Advanced Databases

Aamir Nawab

@00420807

TABLE OF CONTENTS

ABSTRACT

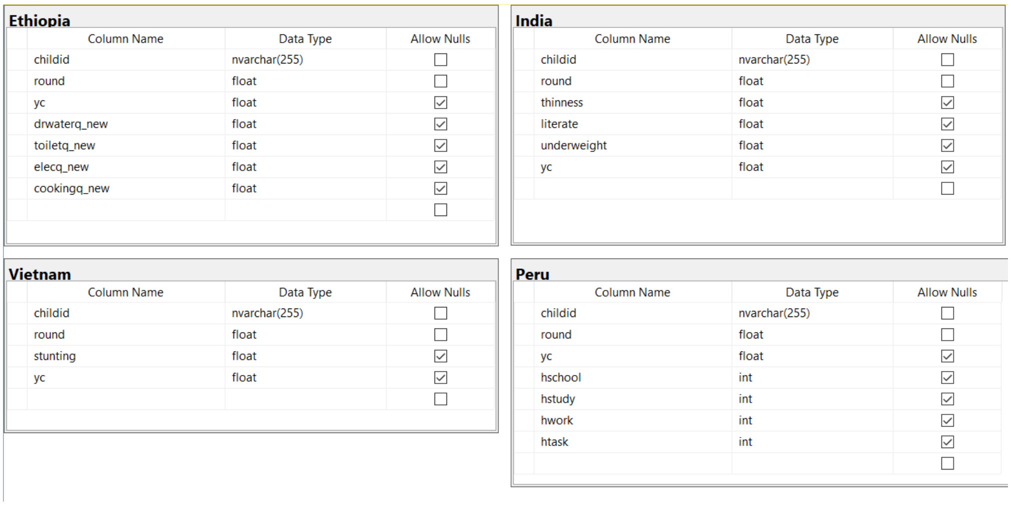
The aim of this report is to showcase the design of a database system using the Young Lives dataset. The report will show the T-SQL queries used to create the database system as well as the queries used to retrieve data. In addition to this, chart representation of the T-SQL queries was created with filtering, sorting, and grouping functionality alongside various search facilities that can be used in the Child Well-Being monitor.

INTRODUCTION

Young Lives conducted a survey over 15 years in four countries, India, Vietnam, Peru, and Ethiopia. The aim was to keep a track of the lives of 12,000 children who were surveyed every 3-4 years. The report will use the data of the study to analyse the changing nature of the causes and consequences childhood poverty.

RELATIONAL SCHEMA

The figure below is the schema of the database. To analyse the Young Lives dataset, four tables were created each representing the countries the survey took part in.



Ethiopia

Column Name	Data Type	Allow Nulls
childid	nvarchar(255)	□
round	float	□
yc	float	☑
drwaterq_new	float	☑
toiletq_new	float	☑
elecq_new	float	☑
cookingq_new	float	☑
		□

India

Column Name	Data Type	Allow Nulls
childid	nvarchar(255)	□
round	float	□
thinness	float	☑
literate	float	☑
underweight	float	☑
yc	float	☑
		□

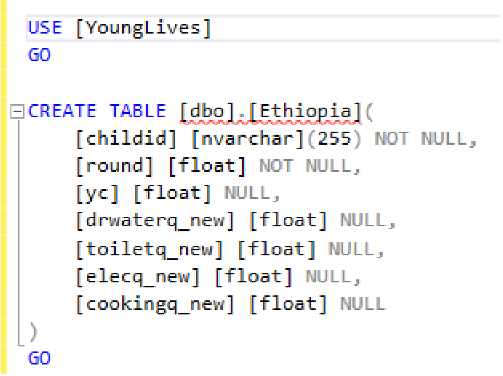
Vietnam

Column Name	Data Type	Allow Nulls
childid	nvarchar(255)	□
round	float	□
stunting	float	☑
yc	float	☑
		□

Peru

Column Name	Data Type	Allow Nulls
childid	nvarchar(255)	□
round	float	□
yc	float	☑
hschool	int	☑
hstudy	int	☑
hwork	int	☑
htask	int	☑
		□

T – SQL statement – Create Table



```
USE [YoungLives]
GO

CREATE TABLE [dbo].[Ethiopia](
    [childid] [nvarchar](255) NOT NULL,
    [round] [float] NOT NULL,
    [yc] [float] NULL,
    [drwaterq_new] [float] NULL,
    [toiletq_new] [float] NULL,
    [elecq_new] [float] NULL,
    [cookingq_new] [float] NULL
)
GO
```

To load the Young Lives dataset on the SQL server, a database named YoungLives was created. After four tables were created each representing the countries the survey took part in. The query above shows how the table Ethiopia was created in MSSQL.

DESIGN CONSIDERATIONS

Normalisation

Database normalisation serves two purposes, to remove useless data and ensure all data dependencies are logical. A database can be normalised by abiding to accepted normal forms.

First Normal Form (1NF): The idea of 1NF when designing a table is that the structured data is divided into tables in which each row each is uniquely identified through a primary key. In addition, the elimination of group repetition is done by placing each group in their individual table and connecting them through a one-to-one relationship. YoungLives database tables adheres to First Normal Form as the tables were created using MSSQL and only have a single valued columns/attributes.

Second Normal Form(2NF): 2NF objective is to prevent repeated data amongst tables. This is done moving data that have partial dependencies on the primary key to another table. The YoungLives database tables adhere to Second Normal Form as all non-key columns for each table is dependent on the table's primary key and have no partial dependencies.

Third Normal Form(3NF): The idea of 3NF is that data that are not dependent on a primary key in table is removed. Moreover, non-key columns can be dependent on each other and a table only meets the requirements of 3NF only if adheres to the 1NF and 2NF. The tables in the YoungLives meet the requirements of all three accepted normal, therefore data is normalised.

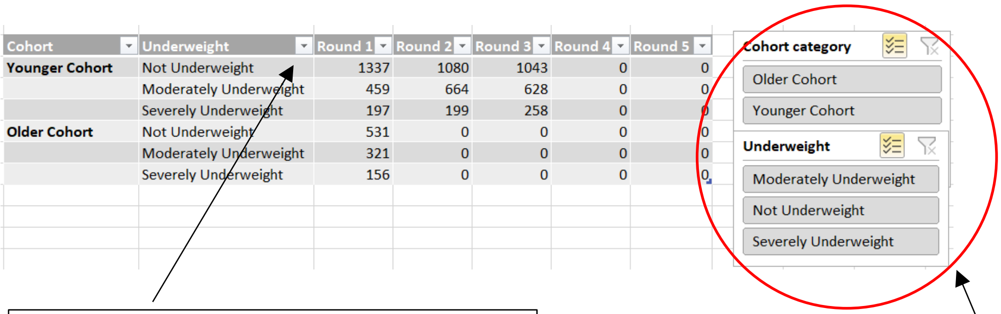
Constraints

Constraints are specified rules or restrictions that limit the type of data that is allowed in a table. These simply ensures the integrity and accuracy of data stored inside a table. The tables in YoungLives database use two types of constraints

Not Null Constraint: The constraint is a rule where a column is not allowed null values i.e. each row in the column where the not null constraint is specified must contain a value.

Primary Key Constraint: This uniquely identifies each row in a column i.e. no two rows in the same can contain the same primary key. This also means that that are no Null values in primary keys.

Explanation of Excel table design for Task 1



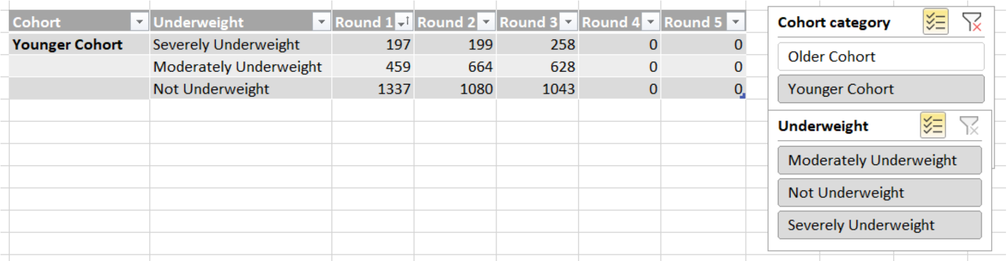
Cohort	Underweight	Round 1	Round 2	Round 3	Round 4	Round 5
Younger Cohort	Not Underweight	1337	1080	1043	0	0
	Moderately Underweight	459	664	628	0	0
	Severely Underweight	197	199	258	0	0
Older Cohort	Not Underweight	531	0	0	0	0
	Moderately Underweight	321	0	0	0	0
	Severely Underweight	156	0	0	0	0

The feature allows to sort and filter data

This is a slicer that allows searching and grouping data to be a lot easier for the client

For example, lets apply the filter, Younger Cohort and sort Round 1 data to the smallest no. of children to the largest

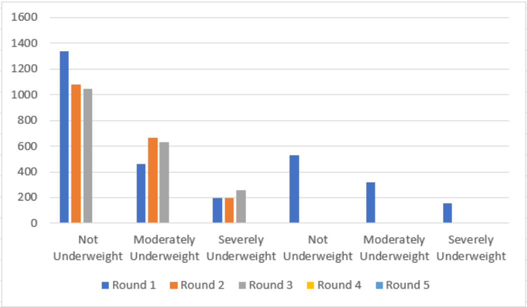
Fig. After filter is applied



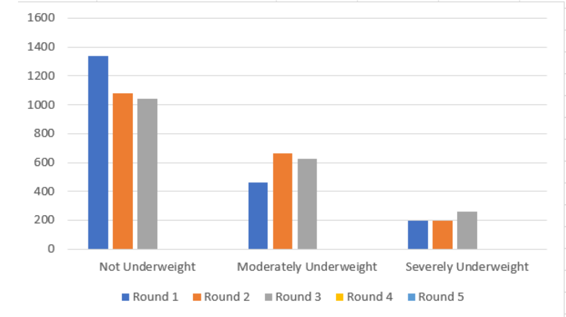
Cohort	Underweight	Round 1	Round 2	Round 3	Round 4	Round 5
Younger Cohort	Severely Underweight	197	199	258	0	0
	Moderately Underweight	459	664	628	0	0
	Not Underweight	1337	1080	1043	0	0

The filter can also change the data of the chart as shown below

Before Filter



After filter



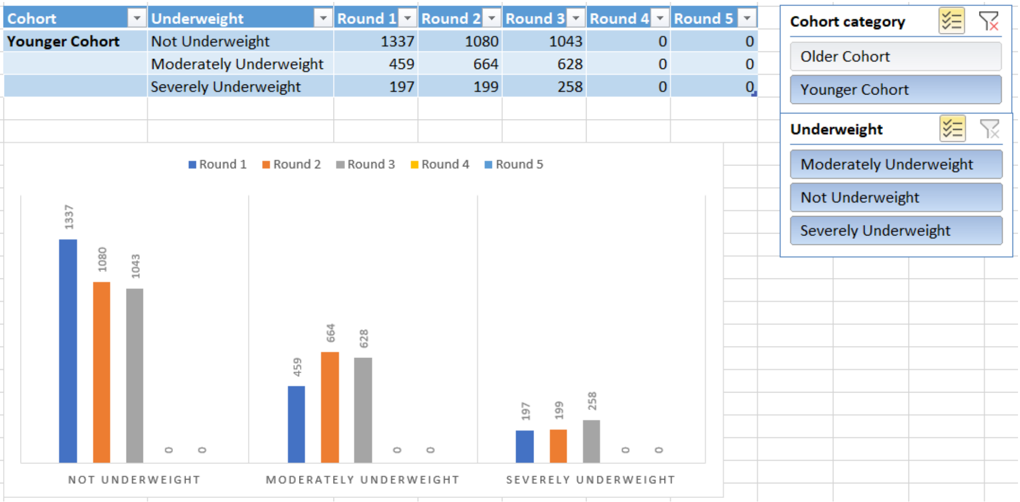
Explanation – The purpose of this to provide various search facilities in the Child Well-Being Monitor as well as filtering, sorting and grouping functionality in the reports.

