Big-Data

Part-2: Coursework

Name: Prakash Dahal

Student Id: 1828421

Date: 2019/19/04

Group Members:

- Prakash Dahal
- Rajan Sapkota (Sharma)

Table of Contents

A. Report

1. Introduction to Big Data

The term Big-Data represents for large amount of data which can be analyzed and processed. The word ‘Big-data’ became popular since it was introduced by John Mashey in 1990s because data become so useful for analysis, to know the behavior or to get extract information out of it. Better operational efficiency, improving customer efficiency, intelligence for decision taking and many more are the best outcome of processing big data. Example; User data, sensor data, satellite data, etc. gives large data each second. Data also can be in structured or semi-structured or unstructured format.

Data is not regarded as big because of its volume but according to IBM, it is represented by three V’s dimensions. They are given below:

a. Volume:
   Volume represents data in large size like hundreds of terabytes, petabytes, yottabytes or zettabytes.
b. Velocity:
   Velocity represents for the speed of data change. At what speed data is being generated.
c. Variety:
   Variety represents different types of data which may include text, audio, video, clicks streams, touch sensors and so on.
   Other different V’s also represent big-data like Veracity, Value and so on (Sriramoju, 2017) (Chitresh Verma, 2016).

2. Evaluation of the Tools and Techniques

Different tools and techniques are there for processing big-data. There are tools available for processing big data. Hadoop, NoSql, Hive, Mongo DB, etc. are some of the popular tools used for handling big-data (M. Sowmya, 2017). Some of these are explained below:

a. Oracle and Excel

Excel is a spread sheet where data can be saved in the form of row and column. Data saved in excel can be directly imported in oracle and use the data. It's easy to store data in excel and load data easily into database but it can store limited number of row and columns only. In other hand Excel is not good for storing various data like image, audio and so on.

Oracle is a relational database management system. It treats data as a unit which helps for the extraction of related data. Oracle is the first database designed for enterprise which is flexible and cost effective to maintain information. Roll back features make it more unique but still in terms of big data oracle is not good for handling large data and variety of data. But Oracle 12c has some updated features (Cyran, 2005) (Rick Greenwald, 2003).

b. MongoDB

MongoDB is open-source and NoSQL database which means Not only Structured Query Language, so it is not restricted by structured query language. It is highly flexible and scalable which make it more popular. It supports for complex data structure and data index. It stores data in JSON format by automatic subdivision and geographical spatial index. In terms of big-data, it has document storage limitation, but it supports structured, semi structured and unstructured data. Convenient storage, functional diversity, reliability etc. makes mongo DB stronger (Chaokui Li, 2014) (Prabagaren, 2014).

c. MapReduce Framework

Map Reduce is a parallel data processing technique. It consists of two main phases mapper and reducer. Mapper reads the file and make mapping. It transforms input record to intermediate key-value pairs. After this reducer aggregate the intermediate result and gives output. Map reduce has merged with HDFS for better performance in HDFS. It does not support stream data processing also processing of complex data analytical is difficult. It performs on homogeneous data, so to solve this issue map join reduce was introduced.

Map reducer processes slowly for small file and not applicable for synchronized data (Ruchi Bhardwaj, 2014).

d. Hadoop

The concept of Hadoop is officially established by Apache which is open source. The journey of Hadoop started from first version (i.e. Hadoop 1.0) has made lots of impact. The technique of cluster computing, grid computing, cloud computing, distributed file system handling etc. and security support and enhancement from Apache Rhino, Apache Ranger, and Apache Knox has made Hadoop stronger. It handles fault-tolerance. MapReduce of Hadoop is dependent on YARN for parallel processing which makes job scheduling and Cluster Resource management. Hadoop supports big-data characteristics like volume, velocity variety and veracity where each has its own role in the framework (Gurjit singh Bhathal, 2018).

Files / Technologies	CSV	Json
Oracle	➔ Data cleaning is essential ➔ Having big amount of data results in slow processing ➔ Data are broken down for better result	➔ No such Task is performed.
MongoDB	➔ No such Task is performed.	➔ Concept of NoSql using Mongo helps to manipulate data easily ➔ Coding and its pattern to check data for human needs are better than Oracle
Map Reduce	➔ Data Cleaning are performed by mapper and reducer program written in Java ➔ Reducer program also helps because it reads whole file by compiling Java file.	➔ No such task is performed
Hadoop	➔ Could have the potential to store read or manipulate large volume of data with effective velocity ➔ Could easily manipulate any forms of data. ➔ Could easily reduce the veracity on the data and results in better output	➔ Could have the potential to store read or manipulate large volume of data with effective velocity ➔ Could easily manipulate any forms of data. Could easily reduce the veracity on the data and results in better output

3. Matrix

	Volume	**Velocity**	**Variety**	**Veracity**
Oracle	Increase in volume slowdowns the performance of the oracle. The processing time periods are a bit longer than Mongo DB and Hadoop.	Daily huge data are generated in an unstructured format where oracle falls back for this.	Oracle is based on relational format and supports structured data and not able to applicable for unstructured data.	Oracle must have cleaned and structured data format but in real world, impure data are generated daily.
Mongo DB	Sharding, fault-tolerance and replication are three major key factors on mongo DB to handle huge data.	It is based on NoSQL and has indexing feature which maintains changing data.	It supports structured to unstructured data like CSV, JSON and so on.	Accurate data can be achieved from Mongo DB since it handles impure data too.
Hadoop	It includes HDFS which work on distributed way and map reduce handles large data easily.	Highly changing data on Yahoo, IBM, Facebook are handled by Hadoop since it can run on multiple nodes and clusters.	It handles all types of data structured, semi structured and unstructured data.	Hadoop supports impure data and can work parallelly on the given data.

4. Conclusions and recommendations

Data are increasing day by day where comes the concept of Big-data. Data must be managed in the systematic way. Big data is represented by volume, velocity, variety or veracity. Oracle, MongoDB and Hadoop are highly used for handling large data.

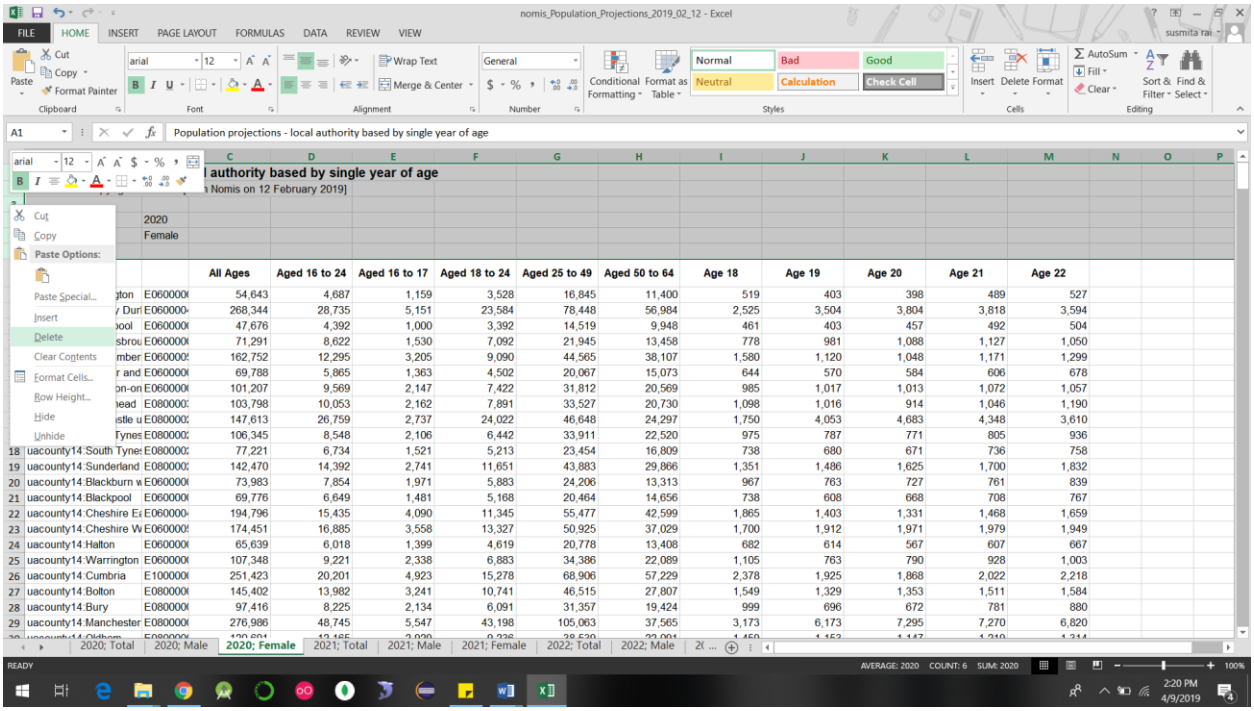
All tools have its own merits and demerits. Oracle and mongo DB are suitable for small data where large volume data can be handled by Hadoop since it used map reduce technique. Mongo also supports for semi and unstructured data where oracle lags.

B. Investigation

a. Cleaning

Data cleaning is necessary for investigation so excel is used for cleaning. Cleaning Female data 2020:

