LEVEL 7: MSC IN DATA SCIENCE

COURSE: ADVANCED DATABASE

PROGRAMME G430 M0052

STEPHEN OKPIAIFOH

STUDENT ID: @00643629

Table of Contents

5. T-SQL Statements
6. Report Design
7. Extra Features Implemented
8. Database backup and Restore Strategy
9. Database Science/Business Intelligence Techniques
10. Database Privacy
11. Conclusion

TASK 1

Abstract

The Young Lives project is a long-term study of how child poverty is changing in four emerging countries. The purpose of the project is to better know the cause and implications of childhood poverty, and to also investigate how laws affect children's well-being, to better inform future policy creation and better focus child welfare efforts. Ethiopia, India, Peru, and Vietnam have all agreed to take part in the study. These countries were chosen because they reflect a wide range of cultural, geopolitical, and economical situations, as well as other issues facing the poorer nations, such as large debt burden, post-conflict reconstruction, and deterioration of the environment like drought and flood. Over the course of a year, the Young Lives research will utilize the lives of 12,000 youngsters.

The study's three essential characteristics are a kid questionnaire, a household survey, and a community survey. Other cross-sectional datasets, such as the World Bank's Standard of Living Conduct various, are like the data collected in the house. Discussed in the following include household makeup, employment and wealth, household consumption, child health and basic services access, and education. There are also queries about caregiver viewpoints, emotions, and goals for their child and family. Young Lives also collects thorough time-use data for all family and friends, as well as statistics on the child's weight and height (as well as caregivers') and gives children school-related exams (language comprehension and mathematics). The children are questioned about their everyday routines, which is an essential aspect of the research. Boyden, J. (2018). *Young Lives: An International Study of Childhood Poverty: Round 2, 2006*. [data collection]. *3rd Edition.* UK Data Service. SN: 6852, DOI: 10.5255/UKDA-SN-6852-3

1. Introduction

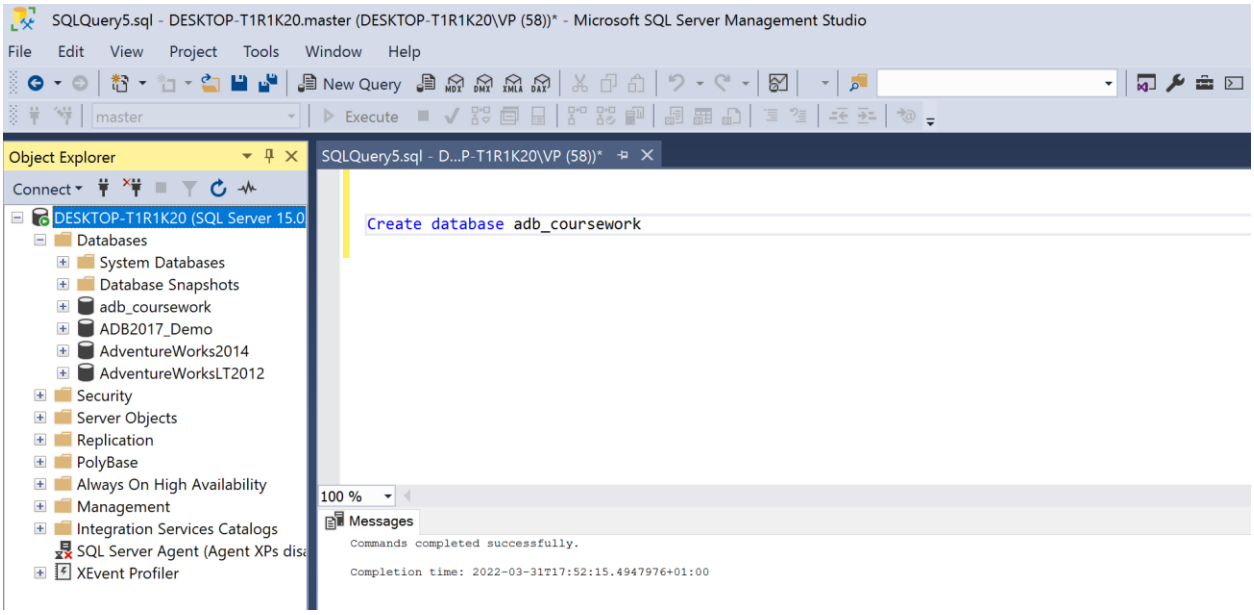
In the battle against poverty as an income gap, child poverty is commonly defined as children who live in low-income families and are well below the poverty line. However, it is important to recognize that a simple economic study of child poverty is insufficient. Poverty affects a cognitive upbringing because it determines their attitudes, behaviors, attitudes, and goals, as well as their physical, social, emotional, and spiritual well-being.

The goal of this work is to create a low-cost custom application reporting tool called Child Well-Being Monitor for analyzing data on child poverty. Software developers can quickly create front-end reporting tools. The Child Well-Being Monitor is a low-cost, simple-to-use web-based reporting tool for examining child poverty. It allows users to retrieve and visualize data in plots, tables, and other visualizations, making it easier for them to find what they need. Advanced Databases Assignment - Develop Specialised Knowledge - Business Assignment Help (tvassignmenthelp.com)

2. Importing Data

Data was imported from Young Lives web site through the international study of Childhood poverty full round series as a tab file into my computer. Tab file was converted and saved as comma delimiter file in my system. Each file was named by the country name for easy identification in SQL Server management studio, as India_table, Ethiopia_table, Peru_table, Vietnam_table.

Firstly, is to create a Database to load the dataset into SQL Server and using the insert and create query statement to load data to SQL studio.



Creating Peru table using create and insert query as peru_table

```
USE [adb_coursework]
GO
/****** database Object:  Table [dbo].[peru_table]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[peru_table](
    [childid] [nvarchar](50) NULL,
    [yc] [float] NULL,
    [round] [float] NULL,
    [inround] [float] NULL,
    [panel12345] [float] NULL,
    [deceased] [float] NULL,
```

Creating Vietnam table using create and insert query as vietnam_table

```
USE [adb_coursework]
GO
/******database Object:  Table [dbo].[vietnam_table]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[vietnam_table](
    [childid] [nvarchar](50) NULL,
    [yc] [float] NULL,
    [round] [float] NULL,
    [inround] [float] NULL,
    [panel12345] [float] NULL,
    [deceased] [float] NULL,
```

Creating India table using create and insert query as india_table

```
USE [adb.coursework]
GO
/****** Object:  Table [dbo].[india_table]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[india_table](
    [childid] [nvarchar](50) NULL,
    [yc] [float] NULL,
    [round] [float] NULL,
    [inround] [float] NULL,
    [panel12345] [float] NULL,
    [deceased] [float] NULL,
    [dint] [date] NULL,
    [clustid] [float] NULL,
    [commid] [nvarchar](50) NULL,
    [region] [float] NULL,
    [typesite] [float] NULL,
    [childloc] [float] NULL,
    [chsex] [float] NULL,
    [chlang] [float] NULL,
    [chethnic] [float] NULL,
    [chldrel] [float] NULL,
    [agemon] [float] NULL,
    [marrcohab] [float] NULL,
    [marrcohab_age] [float] NULL,
    [birth] [float] NULL,
```

Creating Ethiopia table using create and insert query as ethiopia_table

```
USE [adb_coursework]
GO
/******database Object:  Table [dbo].[ethiopia_table]    Sc
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ethiopia_table](
    [childid] [nvarchar](50) NULL,
    [yc] [float] NULL,
    [round] [float] NULL,
    [inround] [float] NULL,
    [panel] [float] NULL,
    [deceased] [float] NULL,
    [dint] [datetime2](7) NULL,
    [commid] [nvarchar](50) NULL,
    [clustid] [float] NULL,
    [region] [float] NULL,
    [typesite] [float] NULL,
    [childloc] [float] NULL,
    [chsex] [float] NULL,
    [chlang] [float] NULL,
    [chethnic] [float] NULL,
    [chldrel] [float] NULL,
    [agemon] [numeric](18, 0) NULL,
    [marrcohab] [float] NULL,
    [marrcohab_age] [float] NULL,
    [birth] [float] NULL,
```

3. Relational Schema

3.1 Identifying the tables

The below tables were identified based on this task from the metadata of Childhood poverty constructed files from round 1-5 guide. The various column names of each table were used to create the table in SQL Server management studio.

1.Child Identification

2.Child Anthropometric Information

3.Child biological parent feature

4.Childbirth Immunization

5. Child Education

6.household Wealth index and sub-indices

7.Child household Property and ownership

8. Household credit, food security

9.Child time use

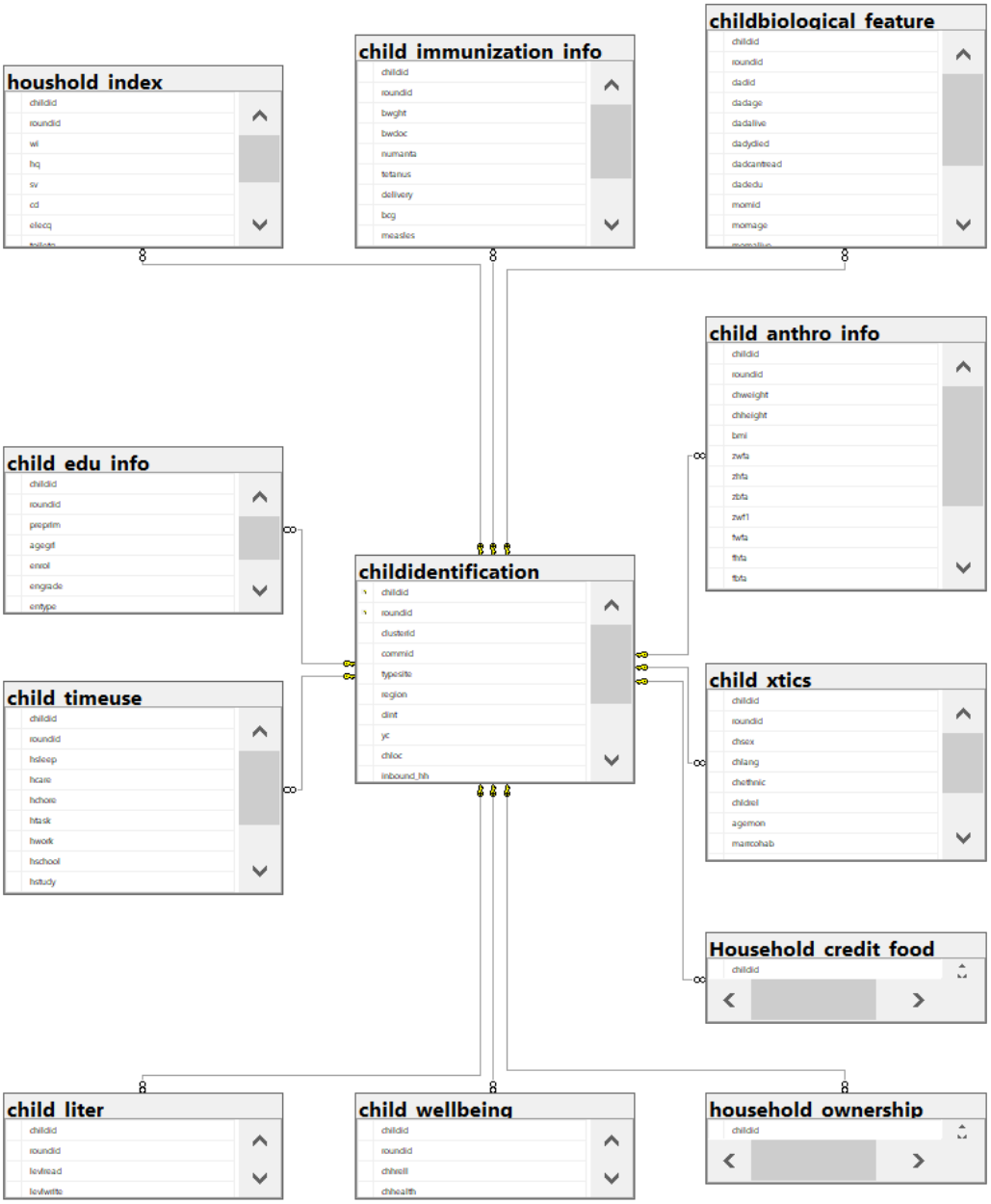
10.Child Subjective well being

11.Child Literature level

12. Child General Feature

3.2 Creating of Normalize table as in section 3.1 above.

3.3 Create relationships between tables



4. Design Rationale

The purpose of this project is to create dependable databases to increase company performance and efficiency. Every business is dependent on database dependability, and it is up to a skilled database developer to provide those firms with new database remedies. It also entails learning how to use the most up-to-date tools and technologies, such as Power Bi, to construct suitable architectures when performing data analysis, as well as thought to provide a simple and standardized database with consistency and integrity. All the tables were interconnected through

one interaction to guarantee the database was relational. The major key table that most of the tables (particularly additional primary key tables in various schemas) referenced was the Basic Child info, Demographic groups table, which contained basic information on the child. A composite primary key was built from the columns, since only the child id and round columns provide data uniqueness. The main keys functioned as foreign keys for other tables in the database, providing several more connections.

5. Design Considerations

5.1 Database Normalization

The technique of efficiently structuring data in a database is known as database normalization. The normalization process is motivated by two factors: Getting rid of unnecessary data, such as storing the same information in multiple tables. Assuring that data dependencies are logical. A childid and roundid column was also introduced to various tables for simple row and column identifying.

Both are worthwhile objectives since they minimize database size and confirm sure data is kept logically. Normalization is a set of rules that will assist you in building a suitable structure of the database.

Normalization is the process of organizing the data. Normalization is a method for cutting the levels of duplication in a partnership or group of relationships. Data collection, upgrade, and deletion are all characteristics that can be disabled. Standardizing separates the main table into separate tables and links them together using connections. The standard process is used to minimize table in the database repetition.

5.2 Constraints

Constraints are rules that are applied to data columns in a table. These are used to restrict the types of information that can be entered into a table. This ensures the database's data is accurate and trustworthy. Constraints can be applied at the column or table level. Top level restrictions are used throughout table, whereas columns level constraints are now only applied to one column. The limits are defined in DDL commands such as "alter" and "create" to guarantee that when an authorized user changes the database, the integrity of data is not disrupted. There are various constraints inside a database system: https://www.geeksforgeeks.org/domain-constraints-in-dbms/.

To this database the below constraints were used

NOT NULL: This was used on the primary key columns. This ensures that null values are not allowed on the columns.

PRIMARY KEY: The main key constraint imposed here was a combination primary key, influenced by the nature of the dataset. On tables that served as references to other items in the database, the primary key was required.

FOREIGN KEY: The foreign key constraint enforced on all the tables except the demographics table was to ensure that all tables were related to another.

CHECK: The check constraint enforced here ensured that before a data is put into a table, the existence of that data on the primary key table is checked and an error thrown if the data within the foreign key column is not present in the primary key table.

UNIQUE Constraint: The UNIQUE Constraint prevents two records from having identical values in a particular column table, for example, you might want to prevent two or more people from having identical age.

5.3 Data Validation

Data verification is a process of checking the accuracy and quality of data. It is accomplished by introducing numerous checks into such a system or report to verify logical consistency of input and local storage. With no or little human interference, data entered automated processes. As a result, it is vital to ensure that the data entering the system is correct and meets the appropriate quality standards. It will be of little benefit if the data is not captured accurately, and it may generate significant downstream reporting issues. Even if digital evidence is submitted correctly, it will cost money to clean, transform, and store it.

For this task, the below types of validation were applied

1. Data Type Check, this confirms that the data entered has the correct data type.

2. Code check a field is selected from a valid list of values or format

3. Range check was to ensure input data falls within a predefined range.

4. Format Check was to ensure data follows a predefined format.

5. Consistency check to ensure data is entered in a logically consistent way.

- Transaction and Concurrency Control (if any)

- Error Handling

5.4 Security

Encryption process is a group of techniques for protecting database systems from harmful information security and unauthorized access. Database leading security guard against abuse, corruption, and invasion not only of the data in a database, but also of the database management system and any programs that access it. A strong, unique password is required for each authorized access on a host machine. If an account is no longer being used, it should be closed and locked. To streamline access to this type of data and monitoring station, I will utilize an identity administration or entry management solution. Every time a user wants to access a database, they would be given just one password with the necessary rights.

Encoding and storage are inadequate to safeguard the confidential material that powers your business. Security professionals must know where sensitive data is housed, who has accessibility to it, but when it is misused to take timely action.

5.5 Comments

Comments are used to explain sections of SQL statements, or to prevent some part of SQL statement from execution. For this task I used single line comments starting with --- and end with ---, any statement between the single line will not executed.

Multi-line comment was also used in my SQL statements starting with /* and end with */, any statement in between the slash will not ignored.

6. T-SQL Statements

The T-SQL statements used in inserting values from Ethiopia ,Peru, Vietnam, and India tables to the tables identified in section 3 which was used to create the entity relationship diagram are show below with values inserted. Also, for Stored procedures, triggers, and views

T-SQL Statements for Tables

```
--- inserting data from india,peru,ethopia and vietnam table into childidentification table---

insert into dbo.childidentification(
childid, roundid, clusterid, commid, typesite,region,
dint, yc, chloc, inround_ch, panel, deceased)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(clustid as tinyint),
cast(commid as nvarchar(15)),
cast(typesite as tinyint),
cast(region as tinyint),
cast(dint as date),
cast (yc as bit),
cast(childloc as bit),
cast(inround as bit),
cast(panel12345 as bit),
cast(deceased as bit) from dbo.india_table
```

```sql
insert into dbo.childidentification(
childid, roundid, clusterid, commid, typesite,region,
dint, yc, chloc, inround_ch, panel, deceased)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(clustid as tinyint),
cast(placeid as nvarchar(15)),
cast(typesite as tinyint),
cast(region as tinyint),
cast(dint as date),
cast (yc as bit),
cast(childloc as bit),
cast(inround as bit),
cast(panel12345 as bit),
cast(deceased as bit) from dbo.peru_table
```

```sql
insert into dbo.childidentification(
childid, roundid, clusterid, commid, typesite,region,
dint, yc, chloc, inround_ch, panel, deceased)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(clustid as tinyint),
cast(commid as nvarchar(15)),
cast(typesite as tinyint),
cast(region as tinyint),
cast(dint as date),
cast (yc as bit),
cast(childloc as bit),
cast(inround as bit),
cast(panel as bit),
cast(deceased as bit) from dbo.ethiopia_table
```

```
insert into dbo.childidentification(
childid, roundid, clusterid, commid, typesite,region,
dint, yc, chloc, inround_ch, panel, deceased)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(clustid as tinyint),
cast(commid as nvarchar(15)),
cast(typesite as tinyint),
cast(region as tinyint),
cast(dint as date),
cast (yc as bit),
cast(childloc as bit),
cast(inround as bit),
cast(panel12345 as bit),
cast(deceased as bit) from dbo.vietnam_table
```

```
---inserting values from india,peru,ethopia and vietnam table into child_anthro_info table---

insert into dbo.child_anthro_info(
childid, roundid, chweight, chheight, bmi,zwfa, zhfa,zbfa, zwf1,fwfa,fhfa,fbfa,fwf1,underweight,stunting,thinness)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast (chweight as smallint),
cast(chheight as smallint),
cast(bmi as decimal(25,20)),
cast(zwfa as decimal(25,20)),
cast(zhfa as decimal(25,20)),
cast(zbfa as decimal(25,20)),
cast(zwfl as decimal(25,20)),
cast(fwfa as tinyint),
cast(fhfa as tinyint),
cast(fbfa as tinyint),
cast(fwfl as tinyint),
cast(underweight as tinyint),
cast(stunting as tinyint),
cast(thinness as tinyint) from dbo.ethiopia_table
```

```
insert into dbo.child_anthro_info(
childid, roundid, chweight, chheight, bmi,zwfa, zhfa,zbfa, zwf1,fwfa,fhfa,fbfa,fwf1,underweight,stunting,thinness)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast (chweight as smallint),
cast(chheight as smallint),
cast(bmi as decimal(25,20)),
cast(zwfa as decimal(25,20)),
cast(zhfa as decimal(25,20)),
cast(zbfa as decimal(25,20)),
cast(zwfl as decimal(25,20)),
cast(fwfa as tinyint),
cast(fhfa as tinyint),
cast(fbfa as tinyint),
cast(fwfl as tinyint),
cast(underweight as tinyint),
cast(stunting as tinyint),
cast(thinness as tinyint) from dbo.peru_table
```

```
insert into dbo.child_anthro_info(
childid, roundid, chweight, chheight, bmi,zwfa, zhfa,zbfa, zwf1,fwfa,fhfa,fbfa,fwf1,underweight,stunting,thinness)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast (chweight as smallint),
cast(chheight as smallint),
cast(bmi as decimal(25,20)),
cast(zwfa as decimal(25,20)),
cast(zhfa as decimal(25,20)),
cast(zbfa as decimal(25,20)),
cast(zwfl as decimal(25,20)),
cast(fwfa as tinyint),
cast(fhfa as tinyint),
cast(fbfa as tinyint),
cast(fwfl as tinyint),
cast(underweight as tinyint),
cast(stunting as tinyint),
cast(thinness as tinyint) from dbo.india_table
```

```
insert into dbo.child_anthro_info(
childid, roundid, chweight, chheight, bmi,zwfa, zhfa,zbfa, zwf1,fwfa,fhfa,fbfa,fwf1,underweight,stunting,thinness)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast (chweight as smallint),
cast(chheight as smallint),
cast(bmi as decimal(25,20)),
cast(zwfa as decimal(25,20)),
cast(zhfa as decimal(25,20)),
cast(zbfa as decimal(25,20)),
cast(zwfl as decimal(25,20)),
cast(fwfa as tinyint),
cast(fhfa as tinyint),
cast(fbfa as tinyint),
cast(fwfl as tinyint),
cast(underweight as tinyint),
cast(stunting as tinyint),
cast(thinness as tinyint) from dbo.vietnam_table
```

```sql
---insert vaues from peru,indian,vietnam and ethopia table into child_immunization_info table---

insert into dbo.child_immunization_info
(childid, roundid, bwght, bwdoc, numanta, tetanus, delivery, bcg, measles, polio, dpt, hib)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(bwght as int),
cast(bwdoc as bit),
cast(numante as tinyint),
cast(tetanus as bit),
cast(delivery as bit),
cast(bcg as bit),
cast(measles as bit),
cast(polio as bit),
cast(dpt as bit),
cast(hib as bit) from dbo.ethiopia_table
```

```sql
insert into dbo.child_immunization_info
(childid, roundid, bwght, bwdoc, numanta, tetanus, delivery, bcg, measles, polio, dpt, hib)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(bwght as int),
cast(bwdoc as bit),
cast(numante as tinyint),
cast(tetanus as bit),
cast(delivery as bit),
cast(bcg as bit),
cast(measles as bit),
cast(polio as bit),
cast(dpt as bit),
cast(hib as bit) from dbo.peru_table
```

```sql
insert into dbo.child_immunization_info
(childid, roundid, bwght, bwdoc, numanta, tetanus, delivery, bcg, measles, polio, dpt, hib)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(bwght as int),
cast(bwdoc as bit),
cast(numante as tinyint),
cast(tetanus as bit),
cast(delivery as bit),
cast(bcg as bit),
cast(measles as bit),
cast(polio as bit),
cast(dpt as bit),
cast(hib as bit) from dbo.vietnam_table
```

```
insert into dbo.child_immunization_info
(childid, roundid, bwght, bwdoc, numanta, tetanus, delivery, bcg, measles, polio, dpt, hib)
select cast(childid as nvarchar(15)),
cast (round as tinyint),
cast(bwght as int),
cast(bwdoc as bit),
cast(numante as tinyint),
cast(tetanus as bit),
cast(delivery as bit),
cast(bcg as bit),
cast(measles as bit),
cast(polio as bit),
cast(dpt as bit),
cast(hib as bit) from dbo.india_table
```

```
---inserting values from peru,india,vietnam and ethiopia table into child_xtics table---

insert into dbo.child_xtics
(childid, roundid, chsex, chlang, chethnic, chldrel, agemon, marrcohab, marrcohab_age, birth, birth_age)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chsex as tinyint),
cast(chlang as tinyint),
cast(chethnic as tinyint),
cast(chldrel as tinyint),
cast(agemon as smallint),
cast (marrcohab as bit),
cast (marrcohab_age as tinyint),
cast(birth as bit),
cast(birth_age as tinyint) from dbo.india_table
```

```
insert into dbo.child_xtics
(childid, roundid, chsex, chlang, chethnic, chldrel, agemon, marrcohab, marrcohab_age, birth, birth_age)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chsex as tinyint),
cast(chlang as tinyint),
cast(chethnic as tinyint),
cast(chldrel as tinyint),
cast(agemon as smallint),
cast (marrcohab as bit),
cast (marrcohab_age as tinyint),
cast(birth as bit),
cast(birth_age as tinyint) from dbo.peru_table
```

```sql
insert into dbo.child_xtics
(childid, roundid, chsex, chlang, chethnic, chldrel, agemon, marrcohab, marrcohab_age, birth, birth_age)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chsex as tinyint),
cast(chlang as tinyint),
cast(chethnic as tinyint),
cast(chldrel as tinyint),
cast(agemon as smallint),
cast (marrcohab as bit),
cast (marrcohab_age as tinyint),
cast(birth as bit),
cast(birth_age as tinyint) from dbo.vietnam_table
```

```sql
insert into dbo.child_xtics
(childid, roundid, chsex, chlang, chethnic, chldrel, agemon, marrcohab, marrcohab_age, birth, birth_age)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chsex as tinyint),
cast(chlang as tinyint),
cast(chethnic as tinyint),
cast(chldrel as tinyint),
cast(agemon as smallint),
cast (marrcohab as bit),
cast (marrcohab_age as tinyint),
cast(birth as bit),
cast(birth_age as tinyint) from dbo.ethiopia_table
```

```sql
---inserting values from peru,vietnam,ethiopia and india table into child_wellbeing table---

insert into dbo.child_wellbeing
(childid, roundid, chhrel, chhealth, cladder)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chhrel as tinyint),
cast(chhealth as tinyint),
cast(cladder as tinyint) from dbo.ethiopia_table

insert into dbo.child_wellbeing
(childid, roundid, chhrel, chhealth, cladder)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chhrel as tinyint),
cast(chhealth as tinyint),
cast(cladder as tinyint) from dbo.peru_table
```

```
insert into dbo.child_wellbeing
(childid, roundid, chhrel, chhealth, cladder)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chhrel as tinyint),
cast(chhealth as tinyint),
cast(cladder as tinyint) from dbo.vietnam_table

insert into dbo.child_wellbeing
(childid, roundid, chhrel, chhealth, cladder)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(chhrel as tinyint),
cast(chhealth as tinyint),
cast(cladder as tinyint) from dbo.india_table
```

```
-- inserting data from peru,india,ethiopia table and vietnam into table child_timeuse table---

insert into dbo.child_timeuse
(childid, roundid, hsleep, hcare,hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(hsleep as int),
cast(hcare as int),
cast(hchore as int),
cast (htask as int),
cast(hwork as int),
cast(hschool as int),
cast(hstudy as int),
cast(hplay as int),
cast(commwork as int),
cast(commsch as int) from dbo.ethiopia_table
```

```
insert into dbo.child_timeuse
(childid, roundid, hsleep, hcare,hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(hsleep as int),
cast(hcare as int),
cast(hchore as int),
cast (htask as int),
cast(hwork as int),
cast(hschool as int),
cast(hstudy as int),
cast(hplay as int),
cast(commwork as int),
cast(commsch as int) from dbo.peru_table
```

```
insert into dbo.child_timeuse
(childid, roundid, hsleep, hcare,hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(hsleep as int),
cast(hcare as int),
cast(hchore as int),
cast (htask as int),
cast(hwork as int),
cast(hschool as int),
cast(hstudy as int),
cast(hplay as int),
cast(commwork as int),
cast(commsch as int) from dbo.vietnam_table
```

```
insert into dbo.child_timeuse
(childid, roundid, hsleep, hcare,hchore, htask, hwork, hschool, hstudy, hplay, commwork, commsch)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(hsleep as int),
cast(hcare as int),
cast(hchore as int),
cast (htask as int),
cast(hwork as int),
cast(hschool as int),
cast(hstudy as int),
cast(hplay as int),
cast(commwork as int),
cast(commsch as int) from dbo.vietnam_table
```

```
--- insert values from peru,india,vietnam and ethiopia table into Household_credit_food table ---

insert into dbo.Household_credit_food
(childid, roundid, credit, foodsec)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(credit as bit),
cast(foodsec as tinyint) from dbo.ethiopia_table

insert into dbo.Household_credit_food
(childid, roundid, credit, foodsec)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(credit as bit),
cast(foodsec as tinyint) from dbo.peru_table


insert into dbo.Household_credit_food
(childid, roundid, credit, foodsec)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(credit as bit),
cast(foodsec as tinyint) from dbo.vietnam_table

insert into dbo.Household_credit_food
(childid, roundid, credit, foodsec)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(credit as bit),
cast(foodsec as tinyint) from dbo.india_table



---insert values from peru,vietnam,ethiopia and india table into household_ownership table---

insert into dbo.household_ownership
(childid, roundid, ownlandhse, ownhouse)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(ownlandhse as bit),
cast(ownhouse as bit) from dbo.ethiopia_table

insert into dbo.household_ownership
(childid, roundid, ownlandhse, ownhouse)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(ownlandhse as bit),
cast(ownhouse as bit) from dbo.peru_table

insert into dbo.household_ownership
(childid, roundid, ownlandhse, ownhouse)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(ownlandhse as bit),
cast(ownhouse as bit) from dbo.vietnam_table
```

```
insert into dbo.household_ownership
(childid, roundid, ownlandhse, ownhouse)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(ownlandhse as bit),
cast(ownhouse as bit) from dbo.india_table
```

```
---inserting values from peru,ethiopia,vietnam and india table into childbiological_feature table---

insert into dbo.childbiological_feature
(childid, roundid, dadid, dadage, dadalive, dadydied, dadcantread, dadedu, momid, momage,momalive,momyrdied,
momcantread,momedu)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(dadid as tinyint),
cast (dadage as smallint),
cast(dadlive as tinyint),
cast(dadyrdied as tinyint),
cast(dadcantread as bit),
cast(dadedu as tinyint),
cast(momid as tinyint),
cast(momage as smallint),
cast(momlive as tinyint),
cast(momyrdied as tinyint),
cast(momcantread as bit),
cast(momedu as tinyint) from dbo.peru_table
```

```
insert into dbo.childbiological_feature
(childid, roundid, dadid, dadage, dadalive, dadydied, dadcantread, dadedu, momid, momage,momalive,momyrdied,
momcantread,momedu)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(dadid as tinyint),
cast (dadage as smallint),
cast(dadlive as tinyint),
cast(dadyrdied as tinyint),
cast(dadcantread as bit),
cast(dadedu as tinyint),
cast(momid as tinyint),
cast(momage as smallint),
cast(momlive as tinyint),
cast(momyrdied as tinyint),
cast(momcantread as bit),
cast(momedu as tinyint) from dbo.india_table
```

```
insert into dbo.childbiological_feature
(childid, roundid, dadid, dadage, dadalive, dadydied, dadcantread, dadedu, momid, momage,momalive,momyrdied,
momcantread,momedu)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(dadid as tinyint),
cast (dadage as smallint),
cast(dadlive as tinyint),
cast(dadyrdied as tinyint),
cast(dadcantread as bit),
cast(dadedu as tinyint),
cast(momid as tinyint),
cast(momage as smallint),
cast(momlive as tinyint),
cast(momyrdied as tinyint),
cast(momcantread as bit),
cast(momedu as tinyint) from dbo.vietnam_table
```

```
insert into dbo.childbiological_feature
(childid, roundid, dadid, dadage, dadalive, dadydied, dadcantread, dadedu, momid, momage,momalive,momyrdied,
momcantread,momedu)
select cast(childid as nvarchar(15)),
cast(round as tinyint),
cast(dadid as tinyint),
cast (dadage as smallint),
cast(dadlive as tinyint),
cast(dadyrdied as tinyint),
cast(dadcantread as bit),
cast(dadedu as tinyint),
cast(momid as tinyint),
cast(momage as smallint),
cast(momlive as tinyint),
cast(momyrdied as tinyint),
cast(momcantread as bit),
cast(momedu as tinyint) from dbo.ethiopia_table
```

