↪ before re-making them
```

```
131
132 */
133
134 /* Creating some extra schema to store uncleansed tables, temp tables and also any trash */
135 Create schema [Dirty]
136 GO
137 Create schema [Temp]
138 GO
139 Create schema [Trash]
140 GO
141
142 /*---------------------------
143    Spatial tables cleanse
144 ---------------------------*/
145 CREATE schema [Geo]
146 GO
147 -- Force Area polys (43 Force Areas)
148 ;with makevalids -- cte to build the polys from text and MakeValid()
149 as (
150         SELECT
151                 CASE
152                         WHEN WKT like '"MULTI%' THEN
153                                 (geography::STMPolyFromText(REPLACE(WKT,'"',''),
                                    ↪  4326).MakeValid())
154                         ELSE        (geography::STPolyFromText(REPLACE(WKT,'"',''),
                                    ↪  4326).MakeValid())
155                 END as poly,
156                 *
157         FROM ForceAreaFinal
158         )
159 select
160         -- Selecting only desired columns and renaming to be consistent with other tables
161         pfa16cd AS [Area code]
162         ,pfa16nm AS [Area name]
163         ,CASE -- This is checking to see if we need to ReorientObject(), for some reason some of the
            ↪  polys are broken and some aren't
164                 WHEN poly.EnvelopeAngle() < 180 THEN poly
165                 ELSE poly.ReorientObject()
166          END AS [Geo poly]
167          /* Going to do area in Alteryx in the interests of speed
168         ,round(([Geo poly].STArea())/1000000,6) AS [Area (sq.km)] -- Calculating Force area (square
    ↪  kilometres) */
169 INTO Geo.ForceArea -- Put into a table for comparison down the line
170 FROM makevalids
171 -- LSOA polys (34,753 LSOAs)
172 ;with makevalids2 -- cte to build the polys from text and MakeValid()
173 as (
174         SELECT
175                 (geography::STMPolyFromText(REPLACE(WKT,'"',''), 4326).MakeValid()) as poly,
176                 *
177         FROM dirty.LSOAFinal
178         )
179 ,deletedupes -- have to run a cte with ROW_NUMBER since DISTINCT isn't allowed on Geog
180 as (
181         select
182                 -- Selecting only desired columns and renaming to be consistent with other tables
183                  lsoa11cd AS [LSOA code]
184                 ,lsoa11nm AS [LSOA name]
185                 ,CASE -- This is checking to see if we need to ReorientObject()
```

```
186                         -- for some reason some of the polys are broken and some aren't
187                         WHEN poly.EnvelopeAngle() < 180 THEN poly
188                         ELSE poly.ReorientObject()
189                     END AS [Geo poly]
190                     ,ROW_NUMBER() over (partition by lsoa11cd order by (SELECT 1)) As rowN
191             FROM makevalids2
192     )
193     SELECT
194             [LSOA code]
195             ,[LSOA name]
196             ,[Geo poly]
197             /* Going to do area in Alteryx in the interests of speed
198             ,round(([Geo poly].STArea())/1000000,6) AS [Area (sq.km)] -- Calculating LSOA area (square
    ↪  kilometres) */
199     INTO Geo.LSOA -- Putting into table for comparisons down the line
200     FROM deletedupes
201     WHERE rown=1
202     ALTER SCHEMA Dirty TRANSFER dbo.ForceAreaFinal
203     ALTER SCHEMA Dirty TRANSFER dbo.LSOAFinal
204     -- Joining Alteryx calculated areas into Geo tables
205     ALTER SCHEMA temp TRANSFER Geo.ForceArea
206     ALTER SCHEMA temp TRANSFER Geo.LSOA
207     SELECT
208              fa.id
209             ,fa.[Area name]
210             ,fa.[Area code]
211             ,round(gfa.[Area (sqkm)],2) AS [Area (sqkm)]-- Taking area (square kilometres) and inserting
212             ,fa.[Geo poly]
213     INTO geo.ForceArea
214     FROM temp.ForceArea fa
215     JOIN temp.GeoForceArea gfa
216             on fa.id = gfa.id
217     SELECT
218              l.id
219             ,l.[LSOA name]
220             ,l.[lsoa code]
221             ,round(gl.[Area (sqkm)],6) AS [Area (sqkm)]-- Taking area (square kilometres) and inserting
222             ,l.[Geo poly]
223     INTO geo.LSOA
224     FROM temp.LSOA l
225     JOIN temp.GeoLSOA gl
226             on l.id = gl.id
227
228
229     /*---------------------------------
230         PoliceCrimeData tables cleanse
231     ---------------------------------*/
232     CREATE SCHEMA [Police]
233     GO
234     -- Stop and Search
235     SELECT
236              CAST([Longitude] as float) AS [Longitude]
237             ,CAST([Latitude] as float) AS [Latitude]
238             ,geography::STPointFromText('POINT(' + cast([Longitude] as varchar)
239                     + ' ' + cast([Latitude] as varchar) + ')', 4326) AS [Geo point] -- Creating
                    ↪  Geo-points from Long/Lat
240             ,[Type] as [Search type]
241             ,Cast(Cast([Date]as datetime2) as smalldatetime) AS [DateTimestamp]
242             ,CAST([Part of a policing operation] as bit) AS [Police operation]
```

```
243         ,Gender
244         ,CASE
245                 WHEN [Age range] = 'under 10' THEN '0-10'
246                 WHEN [Age range] = 'over 34' THEN '34+'
247                 ELSE [Age range]
248         END AS [Age range]
249         ,legislation
250         ,CASE
251         WHEN [Object of search] IN ('Article for use in theft','Stolen goods')
252             THEN 'Theft'
253         WHEN [Object of search] IN ('Controlled drugs','Psychoactive substances')
254             THEN 'Drugs'
255         WHEN [Object of search] IN ('Crossbows','Offensive weapons','Anything to threaten or harm
        ↪  anyone','Firearms')
256             THEN 'Weapons'
257         WHEN [Object of search] IN ('Articles for use in criminal damage')
258             THEN 'Criminal damage'
259         ELSE [Object of search]
260      END AS [Object of search]
261         ,Outcome
262 INTO Police.CrimeDataStopSearch
263 FROM dbo.PoliceCrimeDataStopSearch
264
265 -- Street
266 ;with FixShift AS ( -- creating a cte to fix shifted columns and only return desired columns
267 SELECT
268         [Crime ID]
269         ,CAST(left([Month],4) as int) as [Year]
270         ,CAST(right([Month],2) as int) as [Month]
271         ,[Falls within] as [Area]
272         ,CAST([Longitude] as float) AS [Longitude]
273         ,CAST([Latitude] as float) AS [Latitude]
274         ,CASE
275                 WHEN [LSOA code] like '%"'
276                         THEN [LSOA name]
277                 ELSE [LSOA code]
278          END as [LSOA code]
279         ,CASE
280                 WHEN [LSOA code] like '%"'
281                         THEN [Crime type]
282                 ELSE [LSOA name]
283          END as [LSOA name]
284         ,CASE
285                 WHEN [LSOA code] like '%"'
286                         THEN [Last outcome category]
287                 ELSE [Crime type]
288          END as [Crime type]
289         ,CASE
290                 WHEN [LSOA code] like '%"'
291                         THEN [Context]
292                 ELSE [Last outcome category]
293          END as [Last outcome category]
294 FROM PoliceCrimeDataStreet
295 )
296 SELECT -- Selecting from our fixed cte only rows with desired [Crime type]
297         *
298 -- Inserting this into a holding table so I don't have to keep running this 4 minute query
299 INTO temp.PoliceCrimeStreet
300 FROM FixShift
```

```
WHERE 1=1
AND [Crime type] != 'Anti-social behaviour' AND [Crime type] != 'Bicycle theft'
AND [Crime type] != 'Burglary' AND [Crime type] != 'Criminal damage and arson'
AND [Crime type] != 'Other crime' AND [Crime type] != 'Other theft'
AND [Crime type] != 'Public order' AND [Crime type] != 'Shoplifting'
AND [Crime type] != 'Vehicle crime' AND [Crime type] IS NOT NULL
-- Final cleansing of Street data (creating geom points and removing crimes with NULL locations)
SELECT
        *
        ,geography::STPointFromText('POINT(' + cast([Longitude] as varchar)
                + ' ' + cast([Latitude] as varchar) + ')', 4326) AS [Geo point]
INTO Police.CrimeDataStreet
FROM temp.PoliceCrimeDataStreet
WHERE Longitude IS NOT NULL

/* THIS IS ALMOST 100% ALREADY CONTAINED WITHIN 'Street' TABLE SO UNNECESSARY
-- Outcomes
;with FixShift2 AS ( -- creating a cte to fix shifted columns and only return desired columns
SELECT
        [Crime ID]
        ,CAST(left([Month],4) as int) as [Year]
        ,CAST(right([Month],2) as int) as [Month]
        ,[Falls within] as [Area]
        ,CAST([Longitude] as float) AS [Longitude]
        ,CAST([Latitude] as float) AS [Latitude]
        ,CASE
                WHEN [LSOA code] like '%"'
                        THEN [LSOA name]
                ELSE [LSOA code]
         END as [LSOA code]
        ,CASE
                WHEN [LSOA code] like '%"'
                        THEN SUBSTRING([Outcome type],1,charindex(',',[Outcome type])-1)
                ELSE [LSOA name]
         END as [LSOA name]
        ,CASE
                WHEN [LSOA code] like '%"'
                        THEN SUBSTRING([Outcome type],charindex(',',[Outcome type])+1,LEN([Outcome
    ↪  type]))
                ELSE [Outcome type]
         END as [Outcome type]
FROM PoliceCrimeDataOutcomes
)
SELECT -- Selecting from our fixed cte only rows that line up to our Street crime table on ID
        f.*
-- Inserting this into a holding table so I don't have to keep running this query
INTO temp.PoliceCrimeOutcomes
FROM FixShift2 f
JOIN Police.CrimeDataStreet pc
        on f.[Crime ID] = pc.[Crime ID]
*/
-- Tidying up tables appropriately
ALTER SCHEMA Dirty TRANSFER dbo.PoliceCrimeDataOutcomes
ALTER SCHEMA Dirty TRANSFER dbo.PoliceCrimeDataStopSearch
ALTER SCHEMA Dirty TRANSFER dbo.PoliceCrimeDataStreet
ALTER SCHEMA Trash TRANSFER temp.PoliceCrimeOutcomes
ALTER SCHEMA Trash TRANSFER temp.PoliceCrimeStreet


```

```
/*-------------------------------------------
   Spatialisation of PoliceCrimeData Tables
-------------------------------------------*/
ALTER SCHEMA Temp TRANSFER police.CrimeDataStopSearch
ALTER SCHEMA Temp TRANSFER police.CrimeDataStreet
-- Needs Primary Key in order to create spatial index
ALTER TABLE geo.ForceArea
        Add id int identity primary key
ALTER TABLE geo.LSOA
        Add id int identity primary key
ALTER TABLE Temp.CrimeDataStopSearch
        Add id int identity primary key
ALTER TABLE Temp.CrimeDataStreet
        Add id int identity primary key
-- Creating spatial index(s) to speed up STIntersects (B-Trees baby!)
CREATE SPATIAL INDEX SIndx_CrimeDataStopSearch_GeoPoint
   ON Temp.CrimeDataStopSearch([Geo point])
CREATE SPATIAL INDEX SIndx_CrimeDataStreet_GeoPoint
   ON Temp.CrimeDataStreet([Geo point])
--CREATE SPATIAL INDEX SIndx_GeoForceArea_GeoPoly
--   ON Geo.ForceArea([Geo poly])                        -- SMALL NUMBER OF ROWS: unnecessary to
↪  have spatial index
CREATE SPATIAL INDEX SIndx_GeoLSOA_GeoPoly
   ON Geo.LSOA([Geo poly])
-- STOP and SEARCH
        -- Intersecting with Force Areas
        SELECT
                 f.[Area name]
                ,f.[Area code]
                ,c.[Search type]
                ,c.DateTimestamp
                ,c.[Police operation]
                ,c.Gender
                ,c.[Age range]
                ,c.legislation
                ,c.[Object of search]
                ,c.Outcome
                ,c.[Geo point] -- Need this for the next select where we intersect with LSOA
        INTO Temp.MatchedStopSearch -- Putting into temp so we can add in LSOA value as well
        FROM Temp.CrimeDataStopSearch c
        WITH (INDEX(SIndx_CrimeDataStopSearch_GeoPoint))
        join geo.ForceArea f
                --WITH (INDEX(SIndx_GeoForceArea_GeoPoly))
                on c.[Geo point].STIntersects(f.[Geo poly]) = 1
        -- Intersecting with LSOA
                -- Creating new spatial index on our StopSearch table with matched force areas
        ALTER TABLE temp.matchedstopsearch        -- Needs new primary key to create spatial index
                Add id int identity primary key
        CREATE SPATIAL INDEX SIndx_MatchedStopSearch_GeoPoint
                ON Temp.MatchedStopSearch([Geo point])
        SELECT
                 c.[Area name]
                ,c.[Area code]
                ,l.[LSOA name]
                ,l.[LSOA code]
                ,c.DateTimestamp
                ,c.[Search type]
                ,c.[Object of search]
                ,c.[Police operation]
```