Deleting unrequired data



			All Ages	Aged 16 to 24	Aged 16 to 17	Aged 18 to 24	Aged 25 to 49	Aged 50 to 64	Age 18	Age 19	Age 20	Age 21	Age 22
	gton	E060000	54,643	4,687	1,159	3,528	16,845	11,400	519	403	398	489	527
	/ Durl	E060000	268,344	28,735	5,151	23,584	78,448	56,984	2,525	3,504	3,804	3,818	3,594
	ool	E060000	47,676	4,392	1,000	3,392	14,519	9,948	461	403	457	492	504
	sbrou	E060000	71,291	8,622	1,530	7,092	21,945	13,458	778	981	1,088	1,127	1,050
	mber	E060000	162,752	12,295	3,205	9,090	44,565	38,107	1,580	1,120	1,048	1,171	1,299
	r and	E060000	69,788	5,865	1,363	4,502	20,067	15,073	644	570	584	606	678
	on-on	E060000	101,207	9,569	2,147	7,422	31,812	20,569	985	1,017	1,013	1,072	1,057
	head	E080000	103,798	10,053	2,162	7,891	33,527	20,730	1,098	1,016	914	1,046	1,190
	stle u	E080000	147,613	26,759	2,737	24,022	46,648	24,297	1,750	4,053	4,683	4,348	3,610
	Tynes	E080000	106,345	8,548	2,106	6,442	33,911	22,520	975	787	771	805	936
18 uacounty14 South Tyne		E080000	77,221	6,734	1,521	5,213	23,454	16,809	738	680	671	736	758
19 uacounty14 Sunderland		E080000	142,470	14,392	2,741	11,651	43,883	29,866	1,351	1,486	1,625	1,700	1,832
20 uacounty14 Blackburn w		E060000	73,983	7,854	1,971	5,883	24,206	13,313	967	763	727	761	839
21 uacounty14 Blackpool		E060000	69,776	6,649	1,481	5,168	20,464	14,656	738	608	668	708	767
22 uacounty14 Cheshire Ea		E060000	194,796	15,435	4,090	11,345	55,477	42,599	1,865	1,403	1,331	1,468	1,659
23 uacounty14 Cheshire W		E060000	174,451	16,885	3,558	13,327	50,925	37,029	1,700	1,912	1,971	1,979	1,949
24 uacounty14 Halton		E060000	65,639	6,018	1,399	4,619	20,778	13,408	682	614	567	607	667
25 uacounty14 Warrington		E060000	107,348	9,221	2,338	6,883	34,386	22,089	1,105	763	790	928	1,003
26 uacounty14 Cumbria		E100000	251,423	20,201	4,923	15,278	68,906	57,229	2,378	1,925	1,868	2,022	2,218
27 uacounty14 Bolton		E080000	145,402	13,982	3,241	10,741	46,515	27,807	1,549	1,329	1,353	1,511	1,584
28 uacounty14 Bury		E080000	97,416	8,225	2,134	6,091	31,357	19,424	999	696	672	781	880
29 uacounty14 Manchester		E080000	276,986	48,745	5,547	43,198	105,063	37,565	3,173	6,173	7,295	7,270	6,820

nomis_Population_Projections_2019_02_12 - Excel

490	gor:Yorkshire and The H	E120000	2,786,461	302,313	59,132	243,181	859,856	540,961	29,497	34,854	36,630	36,984	36,001
491	gor:East Midlands	E120000	2,447,823	252,884	50,923	201,961	748,314	488,790	25,369	29,004	30,482	30,859	29,170
492	gor:West Midlands	E120000	2,996,752	314,869	65,719	249,150	936,696	566,553	32,032	33,425	34,257	35,565	37,104
493	gor:East	E120000	3,198,428	285,165	67,567	217,598	1,002,345	626,436	31,689	27,644	27,494	29,796	31,800
494	gor:London	E120000	4,568,010	461,680	94,958	366,722	1,822,818	755,309	44,929	42,972	44,294	48,072	55,003
			4,697,632	447,796	101,287	346,509	1,458,607	924,838	48,579	47,419	47,767	49,637	50,117
			2,877,634	276,659	57,824	218,835	832,980	586,612	28,366	31,379	32,245	32,474	31,892
498			28,635,638	2,858,705	602,065	2,256,640	9,231,555	5,502,508	292,037	302,400	311,853	324,385	332,169

um because of rounding.

Cut
Copy
Paste Options:
Paste Special...
Insert
Delete
Clear Contents
Format Cells...
Row Height...
Hide
Unhide

2020; Total | 2020; Male | 2020; Female | 2021; Total | 2021; Male | 2021; Female | 2022; Total | 2022; Male

2:21 PM 4/9/2019

nomis_Population_Projections_2019_02_12 - Excel

Area		Aged 16 to 24	Aged 16 to 17	Aged 18 to 24	Aged 25 to 49	Aged 50 to 64	Age 18	Age 19	Age 20	Age 21	Age 22
uacounty14:Darlington	E06	4,687	1,159	3,528	16,845	11,400	519	403	398	489	527
uacounty14:County Durh	E06	28,735	5,151	23,584	78,448	56,984	2,525	3,504	3,804	3,818	3,594
uacounty14:Hartlepool	E06	4,392	1,000	3,392	14,519	9,948	461	403	457	492	504
uacounty14:Middlesbrou	E06	8,622	1,530	7,092	21,945	13,458	778	981	1,088	1,127	1,050
uacounty14:Northumber	E06	12,295	3,205	9,090	44,565	38,107	1,580	1,120	1,048	1,171	1,299
uacounty14:Redcar and	E06	5,865	1,363	4,502	20,067	15,073	644	570	584	606	678
uacounty14:Stockton-on	E06	9,569	2,147	7,422	31,812	20,569	985	1,017	1,013	1,072	1,057
uacounty14:Gateshead	E08	10,053	2,162	7,891	33,527	20,730	1,098	1,016	914	1,046	1,190
uacounty14:Newcastle u	E08	26,759	2,737	24,022	46,648	24,297	1,750	4,053	4,683	4,348	3,610
uacounty14:North Tynes	E08	8,548	2,106	6,442	33,911	22,520	975	787	771	805	936
uacounty14:South Tynes	E08	6,734	1,521	5,213	23,454	16,809	738	680	671	736	758
uacounty14:Sunderland	E08	14,392	2,741	11,651	43,883	29,866	1,351	1,486	1,625	1,700	1,832
uacounty14:Blackburn w	E06	7,854	1,971	5,883	24,206	13,313	967	763	727	761	839
uacounty14:Blackpool	E06	6,649	1,481	5,168	20,464	14,656	738	608	668	708	767
uacounty14:Cheshire East	E060000	194,796	15,435	4,090	11,345	55,477	42,599	1,865	1,403	1,331	1,468	1,659
uacounty14:Cheshire West	E060000	174,451	16,885	3,558	13,327	50,925	37,029	1,700	1,912	1,971	1,979	1,949
uacounty14:Halton	E060000	65,639	6,018	1,399	4,619	20,778	13,408	682	614	567	607	667
uacounty14:Warrington	E060000	107,348	9,221	2,338	6,883	34,386	22,089	1,105	763	790	928	1,003
uacounty14:Cumbria	E100000	251,423	20,201	4,923	15,278	68,906	57,229	2,378	1,925	1,868	2,022	2,218
uacounty14:Bolton	E080000	145,402	13,982	3,241	10,741	46,515	27,807	1,549	1,329	1,353	1,511	1,584
uacounty14:Bury	E080000	97,416	8,225	2,134	6,091	31,357	19,424	999	696	672	781	880
uacounty14:Manchester	E080000	276,986	48,745	5,547	43,198	105,063	37,565	3,173	6,173	7,295	7,270	6,820
uacounty14:Oldham	E080000	120,691	12,165	2,929	9,236	38,539	22,091	1,459	1,153	1,147	1,219	1,314
uacounty14:Rochdale	E080000	111,655	10,677	2,566	8,111	36,451	21,432	1,172	968	945	1,094	1,194
uacounty14:Salford	E080000	127,542	14,260	2,566	11,694	45,926	21,561	1,280	1,424	1,572	1,657	1,796
uacounty14:Stockport	E080000	150,547	11,923	3,119	8,804	47,903	30,098	1,410	999	987	1,081	1,265
uacounty14:Tameside	E080000	114,975	10,113	2,394	7,719	37,120	23,133	1,165	921	888	1,056	1,109
uacounty14:Trafford	E080000	122,773	9,560	2,951	6,609	40,815	23,971	1,333	702	672	757	917
uacounty14:Wigan	E080000	163,957	14,499	3,487	11,012	52,086	34,164	1,669	1,376	1,337	1,476	1,573

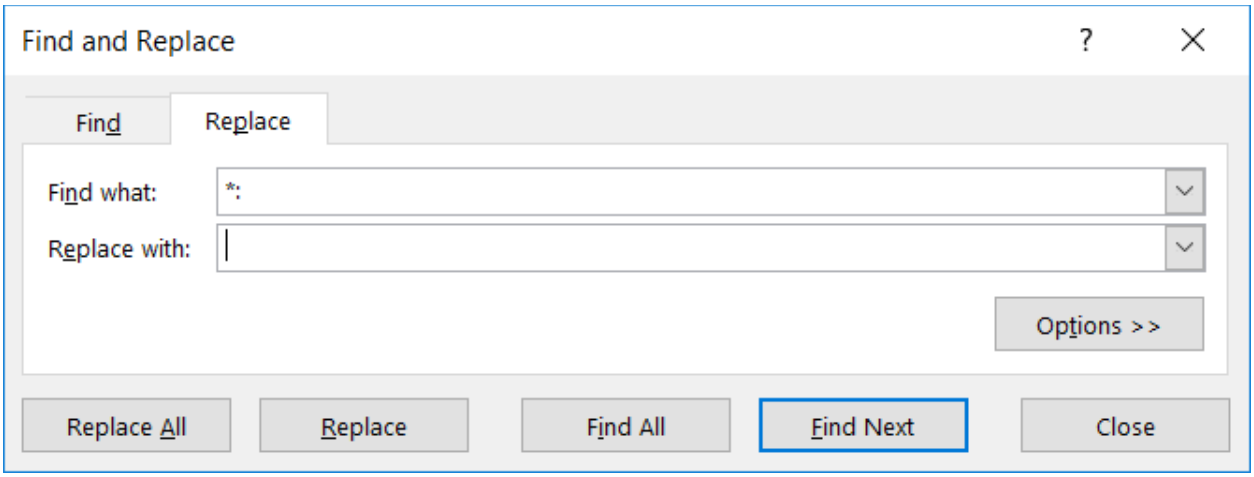
Cut
Copy
Paste Options:
Paste Special...
Insert
Delete
Clear Contents
Format Cells...
Column Width...
Hide
Unhide

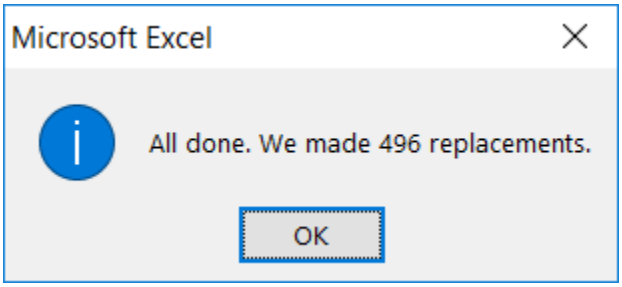
2020; Total | 2020; Male | 2020; Female | 2021; Total | 2021; Male | 2021; Female | 2022; Total | 2022; Male

AVERAGE: 5611.472062 COUNT: 3975 SUM: 19483031

2:23 PM 4/9/2019

Replacing value before ‘**:**’