Report 1 – India

One of the consequences of child poverty in India amongst children is malnutrition (Young Lives, 2020). The following charts and tables is a representation of data gathered by Young Lives on whether a child is underweight. There are two charts one for the Younger Cohort category and Older Cohort category. The purpose of this is to allow a better examination of the changing nature of childhood poverty between two age groups.

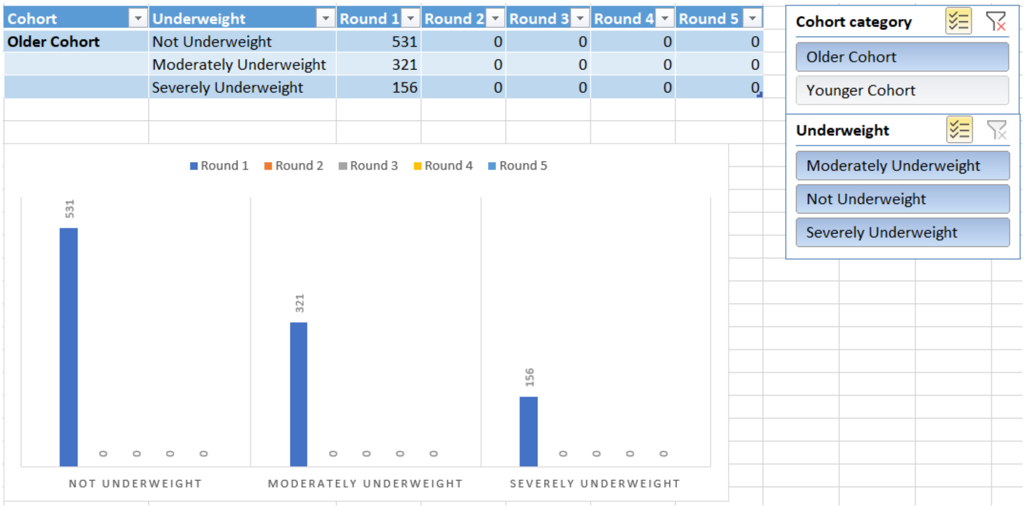
Report 1 – Excel Representation

The chart represents whether a child is underweight for the Younger Cohort category over the five rounds of the survey. The underweight data is split into three categories, not underweight, severely underweight, and moderately underweight.



Cohort	Underweight	Round 1	Round 2	Round 3	Round 4	Round 5
Younger Cohort	Not Underweight	1337	1080	1043	0	0
	Moderately Underweight	459	664	628	0	0
	Severely Underweight	197	199	258	0	0

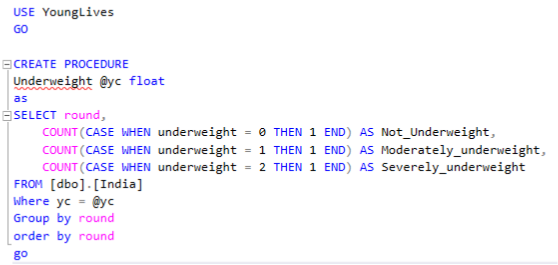
The second chart represents the older cohort and their underweight category



Cohort	Underweight	Round 1	Round 2	Round 3	Round 4	Round 5
Older Cohort	Not Underweight	531	0	0	0	0
	Moderately Underweight	321	0	0	0	0
	Severely Underweight	156	0	0	0	0

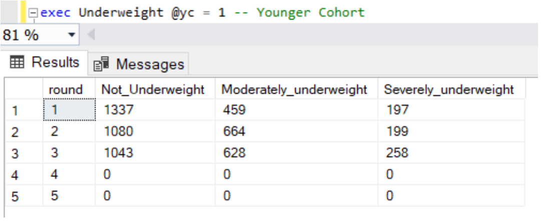
T- SQL query

The following query is stored procedure where it takes yc as a parameter. The purpose of this stored procedure is the count the different categories of underweight based on the @yc parameter given. The query is group by the column ‘round’ therefore SQL will count each individual round of the different categories of underweight.



```
USE YoungLives
GO

CREATE PROCEDURE
Underweight @yc float
as
SELECT round,
    COUNT(CASE WHEN underweight = 0 THEN 1 END) AS Not_Underweight,
    COUNT(CASE WHEN underweight = 1 THEN 1 END) AS Moderately_underweight,
    COUNT(CASE WHEN underweight = 2 THEN 1 END) AS Severely_underweight
FROM [dbo].[India]
Where yc = @yc
Group by round
order by round
go
```

To retrieve data using the stored procedure the following statement would be used. The following query retrieves data of underweight for younger cohort



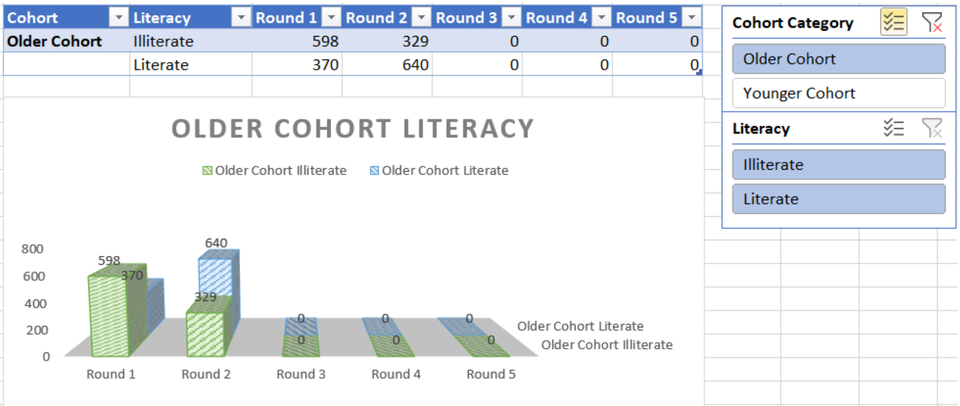
```
exec Underweight @yc = 1 -- Younger Cohort
```

	round	Not_Underweight	Moderately_underweight	Severely_underweight
1	1	1337	459	197
2	2	1080	664	199
3	3	1043	628	258
4	4	0	0	0
5	5	0	0	0

Report 2 – Literacy in India excel representation

Another underlying problem that was highlighted on the Young Lives website was the illiteracy amongst the children in India. The following report aims to measure literacy amongst children in India. The report is split in two charts where it measures the literacy amongst the Younger Cohort and Older Cohort.

The first report is amongst the Older Cohort where it is split into two categories literate and illiterate over 5 rounds



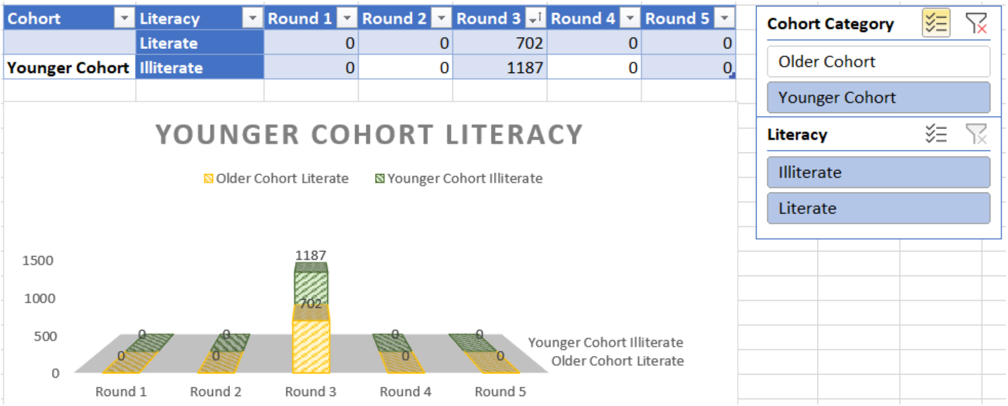
Cohort	Literacy	Round 1	Round 2	Round 3	Round 4	Round 5
Older Cohort	Illiterate	598	329	0	0	0
	Literate	370	640	0	0	0

Cohort Category
Older Cohort
Younger Cohort

Literacy
Illiterate
Literate

Figure. Represents literacy amongst older cohort

The following report measures the literacy amongst the younger cohort which is split into two categories literate and illiterate



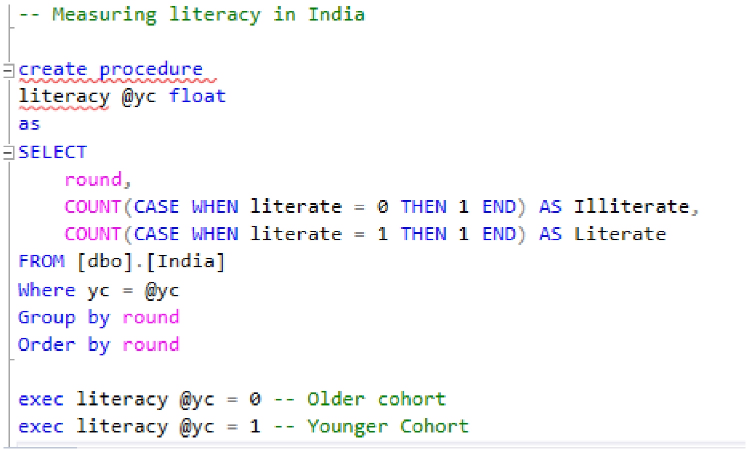
Cohort	Literacy	Round 1	Round 2	Round 3	Round 4	Round 5
	Literate	0	0	702	0	0
Younger Cohort	Illiterate	0	0	1187	0	0

Cohort Category: Older Cohort, Younger Cohort (selected)

Literacy: Illiterate (selected), Literate (selected)

T-SQL query

The following query is stored procedure where it takes yc as a parameter. The purpose of this stored procedure is the count the values 1(literate) and 0(illiterate) for the given yc parameter. The last two line execute the stored procedure by using the desired parameter and retrieves the relevant data.