T-SQLS STATEMENTS FOR SUMMARY REPORTS AND THE TABLES

REPORT 1

```sql
/** Report to measure the child reading abilities in various regions in ethiopia and typesite(urban ar Rural)
from child identification table
and child literature table in round 3 **/

select count(*) as "total children",
case chid.typesite
when 1 then 'urban'
when 2 then 'rural'
else 'unknown' end as 'type of residence',
case chid.region
when 1 then ' Tigray'
when 3 then 'Amhara'
when 5 then 'Somali'
when 12 then 'Gambela'
else 'unknown region' end as 'area',
case lit.levlread
when 1 then 'child cannot read anything'
when 2 then 'child can read letters'
when 3 then 'child can read words'
else 'child can read sentence' end as 'reading abilities'
from dbo.childidentification  as chid join dbo.child_liter as lit with (updlock)
on chid.childid = lit.childid
where chid.region in(1,3,5,12) and lit.levlread is not null
and chid.roundid = 3
group by chid.typesite,chid.region,lit.levlread
order by [total children]
```

Concurrency Control: concurrency control is used to keep the SQL statements in the database consistent even when various users access the data, in this task (updlock) was used as concurrency control.

Results | Messages

	total children	type of residence	area	reading abilities
1	17	urban	Tigray	child can read words
2	19	urban	Amhara	child can read letters
3	21	urban	Amhara	child can read words
4	35	rural	Amhara	child can read words
5	37	urban	Amhara	child cannot read anything
6	42	urban	Tigray	child can read letters
7	51	rural	Tigray	child can read words
8	55	urban	Tigray	child cannot read anything
9	80	rural	Amhara	child can read letters
10	88	urban	Tigray	child can read sentence
11	94	rural	Tigray	child can read letters
12	118	urban	Amhara	child can read sentence
13	135	rural	Tigray	child can read sentence
14	156	rural	Amhara	child can read sentence
15	283	rural	Amhara	child cannot read anything
16	296	rural	Tigray	child cannot read anything

Region Tigray in Ethiopia at round three which means the children are between the ages of 10 - 12 cannot read anything, probably because the region is in the rural setting compared to same Tigray in the urban area.

REPORT 2

```
/** Report to measure food situation and credit level in peru taking some regions and Area of residence,
(urban/rural) into consideration,joining childidentification and household_credit_food table in roundid 4 **/


select count(*) as 'number of children',
case chid.typesite
when 1 then 'urban'
when 2 then 'rural'
else 'not known'  end as 'Area of Residence',
case chid.region
when 32 then 'Sierra'
when 33 then 'Selva'
when 31 then 'Costa'
else 'no region' end as 'region of Residence',
case cf.credit
when 0 then 'no'
```

```
when 1 then 'yes' end as 'credit level',
case cf.foodsec
when 1 then 'we always eat enough of what we want'
when 2 then 'we eat enough but not always what we would like'
when 3 then 'we sometimes do not eat enough'
when 4 then 'we frequently do not eat enough' end as 'food situation'
from dbo.childidentification as chid join dbo.Household_credit_food as cf with (updlock)
on chid.childid = cf.childid
where chid.region in (32,33,31) and  cf.foodsec is not null and cf.credit is not null
and chid.roundid = 4
group by chid .typesite,chid.region,cf.credit,cf.foodsec
order by  [number of children]
```

00 %

Results Messages

	number of children	Area of Residence	region of Residence	credit level	food situation
1	1	rural	Costa	yes	we sometimes do not eat enough
2	1	rural	Selva	yes	we frequently do not eat enough
3	2	urban	Selva	yes	we frequently do not eat enough
4	6	rural	Costa	no	we frequently do not eat enough
5	8	urban	Selva	no	we frequently do not eat enough
6	8	urban	Sierra	yes	we frequently do not eat enough
7	9	urban	Selva	yes	we sometimes do not eat enough
8	13	rural	Costa	yes	we always eat enough of what we want
9	13	urban	Costa	yes	we frequently do not eat enough
10	16	rural	Selva	yes	we sometimes do not eat enough
11	17	rural	Sierra	no	we frequently do not eat enough

This report indicates that the total number of children in both rural and urban areas of this region of residence are all on credit level and frequently do not eat enough.

REPORT 3

```
/** Report to measure food situation and credit level in india taking some regions and Area of residence,
(urban/rural) into consideration,joining childidentification and household_credit_food table in roundid 4
using where to filter roundid, region,coloumns that are null,then group by the non-aggregated columns and finally
order by number of children**/

 select count(*) as 'number of children',
case chid.typesite
when 1 then 'urban'
when 2 then 'rural'
else 'not known'  end as 'Area of Residence',
case chid.region
when 24 then 'Others'
when 23 then 'Telangana' end as 'region of residence',
case cf.credit
when 0 then 'no'
when 1 then 'yes' end as 'credit level',
```

```
when 0 then 'no'
when 1 then 'yes' end as 'credit level',
case cf.foodsec
when 1 then 'we always eat enough of what we want'
when 2 then 'we eat enough but not always what we would like'
when 3 then 'we sometimes do not eat enough'
when 4 then 'we frequently do not eat enough' end as 'food situation'
from dbo.childidentification as chid join dbo.Household_credit_food as cf with (updlock)
on chid.childid = cf.childid
where chid.region in (24,23) and  cf.foodsec is not null and cf.credit is not null
and chid.roundid = 4
group by chid .typesite,chid.region,cf.credit,cf.foodsec
order by  [number of children]
```

sults | Messages

number of children	Area of Residence	region of residence	credit level	food situation
1	rural	Others	yes	we always eat enough of what we want
1	rural	Others	no	we eat enough but not always what we would like
1	urban	Others	yes	we sometimes do not eat enough
1	urban	Others	no	we eat enough but not always what we would like
1	not known	Telangana	yes	we always eat enough of what we want
1	urban	Others	yes	we always eat enough of what we want
1	rural	Others	yes	we eat enough but not always what we would like
1	not known	Telangana	no	we always eat enough of what we want
2	urban	Telangana	no	we frequently do not eat enough
5	rural	Telangana	no	we frequently do not eat enough
5	not known	Telangana	yes	we eat enough but not always what we would like
3	urban	Others	yes	we eat enough but not always what we would like

Report three also shows children in India of various residence and region do not eat enough of what they want, most are also on credit level

REPORT 4

```
 /** Report to measure food situation and credit level in Vietnam taking some regions and Area of residence,
(urban/rural) into consideration,joining childidentification and household_credit_food table in roundid 4
using where to filter roundid, region,coloumns that are null,then group by the non-aggregated columns and finally
order by number of children**/

 select count(*) as 'number of children',
case chid.typesite
when 1 then 'urban'
when 2 then 'rural'
else 'not known'  end as 'Area of Residence',
case chid.region
when 51 then 'Northern Uplands'
when 52 then 'Red River Delta'
when 55 then ' Highlands'
when 56 then 'South Eastern' end as 'region of residence',
case cf.credit
```

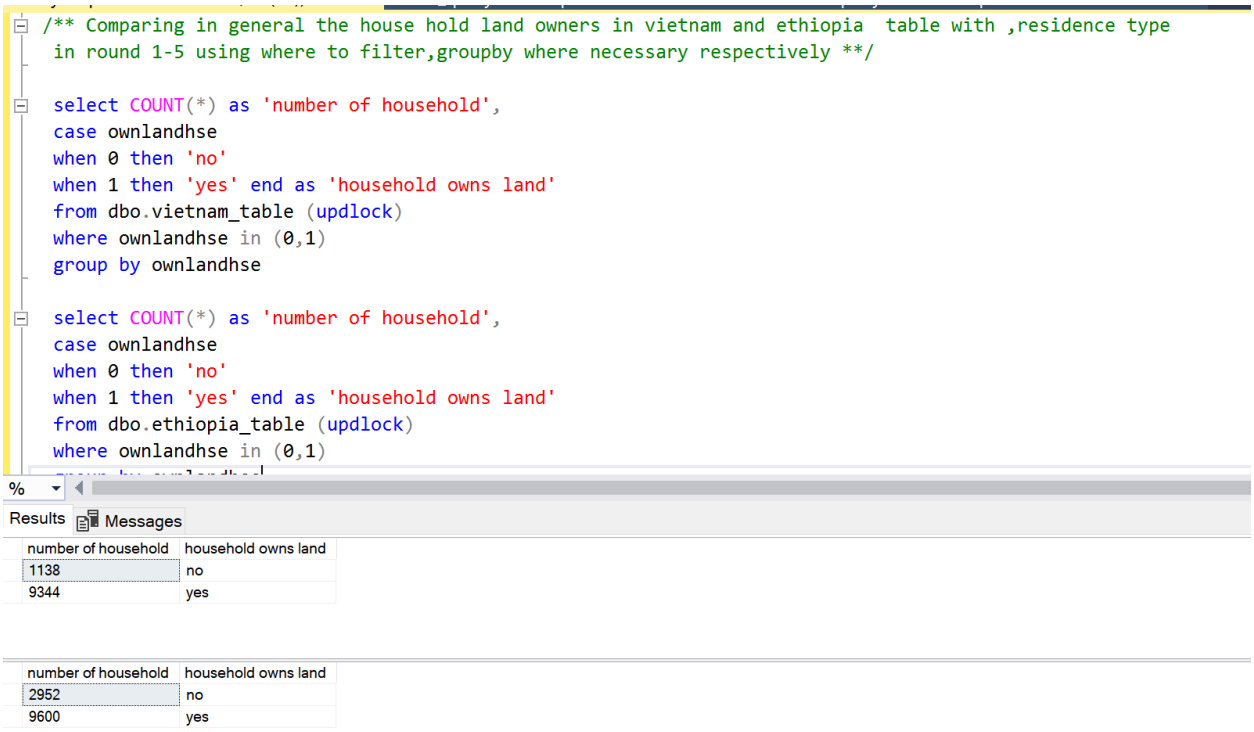
```
case cf.foodsec
when 1 then 'we always eat enough of what we want'
when 2 then 'we eat enough but not always what we would like'
when 3 then 'we sometimes do not eat enough'
when 4 then 'we frequently do not eat enough' end as 'food situation'
from dbo.childidentification as chid join dbo.Household_credit_food as cf with (updlock)
on chid.childid = cf.childid
where chid.region in (51,52,55,56) and  cf.foodsec is not null and cf.credit is not null
and chid.roundid = 4
group by chid .typesite,chid.region,cf.credit,cf.foodsec
```

sults Messages

umber of children	Area of Residence	region of residence	credit level	food situation
	urban	Northern Uplands	no	we sometimes do not eat enough
2	urban	Red River Delta	yes	we eat enough but not always what we would like
2	urban	Northern Uplands	yes	we sometimes do not eat enough
4	urban	Northern Uplands	no	we eat enough but not always what we would like
4	urban	Northern Uplands	yes	we eat enough but not always what we would like
5	urban	Red River Delta	no	we eat enough but not always what we would like
5	urban	Red River Delta	yes	we always eat enough of what we want
8	urban	Red River Delta	no	we always eat enough of what we want
9	urban	Northern Uplands	yes	we always eat enough of what we want
15	urban	Northern Uplands	no	we always eat enough of what we want
18	rural	Red River Delta	no	we frequently do not eat enough

Vietnam also shows the childhood poverty is high children usually do not get much food to eat as desired.

REPORT 5



```
/** Comparing in general the house hold land owners in vietnam and ethiopia  table with ,residence type
 in round 1-5 using where to filter,groupby where necessary respectively **/

 select COUNT(*) as 'number of household',
 case ownlandhse
 when 0 then 'no'
 when 1 then 'yes' end as 'household owns land'
 from dbo.vietnam_table (updlock)
 where ownlandhse in (0,1)
 group by ownlandhse

 select COUNT(*) as 'number of household',
 case ownlandhse
 when 0 then 'no'
 when 1 then 'yes' end as 'household owns land'
 from dbo.ethiopia_table (updlock)
 where ownlandhse in (0,1)
```

%

Results | Messages

number of household	household owns land
1138	no
9344	yes

number of household	household owns land
2952	no
9600	yes

General measure of the household that owns a land where they live, is obvious both in Vietnam and Ethiopia household own a land.

7. Extra Features Implemented in this task

```sql
---Creating a View called child_zscore_information from india_table considering region of residence,
---childsex and zscore measurement---

create view dbo.child_zscore_information
as(
select round(SUM(zwfa),2) as 'total weight zscore',round(SUM(zhfa),2) as 'total height zscore',
round(SUM(zbfa),2) as 'BMI-for-age z-score',
round(SUM(zwfl),2) as 'Weight-for-length/height z-score',
case chsex
when 1 then 'male'
when 2 then 'female' end as 'gender',
case region
when 23 then 'Telangana' end as 'region of residence'
from india_table
where chsex is not null and region is not null
group by chsex, region)
```

100 %

Results Messages

	total weight zscore	total height zscore	BMI-for-age z-score	Weight-for-length/height z-score	gender	region of residence
1	-1734.47	-4326.77	-1430.59	-460.52	female	NULL
2	-1873.44	-3077.31	-2181.1	-351.29	female	NULL
3	-14.67	-13.85	-16.03	NULL	male	NULL
4	NULL	-17.37	-2	NULL	female	NULL
5	-1921.01	-3295.51	-2321.83	-334.55	female	Telangana
6	NULL	-13	-6.02	NULL	male	NULL
7	-2708.58	-4197.24	-3795.4	-529.99	male	Telangana
8	-1883.47	-2844.22	-2262.24	-531.47	male	NULL

```sql
---- create a view dbo.child_vital_info to show vital sign of child in vietnam---
create view dbo.child_vital_info1
as (
select ROUND(sum(chweight),2) as 'sum weight',ROUND(sum(chheight),2) as 'sum height',
ROUND(sum(bmi),2) as 'sum of bmi',
case chsex when 1 then 'male' else 'female' end as 'gender',
case typesite when 1 then 'urban' else 'rural' end as 'area of residence'
from dbo.vietnam_table
group by chsex,typesite)
```

%

Results Messages

sum weight	sum height	sum of bmi	gender	area of residence
160228.9	702765.7	91983.76	female	rural
170367.45	727729.05	94665.29	male	rural
59266.25	219891.95	29543.16	male	urban
51306.3	201768.3	27411.26	female	urban
2413.1	8182.7	1042.75	female	rural
570.4	1948.5	251.61	male	rural

```sql
---- create a view dbo.child_vital_info to show vital sign of child in india for selected typesite and gender---
create view dbo.child_vital_info
as (
select ROUND(sum(chweight),2) as 'sum weight',ROUND(sum(chheight),2) as 'sum height',
ROUND(sum(bmi),2) as 'sum of bmi',
case chsex when 1 then 'male' else 'female' end as 'gender',
case typesite when 1 then 'urban' else 'rural' end as 'area of residence'
from dbo.india_table
group by chsex,typesite)
```

Results Messages

sum weight	sum height	sum of bmi	gender	area of residence
127447.26	622447.65	91283.36	female	rural
264.7	1398.9	159.67	male	rural
NULL	NULL	NULL	female	rural
149735.85	691412.03	90974.15	male	rural
65090.45	270711.38	34233.27	male	urban
57080.87	235214.7	31265.1	female	urban
315.8	1451	172.91	female	rural
880.7	3400	397.28	female	rural

Stored Procedures

```sql
---create stored procedure  called child_height to show child height from child_anthro_info table, this stored
----  procedure will extract height quickly from large table by assigning value of height to show---

create procedure
dbo.child_height
@chheight decimal
as select * from dbo.child_anthro_info
where chheight = @chheight

 execute dbo.child_height 79
```

Results Messages

childid	roundid	chweight	chheight	bmi	zwfa	zhfa	zbfa
ET130008	1	9.00000000000000000000	79.00000000000000000000	14.85999999999999943157	-1.42999999999999993783	-1.10000000000000008882	-1.08000000000000
ET130054	1	10.00000000000000000000	79.00000000000000000000	17.26000000000000156319	0.46000000000000001998	0.05999999999999999778	0.58999999999999996
ET140051	1	10.00000000000000000000	79.00000000000000000000	16.01999999999999957367	-0.30999999999999999778	-0.13000000000000000444	-0.30999999999999995
ET150007	1	11.00000000000000000000	79.00000000000000000000	17.62999999999999900524	2.41999999999999992895	3.87999999999999989342	0.55000000000000004
ET150062	1	11.00000000000000000000	79.00000000000000000000	18.10999999999999943157	0.66000000000000003109	-0.44000000000000000222	1.26000000000000000
ET150070	1	9.00000000000000000000	79.00000000000000000000	14.41999999999999992895	-0.95999999999999996447	-0.40000000000000002220	-1.06000000000000005
ET150087	1	10.00000000000000000000	79.00000000000000000000	16.17999999999999971578	0.46999999999999997335	1.48999999999999999112	-0.48999999999999995
ET150100	1	9.00000000000000000000	79.00000000000000000000	15.22000000000000063949	0.14999999999999999445	1.11000000000000009770	-0.68000000000000004
ET160001	1	9.00000000000000000000	79.00000000000000000000	15.01999999999999957367	-0.88000000000000000444	-0.11000000000000000056	-1.13999999999999996

The above stored procedure was able to display for only children that height fall into 79

```
--- Creating Stored Procedure from Vietnam table-----
---Trigger if the the Access to electricity is no and ownhouse is yes-----

create procedure identification_ownhouse
as
select childid,
case ownhouse
when 0 then 'no'
when 1 then 'yes' end as 'ownhouse group',
case elecq_new
when 0 then 'no'
when 1 then 'yes' end as 'Access to electricity',
case region
when 57 then 'Mekong River Delta'
when 56 then 'South Eastern' end as 'Region of residence'
from dbo.vietnam_table
where region = 57
```

0 %

Messages

```
Commands completed successfully.

Completion time: 2022-04-18T20:02:16.8671426+01:00
```

Executing the stored procedure for functionality

```
execute identification_ownhouse
```

```
execute identification_ownhouse
```

100 %

Results | Messages

	childid	ownhouse group	Access to electricity	Region of residence
1	VN011003	yes	yes	Mekong River Delta
2	VN011003	no	yes	Mekong River Delta
3	VN040016	no	yes	Mekong River Delta
4	VN040069	yes	yes	Mekong River Delta
5	VN040122	no	yes	Mekong River Delta
6	VN050001	NULL	yes	Mekong River Delta
7	VN050001	NULL	yes	Mekong River Delta

Triggers: The above Stored Procedure has trigger which will trigger if access to electricity is "no" and ownhouse is "yes"

8. Considering data protection issues as part of the design and implementation of my solution

The data protection act 2018 which currently governs the data protection in the UK, as well as the general data protection regulations(GDPR) and other related legislations. These laws affect how we gather, store , use data and individuals right over access to information. These laws were duly considered in implementing this task.

9. Database Security

Data is crucial in today's digital economy, and safeguarding data throughout cloud adoption is essential for minimizing risk. There are diverse ways to secure database to eliminate and reduce risk.

Alignment of security and innovation: To be effective, data architectures change quickly and become more complicated, necessitating a shift in focus away from users and systems.

Access to information about data activity: Safe data migration is accelerated by enterprise-wide visibility.

Protect critical data in one place: Data protection necessitates the discovery, classification, monitoring, and enforcement of policies.

Reduce response time and speed resolution, integrate application and data security with enterprise ecosystem solutions like SIEM, SOAR, and others.

Make Insights that are both clear and actionable

Increase the effectiveness of security personnel by including expert analysis and suggestions in incident summaries.

Patch database servers on a regular basis: Patches must be current. Effective databases patch management is a critical security technique since criminals are constantly looking for new database security gaps, and new malicious software appear regularly.

10. Database Backup and Restore Strategy

The SQL Server backup and restore component is a critical safeguard for safeguarding critical data in SQL Server databases. To reduce the danger of catastrophic data loss, the databases should be backed up on

a regular basis to preserve changes to your data. But for this task weekly full backup is good enough. A well-thought-out backup and restore strategy can help safeguard databases from data loss due to several faults. To prepare to respond effectively to a disaster, evaluate your plan by restoring a set of backups and then recovering your database.

Database administrators must address disaster recovery scenarios. Testing SQL Server backup and restoration techniques at regular intervals is one approach to do this. This ensures that data is recovered in a timely manner. And seamless recovery entails speedy system recovery with little or no data loss. Of course, a database administrator's job also includes protecting data from many types of data failures.

Ways to backup and restore database

1. Automate SQL database backup using window task scheduler
2. Configure long-term backup retention plan for the database.
3. Customizing deferential and transaction log backup.
4. Automate SQL data backups using maintenance plan.
5. Restore an SQL database from windows SQL server to SQL server Linux instance

11. Data Science/Business Intelligence Techniques

Analytics is a type of business intelligence that entails analyzing data to derive useful insights and patterns. It is important to include business intelligence for data reporting or descriptive analysis and data science for data for predictive analytics. This is where R-Programming, Python, SAS comes in to effectively get analytic insight into data. Then business intelligence tools like Microsoft Power BI, Tableau.

Both sectors may "process information," and they commonly employ the assistance of technology experts to transform or transform content outcomes into usable insights or competitive information. Combining BI and Data Science provide dependable choice tools to business owners, managers, and sometimes even front-line workers who are specialists in respective areas of work and rely on data professionals to make data-driven decisions.