```
417                     ,c.Gender
418                     ,c.[Age range]
419                     ,c.legislation
420                     ,c.Outcome
421             INTO Police.CrimeDataStopSearch
422             FROM Temp.MatchedStopSearch c
423             WITH (INDEX(SIndx_MatchedStopSearch_GeoPoint))
424             join geo.LSOA l
425                     --WITH (INDEX(SIndx_GeoLSOA_GeoPoly))
426                     on c.[Geo point].STIntersects(l.[Geo poly]) = 1
427     -- STREET data
428             -- Intersecting with Force Areas
429             SELECT
430                     f.[Area name]
431                     ,f.[Area code]
432                     ,c.[LSOA name]
433                     ,c.[LSOA code]
434                     ,c.[Year]
435                     ,c.[Month]
436                     ,c.[Crime type]
437                     ,c.[Last outcome category]
438                     ,c.[Geo point] -- Need this for the next select where we intersect with LSOA
439             INTO Temp.MatchedStreet -- Putting into temp so we can add in LSOA value as well
440             FROM Temp.CrimeDataStreet c
441             WITH (INDEX(SIndx_CrimeDataStreet_GeoPoint))
442             join geo.ForceArea f
443                     --WITH (INDEX(SIndx_GeoForceArea_GeoPoly))
444                     on c.[Geo point].STIntersects(f.[Geo poly]) = 1
445             -- Intersecting with LSOA
446                     -- Creating new spatial index on our StopSearch table with matched force areas
447             /*
448             ALTER TABLE temp.matchedstreet        -- Needs new primary key to create spatial index
449                     Add id int identity primary key
450             CREATE SPATIAL INDEX SIndx_MatchedStreet_GeoPoint
451                     ON Temp.MatchedStreet([Geo point])
452             SELECT
453                      c.[Area name]
454                     ,c.[Area code]
455                     ,l.[LSOA name]
456                     ,l.[LSOA code]
457                     ,c.DateTimestamp
458                     ,c.[Search type]
459                     ,c.[Object of search]
460                     ,c.[Police operation]
461                     ,c.Gender
462                     ,c.[Age range]
463                     ,c.legislation
464                     ,c.Outcome
465             INTO Police.CrimeDataStopSearch
466             FROM Temp.MatchedStopSearch c
467             WITH (INDEX(SIndx_MatchedStopSearch_GeoPoint))
468             join geo.LSOA l
469                     --WITH (INDEX(SIndx_GeoLSOA_GeoPoly))
470                     on c.[Geo point].STIntersects(l.[Geo poly]) = 1
471             */
472             -- Putting into correct schema
473             ALTER SCHEMA Police TRANSFER temp.matchedstreet
474
475
```

```
/*-----------------------------
   AreaCompare table cleanse
-----------------------------*/
-- Creating a comparison table for LSOA to Force Area to Region
        -- This should form the basis of the DimGeo table
SELECT DISTINCT
         ac.LSOA11NM as [LSOA name]
        ,ac.LSOA11CD as [LSOA code]
        ,b.[Force area] as [Force area]
        ,c.[Area code] as [Force area code]
        ,ac.RGN11NM as [Region name]
        ,ac.RGN11CD as [Region code]
INTO temp.AreaCompare
FROM dirty.AreaCompare ac
join dirty.FirearmDealersForceArea f
        on ac.RGN11NM = f.Region
join police.MatchedStreet c
        on ac.LSOA11NM = c.[LSOA name]
join Drug.SeizuresForceArea b
        on b.[Area code] = c.[Area code]
WHERE ac.RGN11NM != 'Scotland'
AND f.[Police force area] not like '*%'

;with dupelsoadelete -- Need to delete some duplicate LSOAs where they fell into multiple areas
as (
select
        *,
        ROW_NUMBER() over (partition by [LSOA name] order by (SELECT 1)) AS rown
FROM Temp.AreaCompare
)
select
         [LSOA name]
        ,[LSOA code]
        ,[Force area]
        ,[Force area code]
        ,[Region name]
        ,[Region code]
INTO Geo.AreaCompare
FROM dupelsoadelete
WHERE rown=1 -- We end up losing 3 LSOAs where no crime occurred

ALTER SCHEMA dirty TRANSFER dbo.AreaCompare


/*-----------------------------------------------------------------
   Firearm/Shotgun tables cleanse [NOT 'FirearmOffence' tables]
-----------------------------------------------------------------*/
Create schema [Weapons]
Go
Select
        -- Selecting only desired columns
         year('20' + right([Year],2)) AS [Year] -- Fixing '08/09' format to singular year
        ,[Region]
        ,[Police force area]
        -- Renaming these columns to correctly differentiate
        ,Cast([Granted] as int) AS [New applications granted]
        ,Cast([Refused] as int) AS [New applications refused]
        ,Cast([Granted1] as int) AS [Renewal applications granted]
        ,Cast([Refused1] as int) AS [Renewal applications refused]
```

```
535          ,Cast([Granted2] as int) AS [Variation certificate granted]
536          ,Cast([Refused2] as int) AS [Variation certificate refused]
537          ,Cast([Revocations] as int) AS [Revocations]
538          ,Cast([Firearm certificates on issue as at 31 March] as int) AS [Total on issue (31/03)]
539          ,Cast([Firearms covered by certificates on issue as at 31 March] as int) AS [Total firearms
             ↪  (31/03)]
540          ,Cast([Firearms per 100,000 people as at 31 March] as float) AS [Firearms per 100K pop
             ↪  (31/03)]
541  INTO [Guns].FirearmCertificatesForceArea
542  FROM dbo.FirearmCertificatesForceArea
543  WHERE [Police force area] not like '*%' -- Removing rows that are area totals, fortunately prefaced
     ↪  with an asterisk '*'
544  Select
545          -- Selecting only desired columns
546          year('20' + right([Year],2)) AS [Year] -- Fixing '08/09' format to singular year
547          ,[Region]
548          ,[Police force area]
549          -- Renaming these columns to correctly differentiate
550          ,Cast([Granted] as int) AS [New applications granted]
551          ,Cast([Refused] as int) AS [New applications refused]
552          ,Cast([Granted1] as int) AS [Renewal applications granted]
553          ,Cast([Refused1] as int) AS [Renewal applications refused]
554          ,Cast([Revocations] as int) AS [Revocations]
555          ,Cast([Shotgun certificates on issue as at 31 March] as int) AS [Total on issue (31/03)]
556          ,Cast([Shotguns covered by certificates in force as at 31 March] as int) AS [Total shotguns
             ↪  (31/03)]
557          ,Cast([Shotguns per 100,000 people as at 31 March] as float) AS [Shotguns per 100K pop
             ↪  (31/03)]
558  INTO [Guns].ShotgunCertificatesForceArea
559  FROM dbo.ShotgunCertificatesForceArea
560  WHERE [Police force area] not like '*%' -- Removing rows that are area totals, fortunately prefaced
     ↪  with an '*' asterisk
561  Select
562          -- Selecting only desired columns
563          year('20' + right([Year],2)) AS [Year] -- Fixing '08/09' format to singular year
564          ,[Region]
565          ,[Police force area]
566          -- Renaming these columns to correctly differentiate
567          ,Cast([Granted] as int) AS [New license granted]
568          ,Cast([Refused] as int) AS [New license refused]
569          ,Cast([Granted1] as int) AS [Renewal license granted]
570          ,Cast([Refused1] as int) AS [Renewal license refused]
571          ,Cast([Dealers removed from register] as int) AS [Dealers removed]
572          ,Cast([Dealers registered as at 31 March] as int) AS [Total dealers (31/03)]
573  INTO [Guns].FirearmDealersForceArea
574  FROM dbo.FirearmDealersForceArea
575  WHERE [Police force area] not like '*%' -- Removing rows that are area totals, fortunately prefaced
     ↪  with an asterisk '*'
576  Select
577          -- Selecting only desired columns (splitting table into 2, this one for gender)
578          year('20' + right([Year],2)) AS [Year] -- Fixing '08/09' format to singular year
579          ,[Region]
580          ,[Police force area]
581          -- Renaming these columns to correctly differentiate
582          ,Cast([Females] as int) AS [Female firearm certs]
583          ,Cast([Males] as int) AS [Male firearm certs]
584          ,Cast([Gender not _known] as int) AS [GenderNA firearm certs]
585          ,Cast([Females1] as int) AS [Female shotgun certs]
586          ,Cast([Males1] as int) AS [Male shotgun certs]
```

```
587         ,Cast([Gender not known] as int) AS [GenderNA shotgun certs]
588 INTO [Guns].GunsGenderForceArea
589 FROM dbo.FirearmsGenderForceArea
590 WHERE [Police force area] not like '*%' -- Removing rows that are area totals, fortunately prefaced
    ↪  with an asterisk '*'
591 Select
592         -- Selecting only desired columns (splitting table into 2, this one for age brackets)
593          year('20' + right([Year],2)) AS [Year] -- Fixing '08/09' format to singular year
594         ,[Region]
595         ,[Police force area]
596         -- Renaming these columns to correctly differentiate
597         ,Cast([14 to 17] as int) AS [Firearm certs 14-17]
598         ,Cast([18 to 34] as int) AS [Firearm certs 18-34]
599         ,Cast([35 to 49] as int) AS [Firearm certs 35-49]
600         ,Cast([50 to 64] as int) AS [Firearm certs 50-64]
601         ,Cast([65 and _over] as int) AS [Firearm certs 65+]
602         ,Cast([13 and _under1] as int) AS [Shotgun certs 0-13]
603         ,Cast([14 to 171] as int) AS [Shotgun certs 14-17]
604         ,Cast([18 to 341] as int) AS [Shotgun certs 18-34]
605         ,Cast([35 to 491] as int) AS [Shotgun certs 35-49]
606         ,Cast([50 to 641] as int) AS [Shotgun certs 50-64]
607         ,Cast([65 and _over1] as int) AS [Shotgun certs 65+]
608 INTO [Guns].GunsAgeForceArea
609 FROM dbo.FirearmsGenderForceArea
610 WHERE [Police force area] not like '*%' -- Removing rows that are area totals, fortunately prefaced
    ↪  with an asterisk '*'
611 -- Moving original tables into dirty schema
612 ALTER SCHEMA dirty TRANSFER dbo.FirearmCertificatesForceArea
613 ALTER SCHEMA dirty TRANSFER dbo.ShotgunCertificatesForceArea
614 ALTER SCHEMA dirty TRANSFER dbo.FirearmDealersForceArea
615 ALTER SCHEMA dirty TRANSFER dbo.FirearmsGenderForceArea
616
617
618 /*--------------------------------
619    FirearmOffence tables cleanse
620 --------------------------------*/
621 -- OFFENCE BY SEVERITY OF INJURY
622         -- Need to clean lots of unwanted data and also transpose table, so going to pivot/unpivot
623 ;with GunOffenceByInjury
624 AS (
625 SELECT TOP 10 -- TOP 10 so that we only look at firearm offences, not airgun
626         -- Renaming date columns to something more sensible
627         Injuries, [Apr '02 to Mar '03] AS '2003', [Apr '03 to Mar '04] AS '2004', [Apr '04 to Mar
        ↪  '05] AS '2005',
628         [Apr '05 to Mar '06] AS '2006', [Apr '06 to Mar '07] AS '2007', [Apr '07 to Mar '08] AS
        ↪  '2008',
629         [Apr '08 to Mar '09] AS '2009', [Apr '09 to Mar '10] AS '2010', [Apr '10 to Mar '113] AS
        ↪  '2011',
630         [Apr '11 to Mar '12] AS '2012', [Apr '12 to Mar '13] AS '2013', [Apr '13 to Mar '14] AS
        ↪  '2014',
631         [Apr '14 to Mar '15] AS '2015', [Apr '15 to Mar '16] AS '2016', [Apr '16 to Mar '17] AS
        ↪  '2017'
632 FROM dbo.FirearmOffenceByInjury
633 )
634 SELECT
635         -- Better names and casting to suitable types
636          year(Cast([Date (March)] AS date)) AS [Date (March)]
637         ,Cast([Fatal injury4] as int) AS [Fatal]
638         ,Cast([Serious injury5] as int) AS [Serious]
```

```
639         ,Cast([Slight injury] as int) AS [Lesser]
640         ,Cast([No injury] as int) AS [No Injury]
641 INTO Weapons.FirearmOffenceByInjury
642 FROM
643 (SELECT
644         [Injuries],value,[Date (March)]
645         FROM GunOffenceByInjury
646         unpivot (
647                 value for [Date (March)] in
648                 ([2003],[2004],[2005],[2006],[2007],[2008],[2009],[2010],[2011],[2012],[2013],
649                 [2014],[2015],[2016],[2017])
650         ) unpiv
651         ) AS src
652 PIVOT (
653         sum(value)
654         FOR Injuries IN ([Fatal injury4], [Serious injury5], [Slight injury], [No injury])
655         ) AS PivotTable;
656 GO
657 -- OFFENCE BY LOCATION TYPE (ROBBERIES)
658         -- e.g. Post Office/Public highway
659         -- Need to clean some unwanted data and also transpose table, so going to pivot/unpivot
660 ;with GunOffenceByLoc
661 AS (
662 SELECT
663         -- Renaming date columns to something more sensible
664         [Location of offence]
665         ,[Apr '02 to Mar '03] AS '2003', [Apr '03 to Mar '04] AS '2004', [Apr '04 to Mar '05] AS
        ↪  '2005',
666         [Apr '05 to Mar '06] AS '2006', [Apr '06 to Mar '07] AS '2007', [Apr '07 to Mar '08] AS
        ↪  '2008',
667         [Apr '08 to Mar '09] AS '2009', [Apr '09 to Mar '10] AS '2010', [Apr '10 to Mar '11] AS
        ↪  '2011',
668         [Apr '11 to Mar '12] AS '2012', [Apr '12 to Mar '13] AS '2013', [Apr '13 to Mar '14] AS
        ↪  '2014',
669         [Apr '14 to Mar '15] AS '2015', [Apr '15 to Mar '16] AS '2016', [Apr '16 to Mar '17] AS
        ↪  '2017'
670 FROM dbo.FirearmOffenceByLocationType
671 )
672 SELECT
673         -- Better names and casting to suitable types
674          year(Cast([Date (March)] AS date)) AS [Date (March)]
675         ,Cast([Shop, stall etc.] as int) AS [Shop]
676         ,Cast([Garage, service station ] as int) AS [Garage]
677         ,Cast([Post Office] as int) AS [Post office]
678         ,Cast([Bank] as int) + Cast([Building society] as int) AS [Bank]
679         ,Cast([Residential2] as int) AS [Residential]
680         ,Cast([Public highway] as int) AS [Road]
681         ,Cast([Other premises or open space] as int) AS [Other]
682 INTO Weapons.FirearmOffenceByLocationType
683 FROM
684 (SELECT
685         [Location of offence],value,[Date (March)]
686         FROM GunOffenceByLoc
687         unpivot (
688                 value for [Date (March)] in
689                 ([2003],[2004],[2005],[2006],[2007],[2008],[2009],[2010],[2011],[2012],[2013],
690                 [2014],[2015],[2016],[2017])
691         ) unpiv
692         ) AS src
```