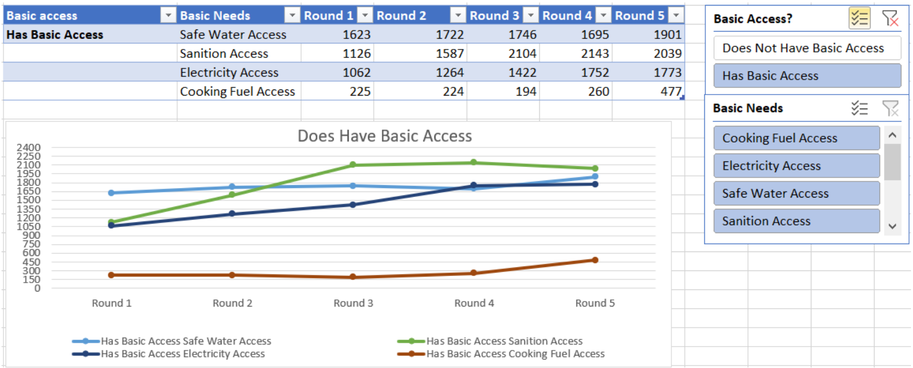


```
-- Measuring literacy in India

create procedure
literacy @yc float
as
SELECT
    round,
    COUNT(CASE WHEN literate = 0 THEN 1 END) AS Illiterate,
    COUNT(CASE WHEN literate = 1 THEN 1 END) AS Literate
FROM [dbo].[India]
Where yc = @yc
Group by round
Order by round

exec literacy @yc = 0 -- Older cohort
exec literacy @yc = 1 -- Younger Cohort
```

Report 3 – Ethiopia

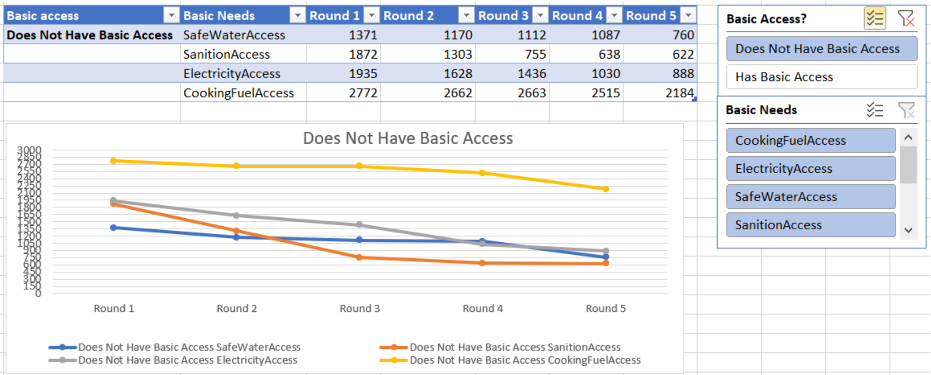
UNICEF identifies the one of the leading causes of child poverty is the lack of basic services in Ethiopia. The following report measures this using the Young Lives dataset and uses charts from Excel to show how the access of basic needs have change over five rounds.

The following chart represents those who have access to basic services. This is broken down into four categories, Safe Water Access, Sanitation Access, Electricity Access and Cooking Fuel Access. This has been measures over five round to see the changing nature of basic services access of Young Lives children in Ethiopia.



Basic access	Basic Needs	Round 1	Round 2	Round 3	Round 4	Round 5
Has Basic Access	Safe Water Access	1623	1722	1746	1695	1901
	Sanition Access	1126	1587	2104	2143	2039
	Electricity Access	1062	1264	1422	1752	1773
	Cooking Fuel Access	225	224	194	260	477

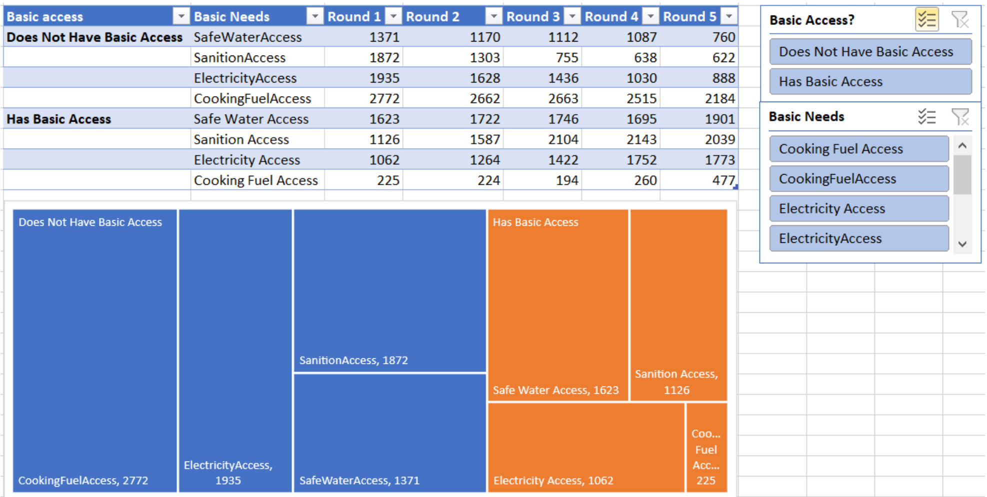
The following chart represents Young Lives children that do not have access to basic services. This again is broken down into four categories, Safe Water Access, Sanitation Access, Electricity Access and Cooking Fuel Access as well as measure across five rounds.



Basic access	Basic Needs	Round 1	Round 2	Round 3	Round 4	Round 5
Does Not Have Basic Access	SafeWaterAccess	1371	1170	1112	1087	760
	SanitionAccess	1872	1303	755	638	622
	ElectricityAccess	1935	1628	1436	1030	888
	CookingFuelAccess	2772	2662	2663	2515	2184

Does Not Have Basic Access
3000
2850
2700
2550
2400
2250
2100
1950
1800
1650
1500
1350
1200
1050
900
750
600
450
300
150
0
Round 1
Round 2
Round 3
Round 4
Round 5
Does Not Have Basic Access SafeWaterAccess
Does Not Have Basic Access ElectricityAccess
Does Not Have Basic Access SanitionAccess
Does Not Have Basic Access CookingFuelAccess
Basic Access?
Does Not Have Basic Access
Has Basic Access
Basic Needs
CookingFuelAccess
ElectricityAccess
SafeWaterAccess
SanitionAccess

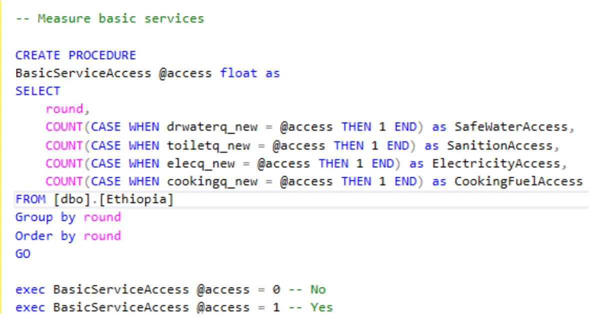
The following is a Treemap chart that compares those who have basic services and those who do not.



Basic access	Basic Needs	Round 1	Round 2	Round 3	Round 4	Round 5
Does Not Have Basic Access	SafeWaterAccess	1371	1170	1112	1087	760
	SanitionAccess	1872	1303	755	638	622
	ElectricityAccess	1935	1628	1436	1030	888
	CookingFuelAccess	2772	2662	2663	2515	2184
Has Basic Access	Safe Water Access	1623	1722	1746	1695	1901
	Sanition Access	1126	1587	2104	2143	2039
	Electricity Access	1062	1264	1422	1752	1773
	Cooking Fuel Access	225	224	194	260	477

T - SQL query

The following stored procedure counts the four categories of basic services. Each count() function requires the @access parameter, in this case 0(no) or 1(yes). The last two lines executes the stored procedure retrieving the count of each category which is group by round.



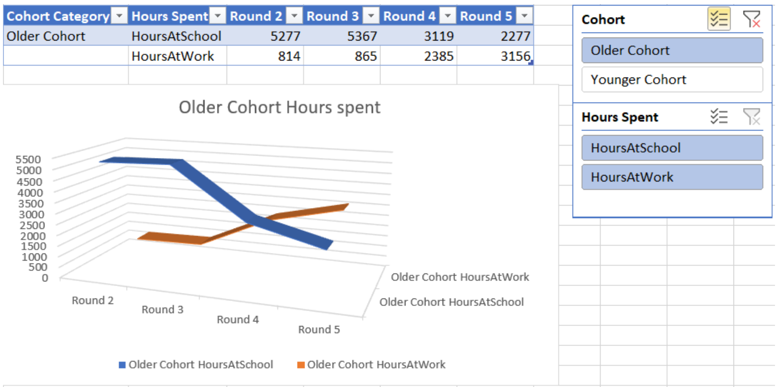
```
-- Measure basic services

CREATE PROCEDURE
BasicServiceAccess @access float as
SELECT
    round,
    COUNT(CASE WHEN drwaterq_new = @access THEN 1 END) as SafeWaterAccess,
    COUNT(CASE WHEN toiletq_new = @access THEN 1 END) as SanitionAccess,
    COUNT(CASE WHEN elecq_new = @access THEN 1 END) as ElectricityAccess,
    COUNT(CASE WHEN cookingq_new = @access THEN 1 END) as CookingFuelAccess
FROM [dbo].[Ethiopia]
Group by round
Order by round
GO

exec BasicServiceAccess @access = 0 -- No
exec BasicServiceAccess @access = 1 -- Yes
```

Report 4 – Peru

A Guardian article in 2012, reveals that a four-year plan was launched which was backed by US funding, in an attempt ‘to get thousands of poor children off work and into full-time education’ in Peru. The report will use the Young Lives dataset to measure time Young Lives children time spent in work against the time spent in studying.

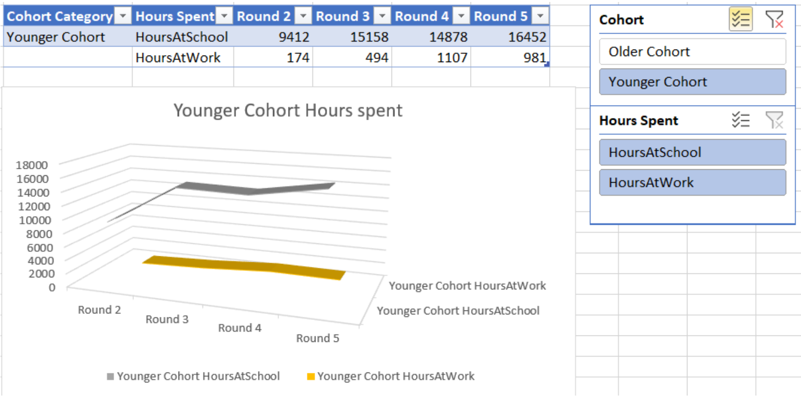
The following report will look at the difference of hours spent on either work or study through two different age groups, Younger Cohort and Older Cohort. The purpose of this is to see the changing nature of the consequences of child poverty by comparing two different generations.

The first chart looks the hours spent by the Older Cohort category, which is broken into two categories Hours at School and Hours at Work.