12. Data Privacy, Ethical and legal issues

The two fundamental concerns with database privacy are the database's actual security and the legal and ethical consequences as to what can be stored on it in the first place. An examination of the inherent ethical burden placed on databases cybersecurity professionals to safeguard a database system should also be made. Database infusion is one of the ways database documentations is leaked through SQL statements. It is important to check statements for syntax errors because users can insert suspicious declarations into nullified user input, which is why database users' access should be restricted to a specific area only accessible by the administrator.

The access and denial function comes into play to validate database users at every point in time. Another way to deal with issues on database is to create stored procedures such that any interaction with a database id prewritten allowing only enough only enough rights as are required for any given command.

Another area of database privacy is to determine which content is for public use. For this task information available on the young life site is just for public use, general information about childhood poverty, it does not in any form display child name and personal information. Database Privacy (ncsu.edu)

13. Conclusion

Based on the studies performed using data from the Young Lives Survey, I believe that this system would be beneficial to policymakers who would use the information gathered from this work to make educated decisions to help reduce child poverty in these nations. Moreover, the front-end developer should have adequate information extracted from the young live data set.

Very importantly is obvious childhood poverty being extremely high in India, Vietnam, Ethiopia, and Peru from my summarized report.

TASK 2

Abstract

The school questionnaire was decided to add to Young Lives in 2010 after the third round of the survey conducted to capture extensive data about children's educational practices and to improve our awareness of the relationship between student achievement and children's home orientations, gender, work, schools, teachers, and school and class cohort.

education performance, such as valuation analysis of school education and comparative study of school-systems equity issues (including gender) in relation to learning consequences and the evolvement of inequalities within education, by analyzing factors having to explain the development of cognitive and non-cognitive skills in school, which include significance evaluation of academic achievement and comparative analysis of school-systems equity issues (including gender) in related to learning outcomes and the transformation of inequality and injustice within education, by analyzing factors describing the development of cognitive and non-cognitive.

The study helps investigators to correlate observational studies on household and child variables from the survey conducted with data on the schools that the Young Lives children go to school and the children's accomplishments both inside and outside of school. It contains regulation data

on the relationship between early childhood development (and its factors) and child's school experiences, such as access, quality, but instead progression. Young Live's competitive advantage is a time-series comprising home, child, and school-level data.(Thang and Hang 2018)

1. Introduction

Vietnam has a long history of education and places a high importance on education. Since 1945, early education has also been a major focus in Vietnam. This custom depicts the traditional Vietnamese attachment between mother and kid. Which is to look after a newborn until he or she is old enough to attend elementary school.(Vu 2021) This Task emphasis the process of creation and development of early child education in Vietnam, before and during the French colonial period(from 1884 to 1945), early child education was not considered as a social task. So, there was no formal educational system or curriculum for pre-school children currently.

This study outlines the thought process, analysis, and decision-making that went into developing this frontend-based reporting system that would provide users with information about Vietnam's educational sector. Tables, views, and charts are displayed in this system to provide more useful data for analytical purposes.

2. Design Rationale

The rationale behind this task is to carefully bring out and emphasis early childhood education in Vietnam considering the life of a child growing up with more or less educational system and curriculum. For this task base on the large dataset to be imported, processed, and analyzed, some rationale was closely considered.

The table was carefully created with appropriate datatypes and constraint of foreign with data description which states if column should be Null or not Null and primary key to avoid wrong data from been inserted in the table.

Also, data Normalization was considered to ensure all created tables are relational with both with a parent table referencing other tables. Stored procedures were considered to check ingestion of data into database are safe for privacy purpose.

3. Design Considerations

Database Normalization

The Vietnam data was carefully organized by arranging the columns and table relations of a database to ensure dependencies are properly enforced by database integrity constraints such as constraints Null and not Null.

Essentially the normalization was carried out to reduce redundancy and avoid dependencies anomalies, this will also help to avoid slow loss of data integrity and ensures meaningful queries and analysis.

3.1 Constraints

Constraints are set of property which is assigned to column or columns to prevent inconsistency in the database. Some constraints like null, not null, primary key and foreign key were used in the task. Constraints are used to enhance the data integrity. This enforces the reliability and accuracy, consistence of the data stored in the database.

3.2 Data Validation

Data validation is used to verify the correctness and quality of data . It is done by incorporating various checks into a system or report to ensure that input and stored data are logically consistent. Various checks on this task were implemented to ensure data inserted into the table remains quality.

uniqueness check, some data, such as IDs are unique. These fields should certainly have unique entries in a database. A uniqueness check ensures that an item is not entered into a database more than once, this was used mostly as primary key and foreign key in tables.

A data type check was used when creating tables and verifies that the data entered is of the proper type. A field, for example, may only take numeric values. If this is the case, the system should reject any data that contains additional characters, such as letters or special symbols.

3.3 Transaction and Concurrency Control

Transaction and concurrency control in SQL database is used to maintain the consistency and liability of the database object, data administrator should consider concurrency control as part database design and implementation

3.4 Security

Database security entails allowing or disallowing user actions on the database and the objects within it. In addition to putting the data in schema's it is important to do the following to ensure that the database is secured:

- Restrict physical access to the physical location of the database
- Enforce tight security
- Login should be reduced to one person per group
- Granting least privilege access to users to avoid unwanted changes to the databases.
- Use of strong passwords, a mix of Capitals, lower and symbols

- Password hashes should be stored encrypted and salted
- An account lock should occur after three or four login attempts

3.5.Comments

Comment in SQL query is to describe the intent of the programmer on that statement. We have two types of comment which are multiline and single line comment. For the task single line comment is implemented.

4. T-SQL Statements

Tables

The Vietnam wave one and wave two data set was downloaded from the young live website as TAB delimiter and was converted and saved as csv delimiter in my laptop.

The full Vietnam wave 1 and wave two table data set was inserted into SQL server with the create and insert statement.

```
USE [adb.coursework]
GO
/****** creating database Object:  Table [dbo].[vietnamwave_1] with create and insert statement   Sc
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[vietnamwave_1](
    [UNIQUEID] [nvarchar](50) NOT NULL,
    [SCHOOLID] [int] NULL,
    [CLASSID] [int] NULL,
    [STUDENTID] [int] NULL,
    [YLCHILDID] [nvarchar](50) NULL,
    [PROVINCE] [float] NULL,
    [DISTRICTCODE] [float] NULL,
    [LOCALITY] [float] NULL,
    [GENDER] [float] NULL,
    [AGE] [int] NULL,
    [ETHNICITY] [float] NULL,
    [ABSENT_DAYS] [float] NULL,
    [MOM_READ] [float] NULL,
    [MOM_EDUC] [float] NULL,
    [DAD_READ] [float] NULL,
    [DAD_EDUC] [float] NULL,
    [STDYLCHD] [float] NULL,
    [STDCMPLT] [float] NULL,
    [STDDINT] [nvarchar](50) NULL,
    [STDLIV] [float] NULL,
```

```
USE [adb_coursework]
GO
/****** create database Object:  Table [dbo].[vietnamwave_2] with create and insert statement
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[vietnamwave_2](
    [UNIQUEID] [varchar](50) NULL,
    [SCHOOLID] [varchar](50) NULL,
    [CLASSID] [varchar](50) NULL,
    [STUDENTID] [varchar](50) NULL,
    [STNTCMP] [varchar](50) NULL,
    [STDINT] [varchar](50) NULL,
    [STMTHWRK] [varchar](50) NULL,
    [STMWRKCH] [varchar](50) NULL,
    [STMWRKCM] [varchar](50) NULL,
    [STETHWRK] [varchar](50) NULL,
    [STEWRKCH] [varchar](50) NULL,
    [STEWRKCM] [varchar](50) NULL,
    [STTMMWRK] [varchar](50) NULL,
    [STTMEWRK] [varchar](50) NULL,
    [STCMPHME] [varchar](50) NULL,
    [STCMPSCH] [varchar](50) NULL,
    [STCMPOTH] [varchar](50) NULL,
    [STLTESCH] [varchar](50) NULL,
    [STMSSDAY] [varchar](50) NULL,
    [STMSSCLS] [varchar](50) NULL,
```

The two tables created are:

1. Vietnamwave_1
2. Vietnamwave_2

```
select * from dbo.vietnamwave_1
```

100 %

Results | Messages

	UNIQUEID	SCHOOLID	CLASSID	STUDENTID	YLCHILDID	PROVINCE	DISTRICTCODE	LOCALITY	GENDER	AGE	ETHNICITY	ABSENT_DAYS	MOM_READ	MOM_EDUC	DAD_READ	DAD_EI
4	VN1203104	1203	1	4	NULL	5	1	1	2	15	1	0	1	2	1	1
5	VN1203105	1203	1	5	NULL	5	1	1	1	16	1	0	1	4	1	2
6	VN1203106	1203	1	6	NULL	5	1	1	2	15	1	0	1	1	1	1
7	VN1203107	1203	1	7	NULL	5	1	1	2	15	1	0	1	6	1	6
8	VN1203108	1203	1	8	NULL	5	1	1	2	15	1	0	1	2	1	2
9	VN1203109	1203	1	9	NULL	5	1	1	2	16	1	0	1	1	1	6
10	VN1203110	1203	1	10	NULL	5	1	1	2	16	1	0	1	6	1	6
11	VN1203111	1203	1	11	NULL	5	1	1	2	15	1	0	1	6	1	6
12	VN1203112	1203	1	12	NULL	5	1	1	1	15	1	0	1	6	1	6
13	VN1203113	1203	1	13	NULL	5	1	1	2	16	1	0	1	2	1	1
14	VN1203114	1203	1	14	NULL	5	1	1	2	15	1	0	0	0	1	2
15	VN1203115	1203	1	15	NULL	5	1	1	2	15	1	0	1	1	1	1
16	VN1203116	1203	1	16	NULL	5	1	1	1	15	1	0	1	2	0	0
17	VN1203117	1203	1	17	NULL	5	1	1	1	15	1	0	1	3	1	2
18	VN1203118	1203	1	18	NULL	5	1	1	1	16	1	0	1	6	1	6

```
select * from dbo.vietnamwave_2
```

100 %

Results | Messages

	UNIQUEID	SCHOOLID	CLASSID	STUDENTID	STNTCMP	STDINT	STMTHWRK	STMWRKCH	STMWRKCM	STETHWRK	STEWRKCH	STEWRKCM	STTMMWRK	STTMEWRK	STCMPHME
1	VN1203101	1203	1	1	88	10/3/2017	2	2	2	0	0	0	2	2	2
2	VN1203102	1203	1	2	88	10/3/2017	3	3	3	2	3	3	2	2	3
3	VN1203103	1203	1	3	88	10/3/2017	4	2	2	2	2	2	2	2	2
4	VN1203104	1203	1	4	88	10/3/2017	3	3	3	1	3	3	2	2	4
5	VN1203105	1203	1	5	88	10/3/2017	4	2	2	2	2	2	2	2	3
6	VN1203106	1203	1	6	88	10/3/2017	2	2	2	1	2	2	2	2	1
7	VN1203107	1203	1	7	88	10/3/2017	2	2	1	1	2	2	1	1	2
8	VN1203108	1203	1	8	88	10/3/2017	3	2	2	2	2	3	2	2	3
9	VN1203109	1203	1	9	88	10/3/2017	2	2	2	2	2	2	2	1	1
10	VN1203110	1203	1	10	88	10/3/2017	4	3	3	3	2	2	2	3	3

Views

T-SQL STATEMENT FOR VIEWS

four views created, from these views summarized report with chart will be implemented.

1. vietnamwave_combined

2. maths_score

3. English_score

4. maths_rawscore

```
--- Creating view to combine vietnam wave 1 and vietnam wave 2 together and generate a report picking column
---  names from both wave to create a view and summarized report using join to know the education rate of both mother
--- and father, also the rate at which student use computer in school----

create view vietnamwave_combined
as(
select a.uniqueid ,
case a.gender
when 1 then 'male'
when 2 then 'female' end as 'gender',
case a.province
when 4 then 'Lao Cai'
when 5 then 'Phu Yen'
when 3 then ' Hung Yen' end as 'Province',
case a.MOM_EDUC
when 0 then 'Never been to school'
when 1 then ' Primary school (Grades 1-5)'
when 3 then ' Intermediate vocational training ' end as 'Mothers Education',
case a.DAD_EDUC
when 0 then 'Never been to school'
when 1 then ' Primary school (Grades 1-5)'
when 3 then ' Intermediate vocational training ' end as 'Fathers Education',
case b.STMTHWRK
when 0 then'Rarely/Never '
when 1 then ' Once a week '
when 2 then ' 2 times per week'
when 3 then '3 times per week'
```

```
when 4 then ' More than 3 times per week' end as 'frequency of maths homework',
case b.STCMPSCH
when 1 then ' Never or almost never'
when 2 then 'Once or twice a month'
when 3 then 'Once or twice a week '
when 4 then 'Every day or almost every day ' end as 'Use of computer for schoolwork at school'
from dbo.vietnamwave_1 a join dbo.vietnamwave_2 b
on a.UNIQUEID = b.UNIQUEID
where gender in (1,2))
```

```
--- Creating views to query and get chart from various views considering gender column,Ethnicity coulmn
---- filtering with gender to know the maths score of childern in this ethnic group---

create view math_score
as(
select case gender
when 1 then 'male'
when 2 then 'female' end as 'Gender',
case Ethnicity
when 3 then 'Cham-HRoi'
when 4 then ' Ede'
when 6 then ' Nung'end as 'Ethinicity',
ROUND(avg(math_rawscore),2) as 'Average maths score'
from dbo.vietnamwave_1
where gender in (1,2)
group by Ethnicity,gender)
```

```
--- Creating views to query and get chart from various views considering gender column,Ethnicity coulmn
---- filtering with gender to know the english score of childern in this ethnic group---

create view English_score
as(
select case gender
when 1 then 'male'
when 2 then 'female' end as 'Gender',
case Ethnicity
when 3 then 'Cham-HRoi'
when 4 then ' Ede'
when 6 then ' Nung'end as 'Ethinicity',
ROUND(avg(eng_rawscore),2) as 'Average english score'
from dbo.vietnamwave_1
where gender in (1,2)
group by Ethnicity,gender)
```

```
--- Creating views to query and get chart from various views considering gender column,province column,
---- Ethnicity column,filtering with gender to know the maths score of childern in this ethnic group---

create view maths_rawscore
as(
select case gender
when 1 then 'male'
when 2 then 'female' end as 'Gender',
case Ethnicity
when 3 then 'Cham-HRoi'
when 4 then ' Ede'
when 6 then ' Nung'end as 'Ethinicity',
case Province
when 1 then 'Ben Tre'
when 2 then ' Da Nang' end as 'Province',
ROUND(avg(math_rawscore),2) as 'Average maths score'
from dbo.vietnamwave_1
where gender in (1,2)
group by Ethnicity,gender,Province)
```

```
--- Creating views to query and get chart from various views considering
gender column, Ethnicity column
---- filtering with gender to know the English score of children in this
ethnic group---

create view English_score
as(
select case gender
when 1 then 'male'
when 2 then 'female' end as 'Gender',
case Ethnicity
when 3 then 'Cham-HRoi'
when 4 then ' Ede'
when 6 then ' Nung'end as 'Ethinicity',
ROUND(avg(eng_rawscore),2) as 'Average english score'
from dbo.vietnamwave_1
where gender in (1,2)
group by Ethnicity,gender)
```

Stored Procedures

A stored procedure is a prepared SQL code like a function which can be saved to be used repeatedly. It is just to call it when needed. For this task, some stored procedures were implemented as shown on the extra features section.

Triggers

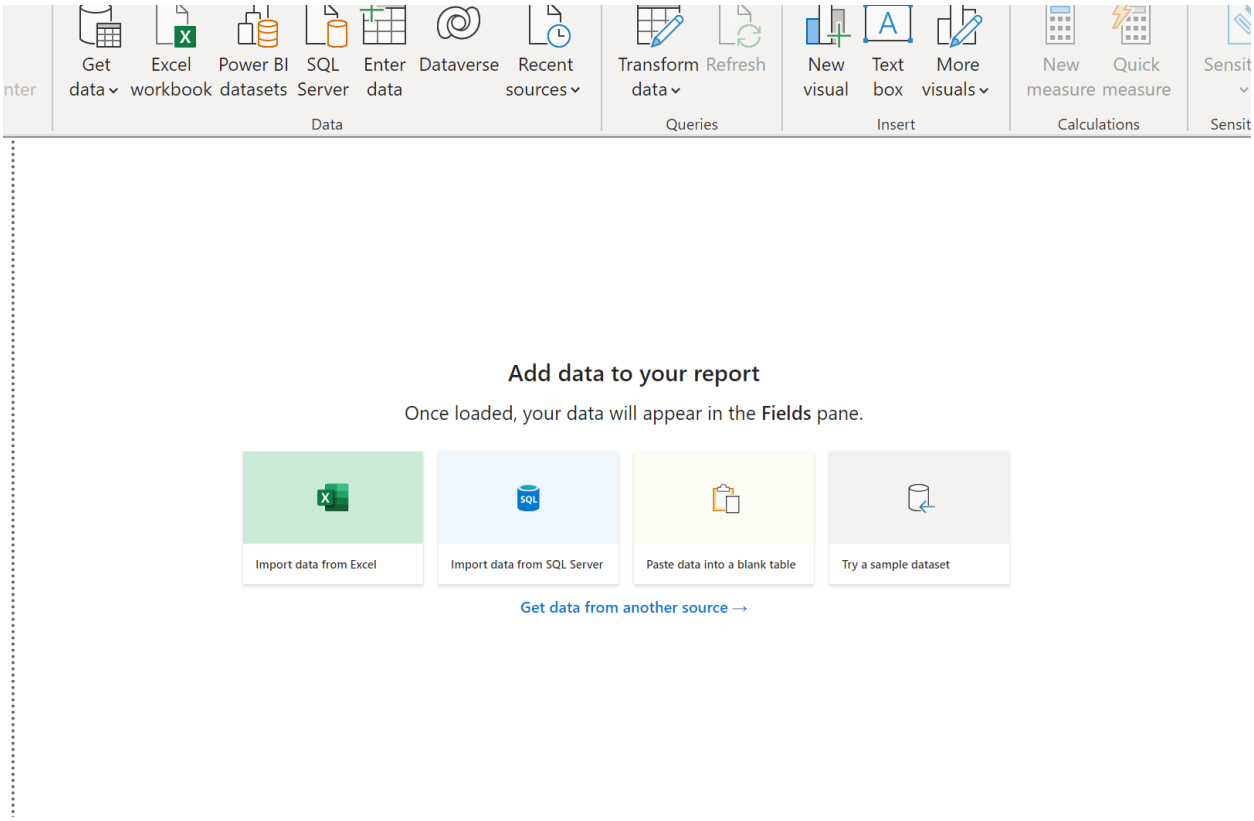
A database trigger is a stored program which will automatically trigger when events occur. It can be defined in table, schema, or view as the case. A trigger can occur in response to following

Data manipulation (DML) statement like delete, update, and insert, data definition(DDL) statements in alter, drop, and create.

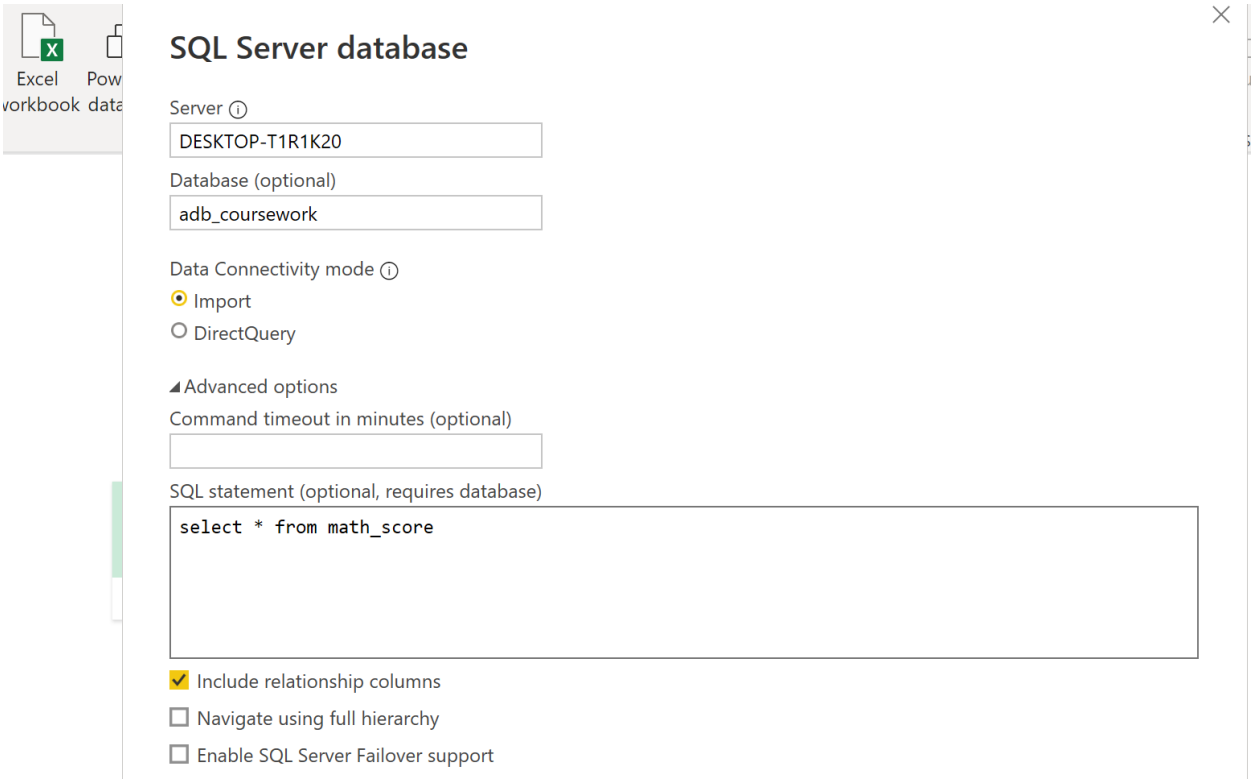
5. Report Design

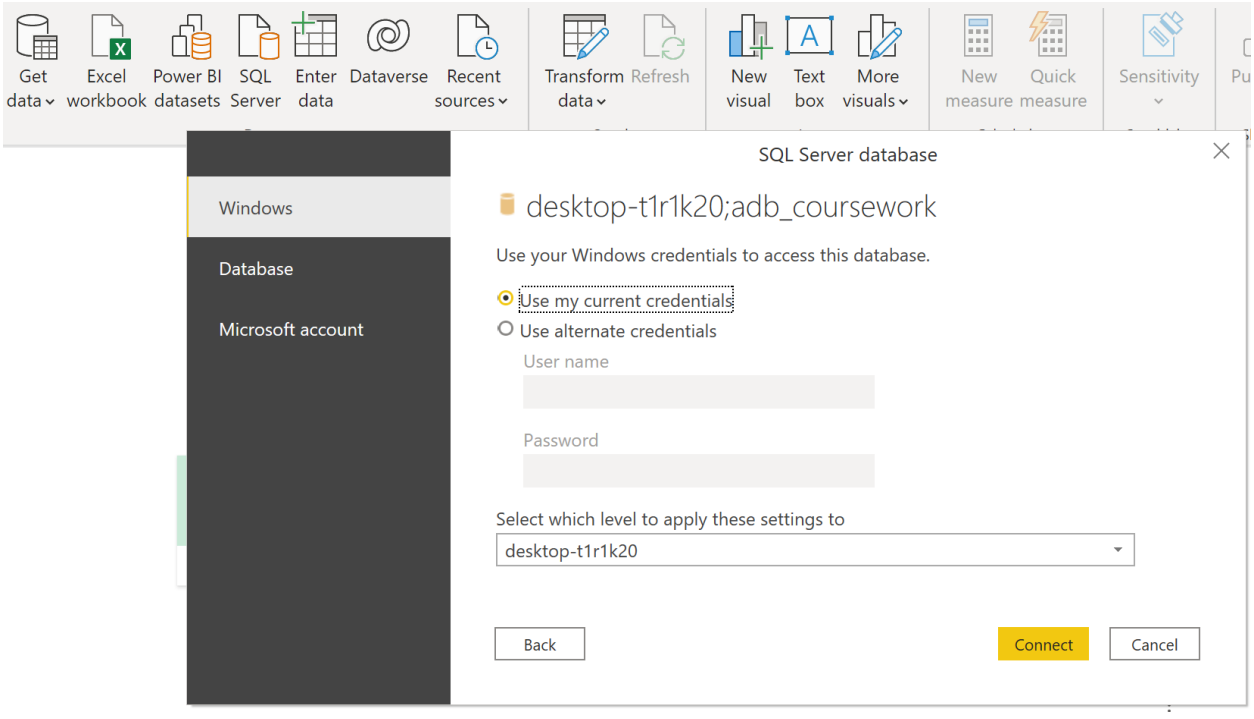
The report was design by connecting SQL server to Microsoft Power Bi with the views created. To be able to generate a report for visualization at the front-by-front end developers.

Design showing how to get data with Microsoft power BI

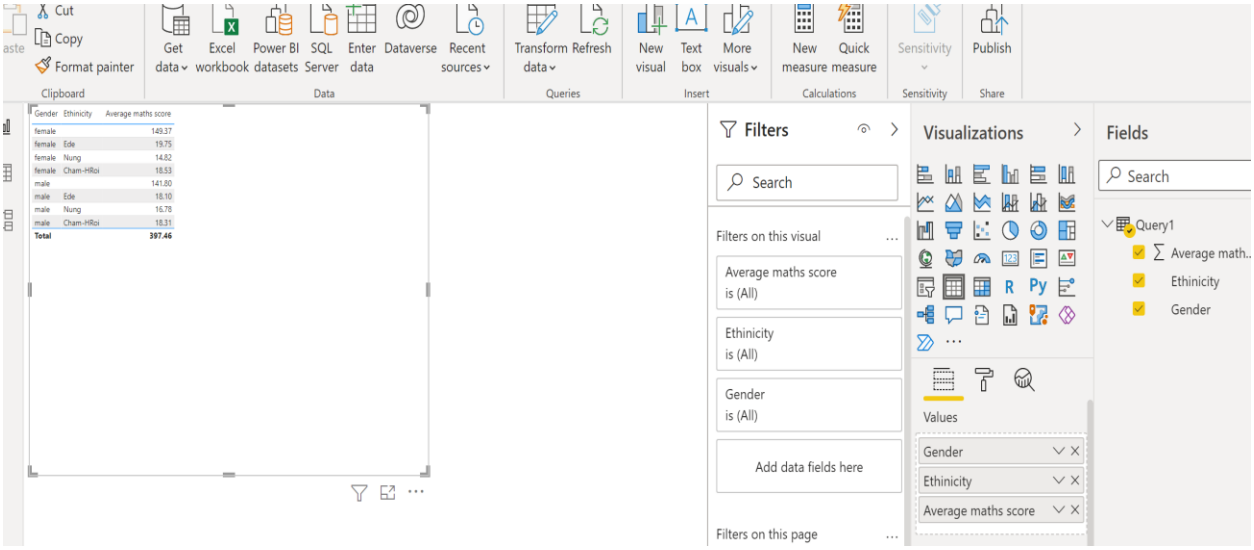


Selecting SQL server and database name to connect the views in SQL server to Microsoft power BI for the purpose of data reporting through visualization.


