



# Advanced Databases Assignment

*A detailed report on the wellbeing of Young Lives children (2002-16), the inequality in education sector in Vietnam (2016-17), and the extent of crime in Manchester (2017-18)*

# Table of Contents

Table of Contents .....	1
<b>Task 1 - Database creation for Child Well-being monitor - Young Lives (2002-16).....</b>	<b>4</b>
<b>1.1 Abstract.....</b>	<b>4</b>
<b>1.2 Introduction .....</b>	<b>4</b>
<b>1.3 Relational Schema.....</b>	<b>4</b>
1.3.1 Identify the tables.....	4
1.3.2 Create the tables .....	5
1.3.3 Create relationships between tables .....	6
<b>1.4 Design Rationale .....</b>	<b>7</b>
<b>1.5 Design Considerations .....</b>	<b>7</b>
1.5.1 Database Normalisation .....	7
1.5.2 Constraints.....	7
1.5.3 Data Validation .....	8
1.5.4 Transaction and Concurrency Control (if any) .....	8
1.5.5 Error Handling.....	9
1.5.6 Security .....	9
1.5.7 Comments.....	10
<b>1.6 T-SQL Statements.....</b>	<b>11</b>
1.6.1 Tables.....	11
1.6.2 Views.....	24
1.6.3 Stored Procedures (if any) .....	40
1.6.4 Triggers (if any) .....	43
1.6.5 User Define Functions (if any) .....	46
<b>1.7. Report Design (Only for task 2 and task 3).....</b>	<b>49</b>
<b>1.8. Database Security .....</b>	<b>49</b>
<b>1.9 Database Backup and Restore Strategy .....</b>	<b>50</b>
<b>1.10 Data Science/Business Intelligence Techniques .....</b>	<b>50</b>
<b>1.11 Data Privacy, Ethical and legal issues .....</b>	<b>51</b>
<b>1.12 Conclusion .....</b>	<b>52</b>
Task 2 - Database and front-end reporting for Inequality in Education – Vietnam (2016-17) .....	54
<b>2.1 Abstract .....</b>	<b>54</b>
<b>2.2 Introduction .....</b>	<b>54</b>
<b>2.3 Relational Schema.....</b>	<b>54</b>
2.3.1 Identify the tables.....	54
2.3.2 Create the tables .....	54
2.3.3 Create relationships between tables .....	55
<b>2.4 Design Rationale .....</b>	<b>56</b>

<b>2.5 Design Considerations .....</b>	56
2.5.1 Database Normalisation .....	56
2.5.2 Constraints.....	56
2.5.3 Data Validation .....	57
2.5.4 Transaction and Concurrency Control (if any) .....	57
2.5.5 Error Handling.....	57
2.5.6 Security .....	58
2.5.7 Comments.....	58
<b>2.6 T-SQL Statements.....</b>	59
2.6.1 Tables.....	59
2.6.2 Views.....	68
2.6.3 Stored Procedures (if any) .....	76
2.6.4 Triggers (if any) .....	76
2.6.5 User Define Functions (if any) .....	79
<b>2.7. Report Design (Only for task 2 and task 3).....</b>	79
<b>2.8. Database Security .....</b>	85
<b>2.9 Database Backup and Restore Strategy .....</b>	85
<b>2.10 Data Science/Business Intelligence Techniques .....</b>	86
<b>2.11 Data Privacy, Ethical and legal issues .....</b>	88
<b>2.12 Conclusion .....</b>	89
Task 3 - Database and front-end reporting for Crime in Manchester (2017-18).....	90
<b>3.1 Abstract .....</b>	90
<b>3.2 Introduction .....</b>	90
<b>3.3 Relational Schema.....</b>	90
3.3.1 Identify the tables.....	90
3.3.2 Create the tables .....	91
3.3.3 Create relationships between tables .....	91
<b>3.4 Design Rationale .....</b>	91
<b>3.5 Design Considerations .....</b>	92
3.5.1 Database Normalisation .....	92
3.5.2 Constraints.....	92
3.5.3 Data Validation .....	92
3.5.4 Transaction and Concurrency Control (if any) .....	93
3.5.5 Error Handling.....	93
3.5.6 Security .....	93
3.5.7 Comments.....	93
<b>3.6 T-SQL Statements.....</b>	94
3.6.1 Tables.....	95

3.6.2 Views.....	99
3.6.3 Stored Procedures (if any) .....	103
3.6.4 Triggers (if any) .....	104
3.6.5 User Define Functions (if any) .....	104
<b>3.7. Report Design (Only for task 2 and task 3)</b> .....	105
<b>3.8. Database Security</b> .....	111
<b>3.9 Database Backup and Restore Strategy</b> .....	111
<b>3.10 Data Science/Business Intelligence Techniques</b> .....	111
<b>3.11 Data Privacy, Ethical and legal issues</b> .....	112
<b>3.12 Conclusion</b> .....	113
References .....	115
Appendix A - Abbreviations .....	116
Appendix B - Formulae .....	116

# Task 1 - Database creation for Child Well-being monitor - Young Lives (2002-16)

## 1.1 Abstract

[1] Young Lives is a global initiative on childhood poverty that began its roots in the year 2000. The initiative has been taken by the Oxford's Department of International Development (ODID) at the University of Oxford, United Kingdom. By its mission, YL has been tracking the progress of 12000 children for 15 years in 5 rounds, i.e., around 3000 children in each of the countries - Ethiopia (ET), India – Andhra Pradesh state (IN), Peru (PE), and Vietnam (VN), this was done to assess progress towards eradicating poverty by setting ambitious futuristic goals with the help of the United Nations (UN). YL cohorts have been split into two groups - namely younger cohorts (YC) and older cohorts (OC). There are 2000 younger cohorts and 1000 older cohorts per each country listed. YC has been the group who were born between the years 2001 and 2002 and OC group children were born between the years 1994 and 1995. All these children could be either from a rural or an urban area, and there are totally 20 sentinel sites per country and a total average of 147 children were recorded per sentinel site in each of these 4 countries. All these children are from varied backgrounds each with a different setback intertwined with wide-ranging causes and implications. Based on the data given, we try to demystify some or most of these problems in their lives in order to understand the causes and consequences.

## 1.2 Introduction

- A database named 'CP001' is created in which we store all the YL data from 2002 to 2016.
- Initially 4 .tab files were given in raw format, which needs to be imported into SSMS.
- Mode of raw source data upload – Direct import using 'SSMS Import and Export wizard'.

Option	Input Value
Source format of the files	Flat file
Header row delimiter	Tab {t}
Destination	Microsoft OLE DB Provider for SQL Server
Server name	LAPTOP-MK-2\SQLEXPRESS
Authentication	Windows / SQL server authentication
Database	CP001
Schema	dbo
Total number of files	4
Names of the raw tables	[dbo].[ethiopia_constructed] [dbo].[india_constructed] [dbo].[peru_constructed] [dbo].[vietnam_constructed]

## 1.3 Relational Schema

### 1.3.1 Identify the tables

- Looking at the data dictionaries, I decided to create/split the master data tables and transaction data tables from the existing raw tables, as part of the data modelling.
- The table identification is done as per the guidance from the documentation, where student's historical data is present, for example child identification and general characters, parents' or guardians' information, their education levels along with the child's, children's medical history, their household financial, socioeconomic status, effects on the family due to shocks they have encountered with, public programmes which benefitted the household, livestock ownership, etc.
- All the fields from the 4 countries have been used in order to complete this task.
- A schema has been defined = 'dbCare'

### 1.3.2 Create the tables

#### 1.3.2.1 Raw Combined Table creation – INPEVIETH

- In order to simplify the task, a common table was created using T-SQL which is called dbo.INPEVIETH which is a combined table for all the data present in each of raw tables uploaded.
- panel12345 was renamed to panel
- wealth index & sub-indices fields "\_new" was removed in some cases
- Peru placeid was renamed to commid in order to maintain consistency.
- Instead of adding null columns to all 4 tables, I'm simply adding one table after another into the main table.
- Inserted columns & selected columns should be same while inserting fields into a table.
- One challenge I faced during this table creation included mismatch of the total number of columns and table definition, and also because the public programmes were not the same for each country. So, all the columns from all the raw tables were listed first and the duplicates were removed in order to maintain completeness.
- The second challenge here is that this is a combined table in raw format, without a primary key. In order to tackle this issue, we create proper tables with normalization techniques using this final raw table only.
- The total number of rows and unique child IDs were validated before and after the merge.

#### 1.3.2.2 Master Data Tables Creation

- Wherever there are more than 5 labels for a value within the dictionary for the raw dataset, that dictionary has been used to create a separate table for example – Ethnicity, Language, Religion, Region, etc., each of which have only 2 columns – a value and a label. These values are present in our transaction tables and they can refer to these master data tables for labels.
- During this process sometimes the same key has been observed multiple times in different countries. In order to streamline this issue, I have created a composite key that has 2-digit country code, '\_', key as a combination key so that we could use all 4 tables at the same time from master data perspective.
- List of all master data tables created from the dictionaries (all 4 countries)
  - [dbCare].[Region]
  - [dbCare].[Language]
  - [dbCare].[Religion]
  - [dbCare].[Disability\_Scale]
  - [dbCare].[Ethnicity]
  - [dbCare].[Current\_Grade]
  - [dbCare].[School\_Type]
  - [dbCare].[Highest\_Grade]
  - [dbCare].[Relationship]

#### 1.3.2.3 Raw Table clean-up to setup relationship between Master Data & Transaction Data tables

- During validation, multiple checks were done to ensure data accuracy.
- If the years where the child's mother & father died were nulls or incorrect numbers, they would have been ignored, but we cannot manipulate their deaths, logically speaking. We are supposed to rectify these numbers in order to streamline with other deaths, and we cannot ignore these relevant numbers. They were disinclined in 2 situations,
  - For childid = 'VN071028', the father died in '88', which is updated to '1988'
  - For childid = 'IN061045', the mother died in '2', which is updated to '2002'
- Ethnicity, Language, Current grade, School Type, Highest grade, Caregiver's education, Dad's education, Mom's education, Household head's education – all these columns are now having a constant number of values by adding / maintaining preceding zeros and 'NaN' which means 'not a

number'. These values later will be used to create a link between master data and transaction data tables.

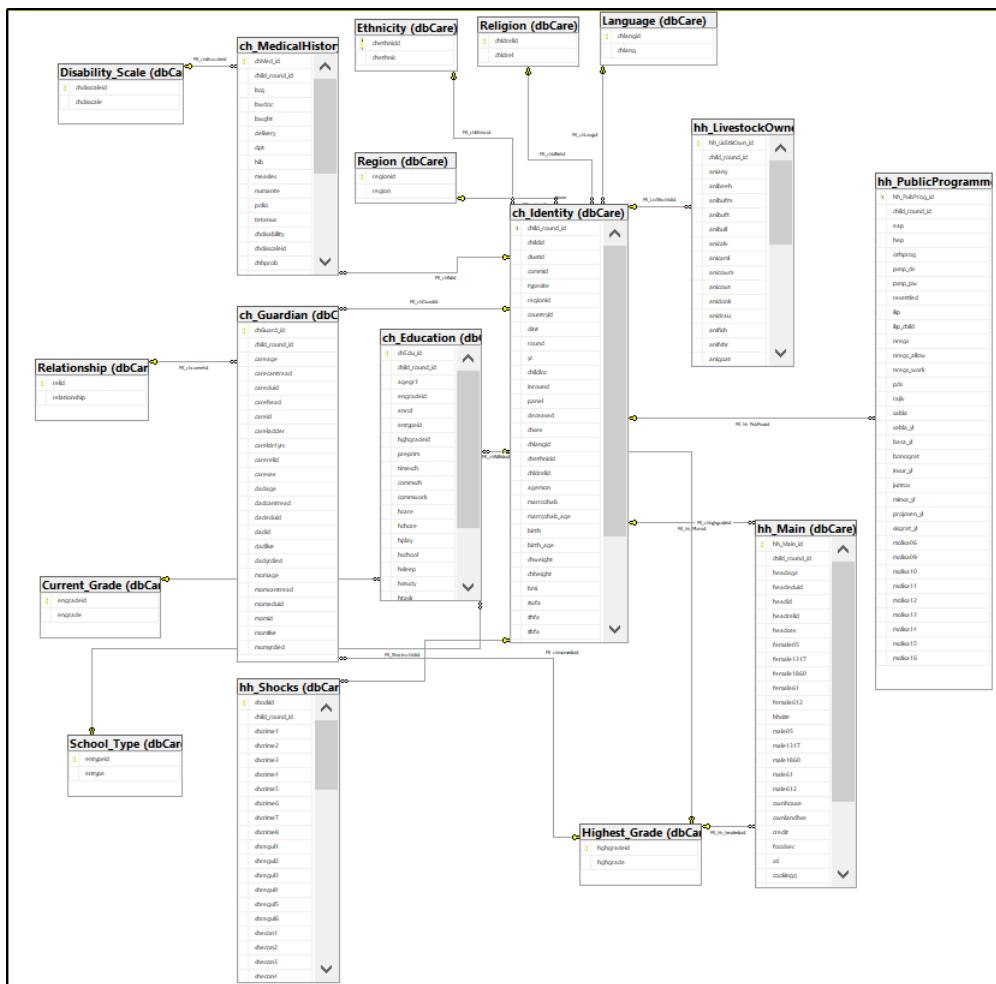
- Religion, Region, Disability Scale – these columns are adjusted for '99' or '88' which is either 'missing' or 'NA' where the length of the value is zero.

#### 1.3.2.4 Transactional Data Tables Creation

- Before creating the tables, the maximum amount of row-length was checked for each of the fields given, in order to determine which datatype would be suitable, also a thorough check was performed before a datatype is finalized for a field.
- After the necessary clean-up is complete as mentioned above, 8 transaction tables were created.
- 4 tables for child attributes, and 4 tables for household attributes are shown as below.
  - [dbCare].[ch\_Identity]
  - [dbCare].[ch\_MedicalHistory]
  - [dbCare].[ch\_Education]
  - [dbCare].[ch\_Guardian]
  - [dbCare].[hh\_Main]
  - [dbCare].[hh\_LivestockOwnership]
  - [dbCare].[hh\_PublicProgrammes]
  - [dbCare].[hh\_Shocks]

#### 1.3.3 Create relationships between tables

- The child based TD tables are prefixed as ch\_ and the household based TD tables are prefixed as hh\_
- All the master data tables that have a unique key are shown around the ch\_ and hh\_ tables.



## 1.4 Design Rationale

- In order to create a relational model, the MD tables and the TD tables have been segregated.
- The segmented transactional tables have specific purpose for each of their segments. For example, ch\_Identity - We can extract child's country, region, younger / older cohort, lives in a rural / urban area, gender, bodily features like height, weight, bmi, physical appearance, age, ethnicity, religion, language, etc.
- There are 253 columns totally from the raw table, which were logically split into 8 segments, without disrupting the completeness or the integrity of the information.
- It is imperative to first begin with MD tables because that is where we define our primary keys, and in order to logically map the rest of the tables and setup foreign keys.
- Whereas, the TD tables were connected using a child\_round\_id which is a composite key – a combination of childid and round. We remember that all these children have been interviewed on 5 occasions from 2002 to 2016. During these 5 rounds the children didn't change, hence this round value also needs to be considered for maintaining the uniqueness of the record.
- The central node / primary key of child\_round\_id is present in the ch\_Identity table and all the other TD tables refer to this tabular key while having a constraint over their child\_round\_id as foreign keys.
- According to this design, the views are simply linked / joined tables in which we can view the underlying information within combinational tables. For example, v01\_WellBeing is a view which has 2 TD tables (Child's Identity and Medical History) and 3 MD tables – (Disability Scale, region, ethnicity). Thus, we could infer the general wellbeing of a child using this structure. Similarly, all the other views have been built.
- All the reports are basically views, however, they have an underlying subquery that could have joined, unionized or even pivoted/unpivoted a combination of tables or views or even both sometimes as part of the blend. Each of these reports have aggregations based on the preliminary exploratory analysis using the raw table.
- A schema 'dbCare' is also defined here to show that this logical collection of database objects is segregated, if needed for multiple applications, accessibility, managing database security, etc. In our case, a schema owner can use only the database tables as the schema is defined for tables alone.
- In terms of performance, we don't have to do many joins in this model as the data is decentralized, as checked in terms of execution plan, time efficiency and statistics I/O, thus making it a performance-driven, normalized and scalable database that can handle heavy duty / big data as well at ease.
- A database role for stored procedures access only was created, and also a user who can access only the reports has been created. This means the data within the database is fully secure, and depending upon the type of the user, this security level may change.

## 1.5 Design Considerations

### 1.5.1 Database Normalisation

- Normalization is generally done to reduce redundant data and improve the data consistency within the database.
- The raw data table was initially given in 1NF level, and it was queried and validated multiple times to split into master data and transactional data tables.
- In terms of master data, we have a unique primary key which acts as foreign key in TD tables, and the TD tables have their individual primary keys all linked to the central table ch\_Identity through child\_round\_id. Through this linking we managed to achieve 3NF level within the schema which was initially 2NF in TD, and thus the normalization has been achieved in this relational model.

### 1.5.2 Constraints

- Constraints are generally used to restrict the type of data that can be inserted into a table, which enables data accuracy and increases reliability on the underlying implementation.

- Constraints are created during the creation of transaction data tables itself. All the constraints shown below are referencing either MD or TD tables previously created.

Constraint	Present in table	Referencing table(key)
FK_chEthnicid	dbCare.ch_Identity	[dbCare].[Ethnicity](chethnicid)
FK_chLangid		[dbCare].[Language](chlangid)
FK_regionid		[dbCare].[Region](regionid)
FK_chldRelid		[dbCare].[Religion](chldreid)
FK_chRdid	dbCare.ch_MedicalHistory	[dbCare].[ch_Identity](child_round_id)
FK_chdisscaleid		[dbCare].[Disability_Scale](chdisscaleid)
FK_chRdEduid	dbCare.ch_Education	[dbCare].[ch_Identity](child_round_id)
FK_chengradeid		[dbCare].[Current_Grade](engradeid)
FK_chentypeid		[dbCare].[School_Type](entypeid)
FK_chhghgradeid		[dbCare].[Highest_Grade](hghgradeid)
FK_chGuardid	dbCare.ch_Guardian	[dbCare].[ch_Identity](child_round_id)
FK_chcareuid		[dbCare].[Highest_Grade](hghgradeid)
FK_chdadeduid		[dbCare].[Highest_Grade](hghgradeid)
FK_chmomeduid		[dbCare].[Highest_Grade](hghgradeid)
FK_chcarerelid		[dbCare].[Relationship] (relid)
FK_hh_Mainid	dbCare.hh_Main	[dbCare].[ch_Identity](child_round_id)
FK_hh_headeduid		[dbCare].[Highest_Grade](hghgradeid)
FK_LivStkchildd	hh_LivestockOwnership	[dbCare].[ch_Identity](child_round_id)
FK_hh_PubProgid	hh_PublicProgrammes	[dbCare].[ch_Identity](child_round_id)
FK_Shockschildid	hh_Shocks	[dbCare].[ch_Identity](child_round_id)

### 1.5.3 Data Validation

- The number of records in [dbo].[ethiopia\_constructed] = 14995
- The number of records in [dbo].[india\_constructed] = 15097
- The number of records in [dbo].[peru\_constructed] = 13830
- The number of records in [dbo].[vietnam\_constructed] = 15000
- Totally = 58922 (raw data) -> INPEVIETH table
- However, we have a problem with the India dataset where there are 2 records with childid = 'childid' and this problem is resolved by deleting both records as they are basically empty records, so it is redundant to save them using a unique key.
- After deleting the 2 dummy records, totally we have
  - 58920 rows in INPEVIETH table.
  - 11784 unique childid values.
- The necessary validation has been done as and when each database object has been created.

### 1.5.4 Transaction and Concurrency Control (if any)

- It is assumed that for this task, I would not want to allow dirty reads or non-repeatable reads. Hence, in this scenario, a transaction isolation level of repeatable read has been given. This is to pre-empt lost updates that may take place when multiple transactions try to retrieve the same row.
- In the database CP001, all the non-admin users will only have access to the reports or stored procedures only, for which the isolation level has been setup.
  - Usually when two transactions are running in parallel, if one transaction is authorized to read the data that is in the background being altered by another transaction as a concurrent session without committing itself, this is known as a dirty read.
  - If a transaction reads the same row twice, but if the results are different, then this is known as a non-repeatable read.

- If two absolutely matching queries are executed at the same time and the rows retrieved by them are unlike, this is known as phantom read.
- To solve these problems, we setup isolation levels.
  - We could go with either of the two options in our case – Repeatable Read or Serializable isolation levels. Serializable is the highest restriction where operations are serially executing.
  - But we used Repeatable read to hold read locks on referenced rows and hold write locks on all rows where it does either an insert, or an update or a delete. Because the latter transaction cannot read or change these rows, it obviously prevents non-repeatable read.

### 1.5.5 Error Handling

- All the TD tables in task 1 have a try and catch exception block to display errors.
- BEGIN TRY and END TRY is where we try to insert our data from the raw table, and when it fails, we catch and return the exceptions.
- BEGIN CATCH and END CATCH is where we select the following system functions to comprehend the error occurred.
  - ERROR\_MESSAGE() is basically the message given by the error
  - ERROR\_LINE() is where the error has occurred in, the line number.
  - ERROR\_NUMBER() is the internally given number as per the records in SSMS.
  - ERROR\_SEVERITY() shows the severity of the error to understand the extent of the damage.
  - ERROR\_STATE() is the state of the error.

### 1.5.6 Security

- Based on the given task from the DML list, a user can either be granted or denied access to either SELECT, INSERT, UPDATE, DELETE, REFERENCES commands.

```
/*Grant or Deny access to tables / fields - cpuser1*/
GRANT SELECT ON dbCare.ch_LivestockOwnership TO cpuser1;
GO
DENY SELECT ON dbCare.ch_MedicalHistory (chrephealth1, chrephealth2, chrephealth3) TO cpuser1;
GO
```

- This can be achieved with the following which will be column level permissions.
- SQL Server Row level security is another option which needs a predicate function, security predicate, and a security policy. This would be part of the database admin's job to predetermine the solution to prevent data spilling and cyber-attacks.
- In order to perform this, we first create a filter predicate which checks for 'sysadmin' in the username. And later we create a security policy and attach to the predicate function.

```
/* Row level security - Predicate function*/
CREATE FUNCTION dbCare.predFunc (@user1 AS SYSNAME)
RETURNS TABLE
WITH SCHEMABINDING AS
RETURN SELECT 1 AS Access_Right
WHERE @user1 = USER_NAME() OR USER_NAME() = 'sysadmin'
GO
```

```
/*Create security policy and attach */
CREATE SECURITY POLICY Privacy_Policy
ADD FILTER PREDICATE TO dbCare.predFunc(adbuser1) ON dbCare.ch_Identity
    WITH (STATE = ON);
```

- Alternatively, we could also setup column level encryption which use symmetric keys for encrypting the data. There are multiple ways of anonymising the data.

### 1.5.7 Comments

- In order to ensure data integrity, the nulls are not filled with previous row values.
- Alternatively, we could write reusable script pointing to the method OBJECT\_ID('view\_name') with an IF clause to check if this object is null, and if yes, we CREATE the view from a table or else we simply ALTER the view from the table. Each of these DDL tasks can be achieved within BEGIN and END keywords inside the IF-ELSE logic.

## 1.6 T-SQL Statements

- Create the database CP001.

```
USE master;  
GO  
  
CREATE DATABASE CP001;
```

### 1.6.1 Tables

- Create combined raw table – INPEVIETH

```

USE CP001; GO

CREATE TABLE INPEVIETH (
    childid VARCHAR(50), clustid VARCHAR(50), commid VARCHAR(50), dadid VARCHAR(50), headid VARCHAR(50), momid VARCHAR(50),
    careid VARCHAR(50), agegrl VARCHAR(50), agemgn VARCHAR(50), aniany VARCHAR(50), anibeeh VARCHAR(50), anibusf VARCHAR(50),
    anibuf VARCHAR(50), anibulf VARCHAR(50), anicalv VARCHAR(50), anicam1 VARCHAR(50), anicown VARCHAR(50), anicowt VARCHAR(50),
    anidonk VARCHAR(50), anidrau VARCHAR(50), anifish VARCHAR(50), anifshr VARCHAR(50), anigoat VARCHAR(50), aniguin VARCHAR(50),
    anihu VARCHAR(50), anihieb VARCHAR(50), anihief VARCHAR(50), anillam VARCHAR(50), animilk VARCHAR(50), aniothar VARCHAR(50), anioxen VARCHAR(50),
    anipigs VARCHAR(50), anipoul VARCHAR(50), anirabb VARCHAR(50), anirumi VARCHAR(50), anishee VARCHAR(50), anishri VARCHAR(50),
    anisnal VARCHAR(50), anispes VARCHAR(50), aniybul VARCHAR(50), bcg VARCHAR(50), beca_y1 VARCHAR(50), birth VARCHAR(50),
    birth_age VARCHAR(50), bmi VARCHAR(50), bonograt VARCHAR(50), bwdoc VARCHAR(50), bwght VARCHAR(50), careage VARCHAR(50),
    carecanread VARCHAR(50), caredu VARCHAR(50), carehead VARCHAR(50), careladder VARCHAR(50), careldry4rs VARCHAR(50), carerel VARCHAR(50),
    caresex VARCHAR(50), cd VARCHAR(50), chalcohol VARCHAR(50), chdisability VARCHAR(50), chdisscale VARCHAR(50), chethnic VARCHAR(50),
    chhealth VARCHAR(50), chheight VARCHAR(50), chhrprob VARCHAR(50), chhrel VARCHAR(50), childloc VARCHAR(50), chillness VARCHAR(50),
    chinjury VARCHAR(50), chlang VARCHAR(50), chlrdrel VARCHAR(50), chmghtdie VARCHAR(50), chrepehealth1 VARCHAR(50),
    chrepehealth2 VARCHAR(50), chrepehealth3 VARCHAR(50), chrepehealth4 VARCHAR(50), chsex VARCHAR(50), chsmoke VARCHAR(50), chweight VARCHAR(50),
    cladden VARCHAR(50), commsch VARCHAR(50), comwork VARCHAR(50), cookinga VARCHAR(50), credit VARCHAR(50), dadage VARCHAR(50),
    dadcantread VARCHAR(50), dadedu VARCHAR(50), dadlive VARCHAR(50), dadydrdied VARCHAR(50), deceased VARCHAR(50), delivery VARCHAR(50),
    dini VARCHAR(50), dpt VARCHAR(50), drwaterd VARCHAR(50), eap VARCHAR(50), eleca VARCHAR(50), engrade VARCHAR(50), enrol VARCHAR(50),
    entype VARCHAR(50), fbfa VARCHAR(50), female05 VARCHAR(50), female1317 VARCHAR(50), female1860 VARCHAR(50), female61 VARCHAR(50),
    female612 VARCHAR(50), fhfa VARCHAR(50), foodsec VARCHAR(50), fwfa VARCHAR(50), fwf1 VARCHAR(50), hcare VARCHAR(50),
    hchore VARCHAR(50), headage VARCHAR(50), headedu VARCHAR(50), headrel VARCHAR(50), headsex VARCHAR(50), hep VARCHAR(50), hghgrade VARCHAR(50),
    hhsize VARCHAR(50), hib VARCHAR(50), hplay VARCHAR(50), hq VARCHAR(50), hschool VARCHAR(50), hsleep VARCHAR(50), hstdy VARCHAR(50),
    htask VARCHAR(50), hwrok VARCHAR(50), ikp VARCHAR(50), ikp_chilc VARCHAR(50), inround VARCHAR(50), insur_y1 VARCHAR(50),
    juntos VARCHAR(50), levread VARCHAR(50), levlwrit VARCHAR(50), literate VARCHAR(50), male05 VARCHAR(50), male1317 VARCHAR(50),
    male1860 VARCHAR(50), male61 VARCHAR(50), male612 VARCHAR(50), marrcohah VARCHAR(50), marrcohah_age VARCHAR(50),
    measles VARCHAR(50), mins_y1 VARCHAR(50), molisa06 VARCHAR(50), molisa09 VARCHAR(50), molisa10 VARCHAR(50), molisai1 VARCHAR(50),
    molisa12 VARCHAR(50), molisa13 VARCHAR(50), molisa14 VARCHAR(50), molisa15 VARCHAR(50), molisa16 VARCHAR(50), momage VARCHAR(50),
    momcantread VARCHAR(50), momedu VARCHAR(50), momlive VARCHAR(50), momdyrdied VARCHAR(50), nregs VARCHAR(50), nregs_allow VARCHAR(50),
    nregs_work VARCHAR(50), numante VARCHAR(50), othprog VARCHAR(50), ownhouse VARCHAR(50), ownlandse VARCHAR(50), panel VARCHAR(50),
    pds VARCHAR(50), polis VARCHAR(50), preprim VARCHAR(50), projoven_y1 VARCHAR(50), psnp_d5 VARCHAR(50), psnp_pw VARCHAR(50), rajiv VARCHAR(50),
    region VARCHAR(50), resettled VARCHAR(50), round VARCHAR(50), sabla VARCHAR(50), sabla_y1 VARCHAR(50), shcrime1 VARCHAR(50),
    shcrime2 VARCHAR(50), shcrime3 VARCHAR(50), shcrime4 VARCHAR(50), shcrime5 VARCHAR(50), shcrime6 VARCHAR(50), shcrime7 VARCHAR(50),
    shcrime8 VARCHAR(50), shecon1 VARCHAR(50), shecon2 VARCHAR(50), shecon3 VARCHAR(50), shecon4 VARCHAR(50), shecon5 VARCHAR(50),
    shecon6 VARCHAR(50), shecon7 VARCHAR(50), shecon8 VARCHAR(50), shecon9 VARCHAR(50), shecon10 VARCHAR(50), shecon11 VARCHAR(50),
    shecon12 VARCHAR(50), shecon13 VARCHAR(50), shecon14 VARCHAR(50), shenv1 VARCHAR(50), shenv2 VARCHAR(50), shenv3 VARCHAR(50),
    shenv4 VARCHAR(50), shenv5 VARCHAR(50), shenv6 VARCHAR(50), shenv7 VARCHAR(50), shenv8 VARCHAR(50), shenv9 VARCHAR(50),
    shenv10 VARCHAR(50), shenv11 VARCHAR(50), shenv12 VARCHAR(50), shenv13 VARCHAR(50), shenv14 VARCHAR(50), shenv15 VARCHAR(50), shenv16 VARCHAR(50),
    shfm1 VARCHAR(50), shfm2 VARCHAR(50), shfm3 VARCHAR(50), shfm4 VARCHAR(50), shfm5 VARCHAR(50), shfm6 VARCHAR(50), shfm7 VARCHAR(50), shfm8 VARCHAR(50), shfm9 VARCHAR(50),
    shfm10 VARCHAR(50), shfm11 VARCHAR(50), shfm12 VARCHAR(50), shfm13 VARCHAR(50), shfm14 VARCHAR(50), shfm15 VARCHAR(50),
    shfm16 VARCHAR(50), shfm17 VARCHAR(50), shfm18 VARCHAR(50), shhouse1 VARCHAR(50), shhouse2 VARCHAR(50), shhouse3 VARCHAR(50),
    shother VARCHAR(50), shregul1 VARCHAR(50), shregul2 VARCHAR(50), shregul3 VARCHAR(50), shregul4 VARCHAR(50), shregul5 VARCHAR(50),
    shregul6 VARCHAR(50), sisgrat_y1 VARCHAR(50), stunting VARCHAR(50), sv VARCHAR(50), tetanus VARCHAR(50), thinness VARCHAR(50),
    timesch VARCHAR(50), toiletq VARCHAR(50), typesite VARCHAR(50), underweight VARCHAR(50), wi VARCHAR(50), yc VARCHAR(50), zbeta VARCHAR(50),
    zhfa VARCHAR(50), zwfa VARCHAR(50), zwfl VARCHAR(50),
)
```

- Insert data into INPEVIETH Table from Ethiopia's raw table

- Insert data into INPEVIETH Table from India's raw table

```
USE CP001; GO
```

```
INSERT INTO [CP001].[dbCare].[INPEVIETH] ( childid, yc, round, inround, panel, deceased, dint, clustid, commid, region, typesite, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohab, marrcohab_age, birth, birth_age, chweight, chheight, bmi, zwfa, zhfa, zbf, zwfl, fwfl, fhfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, polio, dpt, hib, chmichtdie, chillness, chinjury, chprob, chdisability, chdisscale, chsmoke, chalcohol, chrepehealth1, chrepehealth2, chrepehealth3, chrepehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, hghgrade, timesch, levrlread, levrlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dаддаге, dadcantread, dаддаге, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headsex, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi, hq, sv, cd, drwaterq, toiletq, elec, cookingq, aniany, animilk, anidrau, anirumi, anicowm, anicowt, anicalv, anibufm, anibuft, anibull, anihebu, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anispec, anihelf, aniybul, anibeeh, anifish, anishri, anifshr, aniothr, ownlandse, ownhouse, pds, nregs, nregs_work, nregs_allow, rajiv, sabla, sabla_y1, ikp, ikp_child, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shcrime8, shcrime9, shcon6, shcon7, shcon8, shcon9, shcon10, shcon11, shcon12, shcon13, shcon14, shen1, shen2, shen3, shen4, shen5, shen6, shen7, shen8, shen9, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12 )  
SELECT childid, yc, round, inround, panel12345 AS panel, deceased, dint, clustid, commid, region, typesite, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohab, marrcohab_age, birth, birth_age, chweight, chheight, bmi, zwfa, zhfa, zbf, zwfl, fwfl, fhfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, polio, dpt, hib, chmichtdie, chillness, chinjury, chprob, chdisability, chdisscale, chsmoke, chalcohol, chrepehealth1, chrepehealth2, chrepehealth3, chrepehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, hghgrade, timesch, levrlread, levrlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dаддаге, dadcantread, dаддаге, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi, hq, sv, cd, drwaterq, toiletq, elec, cookingq, aniany, animilk, anidrau, anirumi, anicowm, anicowt, anicalv, anibufm, anibuft, anibull, anihebu, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anispec, anihelf, aniybul, anibeeh, anifish, anishri, anifshr, aniothr, ownlandse, ownhouse, pds, nregs, nregs_work, nregs_allow, rajiv, sabla, sabla_y1, ikp, ikp_child, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shcrime8, shregul1, shregul2, shregul3, shregul4, shregul5, shregul6, shcon1, shcon2, shcon3, shcon4, shcon5, shcon6, shcon7, shcon8, shcon9, shcon10, shcon11, shcon12, shcon13, shcon14, shen1, shen2, shen3, shen4, shen5, shen6, shen7, shen8, shen9, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12 )  
FROM dbo.india_constructed;
```

- Insert data into INPEVIETH Table from Peru's raw table

```
USE CP001; GO
```

```
INSERT INTO [CP001].[dbCare].[INPEVIETH] ( childid, yc, round, inround, panel, deceased, dint, placeid AS commid, clustid, typesite, region, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohab, marrcohab_age, birth, birth_age, chweight, chheight, bmi, zwfa, zhfa, zbf, zwfl, fwfl, fhfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, polio, dpt, hib, chmichtdie, chillness, chinjury, chprob, chdisability, chdisscale, chsmoke, chalcohol, chrepehealth1, chrepehealth2, chrepehealth3, chrepehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, timesch, levrlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dаддаге, dadcantread, dаддаге, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi, hq, sv, cd, drwaterq, toiletq, elec, cookingq, aniany, animilk, anidrau, anirumi, anispec, anicowm, anicowt, anioxen, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anilam, aniguin, anisnai, anibeeh, anifish, anishri, anifshr, aniothr, ownlandse, ownhouse, juntos, bonograt, sisgrat_y1, mins_y1, insur_y1, beca_y1, projoven_y1, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shcrime8, shregul1, shregul2, shregul3, shregul4, shregul5, shregul6, shcon1, shcon2, shcon3, shcon4, shcon5, shcon6, shcon7, shcon8, shcon9, shcon10, shcon11, shcon12, shcon13, shcon14, shen1, shen2, shen3, shen4, shen5, shen6, shen7, shen8, shen9, shen10, shen11, shen12, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12, shfam13, shfam14, shfam15, shfam16, shfam17, shfam18, shother )  
SELECT childid, yc, round, inround, panel12345 AS panel, deceased, dint, placeid AS commid, clustid, typesite, region, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohab, marrcohab_age, birth, birth_age, chweight, chheight, bmi, zwfa, zhfa, zbf, zwfl, fwfl, fhfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, polio, dpt, hib, chmichtdie, chillness, chinjury, chprob, chdisability, chdisscale, chsmoke, chalcohol, chrepehealth1, chrepehealth2, chrepehealth3, chrepehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, timesch, levrlread, levrlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dаддаге, dadcantread, dаддаге, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi, hq, sv, cd, drwaterq, toiletq, elec, cookingq, aniany, animilk, anidrau, anirumi, anispec, anicowm, anicowt, anioxen, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anilam, aniguin, anisnai, anibeeh, anifish, anishri, anifshr, aniothr, ownlandse, ownhouse, juntos, bonograt, sisgrat_y1, mins_y1, insur_y1, beca_y1, projoven_y1, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shcrime8, shregul1, shregul2, shregul3, shregul4, shregul5, shregul6, shcon1, shcon2, shcon3, shcon4, shcon5, shcon6, shcon7, shcon8, shcon9, shcon10, shcon11, shcon12, shcon13, shcon14, shen1, shen2, shen3, shen4, shen5, shen6, shen7, shen8, shen9, shen10, shen11, shen12, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12, shfam13, shfam14, shfam15, shfam16, shfam17, shfam18, shother )  
FROM dbo.peru_constructed;
```

- Insert data into INPEVIETH Table from Vietnam's raw table

```

USE CP001; GO

INSERT INTO [CP001].[dbCare].[INPEVIETH] ( childid, yc, round, inround, panel, deceased, dint, commid, clustid, typesite, region, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohob, marrcohob_age, birth, birth_age, chweight, chheight, bmi, zwfa, zfha, zbfra, zwfl, fwfl, fhfa, fwfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, dpt, polio, hib, chmightdie, chillness, chinjury, chprob, chdisability, chdissscale, chsmoke, chalcohol, chrehealth1, chrehealth2, chrehealth3, chrehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, hghgrade, timesch, levlrread, levlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dadage, dadcantread, dadyrdied, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headsex, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi, hq, sv, cd, drwaterq, toiletq, elecq, cookingq, aniany, animilk, anidrau, anirumi, anicowm, anicowt, anicalv, anibufm, anibull, anihebu, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anispec, aniothr, ownlandhse, ownhouse, molisa06, molisa09, molisa10, molisa11, molisa12, molisa13, molisa14, molisa15, molisa16, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shcrime8, shregul1, shregul2, shregul3, shregul5, shecon1, shecon2, shecon3, shecon4, shecon5, shecon6, shecon7, shecon8, shecon9, shecon10, shecon11, shecon12, shecon13, shecon14, shenv1, shenv2, shenv3, shenv4, shenv5, shenv6, shenv7, shenv8, shenv9, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12, shfam13, shfam14, shfam18, shother, shenv13 )
SELECT childid, yc, round, inround, panel12345 as panel, deceased, dint, commid, clustid, typesite, region, childloc, chsex, chlang, chethnic, chdrel, agemon, marrcohob, marrcohob_age, birth, birth_age, chweight, chheight, bmi, zwfa, zfha, zbfra, zwfl, fwfl, fhfa, fwfa, fbfa, underweight, stunting, thinness, bwght, bwdoc, numante, delivery, tetanus, bcg, measles, dpt, polio, hib, chmightdie, chillness, chinjury, chprob, chdisability, chdissscale, chsmoke, chalcohol, chrehealth1, chrehealth2, chrehealth3, chrehealth4, chhrel, chhealth, cladder, hsleep, hcare, hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch, preprim, agegr1, enrol, engrade, entype, hghgrade, timesch, levlrread, levlwrit, literate, careid, caredu, carehead, careage, caresex, carerel, carecantread, careladder, careldr4yrs, dadid, dadedu, dadlive, dadage, dadcantread, dadyrdied, momid, momedu, momlive, momage, momcantread, momyrdied, headid, headedu, headage, headsex, headrel, hysize, male05, male612, male1317, male1860, male61, female05, female612, female1317, female1860, female61, wi_new as wi, hq_new as hq, sv_new as sv, cd_new as cd, drwaterq_new as drwaterq, toiletq_new as toiletq, elecq_new as elecq, cookingq_new as cookingq, aniany, animilk, anidrau, anirumi, anicowm, anicowt, anicalv, anibufm, anibull, anihebu, anidonk, anishee, anigoat, anipigs, anipoul, anirabb, anispec, aniothr, ownlandhse, ownhouse, molisa06, molisa09, molisa10, molisa11, molisa12, molisa13, molisa14, molisa15, molisa16, credit, foodsec, shcrime1, shcrime2, shcrime3, shcrime4, shcrime5, shcrime6, shcrime7, shregul1, shregul2, shregul3, shregul5, shecon1, shecon2, shecon3, shecon4, shecon5, shecon6, shecon7, shecon8, shecon9, shecon10, shecon11, shecon12, shecon13, shecon14, shenv1, shenv2, shenv3, shenv4, shenv5, shenv6, shenv7, shenv8, shenv9, shhouse1, shhouse2, shhouse3, shfam1, shfam2, shfam3, shfam4, shfam5, shfam6, shfam7, shfam8, shfam9, shfam10, shfam11, shfam12, shfam13, shfam14, shfam18, shother, shenv13
FROM dbo.vietnam_constructed;

```

- Data Error checks – delete unnecessary records as part of data cleaning

```

/*Delete unwanted records from the INDIA - dataset*/
-----  

DELETE FROM dbCare.INPEVIETH  

WHERE childid = 'childid';  

--(2 rows affected)  

--Completion time: 2021-03-16T22:30:49.0138037+00:00
-----  


```

- Create Master Data table - Region

```

USE CP001; GO  

-- Create Master Data - Region  

CREATE TABLE Region ( regionid INT PRIMARY KEY, region varchar(50) )  

-- Insert into MD - Region  

INSERT INTO [CP001].[dbCare].[Region](regionid, region) VALUES (1 , 'Tigray'), (2 , 'Afar'), (3 , 'Amhara'), (4 , 'Oromiya'), (5 , 'Somali'), (6 , 'Benshangul Gumz'), (7 , 'SNNP'), (12 , 'Gambela'), (13 , 'Harari'), (14 , 'Addis Ababa City Administration'), (15 , 'Dire Dawa City Administration'), (21 , 'Coastal Andhra'), (22 , 'Rayalseema'), (23 , 'Telangana'), (24 , 'Others'), (31 , 'Costa'), (32 , 'Sierra'), (33 , 'Selva'), (51 , 'Northern Uplands'), (52 , 'Red River Delta'), (53 , 'Phu Yen'), (54 , 'Da Nang'), (55 , 'Highlands'), (56 , 'South Eastern'), (57 , 'Mekong River Delta'), (58 , 'Other'), (77 , 'Not known'), /* India */ (88 , 'Not known') /* Peru */ ;

```

- Create Master Data table - Language

```
USE CP001; GO

-- Create Master Data table - Language

CREATE TABLE Language ( chlangid varchar(6) PRIMARY KEY, chlang varchar(50) )

-- Insert into MD table - Region

INSERT INTO [CP001].[dbCare].[Language](chlangid, chlang) VALUES ('ET_001' , 'Afarigna'), ('ET_010' , 'Kembategna'),
('ET_011' , 'Oromifa'), ('ET_012' , 'Sidamigna'), ('ET_013' , 'Siltigna'), ('ET_014' , 'Somaligna'), ('ET_015' , 'Tigrigna'),
('ET_016' , 'Welayitegna'), ('ET_017' , 'Zayigna'), ('ET_002' , 'Amarigna'), ('ET_020' , 'Other'), ('ET_003' , 'Agewigna'),
('ET_004' , 'Dawerogna'), ('ET_005' , 'Gedeognna'), ('ET_006' , 'Guraghigna'), ('ET_007' , 'Hadiyigna'), ('ET_077' , 'Not Known'),
('ET_079' , 'Refused To Answer'), ('ET_008' , 'Harari'), ('ET_088' , 'N/A'), ('ET_009' , 'Kefigna'), ('ET_099' ,
'Missing'), ('IN_010' , 'Other'), ('IN_021' , 'Telugu'), ('IN_022' , 'Hindi'), ('IN_023' , 'Urdu'), ('IN_024' , 'Oria'),
('IN_025' , 'Kannada'), ('IN_026' , 'Marati'), ('IN_027' , 'Tamil'), ('IN_028' , 'Local Dialect'), ('IN_077' , 'Not Known'),
('IN_079' , 'Refused To Answer'), ('IN_088' , 'N/A'), ('IN_099' , 'Missing'), ('PE_010' , 'Other'), ('PE_031' , 'Spanish'),
('PE_032' , 'Quechua'), ('PE_033' , 'Aymara'), ('PE_034' , 'Native From Jungle'), ('PE_035' , 'Spanish & Quechua'), ('PE_036' ,
'Spanish & Aymara'), ('PE_037' , 'Nomatsiguenga (Native Language In Jungle)'), ('PE_050' , 'Mute Or Difficulty Speaking'),
('PE_077' , 'Not Known'), ('PE_079' , 'Refused To Answer'), ('PE_088' , 'N/A'), ('PE_099' , 'Missing'), ('VN_010' , 'Other'),
('VN_041' , 'Vietnamese'), ('VN_042' , 'Chinese'), ('VN_043' , 'Tay'), ('VN_044' , 'H'Mong'), ('VN_045' , 'Nung'), ('VN_046' ,
'Ede'), ('VN_047' , 'Thai'), ('VN_048' , 'Dao'), ('VN_049' , 'Giay'), ('VN_077' , 'Not Known'), ('VN_079' , 'Refused To Answer'),
('VN_088' , 'N/A'), ('VN_099' , 'Missing'), ('ET_NaN' , 'Missing'), ('IN_NaN' , 'Missing'), ('PE_NaN' , 'Missing'),
('VN_NaN' , 'Missing');
```

- Create Master Data table - Religion

```
USE CP001; GO

-- Create Master Data table - Religion

CREATE TABLE Religion ( chdrelid INT PRIMARY KEY, chdrel varchar(50) )

-- Insert into MD table - Religion

INSERT INTO [CP001].[dbCare].[Religion](chdrelid, chdrel) VALUES (1 , 'Christian'), (2 , 'Muslim'), (3 ,
'Buddhist'), (4 , 'Hindu'), (5 , 'Catholic'), (6 , 'Protestant'), (7 , 'Orthodox'), (8 , 'Sikh'), (9 ,
'Evangelist'), (10 , 'Mormon'), (11 , 'Ancestor Worship'), (12 , 'Hao Hao'), (13 , 'Cao Dai'), (14 , 'None'),
(15 , 'Other'), (99 , 'Not Known');

;
```

- Create Master Data table - Disability\_Scale

```
USE CP001; GO

-- Create Master Data table - [Disability_Scale]
CREATE TABLE Disability_Scale ( chdisscaleid INT PRIMARY KEY, chdisscale varchar(100) )

-- Insert into MD table - [Disability_Scale]
INSERT INTO [CP001].[dbCare].[Disability_Scale](chdisscaleid, chdisscale) VALUES (0 , 'Able to work same as others of this age'), (1 , 'Capable of most types of full-time work but some difficulty with physical work'), (2 , 'Able to work full-time but only work requiring no physical activity'), (3 , 'Can only do light work on a part-time basis'), (4 , 'Cannot work but able to care for themselves (e.g. dress themselves, etc.'), (5 , 'Cannot work and needs help with daily activities such as dressing, washing, etc.'), (6 , 'Other'), (99 , 'Missing');
```

- Create Master Data table - Ethnicity

```
USE CP001; GO

-- Create Master Data table - Ethnicity

CREATE TABLE Ethnicity ( chethnicid varchar(6) PRIMARY KEY, chethnic varchar(50) )

-- Insert into MD table - Ethnicity

INSERT INTO [CP001].[dbCare].[Ethnicity](chethnicid, chethnic) VALUES ('ET_010' , 'Other'), ('ET_011' , 'Agew'), ('ET_012' ,
'Amhara'), ('ET_013' , 'Gurage'), ('ET_014' , 'Hadiva'), ('ET_015' , 'Kambata'), ('ET_016' , 'Oromo'), ('ET_017' ,
'Sidama'), ('ET_018' , 'Tigrian'), ('ET_019' , 'Wolavta'), ('ET_099' , 'NK'), ('IN_100' , 'Other, Hindu'), ('IN_101' ,
'Other, Muslim'), ('IN_102' , 'Other, Buddhist'), ('IN_104' , 'Other, Christian'), ('IN_014' , 'None'), ('IN_021' ,
'SC'), ('IN_022' , 'ST'), ('IN_023' , 'BC'), ('IN_024' , 'Other'), ('IN_099' , 'NK'), ('PE_031' , 'White'), ('PE_032' ,
'Mestizo'), ('PE_033' , 'Native of the Amazon'), ('PE_034' , 'Negro'), ('PE_035' , 'Asiatic'), ('VN_010' , 'Other'), ('VN_041' ,
'Kinh'), ('VN_042' , 'H'Mong'), ('VN_043' , 'Cham'), ('VN_044' , 'Ede'), ('VN_045' , 'Bana'), ('VN_046' ,
'Nung'), ('VN_047' , 'Tay'), ('VN_048' , 'Dao'), ('VN_099' , 'NK'), ('ET_NaN' , 'Missing'), ('IN_NaN' , 'Missing'),
('PE_NaN' , 'Missing'), ('VN_NaN' , 'Missing');
```

- Create Master Data table – Current\_Grade

```
USE CP001; GO

-- Create Master Data table - Current_Grade

CREATE TABLE Current_Grade ( engradeid varchar(6) PRIMARY KEY, engrade varchar(150) )

-- Insert into MD table - Current_Grade

INSERT INTO [CP001].[dbCare].[Current_Grade](engradeid, engrade) VALUES ('ET_000', 'None'), ('ET_001', 'Grade 01'), ('ET_010', 'Grade 10'), ('ET_100', 'Pre-primary'), ('ET_011', 'Grade 11 (Secondary Second Cycle Preparatory Programme)'), ('ET_012', 'Grade 12 (Secondary Second Cycle Preparatory Programme)'), ('ET_013', 'First cycle of primary teaching certificate (grade 1-4)/1st year'), ('ET_014', 'First cycle of primary teaching certificate (grade 1-4)/2nd year'), ('ET_015', 'Second cycle of primary teaching certificate (grades 5-8)/1st year'), ('ET_016', 'Second cycle of primary teaching certificate (grades 5-8)/2nd year'), ('ET_021', 'TVET/1st year/level (include diplomas such as accounting diploma)'), ('ET_022', 'TVET/2nd year/level (include diplomas such as accounting diploma)'), ('ET_023', 'TVET/3rd year/level (include diplomas such as accounting diploma)'), ('ET_024', 'TVET/4th year/level (include diplomas such as accounting diploma)'), ('ET_025', 'Secondary education, teacher (diploma holder)/1st year'), ('ET_026', 'Secondary education, teacher (diploma holder)/2nd year'), ('ET_027', 'Secondary education, teacher (bachelor's degree holder and above)/1st year'), ('ET_028', 'Secondary education, teacher (bachelor's degree holder and above)/3rd year'), ('ET_029', 'Secondary education, teacher (bachelor's degree holder and above)/2nd year'), ('ET_031', 'Preschool teacher certificate (6 months to one year)'), ('ET_032', 'Undergraduate degree (1st year, regular (R))'), ('ET_033', 'Undergraduate degree (2nd year, regular (R))'), ('ET_034', 'Undergraduate degree (3rd year, regular (R))'), ('ET_035', 'Undergraduate degree (4th year, regular (R))'), ('ET_036', 'Undergraduate degree (1st year or equivalent, non-regular (NR), summer/distant/ evening/weekend student)'), ('ET_037', 'Undergraduate degree (2nd year or equivalent, non-regular (NR), summer/distant/ evening/weekend student)'), ('ET_038', 'Undergraduate degree (3rd year or equivalent, non-regular (NR), summer/distant/ evening/weekend student)'), ('ET_039', 'Undergraduate degree (4th year or equivalent, non-regular (NR), summer/distant/ evening/weekend student)'), ('ET_040', 'Masters or doctoral at university'), ('ET_041', 'Others'), ('ET_005', 'Grade 05'), ('ET_006', 'Grade 06'), ('ET_007', 'Grade 07'), ('ET_077', 'Not Known'), ('Refused to answer'), ('ET_079', 'N/A'), ('ET_009', 'Grade 09'), ('IN_000', 'None'), ('IN_001', 'Grade 01'), ('IN_010', 'Grade 10'), ('IN_009', 'Pre-primary'), ('IN_011', 'Grade 11'), ('IN_012', 'Grade 12'), ('IN_013', 'Post-secondary / technological institute'), ('IN_014', 'Vocational'), ('IN_015', 'University degree (graduate)'), ('IN_016', 'University degree (postgraduate)'), ('IN_022', 'Grade 02'), ('IN_028', 'Adult literacy'), ('IN_029', 'Religious education'), ('IN_003', 'Grade 03'), ('IN_004', 'Grade 04'), ('IN_005', 'Grade 05'), ('IN_006', 'Grade 06'), ('IN_007', 'Grade 07'), ('IN_077', 'Not Known'), ('IN_008', 'Grade 08'), ('IN_009', 'Grade 09'), ('PE_001', 'grade 1 (Primary, Grade 1)'), ('PE_010', 'grade 10 (Secondary, Year 4)'), ('PE_100', 'Pre-primary'), ('PE_011', 'grade 11 (Secondary, Year 5)'), ('PE_013', 'Incomplete technical or pedagogical institute'), ('PE_014', 'Complete technical or pedagogical institute'), ('PE_015', 'Incomplete university'), ('PE_016', 'Complete university'), ('PE_017', 'Adult literacy'), ('PE_018', 'Other (specify)'), ('PE_019', 'Masters or doctoral at university'), ('PE_002', 'grade 2 (Primary, Grade 2)'), ('PE_020', 'Some form of formal or informal preschool'), ('PE_021', 'Incomplete Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO'), ('PE_022', 'Complete Cent. Tecnico Productivo CETPRO/ Cent. Edu. Ocupacional CEO'), ('PE_003', 'grade 3 (Primary, Grade 3)'), ('PE_004', 'grade 4 (Primary, Grade 4)'), ('PE_005', 'grade 5 (Primary, Grade 5)'), ('PE_006', 'grade 6 (Primary, Grade 6)'), ('PE_007', 'grade 7 (Secondary, Year 1)'), ('PE_077', 'Not Known'), ('PE_079', 'N/A'), ('PE_008', 'grade 8 (Secondary, Year 2)'), ('PE_088', 'N/A'), ('PE_009', 'grade 9 (Secondary, Year 3)'), ('VN_000', 'None'), ('VN_001', 'Grade 1'), ('VN_010', 'Grade 10'), ('VN_011', 'Grade 11'), ('VN_012', 'Grade 12'), ('VN_013', 'Short term Vocational Training'), ('VN_014', 'Vocational Secondary School (1st year)'), ('VN_015', 'Vocational Secondary School (2nd year)'), ('VN_016', 'Vocational Secondary School completion'), ('VN_017', 'Professional Secondary (1st years)'), ('VN_018', 'Professional Secondary (2nd years)'), ('VN_019', 'Professional Secondary (3rd years)'), ('VN_002', 'Grade 2'), ('VN_020', 'Professional Secondary completion'), ('VN_021', 'Vocational College (1st year)'), ('VN_022', 'Vocational College (2nd year)'), ('VN_023', 'Vocational college completion'), ('VN_024', 'College education (1st year)'), ('VN_025', 'College education (2nd year)'), ('VN_026', 'College education completion'), ('VN_027', 'In the job, evening/weekend college education'), ('VN_028', 'In the job, evening/weekend undergraduate in university'), ('VN_029', 'University education (undergraduate 1st year)'), ('VN_030', 'University education (undergraduate 2nd year)'), ('VN_031', 'University education (undergraduate 3rd year)'), ('VN_032', 'University education (undergraduate 4th year)'), ('VN_033', 'University education (undergraduate 5th year)'), ('VN_034', 'University education completion'), ('VN_035', 'Post-graduate education'), ('VN_036', 'Post-graduate completion'), ('VN_037', 'Centre for continued education (non-formal student)'), ('VN_038', 'Other, (specify)'), ('VN_004', 'Grade 4'), ('VN_005', 'Grade 5'), ('VN_050', 'Any pre-primary grade'), ('VN_006', 'Grade 6'), ('VN_007', 'Grade 7'), ('VN_077', 'Not Known'), ('VN_008', 'Grade 8'), ('VN_009', 'Grade 9'), ('ET_NaN', 'Missing'), ('IN_NaN', 'Missing'), ('PE_NaN', 'Missing'), ('VN_NaN', 'Missing');
```