Cohort Category	Hours Spent	Round 2	Round 3	Round 4	Round 5
Older Cohort	HoursAtSchool	5277	5367	3119	2277
	HoursAtWork	814	865	2385	3156

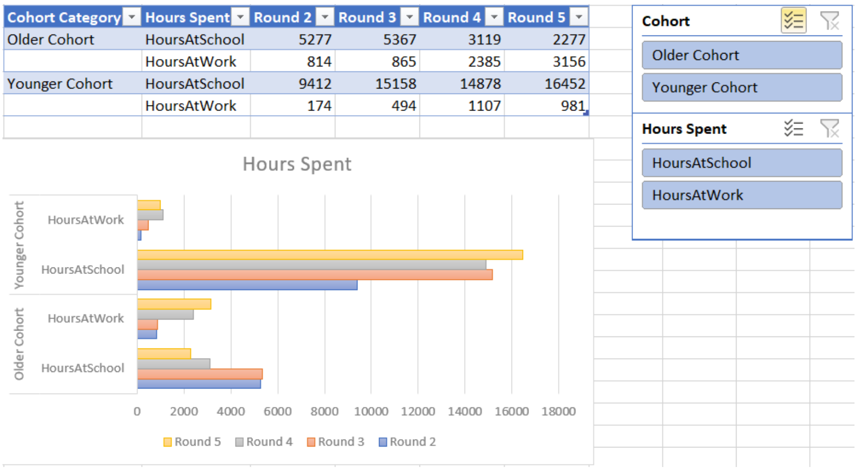
The second chart looks at the hours spent by the Younger Cohort category, which is also broken down in to two categories Hours at School and Hours at Work



Cohort Category	Hours Spent	Round 2	Round 3	Round 4	Round 5
Younger Cohort	HoursAtSchool	9412	15158	14878	16452
	HoursAtWork	174	494	1107	981

Younger Cohort Hours spent
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Round 2
Round 3
Round 4
Round 5
Younger Cohort HoursAtWork
Younger Cohort HoursAtSchool
■ Younger Cohort HoursAtSchool
■ Younger Cohort HoursAtWork
Cohort
Older Cohort
Younger Cohort
Hours Spent
HoursAtSchool
HoursAtWork

The final chart compares both Younger Cohort category and Older Cohort category



Cohort Category	Hours Spent	Round 2	Round 3	Round 4	Round 5
Older Cohort	HoursAtSchool	5277	5367	3119	2277
	HoursAtWork	814	865	2385	3156
Younger Cohort	HoursAtSchool	9412	15158	14878	16452
	HoursAtWork	174	494	1107	981

Cohort

Older Cohort

Younger Cohort

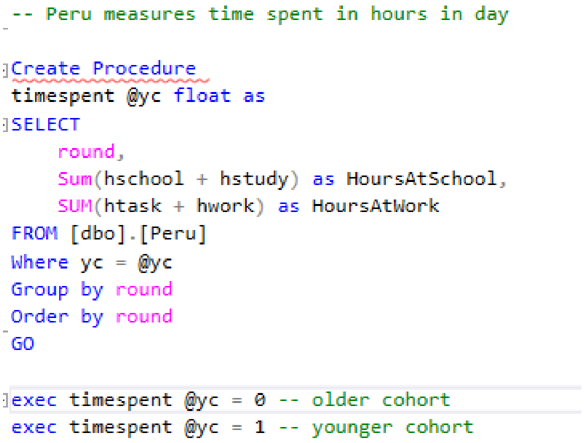
Hours Spent

HoursAtSchool

HoursAtWork

T-SQL

The following T-SQL query was used to retrieve the chart data above. A stored procedure was created name timespent where it takes yc as parameter. The purpose of the stored procedure is sum up the hours spent studying for one of the columns and hours spent at work in another for each round. The last two lines execute the procedure by taking a @yc parameter.

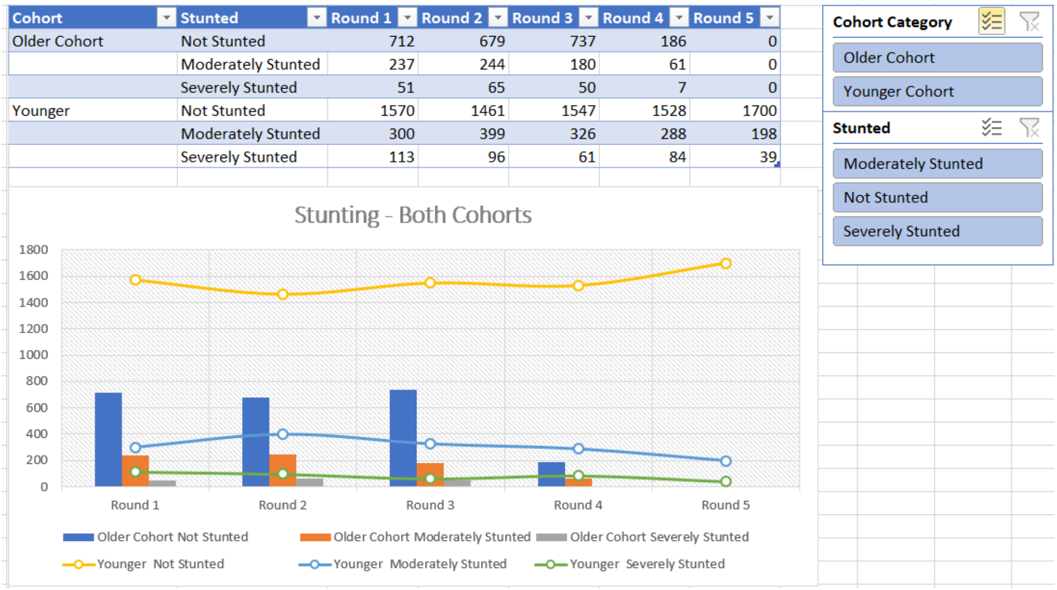


```
-- Peru measures time spent in hours in day

Create Procedure
timespent @yc float as
SELECT
    round,
    Sum(hschool + hstudy) as HoursAtSchool,
    SUM(htask + hwork) as HoursAtWork
FROM [dbo].[Peru]
Where yc = @yc
Group by round
Order by round
GO

exec timespent @yc = 0 -- older cohort
exec timespent @yc = 1 -- younger cohort
```

Report 5 – Vietnam stunting

The following report represents the stunted children in Vietnam. The report broken down in two where it measures the stunting amongst the younger cohort and the older cohort. The stunting has three categories not stunted, stunted and severely stunted.

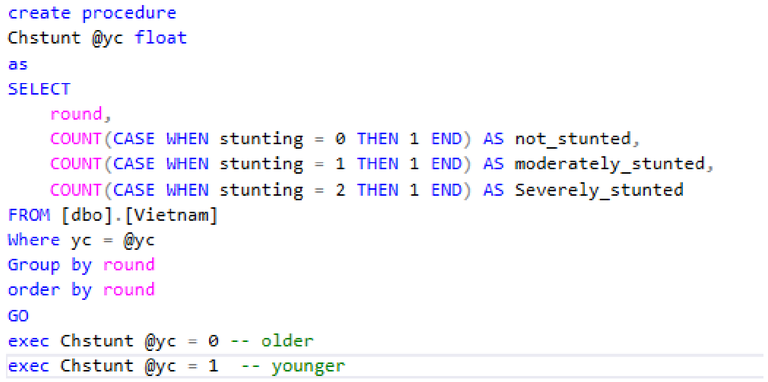
The following chart combines both the Older Cohort category and Younger Cohort category stunting data into one chart. The bar chart represents the older cohort category and the scatter with lines represents the Younger Cohort category.



Cohort	Stunted	Round 1	Round 2	Round 3	Round 4	Round 5
Older Cohort	Not Stunted	712	679	737	186	0
	Moderately Stunted	237	244	180	61	0
	Severely Stunted	51	65	50	7	0
Younger	Not Stunted	1570	1461	1547	1528	1700
	Moderately Stunted	300	399	326	288	198
	Severely Stunted	113	96	61	84	39

T-SQL statement

The following statement was used to retrieve the stunting data for both older cohort and younger cohort. A stored procedure named Chstunt was created taking @yc as a parameter. The purpose of this procedure is to count the three different categories of stunting for each round for the desired @yc parameter. The last two lines execute the procedure by taking a @yc parameter.



```
create procedure
Chstunt @yc float
as
SELECT
    round,
    COUNT(CASE WHEN stunting = 0 THEN 1 END) AS not_stunted,
    COUNT(CASE WHEN stunting = 1 THEN 1 END) AS moderately_stunted,
    COUNT(CASE WHEN stunting = 2 THEN 1 END) AS Severely_stunted
FROM [dbo].[Vietnam]
Where yc = @yc
Group by round
order by round
GO
exec Chstunt @yc = 0 -- older
exec Chstunt @yc = 1  -- younger
```

Database security

For the security of a database there two types of security to consider:

- **Data security** - When designing a database, it is important to protect data against malicious use or loss of data accidentally or intentionally. For any organisation data is a valuable and requires to be managed and controlled.
- **Database security –** a database must be protected against unauthorised access and those who wish to access the database with malicious intent. This can be both computer-based and non-computer based.

Some of the measures that can be taken are:

- **Authorisation:** enabling certain people to have certain access to specific areas of the database.
- **Authentication:** This is where person is required to sign in into a system to confirm they are who they say they are. This is usually done through a unique login and password only the user would know.
- **Views:** These are virtual tables that only show specified data from the database. This means it can restrict access and only show data that only needs to be seen.

The YoungLives database has strictly limited access to the physical server and hardware as well required user authentication to access database.

Database backup and restore strategy

It is important to have a backup and restore plan to prevent the loss of data by making sure the database designed to have the ability to recover lost or damaged data. This can be done through creating a copy of the database and being able to retrieve the copied database in the case of loss or damaged data. These are the different types of database backups in MSSQL:

Full Database Backup: This type of backup simply backups the entire database. This will include part of the transaction log so when restoring the database, the full database can be recovered. Every database contains on LDF file (transaction log file) and MDF file (data file) in which a full database backup will backup both these files

Differential Backup: This type of backup simply contains all the changes since the full last backup. The purpose of this backup is to reduce storage space and reduce the time it takes to backup

Transaction Log backup: This type of backup simply backup all transactions logs. Transactions logs as are basically a history of every modification made on the data in the database. This also requires a previous Full backup and log files cannot be recovered with a simple recovery model.

The YoungLives database has full database backup, where it contains a backup file with an LDF and MDF file. This means the database can recovered in case of loss or damaged data.

Ethical and Legal Data issues

Since the Young Lives dataset has been anonymised it does not need to comply with the Data Protection Act 1998. In addition, The Young Lives is an organisation in the public sector, therefore The UK Freedom Act (2000) allows a UK citizen request or obtain research data the company has.

Additionally, unique ideas used can be protected through intellectual property rights. This means that the design or look of products or the things written, made or produce by a person(s) can be protected legally against those who wish to copy or steal. To qualify for intellectual property rights:

- The invention is unique
- The invention must serve some type of purpose and can be used
- The invention involves an inventive step

However, if the child well being monitor is to be used commercially it will require permission from Young Lives. The UK Data Service where the Young Lives dataset is stored requires an administration fee to use the data. If the data is used is for non-commercial use e.g. publication, then a citation is required. However, this is where the ethical issue occurs. Even though, it is legal requirement to still acknowledge and ask permission to use public research for commercial use. It is still possible to disguise the research as one or not credit the data creators in a publication. It is important to ask permission from data creators as an organisation can face legal action and even patents can be refused as the organisation does not own the legal rights of the data. In addition, it is important to cite data used from data owners as enables tracking which demonstrates value to funders and potential refunding.

Business Intelligence Techniques

Business Intelligence is essentially the use a of various tools to analyse data that can help in decision making. In task 1, the data retrieved from T-SQL queries was converted to charts and tables using Excel.

Conclusion

In conclusion, 5 reports were created with sorting, filtering and group functionality as well as various search facilities for the child well-being monitor. These reports use data from the young lives dataset to analyse the causes and consequences of childhood poverty

Task 2

ABSTRACT

In this task the report will showcase the design and implementation using the Vietnam 2016-2017 dataset as well as using Excel to create a reporting system. The purpose of this is to analyse the inequality of the education sector in Vietnam.

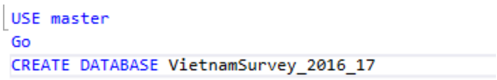
INTRODUCTION

Young Lives conducted a survey in two waves. The first wave of data was collected between September to October 2016 and the second between March to April 2017. The report is aim is to use the dataset and implement it into a SQL server and use T-SQL queries to analyse the data.