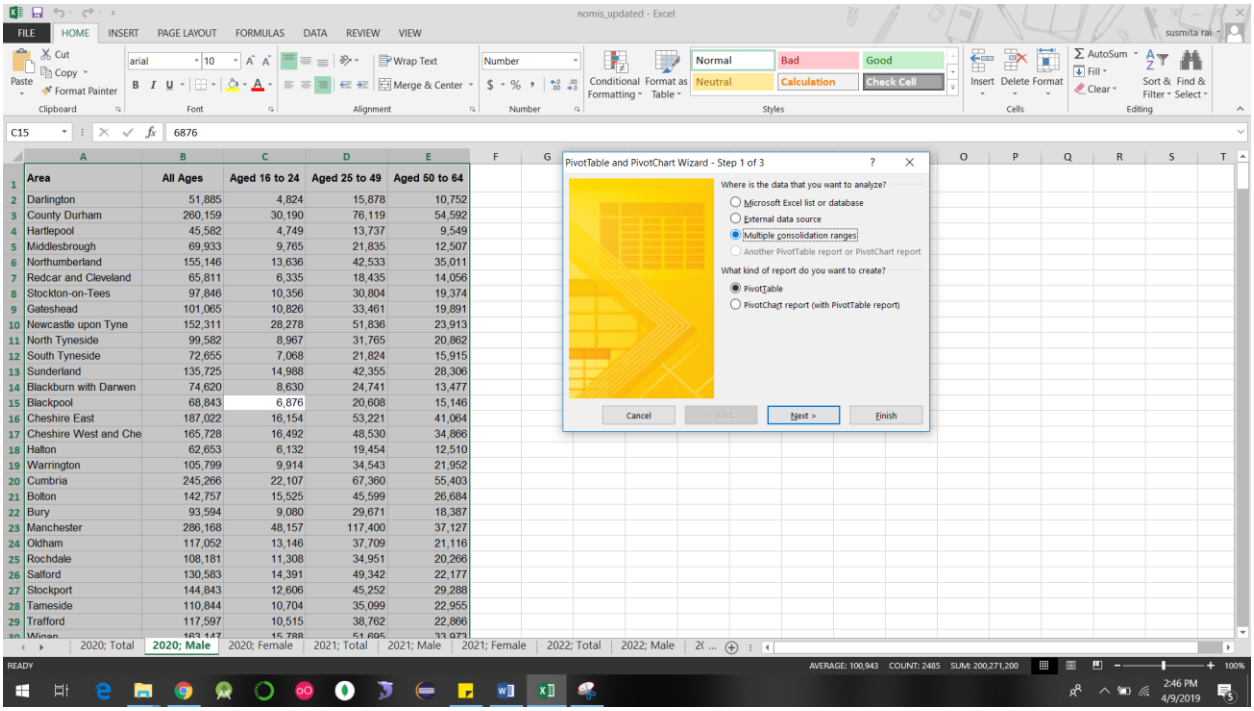


	A	B	C	D	E
1	**Area**	**All Ages**	**Aged 16 to 24**	**Aged 25 to 49**	**Aged 50 to 64**
2	Darlington	54,643	4,687	16,845	11,400
3	County Durham	268,344	28,735	78,448	56,984
4	Hartlepool	47,676	4,392	14,519	9,948
5	Middlesbrough	71,291	8,622	21,945	13,458
6	Northumberland	162,752	12,295	44,565	38,107
7	Redcar and Cleveland	69,788	5,865	20,067	15,073
8	Stockton-on-Tees	101,207	9,569	31,812	20,569
9	Gateshead	103,798	10,053	33,527	20,730
10	Newcastle upon Tyne	147,613	26,759	46,648	24,297
11	North Tyneside	106,345	8,548	33,911	22,520
12	South Tyneside	77,221	6,734	23,454	16,809
13	Sunderland	142,470	14,392	43,883	29,866
14	Blackburn with Darwen	73,983	7,854	24,206	13,313
15	Blackpool	69,776	6,649	20,464	14,656
16	Cheshire East	194,796	15,435	55,477	42,599
17	Cheshire West and Che	174,451	16,885	50,925	37,029
18	Halton	65,639	6,018	20,778	13,408
19	Warrington	107,348	9,221	34,386	22,089
20	Cumbria	251,423	20,201	68,906	57,229
21	Bolton	145,402	13,982	46,515	27,807
22	Bury	97,416	8,225	31,357	19,424
23	Manchester	276,986	48,745	105,063	37,565
24	Oldham	120,691	12,165	38,539	22,091
25	Rochdale	111,655	10,677	36,451	21,432
26	Salford	127,542	14,260	45,926	21,561
27	Stockport	150,547	11,923	47,903	30,098
28	Tameside	114,975	10,113	37,120	23,133

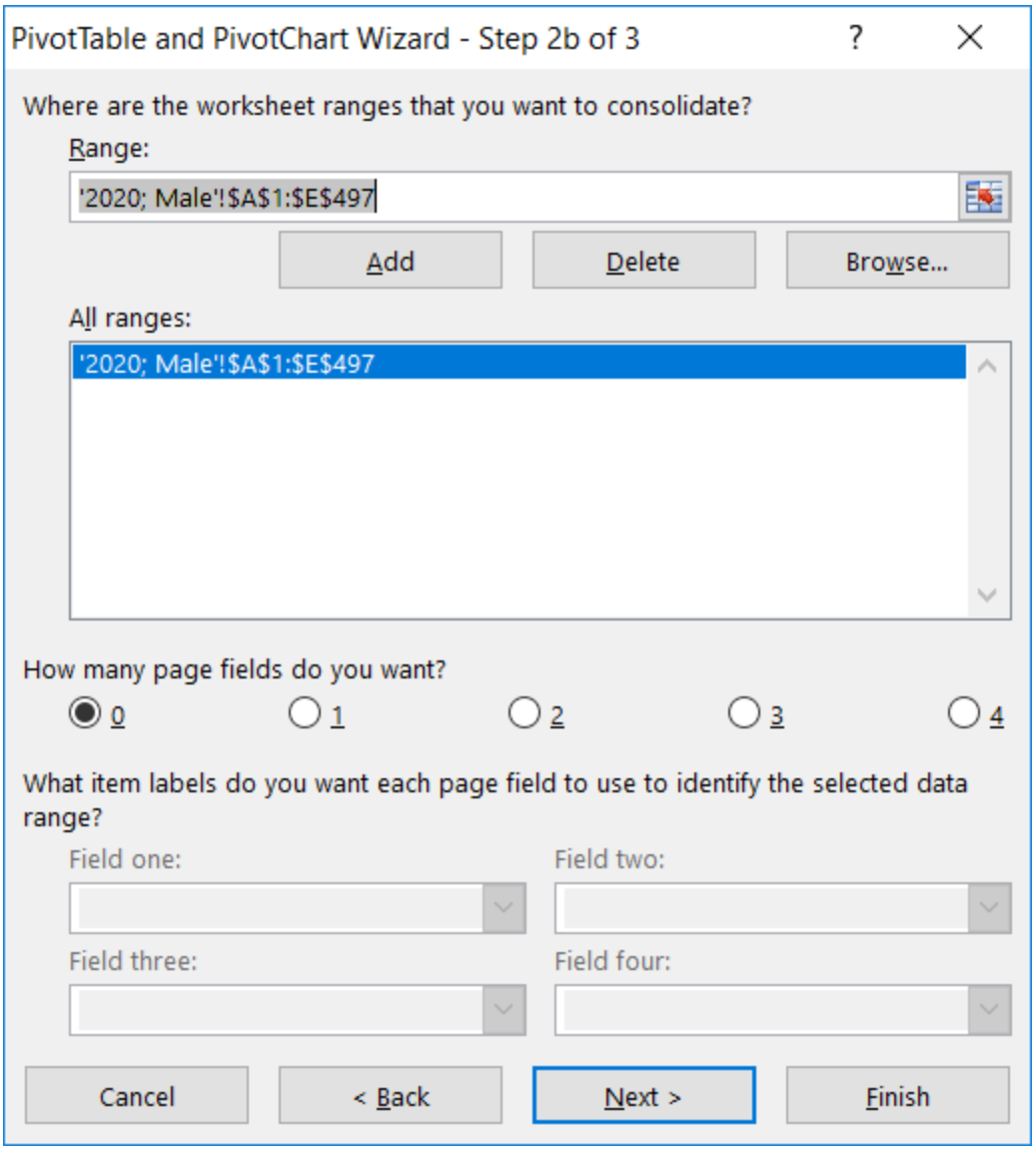
Similarly, other data were cleaned.

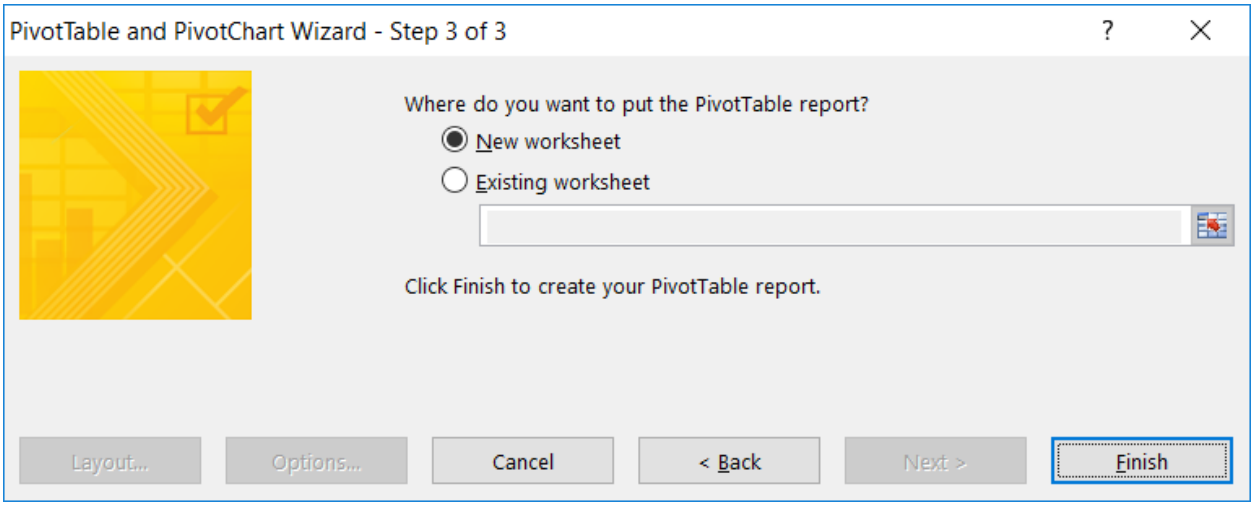
Manipulation:

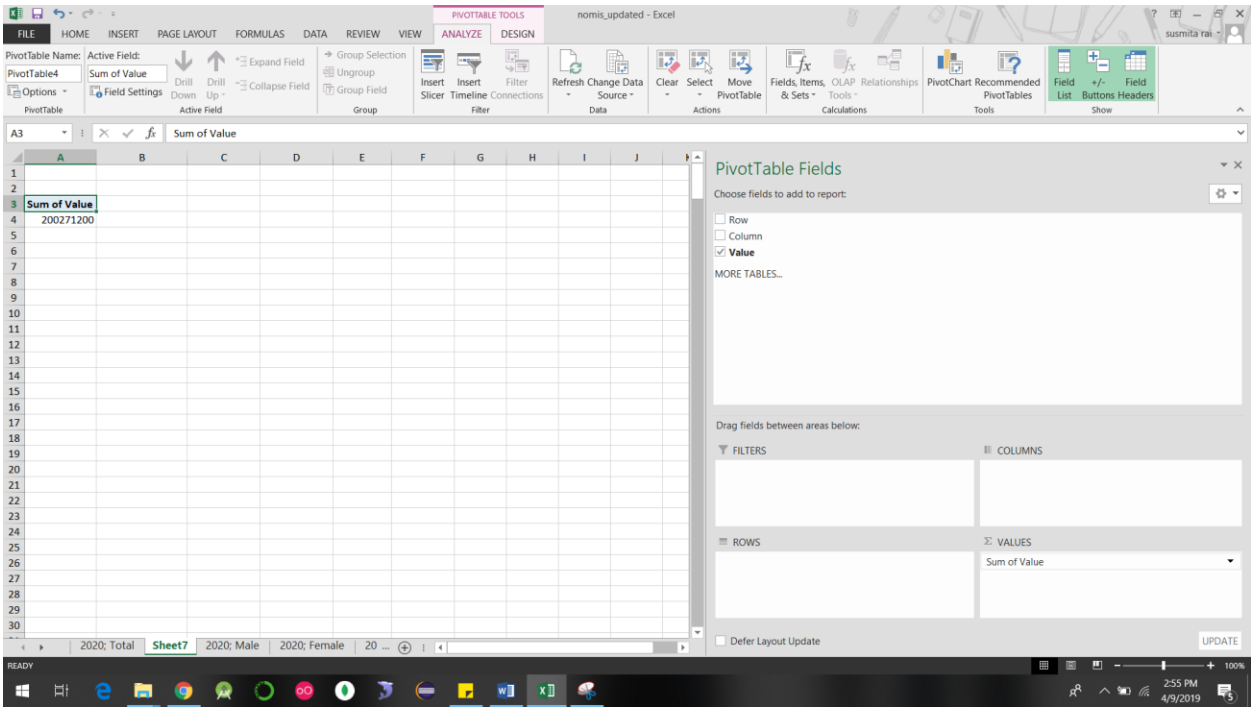
Data need to be converted into required format changing it into pivot.

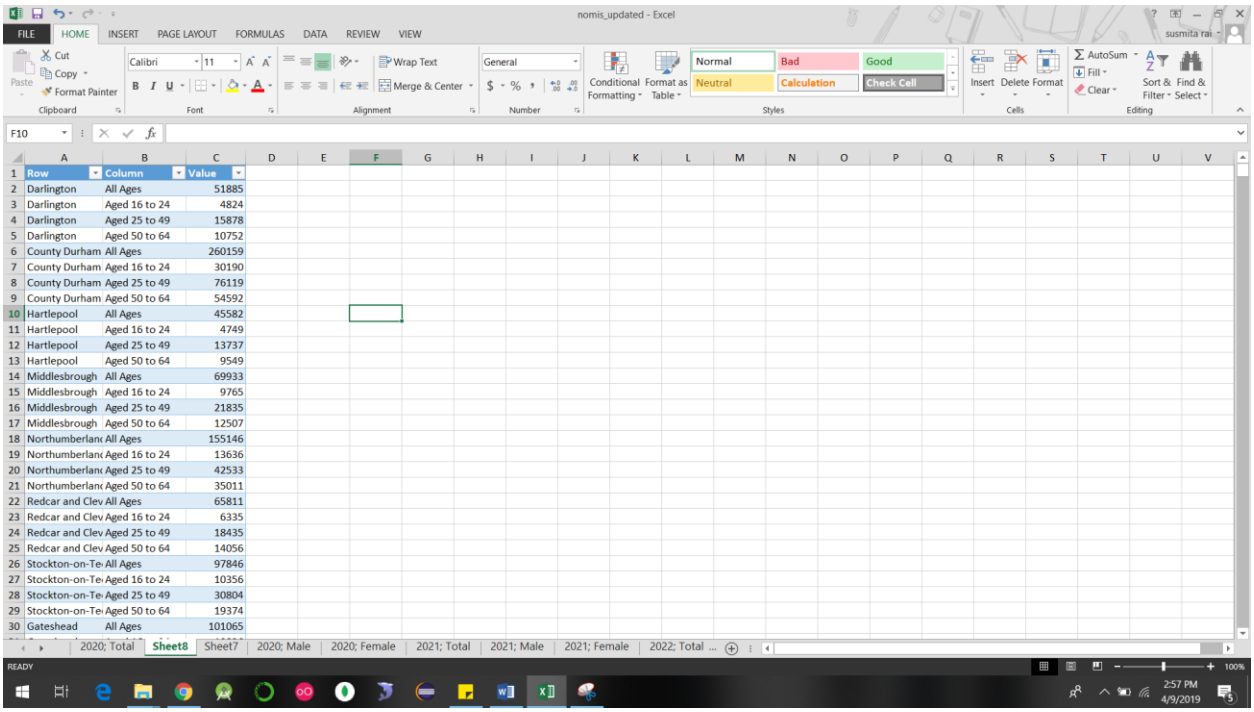


Area	All Ages	Aged 16 to 24	Aged 25 to 49	Aged 50 to 64
Darlington	51,885	4,824	15,878	10,752
County Durham	260,159	30,190	76,119	54,592
Hartlepool	45,582	4,749	13,737	9,549
Middlesbrough	69,933	9,765	21,835	12,507
Northumberland	155,146	13,636	42,533	35,011
Redcar and Cleveland	65,811	6,335	18,435	14,056
Stockton-on-Tees	97,846	10,356	30,804	19,374
Gateshead	101,065	10,826	33,461	19,891
Newcastle upon Tyne	152,311	28,278	51,836	23,913
North Tyneside	99,582	8,967	31,765	20,862
South Tyneside	72,655	7,068	21,824	15,915
Sunderland	135,725	14,988	42,355	28,306
Blackburn with Darwen	74,620	8,630	24,741	13,477
Blackpool	68,843	6,876	20,608	15,146
Cheshire East	187,022	16,154	53,221	41,064
Cheshire West and Che	165,728	16,492	48,530	34,866
Halton	62,653	6,132	19,454	12,510
Warrington	105,799	9,914	34,543	21,952
Cumbria	245,266	22,107	67,360	55,403
Bolton	142,757	15,525	45,599	26,684
Bury	93,594	9,080	29,671	18,387
Manchester	286,168	48,157	117,400	37,127
Oldham	117,052	13,146	37,709	21,116
Rochdale	108,181	11,308	34,951	20,266
Salford	130,583	14,391	49,342	22,177
Stockport	144,843	12,606	45,252	29,288
Tameside	110,844	10,704	35,099	22,955
Trafford	117,597	10,515	38,762	22,866

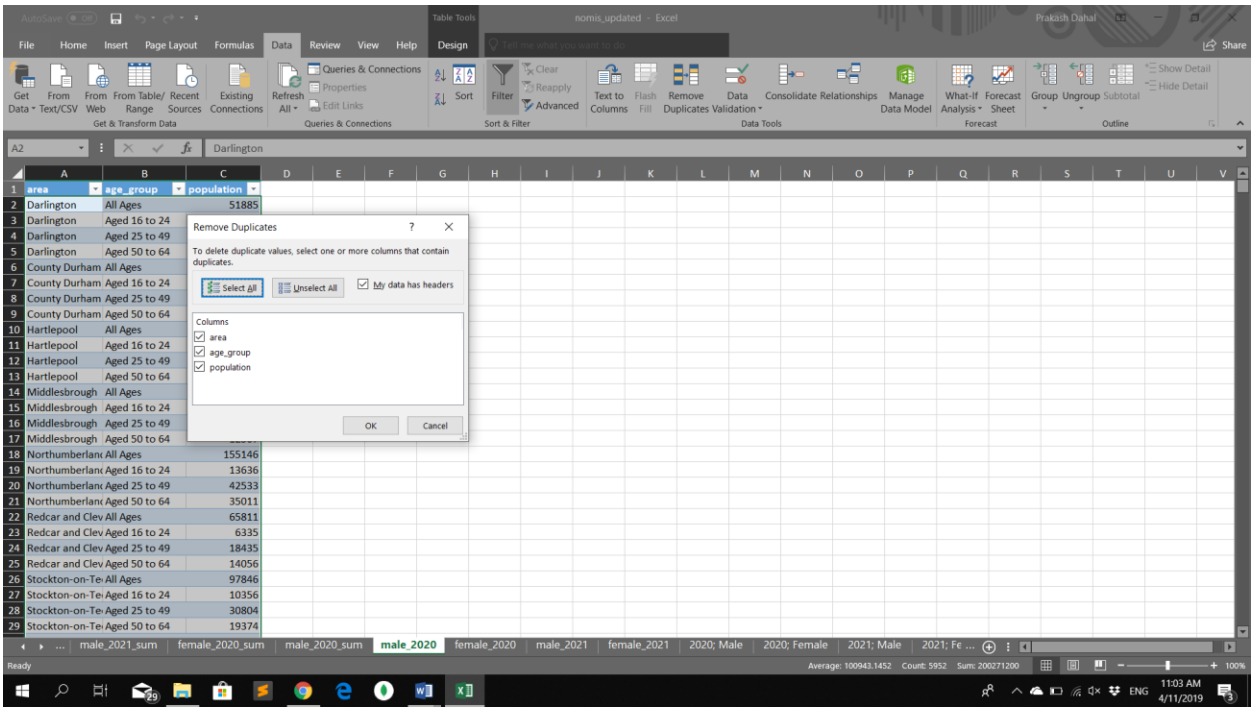

PivotTable and PivotChart Wizard - Step 2b of 3
Where are the worksheet ranges that you want to consolidate?
Range:
'2020; Male'!$A$1:$E$497
Add
Delete
Browse...
All ranges:
'2020; Male'!$A$1:$E$497
How many page fields do you want?
0
1
2
3
4
What item labels do you want each page field to use to identify the selected data range?
Field one:
Field two:
Field three:
Field four:
Cancel
< Back
Next >
Finish

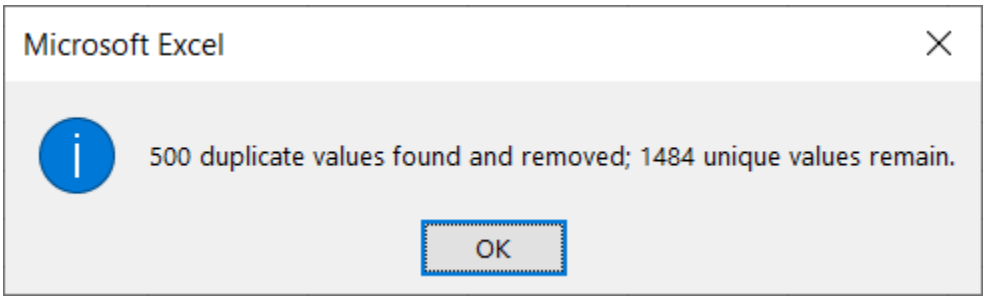

PivotTable and PivotChart Wizard - Step 3 of 3
Where do you want to put the PivotTable report?
New worksheet
Existing worksheet
Click Finish to create your PivotTable report.
Layout...
Options...
Cancel
< Back
Next >
Finish





After analyzing these data, redundant rows are cleaned again.



