

## **Abstract:**

In this report, I have designed a Child Wellbeing database for childhood poverty in four countries namely Ethiopia, Peru, Vietnam and India which have supposedly been plagued by Poverty. Using data from the Young Lives study which began in 2002 through 2016 involving 12,000 children, this study aims to find relationship between poverty and Health alongside Education with the Wealth Index of all countries put into consideration.

I also have designed a database focused on the education sector of Vietnam involving 8740 children. This database consists of 3 different sections :- The 8740 children, 52 schools and 220 classes. The results have helped to draw up relationships using the gender, Math and English grades, household contributions to children education. Data on the individual schools and teachers is provided as well for more research.

I also have the Crimes and LSOA database which focuses on the street crimes data in Greater Manchester from January 2017 till December 2018 and a population data for 2018. I have also employed the use of QGIS for visualizing the vehicle crimes and anti-social behaviour in Salford.

## **Introduction:**

According to the [Ethiopia Poverty Assessment](#), poverty is at 24% in 2016 which marks the end of the Young Lives survey. As at 2000, Ethiopia had one of the highest poverty rates in the world and recent studies have shown a great declination over decades.

In India, as at 2016, 270 million Indians were counted poor, signifying 1 in 5 people being poor. With 61% of people not having access to electricity, it was also found that the poor owned fewer assets such as cars, mobile phones, televisions etc and they spend more on food, fuel and light.

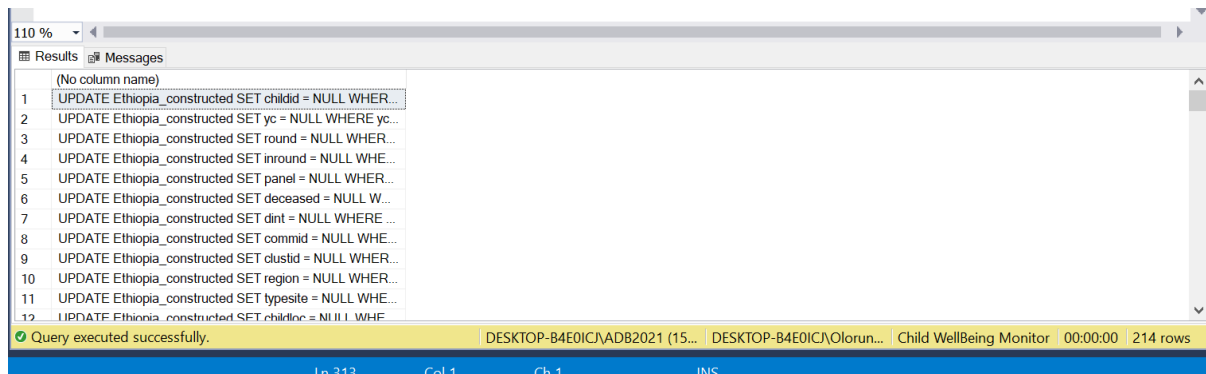
## **TASK 1.**

From the Young Lives dataset available, I downloaded the .tab format of constructed files as it was easier to work with considering it had all 5 rounds together. I approached both methods of importing the .tab file by importing directly to SSMS and taking back to Excel. Going by Excel, I noticed it produced a few empty spaces thereby distorting the

records but came with better datatypes. On using the .tab format, it all came out as varchar but I decided to go with it to have more available data to work with.

On importing, I NULLED the constructed tables before splitting into the various tables to help my analysis using:

```
SELECT 'UPDATE Ethiopia_constructed SET ' + name + ' = NULL WHERE ' + name + ' = ''';'
FROM syscolumns
WHERE id = object_id('ethiopia_countructed')
AND isnullable = 1;
```



The result yielded and 'UPDATE' for all columns available which I copied out and executed, after which I replaced 'ethiopia\_constructed' with all other countries and executed same way.

I made sure to pick out the more relevant columns from the 200+ to draw more conclusive results to create my tables:

```
CREATE SCHEMA Ethiopia
```

```
-- Create table for Child's location
```

```
SELECT childid, yc, round, inround, panel, deceased, dint, region, typesite, childloc
INTO Ethiopia.childlocation
FROM Ethiopia_constructed
```

```
-- Create table for child's bio
```

```
SELECT childid, round, chsex, chethnic, agemon
INTO Ethiopia.ChildBio
FROM Ethiopia_constructed
```

```
-- Create table for child's health
```

```
SELECT childid, round, bmi, zbfa, underweight, stunting, thinness, bwght, chhprob,
chdisability, chdiscale, chhrel, chhealth, cladder
INTO Ethiopia.ChildHealth
FROM ethiopia_constructed
```

```
-- Create table for ChildBirth and vaccination
```

```
SELECT childid, round, numante, delivery, tetanus, bcg, measles, dpt, polio, hib
INTO Ethiopia.ChildBirthAndVaccination
FROM Ethiopia_constructed
```

```
-- create table for Child injury through rounds
```

```
SELECT childid, round, chmightdie, chillness, chinjury
INTO Ethiopia.ChildInjuryThroughRounds
```

```

FROM Ethiopia_constructed

-- Create table for Child drug activity and sexual awareness
SELECT childid, round, chsmoke, chalcohol, chrephealth1, chrephealth2, chrephealth3,
chrephealth4
INTO Ethiopia.ChildDrugActivityAndSexualAwareness
FROM Ethiopia_constructed

-- Create table for Child's ducation
SELECT childid, round, htask, hwork, hschool, preprim, engrade, entype, hghgrade,
levlwrit, levread, literate
INTO Ethiopia.ChildEducation
FROM Ethiopia_constructed

-- Create Household Education table
SELECT childid, round, levread, careid, caredu, carecantread, dadid, dadedu, dadlive,
dadcantread, momid, momedu, momlive, momcantread, headid, headedu, headrel
INTO Ethiopia.HouseholdEducation
FROM Ethiopia_constructed

-- Create table for indexes
SELECT childid, round, wi_new, hq_new, sv_new, cd_new
INTO Ethiopia.Indexes
FROM Ethiopia_constructed

-- Create table for Access to resources
SELECT childid, round, drwaterq_new, toiletq_new, elecq_new, cookingq_new, foodsec,
credit
INTO Ethiopia.HouseholdResources
FROM Ethiopia_constructed

-- Create Household beneficiary of programs
SELECT childid, round, psnp_ds, psnp_pw, othprog, hep, resettled, eap
INTO Ethiopia.HouseholdBeneficiary
FROM Ethiopia_constructed

-- Create table for shocks
SELECT childid, round, shecon5, shecon6, shecon13, shecon14, shhouse1, shhouse2,
shhouse3, shfam1, shfam2, shfam7
INTO Ethiopia.HouseholdShocks
FROM Ethiopia_constructed


CREATE SCHEMA India

-- Create table for Child's location
SELECT childid, yc, round, inround, panel12345, deceased, dint, region, typesite,
childloc
INTO india.childlocation
FROM india_constructed

-- Create table for child's bio
SELECT childid, round, chsex, chethnic, agemon AS Age
INTO india.ChildBio
FROM india_constructed

-- Create table for child's health
SELECT childid, round, bmi, zbfa, underweight, stunting, thinness, bwght, chhprob,
chdisability, chdiscale, chhrel, chhealth, cladder
INTO india.ChildHealth
FROM india_constructed

```

```

-- Create table for ChildBirth and vaccination
SELECT childid, round, numante, delivery, tetanus, bcg, measles, dpt, polio, hib
INTO india.ChildBirthAndVaccination
FROM india_constructed

-- create table for Child injury through rounds
SELECT childid, round, chmightdie, chillness, chinjury
INTO india.ChildInjuryThroughRounds
FROM india_constructed

-- Create table for Child drug activity and sexual awareness
SELECT childid, round, chsmoke, chalcohol, chrephealth1, chrephealth2, chrephealth3,
chrephealth4
INTO india.ChildDrugActivityAndSexualAwareness
FROM india_constructed

-- Create table for Child's education
SELECT childid, round, htask, hwork, hschool, preprim, engrade, entype, hghgrade,
levlwrit, levread, literate
INTO india.ChildEducation
FROM india_constructed

-- Create Household Education table
SELECT childid, round, levread, careid, caredu, carecantread, dadid, dadedu, dadlive,
dadcantread, momid, momedu, momlive, momcantread, headid, headedu, headrel
INTO india.HouseholdEducation
FROM india_constructed

-- Create table for indexes
SELECT childid, round, wi, hq, sv, cd
INTO india.Indexes
FROM india_constructed

-- Create table for Access to resources
SELECT childid, round, drwaterq, toiletq , elecq , cookingq , foodsec, credit
INTO india.HouseholdResources
FROM india_constructed

-- Create Household beneficiary of programs
SELECT childid, round, pds,nregs,nregs_allow, nregs_work
INTO india.HouseholdBeneficiary
FROM india_constructed

-- Create table for shocks
SELECT childid, round, shecon5, shecon6, shecon13, shecon14, shhouse1, shhouse2,
shhouse3, shfam1, shfam2, shfam7
INTO india.HouseholdShocks
FROM india_constructed

CREATE SCHEMA peru

-- Create table for Child's location
SELECT childid, yc, round, inround, panel12345, deceased, dint, region, typesite,
childloc
INTO peru.childlocation
FROM peru_constructed

-- Create table for child's bio
SELECT childid, round, chsex, chethnic, agemon AS Age
INTO peru.ChildBio
FROM peru_constructed

```

```

-- Create table for child's health
SELECT childid, round, bmi, zbfa, underweight, stunting, thinness, bwght, chhprob,
chdisability, chdiscale, chhrel, chhealth, cladder
INTO peru.ChildHealth
FROM peru_constructed

-- Create table for ChildBirth and vaccination
SELECT childid, round, numante, delivery, tetanus, bcg, measles, dpt, polio, hib
INTO peru.ChildBirthAndVaccination
FROM peru_constructed

-- create table for Child injury through rounds
SELECT childid, round, chmightdie, chillness, chinjury
INTO peru.ChildInjuryThroughRounds
FROM peru_constructed

-- Create table for Child drug activity and sexual awareness
SELECT childid, round, chsmoke, chalcohol, chrephealth1, chrephealth2, chrephealth3,
chrephealth4
INTO peru.ChildDrugActivityAndSexualAwareness
FROM peru_constructed

-- Create table for Child's education
SELECT childid, round, htask, hwork, hschool, preprim, engrade, entype, enrol,
levlwrit, levread, literate
INTO peru.ChildEducation
FROM peru_constructed

-- Create Household Education table
SELECT childid, round, levread, careid, caredu, carecantread, dadid, dadedu, dadlive,
dadcantread, momid, momedu, momlive, momcantread, headid, headedudu, headrel
INTO peru.HouseholdEducation
FROM peru_constructed

-- Create table for indexes
SELECT childid, round, wi, hq, sv, cd
INTO peru.Indexes
FROM peru_constructed

-- Create table for Access to resources
SELECT childid, round, drwaterq, toiletq , elecq , cookingq , foodsec, credit
INTO peru.HouseholdResources
FROM peru_constructed

-- Create Household beneficiary of programs
SELECT childid, round, bonograt, sisgrat_y1, minsa_y1, insur_y1, beca_y1, projoven_y1
INTO peru.HouseholdBeneficiary
FROM peru_constructed

-- Create table for shocks
SELECT childid, round, shecon5, shecon6, shecon14, shhouse1, shhouse2, shhouse3,
shfam1, shfam2, shfam7
INTO peru.HouseholdShocks
FROM peru_constructed

CREATE SCHEMA Vietnam

-- Create table for Child's location
SELECT childid, yc, round, inround, panel12345, deceased, dint, region, typesite,
childloc
INTO vietnam.childlocation