- Create Master Data table – School\_Type

```
USE CP001; GO

-- Create Master Data table - School_Type

CREATE TABLE School_Type ( entypeid varchar(6) PRIMARY KEY, entype varchar(150) )

-- Insert into MD table - School_Type

INSERT INTO [CP001].[dbCare].[School_Type](entypeid, entype) VALUES ('ET_001', 'Private'), ('ET_002', 'Public (part student fees, part government funded)'), ('ET_003', 'Community (NGO/Charity/Religious)'), ('ET_004', 'Government funded'), ('ET_005', 'Others, specify'), ('ET_077', 'Not Known'), ('ET_079', 'Refused to answer'), ('ET_088', 'N/A'), ('IN_001', 'Private'), ('IN_010', 'vn=main school'), ('IN_011', 'Other'), ('IN_002', 'NGO/Charity/ Religious (not for profit)'), ('IN_003', 'public (government)'), ('IN_004', 'informal or non-formal community (e.g. mother's cooperative)'), ('IN_005', 'vocational school'), ('IN_006', 'in=charitable trust'), ('IN_007', 'india=bridge school'), ('IN_077', 'Not Known'), ('IN_008', 'mix of public and private'), ('IN_008', 'N/A'), ('IN_009', 'vn=branch school'), ('IN_099', 'missing'), ('PE_001', 'Private'), ('PE_002', 'Public'), ('PE_003', 'Half public/half private'), ('PE_004', 'Other'), ('PE_088', 'N/A'), ('VN_001', 'Private'), ('VN_002', 'NGO/Charity/ Religious (not for profit)'), ('VN_003', 'Public'), ('VN_005', 'Others, specify'), ('VN_006', 'Informal'), ('VN_007', 'Half public/half private'), ('VN_077', 'Not Known'), ('VN_008', 'Centre for continuing education'), ('VN_088', 'N/A'), ('ET_NaN', 'Missing'), ('IN_NaN', 'Missing'), ('PE_NaN', 'Missing'), ('VN_NaN', 'Missing');
```

- Create Master Data table – Highest\_Grade

```
USE CP001; GO

-- Create Master Data table - Highest_Grade

CREATE TABLE Highest_Grade ( hghgradeid varchar(6) PRIMARY KEY, hghgrade varchar(100) )

-- Insert into MD table - Highest_Grade

INSERT INTO [CP001].[dbCare].[Highest_Grade](hghgradeid, hghgrade) VALUES ('ET_000', 'None'), ('ET_001'
, 'Grade 1'), ('ET_002', 'Grade 2'), ('ET_003', 'Grade 3'), ('ET_004', 'Grade 4'), ('ET_005', 'Grade
5'), ('ET_006', 'Grade 6'), ('ET_007', 'Grade 7'), ('ET_008', 'Grade 8'), ('ET_009', 'Grade 9'),
('ET_010', 'Grade 10'), ('ET_011', 'Grade 11'), ('ET_012', 'Grade 12'), ('ET_013', 'Post-secondary,
vocational'), ('ET_014', 'University'), ('ET_015', 'Masters, doctorate'), ('ET_028', 'Adult
literacy'), ('ET_029', 'Religious education'), ('ET_030', 'Other'), ('IN_000', 'None'), ('IN_001',
'Grade 1'), ('IN_002', 'Grade 2'), ('IN_003', 'Grade 3'), ('IN_004', 'Grade 4'), ('IN_005', 'Grade
5'), ('IN_006', 'Grade 6'), ('IN_007', 'Grade 7'), ('IN_008', 'Grade 8'), ('IN_009', 'Grade 9'),
('IN_010', 'Grade 10'), ('IN_011', 'Grade 11'), ('IN_012', 'Grade 12'), ('IN_013', 'Vocational,
technical college'), ('IN_014', 'University'), ('IN_015', 'Masters, doctorate'), ('IN_028', 'Adult
literacy'), ('IN_029', 'Religious education'), ('IN_030', 'Other'), ('VN_000', 'None'), ('VN_001',
'Grade 1'), ('VN_002', 'Grade 2'), ('VN_003', 'Grade 3'), ('VN_004', 'Grade 4'), ('VN_005', 'Grade
5'), ('VN_006', 'Grade 6'), ('VN_007', 'Grade 7'), ('VN_008', 'Grade 8'), ('VN_009', 'Grade 9'),
('VN_010', 'Grade 10'), ('VN_011', 'Grade 11'), ('VN_012', 'Grade 12'), ('VN_013', 'Post-secondary,
vocational'), ('VN_014', 'University'), ('VN_015', 'Masters, doctorate'), ('VN_028', 'Adult
literacy'), ('VN_029', 'Religious education'), ('VN_030', 'Other'), ('PE_000', 'None'), ('PE_001',
'Grade 1'), ('PE_002', 'Grade 2'), ('PE_003', 'Grade 3'), ('PE_004', 'Grade 4'), ('PE_005', 'Grade
5'), ('PE_006', 'Grade 6'), ('PE_007', 'Grade 7'), ('PE_008', 'Grade 8'), ('PE_009', 'Grade 9'),
('PE_010', 'Grade 10'), ('PE_011', 'Grade 11'), ('PE_013', 'Technical, pedagogical, CETPRO
(incomplete)'), ('PE_014', 'Technical, pedagogical, CETPRO (complete)'), ('PE_015', 'University
(incomplete)'), ('PE_016', 'University (complete)'), ('PE_028', 'Adult literacy'), ('PE_030',
'Other'), ('ET_NaN', 'Missing'), ('IN_NaN', 'Missing'), ('PE_NaN', 'Missing'), ('VN_NaN', 'Missing');
```

- Create Master Data table - Relationship

```
USE CP001; GO

-- Create Master Data table - Relationship

CREATE TABLE Relationship ( relid INT PRIMARY KEY, relationship varchar(100) )

-- Insert into MD table - Relationship

INSERT INTO [CP001].[dbCare].[Relationship](relid, relationship) VALUES (0 , 'YL child'), (1 , 'Biological
parent'), (2 , 'Non-biological parent'), (3 , 'Grandparent'), (4 , 'Uncle/aunt'), (5 , 'Sibling'), (6 ,
'Other-relative'), (7 , 'Other-nonrelative'), (8 , 'Partner/spouse of YL child'), (9 , 'Father-in-law/mother-
in-law');
```

- Data Cleaning task – to ensure consistency in MD and TD during normalization

```

USE CP001;
GO

/*Mom year died and Dad year died - Data cleaning*/

UPDATE dbCare.INPEVIETH SET dadyrdied = '1988' WHERE dadyrdied = '88' AND chlidid =
'VN071028';
UPDATE dbCare.INPEVIETH SET momyrdied = '2002' WHERE momyrdied = '2' AND chlidid =
'IN061045';

/*Update Combination IDs from various different tables to be able to create
relationships for TD with MD*/

UPDATE dbCare.INPEVIETH SET chethnic = 'NaN' WHERE LEN(chethnic) = 0;
UPDATE dbCare.INPEVIETH SET chethnic = '00' + chethnic WHERE LEN(chethnic) = 1;
UPDATE dbCare.INPEVIETH SET chethnic = '0' + chethnic WHERE LEN(chethnic) = 2;

UPDATE dbCare.INPEVIETH SET chlang = 'NaN' WHERE LEN(chlang) = 0;
UPDATE dbCare.INPEVIETH SET chlang = '00' + chlang WHERE LEN(chlang) = 1;
UPDATE dbCare.INPEVIETH SET chlang = '0' + chlang WHERE LEN(chlang) = 2;

UPDATE dbCare.INPEVIETH SET engrade = 'NaN' WHERE LEN(engrade) = 0;
UPDATE dbCare.INPEVIETH SET engrade = '00' + engrade WHERE LEN(engrade) = 1;
UPDATE dbCare.INPEVIETH SET engrade = '0' + engrade WHERE LEN(engrade) = 2;

UPDATE dbCare.INPEVIETH SET entype = 'NaN' WHERE LEN(entype) = 0;
UPDATE dbCare.INPEVIETH SET entype = '00' + entype WHERE LEN(entype) = 1;
UPDATE dbCare.INPEVIETH SET entype = '0' + entype WHERE LEN(entype) = 2;

UPDATE dbCare.INPEVIETH SET chldrel = '99' WHERE LEN(LTRIM(RTRIM(chldrel))) = 0;
UPDATE dbCare.INPEVIETH SET region = '88' WHERE LEN(LTRIM(RTRIM(region))) = 0;
UPDATE dbCare.INPEVIETH SET chdisscale = '99' WHERE LEN(LTRIM(RTRIM(chdisscale))) = 0;

/* Education Grades for all - based on Countries */

UPDATE dbCare.INPEVIETH SET hghgrade = 'NaN' WHERE LEN(hghgrade) = 0;
UPDATE dbCare.INPEVIETH SET hghgrade = '00' + hghgrade WHERE LEN(hghgrade) = 1;
UPDATE dbCare.INPEVIETH SET hghgrade = '0' + hghgrade WHERE LEN(hghgrade) = 2;

UPDATE dbCare.INPEVIETH SET caredu = 'NaN' WHERE LEN(caredu) = 0;
UPDATE dbCare.INPEVIETH SET caredu = '00' + caredu WHERE LEN(caredu) = 1;
UPDATE dbCare.INPEVIETH SET caredu = '0' + caredu WHERE LEN(caredu) = 2;

UPDATE dbCare.INPEVIETH SET dadedu = 'NaN' WHERE LEN(dadedu) = 0;
UPDATE dbCare.INPEVIETH SET dadedu = '00' + dadedu WHERE LEN(dadedu) = 1;
UPDATE dbCare.INPEVIETH SET dadedu = '0' + dadedu WHERE LEN(dadedu) = 2;

UPDATE dbCare.INPEVIETH SET momedu = 'NaN' WHERE LEN(momedu) = 0;
UPDATE dbCare.INPEVIETH SET momedu = '00' + momedu WHERE LEN(momedu) = 1;
UPDATE dbCare.INPEVIETH SET momedu = '0' + momedu WHERE LEN(momedu) = 2;

UPDATE dbCare.INPEVIETH SET headedu = 'NaN' WHERE LEN(headedu) = 0;
UPDATE dbCare.INPEVIETH SET headedu = '00' + headedu WHERE LEN(headedu) = 1;
UPDATE dbCare.INPEVIETH SET headedu = '0' + headedu WHERE LEN(headedu) = 2;

```

- Create TD table – ch\_Identity

```

USE CP001; GO

-- Create Transactional Data table - ch_Identity

CREATE TABLE ch_Identity (
    child_round_id VARCHAR(10) PRIMARY KEY ,childdid VARCHAR(8) NOT NULL
,clustid VARCHAR(2) NOT NULL ,commid VARCHAR(10) ,typesite VARCHAR(10) ,regionid INT ,countryid
VARCHAR(2) NOT NULL ,dint DATE ,[round] INT ,yc VARCHAR(2) ,childloc VARCHAR(10) ,inround VARCHAR(10)
,panel VARCHAR(10) ,deceased VARCHAR(10)

,chsex VARCHAR(10) ,chlangid VARCHAR(6) ,chethnidid VARCHAR(6) ,chldrelid INT ,agemon INT ,marrcohab
VARCHAR(10) ,marrcohab_age INT ,birth VARCHAR(10) ,birth_age INT

,chwheight FLOAT ,chheight FLOAT ,bmi FLOAT ,zwfa FLOAT ,zhfa FLOAT ,zbfa FLOAT ,zwfl FLOAT ,fwfa INT
,fhfa INT ,fbfa INT ,fwfl INT ,underweight VARCHAR(30) ,stunting VARCHAR(30) ,thinness VARCHAR(30)

,CONSTRAINT FK_chEthnidid FOREIGN KEY (chethnidid) REFERENCES [dbCare].[Ethnicity](chethnidid)
,CONSTRAINT FK_chLangid FOREIGN KEY (chlangid) REFERENCES [dbCare].[Language](chlangid) ,CONSTRAINT
FK_regionid FOREIGN KEY (regionid) REFERENCES [dbCare].[Region](regionid) ,CONSTRAINT FK_chldRelid
FOREIGN KEY (chldrelid) REFERENCES [dbCare].[Religion](chldrelid) )

-- Insert into TD table - ch_Identity

BEGIN TRY

INSERT INTO [CP001].[dbCare].[ch_Identity] SELECT RTRIM(LTRIM([childdid])) + '_' +
RTRIM(LTRIM([round])) as child_round_id ,RTRIM(LTRIM([childdid])) as [childdid]
,substring(RTRIM(LTRIM([childdid])),3,2) as [clustid] ,RTRIM(LTRIM([commid])) as [commid] ,CASE
[typesite] WHEN '1' THEN 'Urban' WHEN '2' THEN 'Rural' WHEN '77' THEN 'Not known' ELSE [typesite] END
as [typesite] ,[region] as [regionid] ,substring(RTRIM(LTRIM([childdid])),1,2) as [countryid] ,dint
,RTRIM(LTRIM([round])) as [round] ,CASE [yc] WHEN '0' THEN 'OC' WHEN '1' THEN 'YC' ELSE [yc] END as
[yc] ,CASE [childloc] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [childloc] END as [childloc] ,CASE
[inround] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [inround] END as [inround] ,CASE [panel] WHEN
'0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [panel] END as [panel] ,CASE [deceased] WHEN '1' THEN 'Yes'
ELSE [deceased] END as [deceased]

,CASE [chsex] WHEN '1' THEN 'Male' WHEN '2' THEN 'Female' ELSE [chsex] END as [chsex]

,substring(RTRIM(LTRIM([childdid])),1,2) + '_' + [chlang] as [chlangid] -- Here we still get ET_IN_
issues - to be Updated later ,substring(RTRIM(LTRIM([childdid])),1,2) + '_' + [chethnic] as
[chethnidid] -- Here we still get ET_IN_issues - to be Updated later ,[chldrel] as [chldrelid]

,CASE [agemon] WHEN '-9999' THEN 'Missing' ELSE [agemon] END as [agemon]

,CASE [marrcohab] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [marrcohab] END as [marrcohab]

,[marrcohab_age]

,CASE [birth] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [birth] END as [birth]

,[birth_age]

,CASE CAST([chwheight] as FLOAT(1)) WHEN '-9999' THEN NULL -- 'Not known' is variable and cannot be
passed into Float type column ELSE CAST([chwheight] as FLOAT(1)) END as [chwheight]

,CASE CAST([chheight] as FLOAT(1)) WHEN '-9999' THEN NULL -- 'Not known' is variable and cannot be
passed into Float type column ELSE CAST([chheight] as FLOAT(1)) END as [chheight]

,CAST(bmi as FLOAT(1)) as bmi ,CAST(zwfa as FLOAT(1)) as zwfa ,CAST(zhfa as FLOAT(1)) as zhfa
,CAST(zbfa as FLOAT(1)) as zbfa ,CAST(zwfl as FLOAT(1)) as zwfl ,fwfa ,fhfa ,fbfa ,fwfl

,CASE [underweight] WHEN '0' THEN 'Not underweight' WHEN '1' THEN 'Moderately underweight' WHEN '2'
THEN 'Severely underweight' ELSE [underweight] END as [underweight]

,CASE [stunting] WHEN '0' THEN 'Not stunted' WHEN '1' THEN 'Moderately stunted' WHEN '2' THEN
'Severely stunted' ELSE [stunting] END as [stunting]

,CASE [thinness] WHEN '0' THEN 'Not thin' WHEN '1' THEN 'Moderately thin' WHEN '2' THEN 'Severely
thin' ELSE [thinness] END as [thinness]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine
,ERROR_NUMBER() AS [Error Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error
State] END CATCH ;

```

- Create TD table – ch\_MedicalHistory

```

USE CP001; GO

-- Create Transactional Data table- ch_MedicalHistory

CREATE TABLE ch_MedicalHistory ( chMed_id INT IDENTITY(1,1) PRIMARY KEY ,child_round_id VARCHAR(10) NOT NULL ,bcg
VARCHAR(10) ,bwdoc VARCHAR(10) ,bwght INT ,delivery VARCHAR(10) ,dpt VARCHAR(10) ,hib VARCHAR(10) ,measles VARCHAR(10)
,numante INT ,polio VARCHAR(10) ,tetanus VARCHAR(10) ,chdisability VARCHAR(10) ,chdisscaleid INT ,chhprob VARCHAR(10)
,chillness VARCHAR(10) ,chinjury VARCHAR(10) ,chmighthdie VARCHAR(10) ,chalcohol VARCHAR(10) ,chsmoke VARCHAR(30)
,crephealth1 INT ,crephealth2 VARCHAR(10) ,crephealth3 VARCHAR(10) ,crephealth4 VARCHAR(50) ,chhealth VARCHAR(10)
,chlrel VARCHAR(10) ,cladder INT

,CONSTRAINT FK_chRdid FOREIGN KEY (child_round_id) REFERENCES [dbCare].[ch_Identity](child_round_id) ,CONSTRAINT
FK_chdisscaleid FOREIGN KEY (chdisscaleid) REFERENCES [dbCare].[Disability_Scale](chdisscaleid) )

-- Insert into TD table - ch_MedicalHistory

BEGIN TRY

INSERT INTO [CP001].[dbCare].[ch_MedicalHistory] SELECT RTRIM(LTRIM([childid])) + '_' + RTRIM(LTRIM([round])) as
child_round_id

,CASE [bcg] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [bcg] END as [bcg]
,CASE [bwdoc] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [bwdoc] END as [bwdoc]
,CASE [bwght] WHEN '-9999' THEN NULL -- 'Not known' string can't be passed into numeric datatype ELSE [bwght] END as
[bwght]
,CASE [delivery] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [delivery] END as [delivery]
,CASE [dpt] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [dpt] END as [dpt]
,CASE [hib] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [hib] END as [hib]
,CASE [measles] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [measles] END as [measles]
,CASE [numante] WHEN '88' THEN NULL -- 'N/A' can't be passed in numeric type WHEN '99' THEN NULL ELSE [numante] END
as [numante]
,CASE [polio] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [polio] END as [polio]
,CASE [tetanus] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [tetanus] END as [tetanus]
,CASE [chdisability] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chdisability] END as [chdisability]
,[chdisscale] as [chdisscaleid]
,CASE [chhprob] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chhprob] END as [chhprob]
,CASE [chillness] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chillness] END as [chillness]
,CASE [chinjury] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chinjury] END as [chinjury]
,CASE [chmighthdie] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chmighthdie] END as [chmighthdie]
,CASE [chalcohol] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [chalcohol] END as [chalcohol]
,CASE [chsmoke] WHEN '1' THEN 'Every day' WHEN '2' THEN 'At least once a week' WHEN '3' THEN 'At least once a month'
WHEN '4' THEN 'Hardly ever' WHEN '5' THEN 'I never smoke cigarettes' ELSE [chsmoke] END as [chsmokeid]
,[crephealth1]
,CASE [crephealth2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [crephealth2] END as [crephealth2] ,CASE
[crephealth3] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [crephealth3] END as [crephealth3] ,CASE [crephealth4]
WHEN '1' THEN 'Shop or street vendor' WHEN '2' THEN 'Family planning services or health facility' WHEN '3' THEN
'Other' WHEN '4' THEN 'Doesn't know what is a condom or where to buy' ELSE [crephealth4] END as [crephealth4]
,CASE [chhealth] WHEN '1' THEN 'very poor' WHEN '2' THEN 'poor' WHEN '3' THEN 'average' WHEN '4' THEN 'good' WHEN '5'
THEN 'very good' ELSE [chhealth] END as [chhealth]
,CASE [chlrel] WHEN '1' THEN 'Same' WHEN '2' THEN 'Better' WHEN '3' THEN 'Worse' ELSE [chlrel] END as [chlrel]
,[cladder]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine ,ERROR_NUMBER() AS [Error
Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error State] END CATCH ;

```

- Create TD table – ch\_Education

```

USE CP001; GO

-- Create Transactional Data table- ch_Education

CREATE TABLE ch_Education ( chEdu_id INT IDENTITY(1,1) PRIMARY KEY ,child_round_id VARCHAR(10) NOT NULL ,agegr1 INT ,engradeid VARCHAR(6) ,enrol
VARCHAR(30) ,entypeid VARCHAR(6) ,hghgradeid VARCHAR(6) ,preprim VARCHAR(30) ,timesch INT ,commsch INT ,commwork INT ,hcare FLOAT
,hplay FLOAT ,hschool FLOAT ,hsleep FLOAT ,htudy FLOAT ,htask FLOAT ,hwork FLOAT ,levlread VARCHAR(50) ,levlwrit VARCHAR(50) ,literate VARCHAR(30)

,CONSTRAINT FK_chRdEduid FOREIGN KEY (child_round_id) REFERENCES [dbCare].[ch_Identity](child_round_id) ,CONSTRAINT FK_chengradeid FOREIGN KEY
(engradeid) REFERENCES [dbCare].[Current_Grade](engradeid) ,CONSTRAINT FK_chentypeid FOREIGN KEY (entypeid) REFERENCES
[dbCare].[School_Type](entypeid) ,CONSTRAINT FK_chhghgradeid FOREIGN KEY (hghgradeid) REFERENCES [dbCare].[Highest_Grade](hghgradeid) )

-- Insert into TD table - ch_Education

BEGIN TRY

INSERT INTO [CP001].[dbCare].[ch_Education] SELECT RTRIM(LTRIM([childid])) + '_' + RTRIM(LTRIM([round])) as child_round_id

,[agegr1]

,SUBSTRING(RTRIM(LTRIM([childid])),1,2) + '_' + [engrade] as [engradeid] ,CASE [enrol] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '99' THEN
'Missing' WHEN '77' THEN 'Not known' WHEN '88' THEN 'N/A' ELSE [enrol] END as [enrol]

,substring(RTRIM(LTRIM([childid])),1,2) + '_' + [entype] as [entypeid]

,substring(RTRIM(LTRIM([childid])),1,2) + '_' + [hghgrade] as [hghgradeid]

,CASE [preprim] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [preprim] END as [preprim]

,[timesch] ,[commsch] ,[commwork] ,[hcare] ,[hplay] ,[hschool] ,[hsleep] ,[htudy] ,[htask] ,[hwork]

,CASE [levlread] WHEN '1' THEN 'Can't read anything' WHEN '2' THEN 'Reads letters' WHEN '3' THEN 'Reads word' WHEN '4' THEN 'Reads sentence' ELSE
[levlread] END as [levlread] ,CASE [levlwrit] WHEN '1' THEN 'No' WHEN '2' THEN 'Yes with difficulty or errors' WHEN '3' THEN 'Yes without difficulty
or errors' ELSE [levlwrit] END as [levlwrit] ,CASE [literate] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [literate] END as [literate]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine ,ERROR_NUMBER() AS [Error Number] ,ERROR_SEVERITY() AS
[Error Severity] ,ERROR_STATE() AS [Error State] END CATCH;

```

- Create TD table – ch\_Guardian

```

USE CP001; GO

-- Create Transactional Data table- ch_Guardian

CREATE TABLE ch_Guardian ( chGuard_id INT IDENTITY(1,1) PRIMARY KEY ,child_round_id VARCHAR(10) NOT NULL ,careage INT
,carecantread VARCHAR(10) ,careduid VARCHAR(6) ,carehead VARCHAR(50) ,careid INT ,careladder INT ,careldr4yrs INT
,carerelid INT ,caresex VARCHAR(10) ,dadge INT ,dadcantread VARCHAR(10) ,dadeduid VARCHAR(6) ,dadid INT ,dadlive
VARCHAR(50) ,dadyrdied INT ,momage INT ,momcantread VARCHAR(10) ,momeduid VARCHAR(6) ,momid INT ,momlive
VARCHAR(50) ,mommyrdied INT

,CONSTRAINT FK_chGuardid FOREIGN KEY (child_round_id) REFERENCES [dbCare].[ch_Identity](child_round_id) ,CONSTRAINT
FK_chcareduid FOREIGN KEY (careduid) REFERENCES [dbCare].[Highest_Grade](hghgradeid) ,CONSTRAINT FK_chdadeduid FOREIGN
KEY (dadeduid) REFERENCES [dbCare].[Highest_Grade](hghgradeid) ,CONSTRAINT FK_chmomeduid FOREIGN KEY (momeduid)
REFERENCES [dbCare].[Highest_Grade](hghgradeid) ,CONSTRAINT FK_chcarerelid FOREIGN KEY (carerelid) REFERENCES
[dbCare].[Relationship] (relid) )

-- Insert into TD table - ch_Guardian

BEGIN TRY

INSERT INTO [CP001].[dbCare].[ch_Guardian] SELECT RTRIM(LTRIM([childid])) + '_' + RTRIM(LTRIM([round])) as
child_round_id

,CASE [careage] WHEN '-9999' THEN 'Not known' ELSE [careage] END as [careage] ,CASE [carecantread] WHEN '0' THEN 'No'
WHEN '1' THEN 'Yes' ELSE [carecantread] END as [carecantread] ,SUBSTRING(RTRIM(LTRIM([childid])),1,2) + '_' + [caredu]
as [careduid] ,CASE [carehead] WHEN '1' THEN 'Caregiver is household head' WHEN '2' THEN 'Caregiver is partner of
household head' WHEN '3' THEN 'Other' ELSE [carehead] END as [carehead]

,[careid] ,[careladder] ,[careldr4yrs] ,[carerel] as [carerelid] ,CASE [caresex] WHEN '1' THEN 'Male' WHEN '2' THEN
'Female' ELSE [caresex] END as [caresex] ,CASE [dadge] WHEN '-9999' THEN 'Not known' ELSE [dadge] END as [dadge]

,CASE [dadcantread] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [dadcantread] END as [dadcantread]
,SUBSTRING(RTRIM(LTRIM([childid])),1,2) + '_' + [dadedu] as [dadeduid] ,[dadid] ,CASE [dadlive] WHEN '1' THEN 'Lives in
the household' WHEN '2' THEN 'Does not live in household' WHEN '3' THEN 'Has died' ELSE [dadlive] END as [dadlive]
,[dadyrdied]

,CASE [momage] WHEN '-9999' THEN 'Not known' ELSE [momage] END as [momage] ,CASE [momcantread] WHEN '0' THEN 'No' WHEN
'1' THEN 'Yes' ELSE [momcantread] END as [momcantread]

,substring(RTRIM(LTRIM([childid])),1,2) + '_' + [momedu] as [momeduid] ,[momid]

,CASE [momlive] WHEN '1' THEN 'Lives in the household' WHEN '2' THEN 'Does not live in household' WHEN '3' THEN 'Has
died' ELSE [momlive] END as [mommyrdied]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine ,ERROR_NUMBER() AS [Error
Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error State] END CATCH ;

```

- Create TD table – hh\_Main

```

USE CP001; GO

-- Create Transactional Data table- hh_Main

CREATE TABLE hh_Main ( hh_Main_id INT IDENTITY(1,1) PRIMARY KEY , child_round_id VARCHAR(10) NOT NULL , headage INT ,
headeduid VARCHAR(6) , headid INT , headrelid INT , headsex VARCHAR(10) , female05 INT , female1317 INT , female1860 INT ,
female61 INT , female612 INT , hysize INT , male05 INT , male1317 INT , male1860 INT , male61 INT , male612 INT ,
ownhouse VARCHAR(10) , ownlandhse VARCHAR(10) , credit INT , foodsec VARCHAR(50) , cd FLOAT , cookingq VARCHAR(10) ,
drwaterq VARCHAR(10) , elecq VARCHAR(10) , hq FLOAT , sv FLOAT , toiletq VARCHAR(10) , wi FLOAT
,CONSTRAINT FK_hh_Mainid FOREIGN KEY (child_round_id) REFERENCES [dbCare].[ch_Identity](child_round_id) ,CONSTRAINT
FK_hh_headeduid FOREIGN KEY (headeduid) REFERENCES [dbCare].[Highest_Grade](hghgradeid) )

-- Insert into TD table - hh_Main

BEGIN TRY

INSERT INTO [CP001].[dbCare].[hh_Main] SELECT RTRIM(LTRIM([childdid])) + ' ' + RTRIM(LTRIM([round])) as child_round_id
,CASE [headage] WHEN '-9999' THEN 'Not known' ELSE [headage] END as [headage] ,SUBSTRING(RTRIM(LTRIM([childdid])),1,2) +
' ' + [headedu] as [headeduid] ,headid ,[headrel] as [headrelid] ,CASE [headsex] WHEN '1' THEN 'Male' WHEN '2' THEN
'Female' ELSE [headsex] END as [headsex]

,female05 , female612 , female1317 , female1860 , female61

,hysize

,male05 , male612 , male1317 , male1860 , male61 ,CASE [ownhouse] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[ownhouse] END as [ownhouse] ,CASE [ownlandhse] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [ownlandhse] END as
[ownlandhse]

,credit

,CASE [foodsec] WHEN '1' THEN 'We always eat enough of what we want' WHEN '2' THEN 'We eat enough but not always what
we would like' WHEN '3' THEN 'We sometimes do not eat enough' WHEN '4' THEN 'We frequently do not eat enough' ELSE
[foodsec] END as [foodsec]

,[cd] ,CASE [cookingq] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [cookingq] END as [cookingq]

,CASE [drwaterq] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [drwaterq] END as [drwaterq] ,CASE [elecq] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [elecq] END as [elecq] ,[hq] ,[sv] ,CASE [toiletq] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes'
ELSE [toiletq] END as [toiletq] ,[wi]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine ,ERROR_NUMBER() AS [Error
Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error State] END CATCH ;

```

- Create TD table – hh\_LivestockOwnership

```

USE CP001; GO

-- Create Transactional Data table- hh_LivestockOwnership

CREATE TABLE hh_LivestockOwnership ( hh_LivStkOwn_id INT IDENTITY(1,1) PRIMARY KEY ,child_round_id VARCHAR(10) NOT NULL ,
aniany VARCHAR(10) ,anibeeh INT ,anibufm INT ,anibuft INT ,anibull INT ,anicalv INT ,anicaml INT ,anicowm INT ,
anicowt INT ,anidonk INT ,anidrau INT ,anifish INT ,anifshr INT ,anigoat INT ,aniguin INT ,anihebu INT ,aniheif INT
,anillam INT ,animilk INT ,aniothr INT ,anioxen INT ,anipigs INT ,anipoul INT ,anirabb INT ,anirumi INT ,anishee
INT ,anishri INT ,anisnai INT ,anispec INT ,aniybul INT

CONSTRAINT FK_LivStkchilddid FOREIGN KEY (child_round_id) REFERENCES [CP001].[dbCare].[ch_Identity](child_round_id) )

-- Insert into TD table - hh_LivestockOwnership

BEGIN TRY

INSERT INTO [CP001].[dbCare].[hh_LivestockOwnership] SELECT RTRIM(LTRIM([childdid])) + ' ' + RTRIM(LTRIM([round])) as
child_round_id

,CASE [aniany] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [aniany] END as [aniany] ,[anibeeh] ,[anibufm] ,[anibuft]
,[anibull] ,[anicalv] ,[anicaml] ,[anicowm] ,[anicowt] ,[anidonk] ,[anidrau] ,[anifish] ,[anifshr] ,[anigoat]
,[aniguin] ,[anihebu] ,[aniheif] ,[anillam] ,[animilk] ,[aniothr] ,[anioxen] ,[anipigs] ,[anipoul] ,[anirabb]
,[anirumi] ,[anishee] ,[anishri] ,[anisnai] ,[anispec] ,[aniybul]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine ,ERROR_NUMBER() AS [Error Number]
,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error State] END CATCH ;

```

- Create TD table – hh\_PublicProgrammes

```

USE CP001; GO

-- Create Transactional Data table- hh_PublicProgrammes

CREATE TABLE hh_PublicProgrammes ( hh_PubProg_id INT IDENTITY(1,1) PRIMARY KEY ,child_round_id
VARCHAR(10) NOT NULL

, eap VARCHAR(5) , hep VARCHAR(5) , othprog VARCHAR(5) , psnp_ds VARCHAR(5) , psnp_pw VARCHAR(5) ,
resettled VARCHAR(5)

, ikp VARCHAR(5) , ikp_child VARCHAR(5) , nregs VARCHAR(5) , nregs_allow VARCHAR(5) , nregs_work
VARCHAR(5) , pds VARCHAR(5) , rajiv VARCHAR(5) , sabla VARCHAR(5) , sabla_yl VARCHAR(5)

, beca_yl VARCHAR(5) , bonograt VARCHAR(5) , insur_yl VARCHAR(5) , juntos VARCHAR(5) , minsya_yl
VARCHAR(5) , projoven_yl VARCHAR(5) , sisgrat_yl VARCHAR(5)

, molisa06 VARCHAR(5) , molisa09 VARCHAR(5) , molisa10 VARCHAR(5) , molisa11 VARCHAR(5) , molisa12
VARCHAR(5) , molisa13 VARCHAR(5) , molisa14 VARCHAR(5) , molisa15 VARCHAR(5) , molisa16 VARCHAR(5)

,CONSTRAINT FK_hh_PubProg_id FOREIGN KEY (child_round_id) REFERENCES
[dbCare].[ch_Identity](child_round_id) )

-- Insert into TD table - hh_PublicProgrammes

BEGIN TRY

INSERT INTO [CP001].[dbCare].[hh_PublicProgrammes] SELECT RTRIM(LTRIM([childid])) + '_' +
RTRIM(LTRIM([round])) as child_round_id

/*Ethiopia*/ , CASE [eap] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [eap] END as [eap] , CASE [hep]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [hep] END as [hep] , CASE [othprog] WHEN '0' THEN 'No'
WHEN '1' THEN 'Yes' ELSE [othprog] END as [othprog] , CASE [psnp_ds] WHEN '0' THEN 'No' WHEN '1' THEN
'Yes' ELSE [psnp_ds] END as [psnp_ds] , CASE [psnp_pw] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[psnp_pw] END as [psnp_pw] , CASE [resettled] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [resettled]
END as [resettled]

/*India*/ , CASE [ikp] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [ikp] END as [ikp] , CASE
[ikp_child] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [ikp_child] END as [ikp_child] , CASE [nregs]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [nregs] END as [nregs] , CASE [nregs_allow] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [nregs_allow] END as [nregs_allow] , CASE [nregs_work] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [nregs_work] END as [nregs_work] , CASE [pds] WHEN '0' THEN 'No' WHEN
'1' THEN 'Yes' ELSE [pds] END as [pds] , CASE [rajiv] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[rajiv] END as [rajiv] , CASE [sabla] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [sabla] END as
[sabla] , CASE [sabla_yl] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [sabla_yl] END as [sabla_yl]

/*Peru*/
, CASE [beca_yl] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [beca_yl] END as [beca_yl] ,CASE
[bonograt] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [bonograt] END as [bonograt] ,CASE [insur_yl]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [insur_yl] END as [insur_yl] ,CASE [juntos] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [juntos] END as [juntos] ,CASE [minsya_yl] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [minsya_yl] END as [minsya_yl] ,CASE [projoven_yl] WHEN '0' THEN 'No' WHEN '1' THEN
'Yes' ELSE [projoven_yl] END as [projoven_yl] ,CASE [sisgrat_yl] WHEN '0' THEN 'No' WHEN '1' THEN
'Yes' ELSE [sisgrat_yl] END as [sisgrat_yl]

/*Vietnam*/
, CASE [molisa06] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [molisa06] END as [molisa06] ,CASE
[molisa09] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [molisa09] END as [molisa09] ,CASE [molisa10]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [molisa10] END as [molisa10] ,CASE [molisa11] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [molisa11] END as [molisa11] ,CASE [molisa12] WHEN '0' THEN 'No'
WHEN '1' THEN 'Yes' ELSE [molisa12] END as [molisa12] ,CASE [molisa13] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [molisa13] END as [molisa13] ,CASE [molisa14] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes'
ELSE [molisa14] END as [molisa14] ,CASE [molisa15] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[molisa15] END as [molisa15] ,CASE [molisa16] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [molisa16]
END as [molisa16]

FROM CP001.dbCare.INPEVIETH

END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS ErrorLine
,ERROR_NUMBER() AS [Error Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error
State] END CATCH ;

```

- Create TD table – hh\_Shocks

```

USE CP001; GO -- Create Transactional Data table- hh_Shocks
CREATE TABLE hh_Shocks ( shockid BIGINT IDENTITY(1,1) PRIMARY KEY , child_round_id VARCHAR(10) NOT NULL ,
shcrime1 VARCHAR(10) , shcrime2 VARCHAR(10) , shcrime3 VARCHAR(10) , shcrime4 VARCHAR(10) , shcrime5
VARCHAR(10) , shcrime6 VARCHAR(10) , shcrime7 VARCHAR(10) , shcrime8 VARCHAR(10) , shregul1 VARCHAR(10) ,
shregul2 VARCHAR(10) , shregul3 VARCHAR(10) , shregul4 VARCHAR(10) , shregul5 VARCHAR(10) , shregul6
VARCHAR(10) , shecon1 VARCHAR(10) , shecon2 VARCHAR(10) , shecon3 VARCHAR(10) , shecon4 VARCHAR(10) , shecon5
VARCHAR(10) , shecon6 VARCHAR(10) , shecon7 VARCHAR(10) , shecon8 VARCHAR(10) , shecon9 VARCHAR(10) ,
shecon10 VARCHAR(10) , shecon11 VARCHAR(10) , shecon12 VARCHAR(10) , shecon13 VARCHAR(10) , shecon14
VARCHAR(10) , shenv1 VARCHAR(10) , shenv2 VARCHAR(10) , shenv3 VARCHAR(10) , shenv4 VARCHAR(10) , shenv5
VARCHAR(10) , shenv6 VARCHAR(10) , shenv7 VARCHAR(10) , shenv8 VARCHAR(10) , shenv9 VARCHAR(10) , shenv10
VARCHAR(10) , shenv11 VARCHAR(10) , shenv12 VARCHAR(10) , shenv13 VARCHAR(10) , shhouse1 VARCHAR(10) ,
shhouse2 VARCHAR(10) , shhouse3 VARCHAR(10) , shfam1 VARCHAR(10) , shfam2 VARCHAR(10) , shfam3 VARCHAR(10) ,
shfam4 VARCHAR(10) , shfam5 VARCHAR(10) , shfam6 VARCHAR(10) , shfam7 VARCHAR(10) , shfam8 VARCHAR(10) ,
shfam9 VARCHAR(10) , shfam10 VARCHAR(10) , shfam11 VARCHAR(10) , shfam12 VARCHAR(10) , shfam13 VARCHAR(10) ,
shfam14 VARCHAR(10) , shfam15 VARCHAR(10) , shfam16 VARCHAR(10) , shfam17 VARCHAR(10) , shfam18 VARCHAR(10) ,
shother VARCHAR(10) , CONSTRAINT FK_Shockschildid FOREIGN KEY(child_round_id) REFERENCES
[CP001].[dbCare].[ch_Identity](child_round_id) );
-- Insert into TD table - hh_Shocks
BEGIN TRY INSERT INTO [CP001].[dbCare].[hh_Shocks] SELECT RTRIM(LTRIM([childid])) + '_' +
RTRIM(LTRIM([round])) as child_round_id ,CASE [shcrime1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shcrime1] END as [shcrime1] ,CASE [shcrime2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shcrime2] END as
[shcrime2] ,CASE [shcrime3] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shcrime3] END as [shcrime3] ,CASE
[shcrime4] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shcrime4] END as [shcrime4] ,CASE [shcrime5] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shcrime5] END as [shcrime5] ,CASE [shcrime6] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shcrime6] END as [shcrime6] ,CASE [shcrime7] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shcrime7] END as [shcrime7] ,CASE [shcrime8] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shcrime8] END as
[shcrime8] ,CASE [shregul1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shregul1] END as [shregul1] ,CASE
[shregul2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shregul2] END as [shregul2] ,CASE [shregul3] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shregul3] END as [shregul3] ,CASE [shregul4] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shregul4] END as [shregul4] ,CASE [shregul5] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shregul5] END as [shregul5] ,CASE [shregul6] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shregul6] END as
[shregul6] ,CASE [shecon1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon1] END as [shecon1] ,CASE
[shecon2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon2] END as [shecon2] ,CASE [shecon3] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon3] END as [shecon3] ,CASE [shecon4] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shecon4] END as [shecon4] ,CASE [shecon5] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shecon5] END as [shecon5] ,CASE [shecon6] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon6] END as
[shecon6] ,CASE [shecon7] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon7] END as [shecon7] ,CASE
[shecon8] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon8] END as [shecon8] ,CASE [shecon9] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon9] END as [shecon9] ,CASE [shecon10] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shecon10] END as [shecon10] ,CASE [shecon11] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shecon11] END as [shecon11] ,CASE [shecon12] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon12] END as
[shecon12] ,CASE [shecon13] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon13] END as [shecon13] ,CASE
[shecon14] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shecon14] END as [shecon14] ,CASE [shenv1] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv1] END as [shenv1] ,CASE [shenv2] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shenv2] END as [shenv2] ,CASE [shenv3] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shenv3] END as [shenv3] ,CASE [shenv4] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv4] END as [shenv4] ,CASE
[shenv5] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv5] END as [shenv5] ,CASE [shenv6] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [shenv6] END as [shenv6] ,CASE [shenv7] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes'
ELSE [shenv7] END as [shenv7] ,CASE [shenv8] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv8] END as
[shenv8] ,CASE [shenv9] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv9] END as [shenv9] ,CASE [shenv10]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv10] END as [shenv10] ,CASE [shenv11] WHEN '0' THEN 'No'
WHEN '1' THEN 'Yes' ELSE [shenv11] END as [shenv11] ,CASE [shenv12] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes'
ELSE [shenv12] END as [shenv12] ,CASE [shenv13] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shenv13] END as
[shenv13] ,CASE [shhouse1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shhouse1] END as [shhouse1] ,CASE
[shhouse2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shhouse2] END as [shhouse2] ,CASE [shhouse3] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shhouse3] END as [shhouse3] ,CASE [shfam1] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shfam1] END as [shfam1] ,CASE [shfam2] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam2]
END as [shfam2] ,CASE [shfam3] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam3] END as [shfam3] ,CASE
[shfam4] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam4] END as [shfam4] ,CASE [shfam5] WHEN '0' THEN
'No' WHEN '1' THEN 'Yes' ELSE [shfam5] END as [shfam5] ,CASE [shfam6] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes'
ELSE [shfam6] END as [shfam6] ,CASE [shfam7] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam7] END as
[shfam7] ,CASE [shfam8] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam8] END as [shfam8] ,CASE [shfam9]
WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam9] END as [shfam9] ,CASE [shfam10] WHEN '0' THEN 'No' WHEN
'1' THEN 'Yes' ELSE [shfam10] END as [shfam10] ,CASE [shfam11] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shfam11] END as [shfam11] ,CASE [shfam12] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam12] END as
[shfam12] ,CASE [shfam13] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam13] END as [shfam13] ,CASE
[shfam14] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam14] END as [shfam14] ,CASE [shfam15] WHEN '0'
THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam15] END as [shfam15] ,CASE [shfam16] WHEN '0' THEN 'No' WHEN '1'
THEN 'Yes' ELSE [shfam16] END as [shfam16] ,CASE [shfam17] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE
[shfam17] END as [shfam17] ,CASE [shfam18] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shfam18] END as
[shfam18] ,CASE [shother] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [shother] END as [shother] FROM
[CP001].[dbCare].[INPEVIETH] END TRY BEGIN CATCH SELECT ERROR_MESSAGE() AS [Error Message] ,ERROR_LINE() AS
ErrorLine ,ERROR_NUMBER() AS [Error Number] ,ERROR_SEVERITY() AS [Error Severity] ,ERROR_STATE() AS [Error
State] END CATCH ;

```

- Observations during Data Exploration

```

USE CP001;
GO

/*-----*/
/*Observations during Data Exploration*/
/*-----*/
-- Average number of children interviewed per country per sentinel site ID = 147 *
SELECT AVG([Total Children]) as Avg_Children_per_Sentinel_Site
FROM
(SELECT
    clustid,
    CASE countryid
        WHEN 'ET' THEN 'Ethiopia'
        WHEN 'IN' THEN 'India'
        WHEN 'PE' THEN 'Peru'
        WHEN 'VN' THEN 'Vietnam'
    END as Country,
    count(DISTINCT childid) as [Total Children]
FROM (Select DISTINCT childid, countryid, clustid, chsex, yc from dbCare.ch_Identity) as A
GROUP BY countryid, clustid) as B
;

-- Average number of children interviewed per only Sentinel site ID = 589 *
SELECT AVG([Total Children]) as Avg_Children_per_Sentinel_Site
FROM
(SELECT
    count(DISTINCT childid) as [Total Children]
FROM (Select DISTINCT childid, clustid, chsex, yc from dbCare.ch_Identity) as A
GROUP BY clustid) as B
;

/* Validation Queries only - ch_Identity table*/
-- Checking number of children not living in household --> childloc = no
-- Answers - India = 102 ; All countries = 588
SELECT COUNT(DISTINCT child_round_id)
FROM dbCare.ch_Identity
WHERE dbCare.ch_Identity.childloc='no'
AND dbCare.ch_Identity.childid LIKE 'IN%'

```

## 1.6.2 Views

- View 01 – v01\_WellBeing

```

USE CP001; GO

/*********************************/
/*View 01 Creation - v01_WellBeing*/
/*********************************/
CREATE VIEW v01_WellBeing
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.ch_Identity.childid, dbCare.ch_Identity.clustid,
dbCare.ch_Identity.commid, dbCare.ch_Identity.typesite, dbCare.ch_Identity.regionid, dbCare.ch_Identity.countryid,
dbCare.ch_Identity.dint,
        dbCare.ch_Identity.round, dbCare.ch_Identity.yc, dbCare.ch_Identity.childloc,
dbCare.ch_Identity.inround, dbCare.ch_Identity.panel, dbCare.ch_Identity.deceased, dbCare.ch_Identity.chsex,
dbCare.ch_Identity.chlangid,
        dbCare.ch_Identity.chethnid, dbCare.ch_Identity.chlreid, dbCare.ch_Identity.agemon,
dbCare.ch_Identity.marrcohab, dbCare.ch_Identity.marrcohab_age, dbCare.ch_Identity.birth, dbCare.ch_Identity.birth_age,
        dbCare.ch_Identity.chweight, dbCare.ch_Identity.chheight, dbCare.ch_Identity.bmi,
dbCare.ch_Identity.zwfa, dbCare.ch_Identity.zhfa, dbCare.ch_Identity.zbfa, dbCare.ch_Identity.zwfl,
dbCare.ch_Identity.fwfa, dbCare.ch_Identity.fhfa,
        dbCare.ch_Identity.fbfa, dbCare.ch_Identity.fwfl, dbCare.ch_Identity.underweight,
dbCare.ch_Identity.stunting, dbCare.ch_Identity.thinness, dbCare.ch_MedicalHistory.bcg, dbCare.ch_MedicalHistory.bwdoc,
        dbCare.ch_MedicalHistory.bwght, dbCare.ch_MedicalHistory.delivery, dbCare.ch_MedicalHistory.dpt,
dbCare.ch_MedicalHistory.hib, dbCare.ch_MedicalHistory.measles, dbCare.ch_MedicalHistory.numante,
        dbCare.ch_MedicalHistory.polio, dbCare.ch_MedicalHistory.tetanus,
dbCare.ch_MedicalHistory.chdisability, dbCare.ch_MedicalHistory.chdisscaleid, dbCare.ch_MedicalHistory.chhprob,
dbCare.ch_MedicalHistory.chillness,
        dbCare.ch_MedicalHistory.chinjury, dbCare.ch_MedicalHistory.chmighthdie,
dbCare.ch_MedicalHistory.chalcohol, dbCare.ch_MedicalHistory.chsmoke, dbCare.ch_MedicalHistory.chrephealth1,
dbCare.ch_MedicalHistory.chrephealth2,
        dbCare.ch_MedicalHistory.chrephealth3, dbCare.ch_MedicalHistory.chrephealth4,
dbCare.ch_MedicalHistory.chhealth, dbCare.ch_MedicalHistory.chhrel, dbCare.ch_MedicalHistory.cladder,
dbCare.Region.region,
        dbCare.Disability_Scale.chdisscale, dbCare.Ethnicity.chethnic
FROM      dbCare.ch_Identity INNER JOIN
                    dbCare.ch_MedicalHistory ON dbCare.ch_Identity.child_round_id = dbCare.ch_MedicalHistory.child_round_id
INNER JOIN
                    dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid INNER JOIN
dbCare.Disability_Scale ON dbCare.ch_MedicalHistory.chdisscaleid = dbCare.Disability_Scale.chdisscaleid
INNER JOIN
                    dbCare.Ethnicity ON dbCare.ch_Identity.chethnid = dbCare.Ethnicity.chethnid);

```

- View 02 – v02\_Literacy

```

USE CP001;
GO

/*****************/
/*View 02 Creation - v02_Literacy*/
/*****************/
CREATE VIEW v02_Literacy
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.Current_Grade.engrade, dbCare.School_Type.entype,
dbCare.Highest_Grade.hghgrade, dbCare.Language.chlang, dbCare.Religion.chldrel,
dbCare.ch_Identity.chldid,
dbCare.ch_Identity.typesite, dbCare.ch_Identity.clustid,
dbCare.ch_Identity.countryid, dbCare.ch_Identity.round, dbCare.ch_Identity.dint, dbCare.ch_Identity.yc,
dbCare.ch_Identity.chldloc, dbCare.ch_Identity.agemon,
dbCare.ch_Education.agegr1, dbCare.ch_Education.enrol, dbCare.ch_Education.preprim,
dbCare.ch_Education.timesch, dbCare.ch_Education.commsch, dbCare.ch_Education.commwork,
dbCare.ch_Education.hcare,
dbCare.ch_Education.hchore, dbCare.ch_Education.hplay, dbCare.ch_Education.hschool,
dbCare.ch_Education.hsleep, dbCare.ch_Education.hstudy, dbCare.ch_Education.htask,
dbCare.ch_Education.hwork,
dbCare.ch_Education.levelread, dbCare.ch_Education.levelwrit,
dbCare.ch_Education.literate
FROM dbCare.ch_Identity INNER JOIN
dbCare.ch_Education ON dbCare.ch_Identity.child_round_id =
dbCare.ch_Education.child_round_id INNER JOIN
dbCare.Current_Grade ON dbCare.ch_Education.engradeid =
dbCare.Current_Grade.engradeid INNER JOIN
dbCare.Highest_Grade ON dbCare.ch_Education.hghgradeid =
dbCare.Highest_Grade.hghgradeid INNER JOIN
dbCare.Language ON dbCare.ch_Identity.chlangid = dbCare.Language.chlangid INNER JOIN
dbCare.Religion ON dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid INNER
JOIN
dbCare.School_Type ON dbCare.ch_Education.entypeid = dbCare.School_Type.entypeid
INNER JOIN
dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid
)
;

```

- View 03 – v03\_LevelOfCare

```

USE CP001;
GO

/*****************/
/*View 03 Creation - v03_LevelOfCare*/
/*****************/
CREATE VIEW v03_LevelOfCare
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.ch_Identity.chldid, dbCare.ch_Identity.clustid,
dbCare.ch_Identity.typesite, dbCare.ch_Identity.countryid, dbCare.ch_Identity.dint, dbCare.ch_Identity.round,
dbCare.ch_Identity.yc,
dbCare.ch_Identity.chldloc, dbCare.ch_Identity.chsex, dbo.[Dad Education].dadedu, dbo.[Care
Education].caredu, dbo.[Mom Education].momedu, dbCare.Language.chlang, dbCare.Religion.chldrel,
dbCare.ch_Guardian.careage,
dbCare.ch_Guardian.carecantread, dbCare.ch_Guardian.careduid, dbCare.ch_Guardian.carehead,
dbCare.ch_Guardian.careladder, dbCare.ch_Guardian.careldr4yrs, dbCare.ch_Guardian.carerelid, dbCare.ch_Guardian.caresex,
dbCare.ch_Guardian.dadage, dbCare.ch_Guardian.dadcanread, dbCare.ch_Guardian.dadeduid,
dbCare.ch_Guardian.dadlive, dbCare.ch_Guardian.dadyrdied, dbCare.ch_Guardian.momage, dbCare.ch_Guardian.momcantread,
dbCare.ch_Guardian.momeduid, dbCare.ch_Guardian.momlive, dbCare.ch_Guardian.momyrdied,
dbCare.Relationship.relationship
FROM dbCare.ch_Guardian INNER JOIN
dbCare.ch_Identity ON dbCare.ch_Guardian.child_round_id = dbCare.ch_Identity.child_round_id INNER JOIN
dbo.[Dad Education] ON dbCare.ch_Identity.child_round_id = dbo.[Dad Education].child_round_id INNER JOIN
dbo.[Care Education] ON dbCare.ch_Identity.child_round_id = dbo.[Care Education].child_round_id INNER
JOIN
dbo.[Mom Education] ON dbCare.ch_Identity.child_round_id = dbo.[Mom Education].child_round_id INNER JOIN
dbCare.Relationship ON dbCare.ch_Guardian.carerelid = dbCare.Relationship.relid AND
dbCare.ch_Guardian.carerelid = dbCare.Relationship.relid AND dbCare.ch_Guardian.carerelid = dbCare.Relationship.relid AND
dbCare.ch_Guardian.carerelid = dbCare.Relationship.relid INNER JOIN
dbCare.Religion ON dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid AND
dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid AND dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid AND
dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid AND dbCare.ch_Identity.chldrelid = dbCare.Religion.chldrelid AND
dbCare.Language ON dbCare.ch_Identity.chlangid = dbCare.Language.chlangid AND
dbCare.ch_Identity.chlangid = dbCare.Language.chlangid AND dbCare.ch_Identity.chlangid = dbCare.Language.chlangid AND
dbCare.ch_Identity.chlangid = dbCare.Language.chlangid
)

```

- View 04 – v04\_HouseholdOverview

```

USE CP001;
GO

/*****************/
/*View 04 Creation - v04_HouseholdOverview*/
/*****************/
CREATE VIEW v04_HouseholdOverview
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.ch_Identity.childid, dbCare.ch_Identity.clustid,
dbCare.ch_Identity.commid, dbCare.ch_Identity.typesite, dbCare.ch_Identity.round, dbCare.ch_Identity.dint,
dbCare.ch_Identity.yc,
dbCare.ch_Identity.childloc, dbCare.hh_Main.headage, dbCare.hh_Main.headeduid, dbCare.hh_Main.headsex,
dbCare.hh_Main.female05, dbCare.hh_Main.female1317, dbCare.hh_Main.female1800, dbCare.hh_Main.female61,
dbCare.hh_Main.female612, dbCare.hh_Main.hsize, dbCare.hh_Main.male05, dbCare.hh_Main.male1317,
dbCare.hh_Main.male1860, dbCare.hh_Main.male61, dbCare.hh_Main.male612, dbCare.hh_Main.ownhouse,
dbCare.hh_Main.ownlandhse, dbCare.hh_Main.credit, dbCare.hh_Main.foodsec, dbCare.hh_Main.cd,
dbCare.hh_Main.cookingq, dbCare.hh_Main.drwaterq, dbCare.hh_Main.elecq, dbCare.hh_Main.hq, dbCare.hh_Main.sv,
dbCare.hh_Main.toiletq, dbCare.hh_Main.wl, dbCare.hh_LivestockOwnership.aniany,
dbCare.hh_LivestockOwnership.anibeeh, dbCare.hh_LivestockOwnership.anibufm, dbCare.hh_LivestockOwnership.anibuft,
dbCare.hh_LivestockOwnership.anibull, dbCare.hh_LivestockOwnership.anicalv,
dbCare.hh_LivestockOwnership.anicam, dbCare.hh_LivestockOwnership.anicowm, dbCare.hh_LivestockOwnership.anicowt,
dbCare.hh_LivestockOwnership.anidonk, dbCare.hh_LivestockOwnership.anidrau,
dbCare.hh_LivestockOwnership.anifish, dbCare.hh_LivestockOwnership.anifshr, dbCare.hh_LivestockOwnership.anigoat,
dbCare.hh_LivestockOwnership.aniguin, dbCare.hh_LivestockOwnership.anihbu,
dbCare.hh_LivestockOwnership.anihelf, dbCare.hh_LivestockOwnership.anillam, dbCare.hh_LivestockOwnership.animilk,
dbCare.hh_LivestockOwnership.aniothr, dbCare.hh_LivestockOwnership.anioxen,
dbCare.hh_LivestockOwnership.anipigs, dbCare.hh_LivestockOwnership.anipoul, dbCare.hh_LivestockOwnership.anirabb,
dbCare.hh_LivestockOwnership.anirumi, dbCare.hh_LivestockOwnership.anishee,
dbCare.hh_LivestockOwnership.anishri, dbCare.hh_LivestockOwnership.anisnai, dbCare.hh_LivestockOwnership.anispec,
dbCare.hh_LivestockOwnership.aniybul, dbo.[Head Education].headedu
FROM dbCare.hh_Main INNER JOIN
      dbCare.ch_Identity ON dbCare.hh_Main.child_round_id = dbCare.ch_Identity.child_round_id INNER JOIN
      dbCare.hh_LivestockOwnership ON dbCare.ch_Identity.child_round_id =
dbCare.hh_LivestockOwnership.child_round_id INNER JOIN
      dbo.[Head Education] ON dbCare.ch_Identity.child_round_id = dbo.[Head Education].child_round_id
)
;