```
693 PIVOT (
694         sum(value)
695         FOR [Location of offence] IN ([Shop, stall etc.], [Garage, service station ], [Post Office],
            ↪  [Bank]
696                                       ,[Building society], [Residential2], [Public highway],
                                          ↪  [Other premises or open space])
697         ) AS PivotTable;
698 GO
699 -- OFFENCE BY OFFENCE
700         -- e.g. Homicide/Robbery/Possession
701         -- Need to clean some unwanted data and also transpose table, so going to pivot/unpivot
702 ;with GunOffenceByOffence
703 AS (
704 SELECT TOP 17 -- So we only get firearm offences and not also air-weapons
705         -- Renaming date columns to something more sensible
706         [Offence type]
707         ,[Apr '03 to Mar '04] AS '2004', [Apr '04 to Mar '05] AS '2005',
708         [Apr '05 to Mar '062] AS '2006', [Apr '06 to Mar '07] AS '2007', [Apr '07 to Mar '08] AS
            ↪  '2008',
709         [Apr '08 to Mar '09] AS '2009', [Apr '09 to Mar '10] AS '2010', [Apr '10 to Mar '11] AS
            ↪  '2011',
710         [Apr '11 to Mar '12] AS '2012', [Apr '12 to Mar '13] AS '2013', [Apr '13 to Mar '14] AS
            ↪  '2014',
711         [Apr '14 to Mar '15] AS '2015', [Apr '15 to Mar '16] AS '2016', [Apr '16 to Mar '17] AS
            ↪  '2017'
712 FROM dbo.FirearmOffenceByOffence
713 )
714 SELECT
715         -- Better names and casting to suitable types
716          year(Cast([Date (March)] AS date)) AS [Date (March)]
717         ,Cast([Homicide3] as int) AS [Homicide]
718         ,COALESCE(Cast([Attempted murder and other most serious violence] as int),
719                                 Cast([Attempted murder and GBH with intent offences4] as int),
720                                 Cast([Attempted murder, assault with intent to cause serious harm
                                    ↪  and endangering life4] as int)
721         ) AS [Attempted murder]
722         ,Cast([Other] as int) AS [Other violence]
723         ,Cast([Robbery] as int) AS [Robbery]
724         ,Cast([Burglary] as int) AS [Burglary]
725         ,Cast([Criminal damage] as int) AS [Criminal damage]
726         ,Cast([Public fear, alarm or distress] as int) AS [Public fear]
727         ,Cast([Possession of weapons] as int) AS [Possession]
728         ,Cast([Other firearm offences] as int) AS [Other]
729 INTO Weapons.FirearmOffenceByOffence
730 FROM
731 (SELECT
732         [Offence type],value,[Date (March)]
733         FROM GunOffenceByOffence
734         unpivot (
735                 value for [Date (March)] in
736                 ([2004],[2005],[2006],[2007],[2008],[2009],[2010],[2011],[2012],[2013],
737                 [2014],[2015],[2016],[2017])
738         ) unpiv
739         ) AS src
740 PIVOT (
741         sum(value)
742         FOR [Offence type] IN ([Homicide3], [Attempted murder and other most serious violence],
            ↪  [Attempted murder and GBH with intent offences4],
```