RELTIONAL SCHEMA

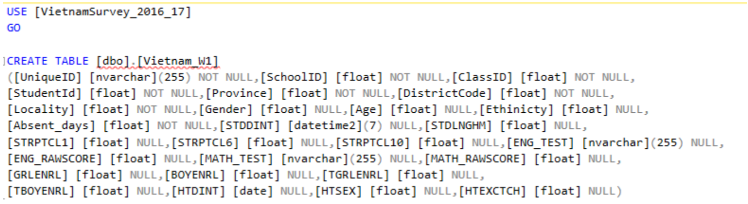
The database VietnamSurvey_2016_17 was created where it contains two tables Vietnam_w1 and Vietnam_w2.

T-SQL to create database



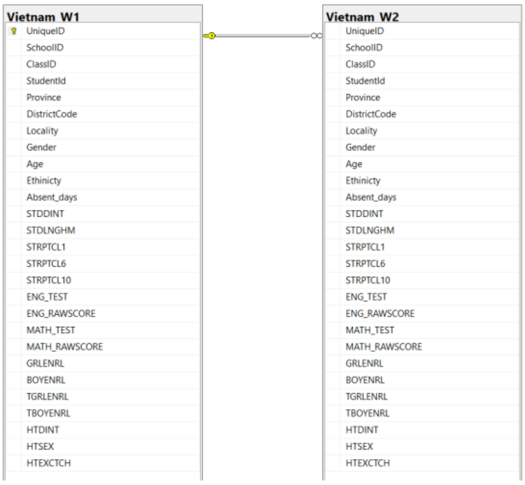
```
USE master
Go
CREATE DATABASE VietnamSurvey_2016_17
```

T-SQL used for Vietnam_w1



```
USE [VietnamSurvey_2016_17]
GO

CREATE TABLE [dbo].[Vietnam_W1]
([UniqueID] [nvarchar](255) NOT NULL,[SchoolID] [float] NOT NULL,[ClassID] [float] NOT NULL,
[StudentId] [float] NOT NULL,[Province] [float] NOT NULL,[DistrictCode] [float] NOT NULL,
[Locality] [float] NOT NULL,[Gender] [float] NULL,[Age] [float] NULL,[Ethinicty] [float] NULL,
[Absent_days] [float] NOT NULL,[STDDINT] [datetime2](7) NULL,[STDLNGHM] [float] NULL,
[STRPTCL1] [float] NULL,[STRPTCL6] [float] NULL,[STRPTCL10] [float] NULL,[ENG_TEST] [nvarchar](255) NULL,
[ENG_RAWSCORE] [float] NULL,[MATH_TEST] [nvarchar](255) NULL,[MATH_RAWSCORE] [float] NULL,
[GRLENRL] [float] NULL,[BOYENRL] [float] NULL,[TGRLENRL] [float] NULL,
[TBOYENRL] [float] NULL,[HTDINT] [date] NULL,[HTSEX] [float] NULL,[HTEXCTCH] [float] NULL)
```

Schema for database VietnamSurvey_2016_17



DESIGN CONSIDERATIONS

Normalisation

First Normal Form (1NF): VietnamSurvey_2016_17 database tables adhere to First Normal Form as the tables were created using MSSQL and only have a single valued columns/attributes.

Second Normal Form(2NF): The VietnamSurvey_2016_17 database tables adhere to Second Normal Form as all non-key columns for each table is dependent on the table's primary key and have no partial dependencies.

Third Normal Form(3NF): The tables in the VietnamSurvey_2016_17 meet the requirements of all three accepted normal, therefore data is normalised.

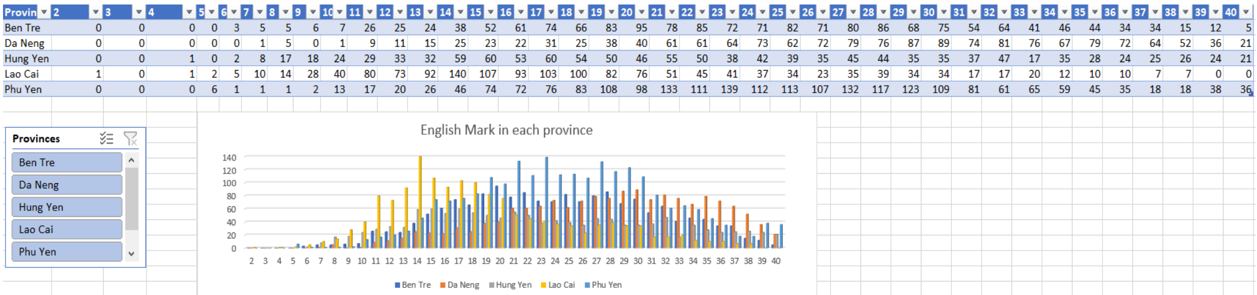
Constraints

The tables in VietnamSurvey_2016_17 database use three types of constraints :

- **Not Null Constraint**
- **Primary Key Constraint**
- **Foreign Key:** This a combination of columns to create a relationship with the data between two tables. The VietnamSurvey_2016_17 database uses a foreign key on the column UniqueID

REPORT 1

One of the ways to measure inequality in the education sector is to test cognitive performance of students. The following report gathers data of the English mark of each Young Lives child and compares against the different provinces. The English Marks ranges from 1-40

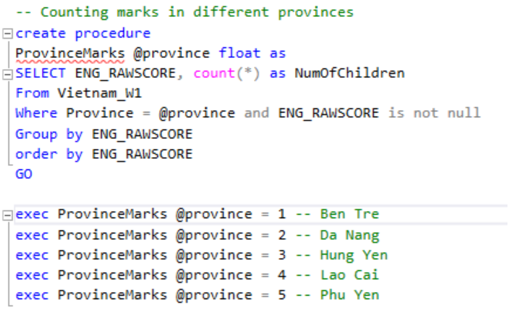


Provin	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ben Tre	0	0	0	0	3	5	5	6	7	26	25	24	38	52	61	74	66	83	95
Da Neng	0	0	0	0	0	1	5	0	1	9	11	15	25	23	22	31	25	38	40
Hung Yen	0	0	1	0	2	8	17	18	24	29	33	32	59	60	53	60	54	50	46
Lao Cai	1	0	1	2	5	10	14	28	40	80	73	92	140	107	93	103	100	82	76
Phu Yen	0	0	0	6	1	1	1	2	13	17	20	26	46	74	72	76	83	108	98

Provin	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Ben Tre	78	85	72	71	82	71	80	86	68	75	54	64	41	46	44	34	34	15	12	5
Da Neng	61	61	64	73	62	72	79	76	87	89	74	81	76	67	79	72	64	52	36	21
Hung Yen	55	50	38	42	39	35	45	44	35	35	37	47	17	35	28	24	25	26	24	21
Lao Cai	51	45	41	37	34	23	35	39	34	34	17	17	20	12	10	10	7	7	0	0
Phu Yen	133	111	139	112	113	107	132	117	123	109	81	61	65	59	45	35	18	18	38	36

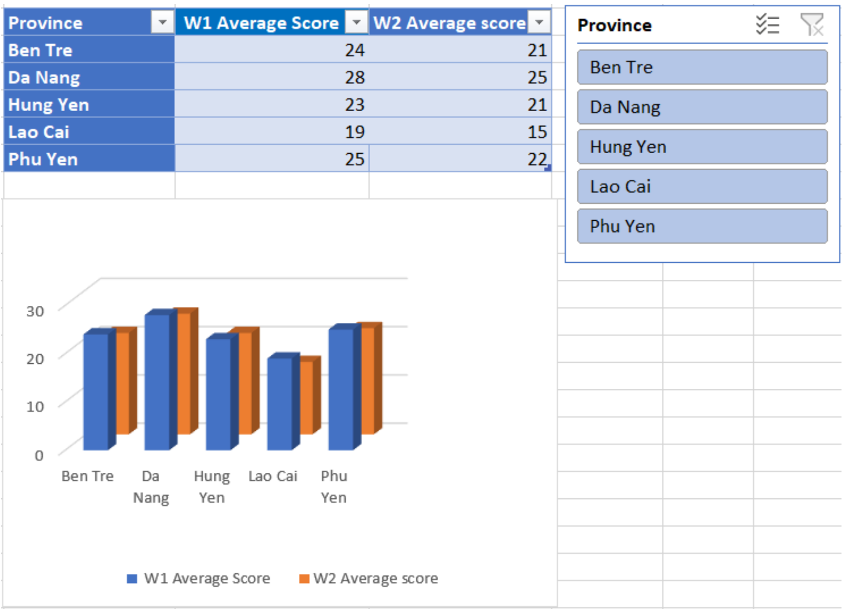
T-SQL statement

The following statement was used to retrieve the the range of English marks in different provinces. A stored procedure named ProvinceMarks was created taking @province as a parameter. The purpose of this procedure is the range of English marks for the desired province. The bottom five lines execute the procedure ProvinceMarks by taking a @province parameter.



```
-- Counting marks in different provinces
create procedure
ProvinceMarks @province float as
SELECT ENG_RAWSCORE, count(*) as NumOfChildren
From Vietnam_W1
Where Province = @province and ENG_RAWSCORE is not null
Group by ENG_RAWSCORE
order by ENG_RAWSCORE
GO

exec ProvinceMarks @province = 1 -- Ben Tre
exec ProvinceMarks @province = 2 -- Da Nang
exec ProvinceMarks @province = 3 -- Hung Yen
exec ProvinceMarks @province = 4 -- Lao Cai
exec ProvinceMarks @province = 5 -- Phu Yen
```

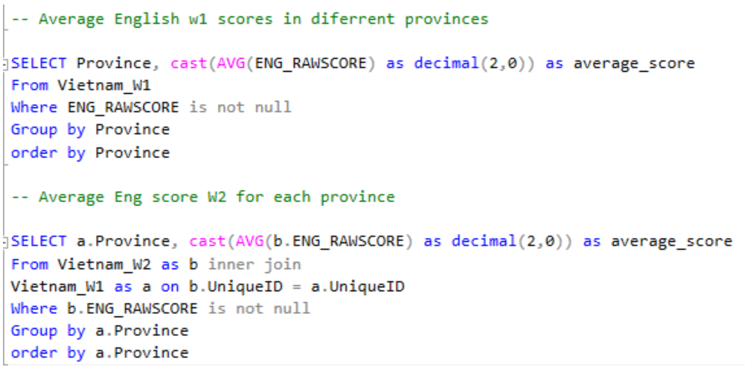
The following chart is summarised the range of English marks from each province by working out the average mark achieved in each province. In addition, the following chart analyses the difference of English result compared from Wave 1 to Wave 2 in averages



Province	W1 Average Score	W2 Average score
Ben Tre	24	21
Da Nang	28	25
Hung Yen	23	21
Lao Cai	19	15
Phu Yen	25	22

T-SQL statement

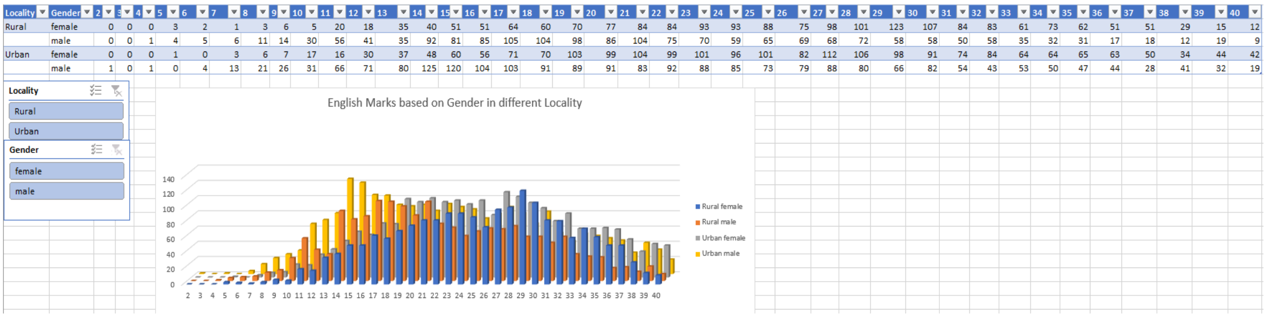
The following is two statements that retrieves average English marks for both wave 1 and wave 2 based on province. The first statement covert ENG_RAWSCORE is datatype, to be able to work out the average English Mark. The data is group by province there it will retrieve the average for each province. The second statement is like the first where average English score is being worked out for each province but for wave 2. However, to work out the average for wave 2 an inner join is required.



```
-- Average English w1 scores in different provinces

SELECT Province, cast(AVG(ENG_RAWSCORE) as decimal(2,0)) as average_score
From Vietnam_W1
Where ENG_RAWSCORE is not null
Group by Province
order by Province

-- Average Eng score W2 for each province

SELECT a.Province, cast(AVG(b.ENG_RAWSCORE) as decimal(2,0)) as average_score
From Vietnam_W2 as b inner join
Vietnam_W1 as a on b.UniqueID = a.UniqueID
Where b.ENG_RAWSCORE is not null
Group by a.Province
order by a.Province
```

REPORT 2

The following chart analyses education inequality by measuring the English marks between gender and the locality. This is done by retrieving the range of English marks by the two genders in rural and urban areas.