```

- View 05 - v05\_PublicProgrammes

```

USE CP001;
GO

/*****************/
/*View 05 Creation - v05_PublicProgrammes*/
/*****************/
CREATE VIEW v05_PublicProgrammes
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.ch_Identity.childid, dbCare.Region.region,
dbCare.ch_Identity.countryid, dbCare.ch_Identity.typesite, dbCare.ch_Identity.yc,
dbCare.ch_Identity.round, dbCare.hh_PublicProgrammes.eap,
dbCare.hh_PublicProgrammes.hep, dbCare.hh_PublicProgrammes.othprog,
dbCare.hh_PublicProgrammes.psnp_ds, dbCare.hh_PublicProgrammes.psnp_pw,
dbCare.hh_PublicProgrammes.resettled, dbCare.hh_PublicProgrammes.ikp,
dbCare.hh_PublicProgrammes.ikp_child, dbCare.hh_PublicProgrammes.nregs,
dbCare.hh_PublicProgrammes.nregs_allow, dbCare.hh_PublicProgrammes.nregs_work,
dbCare.hh_PublicProgrammes.pds,
dbCare.hh_PublicProgrammes.rajiv, dbCare.hh_PublicProgrammes.sabla,
dbCare.hh_PublicProgrammes.sabla_yl, dbCare.hh_PublicProgrammes.beca_yl,
dbCare.hh_PublicProgrammes.bonograt, dbCare.hh_PublicProgrammes.insur_yl,
dbCare.hh_PublicProgrammes.juntos, dbCare.hh_PublicProgrammes.minsa_yl,
dbCare.hh_PublicProgrammes.projoven_yl, dbCare.hh_PublicProgrammes.sisgrat_yl,
dbCare.hh_PublicProgrammes.molisa06,
dbCare.hh_PublicProgrammes.molisa09, dbCare.hh_PublicProgrammes.molisa10,
dbCare.hh_PublicProgrammes.molisa11, dbCare.hh_PublicProgrammes.molisa12,
dbCare.hh_PublicProgrammes.molisa13,
dbCare.hh_PublicProgrammes.molisa14, dbCare.hh_PublicProgrammes.molisa15,
dbCare.hh_PublicProgrammes.molisa16
FROM dbCare.ch_Identity INNER JOIN
      dbCare.hh_PublicProgrammes ON dbCare.ch_Identity.child_round_id =
dbCare.hh_PublicProgrammes.child_round_id INNER JOIN
      dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid
)
;

```

- View 06 – v06\_ShocksEndured

```

USE CP001;
GO

/*****************/
/*View 06 Creation - v06_ShocksEndured*/
/*****************/
CREATE VIEW v06_ShocksEndured
AS
(
SELECT dbCare.ch_Identity.child_round_id, dbCare.ch_Identity.childid, dbCare.ch_Identity.clustid,
dbCare.ch_Identity.typesite, dbCare.ch_Identity.countryid, dbCare.ch_Identity.dint,
dbCare.ch_Identity.round, dbCare.ch_Identity.yc,
dbCare.ch_Identity.childloc, dbCare.ch_Identity.chsex, dbCare.ch_Identity.agemon,
dbCare.Region.region, dbCare.hh_Shocks.shcrime1, dbCare.hh_Shocks.shcrime2, dbCare.hh_Shocks.shcrime3,
dbCare.hh_Shocks.shcrime4,
dbCare.hh_Shocks.shcrime5, dbCare.hh_Shocks.shcrime6, dbCare.hh_Shocks.shcrime7,
dbCare.hh_Shocks.shcrime8, dbCare.hh_Shocks.shregul1, dbCare.hh_Shocks.shregul2,
dbCare.hh_Shocks.shregul3, dbCare.hh_Shocks.shregul4,
dbCare.hh_Shocks.shregul5, dbCare.hh_Shocks.shregul6, dbCare.hh_Shocks.shecon1,
dbCare.hh_Shocks.shecon2, dbCare.hh_Shocks.shecon3, dbCare.hh_Shocks.shecon4,
dbCare.hh_Shocks.shecon5, dbCare.hh_Shocks.shecon6,
dbCare.hh_Shocks.shecon7, dbCare.hh_Shocks.shecon8, dbCare.hh_Shocks.shecon9,
dbCare.hh_Shocks.shecon10, dbCare.hh_Shocks.shecon11, dbCare.hh_Shocks.shecon12,
dbCare.hh_Shocks.shecon13, dbCare.hh_Shocks.shecon14,
dbCare.hh_Shocks.shenv1, dbCare.hh_Shocks.shenv2, dbCare.hh_Shocks.shenv3,
dbCare.hh_Shocks.shenv4, dbCare.hh_Shocks.shenv5, dbCare.hh_Shocks.shenv6, dbCare.hh_Shocks.shenv7,
dbCare.hh_Shocks.shenv8,
dbCare.hh_Shocks.shenv9, dbCare.hh_Shocks.shenv10, dbCare.hh_Shocks.shenv11,
dbCare.hh_Shocks.shenv12, dbCare.hh_Shocks.shenv13, dbCare.hh_Shocks.shhouse1,
dbCare.hh_Shocks.shhouse2, dbCare.hh_Shocks.shhouse3,
dbCare.hh_Shocks.shfam1, dbCare.hh_Shocks.shfam2, dbCare.hh_Shocks.shfam3,
dbCare.hh_Shocks.shfam4, dbCare.hh_Shocks.shfam5, dbCare.hh_Shocks.shfam6, dbCare.hh_Shocks.shfam7,
dbCare.hh_Shocks.shfam8,
dbCare.hh_Shocks.shfam9, dbCare.hh_Shocks.shfam10, dbCare.hh_Shocks.shfam11,
dbCare.hh_Shocks.shfam12, dbCare.hh_Shocks.shfam13, dbCare.hh_Shocks.shfam14,
dbCare.hh_Shocks.shfam15, dbCare.hh_Shocks.shfam16,
dbCare.hh_Shocks.shfam17, dbCare.hh_Shocks.shfam18, dbCare.hh_Shocks.shother,
dbCare.ch_Identity.deceased
FROM dbCare.ch_Identity INNER JOIN
      dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid INNER JOIN
      dbCare.hh_Shocks ON dbCare.ch_Identity.child_round_id =
dbCare.hh_Shocks.child_round_id)
;
```

#### 1.6.2.1 Reports

##### 1. Report 1 - zREP01\_AllChildrenByCtry

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*********************************/
/*Report 01 Creation - zREP01_AllChildrenByCtry*/
/*********************************/
-- Total Poor children per country - M & F, YC & OC
CREATE VIEW zREP01_AllChildrenByCtry
AS
(
SELECT

    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END as Country,
    count(DISTINCT childid) as [Total Children],
-- Total Males vs. Females
    sum(CASE WHEN chsex = 'Male' then 1 END) as [Total Males],
    sum(CASE WHEN chsex = 'Female' then 1 END) as [Total Females],
-- Total Younger cohort vs Older cohort
    sum(CASE WHEN yc = 'YC' then 1 END) as [Younger Cohort],
    sum(CASE WHEN yc = 'OC' then 1 END) as [Older Cohort],
-- Young Males vs Young Females
    sum(CASE WHEN chsex = 'Male' AND yc = 'YC' then 1 END) as [Young Males],
    sum(CASE WHEN chsex = 'Female' AND yc = 'YC' then 1 END) as [Young Females],
-- Old Males vs Old Females
    sum(CASE WHEN chsex = 'Male' AND yc = 'OC' then 1 END) as [Old Males],
    sum(CASE WHEN chsex = 'Female' AND yc = 'OC' then 1 END) as [Old Females]

FROM
    (Select DISTINCT childid, countryid, chsex, yc from dbCare.ch_Identity) as A
GROUP BY countryid
--ORDER BY count(DISTINCT childid) DESC
                                /*Observation - The ORDER BY clause is invalid in views
creation*/
);
GO
COMMIT TRANSACTION;
;

```

- Result Review -

Country	Total Children	Total Males	Total Females	Younger Cohort	Older Cohort	Young Males	Young Females	Old Males	Old Females
1 Vietnam	3000	1528	1472	2000	1000	1027	973	501	499
2 India	3019	1572	1447	2011	1008	1081	930	491	517
3 Ethiopia	2999	1559	1440	1999	1000	1049	950	510	490
4 Peru	2766	1413	1353	2052	714	1027	1025	386	328

## 2. Report 2 - zREP02\_AllChildrenBySentinelSite

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*****************/
/*Report 02 Creation - zREP02_AllChildrenBySentinelSite*/
/*****************/
-- Number of poor children per country per Sentinel Site

CREATE VIEW zREP02_AllChildrenBySentinelSite
AS
(
SELECT
    clustid,
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END AS Country,
    COUNT(DISTINCT childid) AS [Total Children],

    -- Total Males vs. Females
    SUM(CASE WHEN chsex = 'Male' THEN 1 END) AS [Total Males],
    SUM(CASE WHEN chsex = 'Female' THEN 1 END) AS [Total Females],
    -- Total Younger cohort vs Older cohort
    SUM(CASE WHEN yc = 'YC' THEN 1 END) AS [Younger Cohort],
    SUM(CASE WHEN yc = 'OC' THEN 1 END) AS [Older Cohort],
    -- Young Males vs Young Females
    SUM(CASE WHEN chsex = 'Male' AND yc = 'YC' THEN 1 END) AS [YC Males],
    SUM(CASE WHEN chsex = 'Female' AND yc = 'YC' THEN 1 END) AS [YC Females],
    -- Old Males vs Old Females
    SUM(CASE WHEN chsex = 'Male' AND yc = 'OC' THEN 1 END) AS [OC Males],
    SUM(CASE WHEN chsex = 'Female' AND yc = 'OC' THEN 1 END) AS [OC Females]

    FROM (Select DISTINCT childid, countryid, clustid, chsex, yc from dbCare.ch_Identity) AS A
    GROUP BY countryid, clustid
)
;

GO
COMMIT TRANSACTION;

```

- Result Review – An average of 147 number of children per country per sentinel site have been interviewed. There are totally 20 sentinel sites in each of the 4 countries.

	clustid	Country	Total Children	Total Males	Total Females	Younger Cohort	Older Cohort	YC Males	YC Females	OC Males	OC Females
1	01	Ethiopia	150	72	78	100	50	53	47	19	31
2	01	India	150	73	77	100	50	52	48	21	29
3	01	Peru	150	76	74	100	50	50	50	26	24
4	01	Vietnam	150	78	72	100	50	51	49	27	23
5	02	Ethiopia	150	74	76	100	50	51	49	23	27
6	02	India	150	77	73	100	50	55	45	22	28
7	02	Peru	125	65	60	100	25	52	48	13	12

### 3. Report 3 - zREP03\_DeathsByRegionAndEthnicity

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*************************************
/*Report 03 Creation - zREP03_DeathsByRegionAndEthnicity*/
/*************************************
-- Number of deaths by SiteType, Region & Ethnicity

CREATE VIEW zREP03_DeathsByRegionAndEthnicity

AS

SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END AS Country,
    typesite AS [Site Type],
    region AS Region,
    chethnic AS Ethnicity,
    count(DISTINCT childid) AS [Total Children],
    SUM(IIF(round=1 AND deceased='Yes', 1, 0)) AS Deaths_R1,
    SUM(IIF(round=2 AND deceased='Yes', 1, 0))
        - SUM(IIF(round=1 AND deceased='Yes', 1, 0)) AS Deaths_R2,
    SUM(IIF(round=3 AND deceased='Yes', 1, 0))
        - SUM(IIF(round=2 AND deceased='Yes', 1, 0)) AS Deaths_R3,
    SUM(IIF(round=4 AND deceased='Yes', 1, 0))
        - SUM(IIF(round=3 AND deceased='Yes', 1, 0)) AS Deaths_R4,
    SUM(IIF(round=5 AND deceased='Yes', 1, 0))
        - SUM(IIF(round=4 AND deceased='Yes', 1, 0)) AS Deaths_R5,
    SUM(IIF(round=5 AND deceased='Yes', 1, 0)) AS [Total Deaths]

FROM (Select
        DISTINCT childid, countryid, typesite, round, deceased,
        dbCare.Region.region, dbCare.Ethnicity.chethnic
        FROM dbCare.ch_Identity
        INNER JOIN dbCare.Region
            ON dbCare.ch_Identity.regionid =
        dbCare.Region.regionid
        INNER JOIN dbCare.Ethnicity
            ON dbCare.ch_Identity.chethnicid =
        dbCare.Ethnicity.chethnicid) AS A
    GROUP BY countryid, typesite, region, chethnic
;

GO
COMMIT TRANSACTION;

```

- Result Review – The top 5 regions with highest total number of deaths are shown in the screenshot. Also, we have death count per round that can be seen.

	Country	Site Type	Region	Ethnicity	Total Children	Deaths_R1	Deaths_R2	Deaths_R3	Deaths_R4	Deaths_R5	Total Deaths
1	Ethiopia		Not known	Amhara	126	0	23	1	4	3	31
2	India		Not known	BC	87	0	9	7	10	4	30
3	Peru		Not known	Mestizo	345	0	19	5	3	2	29
4	Ethiopia		Not known	Tigrian	74	0	12	3	3	1	19
5	Ethiopia		Not known	Oromo	82	0	10	3	0	0	13

#### 4. Report 4 - zREP04\_ChildPhysiqueReport

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*****************/
/*Report 04 Creation - zREP04_ChildPhysiqueReport*/
/*****************/
-- Country & Region wise - Child body change over the years from 2002 to 2007 segregated by Sex

CREATE VIEW zREP04_ChildPhysiqueReport

AS

SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END as Country,
    region as Region,
    --childid,
    chsex as Sex,
    round as Round,
    --underweight as Underweight,           -- Wrongly classified column
    ROUND(AVG(agemon)/12, 0) as [Avg Age (in Yrs)],
    ROUND(AVG(chweight),2) as [Avg Weight(in kg)],
    ROUND(AVG(chheight)*0.032, 2) as [Avg Height(in ft)],
    ROUND(AVG(bmi),2) as [Avg BMI],
    ROUND(AVG(zwfa),3) as [Avg_Z-score - Weight for Age],
    ROUND(AVG(zhfa),3) as [Avg_Z-score - Height for Age],
    ROUND(AVG(zbfa),3) as [Avg_Z-score - BMI]

FROM (Select
        DISTINCT childid, countryid, chsex, round, agemon, chweight, zwfa,
        chheight, zhfa, bmi, zbfa, dbCare.Region.region
        FROM      dbCare.ch_Identity
        INNER JOIN dbCare.Region
                  ON dbCare.ch_Identity.regionid =
        dbCare.Region.regionid ) as A
    GROUP BY countryid, region, chsex, round
;

GO
COMMIT TRANSACTION;

```

- Result Review – Peru has an abnormal average child weight as 23.6 kg, whereas all the other countries have more than 25 kg as average weight. Also Peru has abnormal average height as 2.96 ft whereas, other countries have around 3.5 ft as average height. Avg BMI is less than 15 for 3 countries, which means they are severely underweight. Only India has an average BMI of 21.69 from the given numbers which can be considered as normal.

## 5. Report 5 - zREP05\_BirthandImmunizations

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*********************************/
/*Report 05 Creation - zREP05_BirthandImmunizations*/
/*********************************/
-- Country & Region wise - Birth and Immunizations report

CREATE VIEW zREP05_BirthandImmunizations

AS

SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END AS Country,
    region AS Region,
    --childid,
    chsex AS Sex,
    --round AS Round,
    --ROUND(AVG(agemon)/12, 0) AS [Avg Age (in Yrs)],
    SUM(IIF([bcg]='Yes', 1, 0)) AS [BCG Yes],
    SUM(IIF([bcg]='No', 1, 0)) AS [BCG No],
    SUM(IIF([bcg]='', 1, 0)) AS [BCG Pending],
    SUM(IIF([dpt]='Yes', 1, 0)) AS [DPT Yes],
    SUM(IIF([dpt]='No', 1, 0)) AS [DPT No],
    SUM(IIF([dpt]='', 1, 0)) AS [DPT Pending],
    SUM(IIF([hib]='Yes', 1, 0)) AS [HIB Yes],
    SUM(IIF([hib]='No', 1, 0)) AS [HIB No],
    SUM(IIF([hib]='', 1, 0)) AS [HIB Pending],
    SUM(IIF([measles]='Yes', 1, 0)) AS [Measles Yes],
    SUM(IIF([measles]='No', 1, 0)) AS [Measles No],
    SUM(IIF([measles]='', 1, 0)) AS [Measles Pending],
    SUM(IIF([polio]='Yes', 1, 0)) AS [Polio Yes],
    SUM(IIF([polio]='No', 1, 0)) AS [Polio No],
    SUM(IIF([polio]='', 1, 0)) AS [Polio Pending],
    SUM(IIF([tetanus]='Yes', 1, 0)) AS [Tetanus Yes],
    SUM(IIF([tetanus]='No', 1, 0)) AS [Tetanus No],
    SUM(IIF([tetanus]='', 1, 0)) AS [Tetanus Pending],
    SUM(IIF([bwdoc]='No', 1, 0)) AS [Birthweight wrongly documented],
    SUM(IIF([delivery]='No', 1, 0)) AS [No skilled personnel during delivery],
    SUM(numante) AS [Num of Antenatal visits]

FROM (Select DISTINCT childid, countryid, chsex, round, agemon, [bcg], [bwdoc], [delivery], [dpt], [hib],
,[measles],[numante],[polio],[tetanus], dbCare.Region.region FROM dbCare.ch_Identity INNER JOIN
dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid INNER JOIN dbCare.ch_MedicalHistory ON
dbCare.ch_Identity.child_round_id = dbCare.ch_MedicalHistory.child_round_id) AS A

GROUP BY countryid, region, chsex
--ORDER BY countryid, region
;

GO
COMMIT TRANSACTION;

```

- Result Review – Almost all the 4 countries are doing fairly better with vaccinations for especially BCG, Measles and Polio. For DPT however, it is still improving, but not so much as it is for the other 3 vaccines mentioned earlier. HIB and Tetanus are severely undermined which needs to be prioritized first.

## 6. Report 6 - zREP06\_OverallHealthandHabits

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO
/************************************************************/
/*Report 06 Creation - zREP06_OverallHealthandHabits*/
/************************************************************/
-- Country & Region wise - Overall Health and Habits

CREATE VIEW zREP06_OverallHealthandHabits
AS
SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END AS Country,
    region AS Region,
    typesite AS [Site Type],
    --childid,
    chsex AS Sex,
    yc AS [Younger/Older Cohort],
    round AS Round,
    AVG(cladder) AS [Avg Subjective Wellbeing],
    COUNT(childid) AS [Total Children],
    SUM(IIF([chalcohol]='Yes', 1, 0)) AS [Alcohol Addicts],
    SUM(IIF([chsmoke]='At least once a month' OR [chsmoke]='Every day' OR [chsmoke]='At least
once a week', 1, 0)) AS [Smoking Addicts],
    SUM(IIF([chdisability]='Yes', 1, 0)) AS [Permanently Disabled],
    SUM(IIF([chhprob]='Yes', 1, 0)) AS [Long-term Health issues],
    SUM(IIF([chillness]='Yes', 1, 0)) AS [Serious illnesses],
    SUM(IIF([chinjury]='Yes', 1, 0)) AS [Serious injuries],
    SUM(IIF([chmightdie]='Yes', 1, 0)) AS [Child Might Die],
    AVG(chrepehealth1) AS [Avg Child's knowledge on Reproductive Health],
    SUM(IIF([chrepehealth2]='Yes', 1, 0)) AS [Knows condom can prevent STD],
    SUM(IIF([chrepehealth3]='Yes', 1, 0)) AS [Knows STDs can pass from healthy people],
    SUM(IIF([chrepehealth4] LIKE 'Doesn''t know%condom%', 1, 0)) AS [Doesn't know what is a
condom]
FROM (Select DISTINCT childid, countryid, typesite, chsex, round, agemon, [cladder]
      ,[chdisability],[chdisscale],[chhprob],[chillness],[chinjury],[chmightdie],[chalcohol]
      ,[chsmoke],[chrepehealth1],[chrepehealth2],[chrepehealth3],[chrepehealth4], dbCare.Region.region
      FROM dbCare.ch_Identity INNER JOIN dbCare.Region ON dbCare.ch_Identity.regionid =
      dbCare.Region.regionid INNER JOIN dbCare.ch_MedicalHistory ON dbCare.ch_Identity.child_round_id =
      dbCare.ch_MedicalHistory.child_round_id INNER JOIN dbCare.Disability_Scale ON
      dbCare.ch_MedicalHistory.chdisscaleid = dbCare.Disability_Scale.chdisscaleid) AS A
GROUP BY countryid, region, typesite, chsex, yc, round;
GO
COMMIT TRANSACTION;

```

- Result Review – Highest alcohol addicts are seen in Ethiopia with 463 cases out of which 325 are OC and 138 are YC. Highest smoking addicts are seen in Peru with 432 cases out of which 354 are OC and 78 are YC. India ranks worst in the subjective well-being with only an average of 2.99. Around 1400 instances were present where the survey answered that the child might die in each of the 3 countries Ethiopia, India and Peru and at least more than 1000 of them are YC. Highest serious injuries and highest serious illnesses are caused in India only. A vast majority of Vietnamese children do not have knowledge on reproductive health. Most of the OC from Vietnam and Ethiopia don't even know what is a condom. Ethiopian children know well about STDs than any other children. Long term health issues are highly seen in Peru, followed by Vietnam. A majority of permanently disabled children are YC, the highest number of disabled are present in Vietnam.

## 7. Report 7 - zREP07\_GuardianOverview

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/*****************/
/*Report 07 Creation - zREP07_GuardianOverview*/
/*****************/
-- Country & Religion wise - Guardian Overview

CREATE VIEW zREP07_GuardianOverview
AS

SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END AS Country,
    chdrel AS Religion,
    typesite AS [Site Type],
    chsex AS Sex,
    yc AS [Younger/Older Cohort],
    round AS Round,
    caredu AS [Caregiver's Education],
    dadedu AS [Father's Education],
    momedu AS [Mother's Education],
    COUNT(chldid) AS [Total Children],
    AVG(careage) AS [Avg Caregiver's age(in yrs)],
    SUM(IIF(carecantread='Yes', 1,0)) AS [Caregivers who cannot read],
    AVG(careladder) AS [Avg Caregiver's Subjective Wellbeing],
    AVG(careldr4yrs) AS [Avg Caregiver's Subj Wellbeing (in 4 years)],
    AVG(dadage) AS [Avg Father's age(in yrs)],
    SUM(IIF(dadcantread='Yes', 1,0)) AS [Fathers who cannot read],
    SUM(IIF(dadlive LIKE '%ive%', 1,0)) AS [Father Alive],
    SUM(IIF(dadlive LIKE '%die%', 1,0)) AS [Father Died],
    AVG(momage) AS [Avg Mother's age(in yrs)],
    SUM(IIF(momcantread='Yes', 1,0)) AS [Mothers who cannot read],
    SUM(IIF(momlive LIKE '%ive%', 1,0)) AS [Mother Alive],
    SUM(IIF(momlive LIKE '%die%', 1,0)) AS [Mother Died]
FROM (Select DISTINCT chldid, countryid, typesite, chsex, round, yc, agemon, [careage] , [carecantread] , [caredu]
      ,[carehead] --,[careid] ,[careladder] ,[careldr4yrs] ,[caresex] ,[dadage] ,[dadcantread] ,[dadedu] --,[dadid]
      ,[dadlive] --,[dadyrdied] ,[momage] ,[momcantread] ,[momedu] --,[momid] ,[momlive] --,[momyrdied]
      , dbCare.Religion.chdrel , dbCare.Language.chlang
      FROM dbCare.ch_Identity INNER JOIN dbCare.Religion ON dbCare.ch_Identity.chdrelid = dbCare.Religion.chdrelid INNER
      JOIN dbCare.Language ON dbCare.ch_Identity.chlangid = dbCare.Language.chlangid INNER JOIN dbCare.ch_Guardian ON
      dbCare.ch_Identity.child_round_id = dbCare.ch_Guardian.child_round_id INNER JOIN dbo.[Care Education] ON
      dbCare.ch_Identity.child_round_id = dbo.[Care Education].child_round_id INNER JOIN dbo.[Dad Education] ON
      dbCare.ch_Identity.child_round_id = dbo.[Dad Education].child_round_id INNER JOIN dbo.[Mom Education] ON
      dbCare.ch_Identity.child_round_id = dbo.[Mom Education].child_round_id AS A
      GROUP BY countryid, chdrel, typesite, chsex, yc, round, caredu, dadedu, momedu
      );
GO
COMMIT TRANSACTION;

```

- Result Review – The subjective well-being of all the caregivers has declined from the last 4 years from 3.60 to 2.77, also individually declined for each country. There have been 323 deaths of fathers of children over the 5 rounds in Ethiopia, and 231 deaths in India which are more prominent than Peru and Vietnam. The fathers who are highly undereducated are from Ethiopia and highly educated are from Peru. The mothers who lack the ability to read are mostly either from Ethiopia and India.

## 8. Report 8 - zREP08\_EducationOverview

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/************************************************************/
/*Report 08 Creation - zREP08_EducationOverview*/
/************************************************************/
-- Country & Region wise - Educational Overview
CREATE VIEW zREP08_EducationOverview
AS
SELECT
    CASE countryid      WHEN 'ET' THEN 'Ethiopia'
                           WHEN 'IN' THEN 'India'
                           WHEN 'PE' THEN 'Peru'
                           WHEN 'VN' THEN 'Vietnam'
    END as Country,
    region as Region,
    typesite as [Site Type],
    entype as [School Type],
    engrade as [Current Grade],
    round as Round,
    yc as [Younger/Older Cohort],
    chsex as Sex,
    COUNT(childid) as [Total Children],
    MAX(agegr1) - AVG(agegr1) as [Years lagging behind the Avg Grade 1 student],
    SUM(IIF([preprim]='Yes', 1, 0)) as [Attended Pre-primary school],
    AVG(timesch) as [Avg Travel time to school(in mins)],
    AVG(commesch) as [Avg Commute time to school(in mins)],
    AVG(commwork) as [Avg Commute time to work(in mins)],
    ROUND(AVG(hcare),2) as [Avg Hours/day spent in caring for household members]
    ,ROUND(AVG(hchore),2) as [Avg Hours/day spent in household chores]
    ,ROUND(AVG(hplay),2) as [Avg Hours/day spent in leisure activities]
    ,ROUND(AVG(hschool),2) as [Avg Hours/day spent at school]
    ,ROUND(AVG(hsleep),2) as [Avg Hours/day spent sleeping]
    ,ROUND(AVG(hstudy),2) as [Avg Hours/day spent studying outside school]
    ,ROUND(AVG(htask),2) as [Avg Hours/day spent in domestic tasks - farming, family business]
    ,ROUND(AVG(hwork),2) as [Avg Hours/day spent in paid activity]

    FROM (Select DISTINCT childid, countryid, typesite, chsex, round, yc, agemon, [agegr1] ,[enrol] ,[entype]
    ,[preprim] ,[timesch] ,[commesch] ,[commwork] ,[hcare] ,[hchore] ,[hplay] ,[hschool] ,[hsleep] ,[hstudy] ,[htask]
    ,[hwork] , dbCare.Region.region
    FROM dbCare.ch_Identity INNER JOIN dbCare.Region ON dbCare.ch_Identity.regionid = dbCare.Region.regionid INNER JOIN
    dbCare.ch_Education ON dbCare.ch_Identity.child_round_id = dbCare.ch_Education.child_round_id INNER JOIN
    dbCare.Current_Grade ON dbCare.ch_Education.engradeid = dbCare.Current_Grade.engradeid INNER JOIN dbCare.School_Type
    ON dbCare.ch_Education.entypeid = dbCare.School_Type.entypeid) as A
    GROUP BY countryid, region, typesite, entype, engrade, round, yc, chsex
    --ORDER BY countryid, region, typesite, entype, engrade, chsex asc
;

GO
COMMIT TRANSACTION;

```

- Result Review – Ethiopia has the highest number of children with the number of years lagging behind the average Grade 1 student (1.83), and the regions like Amhara and Oromiya needs to be targeted to improve first, and also the males are lagging behind more than the females. Similarly Northern Uplands in Vietnam, Telangana in India have children who are lagging behind. Majority of the Ethiopian children do not attend pre-primary school. Coastal Andhra is a region where children take the maximum amount of time to commute to school, and India seems to be majorly lacking in providing schooling in the nearby areas for these children, in addition, most of the children spend their time in school itself in India. SNNP, Amhara, Oromiya in Ethiopia, Costa, Selva, Sierra regions in Peru have the most children spending their time more for caring for household members. The children in Peru spend more time in paid activity than any other country. In Somali, the average hours spent in studying outside school is 5 hours. Average travel time is more in the regions like Afar, Oromiya from Ethiopia, other regions from India and Highlands from Vietnam.

## 9. Report 9 - zREP09\_LiteracyStatus

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/************************************************************/
/*Report 09 Creation - zREP09_LiteracyStatus*/
/************************************************************/
-- Country & Region wise - Child Literacy Status
CREATE VIEW zREP09_LiteracyStatus
AS
SELECT
    CASE countryid WHEN 'ET' THEN 'Ethiopia'
                    WHEN 'IN' THEN 'India'
                    WHEN 'PE' THEN 'Peru'
                    WHEN 'VN' THEN 'Vietnam'
    END as Country,
    region as Region,
    typesite as [Site Type],
    chsex as Sex,
    MAX(chlang) as [Majorly used language],
    hghgrade as [Highest Grade],
    round as Round,
    yc as [Younger/Older Cohort],
    COUNT(childid) as [Total Children],
    SUM(IIF([levlread] LIKE '%word%' OR [levlread] LIKE '%letters%', 1,0)) as [Can read words or letters],
    SUM(IIF([levlread] LIKE '%sent%', 1,0)) as [Can read sentences],
    SUM(IIF([levlread] LIKE '%Can''t read%', 1,0)) as [Cannot read],
    SUM(IIF([levlwrit] LIKE '%with difficulty%', 1,0)) as [Can write but needs help],
    SUM(IIF([levlwrit] LIKE '%without%', 1,0)) as [Can write],
    SUM(IIF([levlwrit] LIKE '%No%', 1,0)) as [Cannot write],
    SUM(IIF([literate] LIKE '%Yes%', 1,0)) as [Literate],
    SUM(IIF([literate] LIKE '%No%', 1,0)) as [Illiterate]
FROM (Select DISTINCT childid, countryid, typesite, chsex, round, yc, agemon, [hghgrade] ,[levlread]
      ,[levlwrit] ,[literate]
      , dbCare.Region.region , dbCare.Language.chlang
      FROM dbCare.ch_Identity INNER JOIN dbCare.Region ON dbCare.ch_Identity.regionid =
dbCare.Region.regionid INNER JOIN dbCare.Language ON dbCare.ch_Identity.chlangid =
dbCare.Language.chlangid INNER JOIN dbCare.ch_Education ON dbCare.ch_Identity.child_round_id =
dbCare.ch_Education.child_round_id LEFT OUTER JOIN dbCare.Highest_Grade ON
dbCare.ch_Education.hghgradeid = dbCare.Highest_Grade.hghgradeid) as A
GROUP BY countryid, region, chsex, typesite, round, yc, chlang, hghgrade
--ORDER BY countryid, region, chsex, typesite, round, yc, chlang, hghgrade asc
;

GO
COMMIT TRANSACTION;

```

- Result Review – In 6882 instances, children conveyed that they are illiterate and in 7944 instances, children conveyed that they are literate. Most of them can read and write sentences, with occasional help. The number of literates improved overall in the first 3 rounds in each of the countries for the children, whereas the number of illiterates dropped in round 2 but gradually increased in round 3 in all the countries.

## 10. Report 10 - zREP10\_HouseholdAttributes

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/************************************************************/
/*Report 10 Creation - zREP10_HouseholdAttributes*/
/************************************************************/
-- Country & Region wise - Household Attributes
CREATE VIEW zREP10_HouseholdAttributes
AS
SELECT CASE countryid WHEN 'ET' THEN 'Ethiopia' WHEN 'IN' THEN 'India' WHEN 'PE' THEN 'Peru' WHEN
'VN' THEN 'Vietnam' END as Country, region as Region, typesite as [Site Type], chsex as Sex,
MAX(chlang) as [Majorly used language], round as Round, yc as [Younger/Older Cohort], headsex as
[Household Head's sex], AVG(headage) as [Avg Age of Household's Head], COUNT(childid) as [Total
Children],
AVG([hhsize]) as [Avg Household size],
AVG([female05]) as [Avg Number of females aged 0-5] ,AVG([female612]) as [Avg Number of females aged
6-12] ,AVG([female1317]) as [Avg Number of females aged 13-17] ,AVG([female1860]) as [Avg Number of
females aged 18-60] ,AVG([female61]) as [Avg Number of females aged 61+] ,AVG([male05]) as [Avg
Number of males aged 0-5] ,AVG([male612]) as [Avg Number of males aged 6-12] ,AVG([male1317]) as [Avg
Number of males aged 13-17] ,AVG([male1860]) as [Avg Number of males aged 18-60] ,AVG([male61]) as
[Avg Number of males aged 61+]
,SUM(IIF([ownlandhse]='Yes', 1,0)) as [Household owns land where house is on]
,SUM(IIF([ownhouse]='Yes', 1,0)) as [Household own the house] ,SUM([credit]) as [Household has
obtained loan or credit] ,SUM(IIF([foodsec] LIKE '%do not eat%', 1,0)) as [People don't eat enough in
the household] ,ROUND(AVG([wi]),2) as [Avg Wealth Index] ,ROUND(AVG([hq]),2) as [Avg Housing Quality
Index] ,ROUND(AVG([sv]),2) as [Avg Access to Services Index] ,ROUND(AVG([cd]),2) as [Avg Consumer
Durables Index] ,SUM(IIF([cookingq]='Yes', 1, 0)) as [Access to adequate fuels for cooking]
,SUM(IIF([cookingq]='No', 1, 0)) as [No Access to adequate fuels for cooking]
,SUM(IIF([drwaterq]='Yes', 1, 0)) as [Access to safe drinking water] ,SUM(IIF([drwaterq]='No', 1,
0)) as [No Access to safe drinking water]
,SUM(IIF([elecq]='Yes', 1, 0)) as [Access to Electricity] ,SUM(IIF([elecq]='No', 1, 0)) as [No
Access to Electricity]
,SUM(IIF([toiletq]='Yes', 1, 0)) as [Access to sanitation] ,SUM(IIF([toiletq]='No', 1, 0)) as [No
Access to sanitation]
,SUM(IIF([aniany]='Yes',1,0)) as [Livestock ownership (in past 12 months)] FROM (Select DISTINCT
childid, countryid, typesite, chsex, round, yc, agemon, [headage] --,[headeduid] --,[headid] --
,[headrelid] ,[headsex] ,[female05] ,[female1317] ,[female1860] ,[female61] ,[female612] ,[hhsize]
,[male05] ,[male1317] ,[male1860] ,[male61] ,[male612] ,[ownhouse] ,[ownlandhse] ,[credit] ,[foodsec]
,[cd] ,[cookingq] ,[drwaterq] ,[elecq] ,[hq] ,[sv] ,[toiletq] ,[wi]
, dbCare.Region.region , dbCare.Language.chlang
,[aniany]

FROM dbCare.ch_Identity INNER JOIN dbCare.Region ON dbCare.ch_Identity.regionid =
dbCare.Region.regionid INNER JOIN dbCare.Language ON dbCare.ch_Identity.chlangid =
dbCare.Language.chlangid INNER JOIN dbCare.hh_Main ON dbCare.ch_Identity.child_round_id =
dbCare.hh_Main.child_round_id INNER JOIN dbCare.hh_LivestockOwnership ON
dbCare.ch_Identity.child_round_id = dbCare.hh_LivestockOwnership.child_round_id) as A

GROUP BY countryid, region, chsex, typesite, round, yc, chlang, headsex ;
GO
COMMIT TRANSACTION;

```

- Result Review – India is lacking behind in terms of access to sanitation. Ethiopian and Vietnamese people mostly don't eat enough in the household. Vietnam majorly doesn't have proper access to safe drinking water. Ethiopia largely lacks electricity and fuels for cooking. Indian and Vietnamese households have majorly obtained loans or credit. The average consumer durables index is 0.25 in Ethiopia, which is quite less. Average housing quality index is only 0.31 in Ethiopia, and 0.34 in Peru. Average number of males and females aged 61 and above are present in larger number of households, especially in India.

## 11. Report 11 - zREP11\_PublicProgrammes

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/************************************************************/
/*Report 11 Creation - zREP11_PublicProgrammes*/
/************************************************************/
-- Country & Region wise - Beneficiaries of Public Programmes

CREATE VIEW zREP11_PublicProgrammes
AS
SELECT CASE countryid WHEN 'ET' THEN 'Ethiopia' WHEN 'IN' THEN 'India' WHEN 'PE' THEN 'Peru' WHEN
'VN' THEN 'Vietnam' END as Country,
region as Region, typesite as [Site Type], chsex as Sex,
round as Round, yc as [Younger/Older Cohort],
COUNT(chilid) as [Total Children], AVG([hhsiz]) as [Avg Household size], SUM(IIF([eap]='Yes',1,0)) as
[Emergency Aid Programme] ,SUM(IIF([hep]='Yes',1,0)) as [Health Extension Programme]
,SUM(IIF([othprog]='Yes',1,0)) as [Other Security Programmes] ,SUM(IIF([psnp_ds]='Yes',1,0)) as
[PSNP-Direct Support Programme] ,SUM(IIF([psnp_pw]='Yes',1,0)) as [PSNP-Public Works Programme]
,SUM(IIF([resettled]='Yes',1,0)) as [Resettled by the government]
,SUM(IIF([ikp]='Yes',1,0)) as [IKP-Credit provision] ,SUM(IIF([ikp_child]='Yes',1,0)) as [IKP-Credit
provision for children] ,SUM(IIF([nregs]='Yes',1,0)) as [Household has a job card under the NREGS]
,SUM(IIF([nregs_allow]='Yes',1,0)) as [Unemployment allowance under NREGS]
,SUM(IIF([nregs_work]='Yes',1,0)) as [Worked for NREGS] ,SUM(IIF([pds]='Yes',1,0)) as [Access to
Public Distribution System] ,SUM(IIF([rajiv]='Yes',1,0)) as [Rajiv/NTR Arogyasri card]
,SUM(IIF([sabla]='Yes',1,0)) as [SABLA programme] ,SUM(IIF([sabla_yl]='Yes',1,0)) as [REGSEAG/SABLA
for children]
,SUM(IIF([beca_yl]='Yes',1,0)) as [Beca 18 programme] ,SUM(IIF([bonograt]='Yes',1,0)) as [Bono de
Gratitud/Pension 65 programme] ,SUM(IIF([insur_yl]='Yes',1,0)) as [YL child has Health Insurance]
,SUM(IIF([juntos]='Yes',1,0)) as [Juntos programme] ,SUM(IIF([minsa_yl]='Yes',1,0)) as [Partial SIS /
SIS independiente (MINSA)] ,SUM(IIF([projoven_yl]='Yes',1,0)) as [ProJoven/ Jovenes a la Obra
programme] ,SUM(IIF([sisgrat_yl]='Yes',1,0)) as [SIS Gratuito programme]

,SUM(IIF([molisa06]='Yes',1,0)) as [MOLISA-2006] ,SUM(IIF([molisa09]='Yes',1,0)) as [MOLISA-2009]
,SUM(IIF([molisa10]='Yes',1,0)) as [MOLISA-2010] ,SUM(IIF([molisa11]='Yes',1,0)) as [MOLISA-2011]
,SUM(IIF([molisa12]='Yes',1,0)) as [MOLISA-2012] ,SUM(IIF([molisa13]='Yes',1,0)) as [MOLISA-2013]
,SUM(IIF([molisa14]='Yes',1,0)) as [MOLISA-2014] ,SUM(IIF([molisa15]='Yes',1,0)) as [MOLISA-2015]
,SUM(IIF([molisa16]='Yes',1,0)) as [MOLISA-2016]

FROM (Select DISTINCT chilid, countryid, typesite, chsex, round, yc, agemon, [hhsiz], [eap], [hep]
,[othprog], [psnp_ds], [psnp_pw], [resettled]
,[ikp], [ikp_child], [nregs], [nregs_allow], [nregs_work], [pds], [rajiv], [sabla], [sabla_yl]
,[beca_yl], [bonograt], [insur_yl], [juntos], [minsa_yl], [projoven_yl], [sisgrat_yl]
,[molisa06], [molisa09], [molisa10], [molisa11], [molisa12], [molisa13], [molisa14], [molisa15]
,[molisa16]

, dbCare.Region.region

FROM dbCare.ch_Identity INNER JOIN dbCare.Region ON dbCare.ch_Identity.regionid =
dbCare.Region.regionid INNER JOIN dbCare.hh_Main ON dbCare.ch_Identity.child_round_id =
dbCare.hh_Main.child_round_id INNER JOIN dbCare.hh_PublicProgrammes ON
dbCare.ch_Identity.child_round_id = dbCare.hh_PublicProgrammes.child_round_id ) as A
GROUP BY countryid, region, chsex, typesite, round, yc
;
GO
COMMIT TRANSACTION;

```

- Result Review – In Peru, SIS Gratuito and Juntos programmes has helped a large number of households. In Vietnam, the number of households getting benefitted through MOLISA-yearly plan are gradually reducing. In 2006, MOLISA helped 545 households which consistently reduced over the years to 342 in 2016. In India, Rajiv/NTR Arogyasri card, NREGS job card or worked for NREGS and IKP-credit provision has largely utilized public programmes. In Ethiopia, Health extension programme and PSNP public works programme have been utilized largely.

## 12. Report 12 - zREP12\_VictimsofVariousShocks

```

USE CP001;
GO

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
GO

BEGIN TRANSACTION;
GO

/************************************************************/
/*Report 12 Creation - zREP12_VictimsofVariousShocks*/
/************************************************************/
-- Country & Region wise - Victims of Various types of Shocks

CREATE VIEW zREP12_VictimsofVariousShocks
AS

SELECT CASE countryid WHEN 'ET' THEN 'Ethiopia' WHEN 'IN' THEN 'India' WHEN 'PE' THEN 'Peru' WHEN 'VN' THEN 'Vietnam'
END as Country,

region as Region, typesite as [Site Type], chsex as Sex,
round as Round, yc as [Younger/Older Cohort],
COUNT(childid) as [Total Children], AVG([hhsize]) as [Avg Household size] ,SUM(IIF([shcrime1]='Yes' OR
[shcrime2]='Yes' OR [shcrime3]='Yes' OR [shcrime4]='Yes' OR [shcrime5]='Yes' OR [shcrime6]='Yes' OR [shcrime7]='Yes'
OR [shcrime8]='Yes',1,0)) as [Type A: Victims of Crimes]
,SUM(IIF([shregul1]='Yes' OR [shregul2]='Yes' OR [shregul3]='Yes' OR [shregul4]='Yes' OR [shregul5]='Yes' OR
[shregul6]='Yes',1,0)) as [Type B: Victims of Regulations/Actions]
,SUM(IIF([shecon1]='Yes' OR [shecon2]='Yes' OR [shecon3]='Yes' OR [shecon4]='Yes' OR [shecon5]='Yes' OR
[shecon6]='Yes' OR [shecon7]='Yes' OR [shecon8]='Yes' OR [shecon9]='Yes' OR [shecon10]='Yes' OR [shecon11]='Yes' OR
[shecon12]='Yes' OR [shecon13]='Yes' OR [shecon14]='Yes',1,0)) as [Type C: Victims of Economic Turmoil]
,SUM(IIF([shenv1]='Yes' OR [shenv2]='Yes' OR [shenv3]='Yes' OR [shenv4]='Yes' OR [shenv5]='Yes' OR [shenv6]='Yes' OR
[shenv7]='Yes' OR [shenv8]='Yes' OR [shenv9]='Yes' OR [shenv10]='Yes' OR [shenv11]='Yes' OR [shenv12]='Yes' OR
[shenv13]='Yes',1,0)) as [Type D: Victims of Natural Disasters]
,SUM(IIF([shhouse1]='Yes' OR [shhouse2]='Yes' OR [shhouse3]='Yes',1,0)) as [Type E: Victims of Accidents in Building]
,SUM(IIF([shfam1]='Yes' OR [shfam2]='Yes' OR [shfam3]='Yes' OR [shfam4]='Yes' OR [shfam5]='Yes' OR [shfam6]='Yes' OR
[shfam7]='Yes' OR [shfam8]='Yes' OR [shfam9]='Yes' OR [shfam10]='Yes' OR [shfam11]='Yes' OR [shfam12]='Yes' OR
[shfam13]='Yes' OR [shfam14]='Yes' OR [shfam15]='Yes' OR [shfam16]='Yes' OR [shfam17]='Yes' OR [shfam18]='Yes',1,0))
as [Type F: Victims of Family Issues]
,SUM(IIF([shother]='Yes',1,0)) as [Type G: Victims of other issues]

FROM (Select DISTINCT childid, countryid, typesite, chsex, round, yc, agemon, [hhsize] ,[shcrime1] ,[shcrime2]
,[shcrime3],[shcrime4],[shcrime5],[shcrime6],[shcrime7],[shcrime8]
,[shregul1],[shregul2],[shregul3],[shregul4],[shregul5],[shregul6]
,[shecon1],[shecon2],[shecon3],[shecon4],[shecon5],[shecon6],[shecon7],[shecon8],[shecon9],[shecon10]
,[shecon11],[shecon12],[shecon13],[shecon14]
,[shenv1],[shenv2],[shenv3],[shenv4],[shenv5],[shenv6],[shenv7],[shenv8],[shenv9],[shenv10],[shenv11]
,[shenv12],[shenv13]
,[shhouse1],[shhouse2],[shhouse3]
,[shfam1],[shfam2],[shfam3],[shfam4],[shfam5],[shfam6],[shfam7],[shfam8],[shfam9],[shfam10],[shfam11]
,[shfam12],[shfam13],[shfam14],[shfam15],[shfam16],[shfam17],[shfam18]
,[shother]
, dbCare.Region.region
GROUP BY countryid, region, chsex, typesite, round, yc ;
GO
COMMIT TRANSACTION;

```

- Result Review – Highest number of victims have undergone Type C – Economic turmoil or Type D – Natural disaster or Type-F – Family issues.

### 1.6.3 Stored Procedures (if any)

#### 1. Stored Procedure 1 - Number of Children per TypeSite

```
-- =====
-- SP1 - Number of Children per TypeSite
-- =====
-- Author:      <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <ChildrenPerTypeSite>
-- =====
CREATE PROCEDURE spChildrenPerTypeSite
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT typesite, count(distinct childid) as Num_of_Children
        FROM dbo.v01_WellBeing
        GROUP BY typesite
END
GO
```

#### 2. Stored Procedure 2 - Children per Typesite (Parameterized)

```
-- =====
-- SP2 - Search facility - Children per Typesite (Parameterized)
-- =====
-- Author:      <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <ChildrenPerTypeSite>
-- Values that can be passed: Rural / Urban
-- =====
CREATE PROCEDURE spFind_ChildrenPerTypeSite
    @TypeSite varchar(10)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT distinct countryid, typesite, childid
        FROM dbo.v01_WellBeing
        WHERE typesite = @TypeSite
        ORDER BY countryid
END
GO
```

#### 3. Stored Procedure 3 - Children under the weight, height & bmi (Parameterized)

```
-- =====
-- SP3 - Search facility - Children under the weight, height & bmi (Parameterized)
-- =====
-- Author:      <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <Count of children less than given input>
-- Values that can be passed: Any Integers
-- =====
CREATE PROCEDURE spFind_ChildrenlessthanWHB
    @inputweight int,
    @inputheight int,
    @inputbmi int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT countryid, typesite, chsex, count(distinct childid) as [Number of children]
        FROM dbo.v01_WellBeing
        WHERE chweight < @inputweight
            AND chheight < @inputheight
            AND bmi < @inputbmi
        GROUP BY countryid, typesite, chsex
END
GO
```

#### 4. Stored Procedure 4 - Number of Cohorts who received funding by Age Range (Parameterized)

```
-- =====
-- SP4 - Search facility - Number of Cohorts who received funding by Age Range (Parameterized)
-- =====

CREATE PROCEDURE spFind_Cohorts_AgeRange_Funding
    @yc varchar(2),
    @agemonmin tinyint,
    @agemonmax tinyint
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT countryid, yc, entype, count(distinct childid) as [Number of children]
    FROM [dbo].[v02_Literacy]
    WHERE yc = @yc
        AND agemon BETWEEN @agemonmin AND @agemonmax
    GROUP BY countryid, yc, entype
END
GO
```

#### 5. Stored Procedure 5 - Number of Cohorts who speak a language by Age Range (Parameterized)

```
-- =====
-- SP5 - Search facility - Number of Cohorts who speak a language by Age Range (Parameterized)
-- =====

CREATE PROCEDURE spFind_Cohorts_AgeRange_Language
    @yc varchar(2),
    @language varchar(20),
    @agemonmin tinyint,
    @agemonmax tinyint
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT countryid, yc, chlang, count(distinct childid) as [Number of children]
    FROM [dbo].[v02_Literacy]
    WHERE yc = @yc
        AND chlang = @language
        AND agemon BETWEEN @agemonmin AND @agemonmax
    GROUP BY countryid, yc, chlang
END
GO
```

#### 6. Stored Procedure 6 - Top 15 children - Travel time to school by Grade (Parameterized)

```
-- =====
-- SP6 - Search facility - Top 15 children - Travel time to school by Grade (Parameterized)
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <Top 15 children - Travel time to school by Grade>
-- =====

CREATE PROCEDURE spFind_TravelTimettoSchool_Grade
    @engrade varchar(20)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select top 15 countryid, childid, engrade, max(timesch) / 60 as [travel time to school (in hours)]
    from [dbo].[v02_literacy]
    Where engrade = @engrade
    group by countryid, childid, engrade
    order by max(timesch) desc
END
GO;
```

## 7. Stored Procedure 7 - Number of Cohorts with Guardian's age range (Parameterized)

```
-- =====
-- SP7 - Search facility - Number of Cohorts with Guardian's age range (Parameterized)
-- =====
CREATE PROCEDURE spFind_Cohorts_Guardian_AgeRange
    @yc varchar(2),
    @careagemin int,
    @careagemax int,
    @carecantread varchar(3),
    @dadagemin int,
    @dadagemax int,
    @dadcantread varchar(3),
    @momagemin int,
    @momagemax int,
    @momcantread varchar(3)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT countryid, yc, childid,
           careage, carecantread,
           momage, momcantread,
           dadage, dadcantread
      FROM [dbo].[v03_LevelOfCare]
     WHERE yc = @yc
       AND carecantread = @carecantread
       AND momcantread = @momcantread
       AND dadcantread = @dadcantread
       AND careage BETWEEN @careagemin AND @careagemax
       AND dadage BETWEEN @dadagemin AND @dadagemax
       AND momage BETWEEN @momagemin AND @momagemax
END
GO
:
```

## 8. Stored Procedure 8 - Ethiopia's Public Programmes (Parameterized)

```
-- =====
-- SP8 - Search facility - Ethiopia's Public Programmes (Parameterized)
-- =====
CREATE PROCEDURE spFind_ET_Household_Economic_KPI
    @resettled varchar(3),
    @eap varchar(3),
    @hep varchar(3),
    @othprog varchar(3),
    @psnp_ds varchar(3),
    @psnp_pw varchar(3)

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select
        countryid,
        childid,
        eap,
        hep,
        othprog,
        psnp_ds,
        psnp_pw,
        resettled
       FROM dbo.v05_PublicProgrammes
      WHERE resettled = @resettled
        AND eap          = @eap
        AND hep         = @hep
        AND othprog     = @othprog
        AND psnp_ds     = @psnp_ds
        AND psnp_pw     = @psnp_pw
        AND countryid = 'ET'

END
GO
:
```

#### 1.6.4 Triggers (if any)

- Triggers have been created to prevent deletion of any records. However, it is implied that the existing records can be updated or even new records can be added as part of the scalability.
- All the transactional data tables have triggers against deleting records.

```
-- =====
-- Trigger 1 - Intercept Deletion
-- =====
-- Author:      Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion
ON [dbCare].[ch_Education]
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO
```

```
-- =====
-- Trigger 2 - Intercept Deletion
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion2
ON [dbCare].[ch_Identity]
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion2` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO
```

```
-- =====
-- Trigger 3 - Intercept Deletion
-- =====
-- Author:      Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion3
ON [dbCare].[ch_MedicalHistory]
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion3` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO
```

```

-- =====
-- Trigger 4 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:     Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion4
    ON [dbCare].[ch_Guardian]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion4` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

```

-- =====
-- Trigger 5 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:     Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion5
    ON [dbCare].[hh_Main]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion5` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

```

-- =====
-- Trigger 6 - Intercept Deletion
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion6
    ON [dbCare].[hh_PublicProgrammes]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion6` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

```

-- =====
-- Trigger 7 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:    Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion7
ON [dbCare].[hh_LivestockOwnership]
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion7` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

```

-- =====
-- Trigger 8 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:    Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCare.trg_Intercept_Deletion8
ON [dbCare].[hh_Shocks]
INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion8` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

### 1.6.5 User Define Functions (if any)

#### 1. Table-valued function 1 - udf\_Ethnicity\_Counter

```
-- =====
-- UDF1 - udf_Ethnicity_Counter
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <Input the country id and ethnicity which returns the filtered table>
-- =====
CREATE FUNCTION udf_Ethnicity_Counter
(
    -- Add the parameters for the function here
    @countryid varchar(2),
    @chethnic varchar(20)
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT countryid, chethnic, count(distinct childid) as Num_of_Children
    FROM dbo.v01_WellBeing
    where countryid = @countryid and chethnic = @chethnic
    group by countryid, chethnic
)
GO
```

#### 2. Table-valued function 2 - udf\_Find\_Child\_Gen\_Characteristics

```
-- =====
-- UDF2 - udf_Find_Child_Gen_Characteristics
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <To find the general characteristics based on childid and roundid>
-- =====
CREATE FUNCTION udf_Find_Child_Gen_Characteristics
(
    -- Add the parameters for the function here
    @childid varchar(10),
    @round tinyint
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT *
    FROM [dbCare].[ch_Identity]
    where childid = @childid and round = @round
)
GO
```

### 3. Scalar-valued function 1 - udf\_Find\_Religion

```
-- =====
-- UDF3 - udf_Find_Religion
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <To bring the scalar value of Religion name from Religion ID>
-- =====
CREATE FUNCTION udf_Find_Religion
(
    -- Add the parameters for the function here
    @chldrelid int
)
RETURNS int
AS
BEGIN
    -- Declare the return variable here
    DECLARE @chldrel varchar(20)

    -- Add the T-SQL statements to compute the return value here
    select @chldrel = chldrel from dbCare.Religion
    where chldrelid = @chldrelid

    -- Return the result of the function
    RETURN @chldrel
END
GO
```

### 4. Table-valued function 3 - udf\_Find\_total\_livestock

```
-- =====
-- UDF4 - udf_Find_total_livestock
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <To find the total livestock in the last 12 months>
-- =====
CREATE FUNCTION udf_Find_total_livestock
(
    -- Add the parameters for the function here
    @countryid varchar(2),
    @childid varchar(10)
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    select A.countryid,
        A.childid,
        sum(iif(J.aniany = 'Yes', 1, 0)) as Any_Live_Stock_in_12_months
    from dbCare.ch_Identity A INNER JOIN dbCare.hh_LivestockOwnership J
    ON A.child_round_id = J.child_round_id
    WHERE A.countryid = @countryid AND A.childid = @childid
    group by A.countryid, A.childid
    having sum(iif(J.aniany = 'Yes', 1, 0)) > 0
)
GO
```

## 5. Table-valued function 4 - udf\_Find\_total\_livestock\_details

```
-- =====
-- UDF5 - udf_Find_total_livestock_details
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <01-May-2021>
-- Description: <To find the total livestock in the last 12 months list all>
-- =====
CREATE FUNCTION udf_Find_total_livestock_details
(
    -- Add the parameters for the function here
    @countryid varchar(2),
    @childid varchar(10)
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    select A.countryid,
           A.childid,
           sum([anibeeh]) as Num_of_BeeHives
      ,sum([anibusfm]) as Num_of_modern_Buffalos
      ,sum([anibusft]) as Num_of_traditional_Buffalos
      ,sum([anibull]) as Num_of_Bullocks
      ,sum([anicalv]) as Num_of_calves
      ,sum([anicam]) as Num_of_camels
      ,sum([anicowm]) as Num_of_modern_cows
      ,sum([anicowt]) as Num_of_traditional_cows
      ,sum([anidonk]) as Num_of_donkeys
      ,sum([anidrau]) as Num_of_draught_animals
      ,sum([anifish]) as Num_of_fish
      ,sum([anifshr]) as Num_of_fisheries
      ,sum([anigoat]) as Num_of_goats
      ,sum([aniguin]) as Num_of_guinea_pigs
      ,sum([anihebu]) as Num_of_hebuffalos
      ,sum([aniheif]) as Num_of_heifers
      ,sum([anillam]) as Num_of_llamas
      ,sum([animilk]) as Num_of_Milk_Animals
      ,sum([aniothr]) as Num_of_other
      ,sum([anioxen]) as Num_of_oxen
      ,sum([anipigs]) as Num_of_pigs
      ,sum([anipoul]) as Num_of_poultry
      ,sum([anirabb]) as Num_of_rabbits
      ,sum([anirumij]) as Num_of_small_ruminants
      ,sum([anishee]) as Num_of_sheep
      ,sum([anishri]) as Num_of_shrimp
      ,sum([anisnai]) as Num_of_snails
      ,sum([anispec]) as Num_of_other_animals
      ,sum([anybul]) as Num_of_young_bulls
        from dbCare.ch_Identity A INNER JOIN dbCare.hh_LivestockOwnership J
       ON A.child_round_id = J.child_round_id
      WHERE A.countryid = @countryid AND A.childid = @childid
        group by A.countryid, A.childid
        having sum(iif(J.aniany = 'Yes', 1, 0)) <> 0
)
GO
```

## 1.7. Report Design (Only for task 2 and task 3)

N/A

## 1.8. Database Security

- Database security is important to ensure deter external attacks like ransomware, or even firewall penetrating and to keep the sensitive information safe. It is important to follow the guidelines of GDPR to protect the data within.
- In our database, we created a role for accessing stored procedures only and also a user is created to be able to access only the reports generated.
- In order to make the database more secure, according to the principle of least privilege, we should restrict the permissions of the service accounts. Based on the SQL server roles like for example, sysadmin, serveradmin, securityadmin, dbcreator, public, etc.