```

```

FROM vietnam_constructed

-- Create table for child's bio
SELECT childid, round, chsex, chethnic, agemon AS Age
INTO vietnam.ChildBio
FROM vietnam_constructed

-- Create table for child's health
SELECT childid, round, bmi, zbfa, underweight, stunting, thinness, bwght, chhprob,
chdisability, chdiscale, chhrel, chhealth, cladder
INTO vietnam.ChildHealth
FROM vietnam_constructed

-- Create table for ChildBirth and vaccination
SELECT childid, round, numante, delivery, tetanus, bcg, measles, dpt, polio, hib
INTO vietnam.ChildBirthAndVaccination
FROM vietnam_constructed

-- create table for Child injury through rounds
SELECT childid, round, chmightdie, chillness, chinjury
INTO vietnam.ChildInjuryThroughRounds
FROM vietnam_constructed

-- Create table for Child drug activity and sexual awareness
SELECT childid, round, chsmoke, chalcohol, chrephealth1, chrephealth2, chrephealth3,
chrephealth4
INTO vietnam.ChildDrugActivityAndSexualAwareness
FROM vietnam_constructed

-- Create table for Child's education
SELECT childid, round, htask, hwork, hschool, preprim, engrade, entype, enrol,
levlwrit, levread, literate
INTO vietnam.ChildEducation
FROM vietnam_constructed

-- Create Household Education table
SELECT childid, round, levread, careid, caredu, carecantread, dadid, dadedu, dadlive,
dadcantread, momid, momedu, momlive, momcantread, headid, headedu, headrel
INTO vietnam.HouseholdEducation
FROM vietnam_constructed

-- Create table for indexes
SELECT childid, round, wi_new, hq_new, sv_new, cd_new
INTO vietnam.Indexes
FROM vietnam_constructed

-- Create table for Access to resources
SELECT childid, round, drwaterq_new, toiletq_new , elecq_new , cookingq_new ,
foodsec, credit
INTO vietnam.HouseholdResources
FROM vietnam_constructed

-- Create Household beneficiary of programs
SELECT childid, round, molisa06, molisa09, molisa11, molisa12, molisa13, molisa14,
molisa15, molisa16
INTO vietnam.HouseholdBeneficiary
FROM vietnam_constructed

-- Create table for shocks
SELECT childid, round, shecon5, shecon6, shecon14, shhouse1, shhouse2, shhouse3,
shfam1, shfam2, shfam7
INTO vietnam.HouseholdShocks

```

```
FROM vietnam_constructed
```

I also individually created composite keys using childid and round after changing from NULLs to NOT NULLs using scripts. I first ran the Ethiopia tables then used the Find and Replace to replace every 'ethiopia' with india, peru and Vietnam.

```
ALTER TABLE ethiopia_constructed  
ADD CONSTRAINT PK_ethiopia_constructed  
PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childbio  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childbirthandvaccination  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childdrugactivityandsexualawareness  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childeducation  
ADD PRIMARY KEY (childid, round)
```

```
ALTER TABLE ethiopia.childhealth  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childinjurythroughrounds  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.childlocation  
ADD PRIMARY KEY (childid, round)
```

```
ALTER TABLE ethiopia.householdbeneficiary  
ADD PRIMARY KEY (childid,round)
```

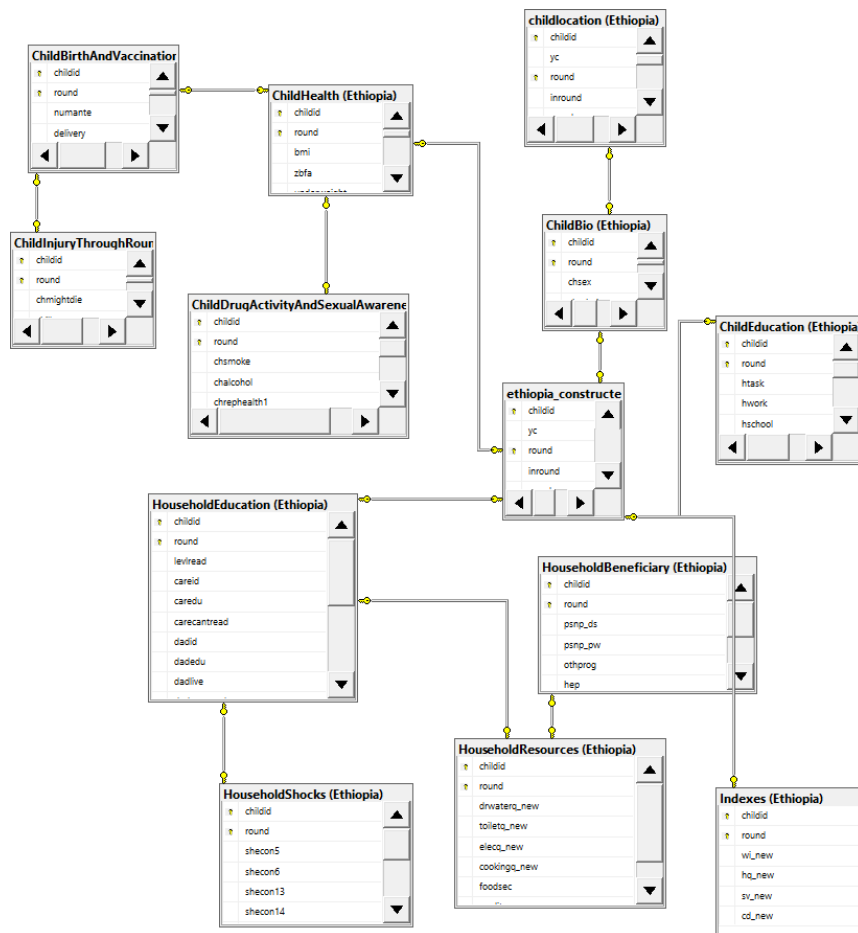
```
ALTER TABLE ethiopia.householdeducation  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.householdresources  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.householdshocks  
ADD PRIMARY KEY (childid,round)
```

```
ALTER TABLE ethiopia.indexes  
ADD PRIMARY KEY (childid,round)
```

With this, I was able to create perfect diagrams for each country, using Ethiopia as an example here. The others can be found in the Diagrams folder of my database.



Also, I had to change the datatypes of certain columns like bmi, zbfa,wi\_new etc using stored procedures which I shall show within the report.

Using Find and Replace, I replaced Ethiopia with every country and editing 'wi\_new, hq\_new, sv\_new, cd\_new' to suit each country's columns.

```
ALTER TABLE ethiopia_constructed ALTER COLUMN numante float
ALTER TABLE ethiopia_constructed ALTER COLUMN hschool float
ALTER TABLE ethiopia_constructed ALTER COLUMN htask float
ALTER TABLE ethiopia_constructed ALTER COLUMN hwork float
ALTER TABLE ethiopia_constructed ALTER COLUMN hschool float
ALTER TABLE ethiopia_constructed ALTER COLUMN bmi float
ALTER TABLE ethiopia_constructed ALTER COLUMN zbfa float
ALTER TABLE ethiopia_constructed ALTER COLUMN bwght float
ALTER TABLE ethiopia_constructed ALTER COLUMN wi_new float
ALTER TABLE ethiopia_constructed ALTER COLUMN hq_new float
ALTER TABLE ethiopia_constructed ALTER COLUMN sv_new float
ALTER TABLE ethiopia_constructed ALTER COLUMN cd_new float
```

I also created a set of queries to analyse each country one by one to further help with creating my Views.



```

CREATE VIEW Antenatals AS
SELECT COUNT(childid) AS 'Total Number', 'Peru' AS 'Country'
FROM peru.ChildBirthAndVaccination
WHERE numante IS NOT NULL AND numante >= 0
UNION
SELECT COUNT(childid) AS 'Total Number', 'Ethiopia' AS 'Country'
FROM ethiopia.childbirthandvaccination
WHERE numante IS NOT NULL AND numante >= 1
UNION
SELECT COUNT(childid) AS 'Total Number', 'India' AS 'Country'
FROM india.ChildBirthAndVaccination
WHERE numante IS NOT NULL AND numante >= 0
UNION
SELECT COUNT(childid) AS 'Total Number', 'Vietnam' AS 'Country'
FROM vietnam.ChildBirthAndVaccination
WHERE numante IS NOT NULL AND numante >= 0

```

```

CREATE VIEW DRUGS AS
SELECT COUNT(childid) AS 'Total Number', 'Ethiopia' AS 'Country'
FROM ethiopia.ChildDrugActivityAndSexualAwareness
WHERE chsmoke IS NOT NULL AND (chsmoke < 3 AND chalcohol = 1)
UNION
SELECT COUNT(childid) AS 'Total Number', 'Peru' AS 'Country'
FROM peru.ChildDrugActivityAndSexualAwareness
WHERE chsmoke IS NOT NULL AND (chsmoke < 3 AND chalcohol = 1)
UNION
SELECT COUNT(childid) AS 'Total Number', 'India' AS 'Country'
FROM india.ChildDrugActivityAndSexualAwareness
WHERE chsmoke IS NOT NULL AND (chsmoke < 3 AND chalcohol = 1)
UNION
SELECT COUNT(childid) AS 'Total Number', 'Vietnam' AS 'Country'
FROM vietnam.ChildDrugActivityAndSexualAwareness
WHERE chsmoke IS NOT NULL AND (chsmoke < 3 AND chalcohol = 1)