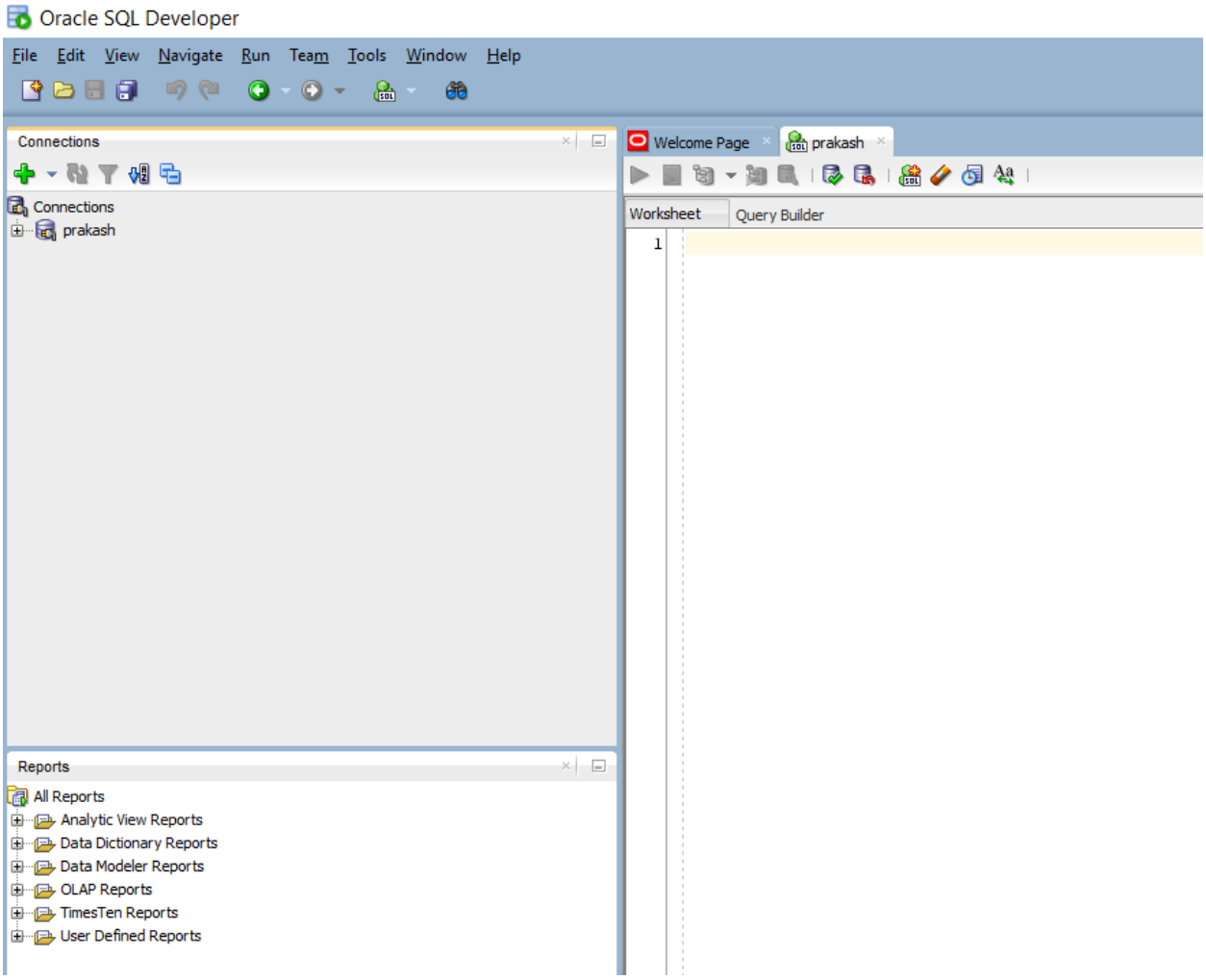


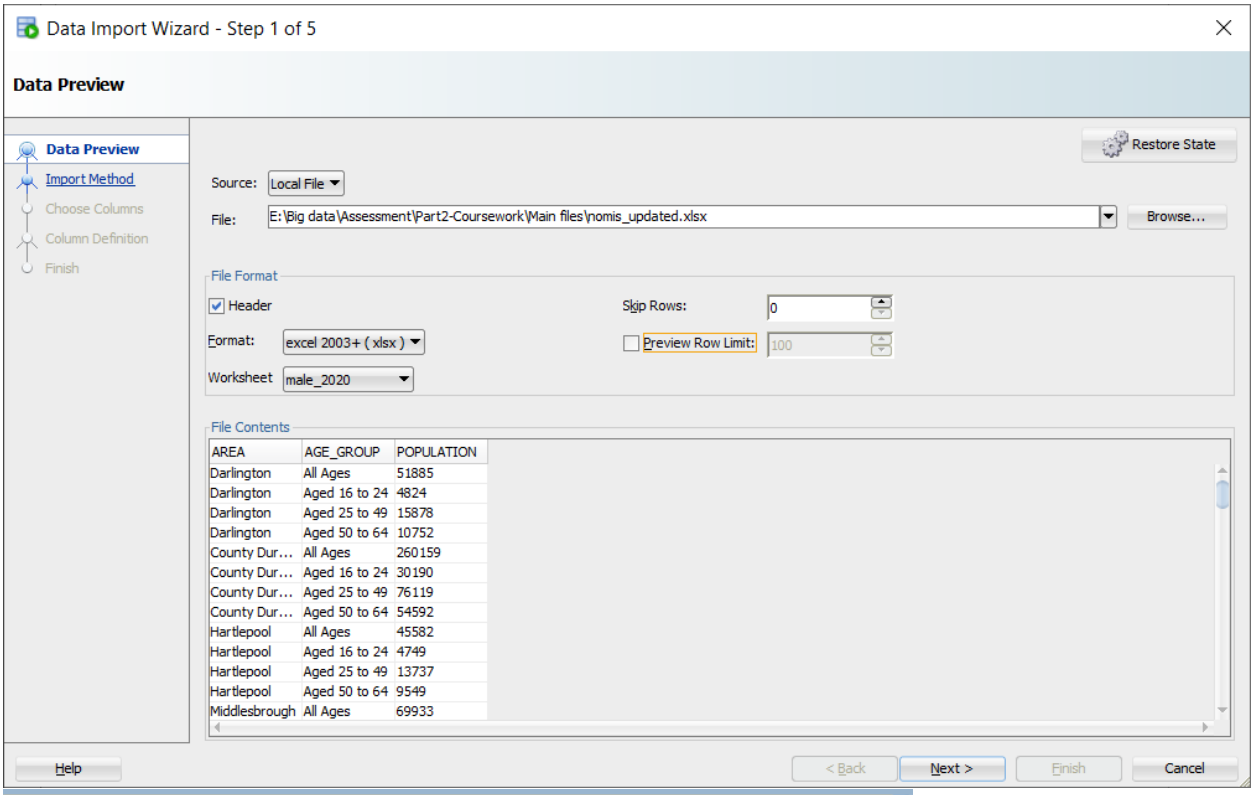
Similarly, other files are changed in and made pivot.

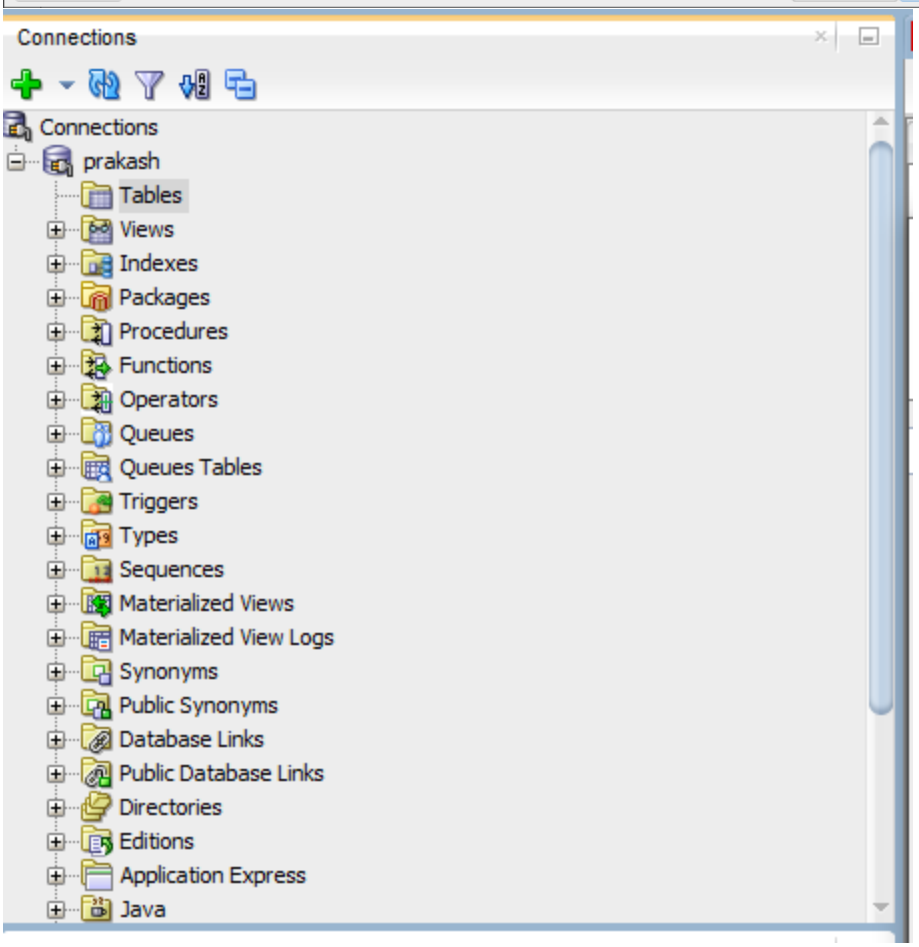
b. Manipulation:

Importing in Oracle:

Importing required data in **prakash** databases




Data Import Wizard - Step 1 of 5
Data Preview
Data Preview
Import Method
Choose Columns
Column Definition
Finish
Restore State
Source: Local File
File: E:\Big data\Assessment\Part2-Coursework\Main files\nomis_updated.xlsx
Browse...
File Format
Header
Skip Rows: 0
Format: excel 2003+ (xlsx)
Preview Row Limit: 100
Worksheet male_2020
File Contents
AREA AGE_GROUP POPULATION
Darlington All Ages 51885
Darlington Aged 16 to 24 4824
Darlington Aged 25 to 49 15878
Darlington Aged 50 to 64 10752
County Dur... All Ages 260159
County Dur... Aged 16 to 24 30190
County Dur... Aged 25 to 49 76119
County Dur... Aged 50 to 64 54592
Hartlepool All Ages 45582
Hartlepool Aged 16 to 24 4749
Hartlepool Aged 25 to 49 13737
Hartlepool Aged 50 to 64 9549
Middlesbrough All Ages 69933
Help
< Back
Next >
Finish
Cancel


Connections
Connections
prakash
Tables
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Application Express
Java

Data Import Wizard - Step 2 of 4

Import Method

Data Preview
Import Method
Column Definition
Finish

Specify the method for importing data. For External Table method, an external table will be created to read the data in the file. For Staging External Table method, an external table will be created as a staging table for importing the target table. For other methods, a new table is created and the data is imported.

Import Method: Insert

Send Create Script to SQL Worksheet

Table Name: male_2020

Import Row Limit: 100

File Contents

AREA	AGE_GROUP	POPULATION
Darlington	All Ages	51885
Darlington	Aged 16 to 24	4824
Darlington	Aged 25 to 49	15878
Darlington	Aged 50 to 64	10752
County Dur...	All Ages	260159
County Dur...	Aged 16 to 24	30190
County Dur...	Aged 25 to 49	76119
County Dur...	Aged 50 to 64	54592
Hartlepool	All Ages	45582
Hartlepool	Aged 16 to 24	4749
Hartlepool	Aged 25 to 49	13737
Hartlepool	Aged 50 to 64	9549
Middlesbrough	All Ages	69933
Middlesbrough	Aged 16 to 24	9765
Middlesbrough	Aged 25 to 49	21835
Middlesbrough	Aged 50 to 64	12507

Help < Back Next > Finish Cancel

Data Import Wizard - Step 3 of 5

Choose Columns

Data Preview
Import Method
Choose Columns
Column Definition
Finish

Select the columns to import from the data set and arrange them in the order you want.

Available Columns

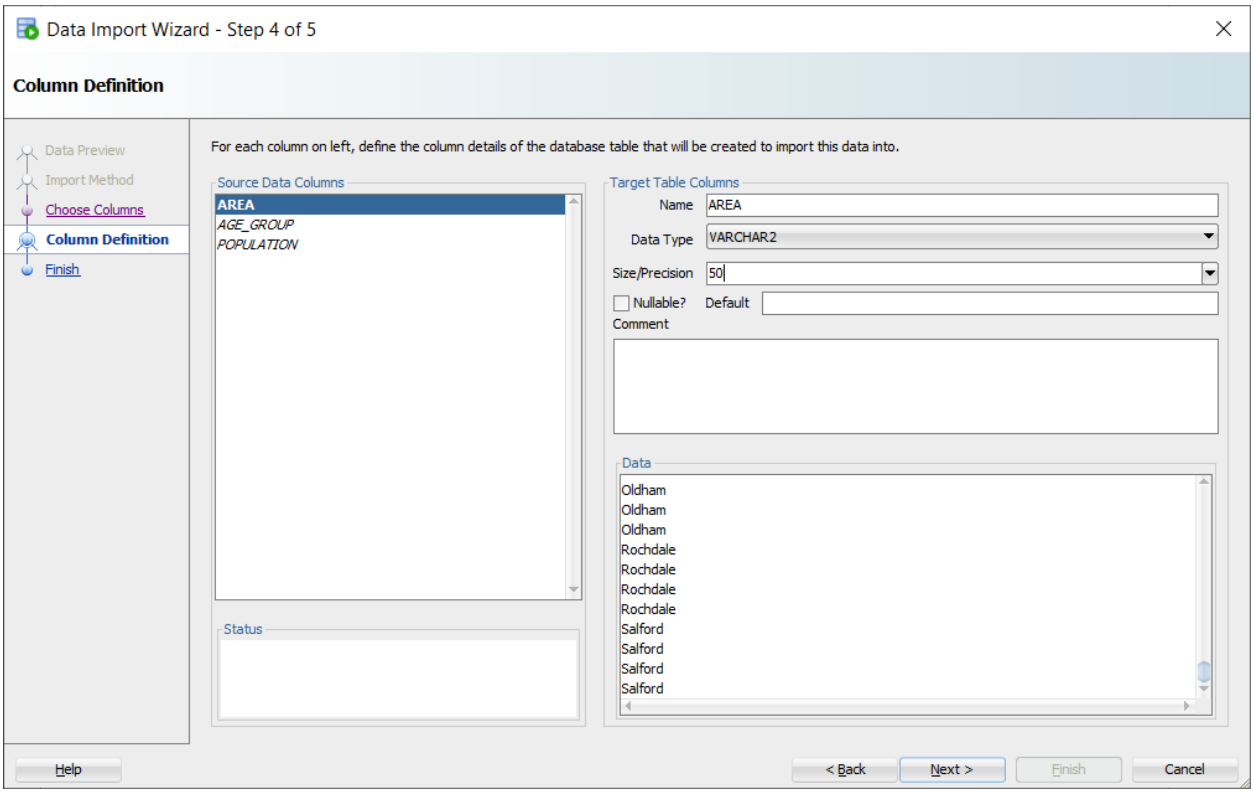
Selected Columns
AREA
AGE_GROUP
POPULATION

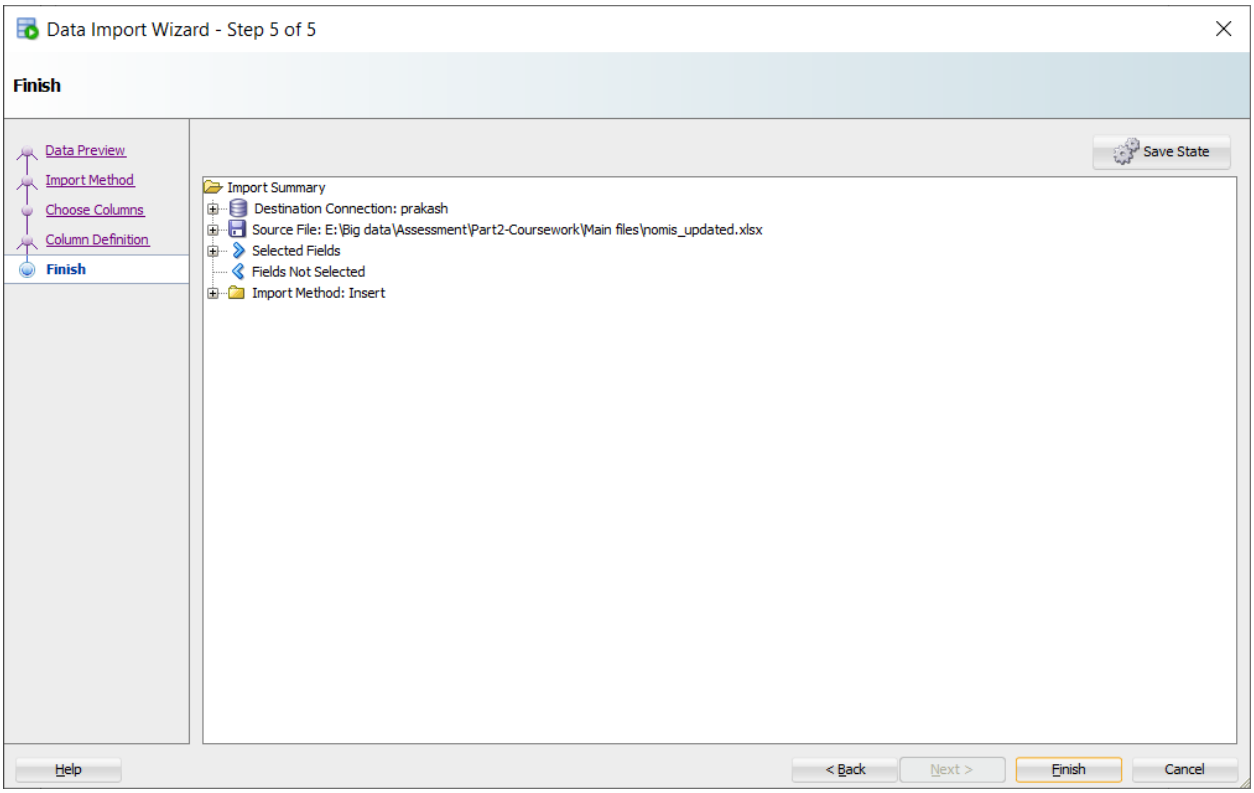
File Contents

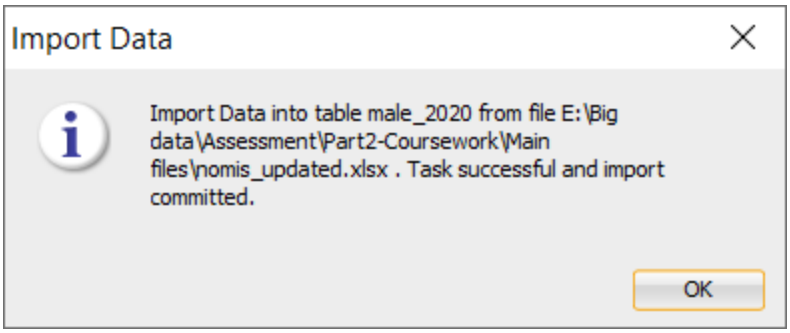
AREA	AGE_GROUP	POPULATION
Darlington	All Ages	51885
Darlington	Aged 16 to 24	4824
Darlington	Aged 25 to 49	15878