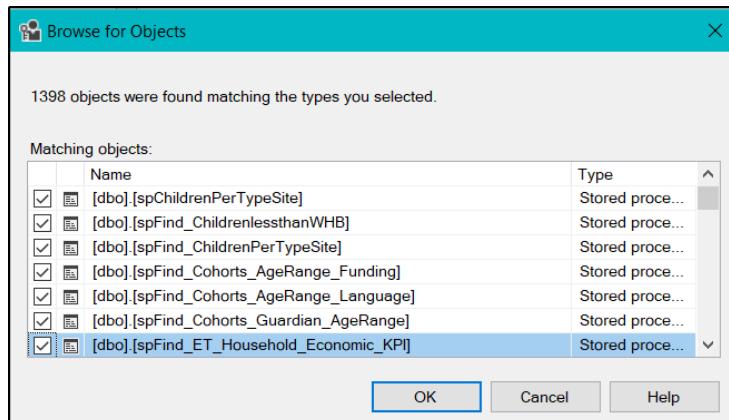
```
USE CP001;
GO

-- =====
-- Create & grant permissions of 'Stored-Procedure-only' Access to a user
-- =====

create role db_sp_Access_only ;

grant execute to db_sp_Access_only;

-- After this is done, the role db_sp_Access_only is opened and
-- assigned the respective securables (datareader and sp only)
```



- Using the system tables we validate the locks given.

```
/*
*Validate the locks using sys tables*
*/

USE CP001;
GO

SELECT a.name, j.name, p.*
FROM sys.dm_tran_locks p
INNER JOIN sys.databases a ON p.resource_database_id = database_id
INNER JOIN sys.objects j ON p.resource_associated_entity_id = object_id
```

	name	name	resource_type	resource_subtype	resource_database_id	resource_description	resource_associated_entity_id	resource_lock_partition	request_mode	request_type	request_status
1	CP001	syschobje	OBJECT		9		34	0	Sch-S	LOCK	GRANT
2	master	sysobjvalues	OBJECT		1		60	0	Sch-S	LOCK	GRANT
3	master	syssingleobjrefs	OBJECT		1		74	0	Sch-S	LOCK	GRANT

## 1.9 Database Backup and Restore Strategy

- A periodical backup plan can be setup, which is done in SQL EXPRESS edition manually, whereas using DEVELOPER edition, we can setup a Maintenance plan in which an automatic scheduler can be setup to perform periodical backup of the database.
- For backup, we may choose either Full or Differential mode, or even overwriting or appending to the original database backup, or perhaps even erase backup can be performed.
- We may backup to either Disk or URL. Usually when we store the data in the cloud, we may sync our cloud with SSMS and store the backup over there to ensure high security and availability.
- In order to check the reliability, we can verify the backup when it is finished, also if an error is thrown, we can enable the backup to continue.
- Depending on our needs, we may choose to set backup compression, so that we optimize our disk / datastore memory.
- Backup can be historically saved with a timestamp, and using any of the timestamp, we can restore a database.

## 1.10 Data Science/Business Intelligence Techniques

- Business Intelligence Techniques using SQL/related tools –
  - DAR Approach (Dashboards, Analysis, Reports), data blending using multiple data sources.
  - Slicing and dicing, to create necessary reports from DB level itself.
  - Analyse the types of dimensions like slowly changing dimensions for example, and then build or query the model based on the type of metadata.
  - Data warehousing – SQL databases (OLTP – Online Transaction Processing) and SQL data warehouses (OLAP – Online Analytical Processing).
  - BI techniques can be applied at Time Intelligence level – using Power BI or even SQL directly, time intelligence can be a powerful feature and can provide excellent key performance indicators which can be projected as well as forecasted meaningfully. For example:
    - Year-to-Date which has the aggregated expression from the beginning of the current year.
    - Year-on-Year which can be used to calculate cumulative growth.
    - Moving annual totals / averages – which can be aggregated cumulative data from this time last year, etc.
  - Additional BI techniques include data mining, model visualization, ETL like in SAP (Extract, Transform, Load), or ELT like in Snowflake (Extract, Load, Transform), etc.
- Data Science Techniques –
  - SQL can be connected to multiple Data Science based tools like Python, R etc. In Python, there is a standard library known as SQLite3 which needs to be imported into the interface and SQL commands can be given to explore the data.
  - SQL can be connected to R studio using 3 packages – RODBC, DBI, ODBC
  - Additional Data Science techniques include predictive modelling, statistical analysis, text mining or even Sentiment analysis, etc.
  - Other techniques using algorithms can also be linked with SQL data like classification, regression, etc.

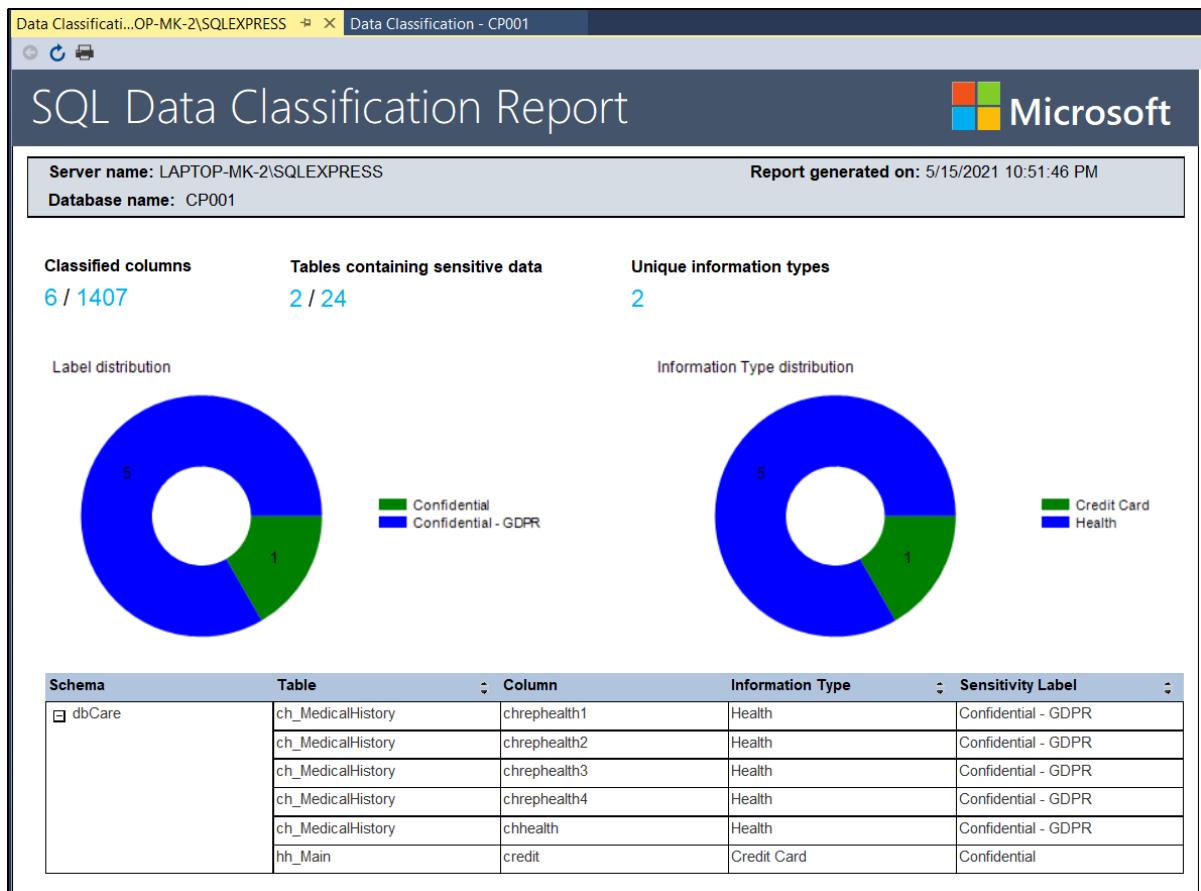
## 1.11 Data Privacy, Ethical and legal issues

- [13] SSMS contains features that are internet-enabled, so they can collect and share anonymous feature usage and diagnostic data to Microsoft.
- [14] In order to comply with GDPR (General Data Protection Regulation policy), Microsoft recommends GDPR compliance using 4 steps – Discover, Manage, Protect & Report.
  - In Discover phase, we discover sensitive data – which table, which database, which object has it – everything needs to be documented. Based on the levels of sensitivity of the personal data, these fields need to be classified.
  - In Manage phase, once all the data locations are documented, we proceed to understand which personal data is accessed by whom and then limit it.
  - In Protect phase, once the present security practices are understood, the efforts to protect the data begin. The goal is to reduce risk and reduce the impact to data because of security and monitoring. Various encryption methods can be used in this phase in order to protect the data.
  - In Report phase, 2 principles apply – transparency and due diligence where in the data is retained of all activities about personal data (metadata only).
- [14] In order to meet the GDPR criteria, an organization must strive to meet the ‘Data protection impact assessment’ which is prerequisite of the GDPR (**GDPR article 35**).
- [14] From SQL point of view – some of the best recommendations / practices are:
  - Windows authentication over SQL server authentication to enable centralized management via Active directory.
  - To use discrete accounts to authenticate users and applications which minimizes risks of malicious activity which can include SQL injection attacks.
- There are ways of protecting data within SSMS. In order to do this, we must discover the data first and later classify it into levels of confidentiality.

The screenshot shows the 'Data Classification - CP001' interface. At the top, there are buttons for Save, Add Classification, View Report, and Refresh. Below this, a message says '36 columns with classification recommendations (click to minimize)'. A section titled '0 classified columns' includes a link to 'Learn more – Getting Started Guide'. Below this is another message: '36 columns with classification recommendations (click to minimize)'. A button labeled 'Accept selected recommendations' is visible. The main table lists 36 columns across various schemas and tables, each with its information type and sensitivity label set to 'Health' or 'Credit Card'.

Schema	Table	Column	Information Type	Sensitivity Label
dbCare	ch_MedicalHistory	chhealth	Health	Confidential
dbCare	ch_MedicalHistory	chrephalth1	Health	Confidential
dbCare	ch_MedicalHistory	chrephalth2	Health	Confidential
dbCare	ch_MedicalHistory	chrephalth3	Health	Confidential
dbCare	ch_MedicalHistory	chrephalth4	Health	Confidential
dbCare	hh_Main	credit	Credit Card	Confidential

Figure 1.1 – Data Discovery & Classification



**Figure 1.2 – Data Classification Report**

## 1.12 Conclusion

- An average of 147 number of children per country per sentinel site have been interviewed. There are totally 20 sentinel sites in each of the 4 countries.
- Child deaths from ethnicity background like Ethiopia - Amhara, India - BC, Peru - Mestizo, Ethiopia - Tigrian, Ethiopia - Oromo have occurred more than any other ethnicity.
- Average BMI is less than 15 for 3 countries, which means they are severely underweight. Only India has an average BMI of 21.69 from the given numbers which can be considered as normal.
- The subjective well-being of all the caregivers has declined from the last 4 years, also individually declined for each country.
- Almost all the 4 countries are doing fairly better with vaccinations for especially BCG, Measles and Polio. For DPT however, it is still improving, but not so much as it is for the other 3 vaccines mentioned earlier. HIB and Tetanus are severely undermined which needs to be prioritized first.
- Most of them can read and write sentences, with occasional help. The number of literates improved overall in the first 3 rounds in each of the countries for the children, whereas the number of illiterates dropped in round 2 but gradually increased in round 3 in all the countries.
- Ethiopia has the highest number of children with the number of years lagging behind the average Grade 1 student (1.83), and the regions like Amhara and Oromiya needs to be targeted to improve first, and also the males are lagging behind more than the females. Similarly Northern Uplands in Vietnam, Telangana in India have children who are lagging behind.
- A vast majority of Vietnamese children do not have knowledge on reproductive health. Most of the OC from Vietnam and Ethiopia don't even know what is a condom. Ethiopian children know well about STDs than any other children.

- There have been 323 deaths of fathers of children over the 5 rounds in Ethiopia, and 231 deaths in India which are more prominent than Peru and Vietnam. The fathers who are highly undereducated are from Ethiopia and highly educated are from Peru.
- The mothers who lack the ability to read are mostly either from Ethiopia and India.
- Majority of the Ethiopian children do not attend pre-primary school.
- Around 1400 instances were present where the survey answered that the child might die in each of the 3 countries Ethiopia, India and Peru and at least more than 1000 of them are YC.
- Highest alcohol addicts are seen in Ethiopia.
- Highest smoking addicts are seen in Peru.
- Peru has an abnormal average child weight, and an abnormal average height.
- A majority of permanently disabled children are YC, the highest number of disabled are present in Vietnam. Long term health issues are highly seen in Peru, followed by Vietnam.
- India ranks worst in the subjective well-being with only an average of 2.99.
- Highest serious injuries and highest serious illnesses are caused in India only.
- Coastal Andhra is a region where children take the maximum amount of time to commute to school, and India seems to be majorly lacking in providing schooling in the nearby areas for these children, in addition, most of the children spend their time in school itself in India.
- SNNP, Amhara, Oromiya in Ethiopia, Costa, Selva, Sierra regions in Peru have the most children spending their time more for caring for household members. The children in Peru spend more time in paid activity than any other country. In Somali, the average hours spent in studying outside school is 5 hours. Average travel time is more in the regions like Afar, Oromiya from Ethiopia, other regions from India and Highlands from Vietnam.
- India is lacking behind in terms of access to sanitation.
- Ethiopian and Vietnamese people mostly don't eat enough in the household.
- Vietnam majorly doesn't have proper access to safe drinking water.
- Ethiopia largely lacks electricity and fuels for cooking.
- Indian and Vietnamese households have majorly obtained loans or credit.
- The average consumer durables index is 0.25 in Ethiopia, which is quite less.
- Average housing quality index is only 0.31 in Ethiopia, and 0.34 in Peru.
- Average number of males and females aged 61 and above are present in larger number of households, especially in India.
- In Peru, SIS Gratuito and Juntos programmes has helped a large number of households.
- In Vietnam, the number of households getting benefitted through MOLISA-yearly plan are gradually reducing. In 2006, MOLISA helped 545 households which consistently reduced over the years to 342 in 2016.
- In India, Rajiv/NTR Arogyasri card, NREGS job card or worked for NREGS and IKP-credit provision has largely utilized public programmes.
- In Ethiopia, Health extension programme and PSNP public works programme have been utilized largely.
- Highest number of victims have undergone Type C – Economic turmoil or Type D – Natural disaster or Type-F – Family issues.

# Task 2 - Database and front-end reporting for Inequality in Education – Vietnam (2016-17)

## 2.1 Abstract

Young Lives specifically focused on the aspect of disparity within the education sector in Vietnam. For this they have planned a survey and organized a fieldwork for 2 years starting from 2016-17. Young Lives is being funded by international donors who work in cooperation with the Government of Vietnam. As part of this study, the data is being collected from Grade 10 students, Grade 10 teachers, and upper secondary schools. A total of 8,740 children have been chosen for this project within 52 schools across 5 provinces in Vietnam. Each student was asked to fill a questionnaire that includes their background, opinions on various topics, etc. The surveys happened twice - once between 12<sup>th</sup> September and 18<sup>th</sup> October 2016 and once between 6<sup>th</sup> March and 15<sup>th</sup> April of 2017 in all provinces except Phu Yen where the field work will take place until 20 April 2017. Students were also asked to complete an English test and a Math test. This cognitive testing happened twice – once in 2016 and once in 2017 to compare their change of ability in 1 year, before the surveys ended. Students may voluntarily participate or withdraw at any given time.

## 2.2 Introduction

- A database named ‘CP002’ is created in which we store all the YL Vietnam data from 2016 to 2017.
- Initially 2 .tab files were given in raw format, which needs to be imported into SSMS.
- Mode of raw source data upload – Direct import using ‘SSMS Import and Export wizard’.

Option	Input Value
Source format of the files	Flat file
Header row delimiter	Tab {t}
Destination	Microsoft OLE DB Provider for SQL Server
Server name	LAPTOP-MK-2\SQLEXPRESS
Authentication	Windows / SQL server authentication
Database	CP002
Schema	dbo
Total number of files	2
Names of the raw tables	[dbo].[vietnam_wave_1] [dbo].[vietnam_wave_2]

## 2.3 Relational Schema

### 2.3.1 Identify the tables

- A different approach has been taken when each of the columns from the raw data tables have been analyzed. It seems that the questionnaires in 2016 and 2017 are entirely different, implying it is a way of looking at how the student has improved within a year.
- The only fields that are common between the two raw tables are UniqueID, ClassID, StudentID, SchoolID besides cognitive tests for Math and English which occurred twice to compare their abilities.
- Due to the given structure of the data from the surveys, each of the tables have been segmented and created one for each of the years 2016 and 2017.
- A schema has been defined = ‘dbEdu’

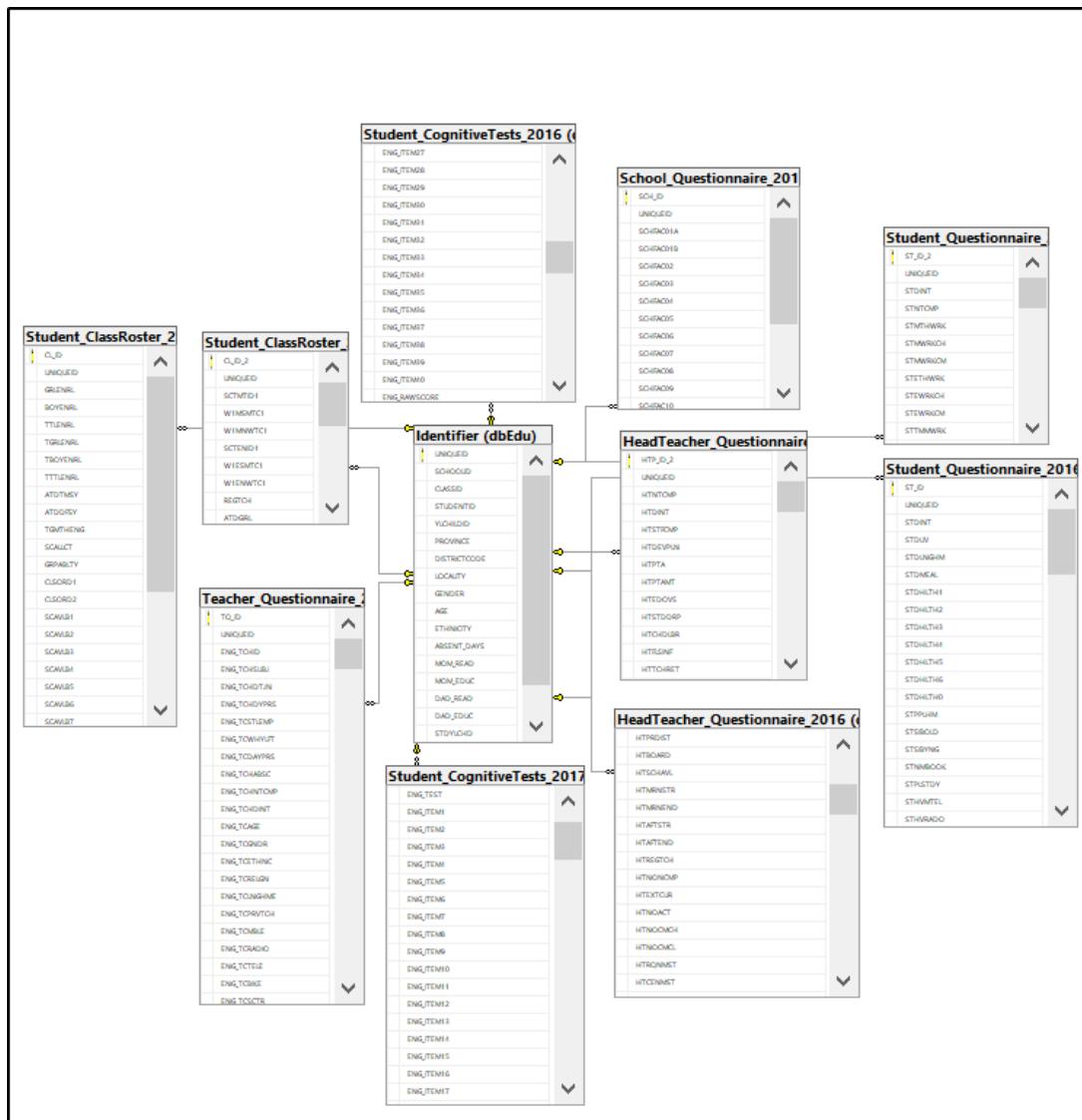
### 2.3.2 Create the tables

- Before creating the tables, the maximum amount of row-length was checked for each of the fields given, in order to determine which datatype would be suitable, also a thorough check was performed before a datatype is finalized for a field.

- The following tables have been created in a way we could identify which questions have been answered in which year.
  - [dbEdu].[Identifier]
  - [dbEdu].[Student\_Questionnaire\_2016]
  - [dbEdu].[Student\_Questionnaire\_2017]
  - [dbEdu].[Student\_ClassRoster\_2016]
  - [dbEdu].[Student\_ClassRoster\_2017]
  - [dbEdu].[Student\_CognitiveTests\_2016]
  - [dbEdu].[Student\_CognitiveTests\_2017]
  - [dbEdu].[HeadTeacher\_Questionnaire\_2016]
  - [dbEdu].[HeadTeacher\_Questionnaire\_2017]
  - [dbEdu].[School\_Questionnaire\_2016]
  - [dbEdu].[Teacher\_Questionnaire\_2017]

### 2.3.3 Create relationships between tables

- All these tables are centrally connected to [dbEdu].[Identifier].
- [dbEdu].[Identifier] table has the primary key – UNIQUEID, which is used as a primary key in this build.
- Later all the other tables refer to this key using a foreign key within their tables.
- Foreign key constraints are defined to ensure data accuracy.



## 2.4 Design Rationale

- In order to build a comprehensive relational model here, I had to consider the columns separately because there is no commonality between 2016 and 2017 except cognitive tests.
- However, each of the surveys have been internally segmented to form groups that have related questions asked in the survey. For example, Identifier table - We can extract student's province, rural/urban living, gender, age, ethnicity, number of days absent, whether the mother and father are from an educated background, if he/she is a YL child, etc.
- There are more than 1000 columns including both 2016 and 2017 surveys, which were logically split into 6 segments, without disrupting the completeness or the integrity of the information.
- We must begin with the identifier table in order to centralize the information, and later we could begin from any of these segments to form the model.
- The central node / primary key of UNIQUEID is present in the Identifier table and all the other tables refer to this tabular key while having a constraint over their UNIQUEID as foreign keys.
- According to this design, the views were linked / joined to give rise to new insights where multiple segments can be analysed together. For example, v01\_ST\_Questionnaire is a view where both the Student Questionnaire tables (2016 and 2017) are connected with Identifier table and now we have province-wise or even site-wise information of each student who has undergone this questionnaire.
- All the views are based on similar logic, except for v06\_ST\_Cogn\_Tests which is not a join between 2 tables, instead it is a unionized table which has the same columns in both the cognitive tests tables of 2016 and 2017.
- All the reports are basically views, however, they have an underlying subquery / sub queries that could have joined, unionized or even pivoted/unpivoted a combination of tables or views or even both sometimes as part of the blend. Each of these reports have aggregations based on the preliminary exploratory analysis using the raw table.
- A schema 'dbEdu' is also defined here to show that this logical collection of database objects is segregated, if needed for multiple applications, accessibility, managing database security, etc. In our case, a schema owner can use only the database tables as the schema is defined for tables alone.
- In terms of performance, this model is not very scalable because if new fields get added every year, then going forward we would have many joins in order to bring forward historical data, and hence there would be a performance bottleneck. This model is built assuming that the survey ends in 2017 and the build is stable to handle many more students perhaps even from different countries.
- The security has been handled by giving necessary access only to a test\_user.

## 2.5 Design Considerations

### 2.5.1 Database Normalisation

- This database has been designed to form a semi 2NF – 3NF normalized model, wherein the same uniqueid has been present in multiple tables as foreign keys but only the Identifier table has it as a primary key.
- The remaining tables also have autogenerated primary keys using IDENTITY (1,1) which would enable the tables to fit in the schema.
- Only the cognitive tests have common fields which were also separated due to transferable skills tests which have taken place only in 2017.

### 2.5.2 Constraints

- Constraints are created during transaction data tables creation. All the constraints shown below are referencing only the [dbEdu].[Identifier] table which has the primary key – UNIQUEID

Constraint	Present in table	Referencing table(key)
FK_STC_ID	Student_CognitiveTests_2016	[dbEdu].[Identifier](UNIQUEID)
FK_STC_ID_2	Student_CognitiveTests_2017	[dbEdu].[Identifier](UNIQUEID)

FK_ST_ID	Student_Questionnaire_2016	[dbEdu].[Identifier](UNIQUEID)
FK_ST_ID_2	Student_Questionnaire_2017	[dbEdu].[Identifier](UNIQUEID)
FK_CL_ID	Student_ClassRoster_2016	[dbEdu].[Identifier](UNIQUEID)
FK_CL_ID_2	Student_ClassRoster_2017	[dbEdu].[Identifier](UNIQUEID)
FK_HTP_ID	HeadTeacher_Questionnaire_2016	[dbEdu].[Identifier](UNIQUEID)
FK_HTP_ID_2	HeadTeacher_Questionnaire_2017	[dbEdu].[Identifier](UNIQUEID)
FK_SCH_ID	School_Questionnaire_2016	[dbEdu].[Identifier](UNIQUEID)
FK_TQ_ID	Teacher_Questionnaire_2017	[dbEdu].[Identifier](UNIQUEID)

### 2.5.3 Data Validation

- Each of the wave 1 and wave 2 tables return 8740 rows as there are 8740 uniqueids too.
- Some of the dates in the column STDDINT / STDINT from both wave 1 and wave 2 have erroneous data which should be rectified / replaced with a suitable correction. This has been done during the secondary tables creation.
- When unionized as well, we see only 8740 rows which means all the 4 columns – UNIQUEID, SCHOOLID, CLASSID, STUDENTID are the same for both the raw datasets.

```

/*Validations*/
SELECT    UNIQUEID
          ,SCHOOLID
          ,CLASSID
          ,STUDENTID
FROM dbo.vietnam_wave_1;
-- Returns 8740 rows

SELECT    UNIQUEID
          ,SCHOOLID
          ,CLASSID
          ,STUDENTID
FROM dbo.vietnam_wave_2;
-- Also returns 8740 rows
--> Now I want to check if both these 8740 rows are common between both tables
SELECT    UNIQUEID
          ,SCHOOLID
          ,CLASSID
          ,STUDENTID
FROM dbo.vietnam_wave_1
UNION
SELECT UNIQUEID
          ,SCHOOLID
          ,CLASSID
          ,STUDENTID
FROM dbo.vietnam_wave_2;
-- This also returns 8740 rows only because UNION returns non-duplicate values in the end.

```

### 2.5.4 Transaction and Concurrency Control (if any)

- N/A

### 2.5.5 Error Handling

- A Try and Catch exception block can be defined to handle the errors just like in Task 1.

## 2.5.6 Security

- ce\_user1 has been granted a SELECT command access to Teacher\_Questionnaire\_2017 table.
- Ce\_user1 has been denied SELECT access to the fields of dbEdu.Identifier – gender, ethnicity.

```
/*Grant access for Teacher_Questionnaire_2017 table*/
GRANT SELECT ON [dbEdu].[Teacher_Questionnaire_2017] TO ce_user1;
GO

/*Deny access for Identifier - Gender and Ethnicity fields*/
DENY SELECT ON dbEdu.Identifier (gender, ethnicity) TO ce_user1;
GO
```

## 2.5.7 Comments

- Comments have been everywhere to annotate what the script is going to be about.

## 2.6 T-SQL Statements

- Create database CP002.

```
USE master;
GO

CREATE DATABASE CP002;
```

### 2.6.1 Tables

- Create table – Identifier

```
USE CP002;
GO

CREATE TABLE Identifier
(
    UNIQUEID           VARCHAR(9) PRIMARY KEY,
    SCHOOLID          VARCHAR(4) NOT NULL,
    CLASSID           TINYINT      NOT NULL,
    STUDENTID         TINYINT      NOT NULL,
    YLCHILDID          VARCHAR(8),
    PROVINCE          VARCHAR(10) NOT NULL,
    DISTRICTCODE      VARCHAR(3) NOT NULL,
    LOCALITY          VARCHAR(5) NOT NULL,
    GENDER             VARCHAR(6),
    AGE                TINYINT,
    ETHNICITY          VARCHAR(10),
    ABSENT_DAYS       TINYINT,
    MOM_READ          VARCHAR(20),
    MOM_EDUC          VARCHAR(60),
    DAD_READ          VARCHAR(20),
    DAD_EDUC          VARCHAR(60),
    STDYLCHD          VARCHAR(3),
    STDCMPLT          VARCHAR(30)
)
```

- Insert rows into dbEdu.Identifier from dbo.vietnam\_wave\_1

```
INSERT INTO [CP002].[dbEdu].[Identifier] SELECT RTRIM(LTRIM([UNIQUEID])) AS UNIQUEID ,RTRIM(LTRIM([SCHOOLID])) AS SCHOOLID
, RTRIM(LTRIM([CLASSID])) AS CLASSID ,RTRIM(LTRIM([STUDENTID])) AS STUDENTID ,RTRIM(LTRIM([YLCHILDID])) AS YLCHILDID
,CASE [PROVINCE] WHEN '1' THEN 'Ben Tre' WHEN '2' THEN 'Da Nang' WHEN '3' THEN 'Hung Yen' WHEN '4' THEN 'Lao Cai' WHEN
'5' THEN 'Phu Yen' ELSE [PROVINCE] END AS [PROVINCE] ,CASE [DISTRICTCODE] WHEN '1' THEN 'PY1' WHEN '2' THEN 'PY2' WHEN '3'
THEN 'PY3' WHEN '4' THEN 'PY4' WHEN '5' THEN 'BT1' WHEN '6' THEN 'BT2' WHEN '7' THEN 'LC1' WHEN '8' THEN 'LC2' WHEN '9'
THEN 'LC3' WHEN '10' THEN 'HY1' WHEN '11' THEN 'HY2' WHEN '12' THEN 'DN1' WHEN '13' THEN 'DN2' WHEN '14' THEN 'DN3' ELSE
[DISTRICTCODE] END AS [DISTRICTCODE]
,CASE [LOCALITY] WHEN '1' THEN 'Rural' WHEN '2' THEN 'Urban' ELSE [LOCALITY] END AS [LOCALITY] ,CASE [GENDER] WHEN '1'
THEN 'Male' WHEN '2' THEN 'Female' ELSE [GENDER] END AS [GENDER]
,[AGE]
,CASE [ETHNICITY] WHEN '1' THEN 'Kinh' WHEN '2' THEN 'H'Mong' WHEN '3' THEN 'Cham-HRoi' WHEN '4' THEN 'Ede' WHEN '5'
THEN 'Ba Na' WHEN '6' THEN 'Nung' WHEN '7' THEN 'Tay' WHEN '8' THEN 'Dao' WHEN '9' THEN 'Giay' WHEN '10' THEN 'Other' WHEN
'99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [ETHNICITY] END AS [ETHNICITY]
,ABSENT_DAYS
,CASE [MOM_READ] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '2' THEN 'Do not know' ELSE [MOM_READ] END AS [MOM_READ]
,CASE [MOM_EDUC] WHEN '0' THEN 'Never been to school' WHEN '1' THEN 'Primary school (Greades 1-5)' WHEN '2' THEN 'Lower
secondary school (Grades 6-9)' WHEN '3' THEN 'Intermediate vocational training' WHEN '4' THEN 'Upper secondary school
(Grades 10-12)' WHEN '5' THEN 'Higher education (e.g. university/college or higher)' WHEN '6' THEN 'Don't know' WHEN '99'
THEN 'Missing' WHEN '88' THEN 'NA' ELSE [MOM_EDUC] END AS [MOM_EDUC]
,CASE [DAD_READ] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '2' THEN 'Do not know' ELSE [DAD_READ] END AS [DAD_READ]
,CASE [DAD_EDUC] WHEN '0' THEN 'Never been to school' WHEN '1' THEN 'Primary school (Greades 1-5)' WHEN '2' THEN 'Lower
secondary school (Grades 6-9)' WHEN '3' THEN 'Intermediate vocational training' WHEN '4' THEN 'Upper secondary school
(Grades 10-12)' WHEN '5' THEN 'Higher education (e.g. university/college or higher)' WHEN '6' THEN 'Don't know' WHEN '99'
THEN 'Missing' WHEN '88' THEN 'NA' ELSE [DAD_EDUC] END AS [DAD_EDUC]
,CASE [STDYLCHD] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' ELSE [STDYLCHD] END AS [STDYLCHD]
,CASE [STDCMPLT] WHEN '88' THEN 'NA' WHEN '1' THEN 'Absent' WHEN '2' THEN 'Refused to participate' WHEN '99' THEN
'Missing' ELSE [STDCMPLT] END AS [STDCMPLT]
FROM CP002.dbo.vietnam_wave_1;
```

- Create table – Student\_CognitiveTests\_2016

```
USE CP002; GO

CREATE TABLE Student_CognitiveTests_2016 ( STC_ID INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[STDINT] VARCHAR(20)

,[ENG_TEST] VARCHAR(3) ,[ENG_ITEM1] BIT ,[ENG_ITEM2] BIT ,[ENG_ITEM3] BIT ,[ENG_ITEM4] BIT ,[ENG_ITEM5] BIT ,[ENG_ITEM6] BIT
,[ENG_ITEM7] BIT ,[ENG_ITEM8] BIT ,[ENG_ITEM9] BIT ,[ENG_ITEM10] BIT ,[ENG_ITEM11] BIT ,[ENG_ITEM12] BIT ,[ENG_ITEM13] BIT
,[ENG_ITEM14] BIT ,[ENG_ITEM15] BIT ,[ENG_ITEM16] BIT ,[ENG_ITEM17] BIT ,[ENG_ITEM18] BIT ,[ENG_ITEM19] BIT ,[ENG_ITEM20] BIT
,[ENG_ITEM21] BIT ,[ENG_ITEM22] BIT ,[ENG_ITEM23] BIT ,[ENG_ITEM24] BIT ,[ENG_ITEM25] BIT ,[ENG_ITEM26] BIT ,[ENG_ITEM27] BIT
,[ENG_ITEM28] BIT ,[ENG_ITEM29] BIT ,[ENG_ITEM30] BIT ,[ENG_ITEM31] BIT ,[ENG_ITEM32] BIT ,[ENG_ITEM33] BIT ,[ENG_ITEM34] BIT
,[ENG_ITEM35] BIT ,[ENG_ITEM36] BIT ,[ENG_ITEM37] BIT ,[ENG_ITEM38] BIT ,[ENG_ITEM39] BIT ,[ENG_ITEM40] BIT ,[ENG_RAWSCORE]
TINYINT

,[MATH_TEST] VARCHAR(3) ,[MATH_ITEM1] BIT ,[MATH_ITEM2] BIT ,[MATH_ITEM3] BIT ,[MATH_ITEM4] BIT ,[MATH_ITEM5] BIT
,[MATH_ITEM6] BIT ,[MATH_ITEM7] BIT ,[MATH_ITEM8] BIT ,[MATH_ITEM9] BIT ,[MATH_ITEM10] BIT ,[MATH_ITEM11] BIT ,[MATH_ITEM12]
BIT ,[MATH_ITEM13] BIT ,[MATH_ITEM14] BIT ,[MATH_ITEM15] BIT ,[MATH_ITEM16] BIT ,[MATH_ITEM17] BIT ,[MATH_ITEM18] BIT
,[MATH_ITEM19] BIT ,[MATH_ITEM20] BIT ,[MATH_ITEM21] BIT ,[MATH_ITEM22] BIT ,[MATH_ITEM23] BIT ,[MATH_ITEM24] BIT
,[MATH_ITEM25] BIT ,[MATH_ITEM26] BIT ,[MATH_ITEM27] BIT ,[MATH_ITEM28] BIT ,[MATH_ITEM29] BIT ,[MATH_ITEM30] BIT
,[MATH_ITEM31] BIT ,[MATH_ITEM32] BIT ,[MATH_ITEM33] BIT ,[MATH_ITEM34] BIT ,[MATH_ITEM35] BIT ,[MATH_ITEM36] BIT
,[MATH_ITEM37] BIT ,[MATH_ITEM38] BIT ,[MATH_ITEM39] BIT ,[MATH_ITEM40] BIT ,[MATH_RAWSCORE] TINYINT

,CONSTRAINT FK_STC_ID FOREIGN KEY (UNIQUEID) REFERENCES [dbEdu].[Identifier](UNIQUEID) )
```

- Insert into table Student\_CognitiveTests\_2016 from dbo.vietnam\_wave\_1

```
INSERT INTO [CP002].[dbEdu].[Student_CognitiveTests_2016]
SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID ,REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
REPLACE( REPLACE( REPLACE( STDINT ,';' ,'/') ,'/','/') ,'+','') ,'' ,'' ) , '2016/' , '2016' ) , '20163' , '2016'
,'/010/' , '10' ) , '/016' , '/2016' ) , '20016' , '2016' ) , '096' , '09' ) , '069' , '06' ) , '179' , '179' ) , '110' , '10' ) as STDINT
,[ENG_TEST] ,[ENG_ITEM1] ,[ENG_ITEM2] ,[ENG_ITEM3] ,[ENG_ITEM4] ,[ENG_ITEM5] ,[ENG_ITEM6] ,[ENG_ITEM7] ,[ENG_ITEM8]
,[ENG_ITEM9] ,[ENG_ITEM10] ,[ENG_ITEM11] ,[ENG_ITEM12] ,[ENG_ITEM13] ,[ENG_ITEM14] ,[ENG_ITEM15] ,[ENG_ITEM16] ,[ENG_ITEM17]
,[ENG_ITEM18] ,[ENG_ITEM19] ,[ENG_ITEM20] ,[ENG_ITEM21] ,[ENG_ITEM22] ,[ENG_ITEM23] ,[ENG_ITEM24] ,[ENG_ITEM25] ,[ENG_ITEM26]
,[ENG_ITEM27] ,[ENG_ITEM28] ,[ENG_ITEM29] ,[ENG_ITEM30] ,[ENG_ITEM31] ,[ENG_ITEM32] ,[ENG_ITEM33] ,[ENG_ITEM34] ,[ENG_ITEM35]
,[ENG_ITEM36] ,[ENG_ITEM37] ,[ENG_ITEM38] ,[ENG_ITEM39] ,[ENG_ITEM40] ,[ENG_RAWSCORE] ,[MATH_TEST] ,[MATH_ITEM1] ,[MATH_ITEM2]
,[MATH_ITEM3] ,[MATH_ITEM4] ,[MATH_ITEM5] ,[MATH_ITEM6] ,[MATH_ITEM7] ,[MATH_ITEM8] ,[MATH_ITEM9] ,[MATH_ITEM10] ,[MATH_ITEM11]
,[MATH_ITEM12] ,[MATH_ITEM13] ,[MATH_ITEM14] ,[MATH_ITEM15] ,[MATH_ITEM16] ,[MATH_ITEM17] ,[MATH_ITEM18] ,[MATH_ITEM19]
,[MATH_ITEM20] ,[MATH_ITEM21] ,[MATH_ITEM22] ,[MATH_ITEM23] ,[MATH_ITEM24] ,[MATH_ITEM25] ,[MATH_ITEM26] ,[MATH_ITEM27]
,[MATH_ITEM28] ,[MATH_ITEM29] ,[MATH_ITEM30] ,[MATH_ITEM31] ,[MATH_ITEM32] ,[MATH_ITEM33] ,[MATH_ITEM34] ,[MATH_ITEM35]
,[MATH_ITEM36] ,[MATH_ITEM37] ,[MATH_ITEM38] ,[MATH_ITEM39] ,[MATH_ITEM40] ,[MATH_RAWSCORE]

FROM CP002.dbo.vietnam_wave_1 --;
```

- Create table – Student\_CognitiveTests\_2017

```
USE CP002; GO

CREATE TABLE Student_CognitiveTests_2017 ( STC_ID_2 INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[STDINT] VARCHAR(20)

,[ENG_TEST] VARCHAR(3) ,[ENG_ITEM1] BIT ,[ENG_ITEM2] BIT ,[ENG_ITEM3] BIT ,[ENG_ITEM4] BIT ,[ENG_ITEM5] BIT ,[ENG_ITEM6]
BIT ,[ENG_ITEM7] BIT ,[ENG_ITEM8] BIT ,[ENG_ITEM9] BIT ,[ENG_ITEM10] BIT ,[ENG_ITEM11] BIT ,[ENG_ITEM12] BIT ,[ENG_ITEM13]
BIT ,[ENG_ITEM14] BIT ,[ENG_ITEM15] BIT ,[ENG_ITEM16] BIT ,[ENG_ITEM17] BIT ,[ENG_ITEM18] BIT ,[ENG_ITEM19] BIT
,[ENG_ITEM20] BIT ,[ENG_ITEM21] BIT ,[ENG_ITEM22] BIT ,[ENG_ITEM23] BIT ,[ENG_ITEM24] BIT ,[ENG_ITEM25] BIT ,[ENG_ITEM26]
BIT ,[ENG_ITEM27] BIT ,[ENG_ITEM28] BIT ,[ENG_ITEM29] BIT ,[ENG_ITEM30] BIT ,[ENG_ITEM31] BIT ,[ENG_ITEM32] BIT
,[ENG_ITEM33] BIT ,[ENG_ITEM34] BIT ,[ENG_ITEM35] BIT ,[ENG_ITEM36] BIT ,[ENG_ITEM37] BIT ,[ENG_ITEM38] BIT ,[ENG_ITEM39]
BIT ,[ENG_ITEM40] BIT ,[ENG_RAWSCORE] TINYINT

,[MATH_TEST] VARCHAR(3) ,[MATH_ITEM1] BIT ,[MATH_ITEM2] BIT ,[MATH_ITEM3] BIT ,[MATH_ITEM4] BIT ,[MATH_ITEM5] BIT
,[MATH_ITEM6] BIT ,[MATH_ITEM7] BIT ,[MATH_ITEM8] BIT ,[MATH_ITEM9] BIT ,[MATH_ITEM10] BIT ,[MATH_ITEM11] BIT ,[MATH_ITEM12]
BIT ,[MATH_ITEM13] BIT ,[MATH_ITEM14] BIT ,[MATH_ITEM15] BIT ,[MATH_ITEM16] BIT ,[MATH_ITEM17] BIT ,[MATH_ITEM18] BIT
,[MATH_ITEM19] BIT ,[MATH_ITEM20] BIT ,[MATH_ITEM21] BIT ,[MATH_ITEM22] BIT ,[MATH_ITEM23] BIT ,[MATH_ITEM24] BIT
,[MATH_ITEM25] BIT ,[MATH_ITEM26] BIT ,[MATH_ITEM27] BIT ,[MATH_ITEM28] BIT ,[MATH_ITEM29] BIT ,[MATH_ITEM30] BIT
,[MATH_ITEM31] BIT ,[MATH_ITEM32] BIT ,[MATH_ITEM33] BIT ,[MATH_ITEM34] BIT ,[MATH_ITEM35] BIT ,[MATH_ITEM36] BIT
,[MATH_ITEM37] BIT ,[MATH_ITEM38] BIT ,[MATH_ITEM39] BIT ,[MATH_ITEM40] BIT ,[MATH_RAWSCORE] TINYINT

,[TS_TEST] VARCHAR(3) ,[TS_ITEM1] BIT ,[TS_ITEM2] BIT ,[TS_ITEM3A] BIT ,[TS_ITEM3B] BIT ,[TS_ITEM3C] BIT ,[TS_ITEM3D] BIT
,[TS_ITEM3E] BIT ,[TS_ITEM3F] BIT ,[TS_ITEM3G] BIT ,[TS_ITEM4] BIT ,[TS_ITEM5] BIT ,[TS_ITEM6] BIT ,[TS_ITEM7] BIT
,[TS_ITEM8A] BIT ,[TS_ITEM8B] BIT ,[TS_ITEM8C] BIT ,[TS_ITEM8] BIT ,[TS_ITEM9] BIT ,[TS_ITEM10] BIT ,[TS_ITEM11] BIT
,[TS_ITEM12] BIT ,[TS_ITEM11_12] BIT ,[TS_ITEM13] BIT ,[TS_ITEM14] BIT ,[TS_ITEM15] BIT ,[TS_ITEM16] BIT ,[TS_ITEM17] BIT
,[TS_ITEM18] BIT ,[TS_ITEM19] BIT ,[TS_ITEM20] BIT ,[TS_ITEM21] BIT ,[TS_ITEM22] BIT ,[TS_ITEM23] BIT

,CONSTRAINT FK_STC_ID_2 FOREIGN KEY (UNIQUEID) REFERENCES [dbEdu].[Identifier](UNIQUEID) )
```

- Insert into table Student\_CognitiveTests\_2017 from dbo.vietnam\_wave\_2

```

INSERT INTO [CP002].[edbU].[Student_CognitiveTests_2017] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID
,REPLACE(LTRIM(RTRIM(STDINT)), '217', '2017') as STDINT

,[ENG_TEST],[ENG_ITEM1],[ENG_ITEM2],[ENG_ITEM3],[ENG_ITEM4],[ENG_ITEM5],[ENG_ITEM6],[ENG_ITEM7],[ENG_ITEM8]
,[ENG_ITEM9],[ENG_ITEM11],[ENG_ITEM12],[ENG_ITEM13],[ENG_ITEM14],[ENG_ITEM15],[ENG_ITEM16],[ENG_ITEM17]
,[ENG_ITEM18],[ENG_ITEM19],[ENG_ITEM20],[ENG_ITEM21],[ENG_ITEM22],[ENG_ITEM23],[ENG_ITEM24],[ENG_ITEM25],[ENG_ITEM26]
,[ENG_ITEM27],[ENG_ITEM28],[ENG_ITEM29],[ENG_ITEM30],[ENG_ITEM31],[ENG_ITEM32],[ENG_ITEM33],[ENG_ITEM34],[ENG_ITEM35]
,[ENG_ITEM36],[ENG_ITEM37],[ENG_ITEM38],[ENG_ITEM39],[ENG_ITEM40],[ENG_RAWSCORE],[MATH_TEST],[MATH_ITEM1],[MATH_ITEM2]
,[MATH_ITEM3],[MATH_ITEM4],[MATH_ITEM5],[MATH_ITEM6],[MATH_ITEM7],[MATH_ITEM8],[MATH_ITEM9],[MATH_ITEM10]
,[MATH_ITEM11],[MATH_ITEM12],[MATH_ITEM13],[MATH_ITEM14],[MATH_ITEM15],[MATH_ITEM16],[MATH_ITEM17],[MATH_ITEM18]
,[MATH_ITEM19],[MATH_ITEM20],[MATH_ITEM21],[MATH_ITEM22],[MATH_ITEM23],[MATH_ITEM24],[MATH_ITEM25],[MATH_ITEM26]
,[MATH_ITEM27],[MATH_ITEM28],[MATH_ITEM29],[MATH_ITEM30],[MATH_ITEM31],[MATH_ITEM32],[MATH_ITEM33],[MATH_ITEM34]
,[MATH_ITEM35],[MATH_ITEM36],[MATH_ITEM37],[MATH_ITEM38],[MATH_ITEM39],[MATH_ITEM40],[MATH_RAWSCORE],[TS_TEST]
,[TS_ITEM1],[TS_ITEM2],[TS_ITEM3A],[TS_ITEM3B],[TS_ITEM3D],[TS_ITEM3E],[TS_ITEM3F],[TS_ITEM3G],[TS_ITEM4]
,[TS_ITEM5],[TS_ITEM6],[TS_ITEM7],[TS_ITEM8A],[TS_ITEM8B],[TS_ITEM8C],[TS_ITEM8B],[TS_ITEM9],[TS_ITEM10],[TS_ITEM11]
,[TS_ITEM12],[TS_ITEM11_12],[TS_ITEM13],[TS_ITEM14],[TS_ITEM15],[TS_ITEM16],[TS_ITEM17],[TS_ITEM18],[TS_ITEM19]
,[TS_ITEM20],[TS_ITEM21],[TS_ITEM22],[TS_ITEM23]

FROM CP002.dbo.vietnam_wave_2 --;

```

- Create table – Student\_Questionnaire\_2016

```

USE CP002; GO

CREATE TABLE Student_Questionnaire_2016 ( ST_ID INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)

,[STDINT] VARCHAR(20)

,[STDLIV] VARCHAR(50) ,[STDLNGHM] VARCHAR(50) ,[STDEAL] VARCHAR(50) ,[STDHLT1] BIT ,[STDHLT2] BIT ,[STDHLT3] BIT
,[STDHLT4] BIT ,[STDHLT5] BIT ,[STDHLT6] BIT ,[STDHLT0] BIT ,[STPPLHD] TINYINT ,[STSIBOLD] TINYINT ,[STSIBYNG] TINYINT
,[STNMBOOK] NVARCHAR(60) ,[STPLSTDY] BIT ,[STHVMTEL] BIT ,[STHVRADO] BIT ,[STHVTLE] BIT ,[STHVBKE] BIT ,[STHVMTRK] BIT
,[STHVDESK] BIT ,[STHVCHR] BIT ,[STHVLMAMP] BIT ,[STHVEFAN] BIT ,[STHVAIRC] BIT ,[STHVCAR] BIT ,[STHVCOMP] BIT ,[STHVMR]
,[STHVFRDG] BIT ,[STHVMCRD] BIT ,[STHVDV] BIT ,[STHVCBLE] BIT ,[STTMSCH] SMALLINT ,[STEATLNC] BIT ,[STPAYLNC] VARCHAR(50)
,[STRPTCL1] VARCHAR(50) ,[STRPTCL6] VARCHAR(50) ,[STRPTCL0] VARCHAR(50) ,[STTLTSCH] VARCHAR(50) ,[STAGEENG] VARCHAR(50)
,[STITMOW1] BIT ,[STITMOW2] BIT ,[STITMOW3] BIT ,[STITMOW4] BIT ,[STITMOW5] BIT ,[STITMOW6] BIT ,[STITMOW7] BIT ,[STITMOW8] BIT
,[STBRWBK] VARCHAR(100) ,[STREADFN] VARCHAR(50) ,[STREADCH] VARCHAR(50) ,[STREADLR] VARCHAR(50) ,[STPLHLRD] VARCHAR(50)
,[STPLHL01] VARCHAR(50) ,[STPLHL02] VARCHAR(50) ,[STPLHL03] VARCHAR(50) ,[STPLHL04] VARCHAR(50) ,[STPLHL05] VARCHAR(50)
,[STPLHL06] VARCHAR(50) ,[STPLHL07] VARCHAR(50) ,[STSPEN01] VARCHAR(50) ,[STSPEN02] VARCHAR(50) ,[STSPEN03] VARCHAR(50)
,[STSPEN04] VARCHAR(50) ,[STSPEN05] VARCHAR(50) ,[STATEN01] VARCHAR(50) ,[STATEN02] VARCHAR(50) ,[STATEN03] VARCHAR(50)
,[STNATEN04] VARCHAR(50) ,[STNONSCL] VARCHAR(50) ,[STGR1001] VARCHAR(50) ,[STGR1002] VARCHAR(50) ,[STGR1003] VARCHAR(50)
,[STGR1004] VARCHAR(50) ,[STGR1005] VARCHAR(50) ,[STGR1006] VARCHAR(50) ,[STGR1007] VARCHAR(50) ,[STGR1008] VARCHAR(50)
,[STGR1009] VARCHAR(50) ,[STGR1011] VARCHAR(50) ,[STGHGHRD] VARCHAR(50)

,CONSTRAINT FK_ST_ID FOREIGN KEY (UNIQUEID) REFERENCES [dbEdu].[Identifier](UNIQUEID) )

```

- Insert into table Student\_Questionnaire\_2016 from dbo.vietnam\_wave\_1

- Create table – Student Questionnaire 2017

- Insert into table Student\_Questionnaire\_2017 from dbo.vietnam\_wave\_2

- Create table – Student\_ClassRoster\_2016
- Insert into table - Student\_ClassRoster\_2016 from dbo.vietnam\_wave\_1

```

USE CP002; GO

CREATE TABLE Student_ClassRoster_2016 ( CL_ID INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9) ,[GRLENRL] TINYINT
,[BOYENRL] TINYINT ,[TTLLENRL] TINYINT ,[TGRLENRL] TINYINT ,[TBOYENRL] TINYINT ,[TTTLENRL] TINYINT ,[ATDTMSY] VARCHAR(30)
,[ATDDFSY] VARCHAR(50) ,[TGTMTHENG] BIT ,[SCALLCT] VARCHAR(50) ,[GRPBALTY] VARCHAR(30) ,[CLSORD1] TINYINT ,[CLSORD2]
TINYINT ,[SCAVLB1] BIT ,[SCAVLB2] BIT ,[SCAVLB3] BIT ,[SCAVLB4] BIT ,[SCAVLB5] BIT ,[SCAVLB6] BIT ,[SCAVLB7] BIT
,[SCAVLB8] BIT ,[SCAVLB9] BIT ,[SCAVLB10] BIT ,[SCPRDDAY] TINYINT ,[SCLNONPR] TINYINT ,[SCMMNTIN] TINYINT ,[SCMNENIN]
TINYINT ,[SCTXTMTH] VARCHAR(30) ,[SCTXTEG] VARCHAR(30)

,CONSTRAINT FK_CL_ID FOREIGN KEY (UNIQUEID) REFERENCES [dbEdu].[Identifier](UNIQUEID) )

--TRUNCATE TABLE [CP002].[dbo].[Student_ClassRoster_2016];

INSERT INTO [CP002].[dbo].[Student_ClassRoster_2016] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID ,[GRLENRL] ,[BOYENRL]
,[TTLLENRL] ,[TGRLENRL] ,[TBOYENRL] ,[TTTLENRL]

,CASE [ATDTMSY] WHEN '1' THEN 'Normal' WHEN '2' THEN 'Higher than normal' WHEN '3' THEN 'Lower than normal' WHEN '99'
THEN 'Missing' WHEN '88' THEN 'NA' ELSE [ATDTMSY] END as [ATDTMSY] ,CASE [ATDDFSY] WHEN '1' THEN 'Patterns of farm work'
WHEN '2' THEN 'Seasonal migration' WHEN '3' THEN 'Recent or forthcoming holiday or celebration' WHEN '4' THEN 'Other' WHEN
'99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [ATDDFSY] END as [ATDDFSY]

,[TGTMTHENG]

,CASE [SCALLCT] WHEN '0' THEN 'There is only one class in Grade 10' WHEN '1' THEN 'Randomly' WHEN '2' THEN
'Alphabetically' WHEN '3' THEN 'By ability (e.g high / middle / low)' WHEN '4' THEN 'According to the level of fees they
pay' WHEN '5' THEN 'By choice of subject' WHEN '6' THEN 'Other method' WHEN '7' THEN 'A combination of two or more of the
above methods' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [SCALLCT] ,CASE [GRPBALTY] WHEN '1' THEN
'Low ability' WHEN '2' THEN 'Medium ability' WHEN '3' THEN 'High ability' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA'
ELSE [GRPBALTY] END as [GRPBALTY]

,[CLSORD1] ,[CLSORD2] ,[SCAVLB1] ,[SCAVLB2] ,[SCAVLB3] ,[SCAVLB4] ,[SCAVLB5] ,[SCAVLB6] ,[SCAVLB7] ,[SCAVLB8]
,[SCAVLB9] ,[SCAVLB10] ,[SCPRDDAY] ,[SCLNONPR] ,[SCMMNTIN] ,[SCMNENIN]

,CASE [SCTXTMTH] WHEN '1' THEN 'Compulsory textbooks' WHEN '2' THEN 'Non-compulsory textbooks' WHEN '99' THEN 'Missing'
WHEN '88' THEN 'NA' ELSE [SCTXTMTH] ,CASE [SCTXTEG] WHEN '1' THEN 'Compulsory textbooks' WHEN '2' THEN
'Non-compulsory textbooks' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [SCTXTEG] END as [SCTXTEG]

FROM CP002.dbo.vietnam_wave_1;