```

```

CREATE VIEW SEX_EDUCATION AS
-- Children with no idea about condoms and how STDs can be spread by healthy looking
persons
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Ethiopia' AS 'Country'
FROM ethiopia.ChildDrugActivityAndSexualAwareness
WHERE chrephealth2 IS NOT NULL AND chrephealth3 IS NOT NULL AND chrephealth2 = 0 AND
chrephealth3 =0
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Peru' AS 'Country'
FROM peru.ChildDrugActivityAndSexualAwareness
WHERE chrephealth2 IS NOT NULL AND chrephealth3 IS NOT NULL AND chrephealth2 = 0
AND chrephealth3 =0
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'India' AS 'Country'
FROM india.ChildDrugActivityAndSexualAwareness
WHERE chrephealth2 IS NOT NULL AND chrephealth3 IS NOT NULL AND chrephealth2 = 0
AND chrephealth3 =0
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Vietnam' AS 'Country'
FROM vietnam.ChildDrugActivityAndSexualAwareness
WHERE chrephealth2 IS NOT NULL AND chrephealth3 IS NOT NULL AND chrephealth2 = 0
AND chrephealth3 =0

```

```

CREATE VIEW Pre_Primary_Education AS
--Number of studnets that did not attend
SELECT COUNT(childid) AS 'Number of Children' , 'Ethiopia' AS 'Country'

```

```

FROM ethiopia.ChildEducation
WHERE preprim = 0 AND round = 1 AND preprim IS NOT NULL
UNION
SELECT COUNT(childid) AS 'Number of Children' , 'India' AS 'Country'
FROM india.ChildEducation
WHERE preprim = 0 AND round = 1 AND preprim IS NOT NULL
UNION
SELECT COUNT(childid) AS 'Number of Children' , 'Peru' AS 'Country'
FROM peru.ChildEducation
WHERE preprim = 0 AND round = 1 AND preprim IS NOT NULL
UNION
SELECT COUNT(childid) AS 'Number of Children' , 'Vietnam' AS 'Country'
FROM vietnam.ChildEducation
WHERE preprim = 0 AND round = 1 AND preprim IS NOT NULL

CREATE VIEW [Permanent_Disability_Scale] AS
-- children with permanent disability who are unable to perform certain duties well
enough
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Ethiopia' AS 'Country'
FROM ethiopia.ChildHealth
WHERE chdissscale IS NOT NULL AND chdissscale BETWEEN 1 AND 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Peru' AS 'Country'
FROM peru.ChildHealth
WHERE chdissscale IS NOT NULL AND chdissscale BETWEEN 1 AND 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'India' AS 'Country'
FROM india.ChildHealth
WHERE chdissscale IS NOT NULL AND chdissscale BETWEEN 1 AND 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Vietnam' AS 'Country'
FROM vietnam.ChildHealth
WHERE chdissscale IS NOT NULL AND chdissscale BETWEEN 1 AND 5

CREATE VIEW [Severely_Thin_Stunted_Underweight] AS
--children who are severely thin, severely stunted and severely underweight
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Ethiopia' AS 'Country'
FROM ethiopia.ChildHealth
WHERE underweight = 2 AND stunting =2 AND thinness = 2
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'India' AS 'Country'
FROM india.ChildHealth
WHERE underweight = 2 AND stunting =2 AND thinness = 2
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Peru' AS 'Country'
FROM peru.ChildHealth
WHERE underweight = 2 AND stunting =2 AND thinness = 2
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Vietnam' AS 'Country'
FROM vietnam.ChildHealth
WHERE underweight = 2 AND stunting =2 AND thinness = 2

CREATE VIEW [Worst_Possible_Lives] AS
-- Worst possible lives
SELECT COUNT (DISTINCT childid) AS 'Number of Children', 'Ethiopia' AS 'Country'
FROM ethiopia.ChildHealth
WHERE cladder = 1
UNION
SELECT COUNT (DISTINCT childid) AS 'Number of Children' , 'India' AS 'Country'
FROM india.ChildHealth
WHERE cladder = 1
UNION
SELECT COUNT (DISTINCT childid) AS 'Number of Children' , 'Peru' AS 'Country'

```

```

FROM peru.ChildHealth
WHERE cladder = 1
UNION
SELECT COUNT (DISTINCT childid) AS 'Number of Children' , 'Vietnam' AS 'Country'
FROM vietnam.ChildHealth
WHERE cladder = 1

```

```

CREATE VIEW [Deceased] AS
SELECT COUNT(DISTINCT childid) AS 'Total', 'Ethiopia' AS 'Country'
FROM ethiopia.childlocation
WHERE deceased IS NOT NULL AND deceased = 1
UNION
SELECT COUNT(DISTINCT childid) AS 'Total', 'India' AS 'Country'
FROM india.childlocation
WHERE deceased IS NOT NULL AND deceased = 1
UNION
SELECT COUNT(DISTINCT childid) AS 'Total', 'Peru' AS 'Country'
FROM peru.childlocation
WHERE deceased IS NOT NULL AND deceased = 1
UNION
SELECT COUNT(DISTINCT childid) AS 'Total' , 'Vietnam' AS 'Country'
FROM vietnam.childlocation
WHERE deceased IS NOT NULL AND deceased = 1

```

```

CREATE VIEW [Programme_Beneficiaries] AS
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdBeneficiary
WHERE '1' IN (psnp_ds, psnp_pw, othprog, hep, resettled, eap) AND eap IS NOT NULL
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'India' AS 'Country'
FROM india.HouseholdBeneficiary
WHERE '1' IN (pds, nregs, nregs_allow, nregs_work, rajiv, sabla, sabla_y1, ikp,
ikp_child) AND round = 5
UNION
SELECT COUNT(childid) AS 'Number of Children', 'Peru' AS 'Country'
FROM peru.HouseholdBeneficiary
WHERE '1' IN (bonograt, sisgrat_y1, minsa_y1, insur_y1, beca_y1, projoven_y1) AND
round = 4
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdBeneficiary
WHERE '1' IN (molisa09, molisa09, molisa11, molisa12, molisa13, molisa14, molisa15,
molisa16)

```

```

CREATE VIEW [Mom and Dad with no education] AS
-- Number of children whose mothers and fathers have no education at all
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdEducation
WHERE dadedu = 0 AND dadedu IS NOT NULL AND momedu = 0 AND momedu IS NOT NULL AND
round = 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'India' AS 'Country'
FROM india.HouseholdEducation
WHERE dadedu = 0 AND dadedu IS NOT NULL AND momedu = 0 AND momedu IS NOT NULL AND
round = 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Peru' AS 'Country'
FROM peru.HouseholdEducation
WHERE dadedu = 0 AND dadedu IS NOT NULL AND momedu = 0 AND momedu IS NOT NULL AND
round = 5
UNION

```

```

SELECT COUNT(DISTINCT childid) AS 'Number of Children', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdEducation
WHERE dadedu = 0 AND dadedu IS NOT NULL AND momedu = 0 AND momedu IS NOT NULL AND
round = 5

```

```

CREATE VIEW [Access to clean water and sanitation, electricity and cooking gas] AS
--access to clean water and sanitation, elctricity and cookng gas
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdResources
WHERE drwaterq_new = 1 AND toiletq_new =1 AND elecq_new =1 AND cookingq_new =1 AND
round = 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'India' AS 'Country'
FROM india.HouseholdResources
WHERE drwaterq = 1 AND toiletq =1 AND elecq =1 AND cookingq =1 AND round = 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Peru' AS 'Country'
FROM peru.HouseholdResources
WHERE drwaterq = 1 AND toiletq =1 AND elecq =1 AND cookingq =1 AND round = 5
UNION
SELECT COUNT(DISTINCT childid) AS 'Total Number', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdResources
WHERE drwaterq_new = 1 AND toiletq_new =1 AND elecq_new =1 AND cookingq_new =1 AND
round = 5

```

```

CREATE VIEW [Loans Obtained since previous round] AS
SELECT COUNT (childid) AS 'Number of Loans', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdResources
WHERE credit = 1
UNION
SELECT COUNT (childid) AS 'Number of Loans', 'India' AS 'Country'
FROM india.HouseholdResources
WHERE credit = 1
UNION
SELECT COUNT (childid) AS 'Number of Loans', 'Peru' AS 'Country'
FROM peru.HouseholdResources
WHERE credit = 1
UNION
SELECT COUNT (childid) AS 'Number of Loans', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdResources
WHERE credit = 1

```

```

CREATE VIEW [Sources of income lost at latest round] AS
SELECT COUNT(childid) AS 'Total', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdShocks
WHERE shecon5 = 1 AND round = 5
UNION
SELECT COUNT(childid) AS 'Total', 'India' AS 'Country'
FROM india.HouseholdShocks
WHERE shecon5 = 1 AND round = 5
UNION
SELECT COUNT(childid) AS 'Total', 'peru' AS 'Country'
FROM peru.HouseholdShocks
WHERE shecon5 = 1 AND round = 5
UNION
SELECT COUNT(childid) AS 'Total', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdShocks
WHERE shecon5 = 1 AND round = 5

```

```

CREATE VIEW [Industrial Actions reported] AS
SELECT COUNT(childid) AS 'Total', 'Ethiopia' AS 'Country'
FROM ethiopia.HouseholdShocks
WHERE shecon6 = 1 AND round = 2
UNION
SELECT COUNT(childid) AS 'Total', 'India' AS 'Country'
FROM india.HouseholdShocks
WHERE shecon6 = 1 AND round = 2
UNION
SELECT COUNT(childid) AS 'Total', 'Peru' AS 'Country'
FROM peru.HouseholdShocks
WHERE shecon6 = 1 AND round = 2
UNION
SELECT COUNT(childid) AS 'Total', 'Vietnam' AS 'Country'
FROM vietnam.HouseholdShocks
WHERE shecon6 = 1 AND round = 2

```

```

CREATE VIEW [DECEASED] AS
SELECT DISTINCT childid, Deceased
FROM Ethiopia.childlocation
WHERE deceased = 1
UNION ALL
SELECT DISTINCT childid, deceased
FROM India.childlocation
WHERE deceased = 1
UNION ALL
SELECT DISTINCT childid, deceased
FROM Peru.childlocation
WHERE deceased = 1
UNION ALL
SELECT DISTINCT childid, deceased
FROM Vietnam.childlocation
WHERE deceased = 1

```

```

CREATE VIEW AS [Reading Level Of All Countries] AS
SELECT COUNT(childid) AS 'Number of children' , levread AS 'Reading Level',
'Ethiopia' AS 'Country'
FROM ethiopia.ChildEducation
WHERE round = 3 AND levread IS NOT NULL
GROUP BY levread
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , levread AS 'Reading Level', 'India'
AS 'Country'
FROM India.ChildEducation
WHERE round = 3 AND levread IS NOT NULL
GROUP BY levread
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , levread AS 'Reading Level', 'Vietnam'
AS 'Country'
FROM Vietnam.ChildEducation
WHERE round = 3 AND levread IS NOT NULL
GROUP BY levread
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , levread AS 'Reading Level', 'Peru' AS
'Country'
FROM Peru.ChildEducation
WHERE round = 3 AND levread IS NOT NULL
GROUP BY levread