```
743         [Attempted murder, assault with intent to cause serious harm and endangering life4],[Other],
            ↪ [Robbery],
744         [Burglary], [Criminal damage],[Public fear, alarm or distress],[Possession of
            ↪ weapons],[Other firearm offences])
745         ) AS PivotTable;
746 GO
747 -- FirearmOffenceByWeapon will remain unused, moving to Trash schema
748 ALTER SCHEMA Trash TRANSFER FirearmOffenceByWeapon
749 -- OFFENCE BY FORCE AREA
750         -- Removed final pivot in order to match area convention with other tables
751         -- e.g. Cumbria/West Mercia/Dorset
752         -- Need to clean some unwanted data, remove totals and also transpose table, so going to
            ↪ pivot/unpivot
753 ;with GunOffenceByArea
754 AS (
755 SELECT
756         [Police force area],
757         -- Renaming date columns to something more sensible
758         [Apr '07 to Mar '08] AS '2008',
759         [Apr '08 to Mar '09] AS '2009', [Apr '09 to Mar '10] AS '2010', [Apr '10 to Mar '11] AS
            ↪ '2011',
760         [Apr '11 to Mar '12] AS '2012', [Apr '12 to Mar '13] AS '2013', [Apr '13 to Mar '14] AS
            ↪ '2014',
761         [Apr '14 to Mar '15] AS '2015', [Apr '15 to Mar '16] AS '2016', [Apr '16 to Mar '17] AS
            ↪ '2017'
762 FROM dirty.FirearmOffenceForceArea
763 WHERE 1=1
764         AND [Police force area] not like 'East '
765         AND [Police force area] not like 'South West '
766         AND [Police force area] not like 'South East '
767         AND [Police force area] not like 'North West '
768         AND [Police force area] not like 'North East '
769         AND [Police force area] not like 'Wales'
770         AND [Police force area] not like 'Yorkshire and The Humber '
771         AND [Police force area] not like 'London '
772         AND [Police force area] not like 'East Midlands '
773 )
774 SELECT
775         -- Better names and casting to suitable types
776         year(Cast([Date (March)] AS date)) AS [Date (March)]
777         ,[Police force area] AS [Area]
778         ,MIN([value]) over (partition by [Date (March)], [Police force area]) AS [Offences]
779 INTO Weapons.FirearmOffenceByArea
780 FROM
781 (SELECT
782         [Police force area],value,[Date (March)]
783         FROM GunOffenceByArea
784         unpivot (
785                 value for [Date (March)] in
786                 ([2008],[2009],[2010],[2011],[2012],[2013],
787                 [2014],[2015],[2016],[2017])
788         ) unpiv
789         ) AS src
790 --PIVOT (
791 --        sum(value)
792 --        FOR [Police force area] IN ([Cleveland], [Durham], [Northumbria],
    ↪ [Cheshire],[Cumbria],[Greater Manchester]
793 --        ,[Lancashire],[Merseyside], [Humberside],[North Yorkshire], [South Yorkshire],[West
    ↪ Yorkshire],[Derbyshire]
```

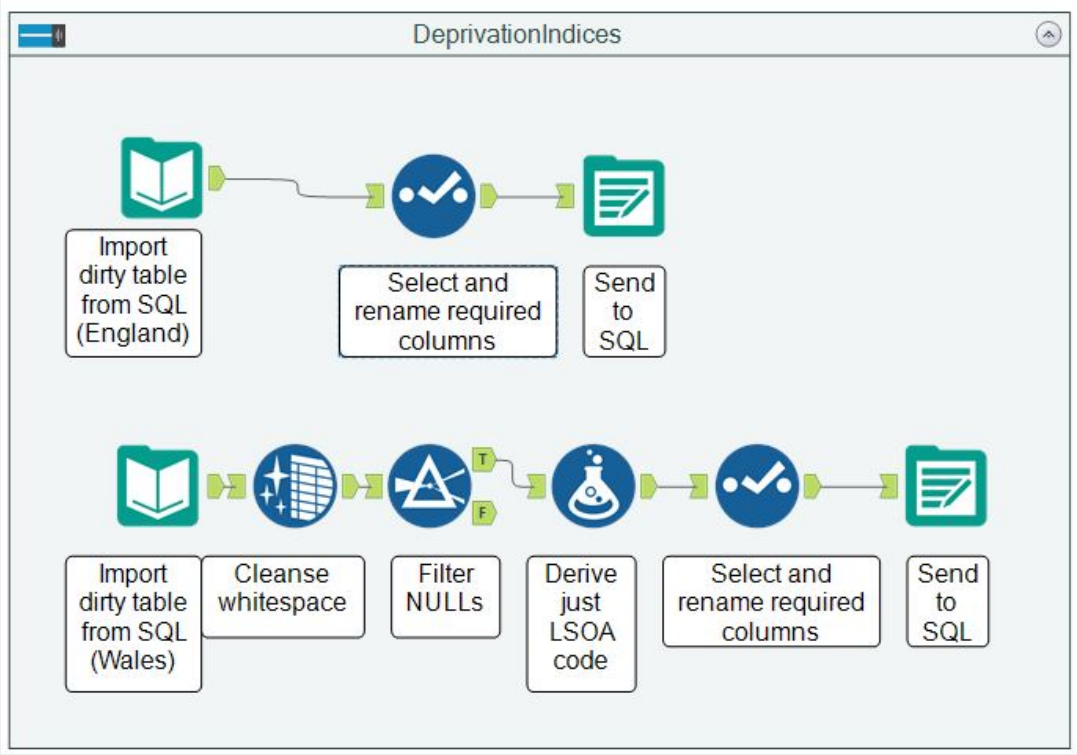
```
794 --        ,[Leicestershire],[Lincolnshire],[Northamptonshire],[Nottinghamshire],[Staffordshire],[Warwickshire],
    ↪ Mercia]
795 --        ,[West Midlands],[Bedfordshire],[Cambridgeshire],[Essex] ,[Hertfordshire], [Norfolk],
    ↪ [Suffolk], [City  of London]
796 --        ,[Metropolitan Police] ,[Hampshire] ,[Kent],[Surrey],[Sussex],[Thames Valley],[Avon and
    ↪ Somerset],[Devon and Cornwall]
797 --        ,[Dorset],[Gloucestershire],[Wiltshire],[Dyfed-Powys],[Gwent],[North Wales],[South Wales])
798 --        ) AS PivotTable;
799 GO
800 -- Removing duplicate West Midland Rows
801 WITH Temp ([Date (March)],Area,Offences,duplicateRecCount)
802 AS
803 (
804 SELECT [Date (March)],Area,Offences,ROW_NUMBER() OVER(PARTITION by [Date (March)],Area,Offences
    ↪ ORDER BY Area)
805 AS duplicateRecCount
806 FROM Weapons.FirearmOffenceByArea
807 )
808 DELETE FROM Temp --Now Delete Duplicate Records
809 WHERE duplicateRecCount > 1
810 -- Moving original tables into dirty schema
811 ALTER SCHEMA dirty TRANSFER dbo.FirearmOffenceByInjury
812 ALTER SCHEMA dirty TRANSFER dbo.FirearmOffenceByLocationType
813 ALTER SCHEMA dirty TRANSFER dbo.FirearmOffenceByOffence
814 ALTER SCHEMA dirty TRANSFER dbo.FirearmOffenceForceArea
815
816
817 /*---------------------------------
818     BladedOffence tables cleanse
819 ---------------------------------*/
820 -- Minor v.s. Adult
821 SELECT
822         -- Case statement to transform quarterly date strings into two columns
823                 -- One for year and one for quarter
824         CASE
825                 WHEN [F1] like 'Q1%' THEN DATEPART(YY,Convert(date,'15-02-'+RIGHT(F1,4),105)) -- Use
                    ↪ Feb 15th for Q1
826                 WHEN [F1] like 'Q2%' THEN DATEPART(YY,Convert(date,'15-05-'+RIGHT(F1,4),105)) -- Use
                    ↪ May 15th for Q2
827                 WHEN [F1] like 'Q3%' THEN DATEPART(YY,Convert(date,'15-08-'+RIGHT(F1,4),105)) -- Use
                    ↪ Aug 15th for Q3
828                 ELSE DATEPART(YY,Convert(date,'15-11-'+RIGHT(F1,4),105)) -- Use Nov 15th for Q4
829         END AS [Year]
830         ,CASE
831                 WHEN [F1] like 'Q1%' THEN DATEPART(QQ,Convert(date,'15-02-'+RIGHT(F1,4),105)) -- Use
                    ↪ Feb 15th for Q1
832                 WHEN [F1] like 'Q2%' THEN DATEPART(QQ,Convert(date,'15-05-'+RIGHT(F1,4),105)) -- Use
                    ↪ May 15th for Q2
833                 WHEN [F1] like 'Q3%' THEN DATEPART(QQ,Convert(date,'15-08-'+RIGHT(F1,4),105)) -- Use
                    ↪ Aug 15th for Q3
834                 ELSE DATEPART(QQ,Convert(date,'15-11-'+RIGHT(F1,4),105)) -- Use Nov 15th for Q4
835         END AS [Quarter]
836         ,CAST([Aged 10 to 17] as int) AS [Minor]
837         ,CAST([Aged 18 and over] as int) AS [Adult]
838 INTO Weapons.BladedOffenceByAge
839 FROM [dbo].[BladedOffenceByAgeOutcomeQuarter]
840 -- Outcomes
841 SELECT
842         -- Case statement to transform quarterly date strings into two columns
```

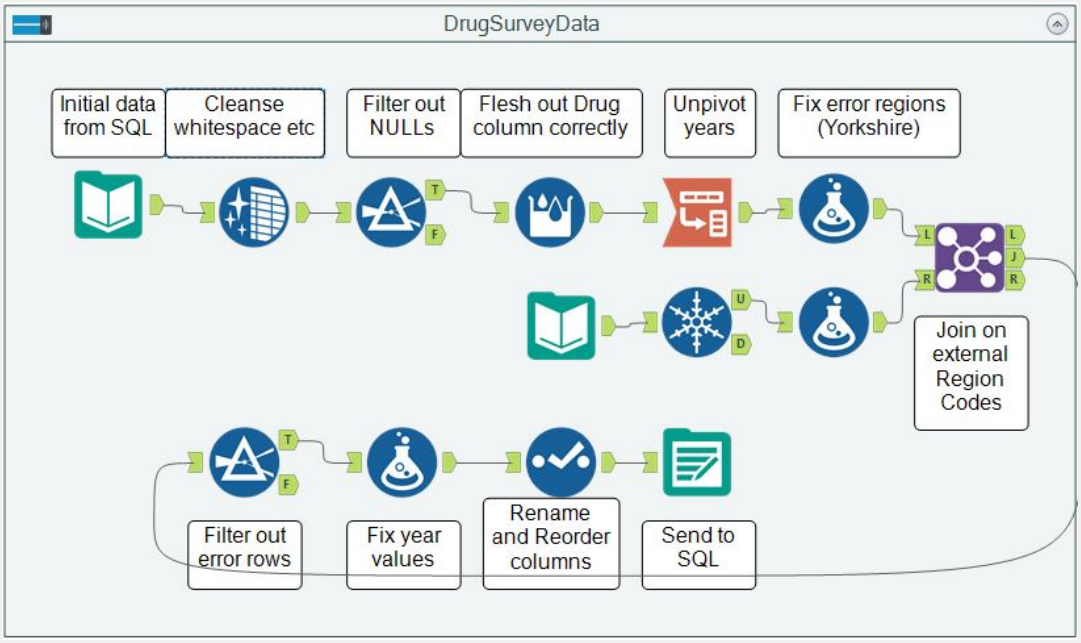
```
843                     -- One for year and one for quarter
844             CASE
845                     WHEN [F1] like 'Q1%' THEN DATEPART(YY,Convert(date,'15-02-'+RIGHT(F1,4),105)) -- Use
                        ↪  Feb 15th for Q1
846                     WHEN [F1] like 'Q2%' THEN DATEPART(YY,Convert(date,'15-05-'+RIGHT(F1,4),105)) -- Use
                        ↪  May 15th for Q2
847                     WHEN [F1] like 'Q3%' THEN DATEPART(YY,Convert(date,'15-08-'+RIGHT(F1,4),105)) -- Use
                        ↪  Aug 15th for Q3
848                     ELSE DATEPART(YY,Convert(date,'15-11-'+RIGHT(F1,4),105)) -- Use Nov 15th for Q4
849             END AS [Year]
850             ,CASE
851                     WHEN [F1] like 'Q1%' THEN DATEPART(QQ,Convert(date,'15-02-'+RIGHT(F1,4),105)) -- Use
                        ↪  Feb 15th for Q1
852                     WHEN [F1] like 'Q2%' THEN DATEPART(QQ,Convert(date,'15-05-'+RIGHT(F1,4),105)) -- Use
                        ↪  May 15th for Q2
853                     WHEN [F1] like 'Q3%' THEN DATEPART(QQ,Convert(date,'15-08-'+RIGHT(F1,4),105)) -- Use
                        ↪  Aug 15th for Q3
854                     ELSE DATEPART(QQ,Convert(date,'15-11-'+RIGHT(F1,4),105)) -- Use Nov 15th for Q4
855             END AS [Quarter]
856             ,CAST([Caution] as int) AS [Caution]
857             ,CAST([Absolute / Conditional discharge] as int) AS [Discharged]
858             ,CAST([Fine] as int) AS [Fine]
859             ,CAST([Community sentence] as int) AS [Community sentence]
860             ,CAST([Suspended sentence] as int) AS [Suspended sentence]
861             ,CAST([Immediate custody] as int) AS [Immediate custody]
862             ,CAST([Other disposal 4] as int) AS [Other]
863     INTO Weapons.BladedOffenceByOutcome
864     FROM [dbo].[BladedOffenceByAgeOutcomeQuarter]
865     -- Offence by Offence
866     BEGIN TRAN -- removing singular erroneous row
867             DELETE FROM [BladedOffenceByOffence]
868             WHERE [Time period  ] like '%Year ending%'
869     COMMIT TRAN
870     SELECT
871              year('20' + right([Time period  ],2)) AS [Year] -- Fixing '2008/09' format to singular year
872             ,CAST([Attempted murder] as int) AS [Attempted murder]
873             ,CAST([Threats to kill] as int) AS [Threats to kill]
874             ,CAST([Assault with injury and intent to cause serious harm] as int) AS [Assault]
875             ,CAST([Robbery] as int) AS [Robbery]
876             ,CAST([Rape] as int) AS [Rape]
877             ,CAST([Sexual assault] as int) AS [Sexual assault]
878             ,CAST([Homicide] as int) AS [Homicide]
879     INTO Weapons.BladedOffenceByOffence
880     FROM [dbo].[BladedOffenceByOffence]
881     -- Offence by Force Area
882             -- Need to unpivot this to match convention for Areas and Years so using cte
883     ;with HalfCleanBladeArea
884     AS (
885     SELECT
886              [F1] AS [Area]
887             ,CAST([Number] as int) AS [2009]
888             ,CAST([Number1] as int) AS [2010]
889             ,CAST([Number2] as int) AS [2011]
890             ,CAST([Number3] as int) AS [2012]
891             ,CAST([Number4] as int) AS [2013]
892             ,CAST([Number5] as int) AS [2014]
893             ,CAST([Number6] as int) AS [2015]
894             ,CAST([Number7] as int) AS [2016]
895             ,CAST([Number8] as int) AS [2017]
```

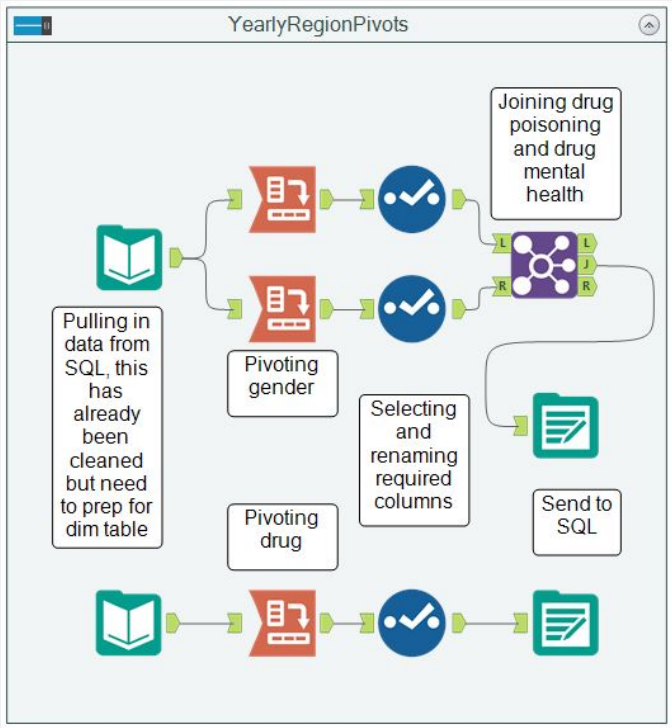
```
FROM [dbo].[BladedOffenceForceArea]
-- Only return relevant rows
WHERE [F1] IS NOT NULL
        AND [F1] NOT LIKE '%Region%'
        AND [F1] NOT LIKE 'WALES'
        AND [F1] NOT LIKE '%England%'
)
SELECT
        -- Better names and casting to suitable types
        year([Date]) AS [Date]
        ,[Area] AS [Area]
        ,MIN([value]) over (partition by [Date], [Area]) AS [Offences]
INTO Weapons.BladedOffenceByArea
FROM
(SELECT
        [Date],[Area],value
        FROM HalfCleanBladeArea
        unpivot (
                value for [Date] in
                ([2009],[2010],[2011],[2012],[2013],
                [2014],[2015],[2016],[2017])
        ) unpiv
        ) AS src
-- Moving original tables into dirty schema
ALTER SCHEMA dirty TRANSFER dbo.BladedOffenceByAgeOutcomeQuarter
ALTER SCHEMA dirty TRANSFER dbo.BladedOffenceByOffence
ALTER SCHEMA dirty TRANSFER dbo.BladedOffenceForceArea


/*----------------------------------
    PoliceTaserUse tables cleanse
----------------------------------*/
-- Completed in Alteryx, moving original data to dirty schema
ALTER SCHEMA Dirty TRANSFER dbo.PoliceTaserUse13
ALTER SCHEMA Dirty TRANSFER dbo.PoliceTaserUse14
ALTER SCHEMA Dirty TRANSFER dbo.PoliceTaserUse15
ALTER SCHEMA Dirty TRANSFER dbo.PoliceTaserUse16

/*-----------------------------------
    DrugDeathByArea table cleanse
-----------------------------------*/
-- Creating new 'Drug' schema
CREATE SCHEMA [Drug]
GO
-- This tabled cleansed in Alteryx and exported here, moving OG data to dirty
ALTER SCHEMA Dirty TRANSFER dbo.drugdeathbyarea
-- Just remove some of the oldest rows
ALTER SCHEMA temp transfer drug.deathbyarea
SELECT
        *
into Drug.DeathByRegion
FROM temp.deathbyarea
WHERE [Year] > 1999