```

- Create table – Student\_ClassRoster\_2017
- Insert into table - Student\_ClassRoster\_2017 from dbo.vietnam\_wave\_2

```

USE CP002; GO

CREATE TABLE Student_ClassRoster_2017 ( CL_ID_2 INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[SCTMTID1] TINYINT ,[W1MSMTC1] VARCHAR(10) ,[W1MNWTC1] TINYINT ,[SCTENID1] TINYINT ,[W1ESMTC1] VARCHAR(10) ,[W1ENWTC1]
TINYINT ,[REGTCH] VARCHAR(30) ,[ATDGIRL] TINYINT ,[ATDBOY] TINYINT ,[ATDTTL] TINYINT ,[SECSIZE] TINYINT ,[NEWCHLD] TINYINT
,[STDSTLENR] VARCHAR(10) ,[STDNOENR] VARCHAR(90) ,[STDNMABS] TINYINT ,[B1LVLSUPP] TINYINT ,[B1MTVSUC] TINYINT ,[B2ACABMT]
TINYINT ,[B2PRCLMT] TINYINT ,[B3ACABEN] TINYINT ,[B3PRCLEN] TINYINT

,CONSTRAINT FK_CL_ID_2 FOREIGN KEY (UNIQUEID) REFERENCES [dbEdu].[Identifier](UNIQUEID) ) --TRUNCATE TABLE
[CP002].[dbo].[Student_ClassRoster_2017];

INSERT INTO [CP002].[dbo].[Student_ClassRoster_2017] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID ,[SCTMTID1] ,CASE
[W1MSMTC1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [W1MSMTC1] END as
[W1MSMTC1] ,[W1MNWTC1] ,[SCTENID1] ,CASE [W1ESMTC1] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '79' THEN 'Missing' WHEN
'88' THEN 'NA' ELSE [W1ESMTC1] END as [W1ESMTC1] ,[W1ENWTC1] ,CASE [REGTCH] WHEN '1' THEN 'Morning' WHEN '2' THEN
'Afternoon' WHEN '3' THEN 'Morning & Afternoon' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [REGTCH] END as [REGTCH]
,[ATDGIRL] ,[ATDBOY] ,[ATDTTL] ,[SECSIZE] ,[NEWCHLD]

,CASE [STDSTLENR] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [STDSTLENR] END
as [STDSTLENR]

,CASE [STDNOENR] WHEN '1' THEN 'Dropped out of school' WHEN '2' THEN 'Moved to a new school' WHEN '3' THEN 'Student is
still enrolled in this school, but in another class' WHEN '4' THEN 'Unknown' WHEN '5' THEN 'Other' WHEN '79' THEN 'Missing'
WHEN '88' THEN 'NA' ELSE [STDNOENR] END as [STDNOENR] ,[STDNMABS] ,[B1LVLSUPP] ,[B1MTVSUC] ,[B2ACABMT] ,[B2PRCLMT]
,[B3ACABEN] ,[B3PRCLEN]

FROM CP002.dbo.vietnam_wave_2 ;

```

- Create table – HeadTeacher\_Questionnaire\_2016

```

CREATE TABLE HeadTeacher_Questionnaire_2016 (HTP_ID INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[HTNTCMP] VARCHAR(30) ,[HTDINT] VARCHAR(20)
,[NUMG10CLS] TINYINT ,[HTAGE] TINYINT ,[HTSEX] VARCHAR(6) ,[HTETHGRP] VARCHAR(10) ,[HTRELGN] VARCHAR(30) ,[HTMTHNG] VARCHAR(20) ,[HTFRMPRV] BIT
,[HTCURREL] VARCHAR(20) ,[HTYRSHT] TINYINT ,[HTLVLED] VARCHAR(60) ,[HTVLVLTCH] VARCHAR(100) ,[HTEXCTCH] VARCHAR(60) ,[HTTYPSC] VARCHAR(20)
,[HTLWSGRD] VARCHAR(10) ,[HTHGHRD] VARCHAR(10) ,[HTMSTMEN] SMALLINT ,[HTNMETST] SMALLINT ,[HTYREST] SMALLINT ,[HTP135CM] VARCHAR(20) ,[HTPRDIST]
VARCHAR(20) ,[HTBOARD] NVARCHAR(200) ,[HTSCHAHL] VARCHAR(150) ,[HTMRNSTR] VARCHAR(30) ,[HTAFTRSTR] VARCHAR(30) ,[HTAFTEND]
VARCHAR(30) ,[HTREGTC] BIT ,[HTNONCMB] BIT ,[HTEXCTR] BIT ,[HTNOACT] BIT ,[HTNCMCH] BIT ,[HTNCMCL] VARCHAR(200) ,[HTRNQMS] SMALLINT ,[HTCENMST]
SMALLINT ,[HTPTNMST] SMALLINT ,[HTROMATH] SMALLINT ,[HTREOENG] SMALLINT ,[HTCHAPP] BIT ,[HTREWTC1] BIT ,[HTREWTC2] BIT ,[HTREWTC4] BIT ,[HTREWTC5] BIT
,[HTREWTC6] BIT ,[HTREWTC7] BIT ,[HTREWTC8] BIT ,[HTDISTC1] BIT ,[HTDISTC2] BIT ,[HTDISTC3] BIT ,[HTDISTC4] BIT ,[HTDISTC6] BIT ,[HTDISTC7] BIT
,[HTDISTC8] BIT ,[HTDISTC8] BIT ,[HTGNADMN] BIT ,[HTACRARE] BIT ,[HTACRAETH] BIT ,[HTACRAGND] BIT ,[HTACRAPER] BIT ,[HTACRAEXM] BIT ,[HTACRAOTH] BIT
,[HTACRAALL] BIT ,[HTAPAIID1] BIGINT ,[HTEXM011] BIT ,[HTEXM021] BIT ,[HTEXM031] BIT ,[HTEXM041] BIT ,[HTEXM051] BIT ,[HTEXM061] BIT ,[HTEXM001] BIT
,[HTAPAIID2] BIGINT ,[HTEXM012] BIT ,[HTEXM022] BIT ,[HTEXM032] BIT ,[HTEXM042] BIT ,[HTEXM052] BIT ,[HTEXM062] BIT ,[HTEXM002] BIT ,[HTAPAIID3]
BIGINT ,[HTEXM013] BIT ,[HTEXM023] BIT ,[HTEXM033] BIT ,[HTEXM043] BIT ,[HTEXM053] BIT ,[HTEXM063] BIT ,[HTAPAIID4] BIGINT ,[HTEXM014]
BIT ,[HTEXM024] BIT ,[HTEXM034] BIT ,[HTEXM044] BIT ,[HTEXM054] BIT ,[HTEXM064] BIT ,[HTEXM004] BIT ,[HTAPAIID5] BIGINT ,[HTEXM015] BIT ,[HTEXM025]
BIT ,[HTEXM035] BIT ,[HTEXM045] BIT ,[HTEXM055] BIT ,[HTEXM065] BIT ,[HTEXM005] BIT ,[HTAPAIID6] BIGINT ,[HTEXM016] BIT ,[HTEXM026] BIT ,[HTEXM036]
BIT ,[HTEXM046] BIT ,[HTEXM056] BIT ,[HTEXM066] BIT ,[HTEXM006] BIT ,[HTAPAIID7] BIGINT ,[HTEXM017] BIT ,[HTEXM027] BIT ,[HTEXM037] BIT ,[HTEXM047]
BIT ,[HTEXM057] BIT ,[HTEXM067] BIT ,[HTEXM007] BIT ,[HTAPAIID8] BIGINT ,[HTEXM018] BIT ,[HTEXM028] BIT ,[HTEXM038] BIT ,[HTEXM048] BIT ,[HTEXM058]
BIT ,[HTEXM068] BIT ,[HTEXM008] BIT ,[HTAPAIID9] BIGINT ,[HTEXM019] BIT ,[HTEXM029] BIT ,[HTEXM039] BIT ,[HTEXM049] BIT ,[HTEXM059] BIT ,[HTEXM069]
BIT ,[HTEXM009] BIT ,[HTAPAIID10] BIGINT ,[HTEXM010] BIT ,[HTEXM020] BIT ,[HTEXM030] BIT ,[HTEXM040] BIT ,[HTEXM050] BIT ,[HTEXM060] BIT
,[HTEXM0010] BIT ,[HTAPAIID11] BIGINT ,[HTEXM011] BIT ,[HTEXM021] BIT ,[HTEXM031] BIT ,[HTEXM041] BIT ,[HTEXM051] BIT ,[HTEXM061] BIT
,[HTEXM0011] BIT ,[HTAPAIID12] BIGINT ,[HTEXM012] BIT ,[HTEXM022] BIT ,[HTEXM032] BIT ,[HTEXM042] BIT ,[HTEXM052] BIT ,[HTEXM062] BIT
,[HTEXM0012] BIT ,[HTENGR10] SMALLINT ,[HTENBY10] SMALLINT ,[HTTLGR10] SMALLINT ,[HTENGR11] SMALLINT ,[HTENBY11] SMALLINT
,[HTTLGR11] SMALLINT ,[HTLBV11] SMALLINT ,[HTENGR12] SMALLINT ,[HTENBY12] SMALLINT ,[HTTLGR12] SMALLINT ,[HTNMC110] SMALLINT
,[HTNMC111] SMALLINT ,[HTNMC112] SMALLINT ,[HTALLC10] VARCHAR(150) ,[HTPRTEXP] VARCHAR(200) ,[HTENTOT10] SMALLINT ,CONSTRAINT FK_HTP_ID FOREIGN KEY
(UNIQUEID) REFERENCES [dbo].[Identifier](UNIQUEID) )

```

- Insert into table – HeadTeacher\_Questionnaire\_2016 from dbo.vietnam\_wave\_1

```

INSERT INTO [CP002].[dbo].[HeadTeacher_Questionnaire_2016] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID
,CASE [HTNTCMP] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTNTCMP] END as [HTNTCMP]
,[HTDINT]
,[NUMG10CLS] ,[HTAGE] ,CASE [HTSEX] WHEN '1' THEN 'Male' WHEN '2' THEN 'Female' ELSE [HTSEX] END as [HTSEX] ,CASE [HTETHGRP] WHEN '1' THEN
'Kinh' WHEN '2' THEN 'H'Mong' WHEN '3' THEN 'Cham-Hroi' WHEN '4' THEN 'Ede' WHEN '5' THEN 'Nung' WHEN '6' THEN 'Tay' WHEN '7' THEN 'Dao'
WHEN '8' THEN 'Giay' WHEN '9' THEN 'Ho'a' WHEN '10' THEN 'Ba Na' WHEN '11' THEN 'Other' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE
[HTETHGRP] END as [HTETHGRP] ,CASE [HTRELGN] WHEN '0' THEN 'religion' WHEN '1' THEN 'Buddhist' WHEN '2' THEN 'Christian (Catholic)' WHEN
'3' THEN 'Christian (Protestant)' WHEN '4' THEN 'Cao Dai' WHEN '5' THEN 'Other' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTRELGN]
END as [HTRELGN] ,CASE [HTMTHNG] WHEN '1' THEN 'Vietnamese' WHEN '2' THEN 'H'Mong' WHEN '3' THEN 'Cham-Hroi' WHEN '4' THEN 'Ede' WHEN '5'
THEN 'Nung' WHEN '6' THEN 'Tay' WHEN '7' THEN 'Dao' WHEN '8' THEN 'Giay' WHEN '9' THEN 'Chinese' WHEN '10' THEN 'Other' WHEN '99' THEN
'Missing' WHEN '88' THEN 'NA' ELSE [HTMTHNG] END as [HTMTHNG] ,[HTFRMPRV]
,CASE [HTCURREL] WHEN '1' THEN 'Principal / Director' WHEN '2' THEN 'Vice-Principal' WHEN '3' THEN 'Another role' WHEN '99' THEN 'Missing'
WHEN '88' THEN 'NA' ELSE [HTCURREL] END as [HTCURREL]
,[HTYRSHT]
,CASE [HTLVLED] WHEN '1' THEN 'Upper secondary or equivalent' WHEN '2' THEN 'Vocational training school' WHEN '3' THEN 'College education'
WHEN '4' THEN 'University education (undergraduate)' WHEN '5' THEN 'University education (postgraduate)' WHEN '99' THEN 'Missing' WHEN '88'
THEN 'NA' ELSE [HTLVLED] END as [HTLVLED]
,CASE [HTLVLVLTCH] WHEN '0' THEN 'I am not trained' WHEN '1' THEN 'Short course or crash course in teaching profession' WHEN '2' THEN 'Teacher
training level (2 years after upper secondary education)' WHEN '3' THEN 'Teacher training college level (3 years after upper secondary
education)' WHEN '4' THEN 'Teacher training university level (4 years after upper secondary education) or higher' WHEN '99' THEN 'Missing'
WHEN '88' THEN 'NA' ELSE [HTLVLVLTCH] END as [HTLVLVLTCH] ,CASE [HTEXCTCH] WHEN '0' THEN 'Never been an excellent teacher' WHEN '1' THEN 'Yes,
school level' WHEN '2' THEN 'Yes, district level' WHEN '3' THEN 'Yes, province level or higher' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA'
ELSE [HTEXCTCH] ,CASE [HTYPSCH] WHEN '2' THEN 'Private' WHEN '3' THEN 'Other' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE
[HTYPSCH] END as [HTYPSCH] ,CASE [HTLWSGRD] WHEN '1' THEN 'Grade 1' WHEN '2' THEN 'Grade 2' WHEN '3'
'Then 'Grade 3' WHEN '4' THEN 'Grade 4' WHEN '5' THEN 'Grade 5' WHEN '6' THEN 'Grade 6' WHEN '7' THEN 'Grade 7' WHEN '8' THEN 'Grade 8' WHEN
'9' THEN 'Grade 9' WHEN '10' THEN 'Grade 10' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTLWSGRD] END as [HTLWSGRD] ,CASE [HTHGHRD]
WHEN '11' THEN 'Grade 11' WHEN '12' THEN 'Grade 12' WHEN '10' THEN 'Grade 10' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTHGHRD]
END as [HTHGHRD] ,[HTNMSTEN] ,[HTTMNSTR] ,[HTAFTRSTR] ,[HTREWTC] ,[HTNCMCH] ,CASE [HTNCMCL] WHEN '0' THEN
'This school does not offer additional classes outside the normal school day' WHEN '1' THEN 'Remedial classes for weaker students' WHEN '2'
THEN 'Additional classes for excellent students' WHEN '3' THEN 'A mixture of remedial classes for weaker students and additional classes for
excellent students' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTNCMCL] END as [HTNCMCL] ,[HTCNMST] ,[HTPTNMST]
,[HTROMATH] ,[HTREOENG] ,[HTCHAPP] ,[HTREWTC1] ,[HTREWTC2] ,[HTREWTC3] ,[HTREWTC4] ,[HTREWTC5] ,[HTREWTC6] ,[HTREWTC7] ,[HTREWTC8] ,[HTDISTC1]
,[HTDISTC2] ,[HTDISTC3] ,[HTDISTC4] ,[HTDISTC5] ,[HTDISTC6] ,[HTDISTC7] ,[HTDISTC8] ,[HTGNADMN] ,[HTACRARE] ,[HTACRAETH] ,[HTACRAGND]
,[HTACRAPER] ,[HTACRAEXM] ,[HTACRAOTH] ,[HTAPAIID1] ,[HTEXM011] ,[HTEXM021] ,[HTEXM031] ,[HTEXM041] ,[HTEXM051] ,[HTEXM061]
,[HTEXM001] ,[HTAPAIID2] ,[HTEXM012] ,[HTEXM022] ,[HTEXM032] ,[HTEXM042] ,[HTEXM052] ,[HTEXM062] ,[HTAPAIID3] ,[HTEXM013]
,[HTEXM023] ,[HTEXM033] ,[HTEXM043] ,[HTEXM053] ,[HTEXM063] ,[HTEXM003] ,[HTAPAIID4] ,[HTEXM014] ,[HTEXM024] ,[HTEXM034]
,[HTEXM044] ,[HTEXM054] ,[HTEXM064] ,[HTAPAIID5] ,[HTEXM015] ,[HTEXM025] ,[HTEXM035] ,[HTEXM045] ,[HTEXM055] ,[HTEXM065]
,[HTEXM005] ,[HTAPAIID6] ,[HTEXM016] ,[HTEXM026] ,[HTEXM036] ,[HTEXM046] ,[HTEXM056] ,[HTEXM066] ,[HTAPAIID7] ,[HTEXM017]
,[HTEXM027] ,[HTEXM037] ,[HTEXM047] ,[HTEXM057] ,[HTEXM067] ,[HTAPAIID8] ,[HTEXM018] ,[HTEXM028] ,[HTEXM038] ,[HTEXM048]
,[HTEXM058] ,[HTEXM068] ,[HTAPAIID9] ,[HTEXM019] ,[HTEXM029] ,[HTEXM039] ,[HTEXM049] ,[HTEXM059] ,[HTEXM069] ,[HTAPAIID10]
,[HTEXM010] ,[HTEXM0210] ,[HTEXM0310] ,[HTEXM0410] ,[HTEXM0510] ,[HTEXM0610] ,[HTEXM0010] ,[HTAPAIID11] ,[HTEXM0111] ,[HTEXM0211]
,[HTEXM0311] ,[HTEXM0411] ,[HTEXM0511] ,[HTEXM0611] ,[HTAPAIID12] ,[HTEXM012] ,[HTEXM0212] ,[HTEXM0312] ,[HTEXM0412]
,[HTEXM0512] ,[HTEXM0612] ,[HTEXM0012] ,[HTENGR10] ,[HTLBV10] ,[HTENGR11] ,[HTENBY11] ,[HTTLGR11] ,[HTLBV11]
,[HTENGR12] ,[HTENBY12] ,[HTTLGR12] ,[HTNMC110] ,[HTALLC10] ,CASE [HTALLC10] WHEN '0' THEN 'There is only one class
in Grade 10' WHEN '1' THEN 'Randomly' WHEN '2' THEN 'Alphabetically' WHEN '3' THEN 'By Ability (e.g. high / middle / low)' WHEN '4' THEN
'According to the level of fees they pay' WHEN '5' THEN 'By choice of subject' WHEN '6' THEN 'Other method' WHEN '7' THEN 'A combination of
two or more of the above methods' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTALLC10] END as [HTALLC10] ,CASE [HTPRTEXP]
WHEN '1' THEN 'There is constant pressure from many parents who expect our school to set academic standards and to have our students ac' WHEN '2' THEN
'Pressure on the school to achieve higher academic standards among students comes from a minority of parents' WHEN '3' THEN 'Pressure from
parents on the school to achieve higher academic standards is largely absent' WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTPRTEXP]
END as [HTPRTEXP]
,[HTENTOT10]

```

FROM CP002.dbo.vietnam\_wave\_1 ;

- Create table – HeadTeacher\_Questionnaire\_2017

```

CREATE TABLE HeadTeacher_Questionnaire_2017 ( HTP_ID_2 INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[HTNTCMP] VARCHAR(30) ,[HTDINT] VARCHAR(20)
,[HTSTFCMP] VARCHAR(50) ,[HTDEVPLN] BIT ,[HTPTA] BIT ,[HTPTAMT] VARCHAR(50) ,[HTEDOVS] TINYINT ,[HTSTDDRP] VARCHAR(20)
,[HTCHDLBR] VARCHAR(20) ,[HTFLSINF] VARCHAR(20) ,[HTTCHRET] VARCHAR(20) ,[HTRCHATT] VARCHAR(20) ,[HTENVEVT] VARCHAR(20)
,[HTHIRPRN] BIT ,[HTHIRTCH] BIT ,[HTHIRBOET] BIT ,[HTHIRDOET] BIT ,[HTHIRMOT] BIT ,[HTFIRPRN] BIT
,[HTFIRTC] BIT ,[HTFIRBOET] BIT ,[HTFIRDOET] BIT ,[HTFIRMOET] BIT ,[HTFIRMOT] BIT ,[HTESTPRN] BIT ,[HTESTTCH] BIT
,[HTESTBOET] BIT ,[HTESTDOET] BIT ,[HTTESTMOET] BIT ,[HTTESTMOTH] BIT ,[HTDETPRN] BIT ,[HTDETCTCH] BIT ,[HTDETBOET] BIT
,[HTDETDOT] BIT ,[HTDETMOT] BIT ,[HTDETMOT] BIT ,[HTBUGPRN] BIT ,[HTBUGTCH] BIT ,[HTBUGBOET] BIT ,[HTBUGDOET] BIT
,[HTBUGMOET] BIT ,[HTBALPRN] BIT ,[HTBALTCH] BIT ,[HTBALBOET] BIT ,[HTBALDOET] BIT ,[HTBALMOET] BIT
,[HTBALMOTH] BIT ,[HTASPRN] BIT ,[HTSASTCH] BIT ,[HTSASBOET] BIT ,[HTSASDOET] BIT ,[HTSASMOET] BIT ,[HTXTPRN] BIT
,[HTXTTCH] BIT ,[HTXTBOET] BIT ,[HTXTDOET] BIT ,[HTXTMOET] BIT ,[HTCNPTRN] BIT ,[HTCNTTCH] BIT ,[HTCNTBOET] BIT
,[HTCNTDOET] BIT ,[HTCNPMT] BIT ,[HTOFRPGL] BIT ,[HTOFRTCH] BIT ,[HTOFRBOET] BIT ,[HTOFRDOET] BIT ,[HTOFRMOET] BIT
,[HTSTPFGL] VARCHAR(50) ,[HTCPFG] VARCHAR(50) ,[HTCWKGL] VARCHAR(50) ,[HTDISGGL] VARCHAR(50) ,[HTDYSTRN] TINYINT
,[HTPMTCRS] VARCHAR(50) ,[HTPRCTP] VARCHAR(50) ,[HTSTDVSK] VARCHAR(50) ,[HTPRBDIS] VARCHAR(50) ,[HTATTBHEH] VARCHAR(50)
,[HTSLVPRB] VARCHAR(50) ,[HTPRDEC] VARCHAR(50) ,[HTREVMNG] VARCHAR(50) ,[HTRSNABS] VARCHAR(50) ,[HTSHRPER] VARCHAR(50)
,[HTSTDENR] SMALLINT ,[HTOKEXM] SMALLINT ,[HTLSPSEX] SMALLINT ,[HTPRCUNI] SMALLINT ,[HTPRCVOC] SMALLINT ,[HTCNLRN] BIT
,[SAMEHEAD] VARCHAR(10) ,[DATECMPL] VARCHAR(20) ,CONSTRAINT FK_HTP_ID_2 FOREIGN KEY (UNIQUEID) REFERENCES
[dbEdu].[Identifier](UNIQUEID) )

```

- Insert into table – HeadTeacher\_Questionnaire\_2017 from dbo.vietnam\_wave\_2

```

INSERT INTO [CP002].[dbo].[HeadTeacher_Questionnaire_2017] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID
,CASE [HTNTCMP] WHEN '1' THEN 'Respondent absent' WHEN '2' THEN 'Respondent refused to participate' WHEN '79' THEN 'Missing' WHEN '88' THEN
'NA' ELSE [HTNTCMP] END as [HTNTCMP] ,[HTDINT]
,CASE [HTSTFCMP] WHEN '1' THEN 'Principal/Director' WHEN '2' THEN 'Vice principal/Vice director' WHEN '3' THEN 'Other member of senior
management staff' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTSTFCMP] END as [HTSTFCMP] ,[HTDEVPLN] ,[HTPTA]
,CASE [HTPTAMT] WHEN '0' THEN 'There is no Parents' Association' WHEN '1' THEN 'Once a week' WHEN '2' THEN 'Once a fortnight' WHEN '3' THEN
'Once a month' WHEN '4' THEN 'Once every 2-3 months' WHEN '5' THEN 'Once every 4-6 months' WHEN '6' THEN 'Once every 7-12 months' WHEN '7'
THEN 'Less than once a year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTPTAMT] END as [HTPTAMT]
,[HTEDOVS]
,CASE [HTSTDDRP] WHEN '1' THEN 'Never/Rarely' WHEN '2' THEN 'Occasionally' WHEN '3' THEN 'Often' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA'
ELSE [HTSTDDRP] END as [HTSTDDRP] ,CASE [HTCHDLBR] WHEN '1' THEN 'Never/Rarely' WHEN '2' THEN 'Occasionally' WHEN '3' THEN 'Often' WHEN '79'
THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTCHDLBR] ,CASE [HTFLSINF] WHEN '1' THEN 'Never/Rarely' WHEN '2' THEN 'Occasionally' WHEN '3' THEN
'Often' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTFLSINF] ,CASE [HTTCHRET] WHEN '1' THEN 'Never/Rarely' WHEN '2' THEN
'Occasionally' WHEN '3' THEN 'Often' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTTCHRET] END as [HTTCHRET] ,CASE [HTRCHATT] WHEN '1'
THEN 'Never/Rarely' WHEN '2' THEN 'Occasionally' WHEN '3' THEN 'Often' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTRCHATT] END as
,[HTENVEVT] ,CASE [HTENVEVT] WHEN '1' THEN 'Never/Rarely' WHEN '2' THEN 'Occasionally' WHEN '3' THEN 'Often' WHEN '79' THEN 'Missing' WHEN '88'
THEN 'NA' ELSE [HTENVEVT] END as [HTENVEVT] ,[HTHIRPRN]
,[HTHIRTCH] ,[HTHIRBOET] ,[HTHIRMOT] ,[HTFIRPRN] ,[HTFIRTC] ,[HTFIRBOET] ,[HTFIRDOET] ,[HTFIRMOET] ,[HTFIRMOT]
,[HTESTPRN] ,[HTESTTCH] ,[HTESTBOET] ,[HTESTDOET] ,[HTESTMOET] ,[HTDETPRN] ,[HTDETCTCH] ,[HTDETBOET] ,[HTDETDOT]
,[HTDETMOT] ,[HTBUGPRN] ,[HTBUGTCH] ,[HTBUGBOET] ,[HTBUGDOET] ,[HTBUGMOT] ,[HTBALPRN] ,[HTBALTCH] ,[HTBALBOET] ,[HTBALDOET]
,[HTBALMOT] ,[HTBALMOTH] ,[HTASPRN] ,[HTSASTCH] ,[HTSASBOET] ,[HTSASDOET] ,[HTSASMOET] ,[HTXTPRN] ,[HTXTTCH] ,[HTXTBOET]
,[HTXTDOET] ,[HTCNPTRN] ,[HTCNTTCH] ,[HTCNPMT] ,[HTCNTDOET] ,[HTOFRPGL] ,[HTOFRTCH] ,[HTOFRBOET] ,[HTOFRDOET] ,[HTOFRMOET]
,[HTSTPFGL] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN 'Once a month'
WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE
[HTSTPFGL] END as [HTSTPFGL] ,CASE [HTCPFG] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week'
WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN
'88' THEN 'NA' ELSE [HTCPFG] ,CASE [HTCWKGL] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week'
WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN
'88' THEN 'NA' ELSE [HTCWKGL] END as [HTCWKGL] ,CASE [HTDISGGL] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN
'2' THEN 'Once a week' WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN
'79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTDISGGL] END as [HTDISGGL] ,[HTDYSTRN]
,CASE [HTPMTCRS] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN 'Once a month'
WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE
[HTPMTCRS] END as [HTPMTCRS] ,CASE [HTPRCTP] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week'
WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN
'88' THEN 'NA' ELSE [HTPRCTP] END as [HTPRCTP] ,CASE [HTSTDVSK] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN
'2' THEN 'Once a week' WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN
'79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTSTDVSK] END as [HTSTDVSK] ,CASE [HTPRBDIS] WHEN '0' THEN 'Did not occur' WHEN '1' THEN
'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN
'1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTPRBDIS] END as [HTPRBDIS] ,CASE [HTATTBHEH] WHEN '0' THEN
'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times
during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTATTBHEH] END as
,[HTATTBHEH] ,CASE [HTSLVPRB] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN
'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN
'NA' ELSE [HTSLVPRB] END as [HTSLVPRB]
,CASE [HTPRDEC] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week' WHEN '3' THEN 'Once a month'
WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE
[HTPRDEC] END as [HTPRDEC] ,CASE [HTREVMNG] WHEN '0' THEN 'Did not occur' WHEN '1' THEN 'More than once a week' WHEN '2' THEN 'Once a week'
WHEN '3' THEN 'Once a month' WHEN '4' THEN '3-4 times during the year' WHEN '5' THEN '1-2 times during the year' WHEN '79' THEN 'Missing' WHEN
'88' THEN 'NA' ELSE [HTREVMNG] END as [HTREVMNG] ,CASE [HTRSNABS] WHEN '0' THEN 'No teacher absence in the last academic year' WHEN '1' THEN
'Leave for personal reasons' WHEN '2' THEN 'Illness / medical leave' WHEN '3' THEN 'Attending training' WHEN '4' THEN 'Official duties' WHEN
'5' THEN 'Teacher faced difficulties traveling to school' WHEN '6' THEN 'Unknown / reason not reported' WHEN '7' THEN 'Other' WHEN '79' THEN
'Missing' WHEN '88' THEN 'NA' ELSE [HTRSNABS] END as [HTRSNABS] ,CASE [HTSHRPER] WHEN '0' THEN 'Never' WHEN '1' THEN 'Once a year' WHEN '2'
THEN '2-3 times a year' WHEN '3' THEN 'More than 3 times a year' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [HTSHRPER] END as
,[HTSHRPER] ,[HTSTDENR] ,[HTOKEXM] ,[HTLSPSEX] ,[HTPRCUNI] ,[HTPRCVOC] ,[HTCNLRN]
,[SAMEHEAD] WHEN '0' THEN 'No' WHEN '1' THEN 'Yes' WHEN '79' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [SAMEHEAD] END as [SAMEHEAD]
,[DATECMPL]
FROM CP002.dbo.vietnam_wave_2 ;

```

- Create table – School\_Questionnaire\_2016
  - Insert into table - School Questionnaire 2016 from dbo.vietnam wave 1

```

CREATE TABLE School_Questionnaire_2016 ( SCH_ID INT IDENTITY(1,1) PRIMARY KEY ,UNIQUEID VARCHAR(9)
,[SCHFAC01A] TINYINT ,[SCHFAC01B] TINYINT ,[SCHFAC02] BIT ,[SCHFAC03] BIT ,[SCHFAC04] BIT ,[SCHFAC05] BIT ,[SCHFAC06] BIT
,[SCHFAC07] BIT ,[SCHFAC08] BIT ,[SCHFAC09] BIT ,[SCHFAC10] SMALLINT ,[SCHFAC11] BIT ,[SCHFAC12] VARCHAR(50) ,[SCHFAC13] BIT
,[SCHFAC14] BIT ,[WEIGHT] FLOAT ,CONSTRAINT FK_SCH_ID FOREIGN KEY (UNIQUEID) REFERENCES [dbo].[Identifier](UNIQUEID) )
--TRUNCATE TABLE [CP002].[dbo].[School_Questionnaire_2016];
INSERT INTO [CP002].[dbo].[School_Questionnaire_2016] SELECT RTRIM(LTRIM([UNIQUEID])) as UNIQUEID ,[SCHFAC01A] ,[SCHFAC01B]
,[SCHFAC02] ,[SCHFAC03] ,[SCHFAC04] ,[SCHFAC05] ,[SCHFAC06] ,[SCHFAC07] ,[SCHFAC08] ,[SCHFAC09] ,[SCHFAC10] ,[SCHFAC11] ,CASE
[SCHFAC12] WHEN '0' THEN 'No drinking water' WHEN '1' THEN 'School's own tap' WHEN '2' THEN 'Public tap' WHEN '3' THEN
'School''s own well' WHEN '4' THEN 'Public well' WHEN '5' THEN 'Rainwater' WHEN '6' THEN 'Spring' WHEN '7' THEN 'Tanker truck'
WHEN '8' THEN 'Bottled water provided to school' WHEN '9' THEN 'Students bring drinking water from home' WHEN '10' THEN 'Other'
WHEN '99' THEN 'Missing' WHEN '88' THEN 'NA' ELSE [SCHFAC12] END as [SCHFAC12] ,[SCHFAC13] ,[SCHFAC14] ,[WEIGHT]
FROM CP002.dbo.vietnam_wave_1 ;

```

- Create table – Teacher\_Questionnaire\_2017

- Insert into table - Teacher Questionnaire 2017 from dbo.vietnam wave 2

## 2.6.2 Views

- View 01 - v01 ST Questionnaire

- View 02 - v02 HT Questionnaire

- View 03 - v03\_ST\_ClassRoster

```

CREATE VIEW v08_ST_ClassRoster AS ( SELECT dbEdu.Identifier.UNIQUEID, dbEdu.Identifier.CLASSID, dbEdu.Student_ClassRoster_2016.GRLNRL,
dbEdu.Student_ClassRoster_2016.BOYENRL, dbEdu.Student_ClassRoster_2016.TTLNRL, dbEdu.Student_ClassRoster_2016.TGRLNR,
dbEdu.Student_ClassRoster_2016.TBOYENRL, dbEdu.Student_ClassRoster_2016.TTLENRL, dbEdu.Student_ClassRoster_2016.ATDMSY,
dbEdu.Student_ClassRoster_2016.ATDDFSY, dbEdu.Student_ClassRoster_2016.TGMTHENG, dbEdu.Student_ClassRoster_2016.SCALLCT,
dbEdu.Student_ClassRoster_2016.GRPABLY, dbEdu.Student_ClassRoster_2016.CLSORD1, dbEdu.Student_ClassRoster_2016.CLSORD2,
dbEdu.Student_ClassRoster_2016.SCAVLB1, dbEdu.Student_ClassRoster_2016.SCAVLB2, dbEdu.Student_ClassRoster_2016.SCAVLB3,
dbEdu.Student_ClassRoster_2016.SCAVLB4, dbEdu.Student_ClassRoster_2016.SCAVLB5, dbEdu.Student_ClassRoster_2016.SCAVLB6,
dbEdu.Student_ClassRoster_2016.SCAVLB7, dbEdu.Student_ClassRoster_2016.SCAVLB8, dbEdu.Student_ClassRoster_2016.SCAVLB9,
dbEdu.Student_ClassRoster_2016.SCAVLB10, dbEdu.Student_ClassRoster_2016.SPCRDAY, dbEdu.Student_ClassRoster_2016.SCLNONPR,
dbEdu.Student_ClassRoster_2016.SCMMTIN, dbEdu.Student_ClassRoster_2016.SCMNENIN, dbEdu.Student_ClassRoster_2016.SCTXTMTH,
dbEdu.Student_ClassRoster_2016.SCTXTENG, dbEdu.Student_ClassRoster_2017.SCTMID1, dbEdu.Student_ClassRoster_2017.W1MSMTC1,
dbEdu.Student_ClassRoster_2017.W1MMWTC1, dbEdu.Student_ClassRoster_2017.SCENID1, dbEdu.Student_ClassRoster_2017.W1ESMTC1,
dbEdu.Student_ClassRoster_2017.WIENWTC1, dbEdu.Student_ClassRoster_2017.REGTH, dbEdu.Student_ClassRoster_2017.ATDGIRL,
dbEdu.Student_ClassRoster_2017.ATDBOY, dbEdu.Student_ClassRoster_2017.ATDTLL, dbEdu.Student_ClassRoster_2017.SECSIZIE,
dbEdu.Student_ClassRoster_2017.NEWCHLD, dbEdu.Student_ClassRoster_2017.STDSTLNLR, dbEdu.Student_ClassRoster_2017.STDNORNE,
dbEdu.Student_ClassRoster_2017.STDNMBAS, dbEdu.Student_ClassRoster_2017.B1LVSLUPP, dbEdu.Student_ClassRoster_2017.B1MVTUSC,
dbEdu.Student_ClassRoster_2017.B2ACABMT, dbEdu.Student_ClassRoster_2017.B2PRCLMT, dbEdu.Student_ClassRoster_2017.B3ACABEN,
dbEdu.Student_ClassRoster_2017.B3PRCLEN, dbEdu.Identifier.SCHOOLID, dbEdu.Identifier.STUDENTID FROM dbEdu.Identifier INNER JOIN
dbEdu.Student_ClassRoster_2016 ON dbEdu.Identifier.UNIQUEID = dbEdu.Student_ClassRoster_2016.UNIQUEID INNER JOIN dbEdu.Student_ClassRoster_2017 ON
dbEdu.Student_ClassRoster_2016.UNIQUEID = dbEdu.Student_ClassRoster_2017.UNIQUEID ) ;

```

- View 04 - v04\_SCH\_Questionnaire

```

CREATE VIEW v04_SCH_Questionnaire AS ( SELECT dbEdu.Identifier.UNIQUEID, dbEdu.Identifier.SCHOOLID, dbEdu.Identifier.PROVINCE,
dbEdu.Identifier.DISTRICTCODE, dbEdu.Identifier.LOCALITY, dbEdu.School_Questionnaire_2016.SCHFAC01A,
dbEdu.School_Questionnaire_2016.SCHFAC01B, dbEdu.School_Questionnaire_2016.SCHFAC02, dbEdu.School_Questionnaire_2016.SCHFAC03,
dbEdu.School_Questionnaire_2016.SCHFAC04, dbEdu.School_Questionnaire_2016.SCHFAC05, dbEdu.School_Questionnaire_2016.SCHFAC06,
dbEdu.School_Questionnaire_2016.SCHFAC07, dbEdu.School_Questionnaire_2016.SCHFAC08, dbEdu.School_Questionnaire_2016.SCHFAC09,
dbEdu.School_Questionnaire_2016.SCHFAC10, dbEdu.School_Questionnaire_2016.SCHFAC11, dbEdu.School_Questionnaire_2016.SCHFAC12,
dbEdu.School_Questionnaire_2016.SCHFAC13, dbEdu.School_Questionnaire_2016.SCHFAC14, dbEdu.School_Questionnaire_2016.WEIGHT FROM
dbEdu.Identifier INNER JOIN dbEdu.School_Questionnaire_2016 ON dbEdu.Identifier.UNIQUEID =
dbEdu.School_Questionnaire_2016.UNIQUEID ) ;

```

- View 05 – v05\_TCH\_Questionnaire

- View 06 – v06\_ST\_Cogn\_Tests

```

CREATE VIEW v06_ST_Cogn_Tests AS ( SELECT [UNIQUEID],[STDINT], '2016' as [Year] ,[ENG_TEST] , [ENG_ITEM1] , [ENG_ITEM2] , [ENG_ITEM3] , [ENG_ITEM4]
,[ENG_ITEMS] , [ENG_ITEM6] , [ENG_ITEM7] , [ENG_ITEM8] , [ENG_ITEM9] , [ENG_ITEM10] , [ENG_ITEM11] , [ENG_ITEM12] , [ENG_ITEM13] , [ENG_ITEM14] , [ENG_ITEM15]
,[ENG_ITEM16] , [ENG_ITEM17] , [ENG_ITEM18] , [ENG_ITEM19] , [ENG_ITEM20] , [ENG_ITEM21] , [ENG_ITEM22] , [ENG_ITEM23] , [ENG_ITEM24] , [ENG_ITEM25]
,[ENG_ITEM26] , [ENG_ITEM27] , [ENG_ITEM28] , [ENG_ITEM29] , [ENG_ITEM30] , [ENG_ITEM31] , [ENG_ITEM32] , [ENG_ITEM33] , [ENG_ITEM34] , [ENG_ITEM35]
,[ENG_ITEM36] , [ENG_ITEM37] , [ENG_ITEM38] , [ENG_ITEM39] , [ENG_ITEM40] , [ENG_RAWSCORE] , [MATH_TEST] , [MATH_ITEM1] , [MATH_ITEM2] , [MATH_ITEM3]
,[MATH_ITEM4] , [MATH_ITEMS] , [MATH_ITEM7] , [MATH_ITEM8] , [MATH_ITEM9] , [MATH_ITEM10] , [MATH_ITEM11] , [MATH_ITEM12] , [MATH_ITEM13]
,[MATH_ITEM14] , [MATH_ITEM15] , [MATH_ITEM16] , [MATH_ITEM17] , [MATH_ITEM18] , [MATH_ITEM19] , [MATH_ITEM20] , [MATH_ITEM21] , [MATH_ITEM22] , [MATH_ITEM23]
,[MATH_ITEM24] , [MATH_ITEM25] , [MATH_ITEM26] , [MATH_ITEM27] , [MATH_ITEM28] , [MATH_ITEM29] , [MATH_ITEM30] , [MATH_ITEM31] , [MATH_ITEM32] , [MATH_ITEM33]
,[MATH_ITEM34] , [MATH_ITEM35] , [MATH_ITEM36] , [MATH_ITEM37] , [MATH_ITEM38] , [MATH_ITEM39] , [MATH_ITEM40] , [MATH_RAWSCORE] FROM
[CP002].[dbedu].[Student_CognitiveTests_2016] UNION SELECT [UNIQUEID] , [STDINT] , '2017' as [Year] ,[ENG_TEST] , [ENG_ITEM1] , [ENG_ITEM2] , [ENG_ITEM3]
,[ENG_ITEM4] , [ENG_ITEMS] , [ENG_ITEM6] , [ENG_ITEM7] , [ENG_ITEM8] , [ENG_ITEM9] , [ENG_ITEM10] , [ENG_ITEM11] , [ENG_ITEM12] , [ENG_ITEM13] , [ENG_ITEM14]
,[ENG_ITEM15] , [ENG_ITEM16] , [ENG_ITEM17] , [ENG_ITEM18] , [ENG_ITEM19] , [ENG_ITEM20] , [ENG_ITEM21] , [ENG_ITEM22] , [ENG_ITEM23] , [ENG_ITEM24]
,[ENG_ITEM25] , [ENG_ITEM26] , [ENG_ITEM27] , [ENG_ITEM28] , [ENG_ITEM29] , [ENG_ITEM30] , [ENG_ITEM31] , [ENG_ITEM32] , [ENG_ITEM33] , [ENG_ITEM34]
,[ENG_ITEM35] , [ENG_ITEM36] , [ENG_ITEM37] , [ENG_ITEM38] , [ENG_ITEM39] , [ENG_ITEM40] , [ENG_RAWSCORE] , [MATH_TEST] , [MATH_ITEM1] , [MATH_ITEM2]
,[MATH_ITEM3] , [MATH_ITEMS] , [MATH_ITEM6] , [MATH_ITEM7] , [MATH_ITEM8] , [MATH_ITEM9] , [MATH_ITEM10] , [MATH_ITEM11] , [MATH_ITEM12]
,[MATH_ITEM13] , [MATH_ITEM14] , [MATH_ITEM15] , [MATH_ITEM16] , [MATH_ITEM17] , [MATH_ITEM18] , [MATH_ITEM19] , [MATH_ITEM20] , [MATH_ITEM21] , [MATH_ITEM22]
,[MATH_ITEM23] , [MATH_ITEM24] , [MATH_ITEM25] , [MATH_ITEM26] , [MATH_ITEM27] , [MATH_ITEM28] , [MATH_ITEM29] , [MATH_ITEM30] , [MATH_ITEM31] , [MATH_ITEM32]
,[MATH_ITEM33] , [MATH_ITEM34] , [MATH_ITEM35] , [MATH_ITEM36] , [MATH_ITEM37] , [MATH_ITEM38] , [MATH_ITEM39] , [MATH_ITEM40] , [MATH_RAWSCORE] FROM
[CP002].[dbedu].[Student_CognitiveTests_2017]
);

```

- View 07 - v07\_ST\_TransfSkills\_Tests

```

CREATE VIEW v07_ST_TransfSkills_Tests AS ( SELECT dbEdu.Identifier.UNIQUEID, dbEdu.Identifier.SCHOOLID, dbEdu.Identifier.STUDENTID,
dbEdu.Identifier.YLCCHILDID, dbEdu.Identifier.PROVINCE, dbEdu.Identifier.DISTRICTCODE, dbEdu.Identifier.LOCALITY, dbEdu.Identifier.ABSENT_DAYS,
dbEdu.Identifier.MOM_READ, dbEdu.Identifier.MOM_EDUC, dbEdu.Identifier.DAD_READ, dbEdu.Identifier.DAD_EDUC, dbEdu.Identifier.STDVLCHD,
dbEdu.Identifier.STDCMPLT, dbEdu.Identifier.ETHNICITY, dbEdu.Identifier.AGE, dbEdu.Identifier.GENDER, dbEdu.Student_CognitiveTests_2017.TS_TEST,
dbEdu.Student_CognitiveTests_2017.TS_ITEM1, dbEdu.Student_CognitiveTests_2017.TS_ITEM2, dbEdu.Student_CognitiveTests_2017.TS_ITEM3A,
dbEdu.Student_CognitiveTests_2017.TS_ITEM3B, dbEdu.Student_CognitiveTests_2017.TS_ITEM3C, dbEdu.Student_CognitiveTests_2017.TS_ITEM3D,
dbEdu.Student_CognitiveTests_2017.TS_ITEM3E, dbEdu.Student_CognitiveTests_2017.TS_ITEM3F, dbEdu.Student_CognitiveTests_2017.TS_ITEM3,
dbEdu.Student_CognitiveTests_2017.TS_ITEM4, dbEdu.Student_CognitiveTests_2017.TS_ITEM5, dbEdu.Student_CognitiveTests_2017.TS_ITEM6,
dbEdu.Student_CognitiveTests_2017.TS_ITEM7, dbEdu.Student_CognitiveTests_2017.TS_ITEM8A, dbEdu.Student_CognitiveTests_2017.TS_ITEM8B,
dbEdu.Student_CognitiveTests_2017.TS_ITEM8C, dbEdu.Student_CognitiveTests_2017.TS_ITEM8B, dbEdu.Student_CognitiveTests_2017.TS_ITEM9,
dbEdu.Student_CognitiveTests_2017.TS_ITEM10, dbEdu.Student_CognitiveTests_2017.TS_ITEM11, dbEdu.Student_CognitiveTests_2017.TS_ITEM12,
dbEdu.Student_CognitiveTests_2017.TS_ITEM11_12, dbEdu.Student_CognitiveTests_2017.TS_ITEM13, dbEdu.Student_CognitiveTests_2017.TS_ITEM15,
dbEdu.Student_CognitiveTests_2017.TS_ITEM14, dbEdu.Student_CognitiveTests_2017.TS_ITEM16, dbEdu.Student_CognitiveTests_2017.TS_ITEM17,
dbEdu.Student_CognitiveTests_2017.TS_ITEM18, dbEdu.Student_CognitiveTests_2017.TS_ITEM19, dbEdu.Student_CognitiveTests_2017.TS_ITEM20,
dbEdu.Student_CognitiveTests_2017.TS_ITEM21, dbEdu.Student_CognitiveTests_2017.TS_ITEM22, dbEdu.Student_CognitiveTests_2017.TS_ITEM23 FROM
dbEdu.Identifier INNER JOIN dbEdu.Student_CognitiveTests_2017 ON dbEdu.Identifier.UNIQUEID = dbEdu.Student_CognitiveTests_2017.UNIQUEID
);

```

### 2.6.2.1 Reports

- Report 01 - yREP01\_ST\_Health\_Concerns

```
/*
/*Report 01 Creation - yREP01_ST_Health_Concerns*/
****

CREATE VIEW yREP01_ST_Health_Concerns
AS

SELECT PROVINCE as Province,
       GENDER as Gender,
       Health_Concern,
       SUM(CAST(Yes as tinyint)) as [Total]
FROM
(
    SELECT PROVINCE, GENDER, UNIQUEID, Health_Concern, Yes
    FROM
        (SELECT PROVINCE, GENDER, UNIQUEID,
               STDHLTH1 as [Sight problems],
               STDHLTH2 as [Hearing problems],
               STDHLTH3 as [Headaches],
               STDHLTH4 as [Fever],
               STDHLTH5 as [Stomach problems],
               STDHLTH6 as [Other problems]
        FROM CP002.dbEdu.v01_ST_Questionnaire) p
    UNPIVOT
        (Yes FOR Health_Concern IN
            ([Sight problems], [Hearing problems], [Headaches], [Fever], [Stomach problems], [Other
problems]))
    ) AS unpvt
) AS level2
WHERE LEN(LTRIM(RTRIM(GENDER))) > 0
GROUP BY PROVINCE, GENDER, Health_Concern
--ORDER BY PROVINCE, GENDER, Health_Concern ASC
GO
;
```

- Report 02 - yREP02\_ST\_Accessories

```
USE CP002;
GO
/*
/*Report 02 Creation - yREP02_ST_Accessories*/
****

CREATE VIEW yREP02_ST_Accessories
AS

SELECT PROVINCE as Province,
       LOCALITY as Locality,
       Accessories,
       SUM(IIF(CAST(Yes as tinyint) = 0, 1, 0)) as [No],
       SUM(CAST(Yes as tinyint)) as [Yes]
FROM
(
    SELECT PROVINCE, LOCALITY, UNIQUEID, Accessories, Yes
    FROM
        (SELECT PROVINCE, LOCALITY, UNIQUEID,
               STITMOW1 as [Compulsory Grade 10 Maths textbooks],
               STITMOW2 as [Compulsory Grade 10 English textbooks],
               STITMOW3 as [Non-compulsory Grade 10 Maths textbooks],
               STITMOW4 as [Non-compulsory Grade 10 English textbooks],
               STITMOW5 as [School bag],
               STITMOW6 as [Ruler],
               STITMOW7 as [Mobile phone],
               STITMOW8 as [Pocket Calculator]
        FROM CP002.dbEdu.v01_ST_Questionnaire) p
    UNPIVOT
        (Yes FOR Accessories IN
            ([Compulsory Grade 10 Maths textbooks],
             [Compulsory Grade 10 English textbooks],
             [Non-compulsory Grade 10 Maths textbooks],
             [Non-compulsory Grade 10 English textbooks],
             [School bag],
             [Ruler],
             [Mobile phone],
             [Pocket Calculator]))
    ) AS unpvt
) AS level2
GROUP BY PROVINCE, LOCALITY, Accessories
GO
;
```

- Report 03 - yREP03\_ST\_Motivation\_Levels

```
*****
/*Report 03 Creation - yREP03_ST_Motivation_Levels */
*****
CREATE VIEW yREP03_ST_Motivation_Levels

AS

SELECT Learning_Motivation, [Strongly disagree], [Disagree], [Agree], [Strongly agree] FROM ( SELECT UNIQUEID, Learning_Motivation, Levels
FROM (SELECT UNIQUEID, [STFEEL18A] as [I study to increase my job opportunities for a good type of work in the future], [STFEEL18B] as [I am
working hard in school to ensure that my future will be financially secure], [STFEEL18C] as [Making an effort in my studies is worth it
because it will help me in the work I want to do later on], [STFEEL18D] as [I want to learn as much as I can in school to help me go on to college/university] ,[STFEEL18F] as [I am
working hard in school to help me gain admission to higher studies] ,[STFEEL18G] as [Making an effort in my studies now is worthwhile because
it will help me in my studies later on] ,[STFEEL18H] as [Learning well in school will improve my work prospects and chances in the future]
,[STFEEL18I] as [I am working hard in school to be able to get work in the future and support my family in the future] ,[STFEEL18J] as [I want
to do well at school to help my brothers and sisters in the future] ,[STFEEL18K] as [Success in examinations will determine how successful I
am in later life] ,[STFEEL18L] as [Since success in the future is uncertain there is little use in learning very much.] ,[STFEEL19A] as [I day
dream a lot in class] ,[STFEEL19B] as [Most of my classmates are smarter than I am] ,[STFEEL19C] as [My teachers feel that I am poor in my
work] ,[STFEEL19D] as [I often forget what I have learnt] ,[STFEEL19E] as [I get frightened when I am asked a question by the teacher]
,[STFEEL19F] as [I often feel like quitting school] ,[STFEEL19G] as [I am always waiting for the lessons to end] ,[STFEEL19H] as [I always do
poorly in tests] ,[STFEEL19I] as [I am not willing to put in more effort in my school work] ,[STFEEL20A] as [I pay attention to the teacher
during lessons] ,[STFEEL20B] as [I am willing to do my best in class] ,[STFEEL20C] as [I study hard for my tests in school] ,[STFEEL20D] as
[If I put in enough effort I can succeed in my studies] ,[STFEEL20E] as [When studying, I work as hard as possible] ,[STFEEL20F] as [When
studying, I keep working even if the material is difficult] ,[STFEEL20G] as [When studying, I try to do my best to acquire the knowledge and
skills taught] ,[STFEEL20H] as [Setbacks in learning do not discourage me] ,[STFEEL20I] as [I am a hard worker in school] ,[STFEEL20J] as [I
am diligent in my studies] ,[STFEEL20K] as [I am conscientious in my studies] ,[STFEEL20L] as [When studying, I want to do as little work as
possible] ,[STFEEL20M] as [I want to do well in my studies, but only if the work is easy] ,[STFEEL20N] as [If the exercises in lessons are
difficult I just don't do them] ,[STFEEL20O] as [I choose easy options in school so that I don't have to work too hard] ,[STFEEL21A] as
[Studying gives me a lot of personal satisfaction] ,[STFEEL21B] as [I like studying because most of my subjects are really interesting]
,[STFEEL21C] as [I spend a lot of time working on topics I am interested in] ,[STFEEL21D] as [Keeping up with my studies helps to develop my
character] ,[STFEEL21E] as [Learning is an important personal experience] ,[STFEEL21F] as [Learning in school teaches me to become self-
disciplined] ,[STFEEL21G] as [I want to do well in school to show myself that I can learn new things] ,[STFEEL21H] as [I want to do well in my
studies to show myself that I can learn difficult school work] ,[STFEEL21I] as [I work hard at school because I am interested in what I am
learning] ,[STFEEL21J] as [Learning in school develops me as a person] ,[STFEEL22A] as [I get good marks in maths] ,[STFEEL22B] as [I'm
hopeless in maths] ,[STFEEL22C] as [Maths is one of my best subjects] ,[STFEEL22D] as [Maths is one of my worst subjects] ,[STFEEL22E] as [I
have always done well in maths] ,[STFEEL22F] as [I have never done well in maths] ,[STFEEL22G] as [I learn things quickly in maths]
,[STFEEL22H] as [I get poor marks in English] ,[STFEEL23A] as [I get good marks in English] ,[STFEEL23B] as [I'm hopeless in English]
,[STFEEL23C] as [English is one of my best subjects] ,[STFEEL23D] as [English is one of my worst subjects] ,[STFEEL23E] as [I have always done
well in English] ,[STFEEL23F] as [I have never done well in English] ,[STFEEL23G] as [I learn things quickly in English] --,[STFEEL23H] as [I
get poor marks in English] ,[STFEEL24A] as [It is important that I enjoy doing maths even if I do not do well in the examinations]
,[STFEEL24B] as [I study maths just to pass the examination] ,[STFEEL24C] as [In maths I study only those topics prescribed in the examination
syllabus] ,[STFEEL24D] as [I study in order to do well in the exam rather than to develop my interest in the subject] ,[STFEEL24E] as [I spend
a lot of time working on maths topics I am interested in even if they are not important for my examinations] ,[STFEEL25A] as [It is important
that I enjoy doing English even if I do not do well in the examinations] ,[STFEEL25B] as [I study English just to pass the examination]
,[STFEEL25C] as [In English I study only those topics prescribed in the examination syllabus] --,[STFEEL25D] as [I study in order to do well
in the exam rather than to develop my interest in the subject] ,[STFEEL25E] as [I spend a lot of time working on English topics I am
interested in even if they are not important for my examinations] ,[STFEEL26A] as [I like to handle situations that require a lot of thinking]
,[STFEEL26B] as [Thinking is not something I find fun] ,[STFEEL26C] as [I would rather do something that requires little thought than
something that will challenge my thinking abilities] ,[STFEEL26D] as [I try to avoid situations where I might have to think in depth about
something] ,[STFEEL26E] as [I like tasks that do not require much thought once I've learned them] ,[STFEEL26F] as [I really enjoy a task that
involves coming up with new solutions to problems] ,[STFEEL26G] as [Learning new ways to think doesn't excite me very much] ,[STFEEL26H] as [I
prefer my life to be filled with puzzles that I must solve] ,[STFEEL26I] as [I feel relief rather than satisfaction after completing a task
that required a lot of mental effort] FROM CP002.dbEdu.v01_ST_Questionnaire) p UNPIVOT (Levels FOR Learning_Motivation IN ( [I study to
increase my job opportunities for a good type of work in the future], [I am working hard in school to ensure that my future will be
financially secure], [Making an effort in my studies is worth it because it will help me in the work I want to do later on], [I want to learn
as much as I can in school to help me get good work in the future], [I want to learn as much as I can in school to help me go on to
college/university], [I am working hard in school to help me gain admission to higher studies], [Making an effort in my studies now is
worthwhile because it will help me in my studies later on], [Learning well in school will improve my work prospects and chances in the
future], [I am working hard in school to be able to get work in the future and support my family in the future], [I always do poorly in tests]
,[I am diligent in my studies], [Success in examinations will determine how successful I am in later life], [Since success in
the future is uncertain there is little use in learning very much.], [I day dream a lot in class], [Most of my classmates are smarter than I
am], [My teachers feel that I am poor in my work], [I often forget what I have learnt], [I get frightened when I am asked a question by the
teacher], [I often feel like quitting school], [I am always waiting for the lessons to end], [I always do poorly in tests], [I am not willing
to put in more effort in my school work], [I pay attention to the teacher during lessons], [I am willing to do my best in class], [I study
hard for my tests in school], [If I put in enough effort I can succeed in my studies], [When studying, I work as hard as possible], [When
studying, I keep working even if the material is difficult], [When studying, I try to do my best to acquire the knowledge and skills taught],
[Setbacks in learning do not discourage me], [I am a hard worker in school], [I am diligent in my studies], [I am conscientious in my
studies], [When studying, I want to do as little work as possible], [I want to do well in my studies, but only if the work is easy], [If the
exercises in lessons are difficult I just don't do them], [I choose easy options in school so that I don't have to work too hard], [Studying
gives me a lot of personal satisfaction], [I like studying because most of my subjects are really interesting], [I spend a lot of time working
on topics I am interested in], [Keeping up with my studies helps to develop my character], [Learning is an important personal experience],
[Learning in school teaches me to become self-disciplined], [I want to do well in school to show myself that I can learn new things], [I want
to do well in my studies to show myself that I can learn difficult school work], [I work hard at school because I am interested in what I am
learning], [Learning in school develops me as a person], [I get good marks in maths], [I'm hopeless in English], [Maths is one of my best
subjects], [Maths is one of my worst subjects], [I have always done well in maths], [I have never done well in maths], [I learn things quickly
in maths], [I get poor marks in English], [I get good marks in English], [I'm hopeless in English], [English is one of my best subjects],
[English is one of my worst subjects], [I have always done well in English], [I have never done well in English], [I learn things quickly in
English], --,[I get poor marks in English], [It is important that I enjoy doing maths even if I do not do well in the examinations], [I study
maths just to pass the examination], [In maths I study only those topics prescribed in the examination syllabus], [I study in order to do well
in the exam rather than to develop my interest in the subject], [I spend a lot of time working on maths topics I am interested in even if they
are not important for my examinations], [It is important that I enjoy doing English even if I do not do well in the examinations], [I study
English just to pass the examination], [In English I study only those topics prescribed in the examination syllabus], --,[I study in order to
do well in the exam rather than to develop my interest in the subject], [I spend a lot of time working on English topics I am interested in
even if they are not important for my examinations], [I like to handle situations that require a lot of thinking], [Thinking is not something
I find fun], [I would rather do something that requires little thought than something that will challenge my thinking abilities], [I try to
avoid situations where I might have to think in depth about something], [I like tasks that do not require much thought once I've learned
them], [I really enjoy a task that involves coming up with new solutions to problems], [Learning new ways to think doesn't excite me very
much], [I prefer my life to be filled with puzzles that I must solve], [I feel relief rather than satisfaction after completing a task that
required a lot of mental effort]
```