Excel workbook
Pow data
SQL Server database
Server
DESKTOP-T1R1K20
Database (optional)
adb_coursework
Data Connectivity mode
Import
DirectQuery
Advanced options
Command timeout in minutes (optional)
SQL statement (optional, requires database)
select * from math_score
Include relationship columns
Navigate using full hierarchy
Enable SQL Server Failover support


Get data
Excel workbook
Power BI datasets
SQL Server
Enter data
Dataverse
Recent sources
Transform data
Refresh
New visual
Text box
More visuals
New measure
Quick measure
Sensitivity
SQL Server database
Windows
Database
Microsoft account
desktop-t1r1k20;adb_coursework
Use your Windows credentials to access this database.
Use my current credentials
Use alternate credentials
User name
Password
Select which level to apply these settings to
desktop-t1r1k20
Back
Connect
Cancel

The view of the summarized report finally connected to SQL server successfully as shown below.



Finally, Visualization in clustered bar-chart and Pie-chart

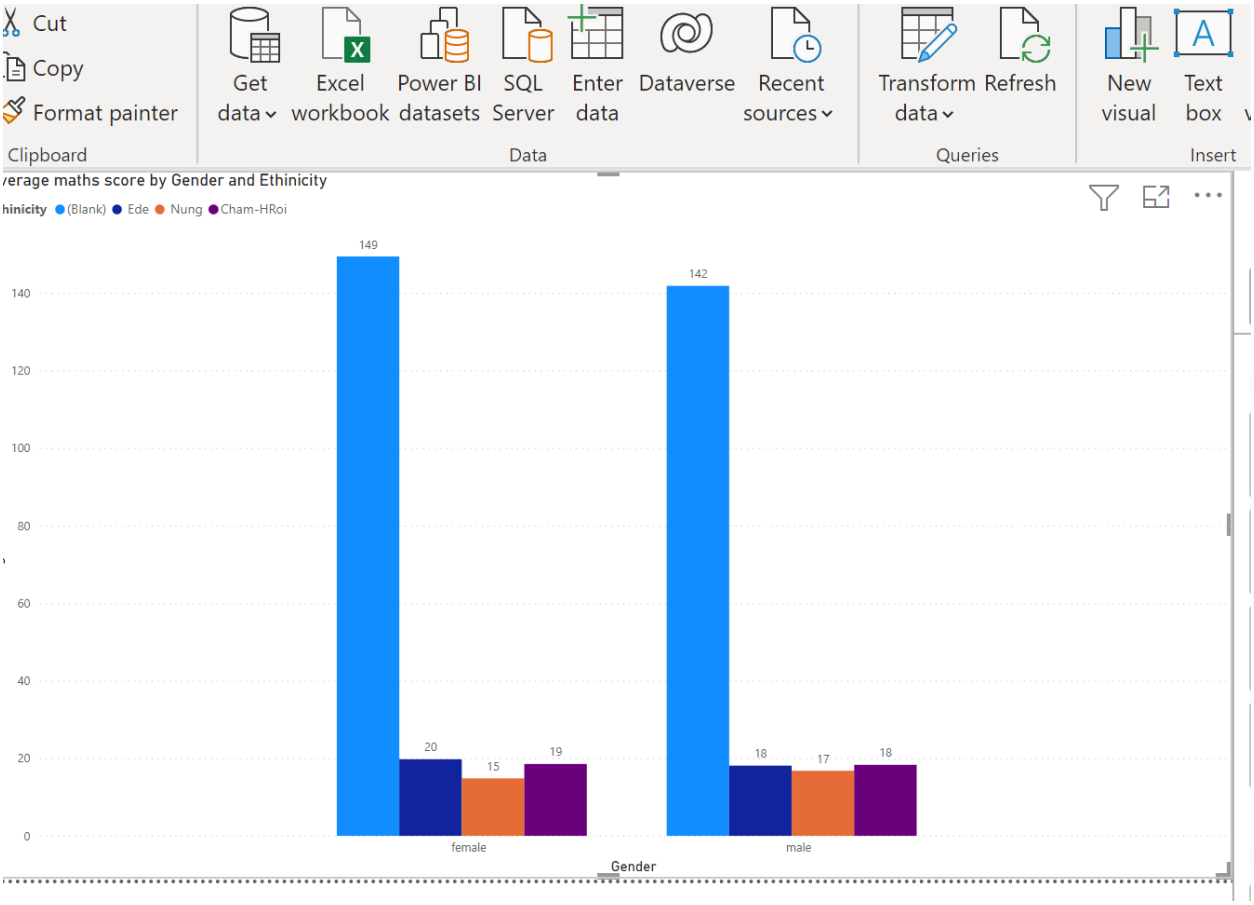


Fig 1.This view is for Average math score for selected ethnicity by gender in Vietnam, I could measure that the female and male in other ethnic of Vietnam score way higher than Cham-HRoi and Nung ethnic.

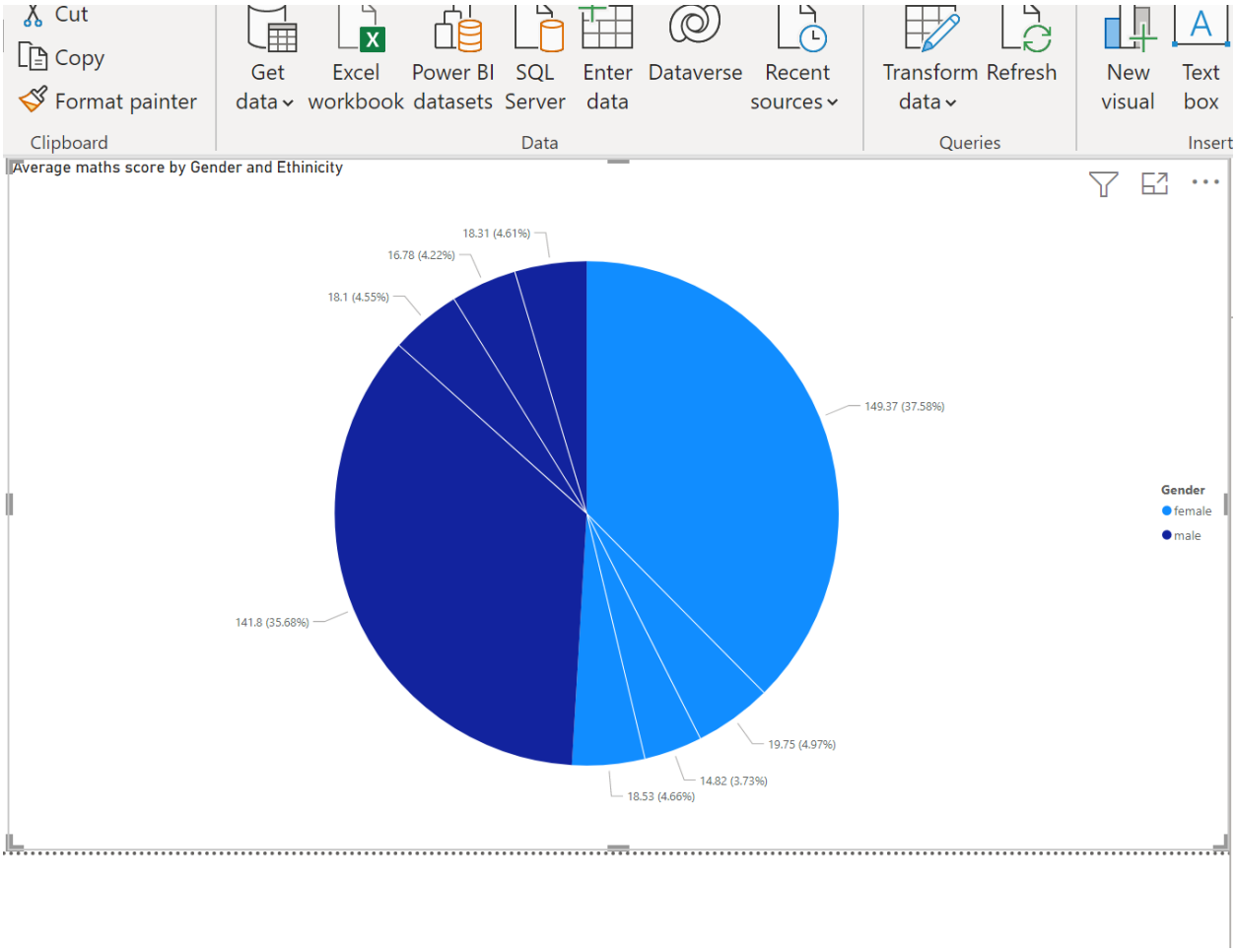
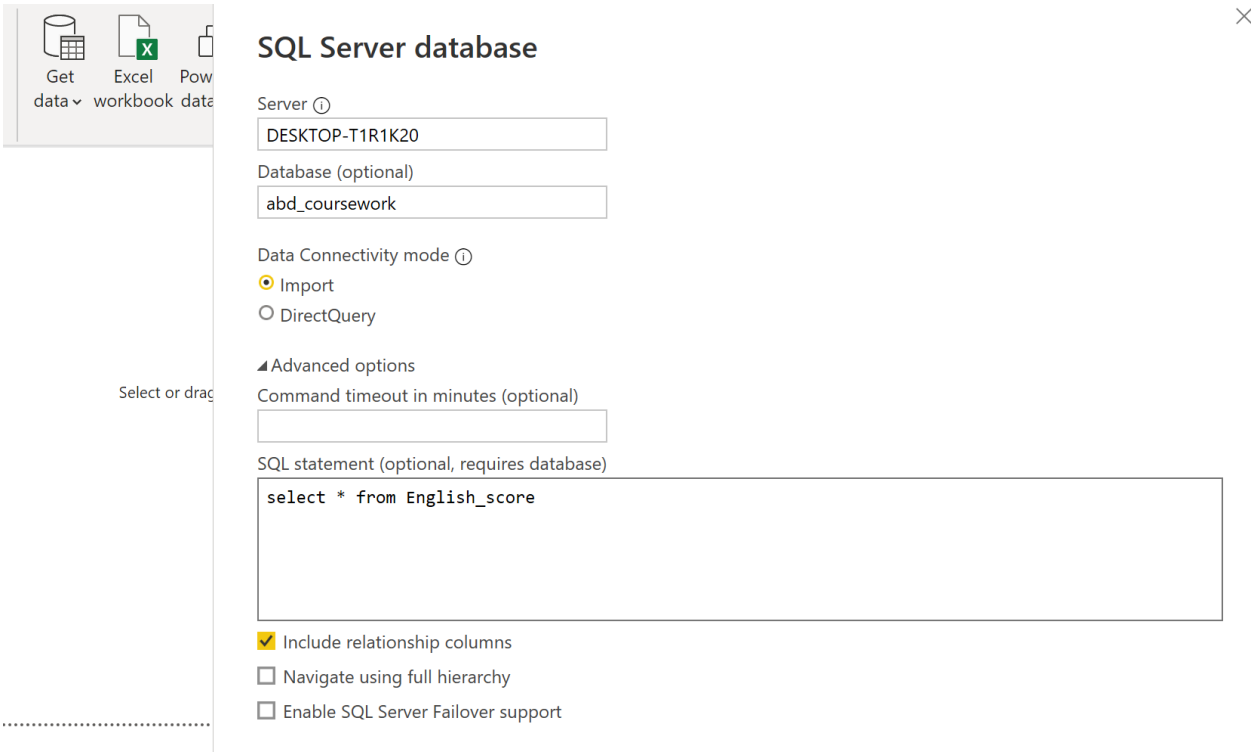
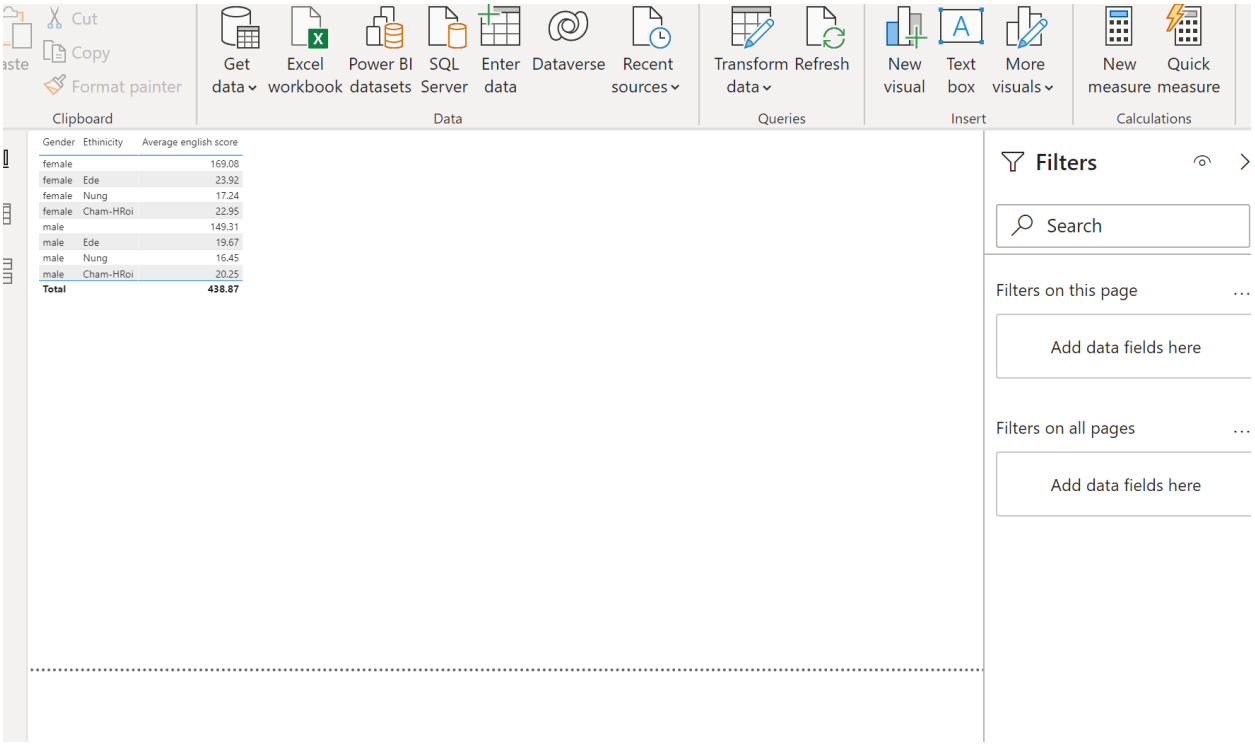


Fig 2 shows the pie chart of average math score by gender and ethnicity in Vietnam.

SUMMARIZED REPORT 2





Report table created by connecting power Bi to SQL server for report 2

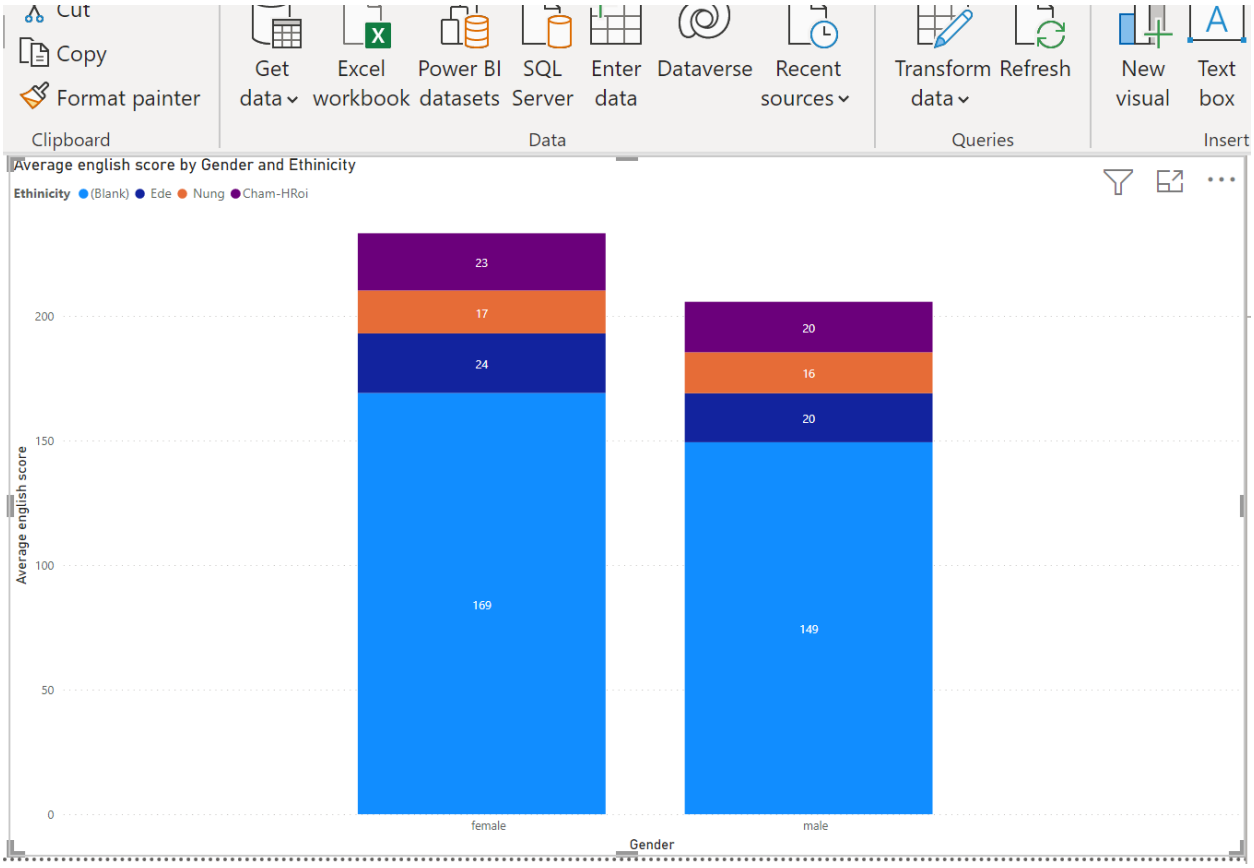


Fig 3.BAR CHART IMPLEMENTATION

This report is like the previous chart the other ethnicity has more English score this around compared to the other two ethnic in both male and female, which means child education in that ethnic has good foundation from the onset.

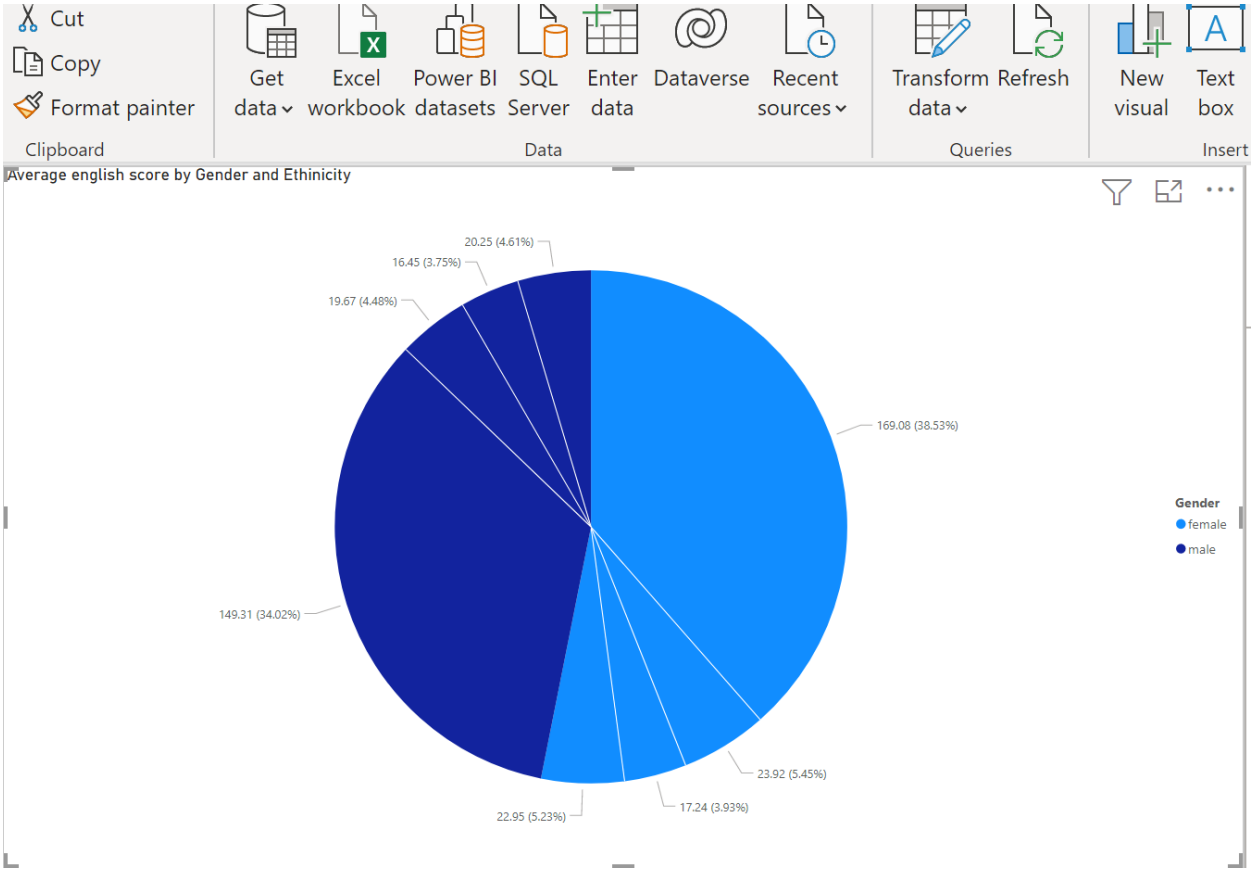
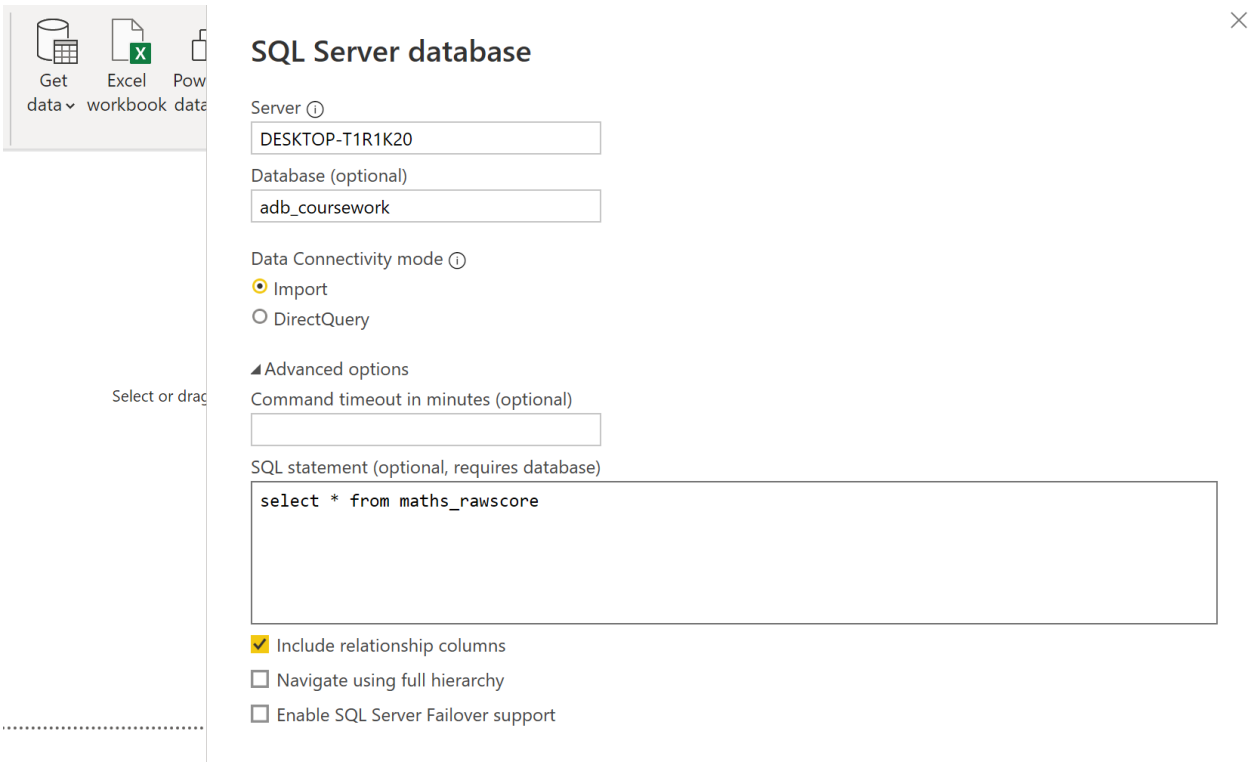
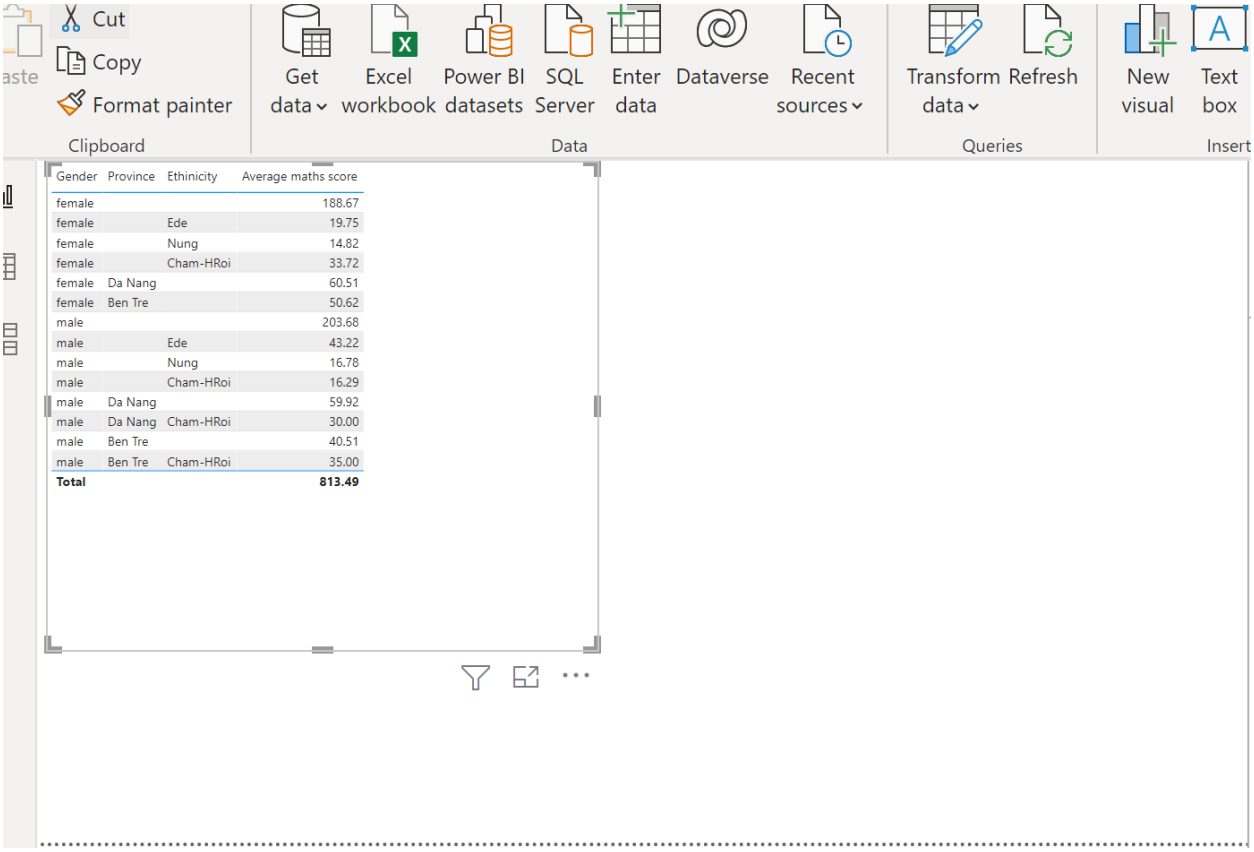


Fig 4.pie chart implementation of REPORT 2 displaying the English score by gender and ethnicity in Vietnam.