/*----------------------------------
    DrugSeizures tables cleanse
----------------------------------*/
-- Completed in Alteryx, moving original data to dirty schema
ALTER SCHEMA Dirty TRANSFER dbo.DrugSeizuresForceArea
```

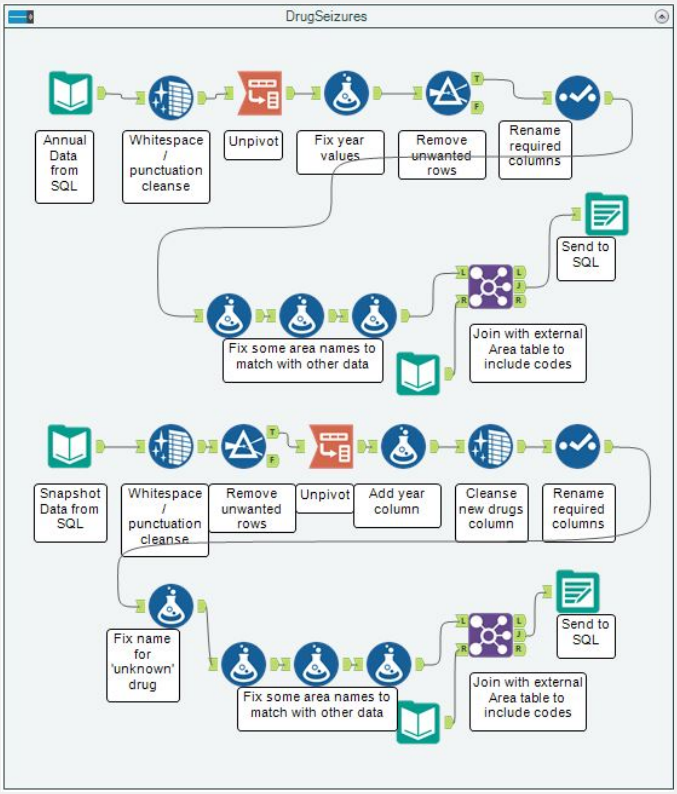
```
955 ALTER SCHEMA Dirty TRANSFER dbo.DrugSeizuresForceAreaSnapshot
956
957 /*-------------------------------------
958     DrugAdmissionsNHS tables cleanse
959 -------------------------------------*/
960 -- Completed in Alteryx, moving original data to dirty schema
961 ALTER SCHEMA Dirty TRANSFER dbo.DrugAdmissionsNHSMental
962 ALTER SCHEMA Dirty TRANSFER dbo.DrugAdmissionsNHSPoison
963
964 /*---------------------------------
965     DrugSurveyData tables cleanse
966 ---------------------------------*/
967 -- Completed in Alteryx, moving original data to dirty schema
968 ALTER SCHEMA Dirty TRANSFER dbo.DrugSurveyData
969
970 /*----------------------------------------
971     DeprivationIndicesLSOA tables cleanse
972 ----------------------------------------*/
973 -- Completed in Alteryx, moving original data to dirty schema
974 ALTER SCHEMA Dirty TRANSFER dbo.DeprivationIndicesLSOA
975 ALTER SCHEMA Dirty TRANSFER dbo.WalesDeprivationRanksLSOA
976
977 /*-----------------------------------
978     PopulationByLSOA tables cleanse
979 -----------------------------------*/
980 -- Completed in Alteryx, moving original data to dirty schema
981 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA11
982 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA12
983 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA13
984 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA14
985 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA15
986 ALTER SCHEMA Dirty TRANSFER dbo.PopulationByLSOA16
987
988 --           ___ _                  _                 ___                        _      _          --
989 --          / __\ | ___  __ _ _ __ (_)_ __   __ _    / __\___  _ __ ___  _ __ | | ___| |_ ___    --
990 --         / /  | |/ _ \/ _` | '_ \| | '_ \ / _` |  / /  / _ \| '_ ` _ \| '_ \| |/ _ \ __/ _ \   --
991 --        / /___| |  __/ (_| | | | | | | | | (_| | / /__| (_) | | | | | | |_) | |  __/ ||  __/   --
992 --        \____/|_|\___|\__,_|_| |_|_|_| |_|\__, | \____/\___/|_| |_| |_| .__/|_|\___|\__\___|   --
993 --                                           |___/                       |_|                      --
```

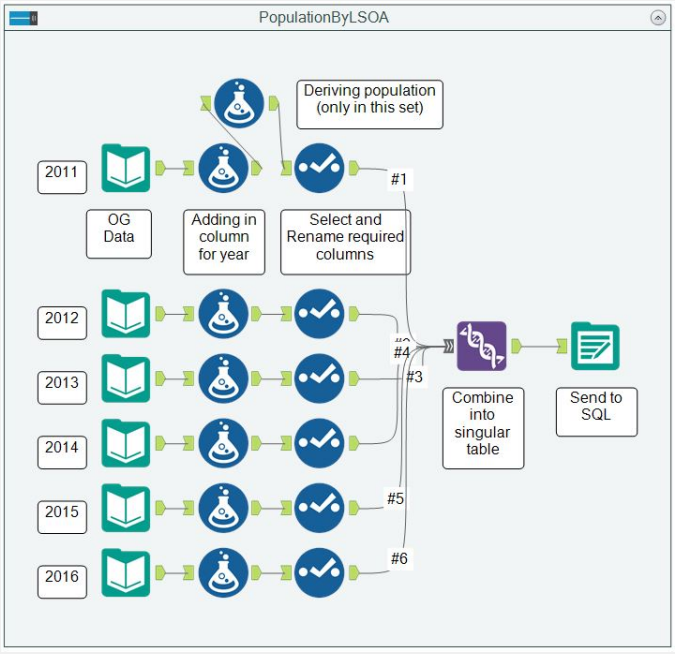
6.2 Alteryx Cleanse

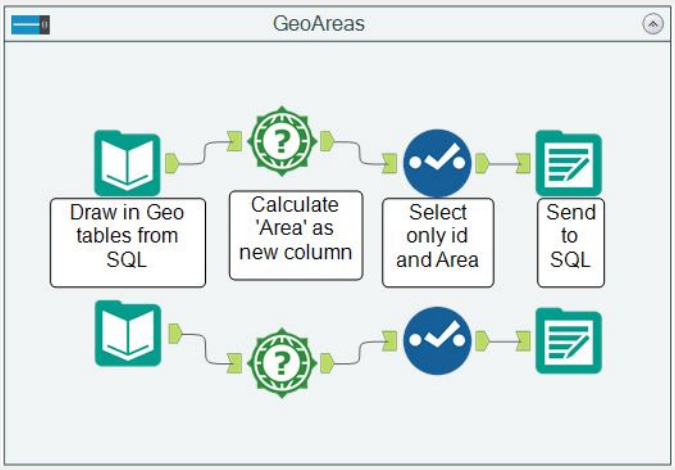


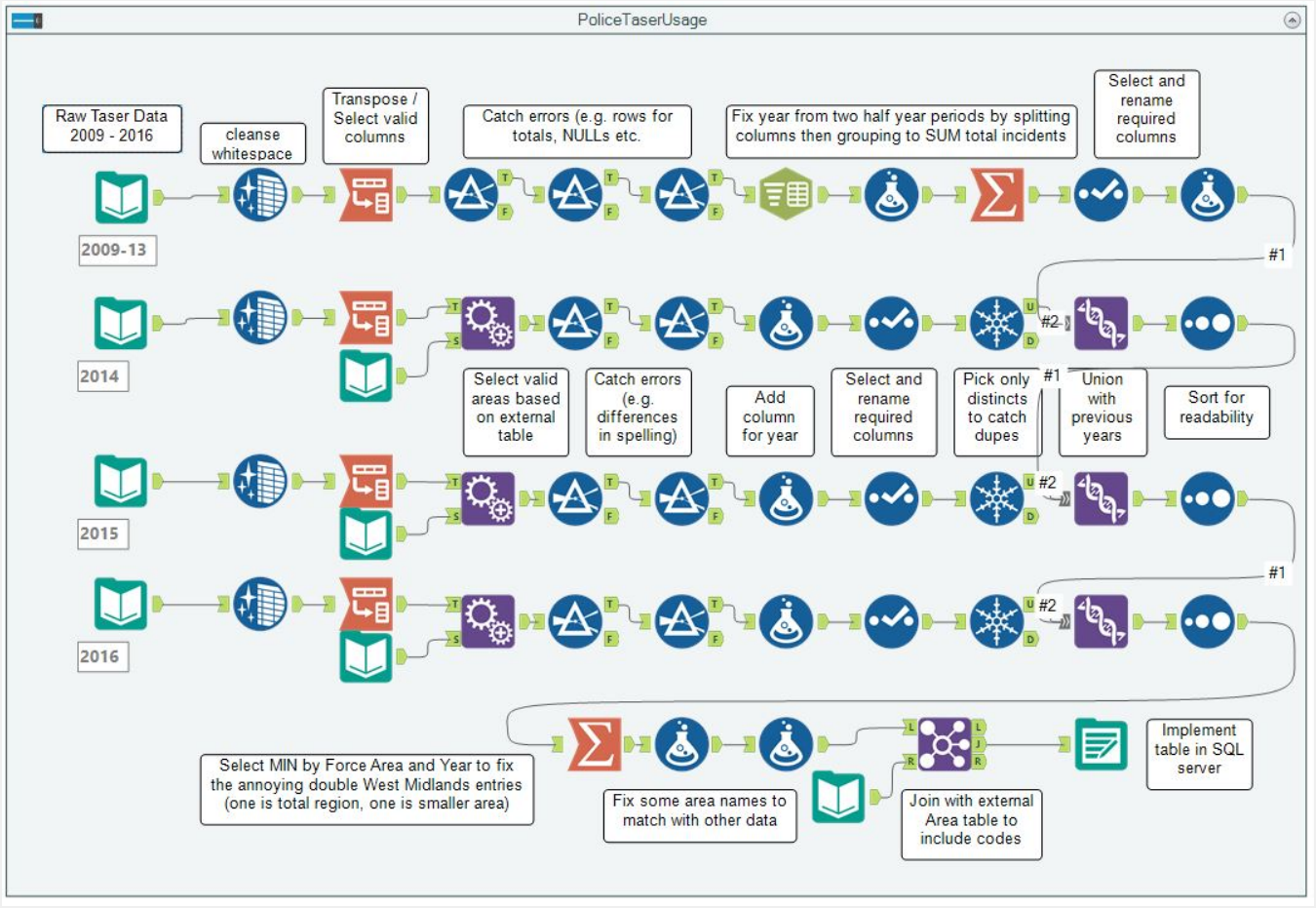


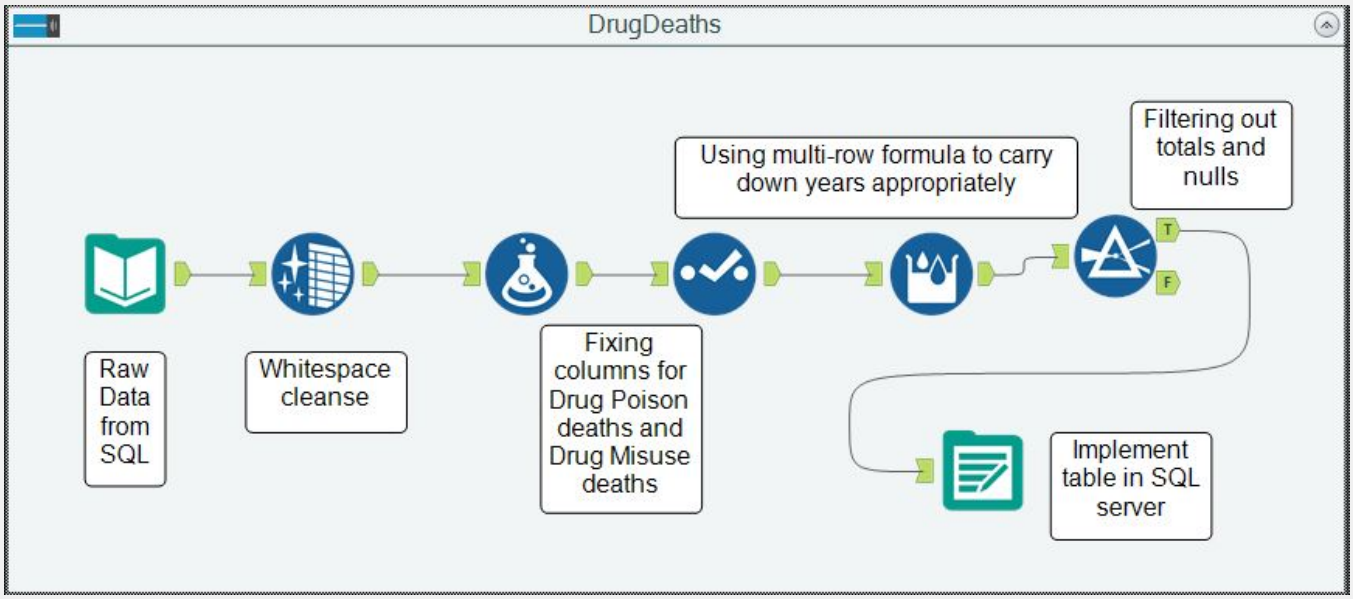

YearlyRegionPivots
Pulling in data from SQL, this has already been cleaned but need to prep for dim table
Pivoting gender
Pivoting drug
Selecting and renaming required columns
Joining drug poisoning and drug mental health
Send to SQL

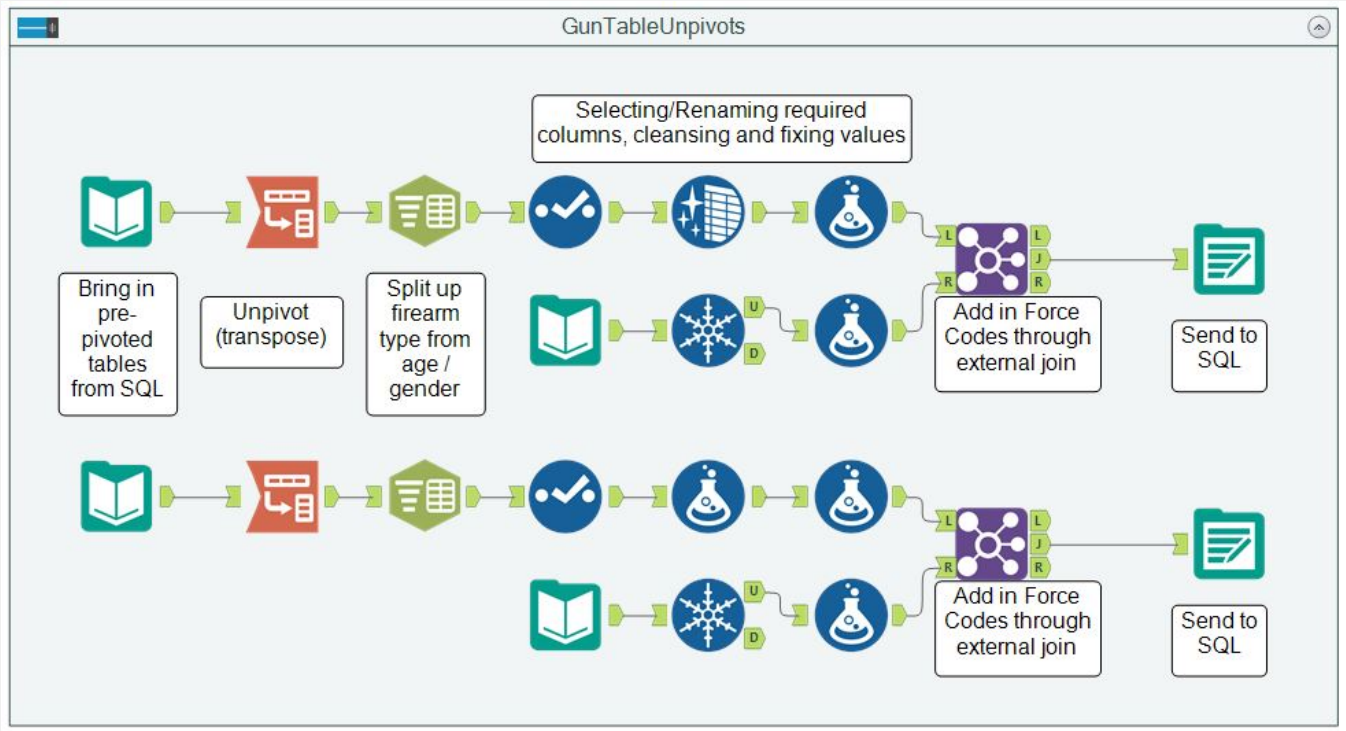

DrugSeizures
Annual Data from SQL
Whitespace / punctuation cleanse
Unpivot
Fix year values
Remove unwanted rows
Rename required columns
Send to SQL
Fix some area names to match with other data
Join with external Area table to include codes
Snapshot Data from SQL
Whitespace / punctuation cleanse
Remove unwanted rows
Unpivot
Add year column
Cleanse new drugs column
Rename required columns
Fix name for 'unknown' drug
Fix some area names to match with other data
Join with external Area table to include codes
Send to SQL

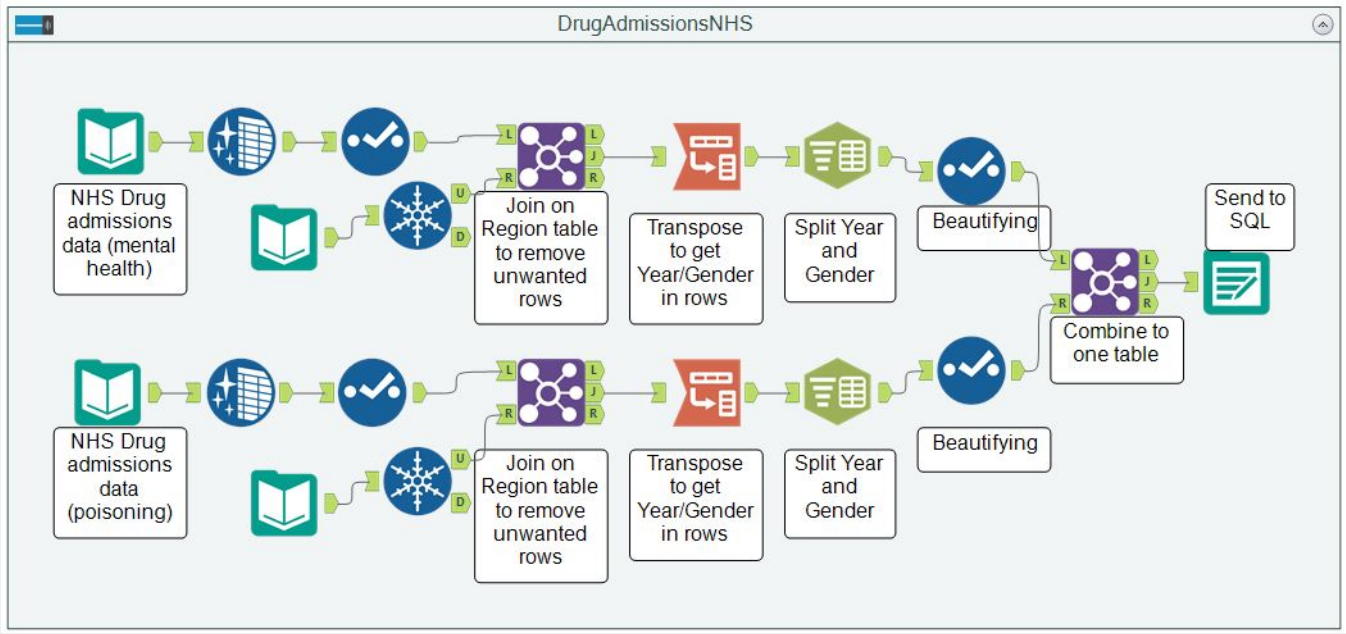

PopulationByLSOA
Deriving population (only in this set)
2011
OG Data
Adding in column for year
Select and Rename required columns
#1
2012
2013
2014
2015
2016
#3
#4
#5
#6
Combine into singular table
Send to SQL


GeoAreas
Draw in Geo tables from SQL
Calculate 'Area' as new column
Select only id and Area
Send to SQL


PoliceTaserUsage
Raw Taser Data 2009 - 2016
cleanse whitespace
Transpose / Select valid columns
Catch errors (e.g. rows for totals, NULLs etc.
Fix year from two half year periods by splitting columns then grouping to SUM total incidents
Select and rename required columns
2009-13
2014
2015
2016
Select valid areas based on external table
Catch errors (e.g. differences in spelling)
Add column for year
Select and rename required columns
Pick only distincts to catch dupes
Union with previous years
Sort for readability
Select MIN by Force Area and Year to fix the annoying double West Midlands entries (one is total region, one is smaller area)
Fix some area names to match with other data
Join with external Area table to include codes
Implement table in SQL server


DrugDeaths
Raw Data from SQL
Whitespace cleanse
Fixing columns for Drug Poison deaths and Drug Misuse deaths
Using multi-row formula to carry down years appropriately
Filtering out totals and nulls
Implement table in SQL server


GunTableUnpivots
Selecting/Renaming required columns, cleansing and fixing values
Bring in pre-pivoted tables from SQL
Unpivot (transpose)
Split up firearm type from age / gender
Add in Force Codes through external join
Send to SQL
Add in Force Codes through external join
Send to SQL


DrugAdmissionsNHS
NHS Drug admissions data (mental health)
Join on Region table to remove unwanted rows
Transpose to get Year/Gender in rows
Split Year and Gender
Beautifying
Combine to one table
Send to SQL
NHS Drug admissions data (poisoning)
Join on Region table to remove unwanted rows
Transpose to get Year/Gender in rows
Split Year and Gender
Beautifying

6.3 SQL Modelling