) AS unpvt ) as unpvt\_level2 PIVOT ( COUNT(UNIQUEID) FOR Levels IN ( [Strongly disagree], [Disagree], [Agree], [Strongly agree] ) ) as pvt --
ORDER BY pvt.Learning\_Motivation GO ;

- Report 04 - yREP04\_SCH\_Pain\_Points

```
/*
*****Report 04 Creation - yREP04_SCH_Pain_Points*****
*****Report 04 Creation - yREP04_SCH_Pain_Points*****
```

```
CREATE VIEW yREP04_SCH_Pain_Points
AS
SELECT PROVINCE AS Province,
       LOCALITY AS Locality,
       SCHOOLID AS School_ID,
       COUNT(UNIQUEID) AS [# Students],
       SCHFAC01A AS [# Classrooms],
       SCHFAC01B AS [# Laboratories],
       SCHFAC10          AS [# Working toilets],
       SCHFAC12          AS [Main Source of Drinking water],
       IIF(CAST(SCHFAC03 AS TINYINT) = 1, 'Yes', 'No') AS [Electricity Supply]
FROM CP002.dbEdu.v04_SCH_Questionnaire
GROUP BY PROVINCE, LOCALITY, SCHOOLID, SCHFAC01A, SCHFAC01B, SCHFAC10, SCHFAC12, SCHFAC03
GO ;
```

- Report 05 - yREP05\_ST\_Cogn\_Eng\_Tests

```
/*
*****Report 05 Creation - yREP05_ST_Cogn_Eng_Tests*****
*****Report 05 Creation - yREP05_ST_Cogn_Eng_Tests*****
```

```
CREATE VIEW yREP05_ST_Cogn_Eng_Tests
AS
SELECT Province+'_'+Locality+'_'+SchoolID+'_'+Gender+'_'+Year AS CK_Test,
       Province, Locality, SchoolID, Gender, Year, Eng_Questions, Num_Crct
FROM
(
    SELECT [PROVINCE] AS [Province],
           [LOCALITY] AS Locality,
           [SCHOOLID],
           IIF(LEN(LTRIM(RTRIM([GENDER]))) > 0, [Gender], 'Neutral') AS [Gender],
           [Year],
           SUM(IIF(CAST([ENG_ITEM1] AS TINYINT) = 1, 1, 0)) AS [Q01_ENG],
           SUM(IIF(CAST([ENG_ITEM2] AS TINYINT) = 1, 1, 0)) AS [Q02_ENG],
           SUM(IIF(CAST([ENG_ITEM3] AS TINYINT) = 1, 1, 0)) AS [Q03_ENG],
           SUM(IIF(CAST([ENG_ITEM4] AS TINYINT) = 1, 1, 0)) AS [Q04_ENG],
           SUM(IIF(CAST([ENG_ITEM5] AS TINYINT) = 1, 1, 0)) AS [Q05_ENG],
           SUM(IIF(CAST([ENG_ITEM6] AS TINYINT) = 1, 1, 0)) AS [Q06_ENG],
           SUM(IIF(CAST([ENG_ITEM7] AS TINYINT) = 1, 1, 0)) AS [Q07_ENG],
           SUM(IIF(CAST([ENG_ITEM8] AS TINYINT) = 1, 1, 0)) AS [Q08_ENG],
           SUM(IIF(CAST([ENG_ITEM9] AS TINYINT) = 1, 1, 0)) AS [Q09_ENG],
           SUM(IIF(CAST([ENG_ITEM10] AS TINYINT) = 1, 1, 0)) AS [Q10_ENG],
           SUM(IIF(CAST([ENG_ITEM11] AS TINYINT) = 1, 1, 0)) AS [Q11_ENG],
           SUM(IIF(CAST([ENG_ITEM12] AS TINYINT) = 1, 1, 0)) AS [Q12_ENG],
           SUM(IIF(CAST([ENG_ITEM13] AS TINYINT) = 1, 1, 0)) AS [Q13_ENG],
           SUM(IIF(CAST([ENG_ITEM14] AS TINYINT) = 1, 1, 0)) AS [Q14_ENG],
           SUM(IIF(CAST([ENG_ITEM15] AS TINYINT) = 1, 1, 0)) AS [Q15_ENG],
           SUM(IIF(CAST([ENG_ITEM16] AS TINYINT) = 1, 1, 0)) AS [Q16_ENG],
           SUM(IIF(CAST([ENG_ITEM17] AS TINYINT) = 1, 1, 0)) AS [Q17_ENG],
           SUM(IIF(CAST([ENG_ITEM18] AS TINYINT) = 1, 1, 0)) AS [Q18_ENG],
           SUM(IIF(CAST([ENG_ITEM19] AS TINYINT) = 1, 1, 0)) AS [Q19_ENG],
           SUM(IIF(CAST([ENG_ITEM20] AS TINYINT) = 1, 1, 0)) AS [Q20_ENG],
           SUM(IIF(CAST([ENG_ITEM21] AS TINYINT) = 1, 1, 0)) AS [Q21_ENG],
           SUM(IIF(CAST([ENG_ITEM22] AS TINYINT) = 1, 1, 0)) AS [Q22_ENG],
           SUM(IIF(CAST([ENG_ITEM23] AS TINYINT) = 1, 1, 0)) AS [Q23_ENG],
           SUM(IIF(CAST([ENG_ITEM24] AS TINYINT) = 1, 1, 0)) AS [Q24_ENG],
           SUM(IIF(CAST([ENG_ITEM25] AS TINYINT) = 1, 1, 0)) AS [Q25_ENG],
           SUM(IIF(CAST([ENG_ITEM26] AS TINYINT) = 1, 1, 0)) AS [Q26_ENG],
           SUM(IIF(CAST([ENG_ITEM27] AS TINYINT) = 1, 1, 0)) AS [Q27_ENG],
           SUM(IIF(CAST([ENG_ITEM28] AS TINYINT) = 1, 1, 0)) AS [Q28_ENG],
           SUM(IIF(CAST([ENG_ITEM29] AS TINYINT) = 1, 1, 0)) AS [Q29_ENG],
           SUM(IIF(CAST([ENG_ITEM30] AS TINYINT) = 1, 1, 0)) AS [Q30_ENG],
           SUM(IIF(CAST([ENG_ITEM31] AS TINYINT) = 1, 1, 0)) AS [Q31_ENG],
           SUM(IIF(CAST([ENG_ITEM32] AS TINYINT) = 1, 1, 0)) AS [Q32_ENG],
           SUM(IIF(CAST([ENG_ITEM33] AS TINYINT) = 1, 1, 0)) AS [Q33_ENG],
           SUM(IIF(CAST([ENG_ITEM34] AS TINYINT) = 1, 1, 0)) AS [Q34_ENG],
           SUM(IIF(CAST([ENG_ITEM35] AS TINYINT) = 1, 1, 0)) AS [Q35_ENG],
           SUM(IIF(CAST([ENG_ITEM36] AS TINYINT) = 1, 1, 0)) AS [Q36_ENG],
           SUM(IIF(CAST([ENG_ITEM37] AS TINYINT) = 1, 1, 0)) AS [Q37_ENG],
           SUM(IIF(CAST([ENG_ITEM38] AS TINYINT) = 1, 1, 0)) AS [Q38_ENG],
           SUM(IIF(CAST([ENG_ITEM39] AS TINYINT) = 1, 1, 0)) AS [Q39_ENG],
           SUM(IIF(CAST([ENG_ITEM40] AS TINYINT) = 1, 1, 0)) AS [Q40_ENG]
           --,ROUND(AVG(CAST([ENG_RAWSCORE] AS REAL)), 3) AS [Average Score in English Test]
        FROM [dbo].[v06_ST_Cogn_Tests] AS A
        INNER JOIN dbEdu Identifier AS J
        ON A.UNIQUEID = J.UNIQUEID
        WHERE [ENG_TEST] = 'Yes'
        GROUP BY [PROVINCE], [LOCALITY], [SCHOOLID], [GENDER], [YEAR] ) AS lvl1
        UNPIVOT
        (Num_Crct FOR Eng_Questions IN
        ([Q01_ENG], [Q02_ENG], [Q03_ENG], [Q04_ENG], [Q05_ENG], [Q06_ENG], [Q07_ENG], [Q08_ENG],
        [Q09_ENG], [Q10_ENG], [Q11_ENG], [Q12_ENG], [Q13_ENG], [Q14_ENG], [Q15_ENG], [Q16_ENG],
        [Q17_ENG], [Q18_ENG], [Q19_ENG], [Q20_ENG], [Q21_ENG], [Q22_ENG], [Q23_ENG], [Q24_ENG],
        [Q25_ENG], [Q26_ENG], [Q27_ENG], [Q28_ENG], [Q29_ENG], [Q30_ENG], [Q31_ENG], [Q32_ENG],
        [Q33_ENG], [Q34_ENG], [Q35_ENG], [Q36_ENG], [Q37_ENG], [Q38_ENG], [Q39_ENG], [Q40_ENG])
        ) AS unpvt GO;
```

- Report 06 - yREP06\_ST\_Cogn\_Math\_Tests

```

/*
*****Report 06 Creation - yREP06_ST_Cogn_Math_Tests*****
*/

CREATE VIEW yREP06_ST_Cogn_Math_Tests
AS
SELECT Province+'_'+Locality+'_'+SchoolID+'_'+Gender+'_'+Year AS CK_Test,
       Province, Locality, SchoolID, Gender, Year, Math_Questions, Num_Crct
FROM
(
    SELECT [PROVINCE] AS [Province],
           [LOCALITY] AS Locality,
           [SCHOOLID],
           IIF(LEN(LTRIM(RTRIM([GENDER]))) > 0, [Gender], 'Neutral') AS [Gender],
           [Year]
           ,SUM(IIF(CAST([MATH_ITEM1] AS TINYINT) = 1, 1, 0)) AS [Q01_MAT]
           ,SUM(IIF(CAST([MATH_ITEM2] AS TINYINT) = 1, 1, 0)) AS [Q02_MAT]
           ,SUM(IIF(CAST([MATH_ITEM3] AS TINYINT) = 1, 1, 0)) AS [Q03_MAT]
           ,SUM(IIF(CAST([MATH_ITEM4] AS TINYINT) = 1, 1, 0)) AS [Q04_MAT]
           ,SUM(IIF(CAST([MATH_ITEM5] AS TINYINT) = 1, 1, 0)) AS [Q05_MAT]
           ,SUM(IIF(CAST([MATH_ITEM6] AS TINYINT) = 1, 1, 0)) AS [Q06_MAT]
           ,SUM(IIF(CAST([MATH_ITEM7] AS TINYINT) = 1, 1, 0)) AS [Q07_MAT]
           ,SUM(IIF(CAST([MATH_ITEM8] AS TINYINT) = 1, 1, 0)) AS [Q08_MAT]
           ,SUM(IIF(CAST([MATH_ITEM9] AS TINYINT) = 1, 1, 0)) AS [Q09_MAT]
           ,SUM(IIF(CAST([MATH_ITEM10] AS TINYINT) = 1, 1, 0)) AS [Q10_MAT]
           ,SUM(IIF(CAST([MATH_ITEM11] AS TINYINT) = 1, 1, 0)) AS [Q11_MAT]
           ,SUM(IIF(CAST([MATH_ITEM12] AS TINYINT) = 1, 1, 0)) AS [Q12_MAT]
           ,SUM(IIF(CAST([MATH_ITEM13] AS TINYINT) = 1, 1, 0)) AS [Q13_MAT]
           ,SUM(IIF(CAST([MATH_ITEM14] AS TINYINT) = 1, 1, 0)) AS [Q14_MAT]
           ,SUM(IIF(CAST([MATH_ITEM15] AS TINYINT) = 1, 1, 0)) AS [Q15_MAT]
           ,SUM(IIF(CAST([MATH_ITEM16] AS TINYINT) = 1, 1, 0)) AS [Q16_MAT]
           ,SUM(IIF(CAST([MATH_ITEM17] AS TINYINT) = 1, 1, 0)) AS [Q17_MAT]
           ,SUM(IIF(CAST([MATH_ITEM18] AS TINYINT) = 1, 1, 0)) AS [Q18_MAT]
           ,SUM(IIF(CAST([MATH_ITEM19] AS TINYINT) = 1, 1, 0)) AS [Q19_MAT]
           ,SUM(IIF(CAST([MATH_ITEM20] AS TINYINT) = 1, 1, 0)) AS [Q20_MAT]
           ,SUM(IIF(CAST([MATH_ITEM21] AS TINYINT) = 1, 1, 0)) AS [Q21_MAT]
           ,SUM(IIF(CAST([MATH_ITEM22] AS TINYINT) = 1, 1, 0)) AS [Q22_MAT]
           ,SUM(IIF(CAST([MATH_ITEM23] AS TINYINT) = 1, 1, 0)) AS [Q23_MAT]
           ,SUM(IIF(CAST([MATH_ITEM24] AS TINYINT) = 1, 1, 0)) AS [Q24_MAT]
           ,SUM(IIF(CAST([MATH_ITEM25] AS TINYINT) = 1, 1, 0)) AS [Q25_MAT]
           ,SUM(IIF(CAST([MATH_ITEM26] AS TINYINT) = 1, 1, 0)) AS [Q26_MAT]
           ,SUM(IIF(CAST([MATH_ITEM27] AS TINYINT) = 1, 1, 0)) AS [Q27_MAT]
           ,SUM(IIF(CAST([MATH_ITEM28] AS TINYINT) = 1, 1, 0)) AS [Q28_MAT]
           ,SUM(IIF(CAST([MATH_ITEM29] AS TINYINT) = 1, 1, 0)) AS [Q29_MAT]
           ,SUM(IIF(CAST([MATH_ITEM30] AS TINYINT) = 1, 1, 0)) AS [Q30_MAT]
           ,SUM(IIF(CAST([MATH_ITEM31] AS TINYINT) = 1, 1, 0)) AS [Q31_MAT]
           ,SUM(IIF(CAST([MATH_ITEM32] AS TINYINT) = 1, 1, 0)) AS [Q32_MAT]
           ,SUM(IIF(CAST([MATH_ITEM33] AS TINYINT) = 1, 1, 0)) AS [Q33_MAT]
           ,SUM(IIF(CAST([MATH_ITEM34] AS TINYINT) = 1, 1, 0)) AS [Q34_MAT]
           ,SUM(IIF(CAST([MATH_ITEM35] AS TINYINT) = 1, 1, 0)) AS [Q35_MAT]
           ,SUM(IIF(CAST([MATH_ITEM36] AS TINYINT) = 1, 1, 0)) AS [Q36_MAT]
           ,SUM(IIF(CAST([MATH_ITEM37] AS TINYINT) = 1, 1, 0)) AS [Q37_MAT]
           ,SUM(IIF(CAST([MATH_ITEM38] AS TINYINT) = 1, 1, 0)) AS [Q38_MAT]
           ,SUM(IIF(CAST([MATH_ITEM39] AS TINYINT) = 1, 1, 0)) AS [Q39_MAT]
           ,SUM(IIF(CAST([MATH_ITEM40] AS TINYINT) = 1, 1, 0)) AS [Q40_MAT]
           ,ROUND(AVG(CAST([MATH_RAWSCORE] AS REAL)), 3) AS [Average Score in Maths Test]
    FROM [dbo].[v06_ST_Cogn_Tests] AS A
    INNER JOIN dbEdu.Identifier AS J
        ON A.UNIQUEID = J.UNIQUEID
    WHERE [ENG_TEST] = 'Yes'
    GROUP BY [PROVINCE], [LOCALITY], [SCHOOLID], [GENDER], [Year]) AS lvl1
    UNPIVOT
        (Num_Crct FOR Math_Questions IN
            ([Q01_MAT], [Q02_MAT], [Q03_MAT], [Q04_MAT], [Q05_MAT], [Q06_MAT],
             [Q07_MAT], [Q08_MAT],
             [Q09_MAT], [Q10_MAT], [Q11_MAT], [Q12_MAT], [Q13_MAT], [Q14_MAT],
             [Q15_MAT], [Q16_MAT],
             [Q17_MAT], [Q18_MAT], [Q19_MAT], [Q20_MAT], [Q21_MAT], [Q22_MAT], [Q23_MAT], [Q24_MAT],
             [Q25_MAT], [Q26_MAT], [Q27_MAT], [Q28_MAT], [Q29_MAT],
             [Q30_MAT], [Q31_MAT], [Q32_MAT],
             [Q33_MAT], [Q34_MAT], [Q35_MAT], [Q36_MAT], [Q37_MAT], [Q38_MAT], [Q39_MAT], [Q40_MAT])
        ) AS unpvt GO;

```

- Report 07 - yREP07\_ST\_Avg\_Test\_Scores

```
/*
*****Report 07 Creation - yREP07_ST_Avg_Test_Scores*****
*/

CREATE VIEW yREP07_ST_Avg_Test_Scores
AS
SELECT t1.Province+'_'+t1.Locality+'_'+t1.SchoolID+'_'+t1.Gender+'_'+t1.Year AS CK_Test,
       t1.Province, t1.Locality, t1.SchoolID, t1.Gender, t1.Year,
       t1.[Average Score in English Test],
       t2.[Average Score in Maths Test]
FROM
    (SELECT [PROVINCE] AS [Province],
            [LOCALITY] AS Locality,
            [SCHOOLID],
            IIF(LEN(LTRIM(RTRIM([GENDER]))) > 0, [Gender], 'Neutral') AS [Gender],
            [Year]
            ,ROUND(AVG(CAST([ENG_RAWSCORE] AS REAL)), 3) AS [Average Score in English Test]
    FROM [dbo].[v06_ST_Cogn_Tests] AS A
    INNER JOIN dbEdu.Identifier AS J
        ON A.UNIQUEID = J.UNIQUEID
    WHERE [ENG_TEST] = 'Yes'
    GROUP BY [PROVINCE],[LOCALITY], [SCHOOLID], [GENDER], [Year] ) AS t1
LEFT JOIN
    (SELECT [PROVINCE] AS [Province],
            [LOCALITY] AS Locality,
            [SCHOOLID],
            IIF(LEN(LTRIM(RTRIM([GENDER]))) > 0, [Gender], 'Neutral') AS [Gender],
            [Year]
            ,ROUND(AVG(CAST([MATH_RAWSCORE] AS REAL)), 3) AS [Average Score in Maths Test]
    FROM [dbo].[v06_ST_Cogn_Tests] AS A
    INNER JOIN dbEdu.Identifier AS J
        ON A.UNIQUEID = J.UNIQUEID
    WHERE [ENG_TEST] = 'Yes'
    GROUP BY [PROVINCE],[LOCALITY], [SCHOOLID], [GENDER], [Year] ) AS t2
ON (t1.[Province]=t2.[Province] AND
    t1.Locality=t2.Locality AND
    t1.SchoolID=t2.SchoolID AND
    t1.Gender=t2.Gender AND
    t1.Year=t2.Year) GO;
```

## 2.6.3 Stored Procedures (if any)

N/A

## 2.6.4 Triggers (if any)

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Trigger 1 - Intercept Deletion
-- =====
-- Author: Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion
    ON [dbEdu].[Identifier]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO
```

```

-- =====
-- Trigger 2 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:     Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion2
    ON [dbEdu].[HeadTeacher_Questionnaire_2016]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion2` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 3 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion3
    ON [dbEdu].[HeadTeacher_Questionnaire_2017]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion3` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 4 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion4
    ON [dbEdu].[School_Questionnaire_2016]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion4` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 5 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion5
    ON [dbEdu].[Student_ClassRoster_2016]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion5` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 6 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion6
    ON [dbEdu].[Student_ClassRoster_2017]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion6` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

```

```

-- =====
-- Trigger 7 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion7
    ON [dbEdu].[Student_CognitiveTests_2016]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion7` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 8 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:     Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion8
    ON [dbEdu].[Student_CognitiveTests_2017]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion8` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 9 - Intercept Deletion
-- =====
-- Author:           Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description:     Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion9
    ON [dbEdu].[Student_Questionnaire_2016]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion9` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 10 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion10
    ON [dbEdu].[Student_Questionnaire_2017]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion10` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 11 - Intercept Deletion
-- =====
CREATE TRIGGER dbEdu.trg_Intercept_Deletion11
    ON [dbEdu].[Teacher_Questionnaire_2017]
    INSTEAD OF DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion11` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

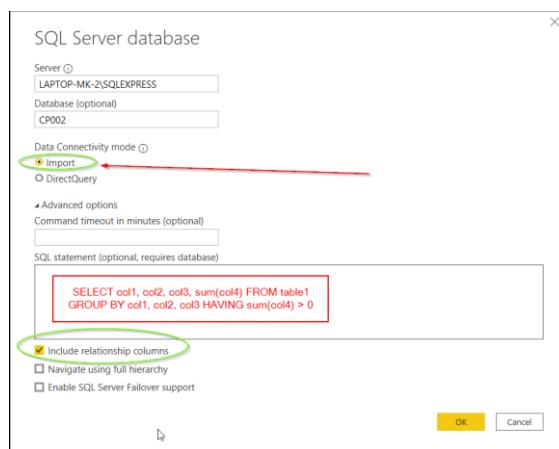
```

## 2.6.5 User Define Functions (if any)

N/A

## 2.7. Report Design (Only for task 2 and task 3)

- The above created views are accessed from Microsoft Power BI / MS Excel. I used both to showcase proficiency.
- Initially, we select SQL server database as my data source for my Power BI report, and select 'Import' as my Data connectivity mode. This will extract the data from SQL and store in the .pbix file.
- Server is 'LAPTOP-MK-2\SQLEXPRESS' and Database is 'CP002'
- Alternatively, I could write queries in the SQL statement text box highlighted below in red. This can enable reporting on the go.
- Reports 01 to 06 are all used to prepare dashboard for Inequality in Education sector – Vietnam.

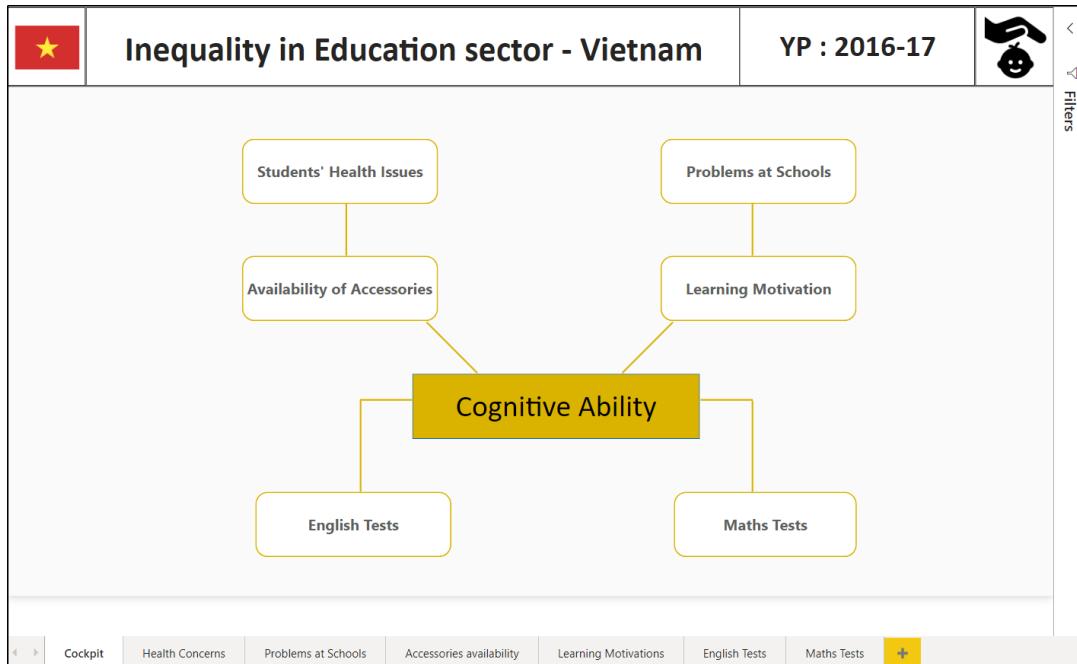


- We select the required reports from the next pop-up screen and the transformation steps open in Power BI. I explicitly didn't transform any data because this assignment would be based on SQL expertise, hence all these reports in the Power BI environment are ready to be visualized. Filtering, sorting, grouping – all these tasks were already performed at the source level itself.

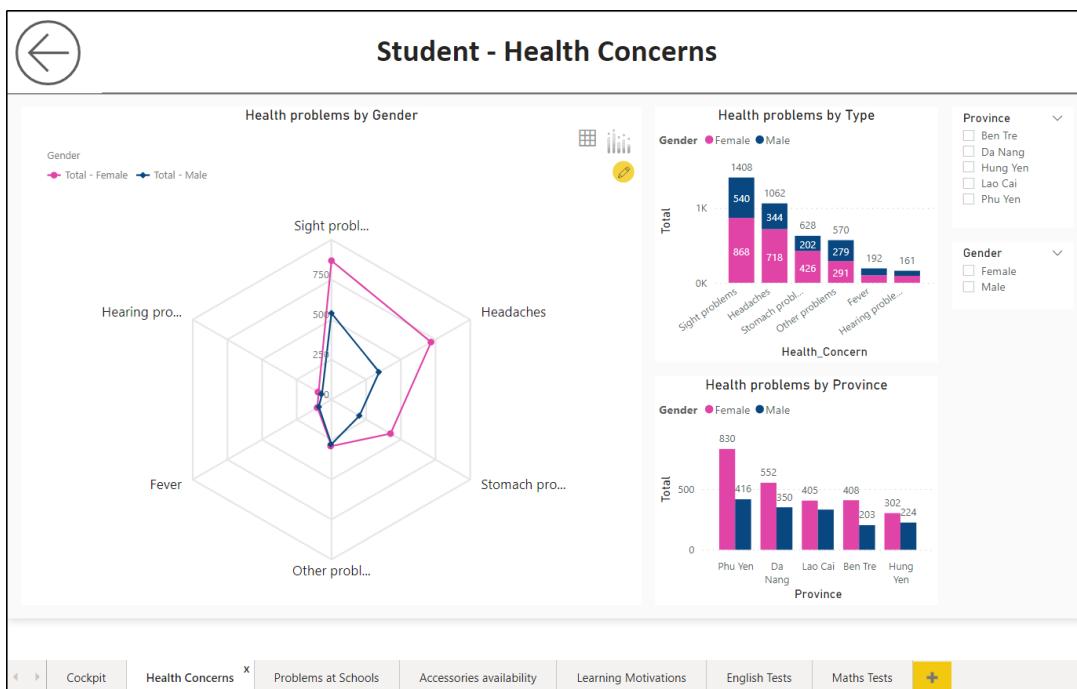
	A <sub>B</sub> Province	A <sub>B</sub> Gender	A <sub>B</sub> Health Concern
1	Lao Cai	Female	Hearing problem
2	Da Nang	Male	Sight problem
3	Lao Cai	Female	Fever
4	Hung Yen	Male	Fever
5	Hung Yen	Male	Stomach pain
6	Phu Yen	Female	Sight problem
7	Phu Yen	Male	Other problem
8	Lao Cai	Male	Fever

- After the designing of the report is completed, this is how the final dashboard looks like.
- We have mainly focused on 5 topics
  - i. Student's Health issues

- ii. Problems at Schools
- iii. Availability of Accessories
- iv. Student's Learning motivation levels
- v. English & Maths – Cognitive ability (2016-17)
- This is the cockpit page where buttons have been setup to be navigated into each of the internal reports.

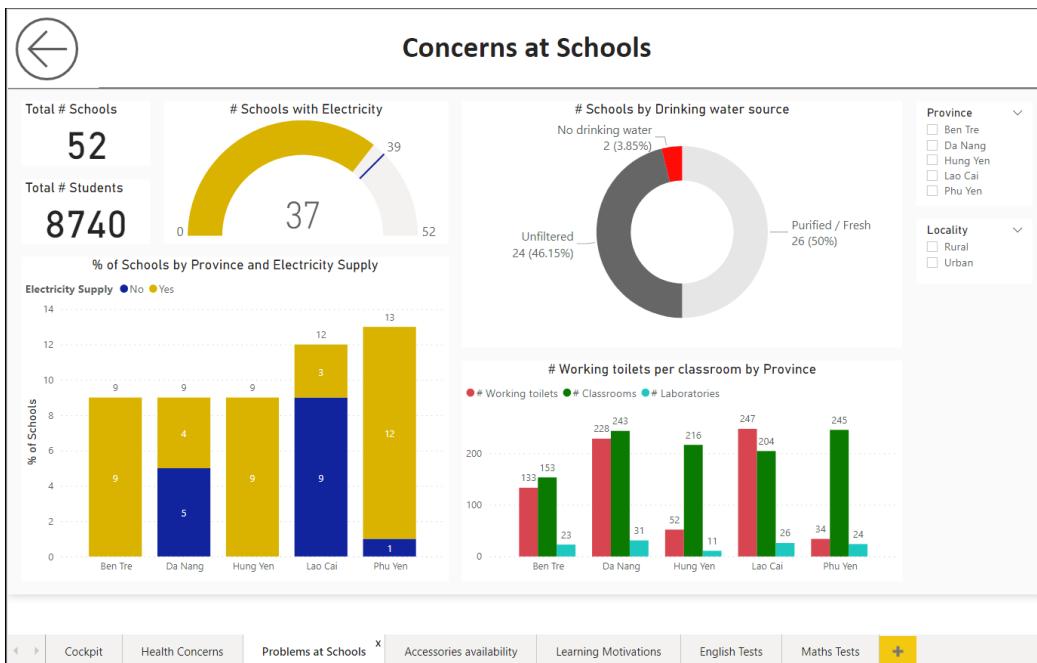


- Student's Health concerns -



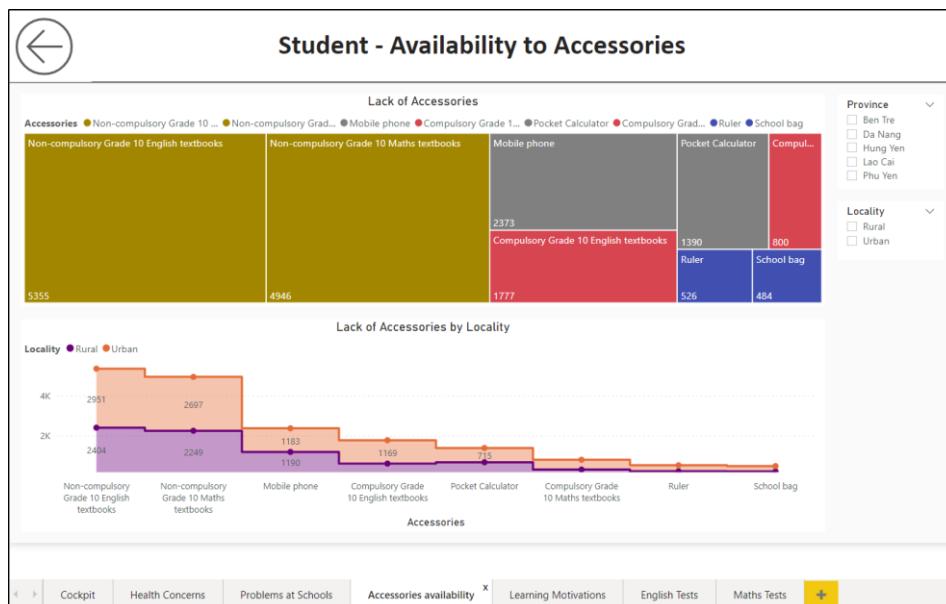
- The female students are facing more challenges than their male counterparts in almost all the health concerns like sight problems, headaches, stomach pains, etc.
- And this is consistent across all the provinces, where females have more health issues.

- **Problems at Schools -**



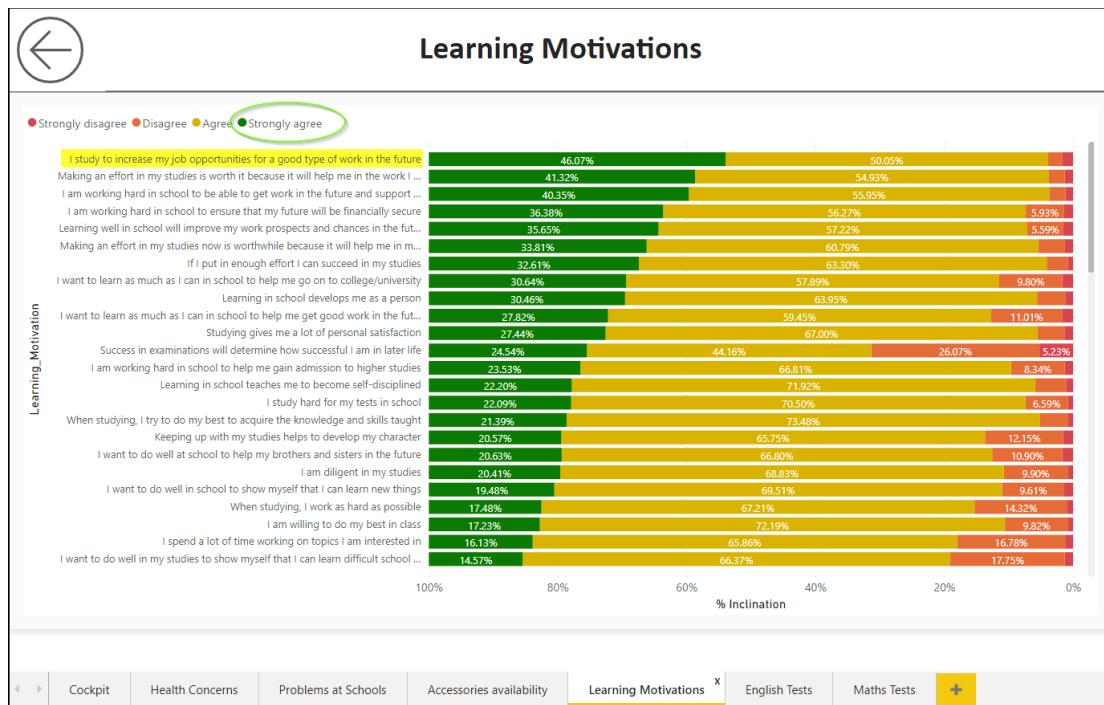
- Among the 52 schools, only 37 of them (less than 75% - 39) are having the electricity access. By province, we could see that Lao Cai, Da Nang primarily are having trouble setting up electricity for more than half of the schools under their radar.
- Number of schools with unfiltered water is 24 and there are 2 schools without even drinking water.
- Phu Yen and Hung Yen have quality problems where there are just above or less than 50 working toilets per an average of 230 classrooms in all the schools together in those provinces. Total number of laboratories in Hung Yen province are just 11.

- **Availability of Accessories -**

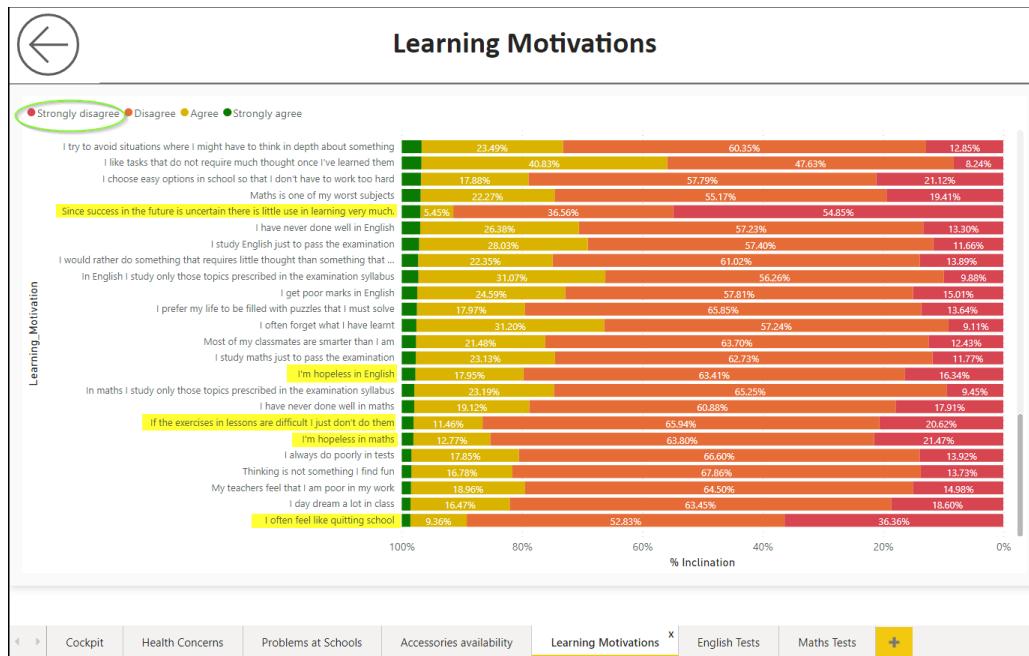


- It seems that 1777 students mainly lack compulsory grade 10 English textbooks and 800 students lack compulsory grade 10 Maths textbooks.
- Also, around 5000 children lack non-compulsory grade 10 English/Maths textbooks.

- Student's Learning Motivations -

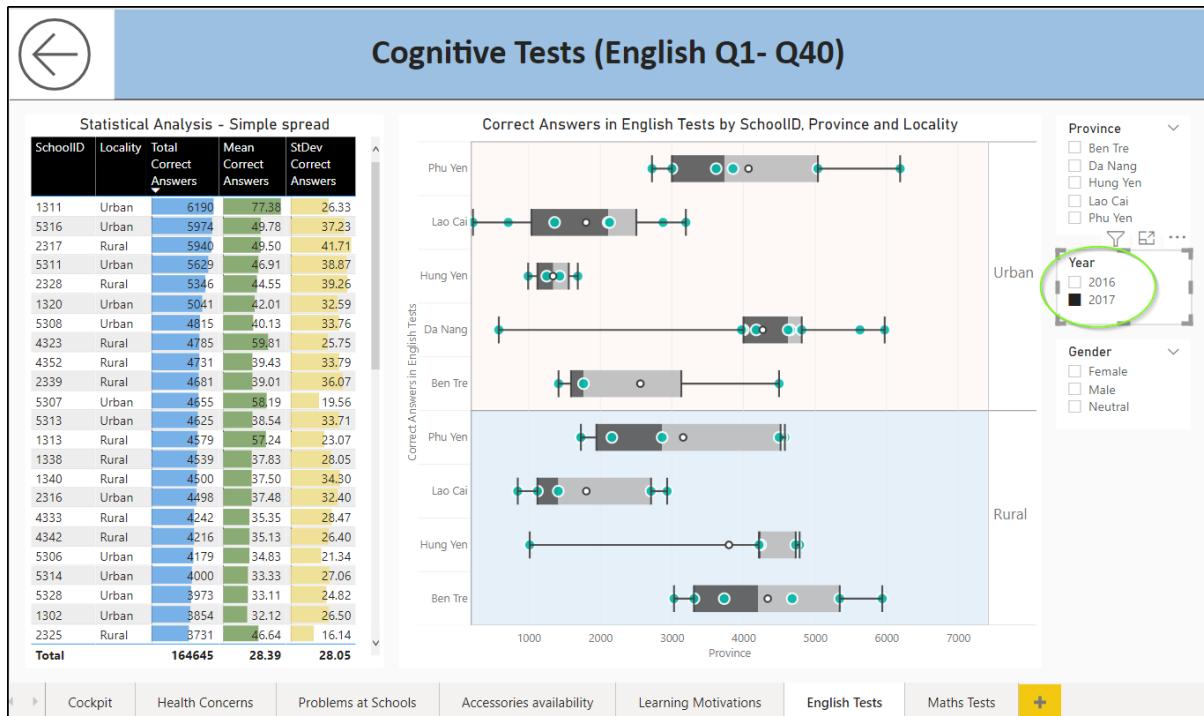
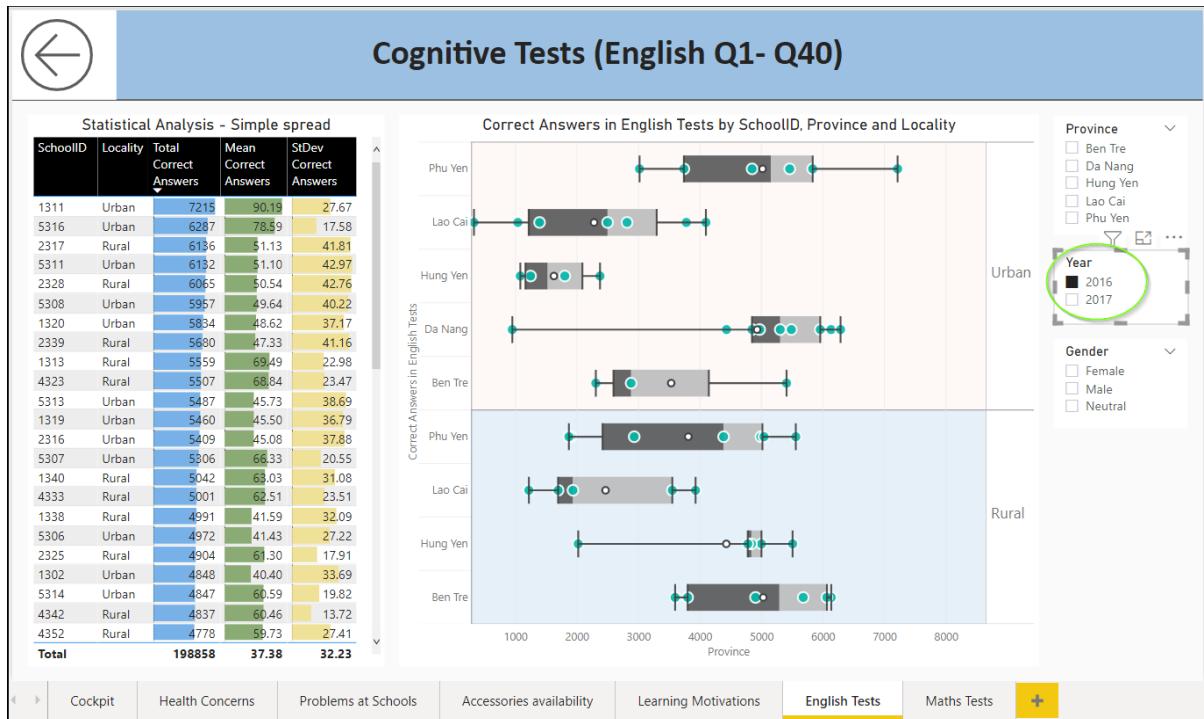


- Children have high enthusiasm and motivation levels towards learning, many of them came forward to agreeing that education improves their life, future and provides better living conditions.



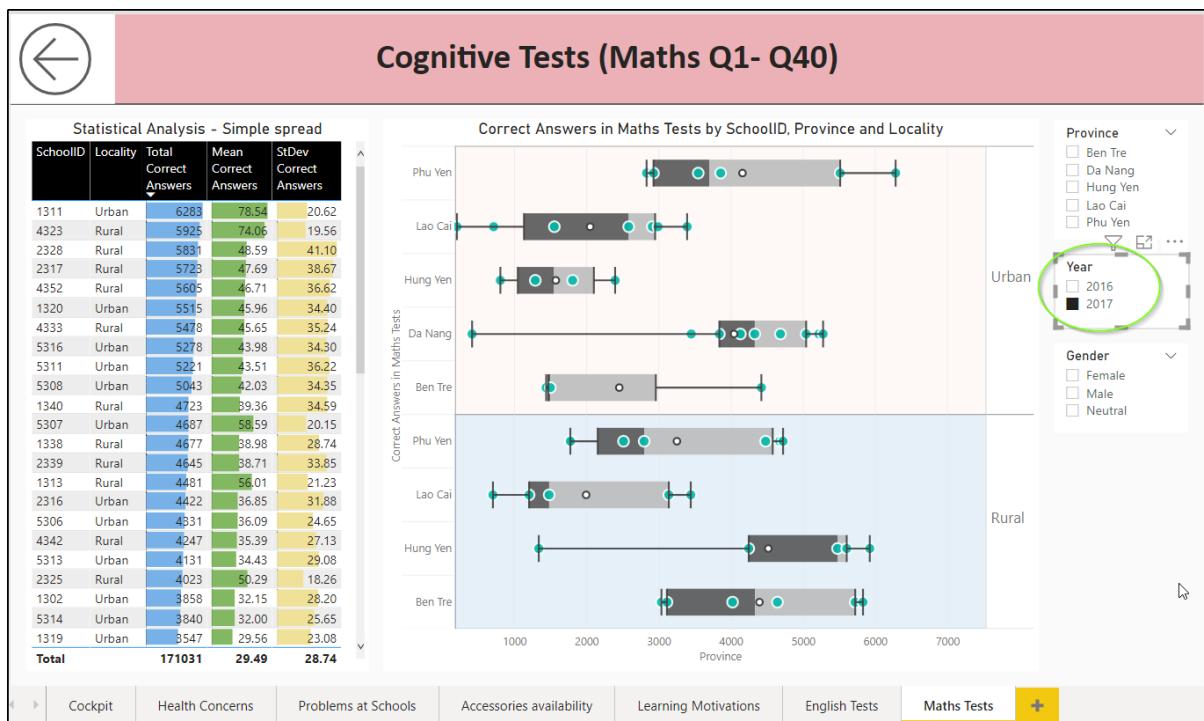
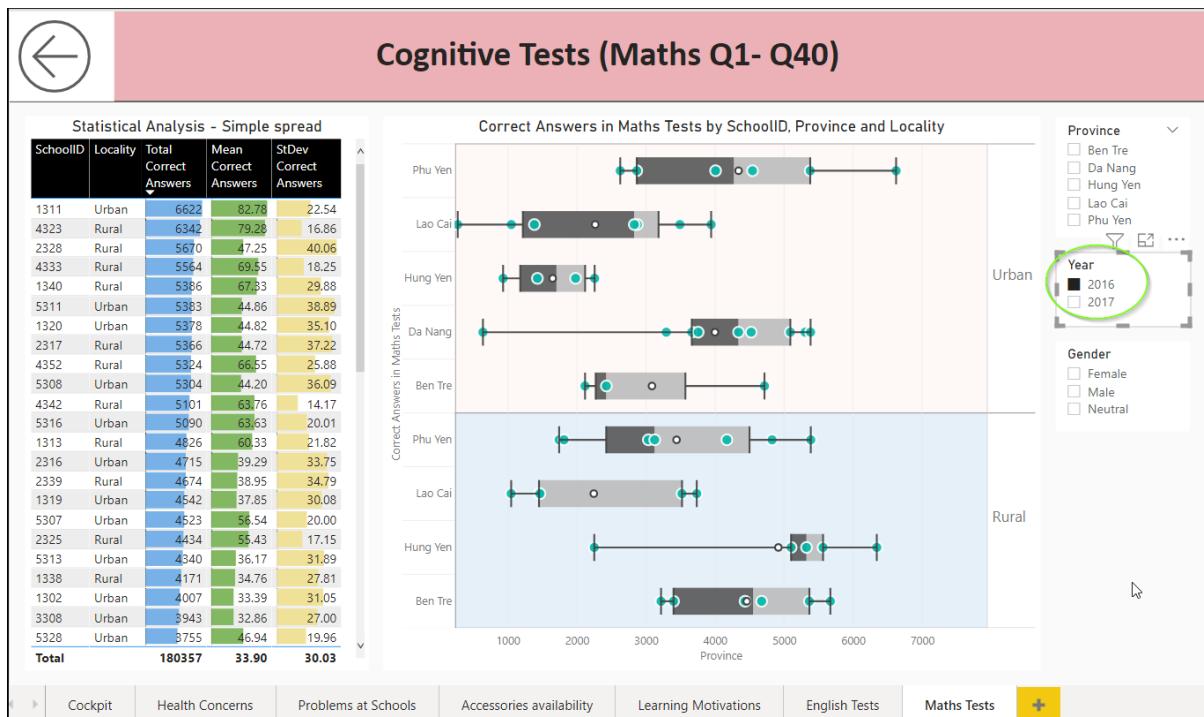
- Also, for the negative questions, they highly disagreed with the questions that they lack in English, hopeless in subjects, or even quitting school.

- Cognitive Tests – English -**



- From 2016 to 2017, it seems that the performance in English tests slightly decreased in almost all the schools. Average correct answers per school in 2016 was 37.38 and in 2017 it dropped to 28.3.

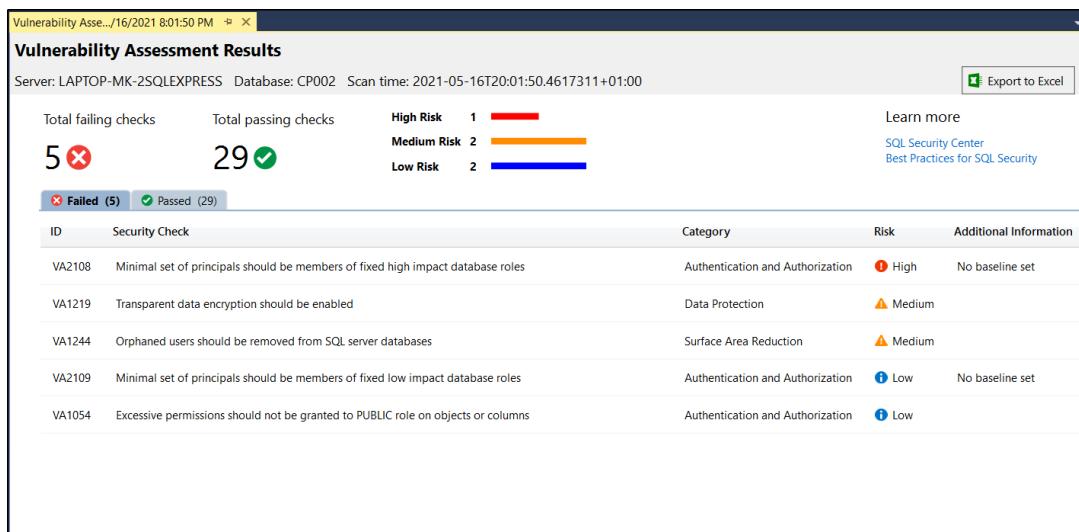
- Cognitive Tests – Maths -**



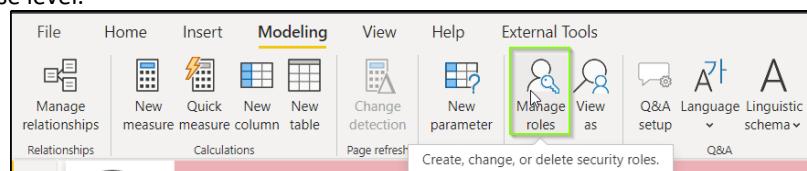
- From 2016 to 2017, it seems that the performance in Maths tests also slightly decreased in almost all the schools, but not to the extent of English. Average correct answers per school in 2016 was 33.90 and in 2017 it dropped to 29.4.

## 2.8. Database Security

- In order to ensure the security in this database, I performed a vulnerability scan/assessment. Below are the results.
- As we could see 5 failing checks here, one advice needed to ensure security and deter ransomware / malware attacks is to rectify these failing checks and to increase the count of passing checks.
- In order to prioritize the rectification process, we could see which failed check is riskier against which category and then we rectify.



- [12] In addition, we could create RLS (Row level security) within Power BI which can segregate users from one another and enable access to only those users who require an access. For example, if this report is being used by analysts who intend to serve their province individually within Vietnam, then the RLS can be setup such that all the analyst users are split by provinces, and Power BI can internally manage roles that are given and each of these users can view only their province-level information, for example – Phu Yen analysts cannot view Ben Tre information because it is restricted.
- In order to enable this feature, we need Power BI Service with a pro/premium license which works at enterprise level.



## 2.9 Database Backup and Restore Strategy

- A periodical backup plan can be setup, which is done in SQL EXPRESS edition manually, whereas using DEVELOPER edition, we can setup a Maintenance plan in which an automatic scheduler can be setup to perform periodical backup of the database.
- For backup, we may choose either Full or Differential, or even overwriting or appending to the original database backup, or perhaps even erase backup.
- We may backup to either Disk or URL. Usually when we store the data in the cloud, we may sync our cloud with SSMS and store the backup over there to ensure high security and availability.
- In order to check the reliability, we can verify the backup when it is finished, also if an error is thrown, we can enable the backup to continue.
- Depending on our needs, we may choose to set backup compression, so that we optimize our disk / datastore memory.
- Backup can be historically saved with a timestamp, and using any of the timestamp, we can restore a database.

## 2.10 Data Science/Business Intelligence Techniques

- In task 2, we use linear regression on student's cognitive abilities, and regression is one of the statistical methods to understand correlation and relationship between 2 variables.
- Report 07 - yREP07\_ST\_Avg\_Test\_Scores is used to create Regression model.
- To analyse if the test scores in 2016 and 2017 are having a correlation between English and Math depending upon the students, schools, provinces, etc.

CK_Test	Province	Locality	SchoolID	Gender	Year	Average Score in English Test	Average Score in Maths Test
2 Ben Tre_Urban_2344_Neutral_2016	Ben Tre	Urban	2344	Neutral	2016	14	0
3 Phu Yen_Rural_1345_Male_2016	Phu Yen	Rural	1345	Male	2016	21.198	15.879
4 Lao Cai_Rural_3326_Neutral_2017	Lao Cai	Rural	3326	Neutral	2017	12.833	8.833
5 Lao Cai_Urban_3305_Male_2016	Lao Cai	Urban	3305	Male	2016	17.414	16.829
6 Hung Yen_Urban_4365_Neutral_2017	Hung Yen	Urban	4365	Neutral	2017	10	15.333
7 Phu Yen_Rural_1338_Male_2017	Phu Yen	Rural	1338	Male	2017	23.937	24.291
8 Phu Yen_Urban_1320_Female_2017	Phu Yen	Urban	1320	Female	2017	27	28.132
9 Ben Tre_Rural_2354_Male_2016	Ben Tre	Rural	2354	Male	2016	19.121	19.516
10 Da Nang_Urban_5307_Male_2017	Da Nang	Urban	5307	Male	2017	24.648	24.8
11 Phu Yen_Rural_1338_Neutral_2016	Phu Yen	Rural	1338	Neutral	2016	23.167	7.5
12 Hung Yen_Rural_4373_Female_2016	Hung Yen	Rural	4373	Female	2016	18.4	17.1
13 Phu Yen_Rural_1336_Female_2017	Phu Yen	Rural	1336	Female	2017	22.025	22.7
14 Lao Cai_Urban_3306_Male_2016	Lao Cai	Urban	3306	Male	2016	18.095	19.959
15 Da Nang_Urban_5313_Neutral_2016	Da Nang	Urban	5313	Neutral	2016	28.5	7.5
16 Phu Yen_Rural_1338_Male_2016	Phu Yen	Rural	1338	Male	2016	26.475	23.238
17 Lao Cai_Urban_3305_Male_2017	Lao Cai	Urban	3305	Male	2017	14.67	15.524
18 Phu Yen_Rural_1345_Male_2017	Phu Yen	Rural	1345	Male	2017	15.515	15.891
19 Phu Yen_Rural_1203_Female_2016	Phu Yen	Rural	1203	Female	2016	22.381	20.286
20 Hung Yen_Urban_4337_Neutral_2017	Hung Yen	Urban	4337	Neutral	2017	13.5	20.5
21 Phu Yen_Urban_1320_Female_2016	Phu Yen	Urban	1320	Female	2016	28.541	26.685
22 Lao Cai_Rural_3240_Male_2017	Lao Cai	Rural	3240	Male	2017	12.127	13.055
23 Phu Yen_Rural_1336_Female_2016	Phu Yen	Rural	1336	Female	2016	22.286	21
24 Da Nang_Urban_5313_Neutral_2017	Da Nang	Urban	5313	Neutral	2017	25	20.75
25 Phu Yen_Rural_1338_Neutral_2017	Phu Yen	Rural	1338	Neutral	2017	18.1	19.2
26 Lao Cai_Urban_3306_Male_2017	Lao Cai	Urban	3306	Male	2017	13.8	17.643
27 Lao Cai_Rural_3315_Neutral_2017	Lao Cai	Rural	3315	Neutral	2017	11	34
28 Lao Cai_Urban_3329_Female_2017	Lao Cai	Urban	3329	Female	2017	12.267	8.533
29 Lao Cai_Rural_3315_Male_2016	Lao Cai	Rural	3315	Male	2016	19.013	19.873

- Initially the correlation is for all the students, without any grouping, where the model failed.
- R squared value and adjusted R-Squared value are very close to 0, that means there is no correlation.
- Also we could observe P-value here which is greater than 0.05, that means this model is statistically insignificant.

A	B	C	D	E	F	G	H	I	J
1	SUMMARY OUTPUT								
2									
3	Regression Statistics								
4	Multiple R	0.45722154							
5	R Square	0.20905							
6	Adjusted R Square	0.20619							
7	Standard Error	5.48696678							
8	Observations	278							
9									
10	ANOVA								
11		df	SS	MS	F	Significance F			
12	Regression	1	2196.235618	2196.23562	72.94814762	9.11364E-16			
13	Residual	276	8309.478039	30.1068045					
14	Total	277	10505.71366						
15									
16		Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
17	Intercept	14.4727048	0.864339366	16.7442388	6.5117E-44	12.77116945	16.1742401	12.7711695	16.1742401
18	X Variable 1	0.36002428	0.042152628	8.54096878	9.11364E-16	0.277042772	0.44300579	0.27704277	0.44300579
19									
20									
21									
22									
23									

yREP07\_ST\_Avg\_Test\_Scores Reg1 - Fail Pivot (SchoolID) Reg2 - Pass Chart +

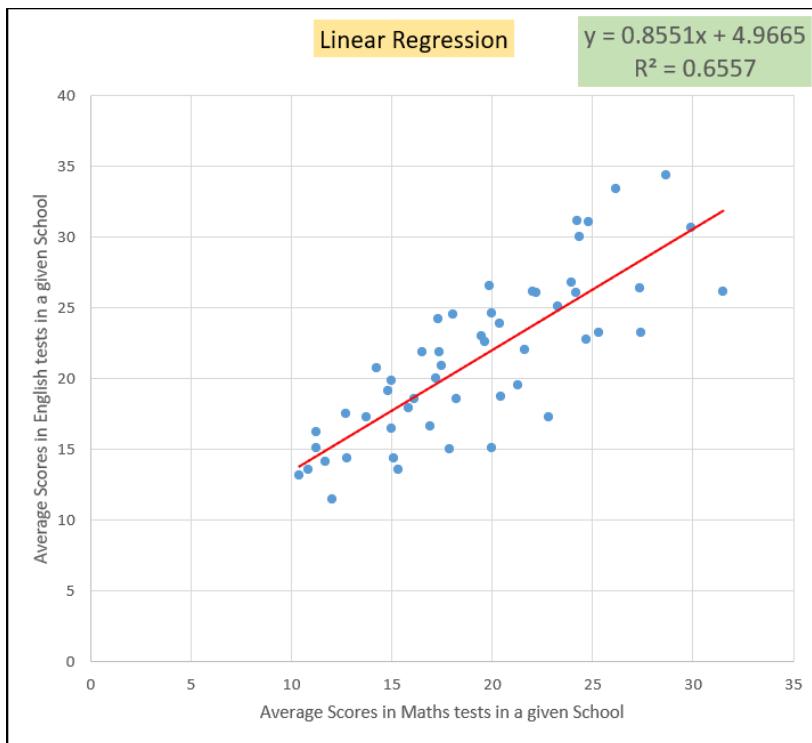
- Later grouping is done on SchoolID and average of Maths & English test scores has been considered for this model.

A	B	C
SchoolID	Average of Score in English Test	Average of Score in Maths Test
1203	18.55833333	16.12783333
1234	20.05883333	17.1785
1302	19.90933333	14.97766667
1311	30.6775	29.913
1313	26.11475	24.182
1319	21.88533333	16.52516667
1320	26.82833333	23.98666667
1321	24.23033333	17.31116667
1336	20.95516667	17.4965
1337	18.7365	20.42825
1338	23.90033333	20.3955
1340	22.7962	24.6776
1345	17.53483333	12.6995
2316	21.86383333	17.3865
2317	31.12883333	24.248
2325	22.01675	21.6305
2328	26.16216667	22.00216667
2329	23.0508	19.478
2339	24.62333333	19.98433333
2343	15.09333333	11.24433333
2344	16.23	11.227
2354	17.3315	13.746
3228	19.5395	21.2815
3240	14.3692	12.7932
3305	16.61083333	16.93633333
3306	17.966	15.8205
3308	18.58766667	18.2125

- Now, the correlation has been moderately consistent.
- R-squared and Adjusted R-squared is relatively closer to 1 and P-value is less than 0.05 which means our model is statistically significant,

A	B	C	D	E	F	G	H	I	
1	SUMMARY OUTPUT								
2									
3	Regression Statistics								
4	Multiple R	0.80974002							
5	R Square	0.65568							
6	Adjusted R Square	0.64879							
7	Standard Error	3.3426691							
8	Observations	52							
9									
10	ANOVA								
11		df	SS	MS	F	Significance F			
12	Regression	1	1063.859705	1063.8597	95.2132928	3.64539E-13			
13	Residual	50	558.6718372	11.1734367					
14	Total	51	1622.531542						
15									
16		Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
17	Intercept	4.96646275	1.748933543	2.83970924	0.006513	1.453626346	8.47929915	1.45362635	8.47929915
18	X Variable 1	0.85505593	0.087628571	9.7577299	3.6454E-13	0.679048764	1.03106309	0.67904876	1.03106309
19									
20									
21									
22									
23									

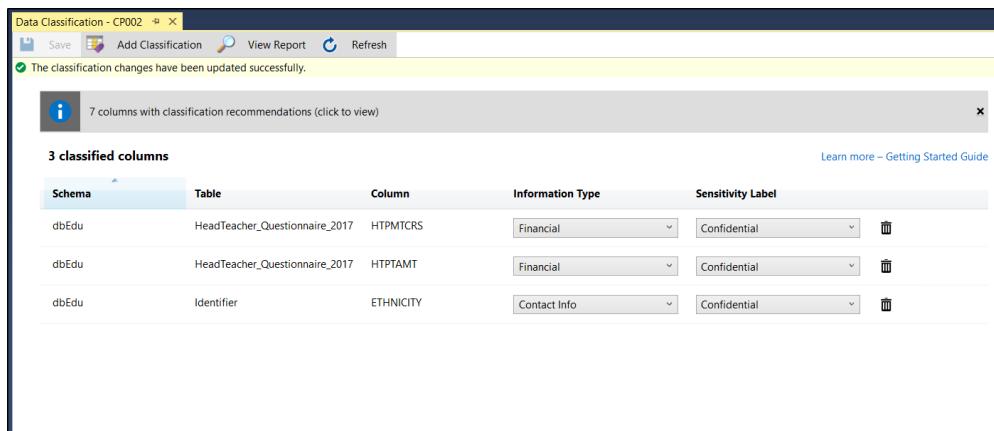
- Now that our model is fairly consistent, we use a scatter plot for Linear regression, as shown below.



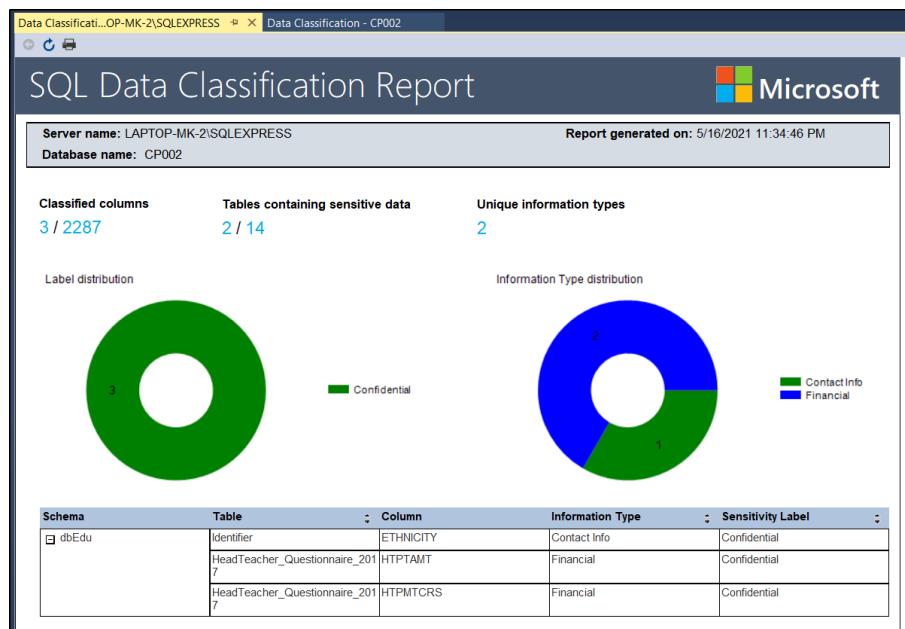
- Conclusion - This means there is a relationship between average test scores from school to school, and based on the school, there is a 65% chance that the student performance can increase or decrease in both the subjects in relation to one another.

## 2.11 Data Privacy, Ethical and legal issues

- [13] SSMS contains features that are internet-enabled, so they can collect and share anonymous feature usage and diagnostic data to Microsoft.
- [14] In order to comply with GDPR (General Data Protection Regulation policy), Microsoft recommends GDPR compliance using 4 steps – Discover, Manage, Protect & Report.
  - In Discover phase, we discover sensitive data – which table, which database, which object has it – everything needs to be documented. Based on the levels of sensitivity of the personal data, these fields need to be classified.
  - In Manage phase, once all the data locations are documented, we proceed to understand which personal data is accessed by whom and then limit it.
  - In Protect phase, once the present security practices are understood, the efforts to protect the data begin. The goal is to reduce risk and reduce the impact to data because of security and monitoring. Various encryption methods can be used in this phase in order to protect the data.
  - In Report phase, 2 principles apply – transparency and data retention of all activities about personal data (metadata only).
- [14] In order to meet the GDPR criteria, an organization must strive to meet the 'Data protection impact assessment' which is prerequisite of the GDPR (**GDPR article 35**).
- [14] From SQL point of view – some of the best recommendations / practices are:
  - Windows authentication over SQL server authentication to enable centralized management via Active directory.
  - To use discrete accounts to authenticate users and applications which minimizes risks of malicious activity which can include SQL injection attacks.
- There are ways of protecting data within SSMS. In order to do this, we must discover the data first and later classify it into levels of confidentiality. **Figure-2.1 shows**



**Figure 2.1 – Data Discovery & Classification**



**Figure 2.2 – Data Classification Report**

## 2.12 Conclusion

- The female students are facing more challenges than their male counterparts in almost all the health concerns like sight problems, headaches, stomach pains, etc, in all the provinces.
- Among the 52 schools, only 37 are having the electricity access. By province, we could see that Lao Cai, Da Nang primarily are having trouble setting up electricity for more than half of the schools under their radar.
- 24 schools operate with unfiltered water and there are 2 schools without even drinking water.
- Phu Yen and Hung Yen have quality problems where there are very less working toilets to classrooms ratio.
- Also the number of laboratories needs to be improved in both Phu Yen and Hung Yen.
- Around 25% of the students lack compulsory grade 10 textbooks for English and Maths.
- Most of the children are highly enthusiastic, and strongly believe that education improves their life, future and provides better living conditions. In addition, they don't want to quit school or they don't tend to get disappointed or depressed with the subjects.
- From 2016 to 2017, the performance in both English and Maths tests have slightly reduced in almost all the schools.

# Task 3 - Database and front-end reporting for Crime in Manchester (2017-18)

## 3.1 Abstract

Manchester is the second largest metropolitan area in the UK outside of London and often is considered the capital of the North. However, crime and notoriety have been prevalent in Manchester from time to time. From violent crimes to petty theft, Manchester city is filled with criminals. Anti-social behavior, burglary, shoplifting, violence and sexual offences, drugs, weaponry are some of the prominent crimes within Manchester. In this task, a database was created to properly populate the crimes data from Jan-2017 to Dec-2018 into the necessary tables - areas, population and crime in streets table where there are latitude and longitude columns as well which can be used to perform geo-spatial analysis. Microsoft Power BI was used for front-end reporting to analyze key performance indicators / metrics like Crime rate per 100K population, Total number of crimes, average number of crimes and crimes outcomes.

## 3.2 Introduction

- A database named 'CR003' is created in which we store all the crimes data in Manchester from 2017 to 2018 along with Population in the United Kingdom.
- Mode of raw source data (crimes) upload – Bulk insert using T-SQL.
- Mode of raw source data (population) upload - Direct import using 'SSMS Import and Export wizard'
- 6 population tables were imported – 2017 males, 2017 females, 2017 persons (sum of 2017 males and females), 2018 males, 2018 females, 2018 persons (sum of 2018 males and females).

Option	Input Value
Source format of the files	Flat file – Comma separated file
Header row delimiter	Comma {,}
Destination	Microsoft OLE DB Provider for SQL Server
Server name	LAPTOP-MK-2\SQLEXPRESS
Authentication	Windows / SQL server authentication
Database	CR003
Schema	dbo
Total number of files	6
Names of the raw tables	[dbo].[Mid-2017 Females] [dbo].[Mid-2017 Males] [dbo].[Mid-2017 Persons] [dbo].[Mid-2018 Females] [dbo].[Mid-2018 Males] [dbo].[Mid-2018 Persons]

## 3.3 Relational Schema

### 3.3.1 Identify the tables

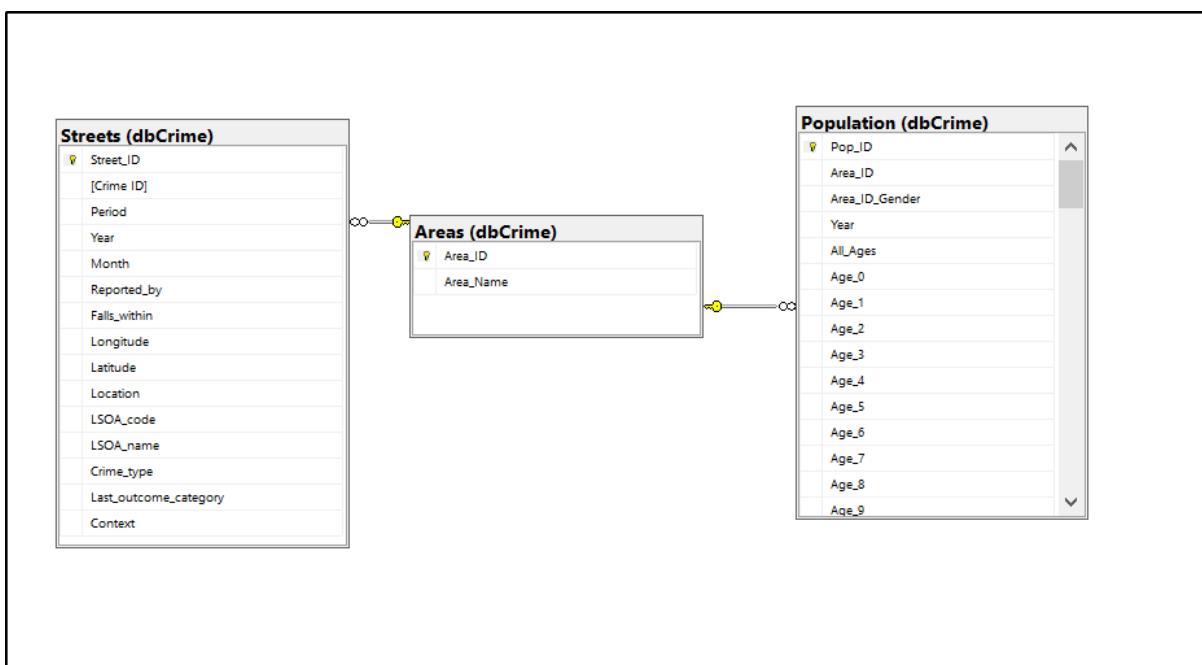
- From the population tables we could see both the counties as well as the areas within the counties against the key value and their sums were also seen for each county. So in order to make sure that we have a normalized model here, it is imperative to separate counties and areas under each of these counties.
- For this reason, a table for Areas is created which has all the areas against the AreaID which is simply LSOA code. Area ID will be the primary key for this model. And Crimes and Population data tables will refer to this central table as a reference table.
- A schema has been defined = 'dbCrime'

### 3.3.2 Create the tables

- Before creating the tables, the maximum amount of row-length was checked for each of the fields given, in order to determine which datatype would be suitable, also a thorough check was performed before a datatype is finalized for a field.
- The following tables are created using T-SQL:
  - [dbCrime].[Areas]
  - [dbCrime].[Population]
  - [dbCrime].[Streets]

### 3.3.3 Create relationships between tables

- Streets table and Population table have individual primary keys as Street\_ID and Pop\_ID respectively, however, each of these tables have a column called either LSOA\_code or Area\_ID which will be mapped to Areas table using a foreign key reference.
- Area\_ID forms the primary key in the Areas table and this table doesn't require any foreign key constraints.
- Foreign key constraints are defined in Streets and Population tables to ensure data accuracy.



## 3.4 Design Rationale

- In order to build a comprehensive relational model here, 2 sets of raw data table extracts are considered and these are spread across 3 final tables.
- Both Streets and Population tables have 2017 and 2018 data within, and Areas table doesn't require a timeline because this is basically a never / slowly changing dimension. Hence Areas table is separately created.
- Areas table is created from the populations table that has the highest number of distinct area names, and then separated.
- Population table need not have the area name anymore, so we can simply create a new table with Pop\_ID as population table and insert the data from combined population table.
- The main challenge here was to segregate persons, male and female population. In order to tackle this while inserting the data, a new column along with the Area code has been introduced such that we could see '`<abc..12>_M`' for male population area code, similarly `_F` for females and `_A` for all population.

- Initially, an inline table called zzfilelist has been created which has the information of the filepaths. In total we have 805855 crimes spread across the 24 months from Jan-2017 to Dec-2018, which means 24 files.
- In order to build a scalable model, a bulk insert script has been written which will use the contents from the zzfilelist table that we created using the files and filepath from our windows-based PC, and later extract everything serially. Here we may even apply Error Handling technique so that if the number of columns or the column headers vary, our script can simply throw an exception and skip the file for that moment, instead of cancelling the batch job.
- All the views that were created are basically using these tables present, simply by logical joins.
- All the reports are basically views, however, they have an underlying subquery / sub queries that could have joined, unionized or even pivoted/unpivoted a combination of tables or views or even both sometimes as part of the blend. Each of these reports have aggregations based on the preliminary exploratory analysis using the raw table.
- A schema 'dbCrime' is also defined here to show that this logical collection of database objects is segregated, if needed for multiple applications, accessibility, managing database security, etc. In our case, a schema owner can use only the database tables as the schema is defined for tables alone.
- In terms of performance, this model is highly scalable as it has a simple yet comprehensive and powerful structure.
- Triggers are given to ensure to prevent inserting, updating or deleting the records from all the tables.
- For front-end reporting, Power BI was used and 2 reports were extracted.
- Since QGIS was problematic during the installation in my PC, I used PowerBI itself for geo-spatial analysis.
- The security has been handled by giving necessary access only to a test\_user.

## 3.5 Design Considerations

### 3.5.1 Database Normalisation

- This is a fully normalized 3NF normal form structure and the schema is very scalable.
- This logical model is highly optimized, and the normalization technique used here will vastly reduce data redundancy and improve data integrity.

### 3.5.2 Constraints

- Constraints are created during transaction data tables creation. All the constraints shown below are referencing only the [dbCrime].[Areas] table which has the primary key – Area\_ID.

Constraint	Present in table	Referencing table(key)
FK_Area_ID	[dbCrime].[Population]	[dbCrime].[Areas](Area_ID)
FK_LSOA_ID	[dbCrime].[Streets]	[dbCrime].[Areas](Area_ID)

### 3.5.3 Data Validation

- There are totally 1695 areas in Manchester and 805855 number of crimes recorded between 2017-18.