SUMMERIZED REPORT 3





Gender	Province	Ethinicity	Average maths score
female			188.67
female		Ede	19.75
female		Nung	14.82
female		Cham-HRoi	33.72
female	Da Nang		60.51
female	Ben Tre		50.62
male			203.68
male		Ede	43.22
male		Nung	16.78
male		Cham-HRoi	16.29
male	Da Nang		59.92
male	Da Nang	Cham-HRoi	30.00
male	Ben Tre		40.51
male	Ben Tre	Cham-HRoi	35.00
Total			**813.49**

REPORT CREATED FROM SQL SERVER of math rawscore

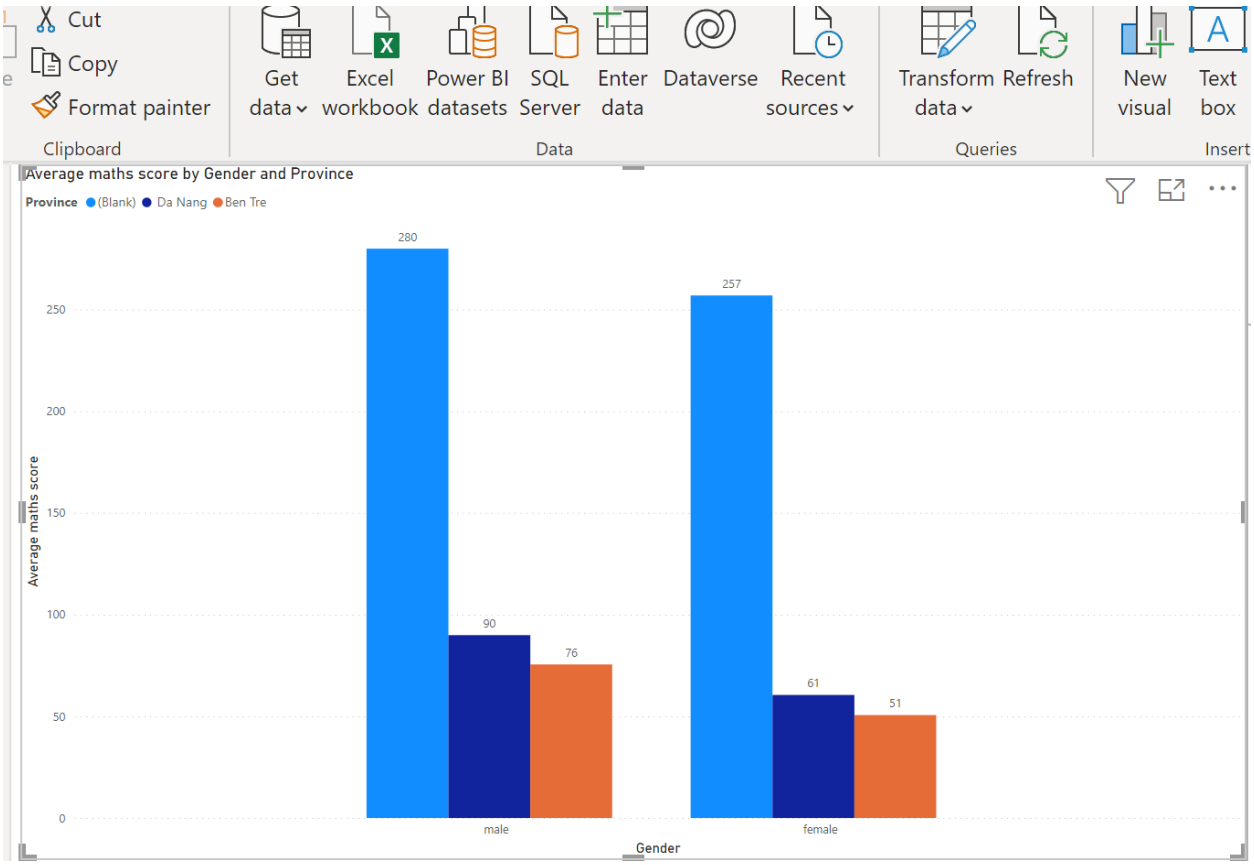


Fig 5. BAR-CHART IMPLEMENTATION

This summarized report shows the male and female in same province scored higher also in math score compared to other selected regions as show on the chart.

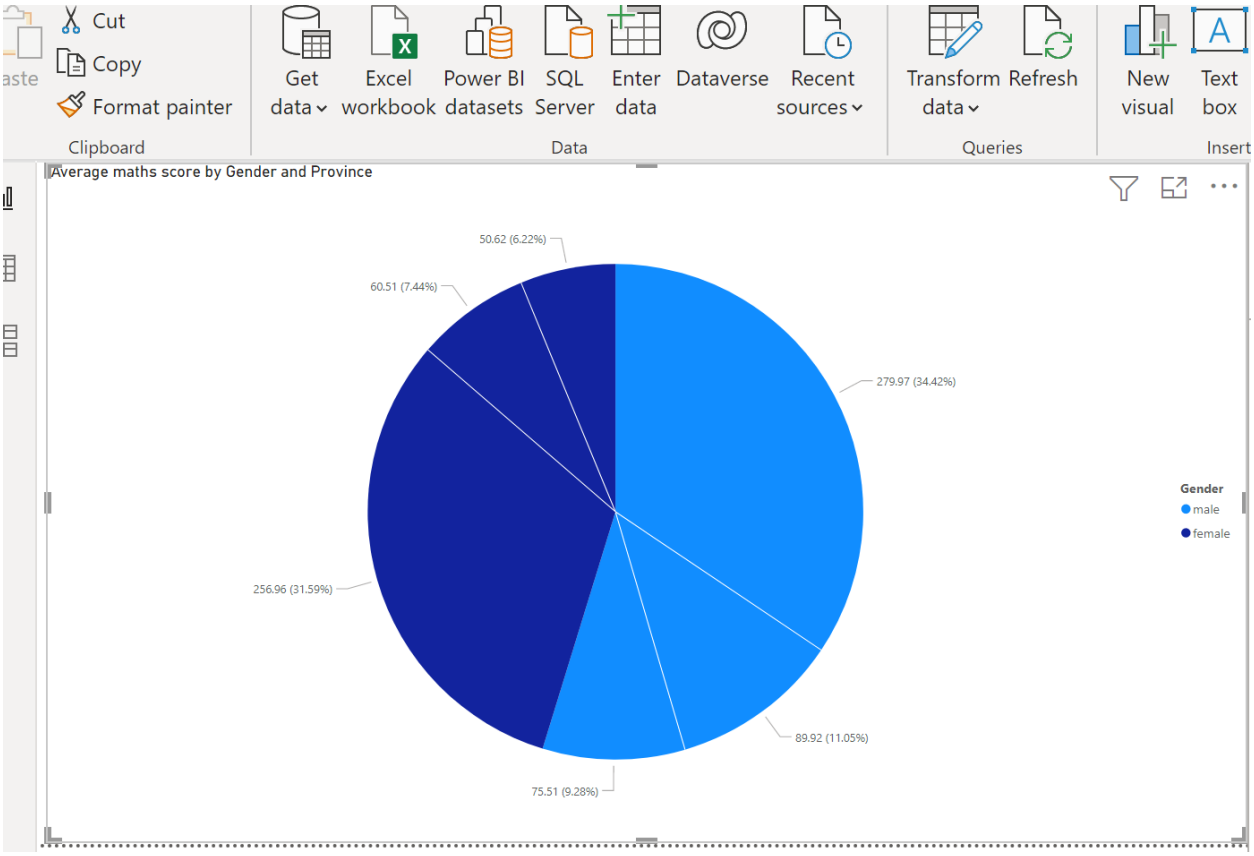
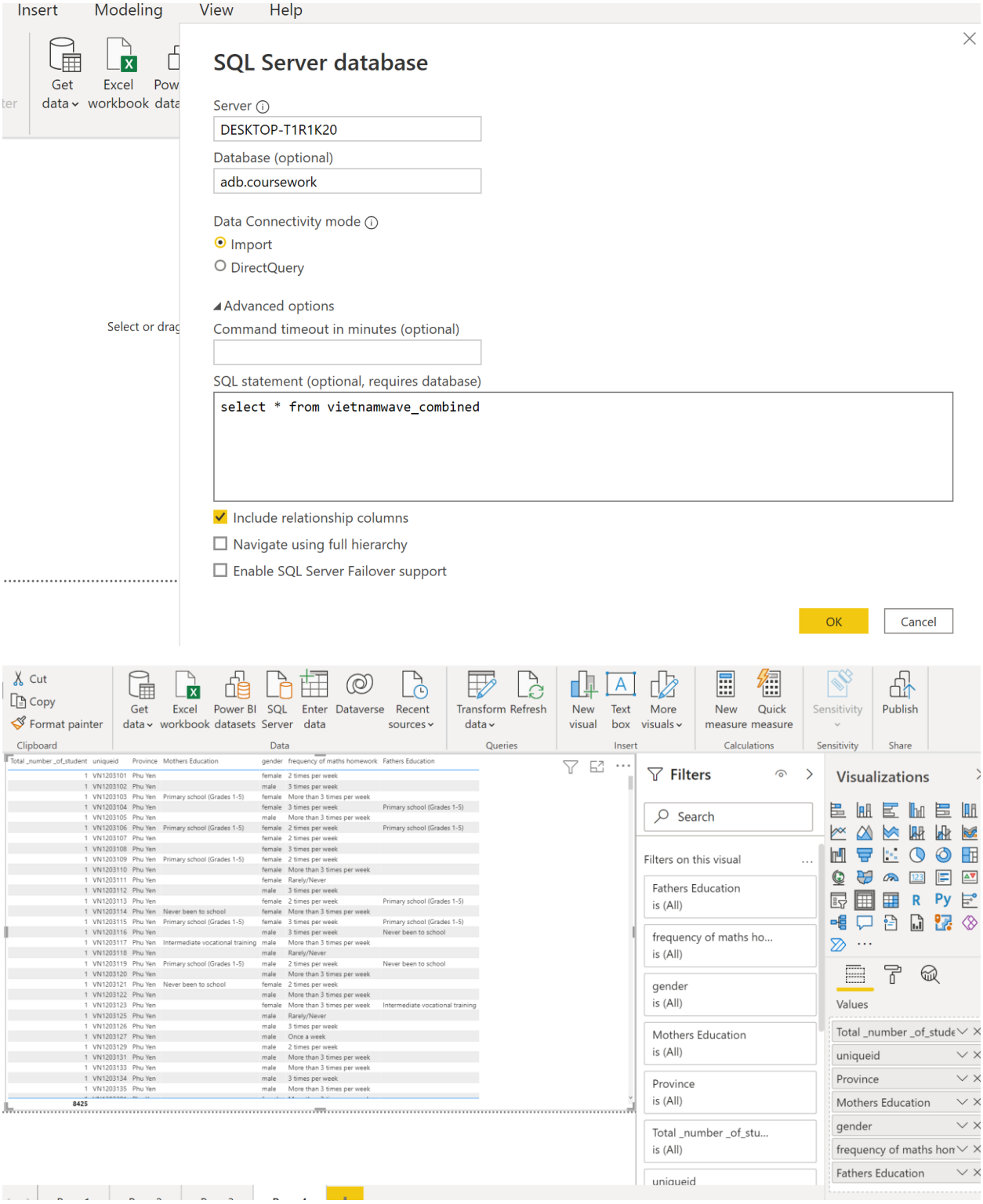


Fig 6. Pie chart of average math score in gender and ethnicity, it shows the percentage of student in other ethnicity to have higher score in English compared to Nung and Ede ethnic group.

SUMMERIZED REPORT 4

Connecting the SQL database to power bi to extract the mother and father education in Vietnam



Summarized report for mother and father education in Vietnam

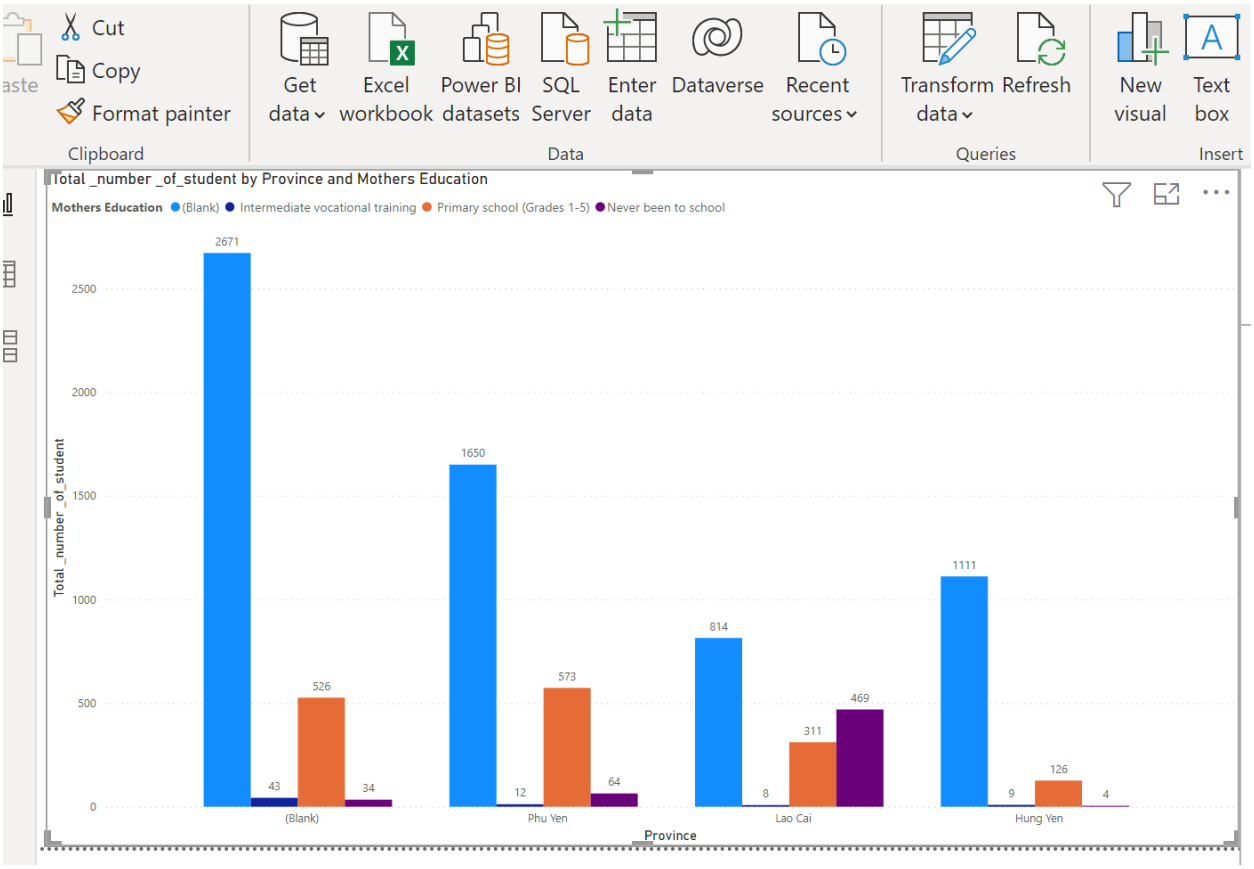
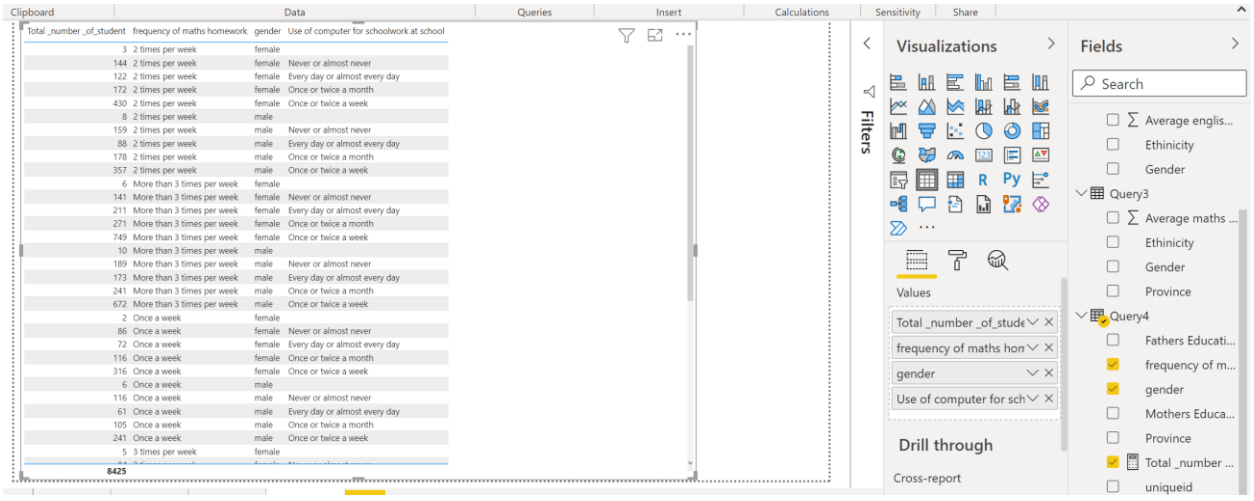


Fig 7. Bar chart showing mother education in various province indicating mother education is higher in other province compared to Lao Cai and Hung Yen, respectively.



Summarized report displaying how students use computer in school

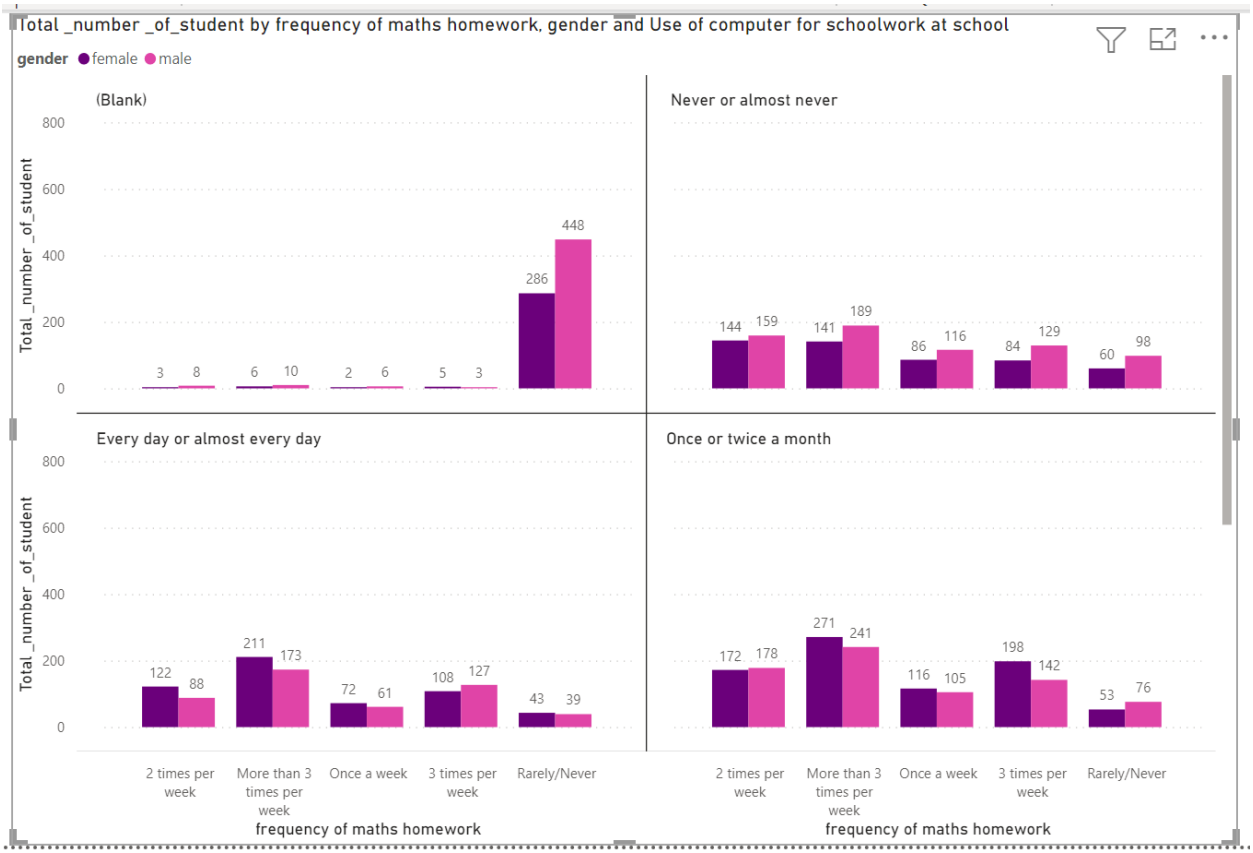


Fig 8. Bar charts indicate student rarely/never use computer in school is high.

6. Database Security

Data needs to be secured to prevent catastrophic breach, addressing the number of growing numbers of privacy mandate is difficult enough, keeping up with reporting can be another hardship, this process can be simplified with automation, analytics, and activity control.

Use of strong password both capital and small letter, reduce access to one or two persons. granting less privilege access to users, account lock was used on the data to lock unnecessary attempts.

7. Database Backup and Restore Strategy

Given how rarely the database is updated, a full database backup can be performed weekly and kept on separate hardware to ensure that if one of the datastores is damaged, the other can be used. The database backup hardware will be encrypted to prevent unauthorized access to the data. The database administrator oversees keeping the encrypted hardware secure and restoring the database when necessary. A stored procedure can also be used to monitor and automate backup and restore procedures.

8.Data Science/Business Intelligence Techniques

Briefly, Business Intelligence is a set of procedures that enable you to make the most of your data. It is made up of different analytics and reporting tools. Data mining, data analysis, data visualization, and key performance indicators are just a many of the things that come to mind. I will recommend the use of Comarch business intelligence, it has a lot of benefit from managing the rapid and increasing volume and velocity of data making analysis more efficient and dependable. Quick and accurate forecasting which leads to easier, and data driven decision making. Tools like R, power Bi and tableau are extremely useful for easy data driven decisions. For this task I personally used power bi for quick data reporting and visualizations.

9. Data Privacy, Ethical and legal issues

Privacy violations cause distrust and threaten dilution or loss of security; they are a display of disobedience to the law and a breach of ethical principles. Data privacy (also known as information privacy or data protection) refers to the legal right of a data subject to access, use, and gather data. Important in terms of contractual considerations of data privacy. Because data is a contract issue, it's important to think about what kind of personal data is being considered sensitive or not sensitive.(Lee, Zankl et al. 2016) for this task the information about childhood education in Vietnam do not contain any sensitive information about the child and parents.

10. Conclusion

The design and implementation of this task was conducted with T-SQL statement, meaningful comments when appropriate, screen shots of visualization from power BI, presentation layers also using power tools with full backup of the database

I was able to use my knowledge and expertise to design and implement a simple reporting system for a front-end developer. The client will be able to evaluate the child inequality situation in Vietnam and make necessary decision for polices that will address the childhood education system**.**

TASK 3

Abstract

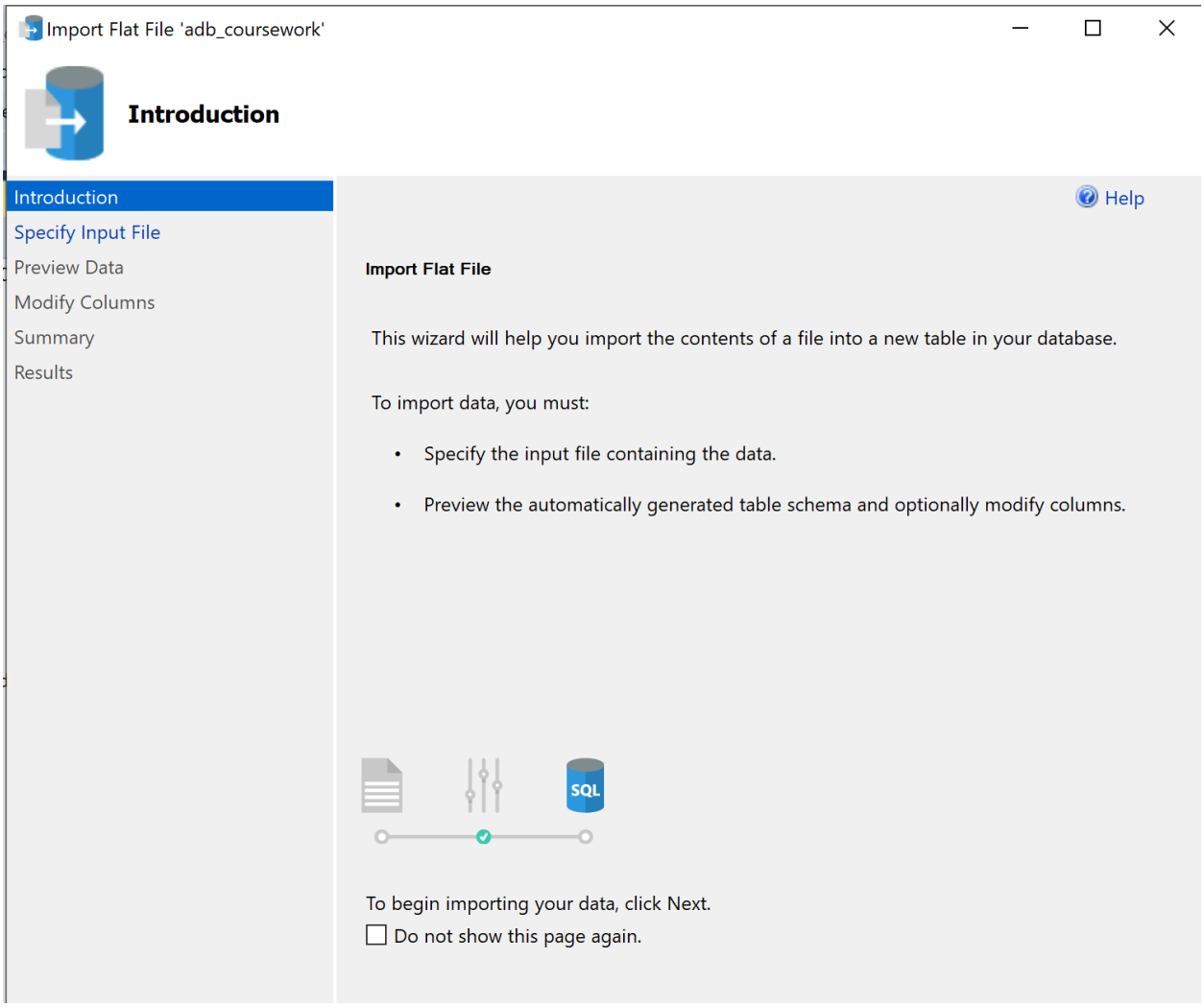
The is task is to provide a survey of the street level crime in Manchester using police statistics. The data published by the police are incident reported to them for that month. This map illustrates whether events are uniformly spread throughout the district or if hotspot exist in specific streets or areas.