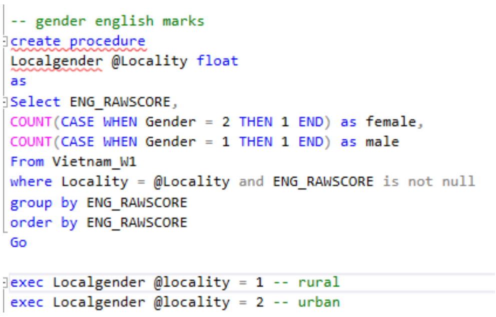


Locality	Gender	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Rural	female	0	0	0	3	2	1	3	6	5	20	18	35	40	51	51	64	60	70	77	84
	male	0	0	1	4	5	6	11	14	30	56	41	35	92	81	85	105	104	98	86	104
Urban	female	0	0	0	1	0	3	6	7	17	16	30	37	48	60	56	71	70	103	99	104
	male	1	0	1	0	4	13	21	26	31	66	71	80	125	120	104	103	91	89	91	83

Locality	Gender	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Rural	female	84	93	93	88	75	98	101	123	107	84	83	61	73	62	51	51	29	15	12
	male	75	70	59	65	69	68	72	58	58	50	58	35	32	31	17	18	12	19	9
Urban	female	99	101	96	101	82	112	106	98	91	74	84	64	64	65	63	50	34	44	42
	male	92	88	85	73	79	88	80	66	82	54	43	53	50	47	44	28	41	32	19

T-SQL statements

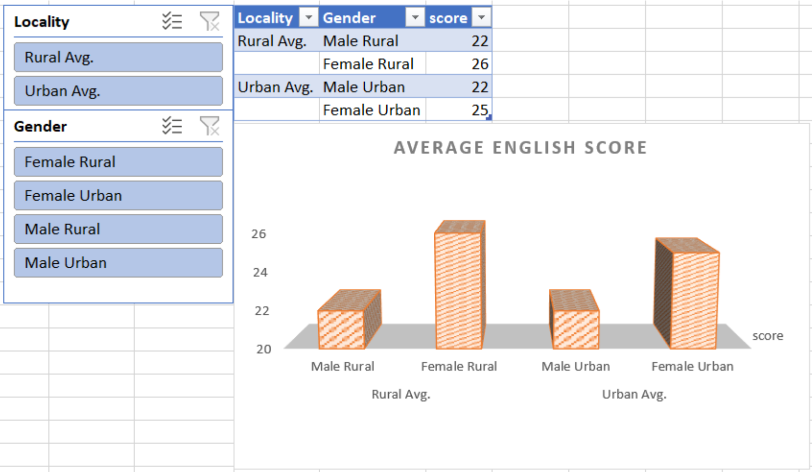
The following statement retrieved the range of English marks by the two genders in rural and urban areas. The stored procedure LocalGender wad created. The purpose of the store procedure is to count the total two genders achieving a range to 1-40 marks in the English test of the desired @locality parameter.



```
-- gender english marks
create procedure
Localgender @Locality float
as
Select ENG_RAWSCORE,
COUNT(CASE WHEN Gender = 2 THEN 1 END) as female,
COUNT(CASE WHEN Gender = 1 THEN 1 END) as male
From Vietnam_W1
where Locality = @Locality and ENG_RAWSCORE is not null
group by ENG_RAWSCORE
order by ENG_RAWSCORE
Go

exec Localgender @locality = 1 -- rural
exec Localgender @locality = 2 -- urban
```

Average Gender difference in English Marks

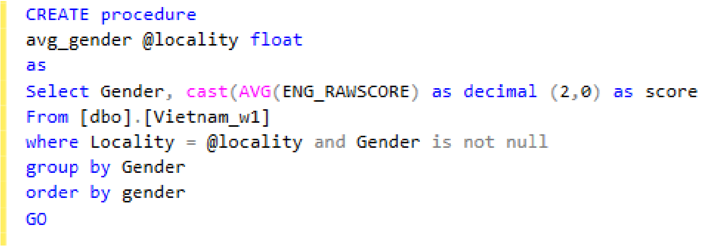
The following chart is a summarised version of data above where it works the averages of the English marks scored.



Locality	Gender	score
Rural Avg.	Male Rural	22
	Female Rural	26
Urban Avg.	Male Urban	22
	Female Urban	25

T-SQL statement

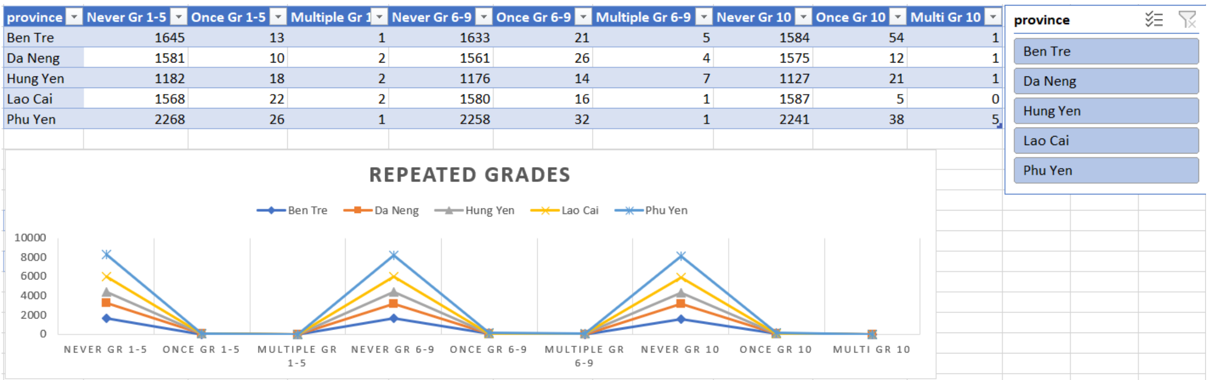
The following statement retrieves the average English score for the Gender in different locality. The stored procedure avg_gender is created, which calculates the average ENG_RAWSCORE for both gender for the given @locality parameter.



```
CREATE procedure
avg_gender @locality float
as
Select Gender, cast(AVG(ENG_RAWSCORE) as decimal (2,0) as score
From [dbo].[Vietnam_w1]
where Locality = @locality and Gender is not null
group by Gender
order by gender
GO
```

REPORT 3

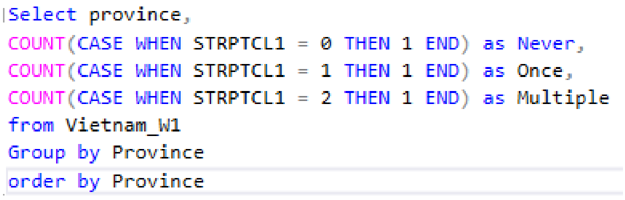
The following report analyses the amount children repeated their grades in different provinces. There are four grade ranges grade 1-5, 6-9, and 10. The repeated grade is broken down into 3 categories Never, Once and Multiple times.



province	Never Gr 1-5	Once Gr 1-5	Multiple Gr 1	Never Gr 6-9	Once Gr 6-9	Multiple Gr 6-9	Never Gr 10	Once Gr 10	Multi Gr 10
Ben Tre	1645	13	1	1633	21	5	1584	54	1
Da Neng	1581	10	2	1561	26	4	1575	12	1
Hung Yen	1182	18	2	1176	14	7	1127	21	1
Lao Cai	1568	22	2	1580	16	1	1587	5	0
Phu Yen	2268	26	1	2258	32	1	2241	38	5

T-SQL statements

The following statement counts the different type of repeated grade for grades 1-5. The data is group by province.



```
Select province,
COUNT(CASE WHEN STRPTCL1 = 0 THEN 1 END) as Never,
COUNT(CASE WHEN STRPTCL1 = 1 THEN 1 END) as Once,
COUNT(CASE WHEN STRPTCL1 = 2 THEN 1 END) as Multiple
from Vietnam_W1
Group by Province
order by Province
```

Database security

For the security of a database there two types of security to consider:

- **Data security**
- **Database Security**

Some of the measures that can be taken are:

- **Authorisation**
- **Authentication**
- **Views**

The VietnamSurvey_2016_17 database has strictly limited access to the physical server and hardware as well required user authentication to access database.

Database backup and restore strategy

The VietnamSurvey_2016_17 database has full database backup, where it contains a backup file with an LDF and MDF file. This means the database can recovered in case of loss or damaged data.

Ethical and Legal Data issues

The UK Freedom Act (2000) allows obtain research data for Young Lives Vietnam survey

To qualify for intellectual property rights:

- The invention is unique
- The invention must serve some type of purpose and can be used
- The invention involves an inventive step

For commercial use of data require permission from Young Lives. The UK Data Service where the Young Lives dataset is stored requires an administration fee to use the data.

Non-commercial use e.g. publication, then a citation is required.

Business Intelligence Techniques

Business Intelligence is essentially the use a of various tools to analyse data that can help in decision making.
In task 2, the data retrieved from T-SQL queries was converted to charts and tables using Excel.

Conclusion

To summarise the above, 3 reports were generated to analyse the inequality of the educational sector in Vietnam. MSSQL was used to create a database to store the dataset obtained from Young Lives Survey. Additionally, T-SQL queries were used to retrieve the appropriate data. MS excel was used to create charts and tables giving the ability to perform an analysis on the data.