```

/*Validation check - 805055 rows in total - Validation complete*/
SELECT COUNT([Period]) FROM dbo.Streets;
/*Validation check - 1695 unique LSOA codes - Validation complete*/
SELECT COUNT( distinct [LSOA_code]) FROM [CR003].[dbCrime].[Population]

```

### 3.5.4 Transaction and Concurrency Control (if any)

- N/A

### 3.5.5 Error Handling

- During the bulk insert, we might face an issue in the future in case this model is being scaled up. Hence, I gave a try and catch exception block to identify the errors and show them to the batch job executioner.

### 3.5.6 Security

- Security-based privileges were given to a test user in order to either provide the privileges or ensure anonymity of the underlying data, in this case, the [reported by] and [falls\_within] columns are anonymised because I wouldn't want to disclose this information to a user.

```
USE CR003;
GO

/*To grant select privilege to cr_user1 - Population*/
GRANT SELECT ON [dbCrime].[Population] TO cr_user1;
GO

/*To deny select privilege to cr_user1 - who reported the issue, and falls_within */
DENY SELECT ON [dbCrime].[Streets] ([Reported by], [Falls_within]) TO cr_user1;
GO
```

### 3.5.7 Comments

- Population table consists of the entire UK population, this can be reduced by using an IF EXISTS by subquerying the records where only Manchester is present. However, this created a performance issue in my PC, hence I decided not to consider this option.
- Necessary comments were added everywhere in all the queries to enable layman readability.

### 3.6 T-SQL Statements

- Create the database – CR003

```
USE master;
GO

CREATE DATABASE CR003;
```

- Perform Data cleaning for the raw Population tables before creating any tables.
- Utilizing the system stored procedure – sp\_rename to rename a table.

```
USE CR003;
GO

/*Tables uploaded for Population have inconsistent names like 'Mid-2017 Females$' for example/
/*We are going to rename them using system stored procedure -
sp_rename*/

EXEC sp_rename 'Mid-2017 Females$', 'Mid-2017 Females';
EXEC sp_rename 'Mid-2017 Males$', 'Mid-2017 Males';
EXEC sp_rename 'Mid-2017 Persons$', 'Mid-2017 Persons';
EXEC sp_rename 'Mid-2018 Females$', 'Mid-2018 Females';
EXEC sp_rename 'Mid-2018 Males$', 'Mid-2018 Males';
EXEC sp_rename 'Mid-2018 Persons$', 'Mid-2018 Persons';

/*Delete Top 4 rows in all the 6 tables extracted, we don't need the
garbage values/

DELETE TOP (4)
    FROM [dbo].[Mid-2017 Females];

DELETE TOP (4)
    FROM [dbo].[Mid-2017 Males];

DELETE TOP (4)
    FROM [dbo].[Mid-2017 Persons];

DELETE TOP (4)
    FROM [dbo].[Mid-2018 Females];

DELETE TOP (4)
    FROM [dbo].[Mid-2018 Males];

DELETE TOP (4)
    FROM [dbo].[Mid-2018 Persons];
/* End of Data cleaning activity for Task 3*/
```

### 3.6.1 Tables

- Create Areas table

```

CREATE TABLE Areas
(
    Area_ID      VARCHAR(10)          PRIMARY KEY,
    Area_Name    VARCHAR(150)         NOT NULL,
)
;
--TRUNCATE TABLE CR003.[dbo].[Areas];
/*Pick any table to insert Area names into the Areas table*/
INSERT INTO CR003.[dbo].[Areas]
SELECT
    LTRIM(RTRIM([Contents])) AS Area_ID,
    LTRIM(RTRIM([F3]))      AS Area_Name
FROM
    [CR003].[dbo].[Mid-2017 Females]
WHERE LTRIM(RTRIM([F3])) <> '' OR [F3] IS NOT NULL
AND LOWER([Contents]) <> 'area codes';

```

- Create table - Population

```

USE CR003; GO

CREATE TABLE [Population] ( Pop_ID INT IDENTITY(1,1) PRIMARY KEY, Area_ID VARCHAR(10) NOT NULL,
Area_ID_Gender VARCHAR(20), [Year] INT

,All_Ages INT ,Age_0 INT ,Age_1 INT ,Age_2 INT ,Age_3 INT ,Age_4 INT ,Age_5 INT ,Age_6 INT ,Age_7 INT ,Age_8 INT ,Age_9 INT
,Age_10 INT ,Age_11 INT ,Age_12 INT ,Age_13 INT ,Age_14 INT ,Age_15 INT ,Age_16 INT ,Age_17 INT ,Age_18 INT ,Age_19 INT ,Age_20
INT ,Age_21 INT ,Age_22 INT ,Age_23 INT ,Age_24 INT ,Age_25 INT ,Age_26 INT ,Age_27 INT ,Age_28 INT ,Age_29 INT ,Age_30 INT
,Age_31 INT ,Age_32 INT ,Age_33 INT ,Age_34 INT ,Age_35 INT ,Age_36 INT ,Age_37 INT ,Age_38 INT ,Age_39 INT ,Age_40 INT ,Age_41
INT ,Age_42 INT ,Age_43 INT ,Age_44 INT ,Age_45 INT ,Age_46 INT ,Age_47 INT ,Age_48 INT ,Age_49 INT ,Age_50 INT ,Age_51 INT
,Age_52 INT ,Age_53 INT ,Age_54 INT ,Age_55 INT ,Age_56 INT ,Age_57 INT ,Age_58 INT ,Age_59 INT ,Age_60 INT ,Age_61 INT ,Age_62
INT ,Age_63 INT ,Age_64 INT ,Age_65 INT ,Age_66 INT ,Age_67 INT ,Age_68 INT ,Age_69 INT ,Age_70 INT ,Age_71 INT ,Age_72 INT
,Age_73 INT ,Age_74 INT ,Age_75 INT ,Age_76 INT ,Age_77 INT ,Age_78 INT ,Age_79 INT ,Age_80 INT ,Age_81 INT ,Age_82 INT ,Age_83
INT ,Age_84 INT ,Age_85 INT ,Age_86 INT ,Age_87 INT ,Age_88 INT ,Age_89 INT ,Age_90_and_above INT

,CONSTRAINT FK_Area_ID FOREIGN KEY (Area_ID) REFERENCES [dbo].[Areas](Area_ID) );

```

- Insert into Population table – from raw tables 2017 Females, 2017 Males, 2017 Persons

```

/*Insert 2017 Population tables gender-wise & year-wise */

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) AS Area_ID, LTRIM(RTRIM([Contents]))+'_F' AS Area_ID_Gender, '2017' AS
'Year', [F4] AS All_Ages, [F5] AS Age_0, [F6] AS Age_1, [F7] AS Age_2, [F8] AS Age_3, [F9] AS Age_4, [F10] AS Age_5, [F11] AS Age_6, [F12] AS
Age_7, [F13] AS Age_8, [F14] AS Age_9, [F15] AS Age_10, [F16] AS Age_11, [F17] AS Age_12, [F18] AS Age_13, [F19] AS Age_14, [F20] AS Age_15, [F21]
AS Age_16, [F22] AS Age_17, [F23] AS Age_18, [F24] AS Age_19, [F25] AS Age_20, [F26] AS Age_21, [F27] AS Age_22, [F28] AS Age_23, [F29] AS Age_24
,[F30] AS Age_25, [F31] AS Age_26, [F32] AS Age_27, [F33] AS Age_28, [F34] AS Age_29, [F35] AS Age_30, [F36] AS Age_31, [F37] AS Age_32, [F38] AS
Age_33, [F39] AS Age_34, [F40] AS Age_35, [F41] AS Age_36, [F42] AS Age_37, [F43] AS Age_38, [F44] AS Age_39, [F45] AS Age_40, [F46] AS Age_41
,[F47] AS Age_42, [F48] AS Age_43, [F49] AS Age_44, [F50] AS Age_45, [F51] AS Age_46, [F52] AS Age_47, [F53] AS Age_48, [F54] AS Age_49, [F55] AS
Age_50, [F56] AS Age_51, [F57] AS Age_52, [F58] AS Age_53, [F59] AS Age_54, [F60] AS Age_55, [F61] AS Age_56, [F62] AS Age_57, [F63] AS Age_58
,[F64] AS Age_59, [F65] AS Age_60, [F66] AS Age_61, [F67] AS Age_62, [F68] AS Age_63, [F69] AS Age_64, [F70] AS Age_65, [F71] AS Age_66, [F72] AS
Age_67, [F73] AS Age_68, [F74] AS Age_69, [F75] AS Age_70, [F76] AS Age_71, [F77] AS Age_72, [F78] AS Age_73, [F79] AS Age_74, [F80] AS Age_75
,[F81] AS Age_76, [F82] AS Age_77, [F83] AS Age_78, [F84] AS Age_79, [F85] AS Age_80, [F86] AS Age_81, [F87] AS Age_82, [F88] AS Age_83, [F89] AS
Age_84, [F90] AS Age_85, [F91] AS Age_86, [F92] AS Age_87, [F93] AS Age_88, [F94] AS Age_89, [F95] AS Age_90_and_above FROM [CR003].[dbo].[Mid-
2017 Females] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) AS Area_ID, LTRIM(RTRIM([Contents]))+'_M' AS Area_ID_Gender, '2017' AS
'Year'

,[F4] AS All_Ages, [F5] AS Age_0, [F6] AS Age_1, [F7] AS Age_2, [F8] AS Age_3, [F9] AS Age_4, [F10] AS Age_5, [F11] AS Age_6, [F12] AS Age_7
,[F13] AS Age_8, [F14] AS Age_9, [F15] AS Age_10, [F16] AS Age_11, [F17] AS Age_12, [F18] AS Age_13, [F19] AS Age_14, [F20] AS Age_15, [F21] AS
Age_16, [F22] AS Age_17, [F23] AS Age_18, [F24] AS Age_19, [F25] AS Age_20, [F26] AS Age_21, [F27] AS Age_22, [F28] AS Age_23, [F29] AS Age_24
,[F30] AS Age_25, [F31] AS Age_26, [F32] AS Age_27, [F33] AS Age_28, [F34] AS Age_29, [F35] AS Age_30, [F36] AS Age_31, [F37] AS Age_32, [F38] AS
Age_33, [F39] AS Age_34, [F40] AS Age_35, [F41] AS Age_36, [F42] AS Age_37, [F43] AS Age_38, [F44] AS Age_39, [F45] AS Age_40, [F46] AS Age_41
,[F47] AS Age_42, [F48] AS Age_43, [F49] AS Age_44, [F50] AS Age_45, [F51] AS Age_46, [F52] AS Age_47, [F53] AS Age_48, [F54] AS Age_49, [F55] AS
Age_50, [F56] AS Age_51, [F57] AS Age_52, [F58] AS Age_53, [F59] AS Age_54, [F60] AS Age_55, [F61] AS Age_56, [F62] AS Age_57, [F63] AS Age_58
,[F64] AS Age_59, [F65] AS Age_60, [F66] AS Age_61, [F67] AS Age_62, [F68] AS Age_63, [F69] AS Age_64, [F70] AS Age_65, [F71] AS Age_66, [F72] AS
Age_67, [F73] AS Age_68, [F74] AS Age_69, [F75] AS Age_70, [F76] AS Age_71, [F77] AS Age_72, [F78] AS Age_73, [F79] AS Age_74, [F80] AS Age_75
,[F81] AS Age_76, [F82] AS Age_77, [F83] AS Age_78, [F84] AS Age_79, [F85] AS Age_80, [F86] AS Age_81, [F87] AS Age_82, [F88] AS Age_83, [F89] AS
Age_84, [F90] AS Age_85, [F91] AS Age_86, [F92] AS Age_87, [F93] AS Age_88, [F94] AS Age_89, [F95] AS Age_90_and_above

FROM [CR003].[dbo].[Mid-2017 Males] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) AS Area_ID, LTRIM(RTRIM([Contents]))+'_A' AS Area_ID_Gender, '2017' AS
'Year'

,[F4] AS All_Ages, [F5] AS Age_0, [F6] AS Age_1, [F7] AS Age_2, [F8] AS Age_3, [F9] AS Age_4, [F10] AS Age_5, [F11] AS Age_6, [F12] AS Age_7
,[F13] AS Age_8, [F14] AS Age_9, [F15] AS Age_10, [F16] AS Age_11, [F17] AS Age_12, [F18] AS Age_13, [F19] AS Age_14, [F20] AS Age_15, [F21] AS
Age_16, [F22] AS Age_17, [F23] AS Age_18, [F24] AS Age_19, [F25] AS Age_20, [F26] AS Age_21, [F27] AS Age_22, [F28] AS Age_23, [F29] AS Age_24
,[F30] AS Age_25, [F31] AS Age_26, [F32] AS Age_27, [F33] AS Age_28, [F34] AS Age_29, [F35] AS Age_30, [F36] AS Age_31, [F37] AS Age_32, [F38] AS
Age_33, [F39] AS Age_34, [F40] AS Age_35, [F41] AS Age_36, [F42] AS Age_37, [F43] AS Age_38, [F44] AS Age_39, [F45] AS Age_40, [F46] AS Age_41
,[F47] AS Age_42, [F48] AS Age_43, [F49] AS Age_44, [F50] AS Age_45, [F51] AS Age_46, [F52] AS Age_47, [F53] AS Age_48, [F54] AS Age_49, [F55] AS
Age_50, [F56] AS Age_51, [F57] AS Age_52, [F58] AS Age_53, [F59] AS Age_54, [F60] AS Age_55, [F61] AS Age_56, [F62] AS Age_57, [F63] AS Age_58
,[F64] AS Age_59, [F65] AS Age_60, [F66] AS Age_61, [F67] AS Age_62, [F68] AS Age_63, [F69] AS Age_64, [F70] AS Age_65, [F71] AS Age_66, [F72] AS
Age_67, [F73] AS Age_68, [F74] AS Age_69, [F75] AS Age_70, [F76] AS Age_71, [F77] AS Age_72, [F78] AS Age_73, [F79] AS Age_74, [F80] AS Age_75
,[F81] AS Age_76, [F82] AS Age_77, [F83] AS Age_78, [F84] AS Age_79, [F85] AS Age_80, [F86] AS Age_81, [F87] AS Age_82, [F88] AS Age_83, [F89] AS
Age_84, [F90] AS Age_85, [F91] AS Age_86, [F92] AS Age_87, [F93] AS Age_88, [F94] AS Age_89, [F95] AS Age_90_and_above

FROM [CR003].[dbo].[Mid-2017 Persons] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

```

- Insert into Population table – from raw tables 2018 Females, 2018 Males, 2018 Persons

```

/*Insert 2018 Population tables gender-wise & year-wise */

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) as Area_ID, LTRIM(RTRIM([Contents]))+'_'+F' as Area_ID_Gender, '2018' as 'Year'

,[F4] as All_Ages ,[F5] as Age_0 ,[F6] as Age_1 ,[F7] as Age_2 ,[F8] as Age_3 ,[F9] as Age_4 ,[F10] as Age_5 ,[F11] as Age_6 ,[F12] as Age_7
,[F13] as Age_8 ,[F14] as Age_9 ,[F15] as Age_10 ,[F16] as Age_11 ,[F17] as Age_12 ,[F18] as Age_13 ,[F19] as Age_14 ,[F20] as Age_15 ,[F21] as
Age_16 ,[F22] as Age_17 ,[F23] as Age_18 ,[F24] as Age_19 ,[F25] as Age_20 ,[F26] as Age_21 ,[F27] as Age_22 ,[F28] as Age_23 ,[F29] as Age_24
,[F30] as Age_25 ,[F31] as Age_26 ,[F32] as Age_27 ,[F33] as Age_28 ,[F34] as Age_29 ,[F35] as Age_30 ,[F36] as Age_31 ,[F37] as Age_32 ,[F38] as
Age_33 ,[F39] as Age_34 ,[F40] as Age_35 ,[F41] as Age_36 ,[F42] as Age_37 ,[F43] as Age_38 ,[F44] as Age_39 ,[F45] as Age_40 ,[F46] as Age_41
,[F47] as Age_42 ,[F48] as Age_43 ,[F49] as Age_44 ,[F50] as Age_45 ,[F51] as Age_46 ,[F52] as Age_47 ,[F53] as Age_48 ,[F54] as Age_49 ,[F55] as
Age_50 ,[F56] as Age_51 ,[F57] as Age_52 ,[F58] as Age_53 ,[F59] as Age_54 ,[F60] as Age_55 ,[F61] as Age_56 ,[F62] as Age_57 ,[F63] as Age_58
,[F64] as Age_59 ,[F65] as Age_60 ,[F66] as Age_61 ,[F67] as Age_62 ,[F68] as Age_63 ,[F69] as Age_64 ,[F70] as Age_65 ,[F71] as Age_66 ,[F72] as
Age_67 ,[F73] as Age_68 ,[F74] as Age_69 ,[F75] as Age_70 ,[F76] as Age_71 ,[F77] as Age_72 ,[F78] as Age_73 ,[F79] as Age_74 ,[F80] as Age_75
,[F81] as Age_76 ,[F82] as Age_77 ,[F83] as Age_78 ,[F84] as Age_79 ,[F85] as Age_80 ,[F86] as Age_81 ,[F87] as Age_82 ,[F88] as Age_83 ,[F89] as
Age_84 ,[F90] as Age_85 ,[F91] as Age_86 ,[F92] as Age_87 ,[F93] as Age_88 ,[F94] as Age_89 ,[F95] as Age_90_and_above

FROM [CR003].[dbo].[Mid-2018 Females] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) as Area_ID, LTRIM(RTRIM([Contents]))+'_'+M' as Area_ID_Gender, '2018' as 'Year'

,[F4] as All_Ages ,[F5] as Age_0 ,[F6] as Age_1 ,[F7] as Age_2 ,[F8] as Age_3 ,[F9] as Age_4 ,[F10] as Age_5 ,[F11] as Age_6 ,[F12] as Age_7
,[F13] as Age_8 ,[F14] as Age_9 ,[F15] as Age_10 ,[F16] as Age_11 ,[F17] as Age_12 ,[F18] as Age_13 ,[F19] as Age_14 ,[F20] as Age_15 ,[F21] as
Age_16 ,[F22] as Age_17 ,[F23] as Age_18 ,[F24] as Age_19 ,[F25] as Age_20 ,[F26] as Age_21 ,[F27] as Age_22 ,[F28] as Age_23 ,[F29] as Age_24
,[F30] as Age_25 ,[F31] as Age_26 ,[F32] as Age_27 ,[F33] as Age_28 ,[F34] as Age_29 ,[F35] as Age_30 ,[F36] as Age_31 ,[F37] as Age_32 ,[F38] as
Age_33 ,[F39] as Age_34 ,[F40] as Age_35 ,[F41] as Age_36 ,[F42] as Age_37 ,[F43] as Age_38 ,[F44] as Age_39 ,[F45] as Age_40 ,[F46] as Age_41
,[F47] as Age_42 ,[F48] as Age_43 ,[F49] as Age_44 ,[F50] as Age_45 ,[F51] as Age_46 ,[F52] as Age_47 ,[F53] as Age_48 ,[F54] as Age_49 ,[F55] as
Age_50 ,[F56] as Age_51 ,[F57] as Age_52 ,[F58] as Age_53 ,[F59] as Age_54 ,[F60] as Age_55 ,[F61] as Age_56 ,[F62] as Age_57 ,[F63] as Age_58
,[F64] as Age_59 ,[F65] as Age_60 ,[F66] as Age_61 ,[F67] as Age_62 ,[F68] as Age_63 ,[F69] as Age_64 ,[F70] as Age_65 ,[F71] as Age_66 ,[F72] as
Age_67 ,[F73] as Age_68 ,[F74] as Age_69 ,[F75] as Age_70 ,[F76] as Age_71 ,[F77] as Age_72 ,[F78] as Age_73 ,[F79] as Age_74 ,[F80] as Age_75
,[F81] as Age_76 ,[F82] as Age_77 ,[F83] as Age_78 ,[F84] as Age_79 ,[F85] as Age_80 ,[F86] as Age_81 ,[F87] as Age_82 ,[F88] as Age_83 ,[F89] as
Age_84 ,[F90] as Age_85 ,[F91] as Age_86 ,[F92] as Age_87 ,[F93] as Age_88 ,[F94] as Age_89 ,[F95] as Age_90_and_above

FROM [CR003].[dbo].[Mid-2018 Males] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

INSERT INTO CR003.[dbo].[Population] SELECT LTRIM(RTRIM([Contents])) as Area_ID, LTRIM(RTRIM([Contents]))+'_'+A' as Area_ID_Gender, '2018' as 'Year'

,[F4] as All_Ages ,[F5] as Age_0 ,[F6] as Age_1 ,[F7] as Age_2 ,[F8] as Age_3 ,[F9] as Age_4 ,[F10] as Age_5 ,[F11] as Age_6 ,[F12] as Age_7
,[F13] as Age_8 ,[F14] as Age_9 ,[F15] as Age_10 ,[F16] as Age_11 ,[F17] as Age_12 ,[F18] as Age_13 ,[F19] as Age_14 ,[F20] as Age_15 ,[F21] as
Age_16 ,[F22] as Age_17 ,[F23] as Age_18 ,[F24] as Age_19 ,[F25] as Age_20 ,[F26] as Age_21 ,[F27] as Age_22 ,[F28] as Age_23 ,[F29] as Age_24
,[F30] as Age_25 ,[F31] as Age_26 ,[F32] as Age_27 ,[F33] as Age_28 ,[F34] as Age_29 ,[F35] as Age_30 ,[F36] as Age_31 ,[F37] as Age_32 ,[F38] as
Age_33 ,[F39] as Age_34 ,[F40] as Age_35 ,[F41] as Age_36 ,[F42] as Age_37 ,[F43] as Age_38 ,[F44] as Age_39 ,[F45] as Age_40 ,[F46] as Age_41
,[F47] as Age_42 ,[F48] as Age_43 ,[F49] as Age_44 ,[F50] as Age_45 ,[F51] as Age_46 ,[F52] as Age_47 ,[F53] as Age_48 ,[F54] as Age_49 ,[F55] as
Age_50 ,[F56] as Age_51 ,[F57] as Age_52 ,[F58] as Age_53 ,[F59] as Age_54 ,[F60] as Age_55 ,[F61] as Age_56 ,[F62] as Age_57 ,[F63] as Age_58
,[F64] as Age_59 ,[F65] as Age_60 ,[F66] as Age_61 ,[F67] as Age_62 ,[F68] as Age_63 ,[F69] as Age_64 ,[F70] as Age_65 ,[F71] as Age_66 ,[F72] as
Age_67 ,[F73] as Age_68 ,[F74] as Age_69 ,[F75] as Age_70 ,[F76] as Age_71 ,[F77] as Age_72 ,[F78] as Age_73 ,[F79] as Age_74 ,[F80] as Age_75
,[F81] as Age_76 ,[F82] as Age_77 ,[F83] as Age_78 ,[F84] as Age_79 ,[F85] as Age_80 ,[F86] as Age_81 ,[F87] as Age_82 ,[F88] as Age_83 ,[F89] as
Age_84 ,[F90] as Age_85 ,[F91] as Age_86 ,[F92] as Age_87 ,[F93] as Age_88 ,[F94] as Age_89 ,[F95] as Age_90_and_above

FROM [CR003].[dbo].[Mid-2018 Persons] WHERE [F3] IS NOT NULL AND LOWER([Contents]) <> 'area codes'

```

- Create table – Streets\_RawBulk\_Extract

```

USE CR003;
GO

CREATE TABLE Streets_RawBulkExtract
(
    -- Primary Key cannot be given into raw extract while bulk insert
    --Street_ID           INT          IDENTITY(1,1)      PRIMARY KEY,
    [Crime_ID]          NVARCHAR(100)
    ,[Month]             NVARCHAR(100)
    ,[Reported_by]       NVARCHAR(100)
    ,[Falls_within]     NVARCHAR(100)
    ,[Longitude]         NVARCHAR(100)
    ,[Latitude]          NVARCHAR(100)
    ,[Location]          NVARCHAR(100)
    ,[LSOA_code]         NVARCHAR(100)
    ,[LSOA_name]         NVARCHAR(100)
    ,[Crime_type]        NVARCHAR(100)
    ,[Last_outcome_category] NVARCHAR(100)
    ,[Context]            NVARCHAR(100)
);

```

- Create a table – zz\_filelist for the list of files present in a windows folder path – this is to enable bulk insert

```

USE CR003;
GO

--TRUNCATE TABLE dbo.Streets_RawBulkExtract;

/* Commenting out the below part, because it only extracts a single file*/

--BULK INSERT dbo.Streets_RawBulkExtract
--FROM 'C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-01-greater-manchester-street.csv'
--WITH
--(
--    FIRSTROW = 2,
--    ROWTERMINATOR = '\n',
--    FIELDTERMINATOR = ','
--)

/* I would like to extract all 24 files at the same time into SSMS - so I'm using the below piece of script*/

TRUNCATE TABLE dbCrime.zz_filelist;

CREATE TABLE zz_filelist (
    ID INT IDENTITY(1,1) PRIMARY KEY,
    FULLFILEPATH VARCHAR(250),
    FILENAME VARCHAR(100)
)

INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-01-greater-manchester-street.csv', '2017-01-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-02-greater-manchester-street.csv', '2017-02-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-03-greater-manchester-street.csv', '2017-03-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-04-greater-manchester-street.csv', '2017-04-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-05-greater-manchester-street.csv', '2017-05-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-06-greater-manchester-street.csv', '2017-06-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-07-greater-manchester-street.csv', '2017-07-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-08-greater-manchester-street.csv', '2017-08-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-09-greater-manchester-street.csv', '2017-09-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-10-greater-manchester-street.csv', '2017-10-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-11-greater-manchester-street.csv', '2017-11-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2017-12-greater-manchester-street.csv', '2017-12-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-01-greater-manchester-street.csv', '2018-01-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-02-greater-manchester-street.csv', '2018-02-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-03-greater-manchester-street.csv', '2018-03-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-04-greater-manchester-street.csv', '2018-04-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-05-greater-manchester-street.csv', '2018-05-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-06-greater-manchester-street.csv', '2018-06-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-07-greater-manchester-street.csv', '2018-07-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-08-greater-manchester-street.csv', '2018-08-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-09-greater-manchester-street.csv', '2018-09-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-10-greater-manchester-street.csv', '2018-10-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-11-greater-manchester-street.csv', '2018-11-greater-manchester-street.csv')
INSERT INTO dbCrime.zz_filelist
    VALUES ('C:\Users\ambar\Desktop\Crime in Greater Manchester\2018-12-greater-manchester-street.csv', '2018-12-greater-manchester-street.csv')

```

- [11] Bulk Insert into table – Streets\_RawBulkExtract from the table zz\_filelist

```

BEGIN TRY

DECLARE @SQLTEXT NVARCHAR(250)
DECLARE @MYPATH NVARCHAR(250)

DECLARE MYCURSOR CURSOR FORWARD_ONLY

FOR SELECT FULLFILEPATH FROM dbCrime.zz_filelist

    OPEN MYCURSOR
    FETCH NEXT FROM MYCURSOR INTO @MYPATH

    WHILE @@FETCH_STATUS = 0

        BEGIN

            SET @SQLTEXT =

                'BULK INSERT dbCrime.Streets_RawBulkExtract FROM ''' +
                @MYPATH +
                ''' WITH
                (
                    FIRSTROW = 2,
                    FIELDTERMINATOR = ',',
                    ROWTERMINATOR = '\n'
                )'

            --PRINT @MYPATH
            --PRINT @SQLTEXT

            EXECUTE SP_EXECUTESQL @SQLTEXT

            FETCH NEXT FROM MYCURSOR INTO @MYPATH

        END

        CLOSE MYCURSOR
        DEALLOCATE MYCURSOR

    END TRY
    BEGIN CATCH
        SELECT ERROR_MESSAGE() AS [Error Message]
        ,ERROR_LINE() AS ErrorLine
        ,ERROR_NUMBER() AS [Error Number]
        ,ERROR_SEVERITY() AS [Error Severity]
        ,ERROR_STATE() AS [Error State]
    END CATCH

```

- Create table – Streets

```

-- After Bulk insert is completed, we can insert into new table using %PK

CREATE TABLE Streets
(
    Street_ID      INT      IDENTITY(1,1)      PRIMARY KEY,
    [Crime_ID]     NVARCHAR(100)
    ,[Period]       NVARCHAR(10)
    ,[Year]         SMALLINT
    ,[Month]        TINYINT
    ,[Reported_by] NVARCHAR(50)
    ,[Falls_within] NVARCHAR(50)
    ,[Longitude]   FLOAT
    ,[Latitude]    FLOAT
    ,[Location]    NVARCHAR(50)
    ,[LSOA_code]   VARCHAR(10)
    ,[LSOA_name]   NVARCHAR(50)
    ,[Crime_type]  NVARCHAR(50)
    ,[Last_outcome_category] NVARCHAR(100)
    ,[Context]      NVARCHAR(1)

    ,CONSTRAINT FK_LSOA_ID FOREIGN KEY ([LSOA_code])
        REFERENCES [dbCrime].[Areas](Area_ID)
);

```

- Insert into table Streets from Streets\_RawBulkExtract which was populated with all the .csv files.

```

INSERT INTO dbo.Streets

SELECT [Crime_ID]
      ,[Month] AS [Period]
      ,CAST(LEFT([MONTH],4) AS INT) AS [Year]
      ,CAST(RIGHT([MONTH],2) AS INT) AS [Month]
      ,[Reported_by]
      ,[Falls_within]
      ,[Longitude]
      ,[Latitude]
      ,[Location]
      ,[LSOA_code]
      ,[LSOA_name]
      ,[Crime_type]
      ,[Last_outcome_category]
      ,[Context]
  FROM [CR003].[dbo].[Streets_RawBulkExtract]
;

/*Validation check - 805055 rows - Validation complete*/
SELECT COUNT([Period]) FROM dbo.Streets;
--TRUNCATE TABLE dbCrime.Streets;

```

### 3.6.2 Views

- View 01 - v01\_AreaWise\_Population\_All

```

USE CR003;
GO

/******************/
/*View 01 Creation - v01_AreaWise_Population_All*/
/******************/

CREATE VIEW v01_Arealwise_Population_All
AS
(
SELECT dbCrime.Areas.Area_ID,
       RTRIM(LTRIM(REPLACE(dbCrime.Areas.Area_Name,RIGHT(dbCrime.Areas.Area_Name , CHARINDEX(' ', REVERSE(dbCrime.Areas.Area_Name))-1), ''))) AS [County],
       RIGHT([Area_Name] , CHARINDEX('_', REVERSE([Area_Name]))-1) AS [Area]
       ,RIGHT(dbCrime.Population.Area_ID_Gender , CHARINDEX('_', REVERSE(dbCrime.Population.Area_ID_Gender))-1) AS [Gender]
       ,dbCrime.Population.Year, dbCrime.Population.All_Ages, dbCrime.Population.Age_0, dbCrime.Population.Age_1,
       dbCrime.Population.Age_2, dbCrime.Population.Age_3, dbCrime.Population.Age_4, dbCrime.Population.Age_5, dbCrime.Population.Age_6,
       dbCrime.Population.Age_7, dbCrime.Population.Age_8, dbCrime.Population.Age_9, dbCrime.Population.Age_10,
       dbCrime.Population.Age_11, dbCrime.Population.Age_12, dbCrime.Population.Age_13, dbCrime.Population.Age_14,
       dbCrime.Population.Age_15, dbCrime.Population.Age_16, dbCrime.Population.Age_17, dbCrime.Population.Age_18,
       dbCrime.Population.Age_19, dbCrime.Population.Age_20, dbCrime.Population.Age_21, dbCrime.Population.Age_22,
       dbCrime.Population.Age_23, dbCrime.Population.Age_24, dbCrime.Population.Age_25, dbCrime.Population.Age_26,
       dbCrime.Population.Age_27, dbCrime.Population.Age_28, dbCrime.Population.Age_29, dbCrime.Population.Age_30,
       dbCrime.Population.Age_31, dbCrime.Population.Age_32, dbCrime.Population.Age_33, dbCrime.Population.Age_34,
       dbCrime.Population.Age_35, dbCrime.Population.Age_36, dbCrime.Population.Age_37, dbCrime.Population.Age_38,
       dbCrime.Population.Age_39, dbCrime.Population.Age_40, dbCrime.Population.Age_41, dbCrime.Population.Age_42,
       dbCrime.Population.Age_43, dbCrime.Population.Age_44, dbCrime.Population.Age_45, dbCrime.Population.Age_46,
       dbCrime.Population.Age_47, dbCrime.Population.Age_48, dbCrime.Population.Age_49, dbCrime.Population.Age_50,
       dbCrime.Population.Age_51, dbCrime.Population.Age_52, dbCrime.Population.Age_53, dbCrime.Population.Age_54,
       dbCrime.Population.Age_55, dbCrime.Population.Age_56, dbCrime.Population.Age_57, dbCrime.Population.Age_58,
       dbCrime.Population.Age_59, dbCrime.Population.Age_60, dbCrime.Population.Age_61, dbCrime.Population.Age_62,
       dbCrime.Population.Age_63, dbCrime.Population.Age_64, dbCrime.Population.Age_65, dbCrime.Population.Age_66,
       dbCrime.Population.Age_67, dbCrime.Population.Age_68, dbCrime.Population.Age_69, dbCrime.Population.Age_70,
       dbCrime.Population.Age_71, dbCrime.Population.Age_72, dbCrime.Population.Age_73, dbCrime.Population.Age_74,
       dbCrime.Population.Age_75, dbCrime.Population.Age_76, dbCrime.Population.Age_77, dbCrime.Population.Age_78,
       dbCrime.Population.Age_79, dbCrime.Population.Age_80, dbCrime.Population.Age_81, dbCrime.Population.Age_82,
       dbCrime.Population.Age_83, dbCrime.Population.Age_84, dbCrime.Population.Age_85, dbCrime.Population.Age_86,
       dbCrime.Population.Age_87, dbCrime.Population.Age_88, dbCrime.Population.Age_89,
       dbCrime.Population.Age_90_and_above
  FROM dbCrime.Areas INNER JOIN
       dbCrime.Population ON dbCrime.Areas.Area_ID = dbCrime.Population.Area_ID
 WHERE RIGHT(dbCrime.Population.Area_ID_Gender , CHARINDEX('_', REVERSE(dbCrime.Population.Area_ID_Gender))-1) = 'A'
)
;
```

- View 02 - v02\_AreaWise\_Population\_Genders

```

USE CR003;
GO

/*****************/
/*View 02 Creation - v02_AreaWise_Population_Genders*/
/*****************/

CREATE VIEW v02_AreaWise_Population_Genders
AS
(
SELECT dbCrime.Areas.Area_ID,
       RTRIM(LTRIM(REPLACE(dbCrime.Areas.Area_Name,RIGHT(dbCrime.Areas.Area_Name , CHARINDEX(' ', REVERSE(dbCrime.Areas.Area_Name)) -1),
        ''))) as [County],
       RIGHT([Area_Name] , CHARINDEX(' ', REVERSE([Area_Name])) -1) as [Area]
      ,RIGHT(dbCrime.Population.Area_ID_Gender , CHARINDEX(' ', REVERSE(dbCrime.Population.Area_ID_Gender)) -1) as [Gender]
      ,dbCrime.Population.Year, dbCrime.Population.All_Ages, dbCrime.Population.Age_0, dbCrime.Population.Age_1,
      dbCrime.Population.Age_2,
      dbCrime.Population.Age_3, dbCrime.Population.Age_4, dbCrime.Population.Age_5, dbCrime.Population.Age_6, dbCrime.Population.Age_7,
      dbCrime.Population.Age_8, dbCrime.Population.Age_9, dbCrime.Population.Age_10,
      dbCrime.Population.Age_11, dbCrime.Population.Age_12, dbCrime.Population.Age_13, dbCrime.Population.Age_14,
      dbCrime.Population.Age_15, dbCrime.Population.Age_16, dbCrime.Population.Age_17, dbCrime.Population.Age_18,
      dbCrime.Population.Age_19, dbCrime.Population.Age_20, dbCrime.Population.Age_21, dbCrime.Population.Age_22,
      dbCrime.Population.Age_23, dbCrime.Population.Age_24, dbCrime.Population.Age_25, dbCrime.Population.Age_26,
      dbCrime.Population.Age_27, dbCrime.Population.Age_28, dbCrime.Population.Age_29, dbCrime.Population.Age_30,
      dbCrime.Population.Age_31, dbCrime.Population.Age_32, dbCrime.Population.Age_33, dbCrime.Population.Age_34,
      dbCrime.Population.Age_35, dbCrime.Population.Age_36, dbCrime.Population.Age_37, dbCrime.Population.Age_38,
      dbCrime.Population.Age_39, dbCrime.Population.Age_40, dbCrime.Population.Age_41, dbCrime.Population.Age_42,
      dbCrime.Population.Age_43, dbCrime.Population.Age_44, dbCrime.Population.Age_45, dbCrime.Population.Age_46,
      dbCrime.Population.Age_47, dbCrime.Population.Age_48, dbCrime.Population.Age_49, dbCrime.Population.Age_50,
      dbCrime.Population.Age_51, dbCrime.Population.Age_52, dbCrime.Population.Age_53, dbCrime.Population.Age_54,
      dbCrime.Population.Age_55, dbCrime.Population.Age_56, dbCrime.Population.Age_57, dbCrime.Population.Age_58,
      dbCrime.Population.Age_59, dbCrime.Population.Age_60, dbCrime.Population.Age_61, dbCrime.Population.Age_62,
      dbCrime.Population.Age_63, dbCrime.Population.Age_64, dbCrime.Population.Age_65, dbCrime.Population.Age_66,
      dbCrime.Population.Age_67, dbCrime.Population.Age_68, dbCrime.Population.Age_69, dbCrime.Population.Age_70,
      dbCrime.Population.Age_71, dbCrime.Population.Age_72, dbCrime.Population.Age_73, dbCrime.Population.Age_74,
      dbCrime.Population.Age_75, dbCrime.Population.Age_76, dbCrime.Population.Age_77, dbCrime.Population.Age_78,
      dbCrime.Population.Age_79, dbCrime.Population.Age_80, dbCrime.Population.Age_81, dbCrime.Population.Age_82,
      dbCrime.Population.Age_83, dbCrime.Population.Age_84, dbCrime.Population.Age_85, dbCrime.Population.Age_86,
      dbCrime.Population.Age_87, dbCrime.Population.Age_88, dbCrime.Population.Age_89,
      dbCrime.Population.Age_90_and_above
FROM   dbCrime.Areas INNER JOIN
      dbCrime.Population ON dbCrime.Areas.Area_ID = dbCrime.Population.Area_ID
      WHERE  RIGHT(dbCrime.Population.Area_ID_Gender , CHARINDEX(' ', REVERSE(dbCrime.Population.Area_ID_Gender)) -1) = 'M'
      OR    RIGHT(dbCrime.Population.Area_ID_Gender , CHARINDEX(' ', REVERSE(dbCrime.Population.Area_ID_Gender)) -1) = 'F';

```

- View 03 - v03\_Streets\_Population\_Merged