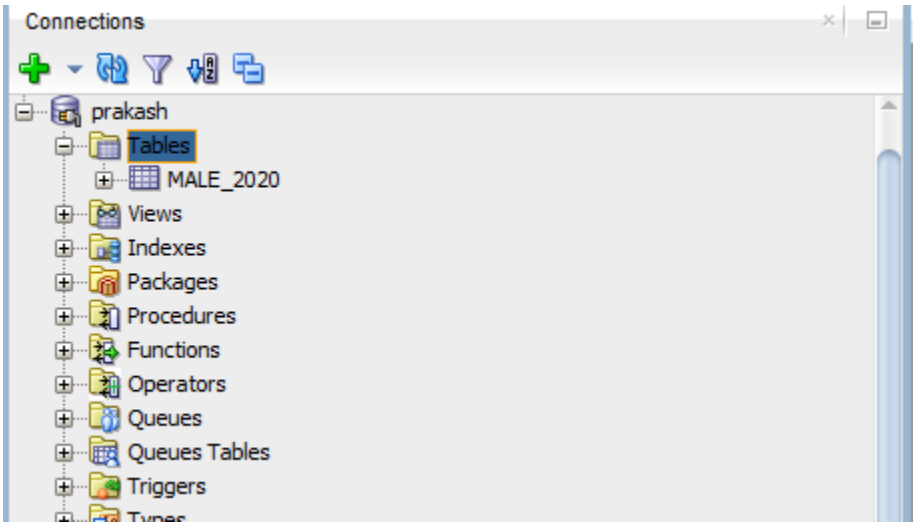
Help < Back Next > Finish Cancel

Changing size to 50 so that the area and age_group field could be longer and making nullable. But leaving population as it is.

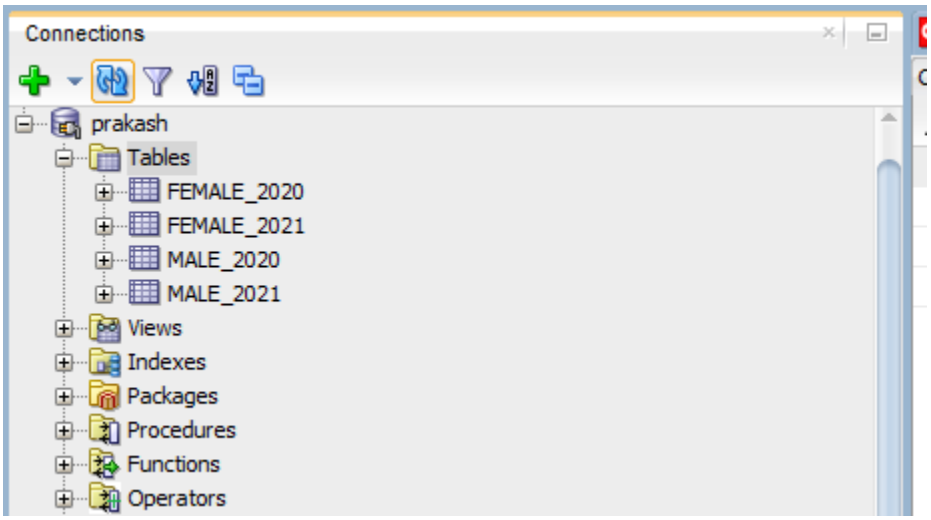




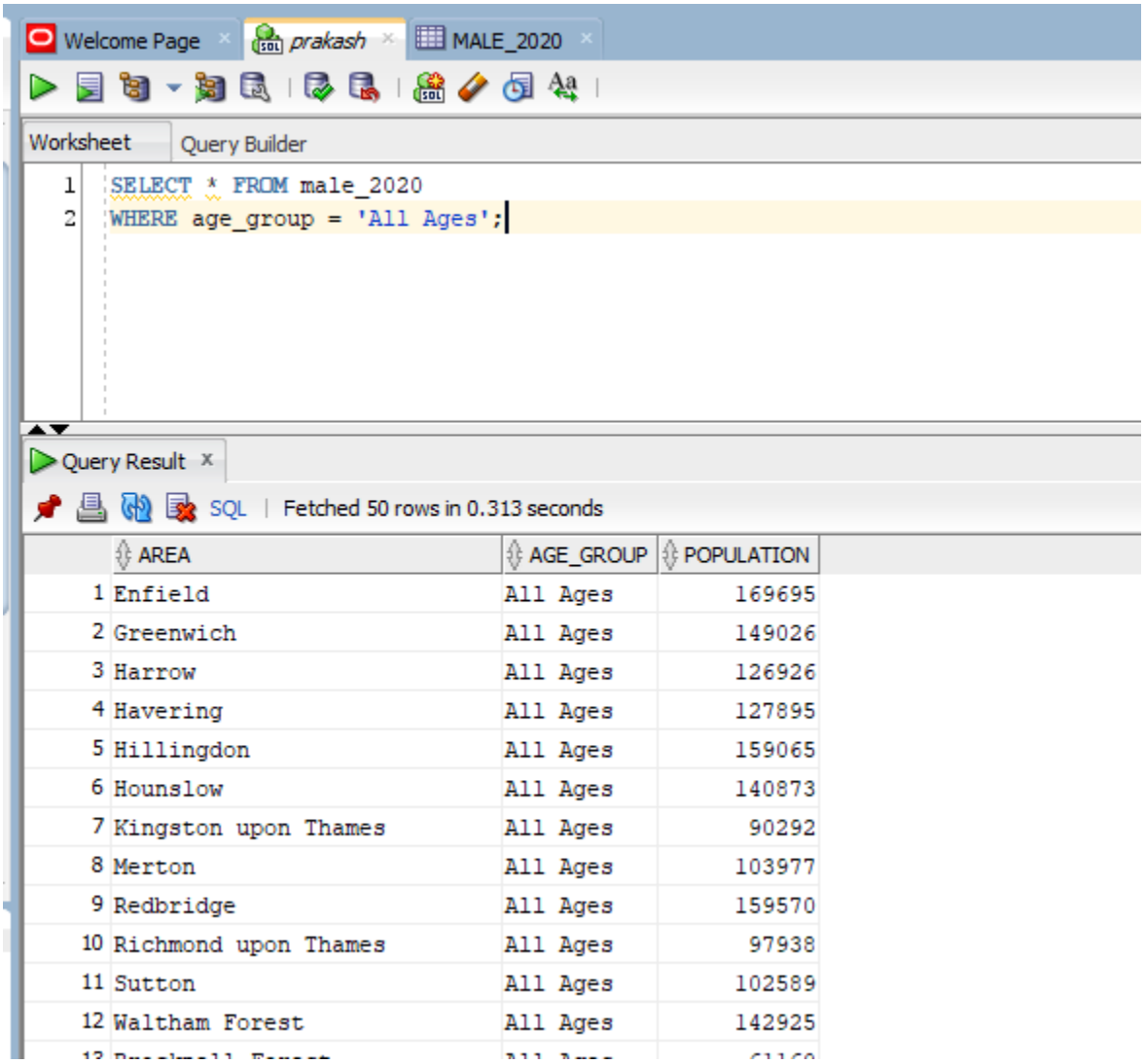




Similarly importing all required sheet.



SQL Query:



	AREA	AGE_GROUP	POPULATION
1	Enfield	All Ages	169695
2	Greenwich	All Ages	149026
3	Harrow	All Ages	126926
4	Havering	All Ages	127895
5	Hillingdon	All Ages	159065
6	Hounslow	All Ages	140873
7	Kingston upon Thames	All Ages	90292
8	Merton	All Ages	103977
9	Redbridge	All Ages	159570
10	Richmond upon Thames	All Ages	97938
11	Sutton	All Ages	102589
12	Waltham Forest	All Ages	142925

Query represents total age of each city of male2020.

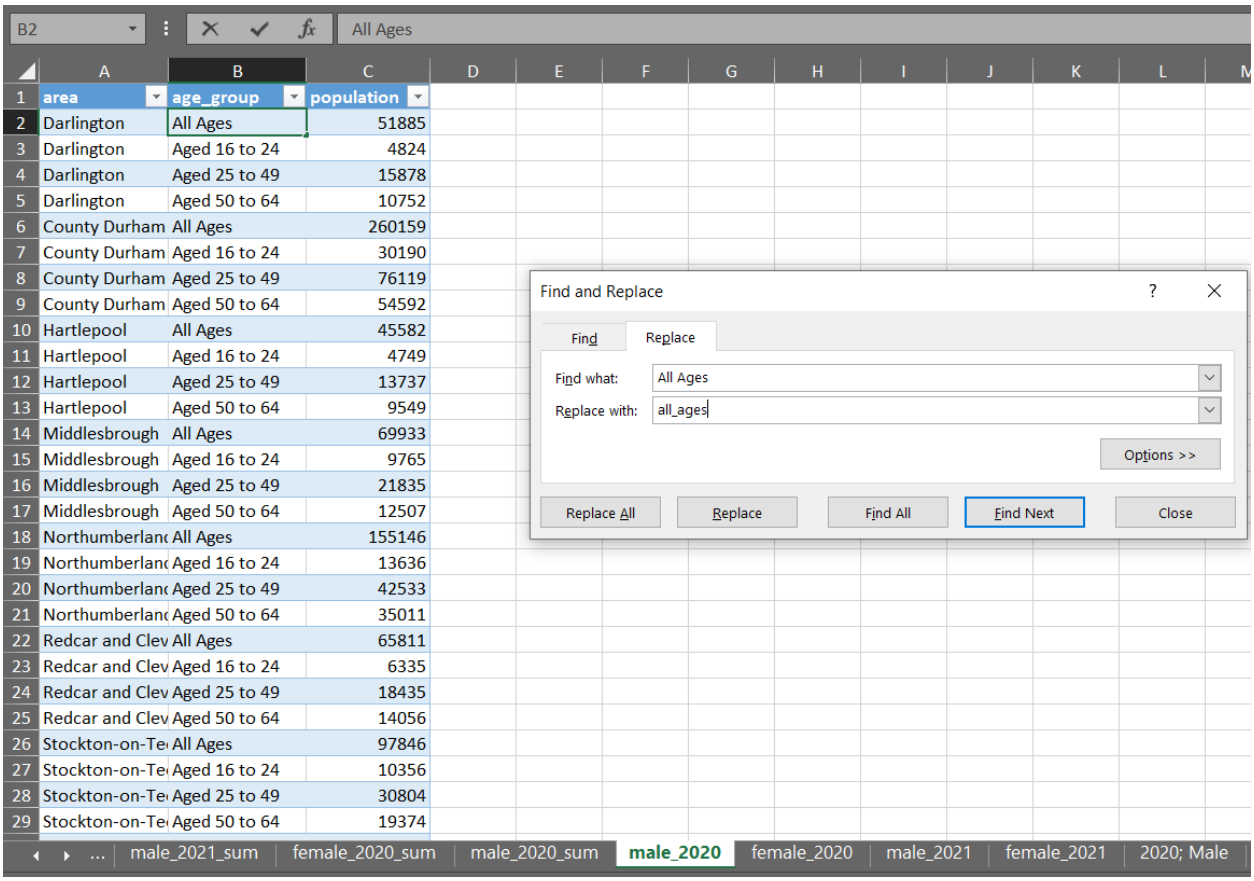
```sql
SELECT unique area AREA, f21.age_group AGE_2021,f21.population POPULATION_2021, f20.population POPULATION_2020
FROM female_2021 f21 JOIN  female_2020 f20
USING (area)
WHERE f20.population<1000 and f21.age_group = f20.age_group
ORDER BY area;
```