Task 3

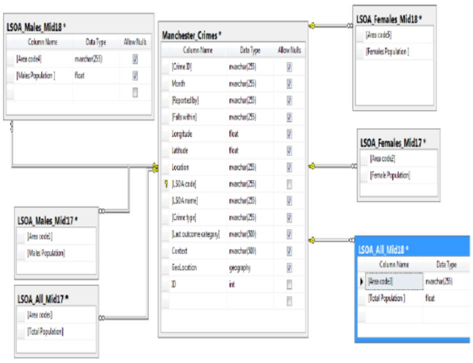
Abstract

In this task the Greater Manchester crimes dataset alongside the LSOA population dataset was used to create summarised LSOAs wise crime report with local population data in Greater Manchester between Jan 2017 and Dec 2018. In addition, QGIS software was used to create GIS Mapping to analyse crimes in the Greater Manchester Area

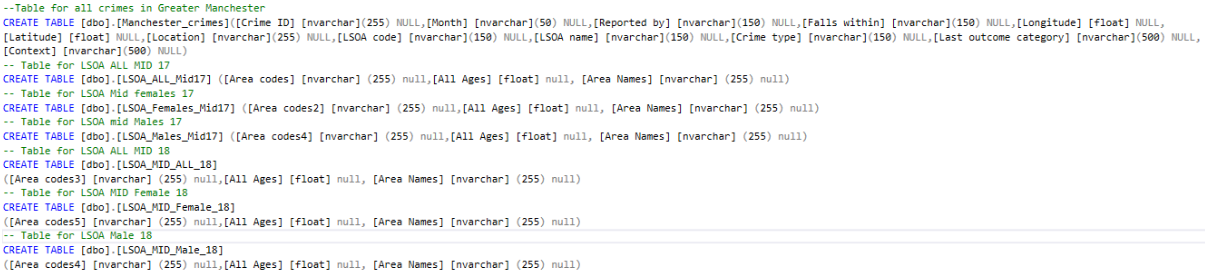
Introduction

The Greater Manchester dataset contains 14 different crime types alongside with the Longitude and Latitude of where each crime type took place. The LSAO dataset contains total population of each are in Greater Manchester. The follow report will analyse the crime report of the local population using GIS mapping.

Relational Schema



T SQL statements used for Tables



```
--Table for all crimes in Greater Manchester
CREATE TABLE [dbo].[Manchester_crimes]([Crime ID] [nvarchar](255) NULL,[Month] [nvarchar](50) NULL,[Reported by] [nvarchar](150) NULL,[Falls within] [nvarchar](150) NULL,[Longitude] [float] NULL,
[Latitude] [float] NULL,[Location] [nvarchar](255) NULL,[LSOA code] [nvarchar](150) NULL,[LSOA name] [nvarchar](150) NULL,[Crime type] [nvarchar](150) NULL,[Last outcome category] [nvarchar](500) NULL,
[Context] [nvarchar](500) NULL)
-- Table for LSOA ALL MID 17
CREATE TABLE [dbo].[LSOA_ALL_Mid17] ([Area codes] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
-- Table for LSOA Mid females 17
CREATE TABLE [dbo].[LSOA_Females_Mid17] ([Area codes2] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
-- Table for LSOA mid Males 17
CREATE TABLE [dbo].[LSOA_Males_Mid17] ([Area codes4] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
-- Table for LSOA ALL MID 18
CREATE TABLE [dbo].[LSOA_MID_ALL_18]
([Area codes3] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
-- Table for LSOA MID Female 18
CREATE TABLE [dbo].[LSOA_MID_Female_18]
([Area codes5] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
-- Table for LSOA Male 18
CREATE TABLE [dbo].[LSOA_MID_Male_18]
([Area codes4] [nvarchar] (255) null,[All Ages] [float] null, [Area Names] [nvarchar] (255) null)
```

DESIGN CONSIDERATIONS

Normalisation

First Normal Form (1NF): ManchesterCrimes database tables adhere to First Normal Form as the tables were created using MSSQL and only have a single valued columns/attributes.

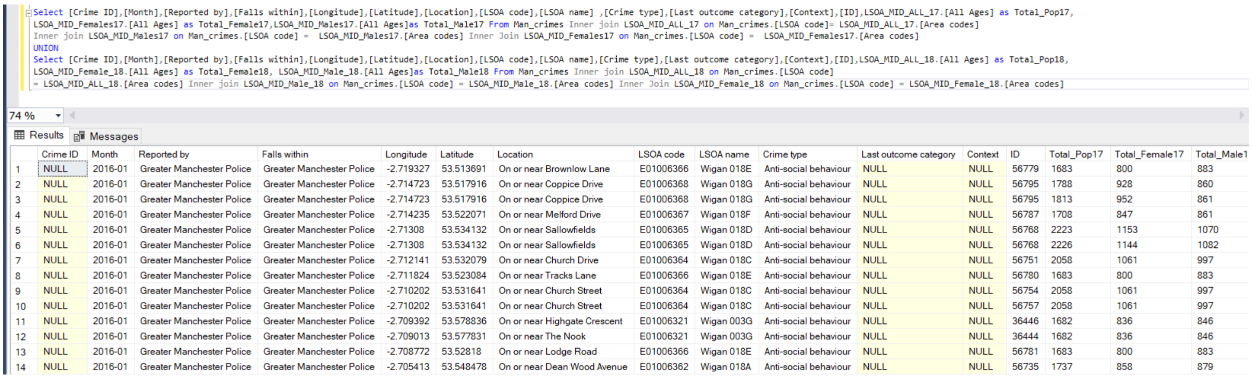
Second Normal Form(2NF): The ManchesterCrimes database tables adhere to Second Normal Form as all non-key columns for each table is dependent on the table's primary key and have no partial dependencies.

Third Normal Form(3NF): The tables in the ManchesterCrimes meet the requirements of all three accepted normal, therefore data is normalised.

Constraints Used

- **Not Null Constraint**
- **Primary Key Constraint**
- **Foreign Key**

A summarised LSOAs wise crime report with local population data in Greater Manchester between Jan 2017 and Dec 2018.

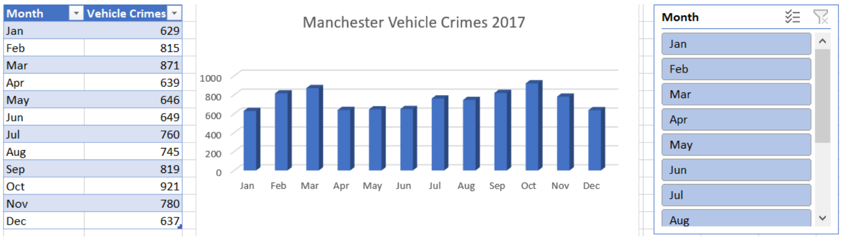


```
Select [Crime ID],[Month],[Reported by],[Falls within],[Longitude],[Latitude],[Location],[LSOA code],[LSOA name] ,[Crime type],[Last outcome category],[Context],[ID],LSOA_MID_ALL_17.[All Ages] as Total_Pop17,
LSOA_MID_Females17.[All Ages] as Total_Female17,LSOA_MID_Males17.[All Ages]as Total_Male17 From Man_crimes Inner join LSOA_MID_ALL_17 on Man_crimes.[LSOA code]= LSOA_MID_ALL_17.[Area codes]
Inner join LSOA_MID_Males17 on Man_crimes.[LSOA code] =  LSOA_MID_Males17.[Area codes] Inner Join LSOA_MID_Females17 on Man_crimes.[LSOA code] =  LSOA_MID_Females17.[Area codes]
UNION
Select [Crime ID],[Month],[Reported by],[Falls within],[Longitude],[Latitude],[Location],[LSOA code],[LSOA name],[Crime type],[Last outcome category],[Context],[ID],LSOA_MID_ALL_18.[All Ages] as Total_Pop18,
LSOA_MID_Female_18.[All Ages] as Total_Female18, LSOA_MID_Male_18.[All Ages]as Total_Male18 From Man_crimes Inner join LSOA_MID_ALL_18 on Man_crimes.[LSOA code]
= LSOA_MID_ALL_18.[Area codes] Inner join LSOA_MID_Male_18 on Man_crimes.[LSOA code] = LSOA_MID_Male_18.[Area codes] Inner Join LSOA_MID_Female_18 on Man_crimes.[LSOA code] = LSOA_MID_Female_18.[Area codes]
```

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type	Last outcome category	Context	ID	Total_Pop17	Total_Female17	Total_Male1
1	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.719327	53.513691	On or near Brownlow Lane	E01006366	Wigan 018E	Anti-social behaviour	NULL	NULL	56779	1683	800	883
2	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.714723	53.517916	On or near Coppice Drive	E01006368	Wigan 018G	Anti-social behaviour	NULL	NULL	56795	1788	928	860
3	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.714723	53.517916	On or near Coppice Drive	E01006368	Wigan 018G	Anti-social behaviour	NULL	NULL	56795	1813	952	861
4	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.714235	53.522071	On or near Melford Drive	E01006367	Wigan 018F	Anti-social behaviour	NULL	NULL	56787	1708	847	861
5	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.71308	53.534132	On or near Sallowfields	E01006365	Wigan 018D	Anti-social behaviour	NULL	NULL	56768	2223	1153	1070
6	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.71308	53.534132	On or near Sallowfields	E01006365	Wigan 018D	Anti-social behaviour	NULL	NULL	56768	2226	1144	1082
7	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.712141	53.532079	On or near Church Drive	E01006364	Wigan 018C	Anti-social behaviour	NULL	NULL	56751	2058	1061	997
8	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.711824	53.523084	On or near Tracks Lane	E01006366	Wigan 018E	Anti-social behaviour	NULL	NULL	56780	1683	800	883
9	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.710202	53.531641	On or near Church Street	E01006364	Wigan 018C	Anti-social behaviour	NULL	NULL	56754	2058	1061	997
10	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.710202	53.531641	On or near Church Street	E01006364	Wigan 018C	Anti-social behaviour	NULL	NULL	56757	2058	1061	997
11	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.709392	53.578836	On or near Highgate Crescent	E01006321	Wigan 003G	Anti-social behaviour	NULL	NULL	36446	1682	836	846
12	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.709013	53.577831	On or near The Nook	E01006321	Wigan 003G	Anti-social behaviour	NULL	NULL	36444	1682	836	846
13	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.708772	53.52818	On or near Lodge Road	E01006366	Wigan 018E	Anti-social behaviour	NULL	NULL	56781	1683	800	883
14	NULL	2016-01	Greater Manchester Police	Greater Manchester Police	-2.705413	53.548478	On or near Dean Wood Avenue	E01006362	Wigan 018A	Anti-social behaviour	NULL	NULL	56735	1737	858	879

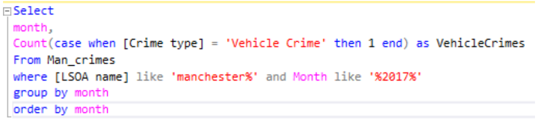
Report 1

The following chart is a representation of all vehicle crimes in Manchester for each month in the year 2017



Month	Vehicle Crimes
Jan	629
Feb	815
Mar	871
Apr	639
May	646
Jun	649
Jul	760
Aug	745
Sep	819
Oct	921
Nov	780
Dec	637

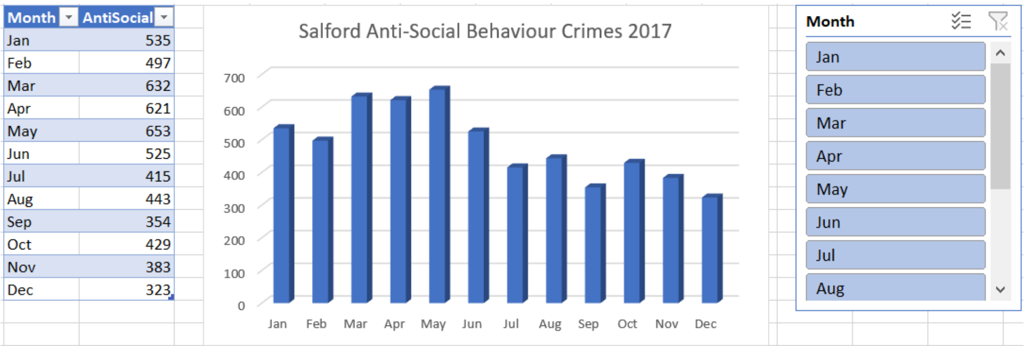
T-SQL statement



```
Select
month,
Count(case when [Crime type] = 'Vehicle Crime' then 1 end) as VehicleCrimes
From Man_crimes
where [LSOA name] like 'manchester%' and Month like '%2017%'
group by month
order by month
```