```

```

CREATE VIEW [Literacy] AS
SELECT COUNT(childid) AS 'Number of children' , literate AS 'Literacy Level',
'Ethiopia' AS 'Country'
FROM ethiopia.ChildEducation
WHERE round = 3 AND literate IS NOT NULL
GROUP BY literate
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , literate AS 'Literacy Level', 'India'
AS 'Country'
FROM India.ChildEducation
WHERE round = 3 AND literate IS NOT NULL
GROUP BY literate
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , literate AS 'Literacy Level', 'Peru'
AS 'Country'
FROM Peru.ChildEducation
WHERE round = 3 AND literate IS NOT NULL
GROUP BY literate
UNION ALL
SELECT COUNT(childid) AS 'Number of children' , literate AS 'Literacy Level',
'Vietnam' AS 'Country'
FROM Vietnam.ChildEducation
WHERE round = 3 AND literate IS NOT NULL
GROUP BY literate

```

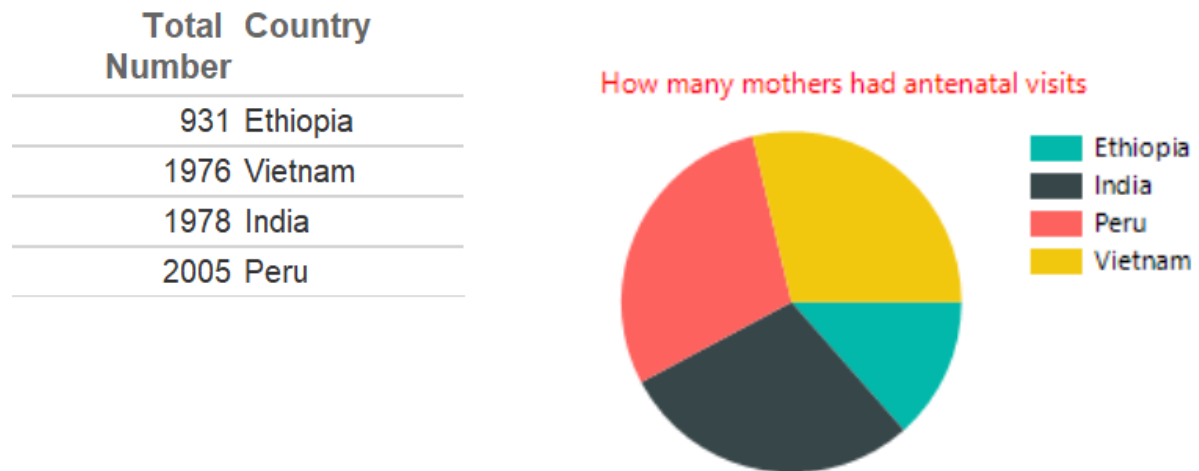
```

CREATE VIEW [Worst Possible Lives] AS
SELECT COUNT (DISTINCT childid) AS 'Total Number' , cladder AS 'Worst Possible Life',
'Ethiopia' AS 'Country'
FROM ethiopia.ChildHealth
WHERE cladder = 1
GROUP BY cladder
UNION
SELECT COUNT (DISTINCT childid) AS 'Total Number' , cladder AS 'Worst Possible Life',
'India' AS 'Country'
FROM India.ChildHealth
WHERE cladder = 1
GROUP BY cladder
UNION
SELECT COUNT (DISTINCT childid) AS 'Total Number' , cladder AS 'Worst Possible Life',
'Peru' AS 'Country'
FROM Peru.ChildHealth
WHERE cladder = 1
GROUP BY cladder
UNION
SELECT COUNT (DISTINCT childid) AS 'Total Number' , cladder AS 'Worst Possible Life',
'Vietnam' AS 'Country'
FROM Vietnam.ChildHealth
WHERE cladder = 1
GROUP BY cladder

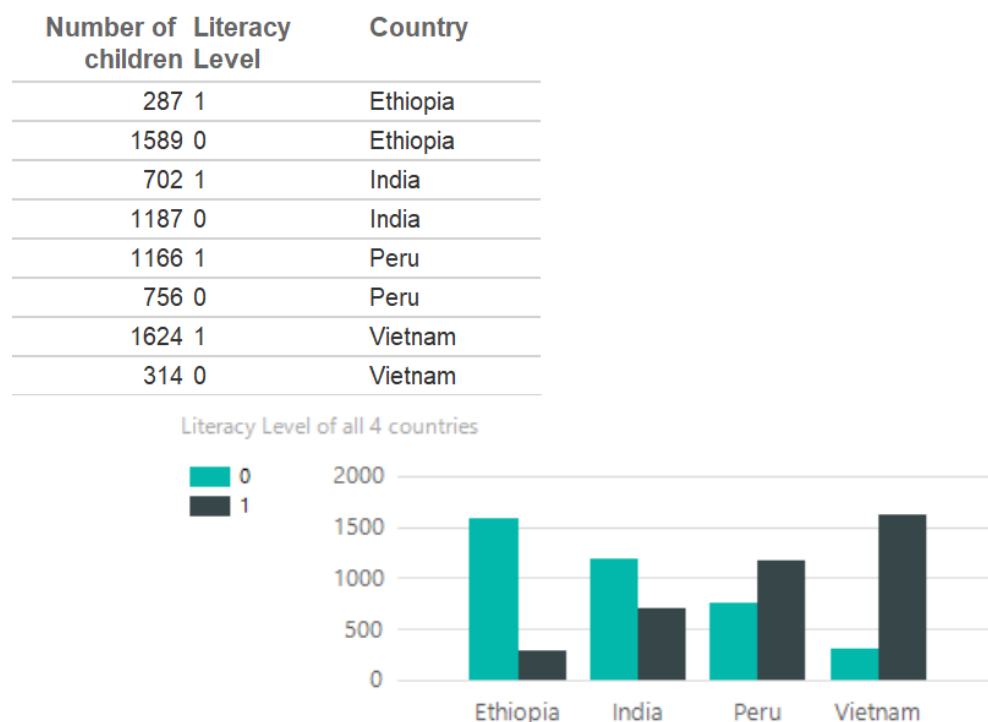
```

As a reporting tool, I made use of SSRS and made five visualizations below:

This chart shows the number of mothers that had antenatal visits with Ethiopia being the least.



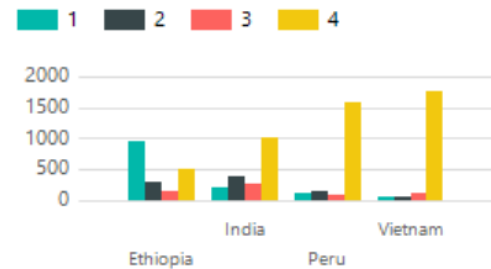
This chart shows the literacy level of the children, where 0 means they can not read and right and 1 means they can read and write. From this, we can deduce an inverse relationship between Ethiopia and Vietnam. Vietnam has the highest number of children that could read and write, while Ethiopia had the lowest.



For the next one, I analysed the child's reading level, where 1 = can't read anything, 2 = reads letter, 3 = reads words, 4 = reads sentences. Here, it can be seen again that Vietnam has the highest amount of children that can read sentences.

Number of children	Reading Level	Country
138	3	Ethiopia
290	2	Ethiopia
949	1	Ethiopia
500	4	Ethiopia
281	3	India
401	2	India
215	1	India
1011	4	India
118	3	Vietnam
46	2	Vietnam
45	1	Vietnam
1742	4	Vietnam
82	3	Peru
152	2	Peru
115	1	Peru
1574	4	Peru

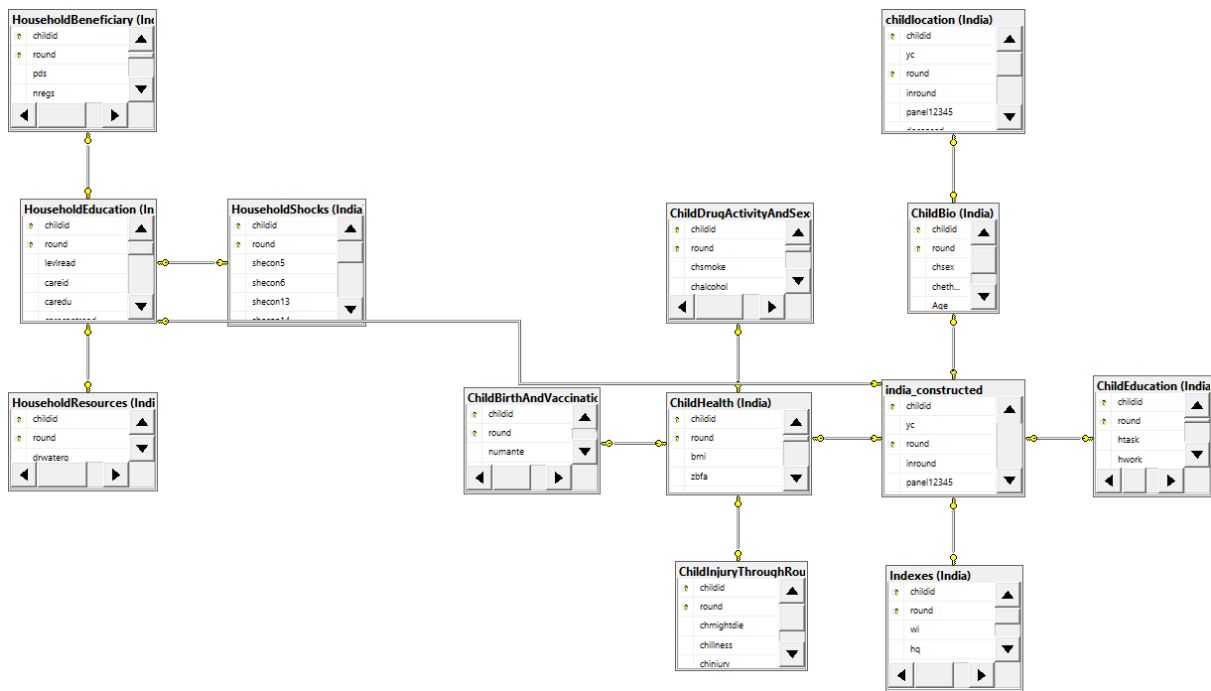
Chart Title



## DESIGN RATIONALE

For Task 1 diagrams, there has to be a relationship between the tables for it to be relational database, so I utilised a one-to-many relationship from my constructed tables to the child health, child bio, household beneficiary and household education tables as it was where I split them from. A one-to-one relationship for the child location and childbio tables. Also, the child health table got a one-to-many relationship as I found the birth and vaccination, injury through rounds and drug activity to be linked to the health of the children. This also applies to the household beneficiary table which I linked to both the Indexes table and the resources table. Below is an image of my India diagram to show this:





As stated earlier, I used a composite key involving both the childid and round columns, due to the fact that childid appeared multiple times based on the multiple rounds. And also, both columns had no reason to be NULL so it was a perfect choice for relationships.