```

USE CR003;
GO

/*****************/
/*View 03 Creation - v03_Streets_Population_Merged*/
/*****************/

CREATE VIEW v03_Streets_Population_Merged AS
(
SELECT dbCrime.Streets.Street_ID, dbCrime.Streets.[Crime ID], dbCrime.Streets.Period, dbCrime.Streets.Year, dbCrime.Streets.Month,
dbCrime.Streets.Reported_By, dbCrime.Streets.Falls_within, dbCrime.Streets.Longitude,
      dbCrime.Streets.Latitude, dbCrime.Streets.Location, dbCrime.Streets.LSOA_code, dbCrime.Streets.LSOA_name,
      dbCrime.Streets.Crime_type, dbCrime.Streets.Last_outcome_category, dbCrime.Streets.Context,
      dbo.v01_AreaWise_Population_All.County, dbo.v01_AreaWise_Population_All.Gender, dbo.v01_AreaWise_Population_All.Age_0,
      dbo.v01_AreaWise_Population_All.Age_1, dbo.v01_AreaWise_Population_All.Age_2, dbo.v01_AreaWise_Population_All.Age_3,
      dbo.v01_AreaWise_Population_All.Age_4, dbo.v01_AreaWise_Population_All.Age_5,
      dbo.v01_AreaWise_Population_All.Age_6, dbo.v01_AreaWise_Population_All.Age_7, dbo.v01_AreaWise_Population_All.Age_8,
      dbo.v01_AreaWise_Population_All.Age_9, dbo.v01_AreaWise_Population_All.Age_10,
      dbo.v01_AreaWise_Population_All.Age_11, dbo.v01_AreaWise_Population_All.Age_12, dbo.v01_AreaWise_Population_All.Age_13,
      dbo.v01_AreaWise_Population_All.Age_14, dbo.v01_AreaWise_Population_All.Age_15,
      dbo.v01_AreaWise_Population_All.Age_16, dbo.v01_AreaWise_Population_All.Age_17, dbo.v01_AreaWise_Population_All.Age_18,
      dbo.v01_AreaWise_Population_All.Age_19, dbo.v01_AreaWise_Population_All.Age_20,
      dbo.v01_AreaWise_Population_All.Age_21, dbo.v01_AreaWise_Population_All.Age_22, dbo.v01_AreaWise_Population_All.Age_23,
      dbo.v01_AreaWise_Population_All.Age_24, dbo.v01_AreaWise_Population_All.Age_25,
      dbo.v01_AreaWise_Population_All.Age_26, dbo.v01_AreaWise_Population_All.Age_27, dbo.v01_AreaWise_Population_All.Age_28,
      dbo.v01_AreaWise_Population_All.Age_29, dbo.v01_AreaWise_Population_All.Age_30,
      dbo.v01_AreaWise_Population_All.Age_31, dbo.v01_AreaWise_Population_All.Age_32, dbo.v01_AreaWise_Population_All.Age_33,
      dbo.v01_AreaWise_Population_All.Age_34, dbo.v01_AreaWise_Population_All.Age_35,
      dbo.v01_AreaWise_Population_All.Age_36, dbo.v01_AreaWise_Population_All.Age_37, dbo.v01_AreaWise_Population_All.Age_38,
      dbo.v01_AreaWise_Population_All.Age_39, dbo.v01_AreaWise_Population_All.Age_40,
      dbo.v01_AreaWise_Population_All.Age_41, dbo.v01_AreaWise_Population_All.Age_42, dbo.v01_AreaWise_Population_All.Age_43,
      dbo.v01_AreaWise_Population_All.Age_44, dbo.v01_AreaWise_Population_All.Age_45,
      dbo.v01_AreaWise_Population_All.Age_46, dbo.v01_AreaWise_Population_All.Age_47, dbo.v01_AreaWise_Population_All.Age_48,
      dbo.v01_AreaWise_Population_All.Age_49, dbo.v01_AreaWise_Population_All.Age_50,
      dbo.v01_AreaWise_Population_All.Age_51, dbo.v01_AreaWise_Population_All.Age_52, dbo.v01_AreaWise_Population_All.Age_53,
      dbo.v01_AreaWise_Population_All.Age_54, dbo.v01_AreaWise_Population_All.Age_55,
      dbo.v01_AreaWise_Population_All.Age_56, dbo.v01_AreaWise_Population_All.Age_57, dbo.v01_AreaWise_Population_All.Age_58,
      dbo.v01_AreaWise_Population_All.Age_59, dbo.v01_AreaWise_Population_All.Age_60,
      dbo.v01_AreaWise_Population_All.Age_61, dbo.v01_AreaWise_Population_All.Age_62, dbo.v01_AreaWise_Population_All.Age_63,
      dbo.v01_AreaWise_Population_All.Age_64, dbo.v01_AreaWise_Population_All.Age_65,
      dbo.v01_AreaWise_Population_All.Age_66, dbo.v01_AreaWise_Population_All.Age_67, dbo.v01_AreaWise_Population_All.Age_68,
      dbo.v01_AreaWise_Population_All.Age_69, dbo.v01_AreaWise_Population_All.Age_70,
      dbo.v01_AreaWise_Population_All.Age_71, dbo.v01_AreaWise_Population_All.Age_72, dbo.v01_AreaWise_Population_All.Age_73,
      dbo.v01_AreaWise_Population_All.Age_74, dbo.v01_AreaWise_Population_All.Age_75,
      dbo.v01_AreaWise_Population_All.Age_76, dbo.v01_AreaWise_Population_All.Age_77, dbo.v01_AreaWise_Population_All.Age_78,
      dbo.v01_AreaWise_Population_All.Age_79, dbo.v01_AreaWise_Population_All.Age_80,
      dbo.v01_AreaWise_Population_All.Age_81, dbo.v01_AreaWise_Population_All.Age_82, dbo.v01_AreaWise_Population_All.Age_83,
      dbo.v01_AreaWise_Population_All.Age_84, dbo.v01_AreaWise_Population_All.Age_85,
      dbo.v01_AreaWise_Population_All.Age_86, dbo.v01_AreaWise_Population_All.Age_87, dbo.v01_AreaWise_Population_All.Age_88,
      dbo.v01_AreaWise_Population_All.Age_89, dbo.v01_AreaWise_Population_All.Age_90_and_above
FROM   dbo.v01_AreaWise_Population_All INNER JOIN
      dbCrime.Streets ON dbo.v01_AreaWise_Population_All.Area_ID = dbCrime.Streets.LSOA_code AND dbo.v01_AreaWise_Population_All.Year
      = dbCrime.Streets.Year;

```

### 3.6.2.1 Reports

- Report 01 - xRep01\_Population\_Classification

```

USE CR003; GO

/*****************/
/*Report 01 Creation - xRep01_Population_Classification*/
/*****************/

CREATE VIEW xRep01_Population_Classification AS ( SELECT tb1.[Area_ID], tb1.[County], tb1.[Area], tb1.[Year],
tb1.[Total_Population], tb1.[Male Population], tb2.[Female Population], 

tb1.[Children_Total(<=12years)], tb1.[Teenages_Total(13-19years)], tb1.[Adults_Total(20-50years)], tb1.[Pre-
Elderly_Total(51-64years)], tb1.[Elderly_Total(65+years)],

tb1.[Children_Male(<=12years)], tb1.[Teenages_Male(13-19years)], tb1.[Adults_Male(20-50years)], tb1.[Pre-
Elderly_Male(51-64years)], tb1.[Elderly_Male(65+years)],

tb2.[Children_Female(<=12years)], tb2.[Teenages_Female(13-19years)], tb2.[Adults_Female(20-50years)], tb2.[Pre-
Elderly_Female(51-64years)], tb2.[Elderly_Female(65+years)] 

FROM (SELECT a.[Area_ID] ,a.[County] ,a.[Area] ,a.[Gender] as [All] ,b.[Gender] as [M&F_FilterM] ,a.[Year]
,a.[All_Ages] as [Total_Population] ,b.[All_Ages] as [Male Population]

,a.[Age_0] + a.[Age_1] + a.[Age_2] + a.[Age_3] + a.[Age_4] + a.[Age_5] + a.[Age_6] + a.[Age_7] + a.[Age_8] +
a.[Age_9] + a.[Age_10] + a.[Age_11] + a.[Age_12] as [Children_Total(<=12years)] ,a.[Age_13] + a.[Age_14] + a.[Age_15] +
a.[Age_16] + a.[Age_17] + a.[Age_18] + a.[Age_19] as [Teenages_Total(13-19years)]

,a.[Age_20] + a.[Age_21] + a.[Age_22] + a.[Age_23] + a.[Age_24] + a.[Age_25] + a.[Age_26] + a.[Age_27] + a.[Age_28] +
a.[Age_29] + a.[Age_30] + a.[Age_31] + a.[Age_32] + a.[Age_33] + a.[Age_34] + a.[Age_35] + a.[Age_36] + a.[Age_37] +
a.[Age_38] + a.[Age_39] + a.[Age_40] + a.[Age_41] + a.[Age_42] + a.[Age_43] + a.[Age_44] + a.[Age_45] + a.[Age_46] +
a.[Age_47] + a.[Age_48] + a.[Age_49] + a.[Age_50] as [Adults_Total(20-50years)]

,a.[Age_51] + a.[Age_52] + a.[Age_53] + a.[Age_54] + a.[Age_55] + a.[Age_56] + a.[Age_57] + a.[Age_58] + a.[Age_59] +
a.[Age_60] + a.[Age_61] + a.[Age_62] + a.[Age_63] + a.[Age_64] as [Pre-Elderly_Total(51-64years)]

,a.[Age_65] + a.[Age_66] + a.[Age_67] + a.[Age_68] + a.[Age_69] + a.[Age_70] + a.[Age_71] + a.[Age_72] + a.[Age_73] +
a.[Age_74] + a.[Age_75] + a.[Age_76] + a.[Age_77] + a.[Age_78] + a.[Age_79] + a.[Age_80] + a.[Age_81] + a.[Age_82] +
a.[Age_83] + a.[Age_84] + a.[Age_85] + a.[Age_86] + a.[Age_87] + a.[Age_88] + a.[Age_89] + a.[Age_90_and_above] as
[Elderly_Total(65+years)]

,b.[Age_0] + b.[Age_1] + b.[Age_2] + b.[Age_3] + b.[Age_4] + b.[Age_5] + b.[Age_6] + b.[Age_7] + b.[Age_8] +
b.[Age_9] + b.[Age_10] + b.[Age_11] + b.[Age_12] as [Children_Male(<=12years)] ,b.[Age_13] + b.[Age_14] + b.[Age_15] +
b.[Age_16] + b.[Age_17] + b.[Age_18] + b.[Age_19] as [Teenages_Male(13-19years)]

,b.[Age_20] + b.[Age_21] + b.[Age_22] + b.[Age_23] + b.[Age_24] + b.[Age_25] + b.[Age_26] + b.[Age_27] + b.[Age_28] +
b.[Age_29] + b.[Age_30] + b.[Age_31] + b.[Age_32] + b.[Age_33] + b.[Age_34] + b.[Age_35] + b.[Age_36] + b.[Age_37] +
b.[Age_38] + b.[Age_39] + b.[Age_40] + b.[Age_41] + b.[Age_42] + b.[Age_43] + b.[Age_44] + b.[Age_45] + b.[Age_46] +
b.[Age_47] + b.[Age_48] + b.[Age_49] + b.[Age_50] as [Adults_Male(20-50years)]

,b.[Age_51] + b.[Age_52] + b.[Age_53] + b.[Age_54] + b.[Age_55] + b.[Age_56] + b.[Age_57] + b.[Age_58] + b.[Age_59] +
b.[Age_60] + b.[Age_61] + b.[Age_62] + b.[Age_63] + b.[Age_64] as [Pre-Elderly_Male(51-64years)]

,b.[Age_65] + b.[Age_66] + b.[Age_67] + b.[Age_68] + b.[Age_69] + b.[Age_70] + b.[Age_71] + b.[Age_72] + b.[Age_73] +
b.[Age_74] + b.[Age_75] + b.[Age_76] + b.[Age_77] + b.[Age_78] + b.[Age_79] + b.[Age_80] + b.[Age_81] + b.[Age_82] +
b.[Age_83] + b.[Age_84] + b.[Age_85] + b.[Age_86] + b.[Age_87] + b.[Age_88] + b.[Age_89] + b.[Age_90_and_above] as
[Elderly_Male(65+years)]

FROM [CR003].[dbo].[v01_AreaWise_Population_All] as a INNER JOIN [dbo].[v02_AreaWise_Population_Genders] as b ON
(a.[Area_ID] = b.[Area_ID] AND a.[Year] = b.[Year]) WHERE b.[Gender] = 'M' ) as tb1 INNER JOIN (SELECT a.[Area_ID]
,a.[County] ,a.[Area] ,a.[Gender] as [All] ,b.[Gender] as [M&F_FilterF] ,a.[Year] ,a.[All_Ages] as [Total_Population]
,b.[All_Ages] as [Female Population]

,b.[Age_0] + b.[Age_1] + b.[Age_2] + b.[Age_3] + b.[Age_4] + b.[Age_5] + b.[Age_6] + b.[Age_7] + b.[Age_8] +
b.[Age_9] + b.[Age_10] + b.[Age_11] + b.[Age_12] as [Children_Female(<=12years)] ,b.[Age_13] + b.[Age_14] + b.[Age_15] +
b.[Age_16] + b.[Age_17] + b.[Age_18] + b.[Age_19] as [Teenages_Female(13-19years)]

,b.[Age_20] + b.[Age_21] + b.[Age_22] + b.[Age_23] + b.[Age_24] + b.[Age_25] + b.[Age_26] + b.[Age_27] + b.[Age_28] +
b.[Age_29] + b.[Age_30] + b.[Age_31] + b.[Age_32] + b.[Age_33] + b.[Age_34] + b.[Age_35] + b.[Age_36] + b.[Age_37] +
b.[Age_38] + b.[Age_39] + b.[Age_40] + b.[Age_41] + b.[Age_42] + b.[Age_43] + b.[Age_44] + b.[Age_45] + b.[Age_46] +
b.[Age_47] + b.[Age_48] + b.[Age_49] + b.[Age_50] as [Adults_Female(20-50years)]

,b.[Age_51] + b.[Age_52] + b.[Age_53] + b.[Age_54] + b.[Age_55] + b.[Age_56] + b.[Age_57] + b.[Age_58] + b.[Age_59] +
b.[Age_60] + b.[Age_61] + b.[Age_62] + b.[Age_63] + b.[Age_64] as [Pre-Elderly_Female(51-64years)]

,b.[Age_65] + b.[Age_66] + b.[Age_67] + b.[Age_68] + b.[Age_69] + b.[Age_70] + b.[Age_71] + b.[Age_72] + b.[Age_73] +
b.[Age_74] + b.[Age_75] + b.[Age_76] + b.[Age_77] + b.[Age_78] + b.[Age_79] + b.[Age_80] + b.[Age_81] + b.[Age_82] +
b.[Age_83] + b.[Age_84] + b.[Age_85] + b.[Age_86] + b.[Age_87] + b.[Age_88] + b.[Age_89] + b.[Age_90_and_above] as
[Elderly_Female(65+years)]


FROM [CR003].[dbo].[v01_AreaWise_Population_All] as a INNER JOIN [dbo].[v02_AreaWise_Population_Genders] as b ON
(a.[Area_ID] = b.[Area_ID] AND a.[Year] = b.[Year]) WHERE b.[Gender] = 'F' ) as tb2 ON (tb1.[Area_ID] = tb2.[Area_ID]
AND tb1.[Year] = tb2.[Year]) );

```

- Report 2 – xRep02\_PopbyAgeGenderCleaned

```

USE CR003;
GO

/*****************/
/*Report 02 Creation - xRep02_PopbyAgeGenderCleaned*/
/*****************/

CREATE VIEW xRep02_PopbyAgeGenderCleaned
AS
(
    SELECT [Area_ID],[County],[Area],[Year],
           IIF(CHARINDEX('Female',Age_Gender_class) > 0, 'Female', 'Male') as Gender,
           Age_Gender_class, Total
    FROM
        (SELECT [Area_ID],[County],[Area],[Year],
               [Children_Male(<=12years)]
              ,[Teenages_Male(13-19years)]
              ,[Adults_Male(20-50years)]
              ,[Pre-Elderly_Male(51-64years)]
              ,[Elderly_Male(65+years)]
              ,[Children_Female(<=12years)]
              ,[Teenages_Female(13-19years)]
              ,[Adults_Female(20-50years)]
              ,[Pre-Elderly_Female(51-64years)]
              ,[Elderly_Female(65+years)]
        FROM xRep01_Population_Classification) p
    UNPIVOT
        (Total FOR Age_Gender_class IN
            ([Children_Male(<=12years)]
             ,[Teenages_Male(13-19years)]
             ,[Adults_Male(20-50years)]
             ,[Pre-Elderly_Male(51-64years)]
             ,[Elderly_Male(65+years)]
             ,[Children_Female(<=12years)]
             ,[Teenages_Female(13-19years)]
             ,[Adults_Female(20-50years)]
             ,[Pre-Elderly_Female(51-64years)]
             ,[Elderly_Female(65+years)])
        )AS unpvt
)
;

```

- Report 3 - xRep03\_Crime\_Rate

```

USE CR003;
GO

/*****************/
/*Report 03 Creation - xRep03_Crime_Rate*/
/*****************/

-- Crime Rate = (Number of reported crimes / Total Population) * 100,000

CREATE VIEW xRep03_Crime_Rate
AS (
    SELECT [County], [Area], [Longitude], [Latitude], [Year], [Month] as Month_num,
           DATENAME(month, DATEADD(month, [Month]-1, CAST('2008-01-01' AS datetime))) as [Month],
           [Period], [Crime_type], [Location], [Last_outcome_category], [LSOA_code]
           ,COUNT(DISTINCT Street_ID) as Total_Crimes
           ,ROUND(CAST((COUNT(DISTINCT Street_ID) / CAST([All_Ages] AS FLOAT)) as FLOAT)*100000, 2) as
           [Crime_Rate(per 100K)]
    FROM [CR003].[dbo].[v03_Streets_Population_Merged]
    GROUP BY [County], [Area], [Longitude], [Latitude], [Year], [Month], [Period], [Crime_type], [All_Ages],
    [Location], [Last_outcome_category], [LSOA_code]
) ;

```

- Report 4 - xRep04\_Crime\_Outcomes

```

USE CR003;
GO

/*****Report 04 Creation - xRep04_Crime_Outcomes*/
/*****Report 04 Creation - xRep04_Crime_Outcomes*/
CREATE VIEW xRep04_Crime_Outcomes
AS (
SELECT [County], [Area], [Longitude], [Latitude], [Year], [Month] AS Month_num, DATENAME(month, DATEADD(month, [Month]-1,
CAST('2008-01-01' AS datetime))) AS [Month], [Period], [Crime_type] , [Last_outcome_category] , [LSOA_code] , COUNT(DISTINCT
Street_ID) AS Total_Crimes
FROM [CR003].[dbo].[v03_Streets_Population_Merged]
GROUP BY [County], [Area], [Longitude], [Latitude], [Year], [Month], [Period], [Crime_type], [All_Ages],
[Last_outcome_category] ,[LSOA_code]
);

```

### 3.6.3 Stored Procedures (if any)

```

-- =====
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: <Ambareesh Jonnavittula>
-- Create date: <10-May-2021>
-- Description: <Total crimes per County>
-- =====
CREATE PROCEDURE spCrimes_per_County
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT County, sum(Total_Crimes)
        FROM [dbo].[xRep03_Crime_Rate]
        GROUP BY County
END
GO

ALTER PROCEDURE spCrimes_per_County
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT County, sum(Total_Crimes) AS Total_Crimes
        FROM [dbo].[xRep03_Crime_Rate]
        GROUP BY County
        ORDER BY sum(Total_Crimes) DESC
END
GO

CREATE PROCEDURE spAvgCrimeRate_per_County
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT County, avg([Crime_Rate(per 100K)]) AS Avg_Crime_Rate
        FROM [dbo].[xRep03_Crime_Rate]
        GROUP BY County
END
GO

```

### 3.6.4 Triggers (if any)

```
USE CR003;
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- =====
-- Trigger 1 - Intercept Deletion
-- =====
-- Author: Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCrime.trg_Intercept_Deletion
ON [dbCrime].[Areas]
INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 2 - Intercept Deletion
-- =====
-- Author: Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCrime.trg_Intercept_Deletion2
ON [dbCrime].[Population]
INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion2` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO

-- =====
-- Trigger 3 - Intercept Deletion
-- =====
-- Author: Ambareesh Jonnavittula
-- Create date: 10-May-2021
-- Description: Intercept Deletion Trigger
-- =====
CREATE TRIGGER dbCrime.trg_Intercept_Deletion3
ON [dbCrime].[Streets]
INSTEAD OF INSERT, UPDATE, DELETE
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    RAISERROR('Trigger `trg_Intercept_Deletion3` needs to be deactivated first', 16, 1)

    Rollback transaction
    Return
END
GO
```

### 3.6.5 User Define Functions (if any)

N/A

### 3.7. Report Design (Only for task 2 and task 3)

- QGIS was not properly installed in my PC. Whenever I open this application, I see the opening screen that 'Setup in progress' and then nothing happens. Since I was unable to install QGIS on my PC, I continued my exploration using Power BI.
- I also checked the same from Event Viewer, and it seems like a windows-based error.

**Application** Number of events: 32,281 (!) New events available

Level	Date and Time	Source	Event ID	Task Category
Information	5/11/2021 7:23:57 PM	Windows Error Reporting	1001	None
Information	5/11/2021 7:23:56 PM	Windows Error Reporting	1001	None
Error	5/11/2021 7:23:54 PM	Application Error	1000 (100)	
Information	5/11/2021 7:23:15 PM	MSSQL\$SQLDEV	18456	Logon
Information	5/11/2021 7:22:15 PM	MSSQL\$SQLDEV	18456	Logon
Information	5/11/2021 7:22:03 PM	Windows Error Reporting	1001	None
Information	5/11/2021 7:22:03 PM	Windows Error Reporting	1001	None
Information	5/11/2021 7:21:15 PM	MSSQL\$SQLDEV	18456	Logon

**Event 1000, Application Error**

General Details

Faulting application name: **qqgis-bin.exe**, version: 0.0.0.0, time stamp: 0x607a2ce3  
 Faulting module name: DbgHelp.dll\_unloaded, version: 10.0.19041.867, time stamp: 0x7c197411  
 Exception code: 0xc0000005  
 Fault offset: 0x0000000000b863e  
 Faulting process id: 0x4410  
 Faulting application start time: 0x01d74692c9a4bbd7  
**Faulting application path: C:\Program Files\QGIS 3.18\bin\qqgis-bin.exe**  
 Faulting module path: DbgHelp.dll  
 Report Id: f27025e5-cba2-4c1c-aea3-9e71265d3119  
 Faulting package full name:  
 Faulting package-relative application ID:

**Windows Application Logs**

Log Name: Application  
 Source: Application Error  
 Event ID: 1000  
 Level: Error  
 User: N/A  
 OpCode: Info  
 More Information: [Event Log Online Help](#)

- A dashboard has been created for the crimes in Manchester from the reports that were created in SSMS. Initially we connect to the server and the database CR003 to extract the necessary data objects.

**SQL Server database**

Server ⓘ LAPTOP-MK-2\SQLEXPRESS

Database (optional) CR003

Data Connectivity mode ⓘ  
 Import  
 DirectQuery

> Advanced options

**File Home Transform Add Column**

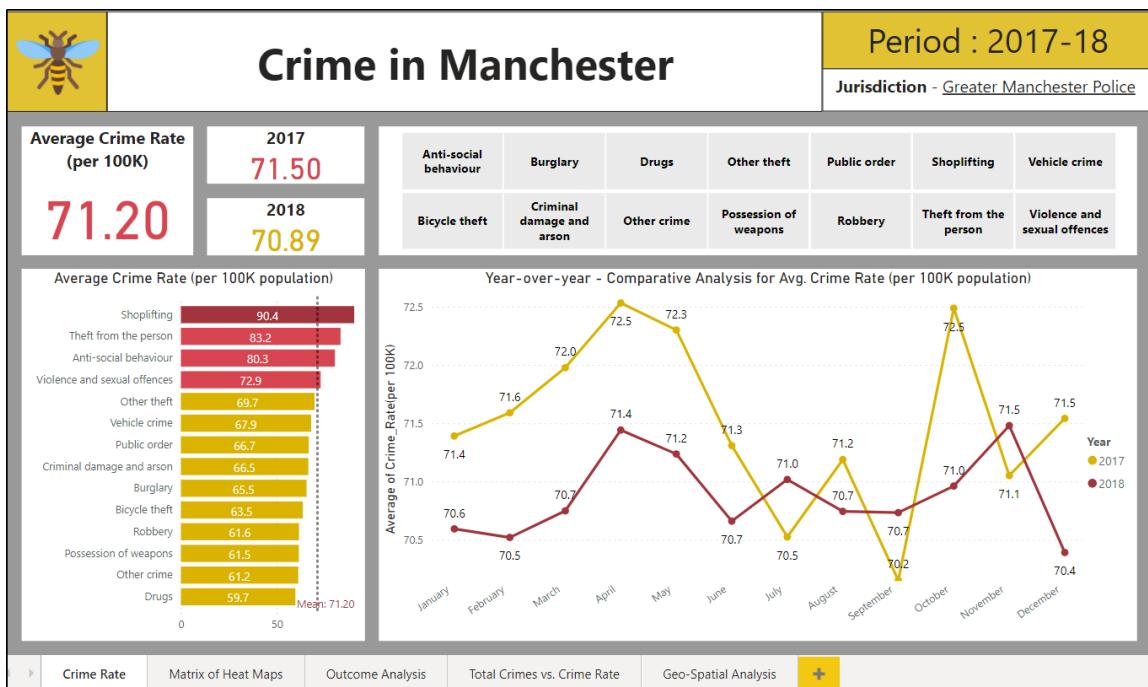
Close & Apply ▾ New Recent Enter Data Data source settings Data Sources Pa

Close New Query Queries [3]

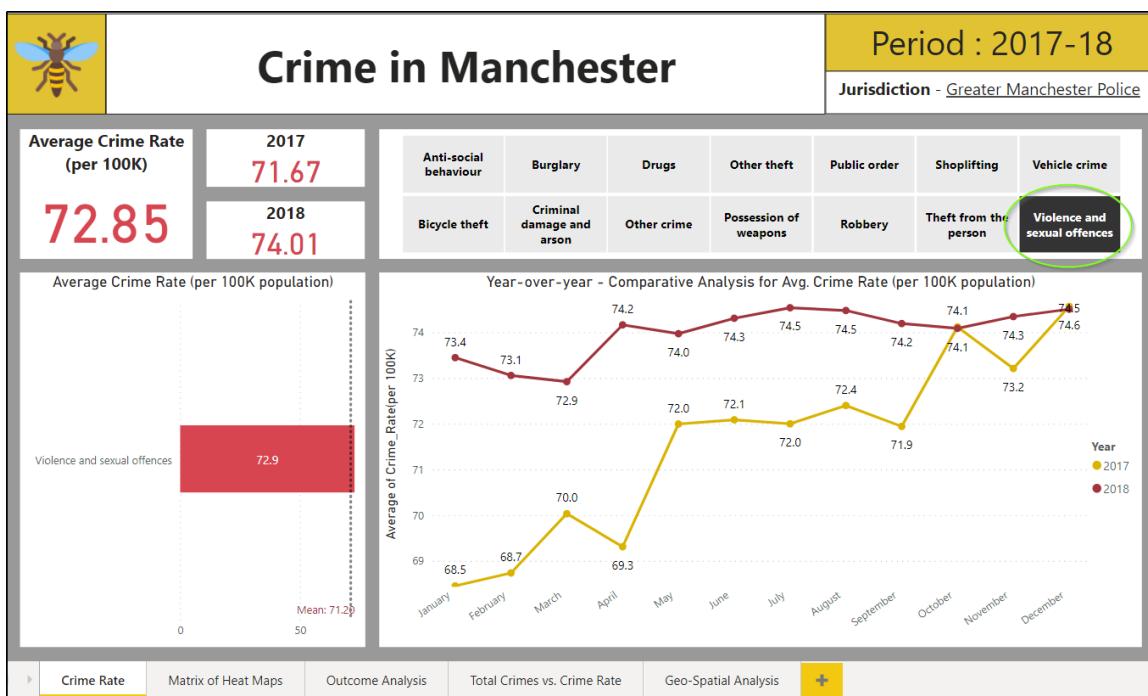
xRep02\_PopbyAgeGenderCleaned  
**xRep03\_Crime\_Rate** (highlighted with a green circle)  
 Months\_Sorted

- **Crime Rate**

- As we could see, the average crime rate (per 100K) is 71.20 and it was the highest for Shoplifting, theft from the person, anti-social behaviour, and violence. All these 4 were grouped as red-flags because their individual mean crime rate is more than the total mean crime rate.
- Under the jurisdiction of Greater Manchester Police, the overall crime rate has gone down from 2017 to 2018 as we can see the average reducing from 71.50 to 70.89.

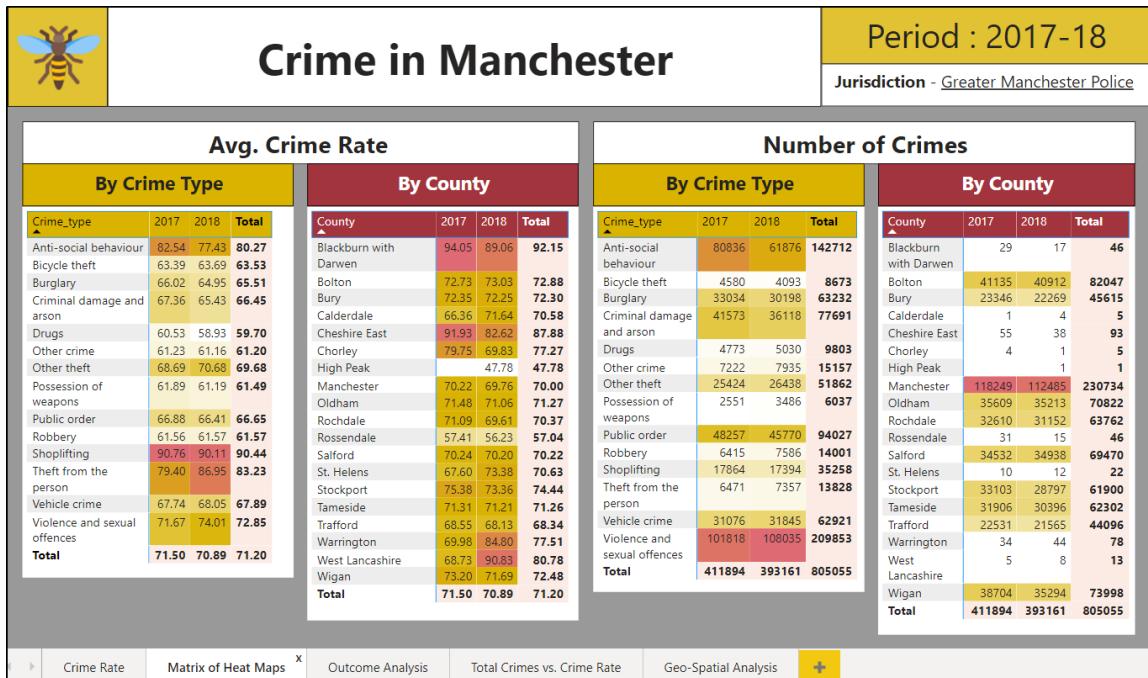


- The trends can be seen as following, and a horizontal slicer has been setup to individually choose the type of crime which we would like to emphasize. For example – Violence and sexual offences is selected as shown below.



- **Matrix of HeatMaps –**

- Average crime rate vs. number of crimes is shown below by crime type and by county.
- As we could observe, shoplifting, anti-social behaviour, violence and sexual offences are showing prominently by crime type.
- By county, Blackburn with Darwen, Cheshire East have dangerous places but reduced from 2017 to 2018, whereas Warrington and West Lancashire have increased crime rate from 2017 to 2018.
- By county, Manchester has the highest the number of crimes, but reduced very slightly from 2017 to 2018.



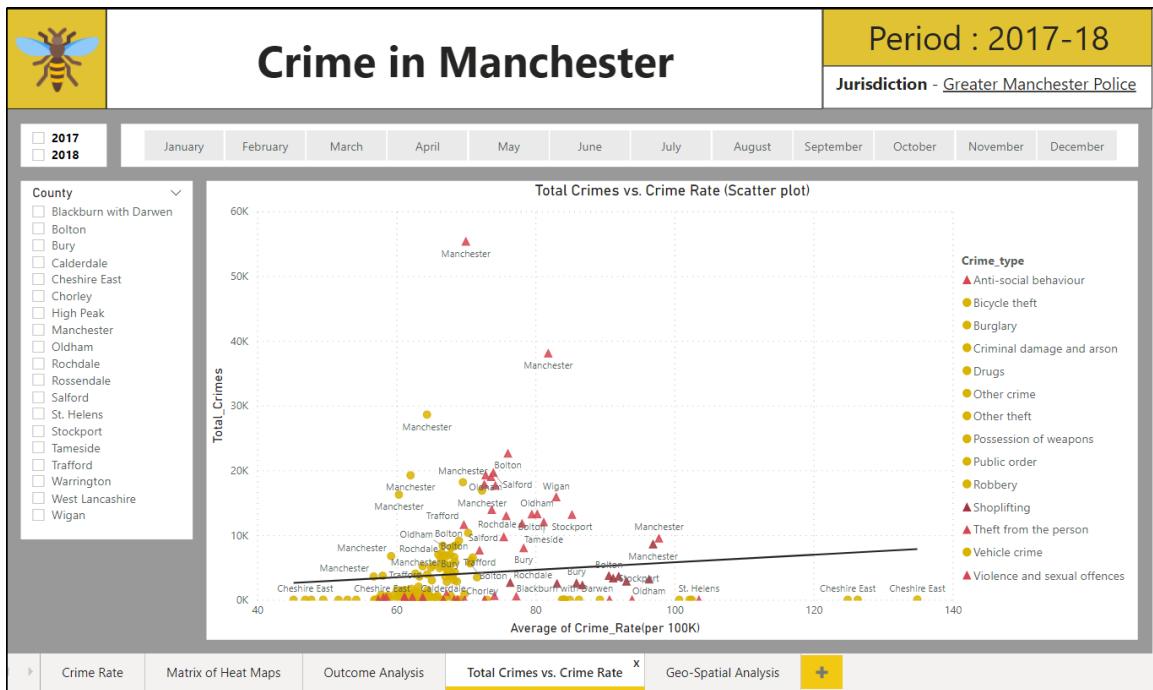
- **Outcome Analysis –**

- Out of the 805855 cases, around 380,000 cases have no suspect identified after the investigation is complete, and around 144,000 are unable to prosecute suspect which means the evidence was inconclusive.



- **Total Crimes vs. Crime Rate –**

- The scatter plot below shows the average crime rate (per 100K population) vs. the total crimes.
- As we could see, the main 4 categories of crime types that we classified are highlighted in red, and among these, the number of cases for violence and sexual offences, anti-social behaviour are the highest, especially in Manchester.
- Whereas the crime rate is maximum in Cheshire East – vehicle crime and Shoplifting

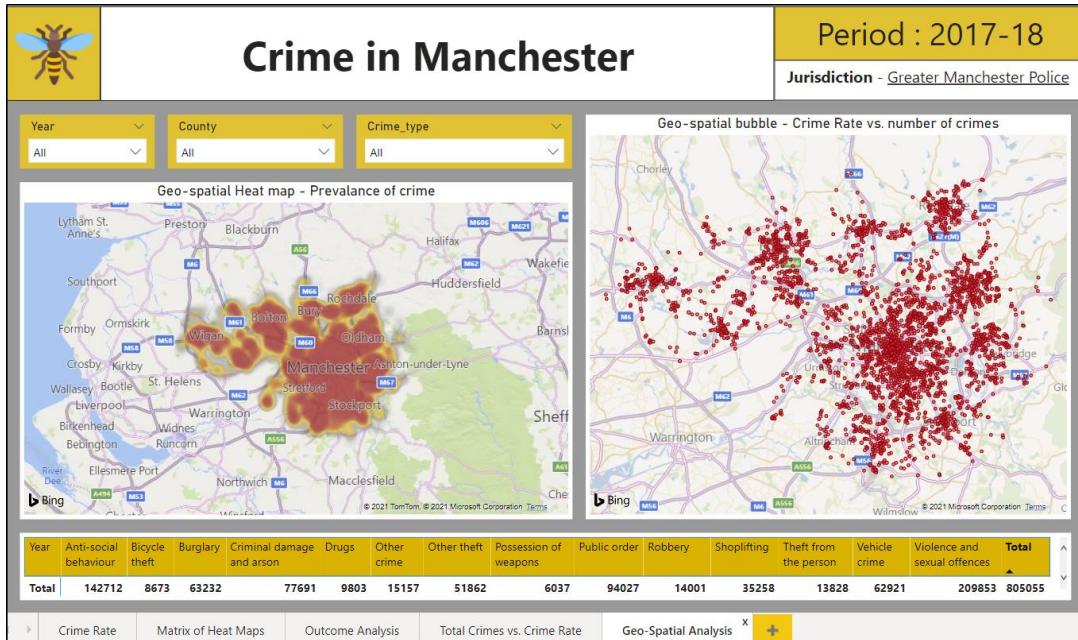


- We could see the total crimes for Violence and sexual offences from 2017-18 are 55356, while the average crime rate here was 70.01.

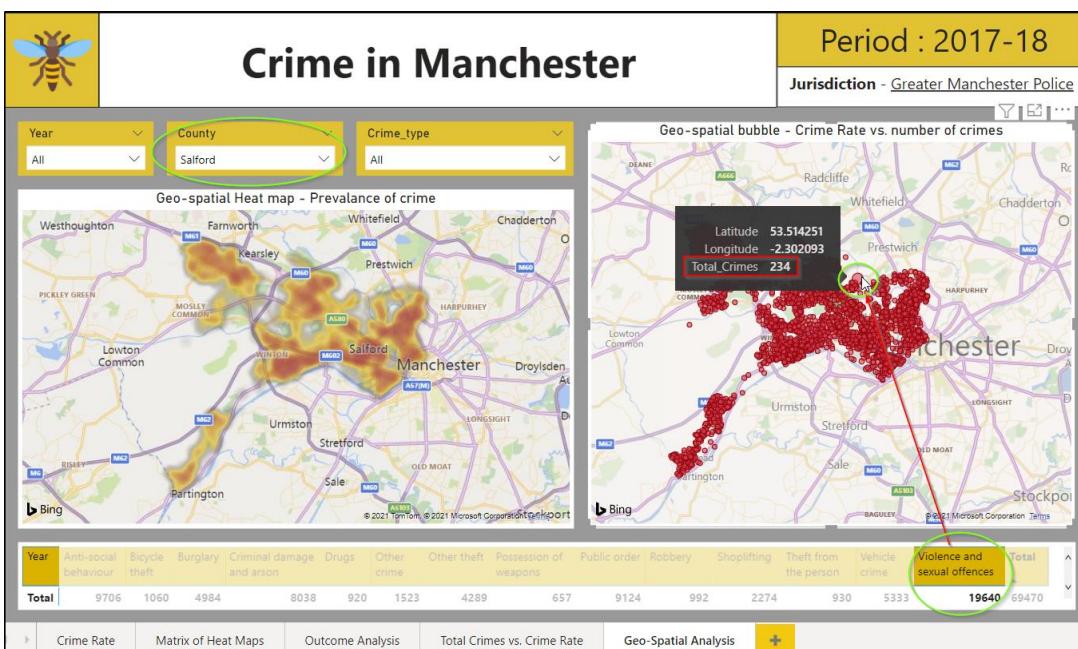


- **Geo-spatial analysis –**

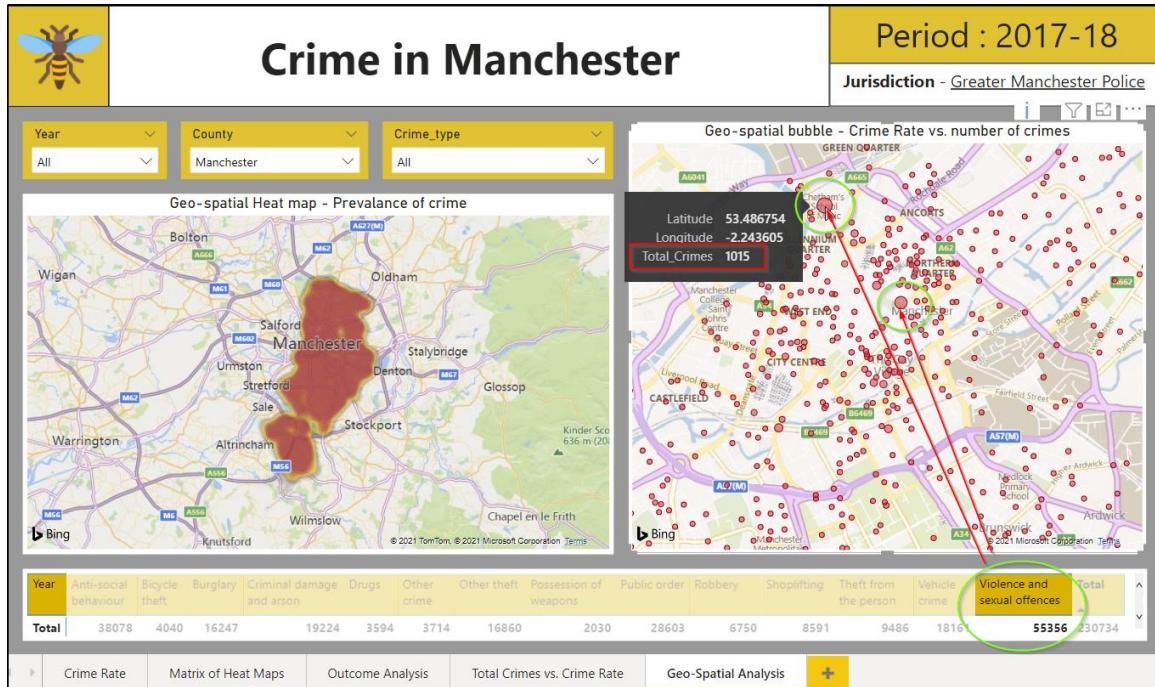
- Two charts were created in terms of maps using Power BI – Heat map and bubble chart.
- On the left, we see a heat map covering the Manchester in red which means crimes are dense in those areas, and along with a yellow border which means the crimes are fairly less in those areas.
- We could filter the necessary Year, County or even Crime type to bring out more meaningful insights.
- The size of the bubble in the bubble map is determined by the number of crimes in that area.



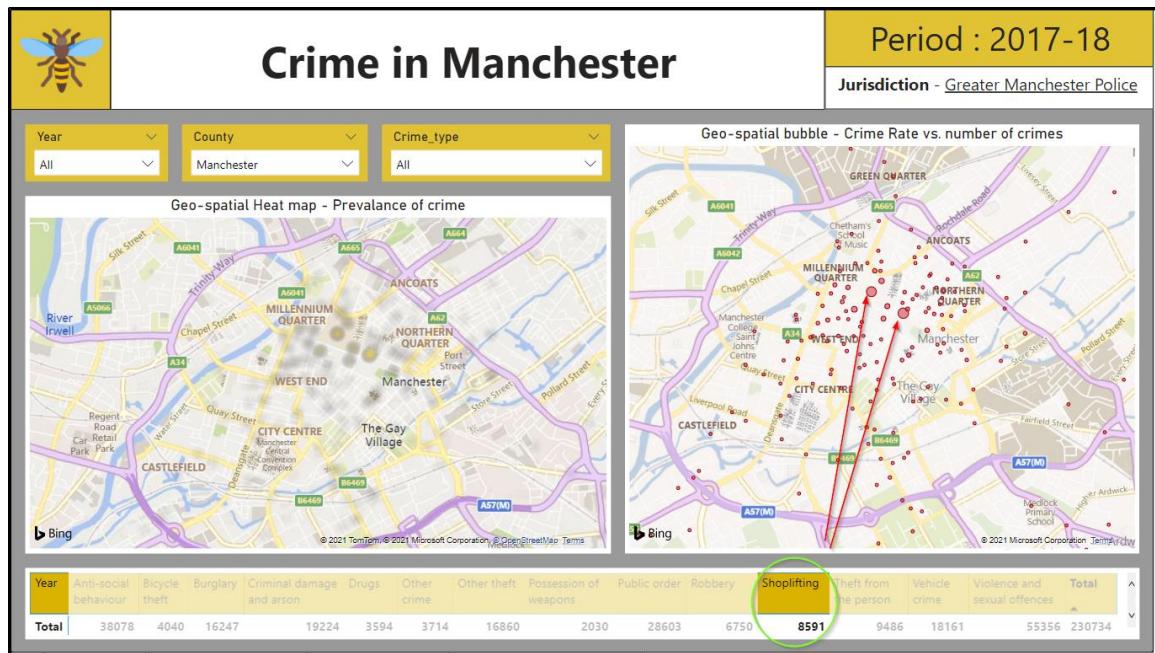
- When we filter for Salford, and Violence and Sexual crimes, we could see the total in the bottom table as 19640, and they are occurring as shown in bubbles below.
- The frequency is more in one area especially, which is highlighted below where in one place alone there are 234 violence/sexual offences. We could highlight this with area and add tool tip to bring out even more information from this visualization, but for the sake of simplicity, I didn't add many tooltips.



- Similarly, in Manchester county, there are 55356 crimes from 2017-18, where there are 2 hotspots for violence and sexual offences, which are shown as below. The total crimes occurring here are more than 1000.

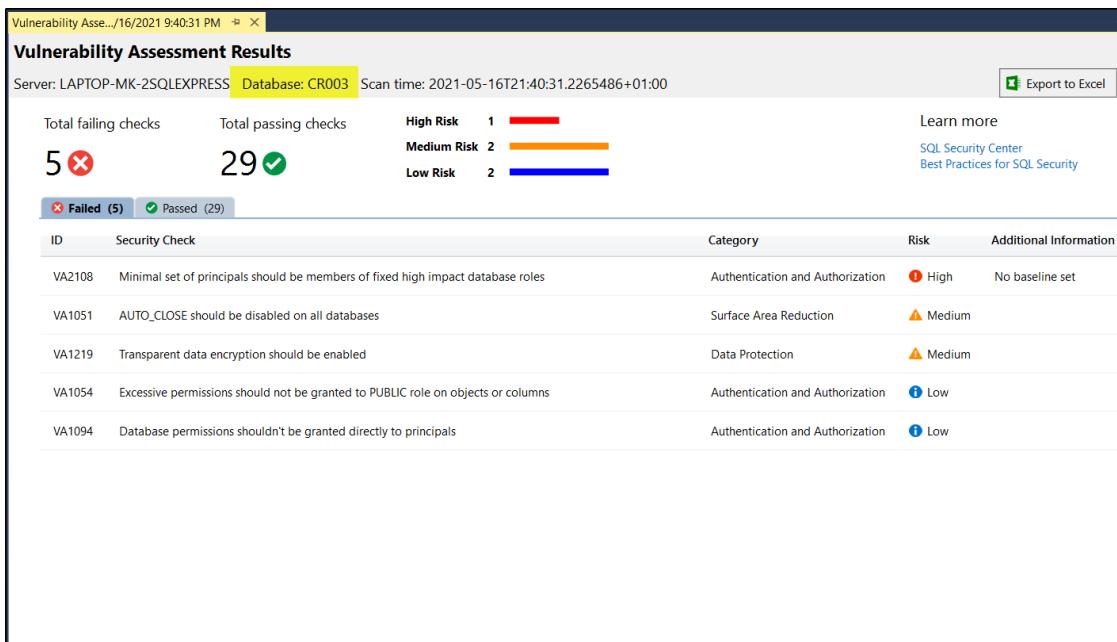


- Shoplifting in Manchester is also prominent in 2 places, and those hotspots have been highlighted in the below graph as we could see.



### 3.8. Database Security

- In order to ensure the security in this database, I performed a vulnerability scan/assessment. Below are the results.
- As we could see 5 failing checks here, one advice needed to ensure security and deter ransomware / malware attacks is to rectify these failing checks and to increase the count of passing checks.
- In order to prioritize the rectification process, we could see which failed check is riskier against which category and then we rectify.



### 3.9 Database Backup and Restore Strategy

- A periodical backup plan can be setup, which is done in SQL EXPRESS edition manually, whereas using DEVELOPER edition, we can setup a Maintenance plan in which an automatic scheduler can be setup to perform periodical backup of the database.
- For backup, we may choose either Full or Differential mode, or even overwriting or appending to the original database backup, or perhaps even erase backup can be performed.
- We may backup to either Disk or URL. Usually when we store the data in the cloud, we may sync our cloud with SSMS and store the backup over there to ensure high security and availability.
- In order to check the reliability, we can verify the backup when it is finished, also if an error is thrown, we can enable the backup to continue.
- Depending on our needs, we may choose to set backup compression, so that we optimize our disk / datastore memory.
- Backup can be historically saved with a timestamp, and using any of the timestamp, we can restore a database.

### 3.10 Data Science/Business Intelligence Techniques

- Business Intelligence Techniques using SQL/related tools –
  - DAR Approach (Dashboards, Analysis, Reports), data blending using multiple data sources.
  - Slicing and dicing, to create necessary reports from DB level itself.
  - Analyse the types of dimensions like slowly changing dimensions for example, and then build or query the model based on the type of metadata.
  - Data warehousing – SQL databases (OLTP – Online Transaction Processing) and SQL data warehouses (OLAP – Online Analytical Processing).

- BI techniques can be applied at Time Intelligence level – using Power BI or even SQL directly, time intelligence can be a powerful feature and can provide excellent key performance indicators which can be projected as well as forecasted meaningfully. For example:
    - Year-to-Date which has the aggregated expression from the beginning of the current year.
    - Year-on-Year which can be used to calculate cumulative growth.
    - Moving annual totals / averages – which can be aggregated cumulative data from this time last year, etc.
  - Additional BI techniques include data mining, model visualization, ETL like in SAP (Extract, Transform, Load), or ELT like in Snowflake (Extract, Load, Transform), etc.
- Data Science Techniques –
  - SQL can be connected to multiple Data Science based tools like Python, R etc. In Python, there is a standard library known as SQLite3 which needs to be imported into the interface and SQL commands can be given to explore the data.
  - SQL can be connected to R studio using 3 packages – RODBC, DBI, ODBC
  - Additional Data Science techniques include predictive modelling, statistical analysis, text mining or even Sentiment analysis, etc.
  - Other techniques using algorithms can also be linked with SQL data like classification, regression, etc.

### 3.11 Data Privacy, Ethical and legal issues

- [13] SSMS contains features that are internet-enabled, so they can collect and share anonymous feature usage and diagnostic data to Microsoft.
- [14] In order to comply with GDPR (General Data Protection Regulation policy), Microsoft recommends GDPR compliance using 4 steps – Discover, Manage, Protect & Report.
  - In Discover phase, we discover sensitive data – which table, which database, which object has it – everything needs to be documented. Based on the levels of sensitivity of the personal data, these fields need to be classified.
  - In Manage phase, once all the data locations are documented, we proceed to understand which personal data is accessed by whom and then limit it.
  - In Protect phase, once the present security practices are understood, the efforts to protect the data begin. The goal is to reduce risk and reduce the impact to data because of security and monitoring. Various encryption methods can be used in this phase in order to protect the data.
  - In Report phase, 2 principles apply – transparency and data retention of all activities about personal data (metadata only).
- [14] In order to meet the GDPR criteria, an organization must strive to meet the ‘Data protection impact assessment’ which is prerequisite of the GDPR (**GDPR article 35**).
- [14] From SQL point of view – some of the best recommendations / practices are:
  - Windows authentication over SQL server authentication to enable centralized management via Active directory.
  - To use discrete accounts to authenticate users and applications which minimizes risks of malicious activity which can include SQL injection attacks.
- There are ways of protecting data within SSMS. In order to do this, we must discover the data first and later classify it into levels of confidentiality.

The screenshot shows the 'Data Classification - CR003' interface. At the top, there are buttons for Save, Add Classification, View Report, and Refresh. A message bar at the top says 'There are pending classification updates. Please save.' Below this, a message box says 'No classification recommendations.' A table titled '2 classified columns' lists two entries:

Schema	Table	Column	Information Type	Sensitivity Label
dbCrime	Streets	Falls_within	National ID	Highly Confidential - GDPR
dbCrime	Streets	Reported_by	Name	Highly Confidential - GDPR

**Figure 3.1 – Data Discovery & Classification**

The screenshot shows the 'SQL Data Classification Report' for server LAPTOP-MK-2\SQLEXPRESS and database CR003. It was generated on 5/17/2021 at 12:08:05 AM. Key statistics shown are:

- Classified columns: 2 / 703
- Tables containing sensitive data: 1 / 12
- Unique information types: 2

Two donut charts are provided:

- Label distribution:** Shows 2 segments in green, representing 'Highly Confidential - GDPR'.
- Information Type distribution:** Shows 1 segment in blue (top) and 1 segment in green (bottom), representing 'Name' and 'National ID' respectively.

A detailed table at the bottom lists the classified columns:

Schema	Table	Column	Information Type	Sensitivity Label
dbCrime	Streets	Reported_by	Name	Highly Confidential - GDPR
dbCrime	Streets	Falls_within	National ID	Highly Confidential - GDPR

**Figure 2.2 – Data Classification Report**

### 3.12 Conclusion

- Both crime rate and number of crimes have slightly reduced from 2017 to 2018.
- The total average crime rate (per 100K) is 71.20.
- The highest crime rate is for Shoplifting, theft from the person, anti-social behaviour, and violence and sexual offences in that order.
- Under the jurisdiction of Greater Manchester Police, the overall crime rate has gone down from 2017 to 2018 as we can see the average reducing from 71.50 to 70.89.
- By county, Blackburn with Darwen, Cheshire East have notorious places for criminal activity but they reduced slightly from 2017 to 2018, whereas Warrington and West Lancashire have increased crime rate from 2017 to 2018.
- As it is the Northern capital of England and being a busy area, Manchester has the highest the number of crimes, but reduced very slightly from 2017 to 2018.

- Out of the 805855 cases, around 380,000 cases have no suspect identified after the investigation is complete, and around 144,000 are unable to prosecute suspect which means the evidence was inconclusive.
- We could see the total crimes for Violence and sexual offences from 2017-18 are 55356, while the average crime rate here was 70.01.
- Violence and Sexual crimes in Salford were 19640.
- 55356 violence and sexual crimes happened from 2017-18 in Manchester area, and there are 2 hotspots for violence and sexual offences. The total crimes occurring here are more than 1000 especially in these 2 areas in the city centre.
- Shoplifting in Manchester is also prominent in 2 hotspots.

## References

- [1] <https://www.younglives.org.uk/>
- [2] "SQL: QuickStart Guide – The Simplified Beginner's Guide To SQL" By Clydebank Technology
- [3] "SQL Practice Problems: 57 beginning, intermediate, and advanced challenges for you to solve using a 'learn-by-doing'" approach by Sylvia Moestl Vasilik
- [4] <https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot?view=sql-server-ver15>
- [5] <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>
- [6] "The Data Warehouse Toolkit" By Ralph Kimball and Margy Ross
- [7] "T-SQL Fundamentals Third Edition" by Itzik Ben-Gunn
- [8] <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>
- [9] <https://docs.microsoft.com/en-us/sql/reporting-services/tools/reporting-services-in-sql-server-management-studio-ssrs?view=sql-server-ver15>
- [10] <https://www.police.uk/pu/your-area/greater-manchester-police/performance/compare-your-area/?tc=MC04>
- [11] <https://docs.microsoft.com/en-us/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-ver15>
- [12] <https://docs.microsoft.com/en-us/power-bi/admin/service-admin-rls>
- [13] <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-telemetry-ssms?view=sql-server-ver15>
- [14] <http://download.microsoft.com/documents/en-gb/microsoft-sql-and-the-gdpr.pdf>
- [15] <https://powerbi.microsoft.com/en-us/blog/power-bi-gdpr-whitepaper-is-now-available/>

## Appendix A - Abbreviations

- Ch / ch\_ - Child
- DB – Database
- DDL – Data Definition Language
- DML – Data Manipulation Language
- ET – Ethiopia
- Hh / hh\_ - Household
- IN – India
- MD – Master Data
- OLAP – Online Analytical Processing
- OLTP – Online Transactional Processing
- PE - Peru
- SP – Stored Procedure
- TBL – Table
- TD – Transactional Data
- TRG – Trigger
- UDF – User-Defined Function
- UN – United Nations
- VN - Vietnam
- YL – Young Lives (foundation)

## Appendix B - Formulae

- Crime Rate = (Number of crimes / Total Number of population) \* 100,000