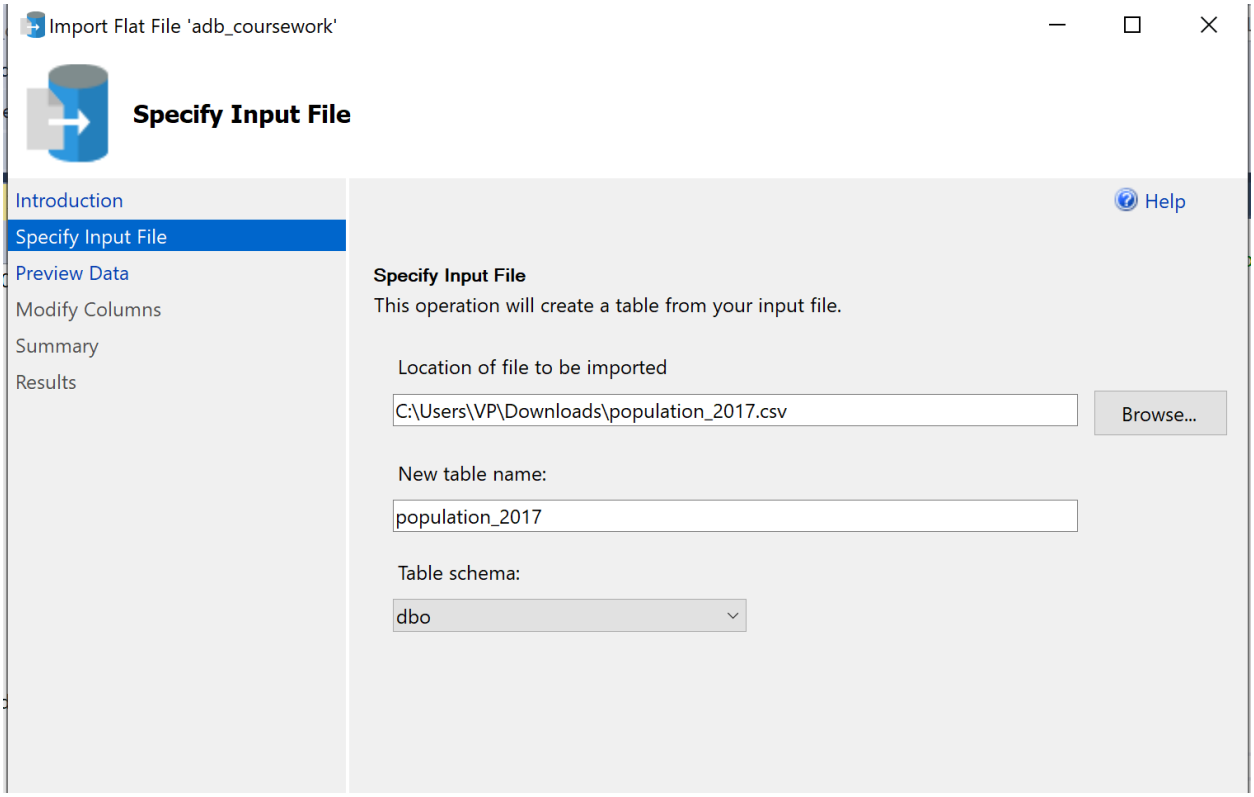
1. Introduction

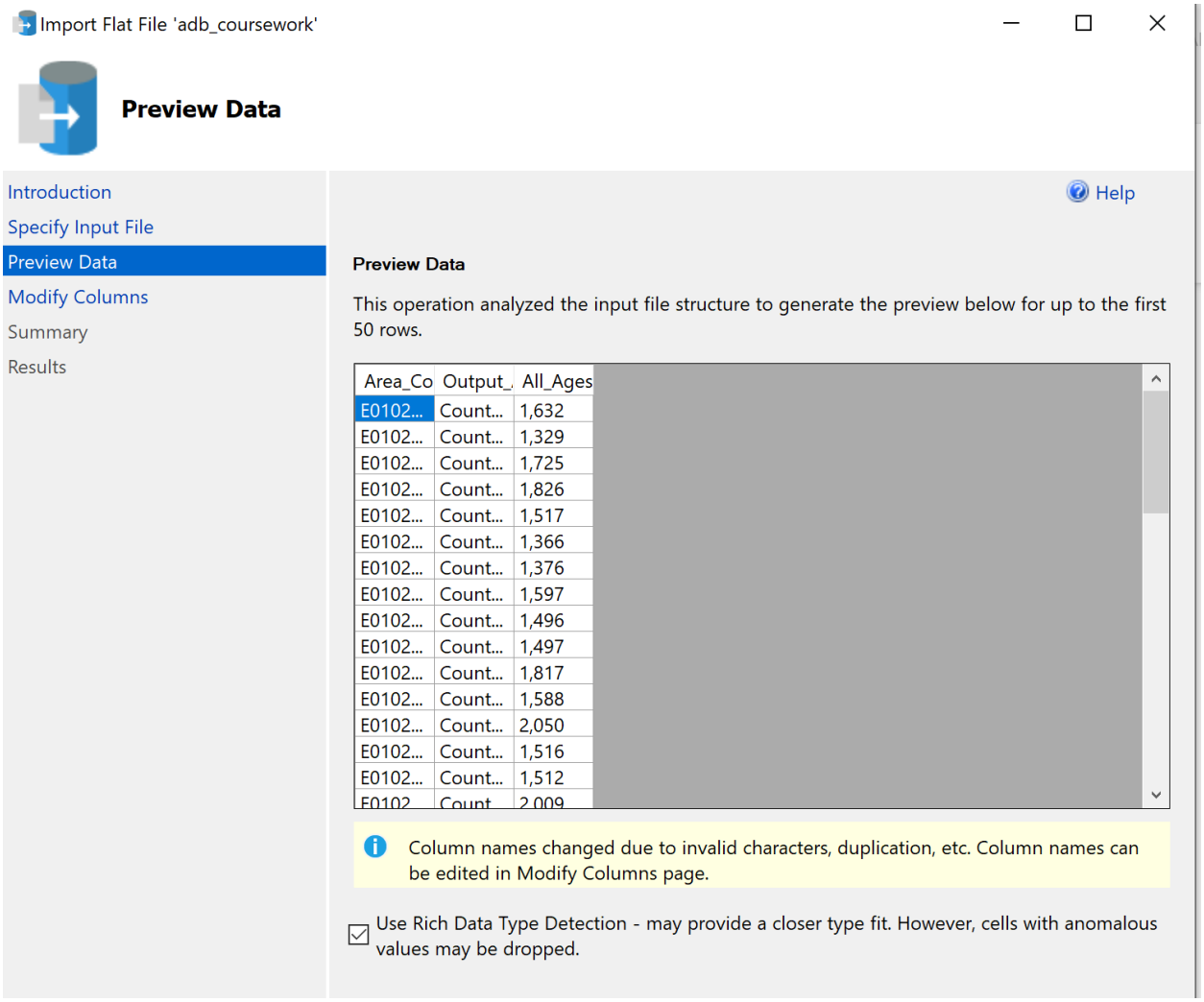
The introduction to this task involves a survey of crime types in some selected LSOAs with high anti-social behavior using SQL and Microsoft power BI to create a front-end design to able to evaluate situation leading to such crime and area at elevated risk.

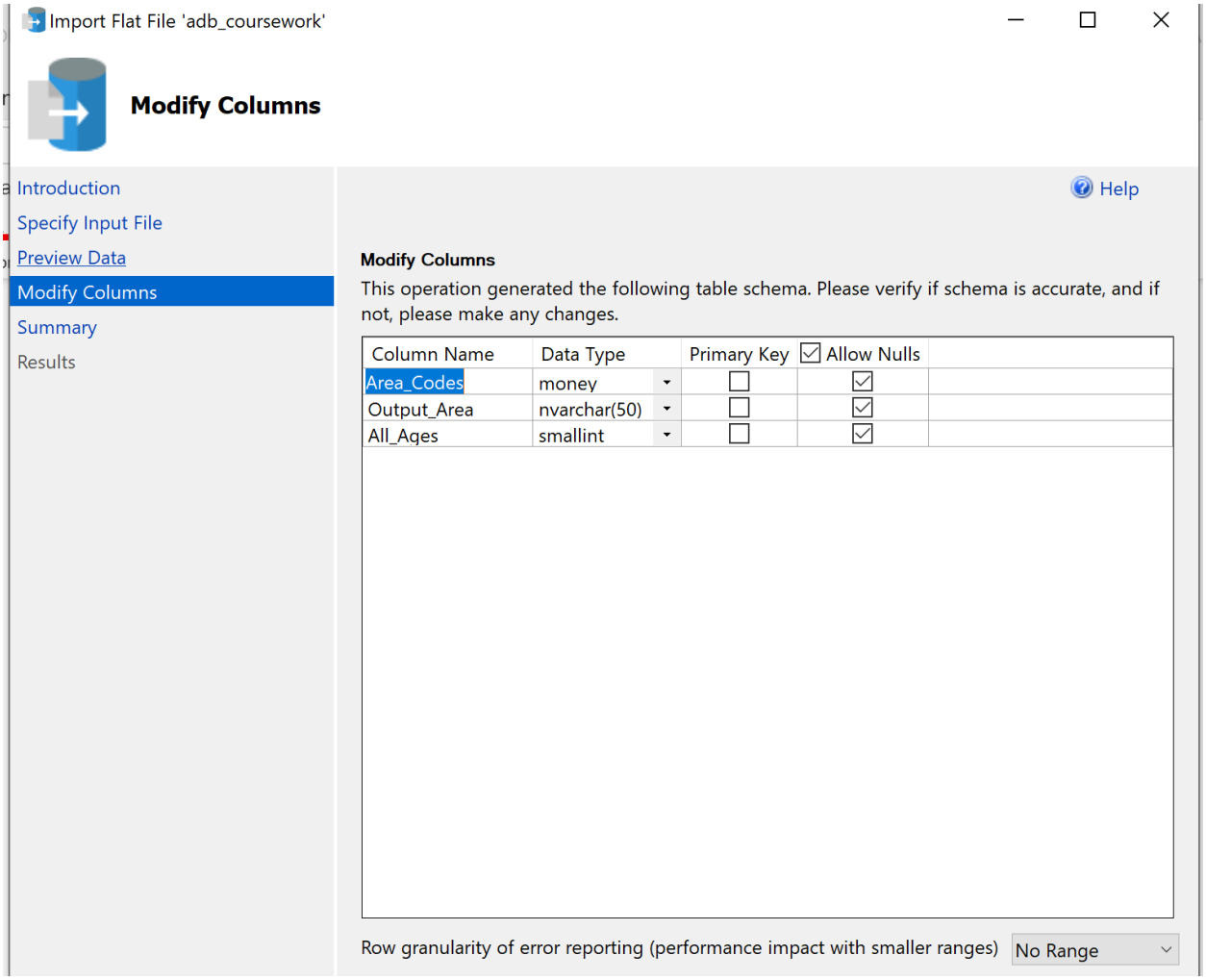
2. Importing tables

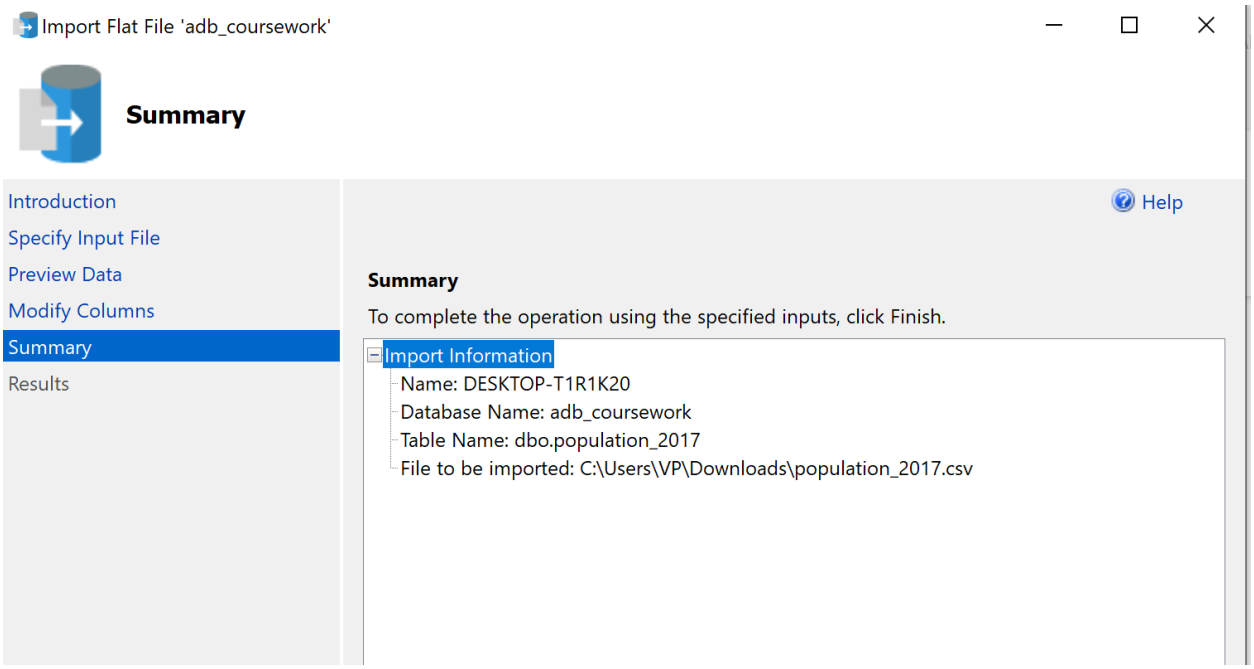
Inserting required Lower Layer Super Output Areas (LSOAs) data to a local SQL Server database from the SQL server using the import data wizard as follows. Firstly, connect my local system to SQL server through data base, task and click import

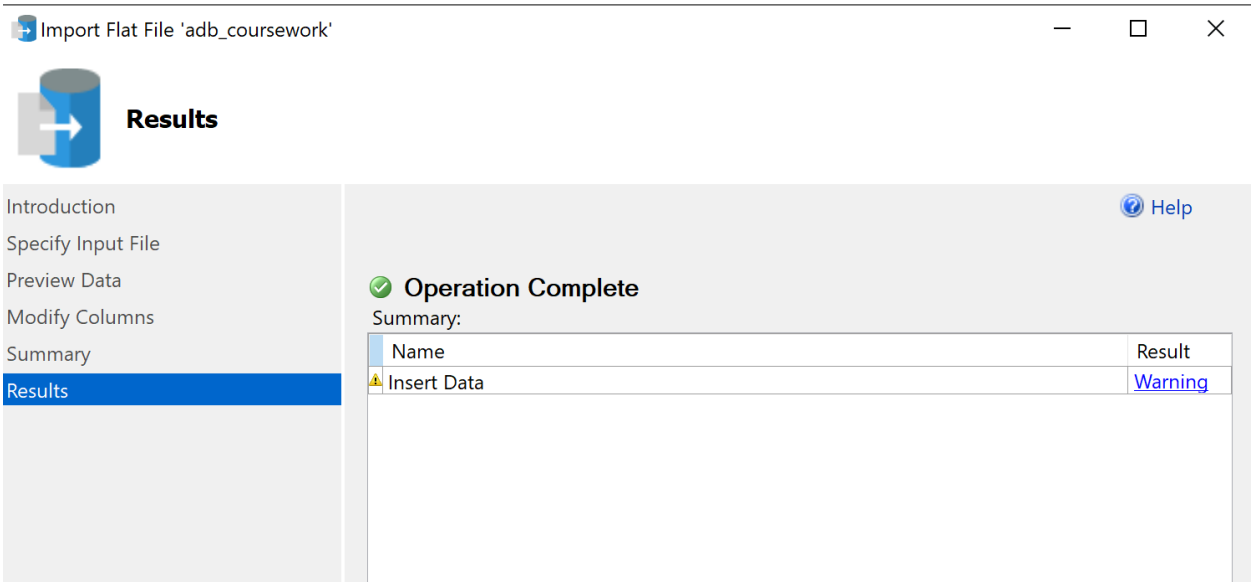

Import Flat File 'adb_coursework'
Introduction
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Import Flat File
This wizard will help you import the contents of a file into a new table in your database.
To import data, you must:
• Specify the input file containing the data.
• Preview the automatically generated table schema and optionally modify columns.
SQL
To begin importing your data, click Next.
Do not show this page again.


Import Flat File 'adb_coursework'
Specify Input File
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Specify Input File
This operation will create a table from your input file.
Location of file to be imported
C:\Users\VP\Downloads\population_2017.csv
Browse...
New table name:
population_2017
Table schema:
dbo

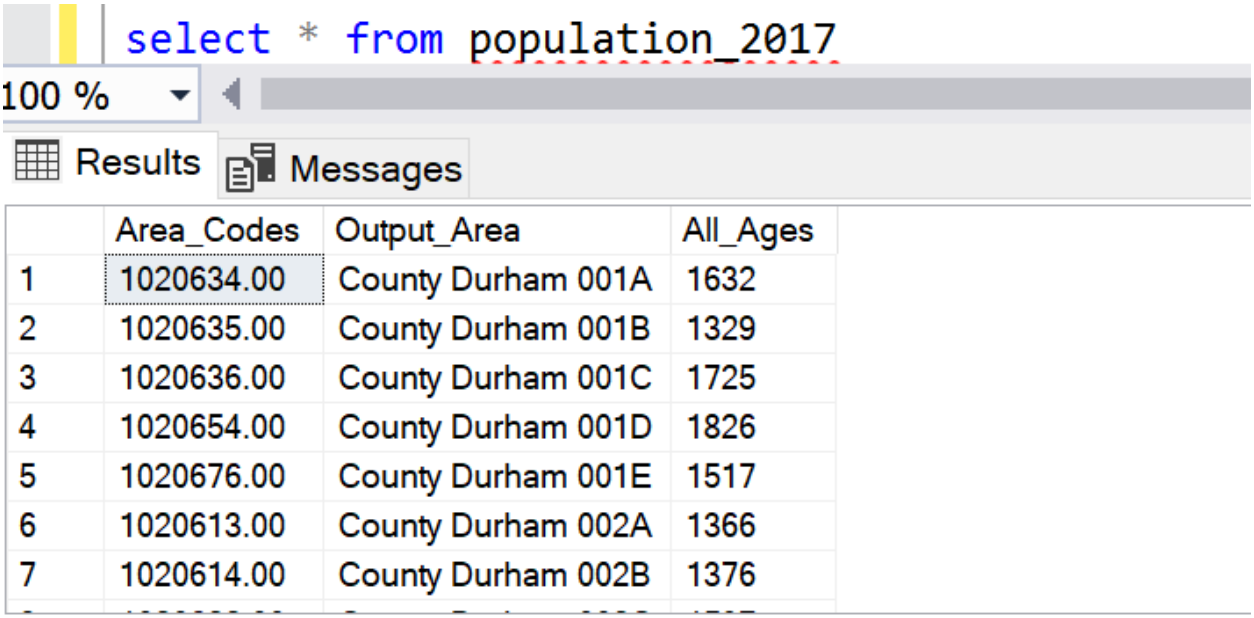

Import Flat File 'adb_coursework'
Preview Data
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Preview Data
This operation analyzed the input file structure to generate the preview below for up to the first 50 rows.
Area_Co Output_ All_Ages
E0102... Count... 1,632
E0102... Count... 1,329
E0102... Count... 1,725
E0102... Count... 1,826
E0102... Count... 1,517
E0102... Count... 1,366
E0102... Count... 1,376
E0102... Count... 1,597
E0102... Count... 1,496
E0102... Count... 1,497
E0102... Count... 1,817
E0102... Count... 1,588
E0102... Count... 2,050
E0102... Count... 1,516
E0102... Count... 1,512
E0102 Count 2,009
Column names changed due to invalid characters, duplication, etc. Column names can be edited in Modify Columns page.
Use Rich Data Type Detection - may provide a closer type fit. However, cells with anomalous values may be dropped.


Import Flat File 'adb_coursework'
Modify Columns
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Modify Columns
This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.
Column Name
Data Type
Primary Key
Allow Nulls
Area_Codes
money
Output_Area
nvarchar(50)
All_Ages
smallint
Row granularity of error reporting (performance impact with smaller ranges)
No Range


Import Flat File 'adb_coursework'
Summary
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Summary
To complete the operation using the specified inputs, click Finish.
Import Information
Name: DESKTOP-T1R1K20
Database Name: adb_coursework
Table Name: dbo.population_2017
File to be imported: C:\Users\VP\Downloads\population_2017.csv

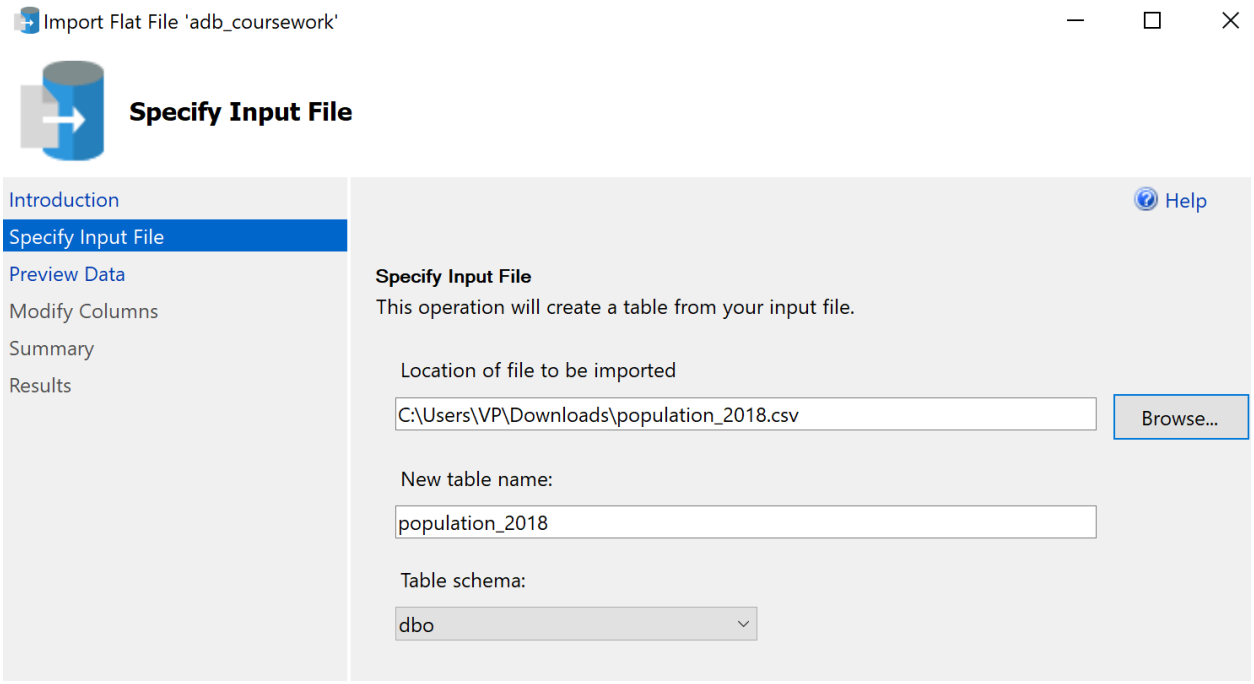


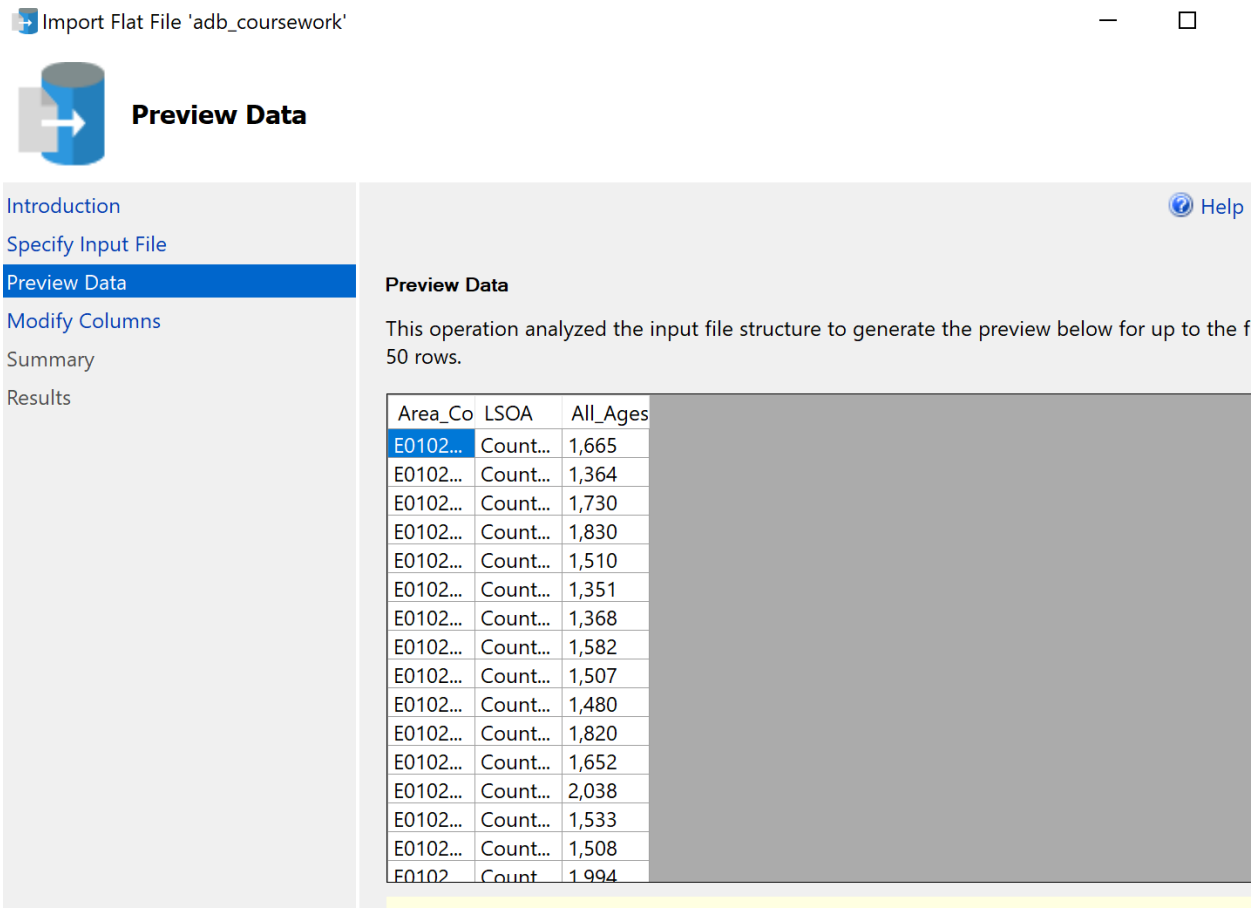
Below displays the select statement from SQL for the imported table population_2017