```
USE CrimeProject
GO

CREATE SCHEMA Dim
GO

--SELECT * FROM Drug.DeathbyRegion -- deaths by YEAR and REGION
--SELECT * FROM Drug.AdmissionsNHSRegion -- admissions by YEAR and REGION and GENDER
--SELECT * FROM Drug.SeizuresForceArea -- seizures by YEAR and FORCE AREA
--SELECT * FROM Drug.SeizuresSnapshot17 -- seizures by FORCE AREA and DRUG (2017)
--SELECT * FROM Drug.SurveyData -- drug use by YEAR and REGION and DRUG
--SELECT * FROM Geo.AreaCompare -- LSOA by FORCE AREA by REGION
--SELECT * FROM Geo.DeprivationRanksLSOA -- deprivation rank by LSOA (England)
--SELECT * FROM Geo.ForceArea -- area size and poly by FORCE AREA
--SELECT * FROM Geo.LSOA -- area size and poly by LSOA
--SELECT * FROM Geo.PopulationLSOA -- population by YEAR and LSOA
--SELECT * FROM Geo.WalesDeprivationRanksLSOA -- deprivation rank by LSOA (Wales)
--SELECT * FROM Police.CrimeDataStopSearch -- data by DATETIME(minute) and LSOA and CRIME TYPE
↪  (object) and GENDER
--SELECT * FROM Police.MatchedStreet -- data by YEAR and MONTH and LSOA and CRIME TYPE
--SELECT * FROM Police.TaserUse -- taser incidents by YEAR and FORCE AREA
--SELECT * FROM Weapons.BladedOffenceByAge -- minor/adult offences by YEAR and QUARTER
--SELECT * FROM Weapons.BladedOffenceByArea -- offences by YEAR and FORCE AREA
--SELECT * FROM Weapons.BladedOffenceByOffence -- various offences by YEAR
--SELECT * FROM Weapons.BladedOffenceByOutcome -- offence outcomes by YEAR and QUARTER
--SELECT * FROM Weapons.FirearmCertificatesForceArea -- certificate applications/totals and firearms
↪  by YEAR and FORCE AREA
--SELECT * FROM Weapons.FirearmDealersForceArea -- dealer license applications/totals by YEAR and
↪  FORCE AREA
--SELECT * FROM Weapons.FirearmOffenceByArea -- offences by YEAR and FORCE AREA
--SELECT * FROM Weapons.FirearmOffenceByInjury -- nationwide injury types by YEAR
--SELECT * FROM Weapons.FirearmOffenceByLocationType -- nationwide offence locations by YEAR
--SELECT * FROM Weapons.FirearmOffenceByOffence -- nationwide offence types by YEAR
--SELECT * FROM Weapons.GunsAgeForceArea -- firearm/shotgun certificates by YEAR and FORCE AREA and
↪  AGE (age needs unpivoting)
--SELECT * FROM Weapons.GunsGenderForceArea -- firearm/shotgun certificates by YEAR and FORCE AREA
↪  and GENDER (gender needs unpivoting)
--SELECT * FROM Weapons.ShotgunCertificatesForceArea -- certificate applications/totals and firearms
↪  by YEAR and FORCE AREA


/*----------------------
    Creating Dim.Geo
----------------------*/
-- For Region to ForceArea to LSOA
        -- including deprivation indice ranks
                -- 1 being the most deprived
-- ENGLAND
SELECT
         a.[Region name]
        ,a.[Region code]
        ,a.[Force area]
        ,a.[Force area code] AS [Force code]
        ,f.[Area (sqkm)] AS [Force size (sqkm)]
```

```
49          ,a.[LSOA name]
50          ,a.[LSOA code]
51          ,l.[Area (sqkm)] AS [LSOA size (sqkm)]
52          ,d.[Multiple deprivation]
53      ,d.[Income deprivation]
54      ,d.[Income deprivation (children)]
55      ,d.[Income deprivation (elderly)]
56      ,d.[Employment deprivation]
57      ,d.[Education/skills/training dep] AS [Education deprivation]
58      ,d.[Health deprivation]
59      ,d.[Crime]
60      ,d.[Barriers to housing/services]
61      ,d.[Living environment dep]
62      ,d.[Youth deprivation]
63      ,d.[Adult skills deprivation]
64      ,d.[Geographical barriers]
65      ,d.[Wider barriers]
66      ,d.[Indoors deprivation]
67      ,d.[Outdoors deprivation]
68  INTO Dim.GeoEngland
69  FROM Geo.AreaCompare a
70  JOIN Geo.LSOA l
71          on a.[LSOA code]=l.[lsoa code]
72  JOIN Geo.ForceArea f
73          on a.[Force area code] = f.[Area code]
74  JOIN Geo.DeprivationRanksLSOA d
75          on d.[LSOA code] = a.[LSOA code]
76  -- Adding primary id key
77  ALTER TABLE Dim.GeoEngland
78          Add id int identity primary key
79
80  -- WALES
81  SELECT
82           a.[Region name]
83          ,a.[Region code]
84          ,a.[Force area]
85          ,a.[Force area code] AS [Force code]
86          ,f.[Area (sqkm)] AS [Force size (sqkm)]
87          ,a.[LSOA name]
88          ,a.[LSOA code]
89          ,l.[Area (sqkm)] AS [LSOA size (sqkm)]
90          ,w.[Multiple deprivation]
91      ,w.[Income deprivation]
92      ,w.[Employment deprivation]
93      ,w.[Education deprivation]
94      ,w.[Health deprivation]
95      ,w.[Crime]
96      ,w.[Barriers to housing]
97          ,w.[Barriers to services]
98      ,w.[Living environment dep]
99  INTO Dim.GeoWales
100 FROM Geo.AreaCompare a
101 JOIN Geo.LSOA l
102         on a.[LSOA code]=l.[lsoa code]
103 JOIN Geo.ForceArea f
104         on a.[Force area code] = f.[Area code]
105 JOIN Geo.WalesDeprivationRanksLSOA w
106         on w.[LSOA code] = a.[LSOA code]
107 -- Adding primary id key
```

```
108 ALTER TABLE Dim.GeoWales
109         Add id int identity primary key
110
111 -- Combining the two (unfortunately lose some data but hey-ho)
112 ;with combinator AS (
113 SELECT
114         [Region name]
115         ,[Region code]
116         ,[Force area]
117         ,[Force code]
118         ,[Force size (sqkm)]
119         ,[LSOA name]
120         ,[LSOA code]
121         ,[LSOA size (sqkm)]
122         ,'England' as [Domain] -- because these are rank values so need to know domain of ranks
123         ,[Multiple deprivation]
124         ,[Income deprivation]
125         ,[Employment deprivation]
126         ,[Education deprivation]
127         ,[Health deprivation]
128         ,[Crime]
129         ,[Barriers to housing/services]
130         ,[Living environment dep]
131 FROM dim.GeoEngland
132 UNION
133 SELECT
134         [Region name]
135         ,[Region code]
136         ,[Force area]
137         ,[Force code]
138         ,[Force size (sqkm)]
139         ,[LSOA name]
140         ,[LSOA code]
141         ,[LSOA size (sqkm)]
142         ,'Wales' as [Domain] -- because these are rank values so need to know domain of ranks
143         ,[Multiple deprivation]
144         ,[Income deprivation]
145         ,[Employment deprivation]
146         ,[Education deprivation]
147         ,[Health deprivation]
148         ,[Crime]
149         ,[Barriers to housing]
150         ,[Living environment dep]
151 FROM dim.GeoWales )
152 SELECT
153         *
154 INTO dim.Geo
155 FROM combinator
156 -- Putting seperate tables in trash
157 ALTER SCHEMA Trash TRANSFER Dim.GeoEngland
158 ALTER SCHEMA Trash TRANSFER Dim.GeoWales
159 -- Adding primary id key
160 ALTER TABLE Dim.Geo
161         Add id int identity primary key
162
163
164 /*--------------------
165    Creating Dim.Pop
166 --------------------*/
```

```sql
SELECT
        *
INTO Dim.Pop
FROM geo.PopulationLSOA
-- Adding primary id key
ALTER TABLE Dim.Pop
        Add id int identity primary key