## CONSTRAINT:

First set of constraint I employed was the use of NOT NULLs on the childid and round columns to indicate there had to be an entry as children could not exist without an ID and also without a round.

I would've implemented a UNIQUE Key but due to the data available, it was impossible as childid had to appear 5 times due to the rounds and defeats the purpose of a Unique Key, so I went with the use of Composite Keys involving the childid and round columns

# DATA VALIDATION

I ensured a data validation process was carried out on all the tables and noticed a flaw in the Ethiopia.childeducation table, which there was an incorrectly placed value of 79 for the levread and levlwrit after checking through the data dictionary and there was no such value related.

```
--There was a value outside the specified range, so to find it
SELECT MAX(levread)
FROM Ethiopia.ChildEducation
WHERE round = 3 AND levread != ''
```

```
--To delete the inappropriate value
DELETE FROM Ethiopia.ChildEducation WHERE levread = 79
```

```
-- Deleting inappropriate value
DELETE FROM Ethiopia.ChildEducation WHERE levlwrit = 79
```

## TRANSACTION AND CONCURRENCY CONTROL

Looking at the database, it is seen to be a research database rather than a transactional database which to implement accurate concurrency controls. Going by this, it was difficult to implement simultaneous execution of transactions thereby limiting the use of Commit and Rollback functions

## Error handling

Errors are usually bound to happen in the coding scene and SQL has a functionality for that as well which an example is shown below.

```
BEGIN TRY
    Insert Into Categories(CategoryID, CategoryName, Description, Picture) Values
    (9, 'Test', 'Test Description', 'Test')
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
```

# SECURITY

Considering the Security level, I created only a Login for the Manager named John and as well granted him permissions. I have also created roles for specific people which are the doctor, analyst and clerk and given them permissions only relating to their jobs they have to carry out on the database.

```
CREATE PROCEDURE [Creating Roles] AS
-- Creates amanager role
CREATE ROLE Manager
-- CREATE Clerk role
CREATE ROLE Clerk
-- Create role for Analyst
CREATE ROLE Analyst
-- CREATE role for Headmaster
CREATE ROLE Headmaster
-- Create role doctor
CREATE ROLE Doctor
GO
```

```
CREATE PROCEDURE [Granting Roles] AS
-- Grants creating tables and views to the manager
GRANT CREATE TABLE, CREATE VIEW
TO Manager
-- Grants inserting, updating and deleting to the clerk
GRANT INSERT, UPDATE, DELETE TO Clerk
-- Grant Select role to Analyst
GRANT SELECT TO Analyst
-- Grant SELECT to headmaster on deducation tables
GRANT SELECT ON ethiopia.childededucation TO Headmaster
GRANT SELECT ON ethiopia.childededucation TO Headmaster
GRANT SELECT ON peru.childededucation TO Headmaster
GRANT SELECT ON vietnam.childededucation TO Headmaster
--Grant roles for doctor
GRANT SELECT ON ethiopia.childhealth TO Doctor
GRANT SELECT ON ethiopia.childhealth TO Doctor
GRANT SELECT ON peru.childhealth TO Doctor
GRANT SELECT ON vietnam.childhealth TO Doctor
GO
```

```
CREATE PROCEDURE [Logins] AS
CREATE LOGIN Manager WITH PASSWORD = 'Childmonitor'
GO
```

```
CREATE PROCEDURE[Users] AS
CREATE USER John for login Manager
GO
```

```
CREATE PROCEDURE[Granting Permissions]
GRANT SELECT ON ethiopia_constructed TO John
GRANT SELECT ON ethiopia_constructed TO John
GRANT SELECT ON peru_constructed TO John
GRANT SELECT ON vietnam_constructed TO John
GO
```

## TASK 2.

I went the same route as Task 1 to import the .tab file directly, NULLing the empty rows and splitting the best columns for analysis. During my analysis, I realised there were 3 sections of it to analyse the students itself using their scores, household, possessions, self development and health. Another section was for the school details which had data concerning headteacher details, admission criteria, provisions they offered etc. There is also another section for Classes in each school to show the enrolment, teaching, and their equipment.

-- Create table for Student Bio

```
SELECT UNIQUEID, SCHOOLID, CLASSID, STUDENTID, YLCHILDID, PROVINCE, DISTRICTCODE,
LOCALITY, GENDER, AGE, ETHNICITY STDDINT, ABSENT_DAYS, STDYLCHD, STDLIV, STDLNHGM,
STDMEAL, STEATLNC, STPAYLNC, STTLTSCH
INTO Vietnam_StudentBio
FROM vietnam_wave_1
```

-- Crate table for Parents Education

```
SELECT UNIQUEID, MOM_EDUC, MOM_READ, DAD_EDUC, DAD_READ
INTO Vietnam_ParentsEducation
FROM vietnam_wave_1
```

-- Create table fOr student health

```
SELECT UNIQUEID, STDHLTH1, STDHLTH2, STDHLTH3, STDHLTH4, STDHLTH5, STDHLTH6, STDHLTH0
INTO Vietnam_StudentHealth
FROM vietnam_wave_1
```

-- Create table for Student's Siblings

```
SELECT UNIQUEID, STPLPHM, STSIBOLD, STSIBYNG
INTO Vietnam_StudentSiblings
FROM vietnam_wave_1
```

-- Create table for Student Possessions

```
SELECT UNIQUEID, STNMBOOK, STPLSTDY, STHVAIRC, STHVBIKE, STHVCAR, STHVCBLE, STHVCHR,
STHVCOMP, STHVDESK, STHVDVD, STHVEFAN, STHVFRDG, STHVINTR, STHVLAMP, STHVMCRO,
STHVMTBK, STHVMTEL, STHVRADO, STHVTELE
INTO Vietnam_StudentPossessions
FROM vietnam_wave_1
```

-- Create table for school provisions

```
SELECT DISTINCT SCHOOLID, SCHFAC03, SCHFAC04, SCHFAC05, SCHFAC06, SCHFAC07, SCHFAC08,
SCHFAC09, SCHFAC10, SCHFAC11, SCHFAC12, SCHFAC13, SCHFAC14
INTO Vietnam_SchoolProvisions
FROM vietnam_wave_1
```

-- Create table for Student Grades

```
SELECT UNIQUEID, STRPTCL1, STRPTCL6, STRPTCL10, STAGEENG
INTO Vietnam_StudentGrades
FROM vietnam_wave_1
```

-- Create table for student's possessions in school

```
SELECT UNIQUEID, STITMOW1, STITMOW2, STITMOW3, STITMOW4, STITMOW5, STITMOW6, STITMOW7,
STITMOW8
INTO Vietnam_StudentSchoolPossessions
FROM vietnam_wave_1
```

-- Create table for student self development

```
SELECT UNIQUEID, STBRWBK, STREADCH, STREADFN, STREADLR
```

```

INTO Vietnam_StudentSelfDevelopment
FROM vietnam_wave_1

-- Create table for Household contribution
SELECT UNIQUEID, STPLHLRD, STPLHL01, STPLHL02, STPLHL03, STPLHL04, STPLHL05, STPLHL06
INTO Vietnam_HouseholdContributions
FROM vietnam_wave_1

-- Create table for English conversations
SELECT UNIQUEID, STSPEN01, STSPEN02, STSPEN03, STSPEN04, STSPEN05, STATEN01, STATEN02,
STATEN03, STATEN04
INTO Vietnam_EnglishConversations
FROM vietnam_wave_1

-- Create table for school decisions
SELECT UNIQUEID, STGR1001, STGR1002, STGR1003, STGR1004, STGR1005, STGR1006, STGR1007,
STGR1008, STGR1009, STGR1011, STHGHGRD
INTO Vietnam_SchoolDecisions
FROM vietnam_wave_1

--Create table for English and Maths tests
SELECT UNIQUEID, ENG_TEST, ENG_RAWSCORE, MATH_TEST, MATH_RAWSCORE
INTO Vietnam_EnglishAndMathTests
FROM vietnam_wave_1

-- Create table for Class Enrolment
SELECT DISTINCT SCHOOLID, CLASSID, GRLENRL, BOYENRL, TTLENRL, TGRLENRL, TBOYENRL,
TTTLENRL, ATDTMSY, ATDDFSY, TGMTHENG, SCALLCT, GRPABLT, CLSORD1, CLSORD2
INTO Vietnam_ClassEnrolment
FROM vietnam_wave_1

-- Create table for class equipments
SELECT DISTINCT SCHOOLID, CLASSID, SCAVLB1, SCAVLB2, SCAVLB3, SCAVLB4, SCAVLB5,
SCAVLB6, SCAVLB7, SCAVLB8, SCAVLB9, SCAVLB10
INTO Vietnam_ClassEquipments
FROM vietnam_wave_1

--Create table for Class Teaching
SELECT DISTINCT SCHOOLID, CLASSID, SCPRDDAY, SCLNONPR, SCMENIN, SCMNMtin, SCTXTENG,
SCTXTMTH
INTO Vietnam_ClassTeaching
FROM vietnam_wave_1

-- Create table for Head Teacher Bio
SELECT DISTINCT SCHOOLID, HTDINT, HTAGE, HTSEX, HTETHGRP, HTRELGN, HTMHTNG, HTFRMPRV,
HTCURRLE
INTO Vietnam_HeadTeacherBio
FROM vietnam_wave_1

-- Create table Head Teacher Education and Experience
SELECT DISTINCT SCHOOLID, HTYRSHT, HTLVLEDC, HTLVLTCH, HTEXCTCH
INTO Vietnam_HeadTeacherEducationAndExperience
FROM vietnam_wave_1

-- Create table for Head Teacher School
SELECT DISTINCT SCHOOLID, HTTYPsch, HTLWSGRD, HTHGHGRD, HTNmetst, HTNMSTEN, HTYREST,
HTP135CM, HTPRDIST, HTBOARD, HTSCHAVL,
HTREGTCH, HTNONCMP, HTEXTCUR, HTNOACT, HTNOCMCH, HTNOCMCL
INTO Vietnam_HeadTeacherSchool
FROM vietnam_wave_1