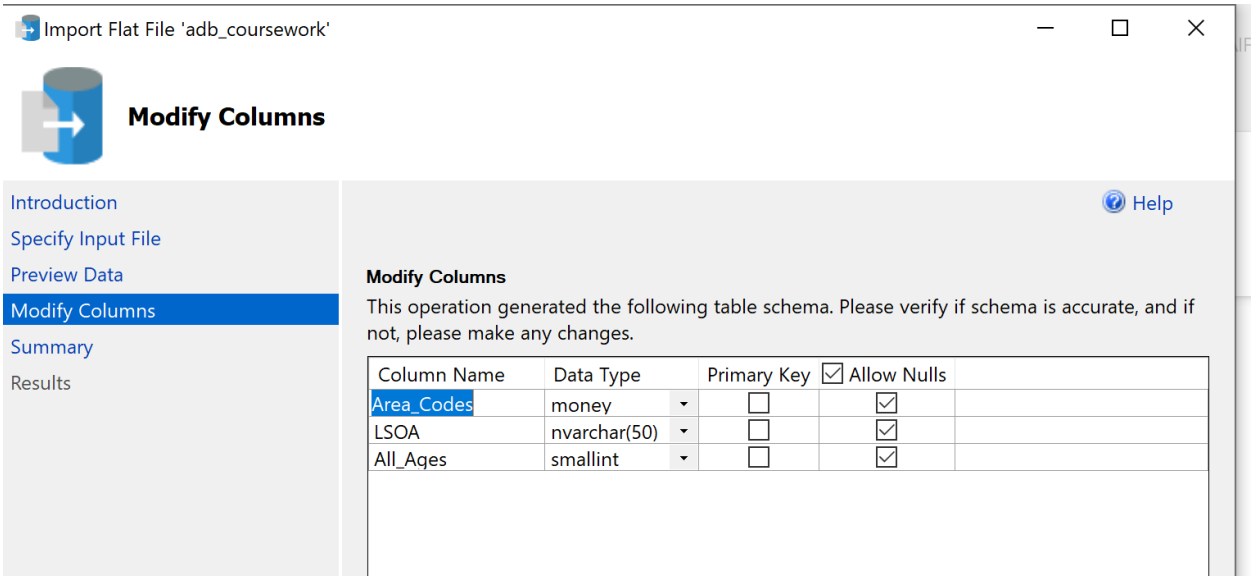


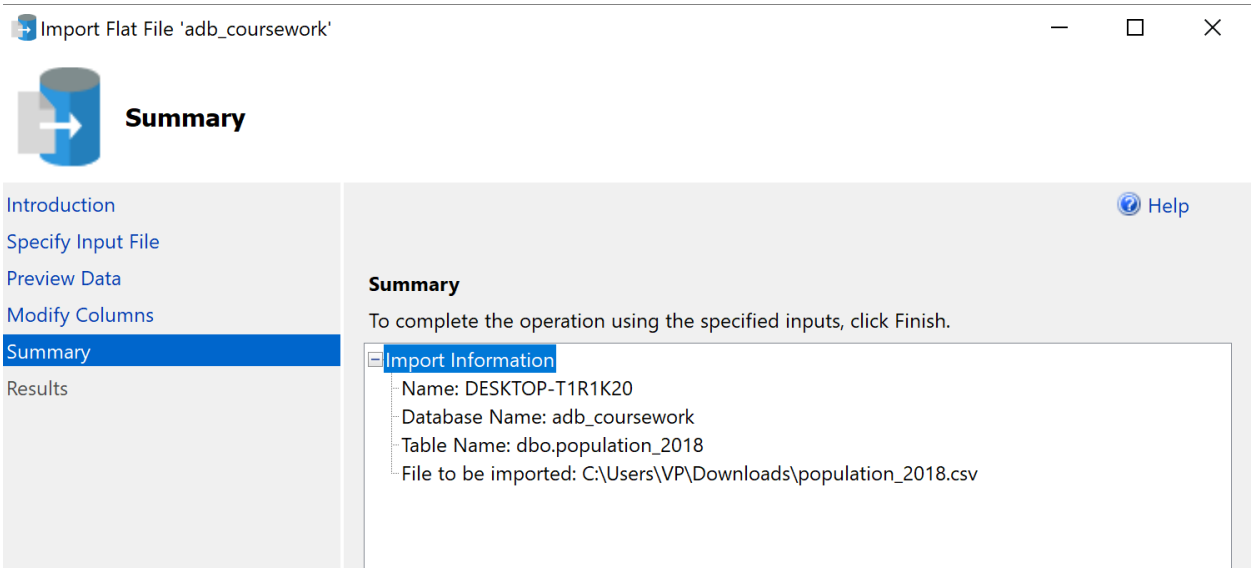
	Area_Codes	Output_Area	All_Ages
1	1020634.00	County Durham 001A	1632
2	1020635.00	County Durham 001B	1329
3	1020636.00	County Durham 001C	1725
4	1020654.00	County Durham 001D	1826
5	1020676.00	County Durham 001E	1517
6	1020613.00	County Durham 002A	1366
7	1020614.00	County Durham 002B	1376

Same import wizard procedure follows for importing population for 2018

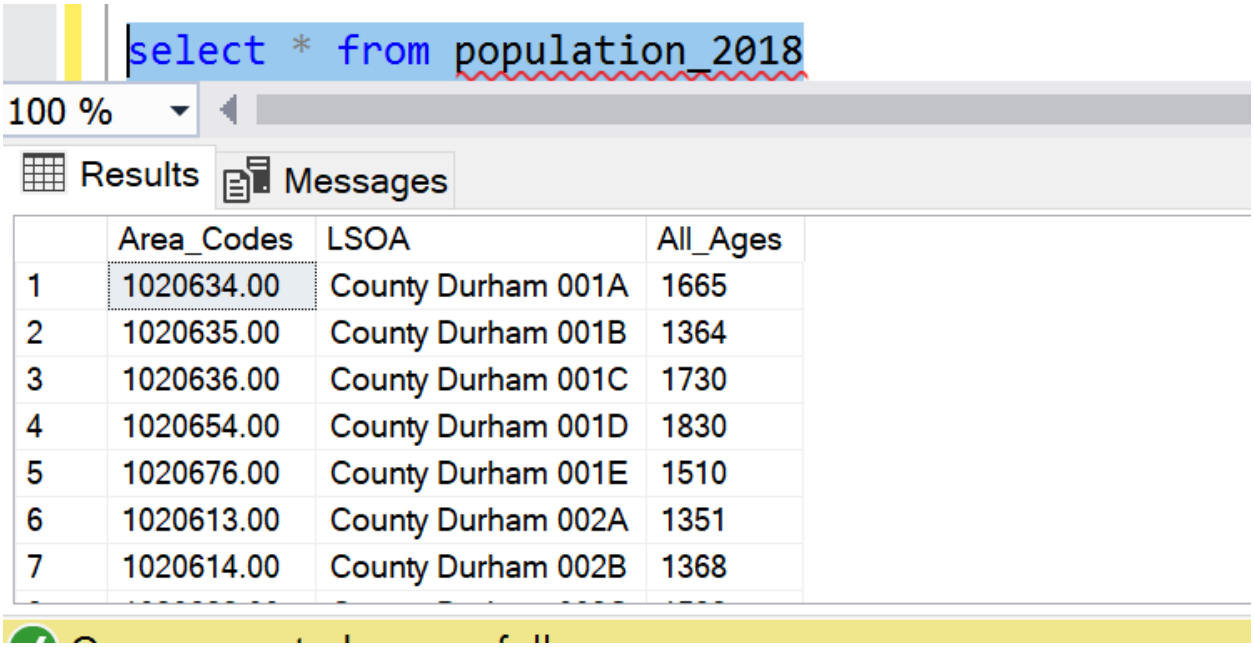

Import Flat File 'adb_coursework'
Specify Input File
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Specify Input File
This operation will create a table from your input file.
Location of file to be imported
C:\Users\VP\Downloads\population_2018.csv
Browse...
New table name:
population_2018
Table schema:
dbo


Import Flat File 'adb_coursework'
Preview Data
Introduction
Specify Input File
Preview Data
Modify Columns
Summary
Results
Help
Preview Data
This operation analyzed the input file structure to generate the preview below for up to the f
50 rows.
Area_Co LSOA All_Ages
E0102... Count... 1,665
E0102... Count... 1,364
E0102... Count... 1,730
E0102... Count... 1,830
E0102... Count... 1,510
E0102... Count... 1,351
E0102... Count... 1,368
E0102... Count... 1,582
E0102... Count... 1,507
E0102... Count... 1,480
E0102... Count... 1,820
E0102... Count... 1,652
E0102... Count... 2,038
E0102... Count... 1,533
E0102... Count... 1,508
E0102 Count 1,994





This is the select statements from SQL for the imported table population_2018



	Area_Codes	LSOA	All_Ages
1	1020634.00	County Durham 001A	1665
2	1020635.00	County Durham 001B	1364
3	1020636.00	County Durham 001C	1730
4	1020654.00	County Durham 001D	1830
5	1020676.00	County Durham 001E	1510
6	1020613.00	County Durham 002A	1351
7	1020614.00	County Durham 002B	1368

Insert required Crimes data from the Police data repository to a local SQL Server database

The data imported using the create and insert query because the data was too large for my SQL Server which I run out of memory.

```
USE [adb_coursework]
GO
/******creating Object:  Table [dbo].[crime012017]    S
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[crime012017](
    [Crime ID] [nvarchar](255) NULL,
    [Month] [nvarchar](255) NULL,
    [Reported by] [nvarchar](255) NULL,
    [Falls within] [nvarchar](255) NULL,
    [Longitude] [float] NULL,
    [Latitude] [float] NULL,
    [Location] [nvarchar](255) NULL,
    [LSOA code] [nvarchar](255) NULL,
    [LSOA name] [nvarchar](255) NULL,
    [Crime type] [nvarchar](255) NULL,
    [Last outcome category] [nvarchar](255) NULL,
    [Context] [nvarchar](255) NULL
) ON [PRIMARY]
GO
```

```
/****** creating Object:  Table [dbo].[Crime012018]    S
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Crime012018](
    [Crime ID] [nvarchar](255) NULL,
    [Month] [nvarchar](255) NULL,
    [Reported by] [nvarchar](255) NULL,
    [Falls within] [nvarchar](255) NULL,
    [Longitude] [float] NULL,
    [Latitude] [float] NULL,
    [Location] [nvarchar](255) NULL,
    [LSOA code] [nvarchar](255) NULL,
    [LSOA name] [nvarchar](255) NULL,
    [Crime type] [nvarchar](255) NULL,
    [Last outcome category] [nvarchar](255) NULL,
    [Context] [nvarchar](255) NULL
) ON [PRIMARY]
GO
```

3. Design Rationale

The rationale behind this task is to create a simple reporting tool with improve performance and efficiency by the development of credible database system using SQL server and power BI tools. Also considered a database that is up to date secured with accurate information.

4. Design Considerations

Constraints

Constraints can be created in SQL with create table statement or alter table statement, either way works fine. But for the purpose of this task constraint was created with create table

statement most cases. They are used to specify rules for data in a table, limiting the type of information that goes into the column or entire table. This help to ensure the accuracy and reliability of data in the table, I used constraints when creating my tables. The constraints used are not Null, check, foreign keys, primary keys. Constraints can be column level or table level, but my work only applies to column level constraint.

Data Validation

When using SQL data validation is the aspect of the database that keeps data consistent and dependable even when many users have access to the database. The main key factor in data integrity that plays this role is constraints. For this task the data validation was done by using check constraint, insert statement. The purpose is to ensure the correct data format has been entered into the database.

Error Handling

Handling error in database base object must be implemented from the beginning of the task, constraints, like primary keys checks, unique, not null ,null is implemented when creating tables, views, stored procedures to eradicate errors in data entry. This to avoid transactions are not aborted query statement runs.

Security

Security in databases is considered as the most important of all the responsibilities of a database administrator.one of the way of conducting security in database is SQL server authentication and authorization. The ability to authenticate and limit users access to specific function on the database. User defined functionality is also a useful tool to secure database objects.

Comments

Comments in SQL are used to explain what you intend to carry out in the query page or to prevent execution of SQL statements. In this task I used single line and multiline comment to explain and prevent SQL from executing certain statements.

5. T-SQL Statements

- **Tables**

```sql
--- creating a table with the join clause statement combining manchester_crime_table to population_total table----
---reporting crime type in Traford area---

select  sum(a.[All_Ages]) as 'Total people',b.[Month],b.[LSOA name],b.[Crime type],
count(*) as 'number of committed crime'
from dbo.manchester_crime_table as b join dbo.population_total as a
on a.[Output_Area] = b.[LSOA name]
where b.[LSOA name] like 'Tra%'
group by b.[Month], b.[Crime type],b.[LSOA name]
order by b.[Month] ASC
```

100 %

Results Messages

	Total people	Month	LSOA name	Crime type	number of committed crime
1	9518	2017-01	Trafford 001E	Anti-social behaviour	4
2	3563	2017-01	Trafford 001B	Anti-social behaviour	2
3	3635	2017-01	Trafford 001C	Anti-social behaviour	2
4	9992	2017-01	Trafford 001A	Anti-social behaviour	8
5	10110	2017-01	Trafford 003A	Anti-social behaviour	6
6	8928	2017-01	Trafford 002C	Anti-social behaviour	6

```sql
---Creating a table to view and report all crime type committed in Bolton grouping by Month,Crime type,
---LSOA name also ordering by Month Ascending---

select  sum(a.[All_Ages]) as 'Total people',b.[Month],b.[LSOA name],b.[Crime type],
count(*) as 'number of committed crime'
from dbo.manchester_crime_table as b join dbo.population_total as a
on a.[Output_Area] = b.[LSOA name]
where b.[LSOA name] like 'Bol%'
group by b.[Month], b.[Crime type],b.[LSOA name]
order by b.[Month] ASC
```

100 %

Results Messages

	Total people	Month	LSOA name	Crime type	number of committed crime
1	3164	2017-01	Bolton 001A	Anti-social behaviour	2
2	5958	2017-01	Bolton 001C	Anti-social behaviour	4
3	12320	2017-01	Bolton 004A	Anti-social behaviour	8
4	14735	2017-01	Bolton 004B	Anti-social behaviour	10
5	9543	2017-01	Bolton 001E	Anti-social behaviour	6
6	2818	2017-01	Bolton 001D	Anti-social behaviour	2

```sql
--- creating a table to view and report Robbery crime in salford from population_total table
--- and manchester_crime_table with where clause and oder by clause,group by LSOA name and Month---

select  sum(a.[All_Ages]) as 'Total people',b.[Month],b.[LSOA name],b.[Crime type],
count(*) as 'number of committed crime'
from dbo.manchester_crime_table as b join dbo.population_total as a
on a.[Output_Area] = b.[LSOA name]
where b.[LSOA name] like 'sal%'
and b.[Crime type] = 'Robbery'
group by b.[Month], b.[Crime type],b.[LSOA name]
order by b.[Month] ASC
```

100 %

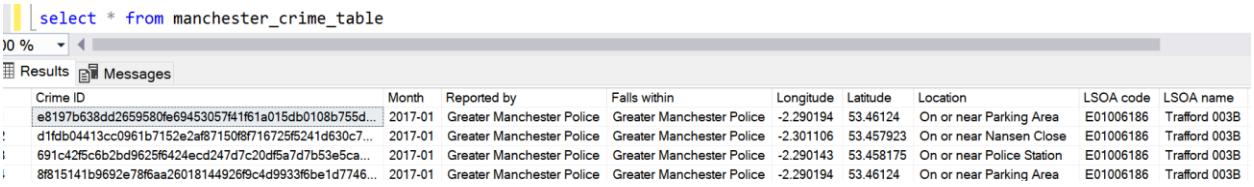
Results Messages

	Total people	Month	LSOA name	Crime type	number of committed crime
1	3138	2017-01	Salford 001A	Robbery	2
2	3848	2017-01	Salford 004A	Robbery	2
3	3506	2017-01	Salford 004B	Robbery	2
4	3240	2017-01	Salford 007F	Robbery	2
5	6362	2017-01	Salford 010A	Robbery	4
6	12357	2017-01	Salford 010D	Robbery	6

```
--- Creating a table called manchester_crime_table to union all the tables in
crime from 2017 January to 2018 December
--- using the union clause statement and select into statement---

select * into dbo.manchester_crime_table
from
dbo.Crime012017 union all select *from dbo.Crime022017
union all select * from dbo.Crime032017
union all select * from dbo.Crime042017
union all select * from dbo.Crime052017
union all select * from dbo.Crime062017
union all select * from dbo.Crime072017
union all select * from dbo.Crime082017
union all select * from dbo.Crime092017
union all select * from dbo.Crime102017
union all select * from dbo.Crime112017
union all select * from dbo.Crime122017
union all select * from dbo.Crime012018
union all select * from dbo.Crime022018
union all select * from dbo.Crime032018
union all select * from dbo.Crime042018
union all select * from dbo.Crime052018
union all select * from dbo.Crime062018
union all select * from dbo.Crime072018
union all select * from dbo.Crime082018
union all select * from dbo.Crime092018
union all select * from dbo.Crime102018
union all select * from dbo.Crime112018
union all select * from dbo.Crime122018
```

Table for Manchester_crime_table shown below from SQL select statement.



```
select * from manchester_crime_table
```

Results | Messages

Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name
e8197b638dd2659580fe69453057f41f61a015db0108b755d...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.290194	53.46124	On or near Parking Area	E01006186	Trafford 003B
d1fdb04413cc0961b7152e2af87150f8f716725f5241d630c7...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.301106	53.457923	On or near Nansen Close	E01006186	Trafford 003B
691c42f5c6b2bd9625f6424ecd247d7c20df5a7d7b53e5ca...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.290143	53.458175	On or near Police Station	E01006186	Trafford 003B
8f815141b9692e78f6aa26018144926f9c4d9933f6be1d7746...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.290194	53.46124	On or near Parking Area	E01006186	Trafford 003B

- **Views**

```
--- Creating a view to union tne population table 2017 and population table 2018 together using
--- the union clause statement---

create view dbo.population_total as (
select * from dbo.population_2017 union all select * from dbo.population_2018)

select * from population_total
```

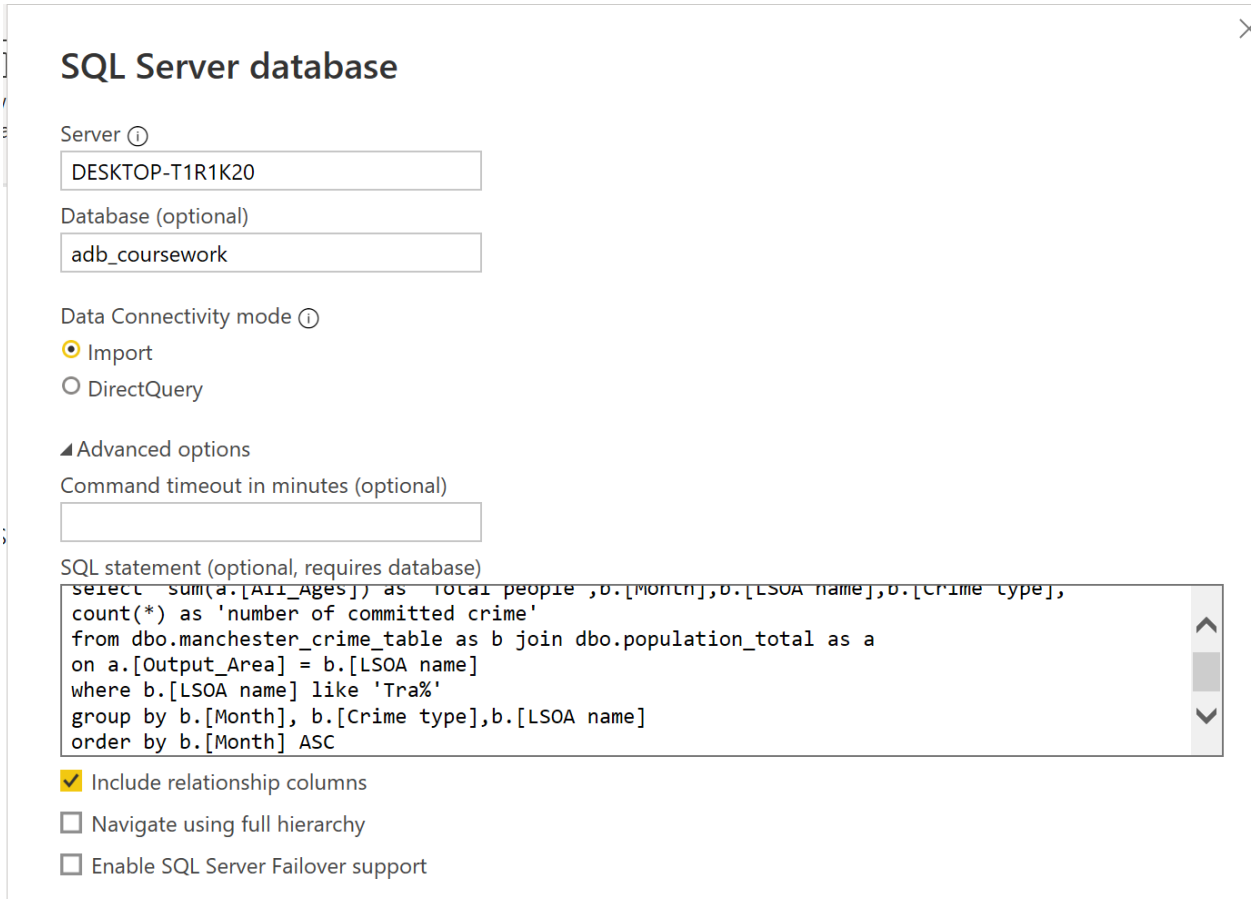
Results | Messages

Area_Codes	Output_Area	All_Ages
1020634.00	County Durham 001A	1632
1020635.00	County Durham 001B	1329
1020636.00	County Durham 001C	1725
1020654.00	County Durham 001D	1826
1020676.00	County Durham 001E	1517
1020613.00	County Durham 002A	1366
1020614.00	County Durham 002B	1376
1020622.00	County Durham 002C	1597
1020623.00	County Durham 002D	1496
1020627.00	County Durham 003A	1497
1020628.00	County Durham 003B	1817
1020629.00	County Durham 003C	1588
1020655.00	County Durham 003D	2050
1020656.00	County Durham 003E	1516
1020657.00	County Durham 003F	1512
1020661.00	County Durham 004A	2009
1020662.00	County Durham 004B	1784
1020663.00	County Durham 004C	1878
1020674.00	County Durham 004D	1585
1020678.00	County Durham 004E	1342
1020606.00	County Durham 005A	1403

6. Report Design/Summarize Report

Various summarized report from crime data called Manchester_crime_table and population data was created, and visualizations conducted on power Bi with process design as shown. The selected statement was connected to power Bi to summarize and visualization the report sees below design.

REPORT 1



The select statement inserted into the SQL server to summarize the report

DESKTOP-T1R1K20: adb_coursework

Total people	Month	LSOA name	Crime type	number of committed crime
9518	2017-01	Trafford 001E	Anti-social behaviour	*4*
3563	2017-01	Trafford 001B	Anti-social behaviour	*2*
3635	2017-01	Trafford 001C	Anti-social behaviour	*2*
9992	2017-01	Trafford 001A	Anti-social behaviour	*8*
10110	2017-01	Trafford 003A	Anti-social behaviour	*6*
8928	2017-01	Trafford 002C	Anti-social behaviour	*6*
7598	2017-01	Trafford 001D	Anti-social behaviour	*4*
3356	2017-01	Trafford 001F	Anti-social behaviour	*2*
3520	2017-01	Trafford 004B	Anti-social behaviour	*2*
3197	2017-01	Trafford 002D	Anti-social behaviour	*2*
10146	2017-01	Trafford 002B	Anti-social behaviour	*6*
15140	2017-01	Trafford 002E	Anti-social behaviour	*10*
6368	2017-01	Trafford 005E	Anti-social behaviour	*4*
30684	2017-01	Trafford 003C	Anti-social behaviour	*12*
16100	2017-01	Trafford 003B	Anti-social behaviour	*8*
3587	2017-01	Trafford 005D	Anti-social behaviour	*2*
3744	2017-01	Trafford 006C	Anti-social behaviour	*2*
4628	2017-01	Trafford 003D	Anti-social behaviour	*2*
3054	2017-01	Trafford 004A	Anti-social behaviour	*2*
6846	2017-01	Trafford 006E	Anti-social behaviour	*4*

Fig 9. report of crime type in Trafford area.