/*--------------------------
    Creating Dim.Date
--------------------------*/
-- CTE to union starting anchor date with dateadds
;WITH DatesUnion
AS
(
--anchor
SELECT
     CAST('2005-01-01' AS DATE) AS [AnchorDate] -- Anchor point
    ,YEAR(CAST('2005-01-01' AS DATE)) [Year] -- Derived year
        ,CASE
        WHEN MONTH(CAST('2005-01-01' AS DATE)) BETWEEN 3 AND 5
        THEN 'Spring'
        WHEN MONTH(CAST('2005-01-01' AS DATE)) BETWEEN 6 AND 8
        THEN 'Summer'
        WHEN MONTH(CAST('2005-01-01' AS DATE)) BETWEEN 9 AND 11
        THEN 'Autumn'
        ELSE 'Winter'
     END AS [Season] -- Derived season
        ,DATEPART(QUARTER,CAST('2005-01-01' AS DATE)) [Quarter] -- Derived quarter
    ,MONTH(CAST('2005-01-01' AS DATE)) [Month] -- Derived month
    /* ALL UNNECCESARY AS WE'RE ONLY GOING AS DEEP AS MONTH
        ,DATEPART(dd, CAST('2005-01-01' AS DATE)) [Day] -- Derived day (of month)
    ,DATENAME(dw, (CAST('2005-01-01' AS DATE))) [DayName] -- Derived day (name)
    ,CASE
        WHEN DATENAME(dw, (CAST('2005-01-01' AS DATE))) LIKE 'S%'
        THEN 0
        ELSE 1
     END as [WeekdayFlag] -- Is it a weekday? 1=y 0=n
         ,DATEPART(hh, CAST('2005-01-01' AS DATETIME)) [Hour] -- Derived hour
         ,DATEPART(mi, CAST('2005-01-01' AS DATETIME)) [Minute] -- Derived minute
         */
UNION ALL
SELECT -- Creating a new row for every monthly interval
     DATEADD(M, 1, AnchorDate)
    ,YEAR(DATEADD(M, 1, AnchorDate))
        ,CASE
        WHEN MONTH(DATEADD(M, 1, AnchorDate)) BETWEEN 3 AND 5
        THEN 'Spring'
        WHEN MONTH(DATEADD(M, 1, AnchorDate)) BETWEEN 6 AND 8
        THEN 'Summer'
        WHEN MONTH(DATEADD(M, 1, AnchorDate)) BETWEEN 9 AND 11
        THEN 'Autumn'
        ELSE 'Winter'
     END AS season
        ,DATEPART(QUARTER,DATEADD(M, 1, AnchorDate))
    ,MONTH(DATEADD(M, 1, AnchorDate))
    /* Not needed to this level of detail
        ,DATEPART(dd, DATEADD(mi, 5, [DateTime]))
```

```
226      ,DATENAME(dw, DATEADD(mi, 5, [DateTime]))
227      ,CASE
228          WHEN DATENAME(dw, DATEADD(mi, 5, [DateTime])) LIKE 'S%'
229          THEN 0
230          ELSE 1
231       END as isweekday
232           ,DATEPART(hh, DATEADD(mi, 5, [DateTime]))
233           ,DATEPART(mi, DATEADD(mi, 5, [DateTime]))
234           */
235  FROM DatesUnion
236  WHERE DATEADD(M, 1, AnchorDate) < GETDATE()
237  )
238  SELECT
239          [Year]
240          ,[Quarter]
241          ,Season
242          ,[Month]
243  INTO Dim.[Date]
244  FROM DatesUnion
245  OPTION (maxrecursion 0)
246  -- Adding primary id key
247  ALTER TABLE Dim.[Date]
248          Add id int identity primary key
249
250
251  /*---------------------------------
252      Creating Dim.YearlyForceArea
253  ---------------------------------*/
254  ;with Years as (
255  SELECT DISTINCT
256          [Year]
257  FROM dim.[Date]
258  WHERE [Year] > 2008
259  )
260  , ForceAreas as (
261  SELECT DISTINCT
262           [Force area]
263          ,[Force code]
264  FROM Dim.Geo
265  )
266  , DimStart as (
267  SELECT
268          *
269  FROM Years
270  cross join ForceAreas
271  )
272  -- Actual dim table builder
273  SELECT
274           dim.*
275          ,ds.[Drug seizures] AS [Police drug seizures]
276          ,pt.[Taser incidents] AS [Police taser incidents]
277          ,bo.Offences AS [Bladed offences]
278          ,fo.Offences AS [Firearm offences]
279          ,fc.[New applications granted] + sc.[New applications granted]
280          +fc.[Renewal applications granted] + sc.[Renewal applications granted]
281           AS [Firearm licenses granted]
282          ,fc.[New applications refused] + sc.[New applications refused]
283          +fc.[Renewal applications refused] + sc.[Renewal applications refused]
284          +fc.Revocations + sc.Revocations AS [Firearm licenses refused/revoked]
```

```
285         ,fc.[Total on issue (31/03)] + sc.[Total on issue (31/03)]
286          AS [Firearm certificates on issue]
287         ,fc.[Total firearms (31/03)] + sc.[Total shotguns (31/03)]
288          AS [Licensed firearms]
289         ,fd.[Total dealers (31/03)] AS [Firearm dealers]
290 INTO dim.YearlyForceArea
291 FROM DimStart dim
292 LEFT JOIN Drug.SeizuresForceArea ds
293         on dim.[Year] = ds.[Year]
294         AND dim.[Force code] = ds.[Area code]
295 LEFT JOIN police.TaserUse pt
296         -- [Year] in taser table is ntext so need to two-step convert
297         on dim.[Year] = Convert(int,Convert(varchar(100),pt.[Year]))
298         AND dim.[Force code] = pt.[Area code]
299 LEFT JOIN Weapons.BladedOffenceByArea bo
300         on dim.[Year] = bo.[Date]
301         AND CASE -- Fixing disparity in force area name (Northumberland v.s. Northumbria)
302                 WHEN dim.[Force area] = 'Northumbria' AND bo.[Area] = 'Northumberland' THEN 1
303                 WHEN dim.[Force area] = bo.[Area] THEN 1
304                 ELSE 0
305         END = 1
306 LEFT JOIN Weapons.FirearmOffenceByArea fo
307         on dim.[Year] = fo.[Date (March)]
308         AND CASE -- Fixing mistake in fo value for City of London
309                 WHEN dim.[Force area] = 'City of London' AND fo.[Area] = 'City  of London' THEN 1
310                 WHEN dim.[Force area] = fo.[Area] THEN 1
311                 ELSE 0
312         END = 1
313 LEFT JOIN Weapons.FirearmCertificatesForceArea fc
314         on dim.[Year] = fc.[Year]
315         AND CASE -- Fixing mistake in fc value for City of London
316                 WHEN dim.[Force area] = 'City of London' AND fc.[Police force area] = 'London, City
                    ↪  of' THEN 1
317                 WHEN dim.[Force area] = fc.[Police force area] THEN 1
318                 ELSE 0
319         END = 1
320 LEFT JOIN Weapons.ShotgunCertificatesForceArea sc
321         on dim.[Year] = sc.[Year]
322         AND CASE -- Fixing mistake in fc value for City of London
323                 WHEN dim.[Force area] = 'City of London' AND sc.[Police force area] = 'London, City
                    ↪  of' THEN 1
324                 WHEN dim.[Force area] = sc.[Police force area] THEN 1
325                 ELSE 0
326         END = 1
327 LEFT JOIN Weapons.FirearmDealersForceArea fd
328         on dim.[Year] = fd.[Year]
329         AND CASE -- Fixing mistake in fc value for City of London
330                 WHEN dim.[Force area] = 'City of London' AND fd.[Police force area] = 'London, City
                    ↪  of' THEN 1
331                 WHEN dim.[Force area] = fd.[Police force area] THEN 1
332                 ELSE 0
333         END = 1
334 ORDER BY [Year], [Force area]
335 -- Adding primary id key
336 ALTER TABLE Dim.YearlyForceArea
337         Add id int identity primary key
338
339
340 /*------------------------------
```

```
341      Creating Dim.YearlyRegion
342  ------------------------------*/
343  -- Pivoting a couple of tables to fit in this dim
344          -- Drug.AdmissionsNHSRegion, pivot gender
345          -- Drug.SurveyData, pivot drug
346  ALTER SCHEMA temp TRANSFER Drug.AdmissionsNHSRegion
347  ALTER SCHEMA temp TRANSFER Drug.SurveyData
348  -- Pivoted in Alteryx, ready to created dim
349  ;with Years as (
350  SELECT DISTINCT
351          [Year]
352  FROM dim.[Date]
353  )
354  , Regions as (
355  SELECT DISTINCT
356           [Region name]
357          ,[Region code]
358  FROM Dim.Geo
359  )
360  , DimStart as (
361  SELECT
362          *
363  FROM Years
364  cross join Regions
365  )
366  -- Actual dim table builder
367  SELECT
368           dim.*
369          ,sd.[Drug users (% of pop)]
370          ,sd.[Cannabis users (% of pop)]
371          ,sd.[Class A users (% of pop)]
372          ,sd.[Cocaine users (% of pop)]
373          ,sd.[Ecstasy users (% of pop)]
374          ,sd.[Amphetamine users (% of pop)]
375          ,sd.[Hallucinogen users (% of pop)]
376          ,dr.[Deaths (misuse)] AS [Drug misuse deaths]
377          ,dr.[Deaths (poison)] AS [Drug poison deaths]
378          ,nhs.[NHS admissions (drug mental health male)]
379          ,nhs.[NHS admissions (drug mental health female)]
380          ,nhs.[NHS admissions (drug poisoning male)]
381          ,nhs.[NHS admissions (drug poisoning female)]
382  INTO dim.YearlyRegion
383  FROM DimStart dim
384  LEFT JOIN drug.DeathByRegion dr
385          on dim.[Year] = dr.[Year]
386          AND dim.[Region code] = dr.[Region code]
387  LEFT JOIN drug.AdmissionsNHSRegion nhs
388          on dim.[Year] = nhs.[Year]
389          AND dim.[Region code] = nhs.[Region code]
390  LEFT JOIN drug.SurveyData sd
391          on dim.[Year] = sd.[Year]
392          AND dim.[Region code] = sd.[Region code]
393  -- Adding primary id key
394  ALTER TABLE Dim.YearlyRegion
395          Add id int identity primary key
396
397
398  /*---------------------------------
399      Creating Dim.YearlyNationwide
```

```
400 ----------------------------------*/
401 ;with Years as (
402 SELECT DISTINCT
403         [Year]
404 FROM dim.[Date]
405 WHERE [Year] < 2018
406 )
407 ,KnifeAge as (
408 SELECT
409         [Year], SUM(Minor) as Minor, SUM(Adult) as Adult
410 FROM Weapons.BladedOffenceByAge
411 WHERE [Year] between 2008 and 2016
412 GROUP BY [Year]
413 )
414 ,KnifeOutcome as (
415 SELECT
416         [Year], SUM(Caution) as Caution, SUM(Discharged) as Discharged,
417         SUM(Fine) as Fine, SUM([Community sentence]) as [Community sentence],
418         SUM([Suspended sentence]) as [Suspended sentence],
419         SUM([Immediate custody]) as [Immediate custody], SUM(Other) as Other
420 FROM Weapons.BladedOffenceByOutcome
421 WHERE [Year] between 2008 and 2016
422 GROUP BY [Year]
423 )
424 -- Actual dim table builder
425 SELECT
426          dim.*
427         ,fo.[Homicide] as [Firearm (homicide)]
428         ,fo.[Attempted murder] as [Firearm (attempted murder)]
429         ,fo.[Other violence] as [Firearm (violence)]
430         ,fo.[Robbery] as [Firearm (robbery)]
431         ,fo.[Burglary] as [Firearm (burglary)]
432         ,fo.[Criminal damage] as [Firearm (criminal damage)]
433         ,fo.[Public fear] as [Firearm (public fear)]
434         ,fo.[Possession] as [Firearm (possession)]
435         ,fo.[Other] as [Firearm (other)]
436         ,fi.[Fatal] as [Firearm injuries (fatal)]
437         ,fi.[Serious] as [Firearm injuries (serious)]
438         ,fi.[Lesser] as [Firearm injuries (lesser)]
439         ,fi.[No Injury] as [Firearm injuries (none)]
440         ,fl.[Shop] as [Firearm location (shop)]
441         ,fl.[Garage] as [Firearm location (garage)]
442         ,fl.[Post office] as [Firearm location (post office)]
443         ,fl.[Bank] as [Firearm location (bank)]
444         ,fl.[Residential] as [Firearm location (residential)]
445         ,fl.[Road] as [Firearm location (road)]
446         ,fl.[Other] as [Firearm location (other)]
447         ,ka.Adult [Adult knife offences]
448         ,ka.Minor [Juvenile knife offences]
449         ,bo.[Homicide] as [Knife (homicide)]
450         ,bo.[Attempted murder] as [Knife (attempted murder)]
451         ,bo.[Threats to kill] as [Knife (threats to kill)]
452         ,bo.[Assault] as [Knife (assault)]
453         ,bo.[Rape] as [Knife (rape)]
454         ,bo.[Sexual assault] as [Knife (sexual assault)]
455         ,bo.[Robbery] as [Knife (robbery)]
456         ,ko.[Immediate custody] as [Knife outcomes (immediate custody)]
457         ,ko.[Fine] as [Knife outcomes (fined)]
458         ,ko.[Community sentence] as [Knife outcomes (community sentence)]
```