```

```

-- Create table for Teacher Details
SELECT DISTINCT SCHOOLID, HTRQNMST, HTCENMST, HTPTNMST, HTRQENG, HTRQMATH, HTTCHAPP,
HTREWTC1, HTREWTC2, HTREWTC4, HTREWTC5, HTREWTC6, HTREWTC7, HTREWTC8, HTDISTC1,
HTDISTC2, HTDISTC3, HTDISTC4, HTDISTC6, HTDISTC7, HTDISTC8, HTDISTC0
INTO Vietnam_TeacherDetails
FROM vietnam_wave_1

-- Create table for Admission Criteria
SELECT DISTINCT SCHOOLID, HTGNADMN, HTACRAALL, HTACRAETH, HTACRAEXM, HTACRAGND,
HTACRAOTH, HTACRAPAY, HTACRARE
INTO Vietnam_AdmissionCriteria
FROM vietnam_wave_1

-- Create table for Tuition Fees
SELECT DISTINCT SCHOOLID, HTEXM0011, HTEXM021, HTEXM031, HTEXM041, HTEXM051, HTEXM061,
HTEXM001
INTO Vietnam_TuitionFees
FROM vietnam_wave_1

-- Create table for Lunch fees
SELECT DISTINCT SCHOOLID, HTAMPAID2, HTEXM0012, HTEXM022, HTEXM032, HTEXM042,
HTEXM052, HTEXM062, HTEXM002
INTO Vietnam_LunchFees
FROM vietnam_wave_1

-- Create table for extra classes fees
SELECT DISTINCT SCHOOLID, HTAMPAID3, HTEXM013, HTEXM023, HTEXM033, HTEXM043, HTEXM053,
HTEXM063, HTEXM003
INTO Vietnam_ExtraClassesFees
FROM vietnam_wave_1

-- Create table for textbook fees
SELECT DISTINCT SCHOOLID, HTAMPAID4, HTEXM014, HTEXM024, HTEXM034, HTEXM044,
HTEXM054, HTEXM064, HTEXM004
INTO Vietnam_TextbookFees
FROM vietnam_wave_1

-- Create table for Uniforms
SELECT DISTINCT SCHOOLID, HTAMPAID5, HTEXM015, HTEXM025, HTEXM035, HTEXM045, HTEXM055,
HTEXM065, HTEXM005
INTO Vietnam_UniformFees
FROM vietnam_wave_1

-- Create table for School construction fees
SELECT DISTINCT SCHOOLID, HTAMPAID6, HTEXM016, HTEXM026, HTEXM036, HTEXM046, HTEXM056,
HTEXM066, HTEXM006
INTO Vietnam_SchoolConstructionFees
FROM vietnam_wave_1

-- Create table for Students parents fund
SELECT DISTINCT SCHOOLID, HTAMPAID7, HTEXM017, HTEXM027, HTEXM037, HTEXM047, HTEXM057,
HTEXM067, HTEXM007
INTO Vietnam_StudentsParentsFundFees
FROM vietnam_wave_1

-- Create table for Traituyen
SELECT DISTINCT SCHOOLID, HTAMPAID8, HTEXM018, HTEXM028, HTEXM038, HTEXM048, HTEXM058,
HTEXM068, HTEXM008
INTO Vietnam_TraituyenFees
FROM vietnam_wave_1

```

```

-- Create table for Health Insurance
SELECT DISTINCT SCHOOLID, HTAMPAID9, HTEXTM019, HTEXTM029, HTEXTM039, HTEXTM049, HTEXTM059,
HTEXTM069, HTEXTM009
INTO Vietnam_HealthInsuranceFees
FROM vietnam_wave_1

-- Create table for School Security Fees
SELECT DISTINCT SCHOOLID, HTAMPAID10, HTEXTM0110, HTEXTM0210, HTEXTM0310, HTEXTM0410,
HTEXTM0510, HTEXTM0610, HTEXTM0010
INTO Vietnam_SchoolSecurityFees
FROM vietnam_wave_1

-- Create table for Exam Fees
SELECT DISTINCT SCHOOLID, HTAMPAID11, HTEXTM0111, HTEXTM0211, HTEXTM0311, HTEXTM0411,
HTEXTM0511, HTEXTM0611, HTEXTM0011
INTO Vietnam_ExamFees
FROM vietnam_wave_1

-- Create table for Extra Curricular Fees
SELECT DISTINCT SCHOOLID, HTAMPAID12, HTEXTM0112, HTEXTM0212, HTEXTM0312, HTEXTM0412,
HTEXTM0512, HTEXTM0612, HTEXTM0012
INTO Vietnam_ExtraCurricularFees
FROM vietnam_wave_1

-- Create table for Grade 10 enrolment
SELECT DISTINCT SCHOOLID, HTENGR10, HTENBY10, HTTLGR10, HTTLBY10, HTNMCL10, HTALLC10,
HTENTOT10, SCHFAC02
INTO Vietnam_Grade10Enrolment
FROM vietnam_wave_1

-- Create table for Grade 11 enrolment
SELECT DISTINCT SCHOOLID, HTENGR11, HTENBY11, HTTLGR11, HTTLBY11, HTNMCL11
INTO Vietnam_Grade11Enrolment
FROM vietnam_wave_1

-- Create table for Grade 12 enrolment
SELECT DISTINCT SCHOOLID, HTENGR12, HTENBY12, HTTLGR12, HTTLBY12, HTNMCL12
INTO Vietnam_Grade12Enrolment
FROM vietnam_wave_1

```

After this, I created my views relating to each section of the data as mentioned earlier:

```

CREATE VIEW [Total Enrolment by School] AS
-- Total enrolment by school
SELECT SCHOOLID, SUM(TTLENRL) AS 'Total Enrolment by School'
FROM Vietnam_ClassEnrolment
GROUP BY SCHOOLID

CREATE VIEW [Student Attendance Level] AS
-- 1 = Normal 2 = Higher than normal 3 = Lower than normal
SELECT COUNT(CLASSID) AS 'Total Number', ATDTMSY AS 'Student Attendance'
FROM Vietnam_ClassEnrolment
WHERE ATDTMSY != ''
GROUP BY ATDTMSY

CREATE VIEW [Student Allocation To Class] AS
/*0 = There is only one class in Grade 10
1 = Randomly
2 = Alphabetically
3 = By ability (e.g. high / middle / low)

```

```

4 = According to the level of fees they pay
5 = By choice of subject
6 = Other method
7 = A combination of two or more of the above*/
SELECT COUNT(CLASSID) AS 'Total Number', SCALLCT AS 'Student Allocation To Class'
FROM Vietnam_ClassEnrolment
GROUP BY SCALLCT

CREATE VIEW [Class Ability] AS
/*1 = Low ability
2 = Medium ability
3 = High ability*/
SELECT COUNT(CLASSID) 'Total Number' , GRPABLT AS 'Class Ability'
FROM Vietnam_ClassEnrolment
GROUP BY GRPABLT

CREATE VIEW [English Lecture Periods] AS
-- Number of hours dedicated to teaching English
SELECT COUNT(CLASSID) AS 'Number of Classes' , SCMENIN AS 'Periods'
FROM Vietnam_ClassTeaching
GROUP BY SCMENIN

CREATE VIEW [Maths Lecture Hours] AS
-- Number of periods deicated to teaching Maths
SELECT COUNT(CLASSID) AS 'Number of classes', SCNMNTIN AS 'Periods'
FROM Vietnam_ClassTeaching
GROUP BY SCNMNTIN

CREATE VIEW [English Test above 30] AS
-- List of tudents who scored above 30 in English
SELECT UNIQUEID, ENG_RAWSCORE
FROM Vietnam_EnglishAndMathTests
WHERE ENG_RAWSCORE > 30

CREATE VIEW [English Test below 10] AS
-- List of students who scored below 10 in English
SELECT UNIQUEID, ENG_RAWSCORE
FROM Vietnam_EnglishAndMathTests
WHERE ENG_RAWSCORE <= 10 AND ENG_RAWSCORE IS NOT NULL

CREATE VIEW [Math Test above 30] AS
-- List of students who scored above 30 in Math
SELECT UNIQUEID, MATH_RAWSCORE
FROM Vietnam_EnglishAndMathTests
WHERE MATH_RAWSCORE > 30

CREATE VIEW [Math Test below 10] AS
-- List of students below 10 in Math
SELECT UNIQUEID, MATH_RAWSCORE
FROM Vietnam_EnglishAndMathTests
WHERE MATH_RAWSCORE <= 10 AND MATH_RAWSCORE != ''

CREATE VIEW [English Speakers] AS
-- Grouping of children's english conversations
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Never' AS 'English Speaking'
FROM Vietnam_EnglishConversations
WHERE STSPEN01 = 0 AND STSPEN02 = 0 AND STSPEN03 = 0 AND STSPEN04 = 0 AND STSPEN05 = 0
AND '' NOT IN (STSPEN01, STSPEN02, STSPEN03, STSPEN04, STSPEN05)
UNION
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Often' AS 'English Speaking'

```



```

FROM Vietnam_EnglishConversations
WHERE STSPEN01 = 1 AND STSPEN02 = 1 AND STSPEN03 = 1 AND STSPEN04 = 1 AND STSPEN05 = 1
UNION
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Occasionally' AS 'English Speaking'
FROM Vietnam_EnglishConversations
WHERE STSPEN01 = 2 AND STSPEN02 = 2 AND STSPEN03 = 2 AND STSPEN04 = 2 AND STSPEN05 = 2

```

```

CREATE VIEW [English Activities] AS
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Occasionally' AS 'English Activities'
FROM Vietnam_EnglishConversations
WHERE STATEN01 = 2 AND STATEN02 = 2 AND STATEN03 = 2 AND STATEN04 = 2
UNION
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Often' AS 'English Activities'
FROM Vietnam_EnglishConversations
WHERE STATEN01 = 1 AND STATEN02 = 1 AND STATEN03 = 1 AND STATEN04 = 1
UNION
SELECT COUNT(UNIQUEID) AS 'Total Number', 'Never' AS 'English Activities'
FROM Vietnam_EnglishConversations
WHERE STATEN01 = 0 AND STATEN02 = 0 AND STATEN03 = 0 AND STATEN04 = 0 AND '' NOT IN
(STATEN01, STATEN02, STATEN03, STATEN04)

```

```

CREATE VIEW [Grade 10 Allocation] AS
/* 0 = There is only one class in Grade 10
1 = Randomly
2 = Alphabetically
3 = By ability (e.g. high / middle / low)
4 = According to the level of fees they payYoung Lives 2016 - 2017. Vietnam School
Survey Data Dictionary (Wave 1).
25
5 = By choice of subject
6 = Other method
7 = A combination of two or more of the above methods*/
SELECT COUNT(HTALLC10) AS 'Total Number' , HTALLC10 'Grade 10 Allocation'
FROM Vietnam_Grade10Enrolment
GROUP BY HTALLC10

```

```

CREATE VIEW [Head Teacher Education Level] AS
/*1 = Upper secondary or equivalent
2 = Vocational training school
3 = College education
4 = University education (undergraduate)
5 = University education (postgraduate) */
SELECT COUNT(HTLVLEDC) AS 'Total Number' , HTLVLEDC AS 'Education Level'
FROM Vietnam_HeadTeacherEducationAndExperience
GROUP BY HTLVLEDC