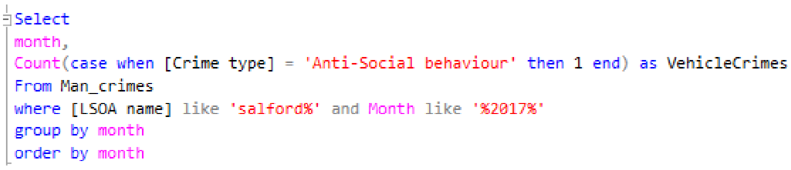
Report 2

The following chart is a representation of all anti-social behaviour crimes in Salford for each month in the year 2017



Month	AntiSocial
Jan	535
Feb	497
Mar	632
Apr	621
May	653
Jun	525
Jul	415
Aug	443
Sep	354
Oct	429
Nov	383
Dec	323

T-SQL statement



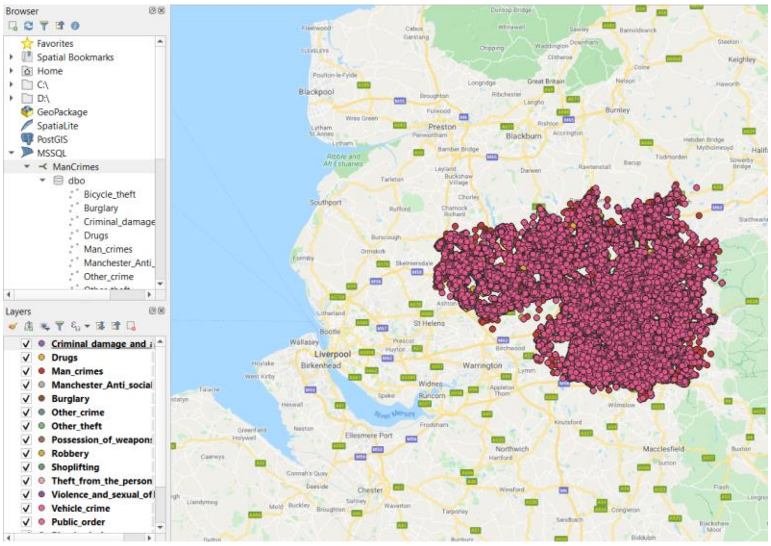
```
Select
month,
Count(case when [Crime type] = 'Anti-Social behaviour' then 1 end) as VehicleCrimes
From Man_crimes
where [LSOA name] like 'salford%' and Month like '%2017%'
group by month
order by month
```

Create Views Statement

```
-- Bicycle theft
CREATE VIEW [dbo].[Bicycle_theft]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Bicycle theft' AND
[GeoLocation] IS NOT NULL
-- Burglary
ALTER VIEW [dbo].[Burglary] AS
SELECT * FROM [dbo].Manchester_crimes] WHERE [Crime type]='Burglary' AND [GeoLocation]
IS NOT NULL
-- Criminal damage and arson
CREATE VIEW [dbo].[Criminal_damage_and_arson]
AS SELECT * FROM [dbo].[Manchester_crime]s WHERE [Crime type]='Criminal damage and
arson' AND [GeoLocation] IS NOT NULL
-- Drugs
CREATE VIEW [dbo].[Drugs]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Drugs' AND
[GeoLocation] IS NOT NULL
-- Anti-social behaviour
CREATE VIEW [dbo].[Manchester_Anti_social]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Anti-social behaviour'
AND [GeoLocation] IS NOT NULL
-- Other crime
CREATE VIEW [dbo].[Other_crime]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Other crime' AND
[GeoLocation] IS NOT NULL
-- Other theft
CREATE VIEW [dbo].[Other_theft]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Other theft' AND
[GeoLocation] IS NOT NULL
-- Possession of weapons
CREATE VIEW [dbo].[Possession_of_weapons]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Possession of weapons'
AND [GeoLocation] IS NOT NULL
-- Public order
CREATE VIEW [dbo].[Public_order]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Public order' AND
[GeoLocation] IS NOT NULL
-- Robbery
CREATE VIEW [dbo].[Robbery]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Robbery'
AND [GeoLocation] IS NOT NULL
-- Shoplifting
CREATE VIEW [dbo].[Shoplifting]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Shoplifting' AND
[GeoLocation] IS NOT NULL
-- Theft from the person
CREATE VIEW [dbo].[Theft_from_the_person]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Theft from the person'
AND [GeoLocation] IS NOT NULL
-- Vehicle crime
CREATE VIEW [dbo].[Vehicle_crime]
AS SELECT * FROM [dbo].Manchester_crimes WHERE [Crime type]='Vehicle crime' AND
[GeoLocation] IS NOT NULL
-- Violence and sexual offences
CREATE VIEW [dbo].[Violence_and_sexual_offences]
AS SELECT * FROM [dbo].[Manchester_crimes] WHERE [Crime type]='Violence and sexual
offences' AND [GeoLocation] IS NOT NULL
```

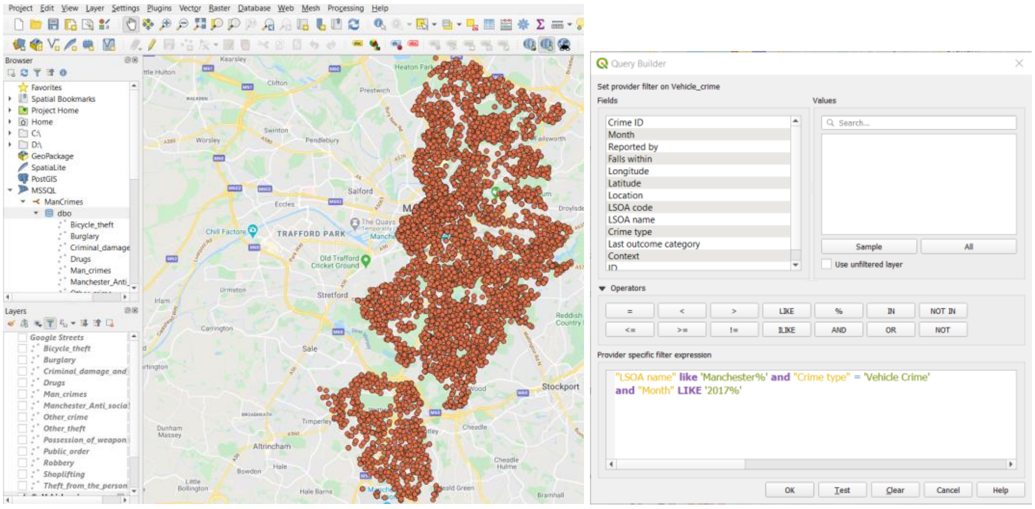
GIS Mapping of all Crimes in Manchester

The following visualises all crime sin Manchester with QGIS using MSSQL Connector. It Uses the OpenStreetMap that comes with the OpenLayers plugin.



GIS Mapping Filter – Manchester Vehicle Crimes

The following GIS Map uses QGIS through MSSQL connector to visualise Report 1



Set provider filter on Vehicle_crime

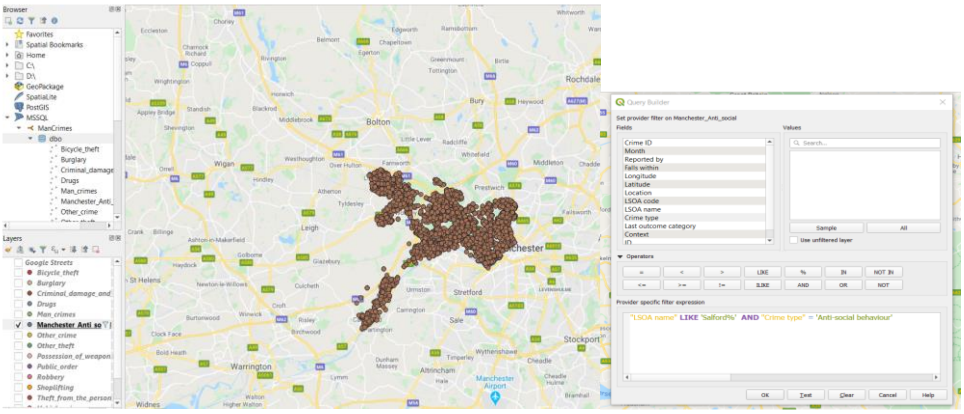
Provider specific filter expression

```
"LSOA name" like 'Manchester%' and "Crime type" = 'Vehicle Crime'
and "Month" LIKE '2017%'
```

This feature on the QGIS allows filtering, sorting, and grouping functionality. The GIS map below shows after the filter "LSOA name" like 'Manchester%' and "Crime type" = 'Vehicle Crime' and "Month" LIKE '2017%'.

GIS Mapping Filter Salford Anti-social crimes

The following GIS Map uses QGIS through MSSQL connector to visualise Report 2



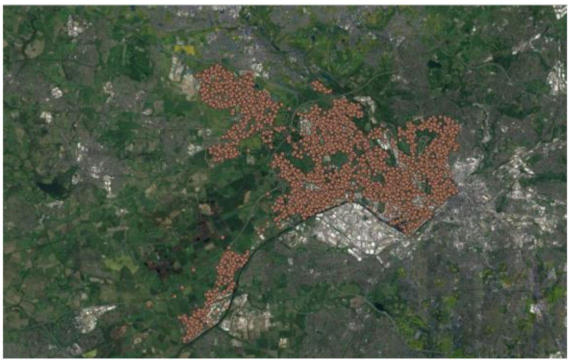
Set provider filter on Manchester_Anti_social

Provider specific filter expression

```
"LSOA name" LIKE 'Salford%'  AND "Crime type" = 'Anti-social behaviour'
```

Salford Anti-social crimes GIS satellite

The following is a satellite view of report 2



Database security

For the security of a database there two types of security to consider:

- **Data security**
- **Database Security**

Some of the measures that can be taken are:

- **Authorisation**
- **Authentication**
- **Views**

The ManchesterCrimes database has strictly limited access to the physical server and hardware as well required user authentication to access database.

Database backup and restore strategy

The ManchesterCrimes database has full database backup, where it contains a backup file with an LDF and MDF file. This means the database can recovered in case of loss or damaged data.

Ethical and Legal Data issues

To qualify for intellectual property rights:

- The invention is unique
- The invention must serve some type of purpose and can be used
- The invention involves an inventive step

Business Intelligence

The task used QGIS to analyses crime report within the local population. This allowed to generate QGIS maps to analyse the datasets.

Conclusion

To conclude, a report of a LSOA wise crime report was created using MSSQL to store and retrieve data. QGIS was used as a form of visualisation of the data to help better understand the crimes within the local population. The QGIS maps include sorting, filtering and grouping functionality for the reporting tool Crime profiler.

Bibliography

Websites

https://www.younglives.org.uk/content/india

https://www.unicef.org/ethiopia/press-releases/nearly-36-million-children-ethiopia-are-poor-and-lack-access-basic-social-services

https://www.theguardian.com/global-development/2012/jul/16/peru-eradicate-child-labour-education

Learning resources

Salford blackboard

Week 6: Database Recovery

Week 7- Database Security

Week 8: Professional, Legal, and Ethical Issues in Data Management

Week 9 - SQL Server Business Intelligence tools