```
459         ,ko.[Suspended sentence] as [Knife outcomes (suspended sentence)]
460         ,ko.[Caution] as [Knife outcomes (caution)]
461         ,ko.[Discharged] as [Knife outcomes (discharged)]
462         ,ko.[Other] as [Knife outcomes (other)]
463 INTO dim.YearlyNationwide
464 FROM Years dim
465 LEFT JOIN KnifeAge ka
466         ON dim.[Year] = ka.[Year]
467 LEFT JOIN KnifeOutcome ko
468         ON dim.[Year] = ko.[Year]
469 LEFT JOIN Weapons.BladedOffenceByOffence bo
470         ON dim.[Year] = bo.[Year]
471 LEFT JOIN Weapons.FirearmOffenceByInjury fi
472         ON dim.[Year] = fi.[Date (March)]
473 LEFT JOIN Weapons.FirearmOffenceByLocationType fl
474         ON dim.[Year] = fl.[Date (March)]
475 LEFT JOIN Weapons.FirearmOffenceByOffence fo
476         ON dim.[Year] = fo.[Date (March)]
477 ORDER BY [Year]
478 -- Adding primary id key
479 ALTER TABLE Dim.YearlyNationwide
480         Add id int identity primary key
481
482
483 /*-----------------------------------
484     Creating Dim.StopSearchExtras
485 -----------------------------------*/
486 -- Holding extra data from the StopSearch table that isn't included within street crime data
487 SELECT DISTINCT
488         [Search type]
489         ,Gender
490         ,[Age range]
491         ,legislation
492 INTO dim.StopSearchExtras
493 FROM police.CrimeDataStopSearch
494 -- adding row of NULLs for Street data that doesn't have this granularity
495 INSERT INTO dim.StopSearchExtras
496 VALUES (NULL,NULL,NULL,NULL)
497 -- Adding primary id key
498 ALTER TABLE Dim.StopSearchExtras
499         Add id int identity primary key
500
501
502 /*-------------------------------------
503     Creating Dim.CrimeTypesOutcomes
504 -------------------------------------*/
505 -- Holding merged extra data (Crime and Outcome) from the crime data tables
506 CREATE TABLE Temp.CrimeTypesOutcomes1 (
507         [Outcome flag] bit not null default(0)
508         )
509 CREATE TABLE Temp.CrimeTypesOutcomes2 (
510         [Crime type] varchar(30) default(NULL)
511         )
512 INSERT INTO Temp.CrimeTypesOutcomes1
513 VALUES (0),(1)
514 INSERT INTO Temp.CrimeTypesOutcomes2
515 VALUES (NULL),('Drugs'),('Weapons'),('Violence & sexual crime'),('Theft & robbery'),('Criminal
    ↪  damage or disorder')
516 -- Cross join to get all combinations
```

```sql
SELECT
        *
INTO Dim.CrimeTypesOutcomes
FROM Temp.CrimeTypesOutcomes2
CROSS JOIN Temp.CrimeTypesOutcomes1
-- Adding primary id key
ALTER TABLE Dim.CrimeTypesOutcomes
        Add id int identity primary key


/*-------------------------------------------------------
    Prepping supplementary tables for YearlyForceArea
-------------------------------------------------------*/
-- Need to unpivot GunsAge and GunsGender, DrugSeizuresSnapshot should be easier
CREATE SCHEMA Supp -- Supplementary

-- Drug Seizures Snapshot
SELECT
         ss.*
        ,yfa.id
INTO Supp.DrugSeizuresSnapshot
FROM drug.SeizuresSnapshot17 ss
left join dim.YearlyForceArea yfa
        ON Convert(int,Convert(varchar(100),ss.[Year])) = yfa.[Year] -- Matching on year
        AND ss.[Area code] = yfa.[Force code] -- Also matching on force code

-- Unpivoted Guns tables in Alteryx, now to join with YearlyForceArea and add foreign keys
ALTER SCHEMA temp TRANSFER Supp.FirearmCertsByAge
ALTER SCHEMA temp TRANSFER Supp.FirearmCertsByGender
-- Age
SELECT
         fca.*
        ,yfa.id
INTO Supp.FirearmCertsByAge
FROM Temp.FirearmCertsByAge fca
left join dim.YearlyForceArea yfa
        ON fca.[Year] = yfa.[Year] -- Matching on year
        AND fca.[Force code] = yfa.[Force code] -- Also matching on force code
-- Gender
SELECT
         fcg.*
        ,yfa.id
INTO Supp.FirearmCertsByGender
FROM Temp.FirearmCertsByGender fcg
left join dim.YearlyForceArea yfa
        ON fcg.[Year] = yfa.[Year] -- Matching on year
        AND fcg.[Force code] = yfa.[Force code] -- Also matching on force code

-- Adding foreign keys to link to YearlyForceArea
ALTER TABLE Supp.FirearmCertsByAge
ADD CONSTRAINT FK_FirearmCertsByAge_YearlyForceArea FOREIGN KEY (id)
    REFERENCES Dim.YearlyForceArea (id)
ALTER TABLE Supp.FirearmCertsByGender
ADD CONSTRAINT FK_FirearmCertsByGender_YearlyForceArea FOREIGN KEY (id)
    REFERENCES Dim.YearlyForceArea (id)
ALTER TABLE Supp.DrugSeizuresSnapshot
ADD CONSTRAINT FK_DrugSeizuresSnapshot_YearlyForceArea FOREIGN KEY (id)
    REFERENCES Dim.YearlyForceArea (id)

```

```sql

/*---------------------
    FACT TABLE BABY!
----------------------*/
-- Rather than do a cte, Street table is much bigger so going to put into 'Temp'
SELECT
         ms.*
        ,oo.[Outcome flag]
        ,cc.[Crime type] AS [New crime type]
INTO temp.ReadyStreetData
FROM police.MatchedStreet ms
join (select distinct * from Temp.OutcomesToOutcomes) oo
        on oo.[OG Outcome] =
                CASE -- Fixing joins on NULLs
                WHEN ms.[Last outcome category] IS NULL THEN 'NULL'
                ELSE ms.[Last outcome category]
        END
join (select distinct * from Temp.CrimeToCrime) cc
        on cc.[OG] =
                CASE -- Fixing joins on NULLs
                WHEN ms.[Crime type] IS NULL THEN 'NULL'
                ELSE ms.[Crime type]
        END

-- Creating a cte to correctly place Outcomes and Crime Types in StopSearch
;with crimetypeoutcome AS (
SELECT
         ss.*
        ,oo.[Outcome flag]
        ,cc.[Crime type]
FROM police.CrimeDataStopSearch ss
join (select distinct * from Temp.OutcomesToOutcomes) oo
        on oo.[OG Outcome] =
                CASE -- Fixing joins on NULLs
                WHEN ss.Outcome IS NULL THEN 'NULL'
                ELSE ss.Outcome
        END
join (select distinct * from Temp.CrimeToCrime) cc
        on cc.[OG] =
                CASE -- Fixing joins on NULLs
                WHEN ss.[Object of search] IS NULL THEN 'NULL'
                ELSE ss.[Object of search]
        END
)
-- Need to select both StopSearch and Street data in same format and union them
, bigboyunion AS (
-- StopSearch
SELECT
         dd.id AS [Date id]
        ,g.id AS [Geo id]
        ,cto.id AS [Crime and outcome id]
        ,sse.id AS [Further details id]
        ,yfa.id AS [Yearly force area id]
        ,yr.id AS [Yearly region id]
        ,yn.id AS [Yearly nationwide id]
        ,p.id AS [Population id]
FROM crimetypeoutcome ss
left join dim.[DateTable] dd
        on Year(ss.DateTimestamp) = dd.[Year]
```

```
635         AND Month(ss.DateTimestamp) = dd.[Month]
636 left join dim.Geo g
637         on ss.[LSOA code] = g.[LSOA code]
638 left join dim.CrimeTypesOutcomes cto
639         on CASE -- Fixing joins on NULLs
640                 WHEN ss.[Crime type] = 'NULL' AND cto.[Crime type] IS NULL THEN 1
641                 WHEN ss.[Crime type] = cto.[Crime type] THEN 1
642                 ELSE 0
643         END = 1
644         AND ss.[Outcome flag] = cto.[Outcome flag]
645 left join dim.StopSearchExtras sse
646         on CASE -- Fixing joins on NULLs
647                 WHEN ss.Gender IS NULL AND sse.Gender IS NULL THEN 1
648                 WHEN ss.Gender = sse.Gender THEN 1
649                 ELSE 0
650         END = 1
651         AND CASE -- Fixing joins on NULLs
652                 WHEN ss.[Search type] IS NULL AND sse.[Search type] IS NULL THEN 1
653                 WHEN ss.[Search type] = sse.[Search type] THEN 1
654                 ELSE 0
655         END = 1
656         AND CASE -- Fixing joins on NULLs
657                 WHEN ss.[Age range] IS NULL AND sse.[Age range] IS NULL THEN 1
658                 WHEN ss.[Age range] = sse.[Age range] THEN 1
659                 ELSE 0
660         END = 1
661         AND CASE -- Fixing joins on NULLs
662                 WHEN ss.legislation IS NULL AND sse.legislation IS NULL THEN 1
663                 WHEN ss.legislation = sse.legislation THEN 1
664                 ELSE 0
665         END = 1
666 left join dim.YearlyForceArea yfa
667         on Year(ss.DateTimestamp) = yfa.[Year]
668         AND g.[Force code] = yfa.[Force code]
669 left join dim.YearlyRegion yr
670         on Year(ss.DateTimestamp) = yr.[Year]
671         AND g.[Region code] = yr.[Region code]
672 left join dim.YearlyNationwide yn
673         on Year(ss.DateTimestamp) = yn.[Year]
674 left join dim.pop p
675         on Year(ss.DateTimestamp) = Convert(int,Convert(varchar(100),p.[Year]))
676         AND ss.[LSOA code] = p.[LSOA code]
677
678 UNION ALL
679
680 -- Street
681 SELECT
682          dd.id AS [Date id]
683         ,g.id AS [Geo id]
684         ,cto.id AS [Crime and outcome id]
685         ,295 AS [Further details id] -- no need to join here as Street doesn't have this data so
        ↪  just = NULL row
686         ,yfa.id AS [Yearly force area id]
687         ,yr.id AS [Yearly region id]
688         ,yn.id AS [Yearly nationwide id]
689         ,p.id AS [Population id]
690 FROM temp.ReadyStreetData sd
691 left join dim.[DateTable] dd
692         on sd.[Year] = dd.[Year]
```

```
693         AND sd.[Month] = dd.[Month]
694 left join dim.Geo g
695         on sd.[LSOA code] = g.[LSOA code]
696 left join dim.CrimeTypesOutcomes cto
697         on CASE -- Fixing joins on NULLs
698                 WHEN sd.[New crime type] = 'NULL' AND cto.[Crime type] IS NULL THEN 1
699                 WHEN sd.[New crime type] = cto.[Crime type] THEN 1
700                 ELSE 0
701         END = 1
702         AND sd.[Outcome flag] = cto.[Outcome flag]
703 left join dim.YearlyForceArea yfa
704         on sd.[Year] = yfa.[Year]
705         AND g.[Force code] = yfa.[Force code]
706 left join dim.YearlyRegion yr
707         on sd.[Year] = yr.[Year]
708         AND g.[Region code] = yr.[Region code]
709 left join dim.YearlyNationwide yn
710         on sd.[Year] = yn.[Year]
711 left join dim.pop p
712         on sd.[Year] = Convert(int,Convert(varchar(100),p.[Year]))
713         AND sd.[LSOA code] = p.[LSOA code]
714 )
715 SELECT
716         *
717 INTO dim.FactTable
718 FROM bigboyunion
719
720 -- Adding foreign key constraints to link to Dim Tables
721 ALTER TABLE dim.FactTable
722 ADD CONSTRAINT FK_FactTable_Date FOREIGN KEY ([Date id])
723     REFERENCES Dim.[DateTable] (id)
724 ALTER TABLE dim.FactTable
725 ADD CONSTRAINT FK_FactTable_Geo FOREIGN KEY ([Geo id])
726     REFERENCES Dim.Geo (id)
727 ALTER TABLE dim.FactTable
728 ADD CONSTRAINT FK_FactTable_CrimeTypesOutcomes FOREIGN KEY ([Crime and outcome id])
729     REFERENCES Dim.CrimeTypesOutcomes (id)
730 ALTER TABLE dim.FactTable
731 ADD CONSTRAINT FK_FactTable_FurtherDetails FOREIGN KEY ([Further details id])
732     REFERENCES Dim.StopSearchExtras (id)
733 ALTER TABLE dim.FactTable
734 ADD CONSTRAINT FK_FactTable_YearlyForceArea FOREIGN KEY ([Yearly force area id])
735     REFERENCES Dim.YearlyForceArea (id)
736 ALTER TABLE dim.FactTable
737 ADD CONSTRAINT FK_FactTable_YearlyRegion FOREIGN KEY ([Yearly region id])
738     REFERENCES Dim.YearlyRegion (id)
739 ALTER TABLE dim.FactTable
740 ADD CONSTRAINT FK_FactTable_YearlyNationwide FOREIGN KEY ([Yearly nationwide id])
741     REFERENCES Dim.YearlyNationwide (id)
742 ALTER TABLE dim.FactTable
743 ADD CONSTRAINT FK_FactTable_Population FOREIGN KEY ([Population id])
744     REFERENCES Dim.Pop (id)
745 -- Adding primary id key for fact table
746 ALTER TABLE dim.FactTable
747         Add [Fact id] int identity primary key
```

6.4 Dimensional Model