```

```

CREATE VIEW [Head Teacher Highest Teaching Qualification] AS
/*0 = I am not trained
1 = Short course or crash course in teaching profession
2 = Teacher training level (2 years after upper secondary education)
3 = Teacher training college level (3 years after upper secondary education)
4 = Teacher training university level (4 years after upper secondary education) or
higher */
SELECT COUNT(HTLVLTCH) AS 'Total Number', HTLVLTCH AS 'Teaching Qualification'
FROM Vietnam_HeadTeacherEducationAndExperience
GROUP BY HTLVLTCH

```

```

CREATE VIEW [Excellent Head Teacher Awards] AS

```

```

/*0 = Never been an excellent teacher
1 = Yes, school level
2 = Yes, district level
3 = Yes, province level or higher*/
SELECT COUNT(HTEXCTCH) AS 'Total Number', HTEXCTCH AS 'Award'
FROM Vietnam_HeadTeacherEducationAndExperience
GROUP BY HTEXCTCH

```

```

CREATE VIEW [Type of School] AS
/*1 = Government
2 = Private
3 = Other*/
SELECT COUNT(HTTYPSCCH) AS 'Total Number', HTTYPSCCH AS 'Type Of School'
FROM Vietnam_HeadTeacherSchool
GROUP BY HTTYPSCCH

```

```

CREATE VIEW [Establishment Year] AS
-- Year school was established and number of students
SELECT SCHOOLID, HTNMSTEN, HTYREST
FROM Vietnam_HeadTeacherSchool

```

```

CREATE VIEW [P135 Commune] AS
/* Is this school located in a former P135 commune?
0 = No
1 = Yes
2= Don't know*/
SELECT COUNT(SCHOOLID) AS 'Total Number', HTP135CM AS 'Category'
FROM Vietnam_HeadTeacherSchool
GROUP BY HTP135CM

```

```

CREATE VIEW [62 Poorest districts] AS
/*0 = No
1 = Yes
2= Don't know*/
SELECT COUNT(SCHOOLID) AS 'Total Number', HTPRDIST AS 'Category'
FROM Vietnam_HeadTeacherSchool
GROUP BY HTPRDIST

```

```

CREATE VIEW [Best Available Schools] AS
/*0 = There are no other upper secondary schools in this district that students could
attend
1 = There is one other upper secondary school in this district that students could
attend
2 = There are two or more other upper secondary schools in this district that students
could attend*/
SELECT COUNT(SCHOOLID) AS 'Total Number', HTSCHAVL AS 'Best Available School'
FROM Vietnam_HeadTeacherSchool
GROUP BY HTSCHAVL

```

```

CREATE VIEW [Extra Classes Offered] AS
/*0 = This school does not offer additional classes outside the normal school day
1 = Remedial classes for weaker studentsYoung Lives 2016 - 2017
2 = Additional classes for excellent students
3 = A mixture of remedial classes for weaker students and additional classes for
excellent students*/
SELECT COUNT(SCHOOLID) AS 'Total Number', HTNOCMCL AS 'Type Of Extra Classes'
FROM Vietnam_HeadTeacherSchool
GROUP BY HTNOCMCL

```

```

CREATE VIEW [School Discussion At Home] AS
-- How often school discussions are held at home
SELECT COUNT(UNIQUEID) AS 'Total Number', STPLHL01 AS 'School discussion at home'
FROM Vietnam_HouseholdContributions
GROUP BY STPLHL01

```

```

CREATE VIEW [Help With Maths At Home] AS
-- How often student gets help with maths at home
SELECT COUNT(UNIQUEID) AS 'Total Number', STPLHL03 AS 'Help with maths'
FROM Vietnam_HouseholdContributions
GROUP BY STPLHL03

```

```

CREATE VIEW [Maths Discussion At Home] AS
-- How often maths discussions are held at home
SELECT COUNT(UNIQUEID) AS 'Total Number', STPLHL04 AS 'Maths Discussions'
FROM Vietnam_HouseholdContributions
GROUP BY STPLHL04

```

```

CREATE VIEW [Mom Education] AS
/* 0 = Never been to school
1 = Primary school (Grades 1-5)
2 = Lower secondary school (Grades 6-9)
3 = Intermediate vocational training
4 = Upper secondary school (Grades 10
5 = Higher education (e.g. university/college)
6 = Don't know */
SELECT COUNT(UNIQUEID) AS 'Total Number' , MOM_EDUC AS 'Mother Education'
FROM Vietnam_ParentsEducation
GROUP BY MOM_EDUC

```

```

CREATE VIEW [Dad Education] AS
/*0 = Never been to school
1 = Primary school (Grades 1-5)
2 = Lower secondary school (Grades 6-9)
3 = Intermediate vocational training
4 = Upper secondary school (Grades 10
5 = Higher education (e.g. university/college)
6 = Don't know */
SELECT COUNT(UNIQUEID) AS 'Total Number', DAD_EDUC as 'Father Education'
FROM Vietnam_ParentsEducation
GROUP BY DAD_EDUC

```

```

CREATE VIEW [Parents with University or College degree] AS
-- MOM AND DAD WITH UNIVERSITY OR COLLEGE DEGREE
SELECT UNIQUEID, DAD_EDUC, MOM_EDUC
FROM Vietnam_ParentsEducation
WHERE DAD_EDUC = 5 AND MOM_EDUC = 5

```

```

CREATE VIEW [Parents with Primary School] AS
-- MOM AND DAD WITH PRIMARY SCHOOL
SELECT UNIQUEID, DAD_EDUC, MOM_EDUC
FROM Vietnam_ParentsEducation
WHERE DAD_EDUC = 0 AND MOM_EDUC = 0 AND '' NOT IN (MOM_EDUC, DAD_EDUC)

```

```

CREATE VIEW [Highest Grade Student Expects] AS
/* highest grade or level of education you expect to complete?
1 = Upper secondary (general)
2 = Upper secondary (vocational)
3 = Professional school
4 = College (non - vocational)
5 = College (vocational)

```

```

6 = Bachelor's degree
7 = Master's degree
8 = Doctorate/PhD */
SELECT count(uniqueid) AS 'Total Number', STHGHGRD AS 'Highest grade student expects'
FROM Vietnam_SchoolDecisions
GROUP BY STHGHGRD

CREATE VIEW [Schools with ELECTRICTY, LIBRARY AND INTERNET COMPUTERS] AS
-- Schools with ELECTRICTY, LIBRARY AND INTERNET COMPUTERS
SELECT SCHOOLID, SCHFAC03, SCHFAC04, SCHFAC05
FROM Vietnam_SchoolProvisions
WHERE SCHFAC03 = 1 AND SCHFAC04 = 1 AND SCHFAC05 = 1

CREATE VIEW [Schools with NONE of ELECTRICTY, LIBRARY AND INTERNET COMPUTERS] AS
-- NONE of ELECTRICTY, LIBRARY AND INTERNET COMPUTERS
SELECT SCHOOLID, SCHFAC03, SCHFAC04, SCHFAC05
FROM Vietnam_SchoolProvisions
WHERE SCHFAC03 = 0 AND SCHFAC04 = 0 AND SCHFAC05 = 0

```

## TASK 3.

I was able to get the requested 24 months ranging from January 2017 to December 2018, which I extracted the data one by one and had it in a single Excel file which I saved as a 97-2003 excel file. This was a faster method than importing each month one by one and saved me some stress.

After this, I made a combined table of all the specified months:

```

SELECT *
INTO Crimes
FROM dbo.['2017-01-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-02-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-03-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-04-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-05-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-06-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-07-greater-manchester-stre$']
UNION ALL
SELECT *
FROM dbo.['2017-08-greater-manchester-stre$']
UNION ALL
SELECT *

```

```

FROM dbo.[ '2017-09-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2017-10-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2017-11-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2017-12-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-01-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-02-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-03-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-04-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-05-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-06-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-07-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-08-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-09-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-10-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-11-greater-manchester-stre$' ]
UNION ALL
SELECT *
FROM dbo.[ '2018-12-greater-manchester-stre$' ]

```

From this, I was able to create Views and split the data by Crime committed and others.

```

CREATE VIEW [Anti-Social Behaviour] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Anti-social behaviour'

CREATE VIEW [Drugs] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Drugs'

```

```

CREATE VIEW [Robbery] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Robbery'

CREATE VIEW [Possession of Weapons] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Possession of weapons'

CREATE VIEW [Shoplifting] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Shoplifting'

CREATE VIEW [Burglary] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Burglary'

CREATE VIEW [Bicycle Theft] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Bicycle Theft'

CREATE VIEW [Other Theft] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Other theft'

CREATE VIEW [Public Order] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Public order'

CREATE VIEW [Vehicle Crimes] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Vehicle crime'

CREATE VIEW [Criminal damage and Arson] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Criminal damage and arson'

CREATE VIEW [Violence and sexual offences] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Violence and sexual offences'

CREATE VIEW [Theft from the person] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Theft from the person'

CREATE VIEW [Other Crime] AS
SELECT *
FROM Crimes
WHERE [Crime type] = 'Other crime'

CREATE VIEW [Anti-Social Behaviour in Salford] AS
SELECT *

```

```
FROM Crimes
WHERE [Crime type] = 'anti-social behaviour' AND [LSOA name] LIKE 'Salford%'
```

```
CREATE VIEW[All crime by month] AS
-- Show all crimes in every month
SELECT COUNT(Month) AS 'Total', Month
FROM Crimes
GROUP BY Month
```

```
CREATE VIEW[Public order by Month] AS
-- Show burglary crimes by month
SELECT COUNT(Month) AS 'Total', Month
FROM Crimes
WHERE [Crime type] = 'public order'
GROUP BY Month
```

```
CREATE VIEW[shoplifting by Month] AS
-- show vehicle crime by month
SELECT COUNT(Month) AS 'Total', Month
FROM Crimes
WHERE [Crime type] = 'shoplifting'
GROUP BY Month
```

```
CREATE VIEW[Burglary] AS
SELECT TOP 10 COUNT(Month) AS 'Total', Month
FROM Crimes
WHERE [Crime type] = 'Burglary' AND [LSOA name] LIKE 'Salford%'
GROUP BY Month
```

```
CREATE VIEW[Highest LSOA Names by Crime] AS
SELECT TOP 10 COUNT([LSOA name]) AS 'TOTAL', [LSOA NAME]
FROM Crimes
GROUP BY [LSOA name]
ORDER BY TOTAL DESC
```

```
CREATE VIEW[crimes in Bury in 2017] AS
SELECT COUNT([CRIME Type]) AS 'Total', Month
FROM Crimes
WHERE Month LIKE '2017%' AND [LSOA name] LIKE 'Bury%'
GROUP BY Month
```

```
CREATE VIEW [Most popular crimes in summer 2017 and 2018] AS
SELECT TOP 10 COUNT([CRIME Type]) AS 'Total', [Crime type]
FROM Crimes
WHERE Month LIKE '2018%' AND [LSOA name] LIKE 'Salford%'
GROUP BY [Crime type]
ORDER BY Total DESC
```