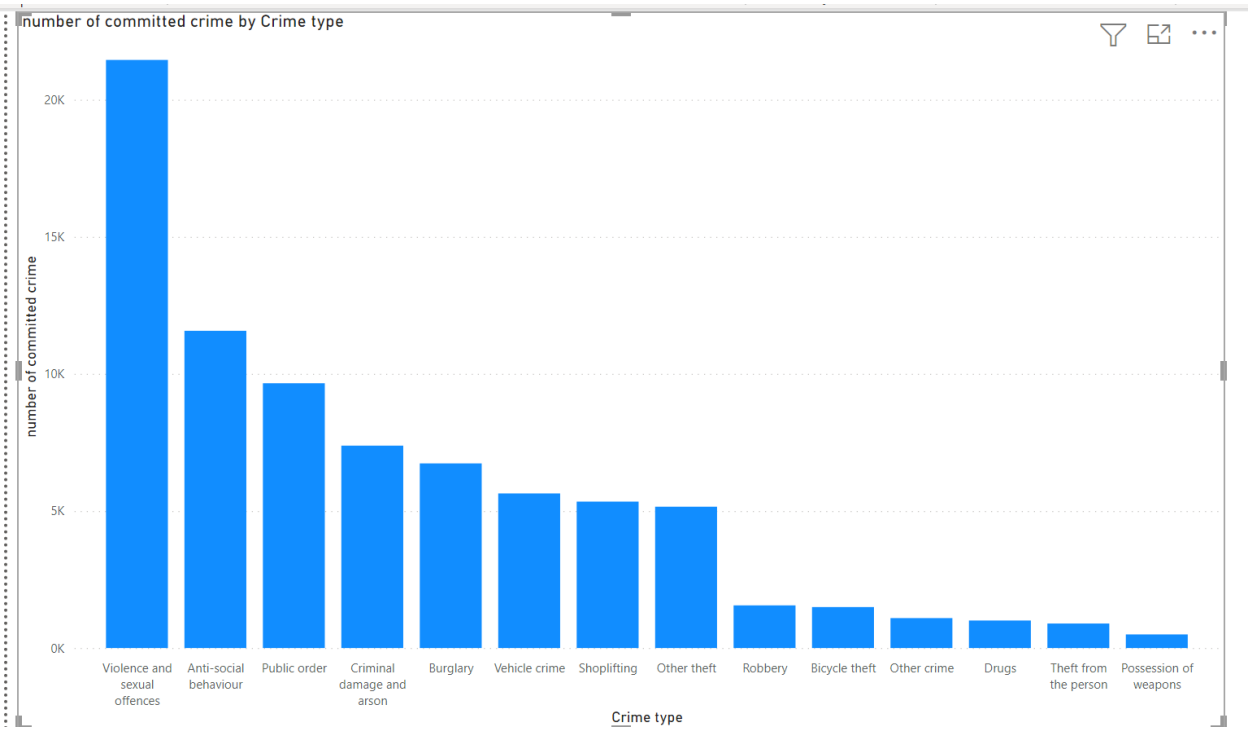


Fig 10 Number of committed crimes by crime type, we see violence and sexual offences is higher in Trafford area.

REPORT 2

SQL Server database

Server
destop-t1r1k20

Database (optional)
adb_coursework

Data Connectivity mode
Import
DirectQuery

Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

```
select  sum(a.[All_Ages]) as 'Total people',b.[Month],b.[LSOA name],b.[Crime type],
count(*) as 'number of committed crime'
from dbo.manchester_crime_table as b join dbo.population_total as a
on a.[Output_Area] = b.[LSOA name]
where b.[LSOA name] like 'Bol%'
group by b.[Month], b.[Crime type],b.[LSOA name]
```

Include relationship columns

Navigate using full hierarchy

Enable SQL Server Failover support

desktop-t1r1k20: adb_coursework

Total people	Month	LSOA name	Crime type	number of committed crime
3164	2017-01	Bolton 001A	Anti-social behaviour	2
5958	2017-01	Bolton 001C	Anti-social behaviour	4
12320	2017-01	Bolton 004A	Anti-social behaviour	8
14735	2017-01	Bolton 004B	Anti-social behaviour	10
9543	2017-01	Bolton 001E	Anti-social behaviour	6
2818	2017-01	Bolton 001D	Anti-social behaviour	2
13935	2017-01	Bolton 004E	Anti-social behaviour	10
5188	2017-01	Bolton 004C	Anti-social behaviour	4
13525	2017-01	Bolton 002B	Anti-social behaviour	10
3267	2017-01	Bolton 002E	Anti-social behaviour	2
14552	2017-01	Bolton 005B	Anti-social behaviour	8
7012	2017-01	Bolton 005E	Anti-social behaviour	4
6364	2017-01	Bolton 002C	Anti-social behaviour	4
2903	2017-01	Bolton 003A	Anti-social behaviour	2
20460	2017-01	Bolton 007B	Anti-social behaviour	10
6268	2017-01	Bolton 008A	Anti-social behaviour	4
5946	2017-01	Bolton 002D	Anti-social behaviour	4
3594	2017-01	Bolton 003E	Anti-social behaviour	2
3246	2017-01	Bolton 008B	Anti-social behaviour	2
8205	2017-01	Bolton 010B	Anti-social behaviour	6

Fig 11 shows Various crime type in Bolton area

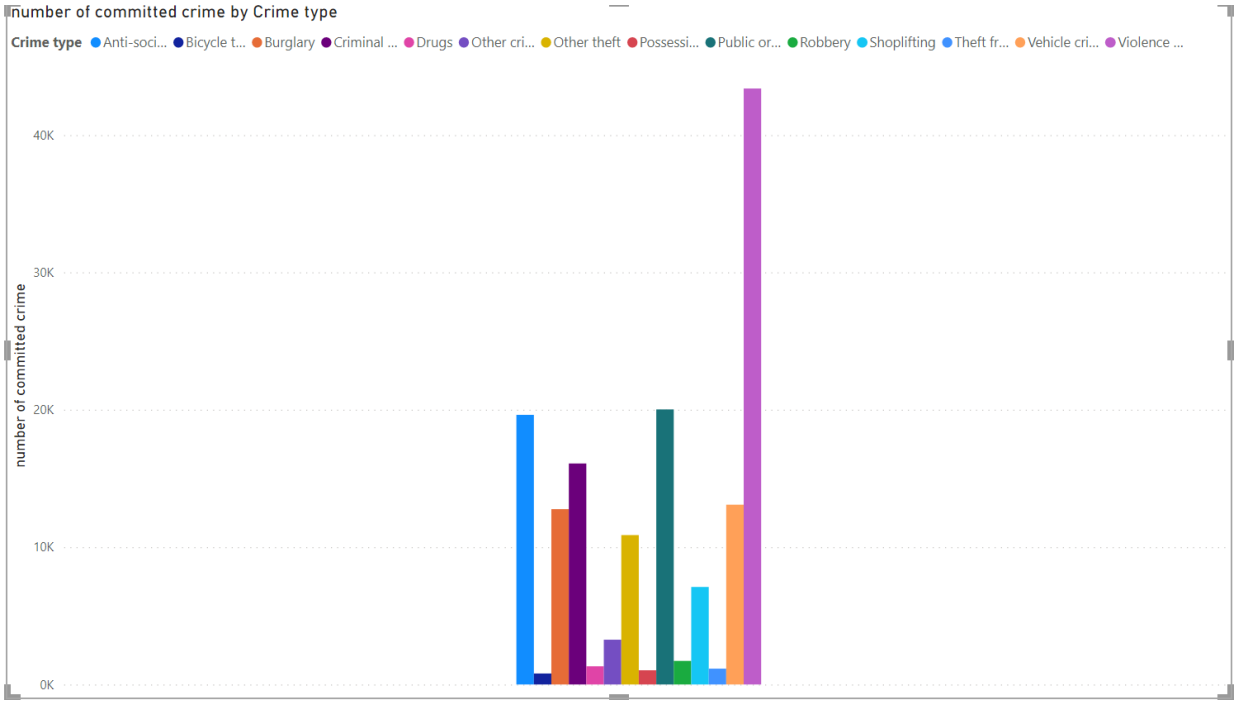


Chart also shows violence and sexual offences is higher in Bolton

REPORT 3

SQL Server database

Server ⓘ

DESKTOP-T1R1K20

Database (optional)

adb_coursework

Data Connectivity mode ⓘ

- (•) Import
- ( ) DirectQuery

▲ Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

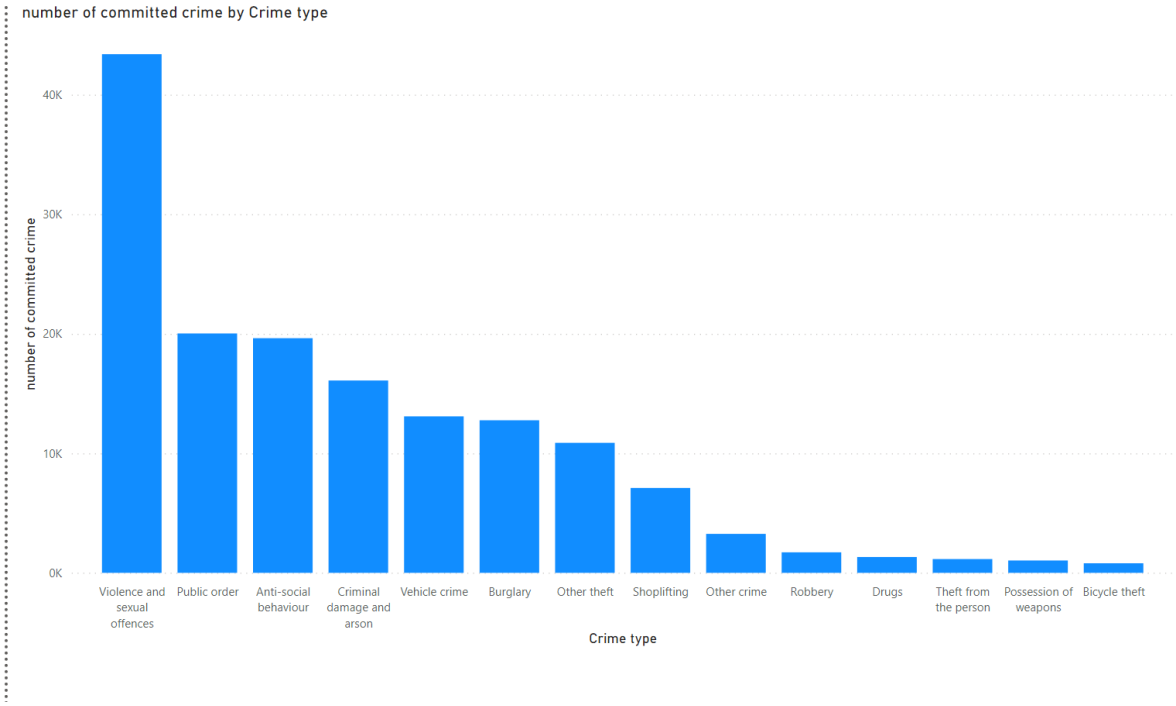
```
select  sum(a.[All_Ages]) as 'Total people',b.[Month],b.[LSOA name],b.[Crime type],
count(*) as 'number of committed crime'
from dbo.manchester_crime_table as b join dbo.population_total as a
on a.[Output_Area] = b.[LSOA name]
where b.[LSOA name] like 'sal%'
and b.[Crime type] = 'Robbery'
```

- [x] Include relationship columns
- [ ] Navigate using full hierarchy
- [ ] Enable SQL Server Failover support

DESKTOP-T1R1K20: adb_coursework

Total people	Month	LSOA name	Crime type	number of committed crime
3138	2017-01	Salford 001A	Robbery	2
3848	2017-01	Salford 004A	Robbery	2
3506	2017-01	Salford 004B	Robbery	2
3240	2017-01	Salford 007F	Robbery	2
6362	2017-01	Salford 010A	Robbery	4
12357	2017-01	Salford 010D	Robbery	6
3111	2017-01	Salford 011D	Robbery	2
3103	2017-01	Salford 012F	Robbery	2
4346	2017-01	Salford 016B	Robbery	2
5533	2017-01	Salford 016E	Robbery	2
6818	2017-01	Salford 017C	Robbery	4
2561	2017-01	Salford 020G	Robbery	2
11748	2017-01	Salford 021D	Robbery	4
2524	2017-01	Salford 022D	Robbery	2
6624	2017-01	Salford 022E	Robbery	4
4308	2017-01	Salford 022F	Robbery	2
2989	2017-01	Salford 022H	Robbery	2
3121	2017-01	Salford 023C	Robbery	2
3054	2017-01	Salford 023D	Robbery	2
7462	2017-01	Salford 024C	Robbery	4

Fig 12 Robbery activities in Salford area



Violence and sexual offences prevalence in Salford area and bicycle theft is minimal.

7. EXTRA FEATURES IMPLEMENTATION

Using the select SQL statement below and where statement to filter crime type which is vehicle crime with LSOA name .

```
select * from dbo.manchester_crime_table
where [Crime type] = 'vehicle crime'
and [LSOA name] like 'Man%'
```

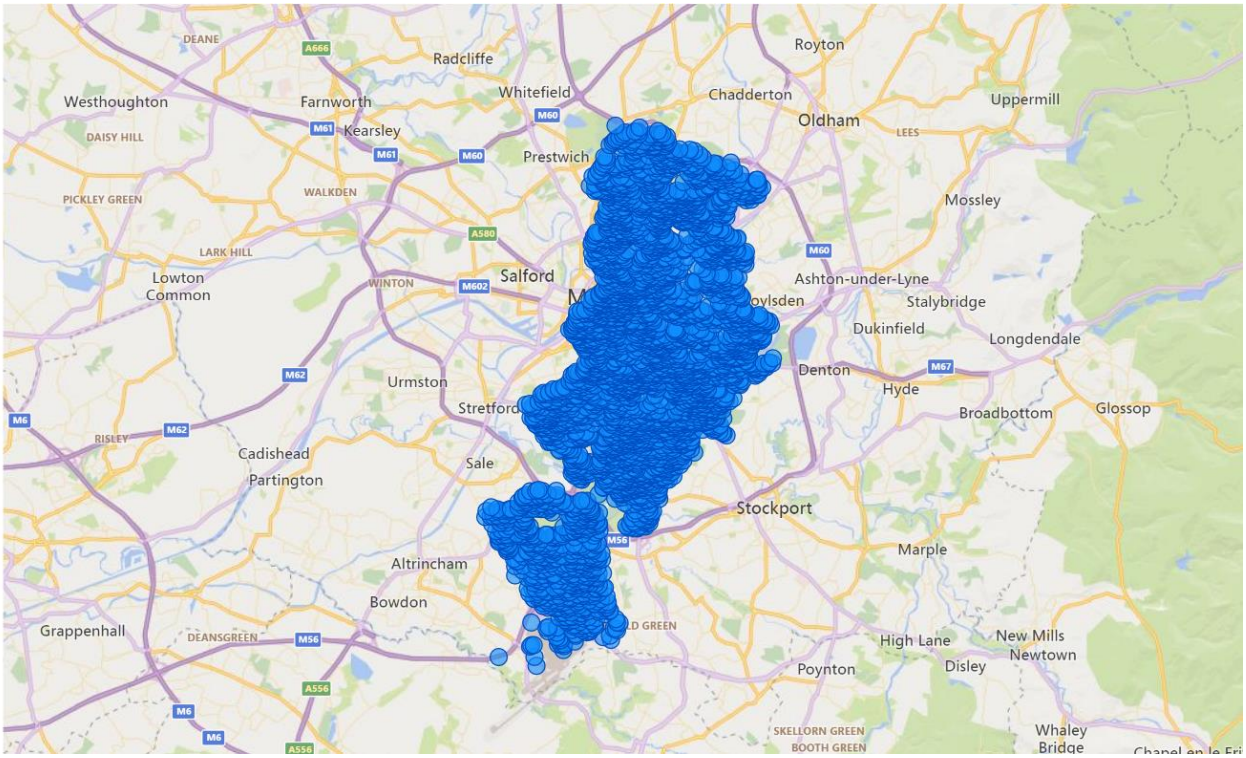
Results Messages

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name
1	95c09c794003293fdac4ba0aa10244e5dd28191171a56e47...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.261459	53.37818	On or near Petrol Station	E01005291	Manchester 050E
2	ec8720d40d57a57068ae183d739715575e344e1ba7ff678b1...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.288433	53.392527	On or near Banbury Road	E01005070	Manchester 051B
3	054ffd8143d4ab3db21ad3771636f1d6c9d4e6bc60d598c94...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.287229	53.383442	On or near Marwood Drive	E01005075	Manchester 051D
4	6141bfba8b5f8cbcca907f878f7911f2d314ff837b06ec03ebb...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.276802	53.377534	On or near Drake Avenue	E01005319	Manchester 051E
5	674759839ae805a5286564ab6610319a6248c22d23913fa6...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.244591	53.495633	On or near Sherborne Street	E01033674	Manchester 058C
6	c9a9e087cdb297a9b36a6eba8863db1b4a9f94779a65e104...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.243178	53.488562	On or near Pedestrian Sub...	E01033674	Manchester 058C
7	d10077e90a5eeff0caf730e9cb2aec2f6f7e7815460c67e369...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.249196	53.491902	On or near Parking Area	E01033674	Manchester 058C
8	f4767988a9cbff625c2794b98d701df39d7bc8316397e18b0d...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.243178	53.488562	On or near Pedestrian Sub...	E01033674	Manchester 058C
9	c145bc459b78700397df2cc82ba67b8caf37ab8c135eba7d4...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.248801	53.491265	On or near Langston Street	E01033674	Manchester 058C
10	70fa5935f93c94bf630515bde2338d83bee221eb88998c8f04...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246328	53.381943	On or near Ashurst Road	E01005290	Manchester 052A

Crime ID	Month	Reported by	Falls within	Longitude	Latitude
95c09c794003293fdac4ba0aa10244e5dd28191171a56...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.261459	53.37818
ec8720d40d57a57068ae183d739715575e344e1ba7ff6...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.288433	53.392527
054ffd8143d4ab3db21ad3771636f1d6c9d4e6bc60d59...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.287229	53.383442
6141bfba8b5f8cbcca907f878f7911f2d314ff837b06ec03...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.276802	53.377534
674759839ae805a5286564ab6610319a6248c22d23913...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.244591	53.495633
c9a9e087cdb297a9b36a6eba8863db1b4a9f94779a65e...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.243178	53.488562
d10077e90a5eeff0caf730e9cb2aec2f6f7e7815460c67e...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.249196	53.491902
f4767988a9cbff625c2794b98d701df39d7bc8316397e1...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.243178	53.488562
c145bc459b78700397df2cc82ba67b8caf37ab8c135eba...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.248801	53.491265
70fa5935f93c94bf630515bde2338d83bee221eb88998c...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246328	53.381943
ba386bcb88ceb85a9d0ddd11fe0bcd89a43c5150f1040...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.251748	53.383244
3ef58a3aaec575232e5a4dcd8b3ff38b7478343f5e6765...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.25264	53.379008
11039ee3264b38e0f043fd858ae0f456d4b4c2beb2348...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.25264	53.379008
8a8fcdcbc1f98f5e8cdc35f284d4b6c1543d6688a04c8f9...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246805	53.375964
90a135d7179fb1f7a3431d510d43b9b002c1b8b734a26...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246441	53.372792
1f3733542e8979cabde0f94e97fe3cf71b5f55ad23d8e11...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.244751	53.377011
07d0a8ce8a26e42ae278931238f2e64057de2ec4547b3...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246441	53.372792
8158213e0da292ac2a1ef9608302518b5b2e587a9fbf4e...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.24859	53.377956
f5f0fe30b390b51f1ef13545118422ee8a73de6a497d1e...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.246441	53.372792
07e0dca1f78e5b35890321e64bf0cc206e879afe2a89b7...	2017-01	Greater Manchester Police	Greater Manchester Police	-2.24859	53.377956

Fig above shows vehicle crime for greater Manchester only.

Visualizing the Vehicle crime in Greater Manchester with QGIS using MSSQL Connector. Use OpenStreetMap as Open Layers plugin



Visualize the Anti-social behaviors crimes in Salford with QGIS using MSSQL Connector. Using Google Satellite map as Open Layers plugin.

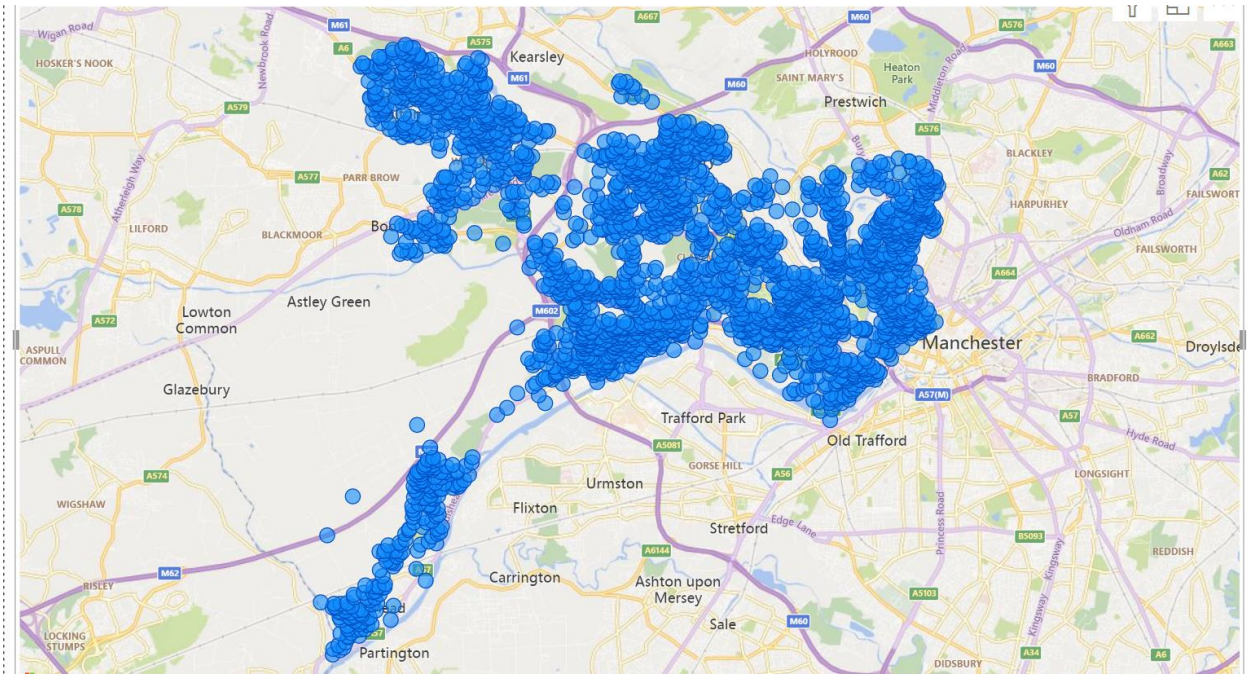
Using the below Select SQL statement and filter to select only crime type 'anti-social behavior' in Salford as below.

```
select * from dbo.manchester_crime_table
where [Crime type] = 'Anti-social behaviour'
and [LSOA name] like 'sal%'
```

100 %

Results | Messages

	Crime ID	Month	Reported by	Falls within	Longitude	Latitude	Location	LSOA code	LSOA name	Crime type
1	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.290913	53.471539	On or near Winnipeg Quay	E01032685	Salford 028E	Anti-social behaviour
2	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.285924	53.469061	On or near Merchants Quay	E01032686	Salford 028F	Anti-social behaviour
3	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.280242	53.468634	On or near Weedall Avenue	E01032912	Salford 028G	Anti-social behaviour
4	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.268679	53.47427	On or near Freya Grove	E01032915	Salford 028I	Anti-social behaviour
5	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.403423	53.454511	On or near Shopping Area	E01005640	Salford 029B	Anti-social behaviour
6	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.438824	53.43089	On or near Allotment Road	E01005617	Salford 030B	Anti-social behaviour
7	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.428914	53.43614	On or near Preston Avenue	E01005620	Salford 030E	Anti-social behaviour
8	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.432477	53.434221	On or near Pedestrian Subway	E01005621	Salford 030F	Anti-social behaviour
9	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.4259	53.535087	On or near Thornfield Crescent	E01005660	Salford 001B	Anti-social behaviour
10	NULL	2017-01	Greater Manchester Police	Greater Manchester Police	-2.432135	53.536898	On or near Brookhurst Lane	E01005664	Salford 001D	Anti-social behaviour



7. Database Security

Database security is not entirely difficult to manage, but consistent vigilance and continues efforts to secure a database is required as often as possible. There are certain things that needs to be put into consideration among are:

1. secure the physical environment if the database is in a physical server system. Limit access of employers and prevent staff from accessing IT related materials.
2. Network Security: database administrator should check every point of the network is strong, the TCP/IP should be changed from default as this a point of attack from trojan horse. Also ensure the fireworks and antivirus software are updated regularly.
3. Avoid over deployment of features. Features not needed should be disabled and right given out to users should be reduced to prevent hackers gaining access to the database.

8. Database Backup and Restore Strategy

Database consists of backup part and restore part, knowing the size, frequency of backup type of backup and place to store the backup media going a long way to facilitate this strategy. For this task it assumed to be full back up on interval of twice in a month. The second part involves the restore part, two things to consider who will restore the database and how the restore will be performed.

9.Data Science/Business Intelligence Techniques

Business intelligence tools help decision makers analyses data so they can get insight from data and result driven solution for various business questions. Most organization uses business intelligence tools and data science related tools also to forecast the future using certain algorithm like unsupervised learning and supervised learning like classification in random forest and k-clustering to cluster a certain location in this task. Business intelligence tools like power BI and tableau, for this task I used Microsoft power BI.

10. Data Privacy, Ethical and legal issues

Ensuring data privacy as default in the first place during design is vital, as individuals are not responsible for protecting data, is an organizational issue. Privacy design involves implementing privacy from start to finish, so any data that enters the database is secure from the beginning and retained safely. For privacy reasons the police data is not specific to an address.

11. Conclusion

The design and implementation of this task made use of screen shots, working T-SQL statement, power BI files and QGIS Maps and using power BI as presentation layer to implement the frontend. I have been able to use my knowledge and expertise to implement and design a simple reporting system using SQL server and Micro-soft Power BI.

The principal investigator will be able to use this straightforward design to further understand the idea behind the survey and communicate to stakeholders accordingly and evaluate or

review policies solutions to eradicate or minimize crime in different area code, from the summarized report of crime in elevated high-risk area.

References

https://www.geeksforgeeks.org/domain-constraints-in-dbms/

Lee, W. W., et al. (2016). "An ethical approach to data privacy protection."

Thang, N. and N. T. T. Hang (2018). "Leaving no one behind in a growing Vietnam: The story from Young Lives." Leaving no one behind in a growing Vietnam: the story from Young Lives.

Vu, T. T. (2021). "Early childhood education in Vietnam, history, and development." International Journal of Child Care and Education Policy **15**(1): 1-18.