I also picked out the districts represented as Greater Manchester which I used to Create relevant tables from the LSOA population data. I noticed there were some NULL values from the start on the ID

```
SELECT [LA (2019 boundaries)], [Area Codes], LSOA, [ALL AGES]
INTO [Manchester Population]
FROM [LSOA 2018]
WHERE [LSOA] LIKE 'Bolton%' OR [LSOA] LIKE 'Bury%' OR [LSOA] LIKE 'Manchester%' OR
```

```
[LSOA] LIKE 'OLDHAM%' OR [LSOA] LIKE 'ROCHDALE%' OR [LSOA] LIKE 'SALFORD%'
OR [LSOA] LIKE 'STOCKPORT%' OR [LSOA] LIKE 'TAMESIDE%' OR [LSOA] LIKE 'TRAFFORD%'
OR [LSOA] LIKE 'WIGAN%'
```

I was also able to make both visualizations in QGIS, but I had to make some minor adjustments to the table with certain statements to get it working: First, a schema called Manchester was created, then a column named Geolocation was added. After this, the Geolocation column was Updated with Longitude and Latitude values. Next, an ID identity was created then a Primary Key created on the Identity.

```
CREATE SCHEMA Manchester
```

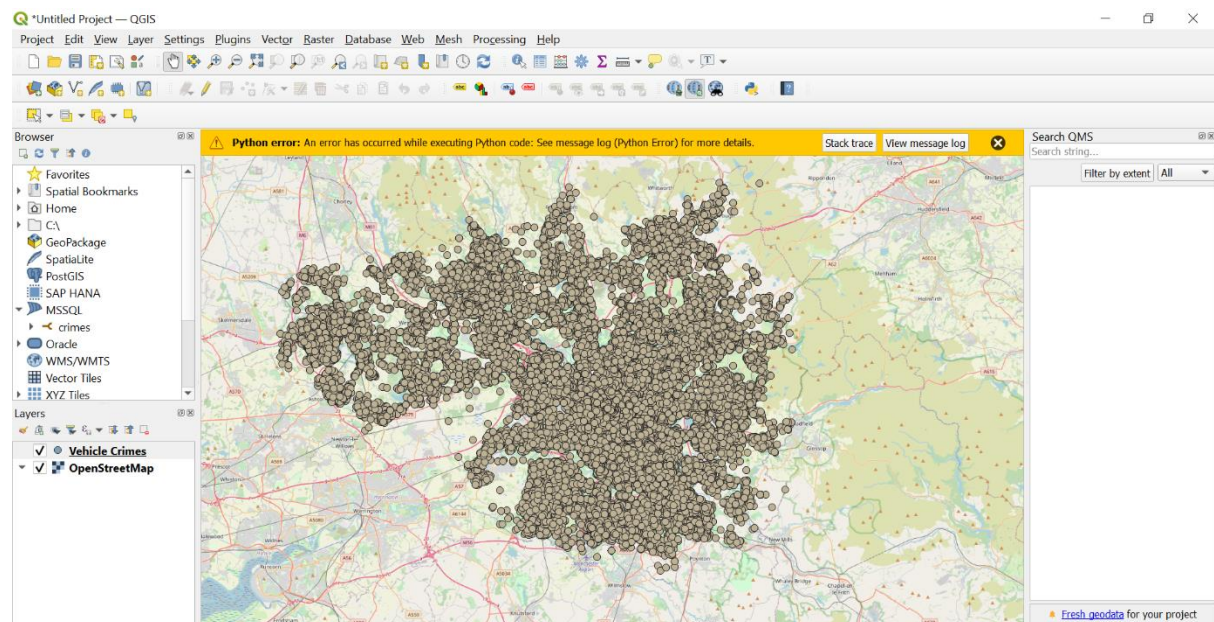
```
ALTER TABLE [dbo].[Crimes]
ADD [GeoLocation] GEOGRAPHY
GO
```

```
UPDATE [dbo].[Crimes]
SET [GeoLocation] = geography::Point([Latitude], [Longitude], 4326)
WHERE [Longitude] IS NOT NULL and [Latitude] IS NOT NULL
GO
```

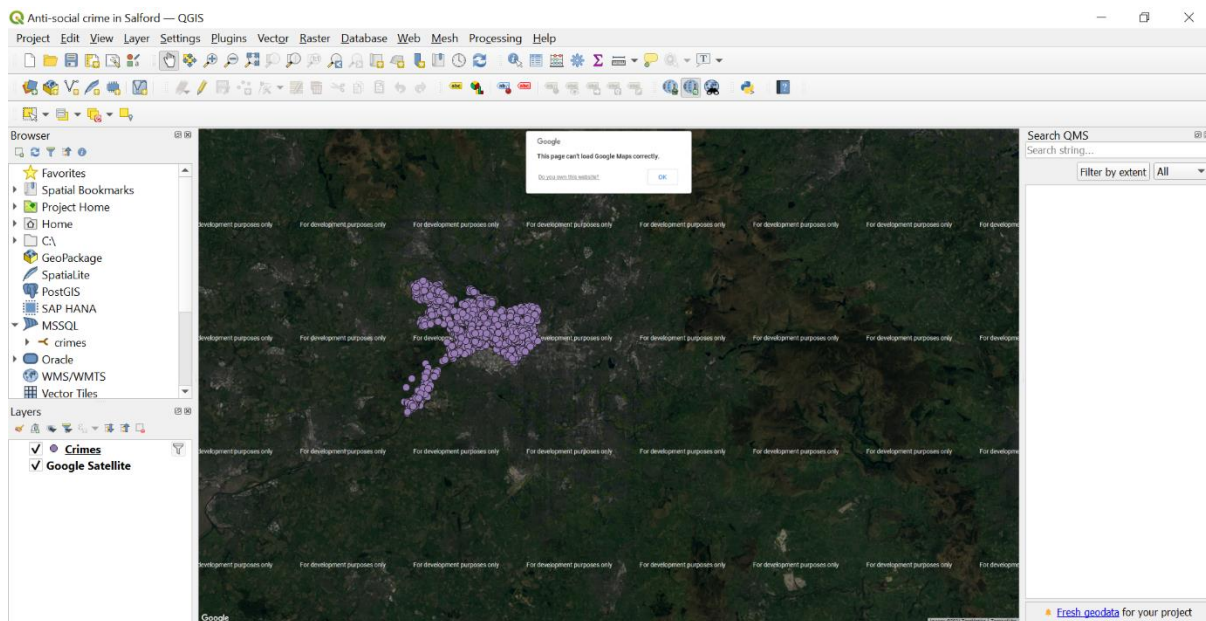
```
ALTER TABLE dbo.Crimes
ADD ID INT IDENTITY;
```

```
ALTER TABLE dbo.Crimes
ADD CONSTRAINT PK_Id PRIMARY KEY NONCLUSTERED (ID);
GO
```

The first shows the Vehicle crime, while the second shows the anti-social behaviour in Salford.







## Database security

Security is a great call for concern with databases and we must always try to protect our data against loss, destruction or misuse. Firewalls could be put in place to help safeguard the database, to limit connections. With firewalls, injection attacks could be reduced to a minimum as they are very rampant to steal data from databases and sell to random people or expose to the public.

Database servers as well need to be kept in secure locations to avoid unauthorised access as someone within the office could have been paid to access parts of the servers. If you decide to use a web hosted server, you would want to go for a Host that takes its security very seriously. A lacklustre one could put your whole management at more risk. Also, to help with access to the servers, a limited number of access should be given and I would suggest to one person per group. I would also advice these:

- Use of strong passwords, a mix of Capitals and symbols
- Password hashes should be stored encrypted and salted
- An account lock should occur after three or four login attempts

As a database administrator, you could also try to run a hack on your database to test its functionality. Try several means to get into it like a real attacker would.

## **Backup and restore strategy:**

Over time, possible data loss or failures may occur at certain points and to safeguard, there needs to be proper safeguarding methods applied. Besides data loss, there could be user input errors as well such as dropping tables, deleting rows, inserting into wrong tables etc. This is where Backup and Restore strategies come to play. In databases, there are three methods that could be applied.

- Full backups
- Differential backups
- Transactional log backups
- 

For the purpose of this database as a research database, we would be utilising the Full backup measures which would be implemented manually. I made backups occasionally after loading the tables, running my views etc. For a simple format, this is the script to go with:

```
BACKUP DATABASE Taiwo_Vietnamschoolsurvey  
TO DISK = 'C:\AdventureWorks.BAK'  
GO
```

It can also be implemented with the use of a password with the following script:

```
BACKUP DATABASE Taiwo_vietnamschoolsurvey  
TO DISK = 'C:\AdventureWorks.BAK'  
WITH PASSWORD = 'Q!W@E#R$'  
GO
```

## **Business intelligence techniques**

Business intelligence over the years, has evolved from basic charts to more interactive models like Power BI but for the purpose of this report, I utilised SSRS and Excel charts. Business intelligence (BI) is mainly use for analysis and converting raw data into better models to acquire more information for use in businesses to help in decision making.

With data, new information keeps coming in day by day and with the inflow, new patterns could surface and would help to find new trends to utilise for your business. I shall discuss two main techniques related to the purpose of this research which are Analytics and Model Visualization.

In analytics, I used the data provided to find the children whose mothers and fathers had no schooling, had a university degree and the possible effect of this on the children, thanks to analytics I also was able to find the wealth index of the countries over the rounds to see if things improved or deteriorated, I was also able to find out how students were admitted to their classes in Vietnam etc

Using model visualization, I was able to compare all four countries on a single chart to find which had the highest or lowest figures at a glance.

## **Data privacy, ethical and legal issues**

Looking at the data for all three tasks, a good job was done with keeping the privacy of the students and crime data anonymous. The students were represented with numbers and alphabets to indicate their respective countries.

## References:

<https://beta.ukdataservice.ac.uk/datacatalogue/series/series?id=2000060#!/resources>

<https://beta.ukdataservice.ac.uk/datacatalogue/studies/study?id=8360>

<https://www.worldbank.org/en/topic/poverty/publication/ethiopia-poverty-assessment>

<https://www.worldbank.org/en/news/infographic/2016/05/27/india-s-poverty-profile>

<https://www.sqlshack.com/sql-server-logins-users-security-identifiers-sids/>

<https://blog.devart.com/sql-database-design-basics-with-example.html>

<https://www.esecurityplanet.com/networks/database-security-best-practices/>

<https://callminer.com/blog/what-is-business-intelligence-definition-techniques-tools-and-tips-from-experts/>