Script Output | Query Result

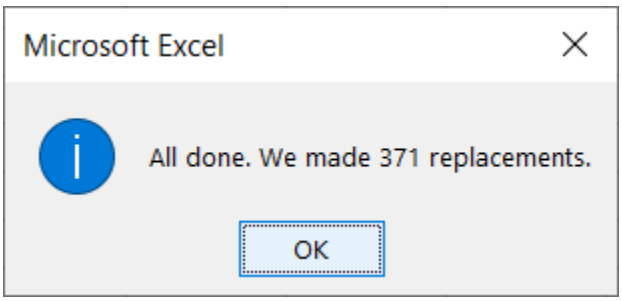
SQL | All Rows Fetched: 6 in 0.301 seconds

	AREA	AGE_2021	POPULATION_2021	POPULATION_2020
1	City of London	Aged 16 to 24	222	222
2	City of London	Aged 25 to 49	658	694
3	City of London	Aged 50 to 64	744	733
4	Isles of Scilly	Aged 16 to 24	89	93
5	Isles of Scilly	Aged 25 to 49	308	315
6	Isles of Scilly	Aged 50 to 64	222	220

While analyzing the cleaned pivot csv file, the name given is not suitable so changing the name for mongo DB and Hadoop.



area	age_group	population
Darlington	All Ages	51885
Darlington	Aged 16 to 24	4824
Darlington	Aged 25 to 49	15878
Darlington	Aged 50 to 64	10752
County Durham	All Ages	260159
County Durham	Aged 16 to 24	30190
County Durham	Aged 25 to 49	76119
County Durham	Aged 50 to 64	54592
Hartlepool	All Ages	45582
Hartlepool	Aged 16 to 24	4749
Hartlepool	Aged 25 to 49	13737
Hartlepool	Aged 50 to 64	9549
Middlesbrough	All Ages	69933
Middlesbrough	Aged 16 to 24	9765
Middlesbrough	Aged 25 to 49	21835
Middlesbrough	Aged 50 to 64	12507
Northumberlan	All Ages	155146
Northumberlan	Aged 16 to 24	13636
Northumberlan	Aged 25 to 49	42533
Northumberlan	Aged 50 to 64	35011
Redcar and Clev	All Ages	65811
Redcar and Clev	Aged 16 to 24	6335
Redcar and Clev	Aged 25 to 49	18435
Redcar and Clev	Aged 50 to 64	14056
Stockton-on-Te	All Ages	97846
Stockton-on-Te	Aged 16 to 24	10356
Stockton-on-Te	Aged 25 to 49	30804
Stockton-on-Te	Aged 50 to 64	19374

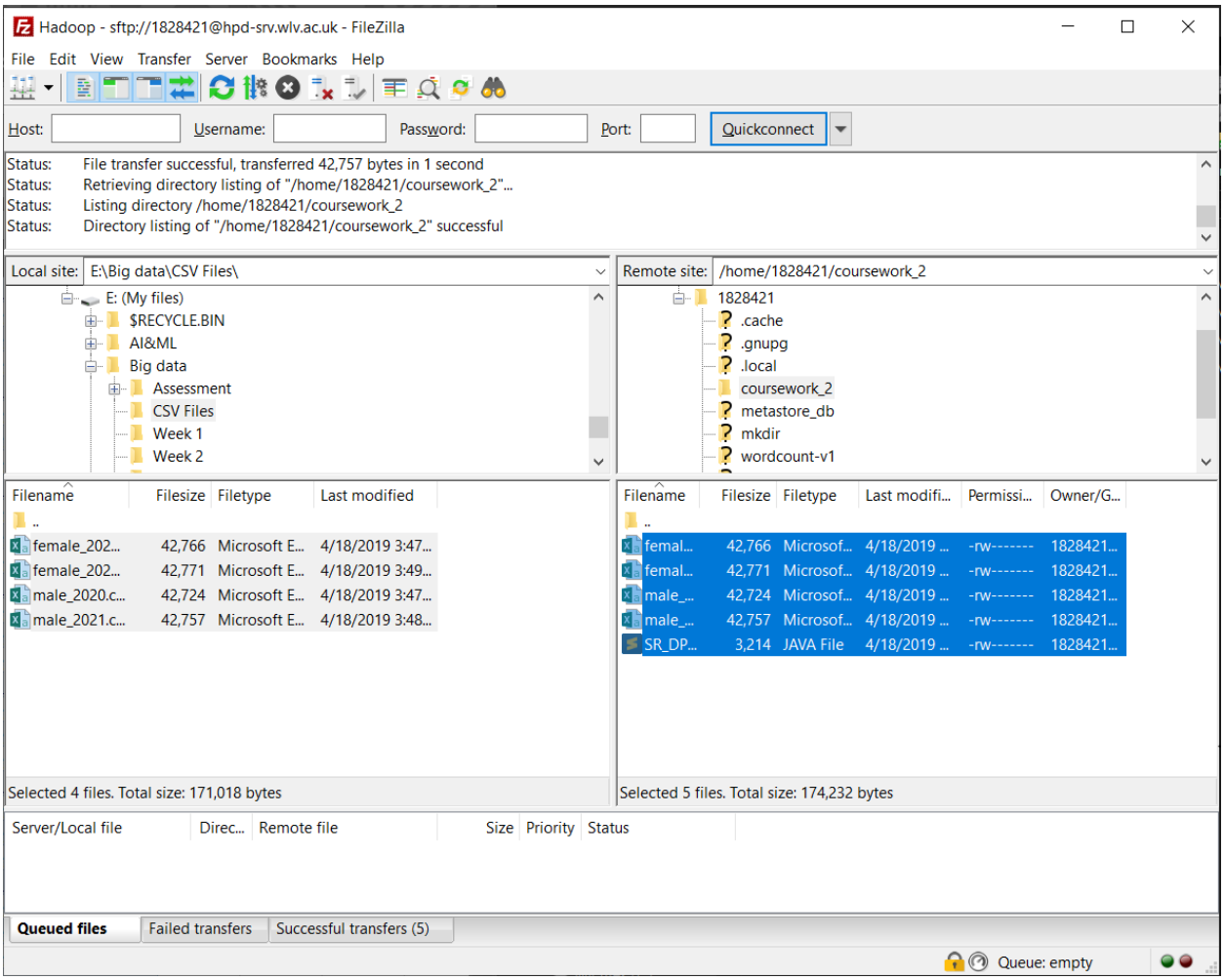


Similarly, giving suitable name for all on each sheet

B5 | age_50_64

	A	B	C	D
1	area	age_group	population	
2	Darlington	all_ages	51885	
3	Darlington	age_16_24	4824	
4	Darlington	age_25_49	15878	
5	Darlington	age_50_64	10752	
6	County Durham	all_ages	260159	
7	County Durham	age_16_24	30190	
8	County Durham	age_25_49	76119	
9	County Durham	age_50_64	54592	
10	Hartlepool	all_ages	45582	
11	Hartlepool	age_16_24	4749	
12	Hartlepool	age_25_49	13737	
13	Hartlepool	age_50_64	9549	
14	Middlesbrough	all_ages	69933	
15	Middlesbrough	age_16_24	9765	
16	Middlesbrough	age_25_49	21835	
17	Middlesbrough	age_50_64	12507	
18	Northumberlan	all_ages	155146	

Hadoop:



Making input -output files and putting file in it.

1828421@sm1: ~/coursework_2

```
1828421@sm1:~/coursework_2$ hdfs dfs -rm -R SR_DP_input
Deleted SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -rm -R SR_DP_output
Deleted SR_DP_output
1828421@sm1:~/coursework_2$ hdfs dfs -mkdir SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -mkdir SR_DP_output
1828421@sm1:~/coursework_2$ hdfs dfs -ls
Found 10 items
drwxr-xr-x   - 1828421 hadoop          0 2019-04-02 06:02 PD_input
drwxr-xr-x   - 1828421 hadoop          0 2019-04-02 06:05 PD_output
drwxr-xr-x   - 1828421 hadoop          0 2019-04-18 11:18 SR_DP_input
drwxr-xr-x   - 1828421 hadoop          0 2019-04-18 11:18 SR_DP_output
drwxr-xr-x   - 1828421 hadoop          0 2019-03-23 12:10 input
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 06:17 input_csv
drwxr-xr-x   - 1828421 hadoop          0 2019-03-23 12:46 input_word
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 06:35 output_csv
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 05:02 output_word
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 08:04 spark_output_word
1828421@sm1:~/coursework_2$
```

1828421@sm1: ~/coursework_2

```
drwxr-xr-x   - 1828421 hadoop          0 2019-04-18 11:18 SR_DP_input
drwxr-xr-x   - 1828421 hadoop          0 2019-04-18 11:18 SR_DP_output
drwxr-xr-x   - 1828421 hadoop          0 2019-03-23 12:10 input
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 06:17 input_csv
drwxr-xr-x   - 1828421 hadoop          0 2019-03-23 12:46 input_word
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 06:35 output_csv
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 05:02 output_word
drwxr-xr-x   - 1828421 hadoop          0 2019-03-29 08:04 spark_output_word
1828421@sm1:~/coursework_2$ hdfs dfs -put male_2020.csv SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -put male_2021.csv SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -put female_2021.csv SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -put female_2020.csv SR_DP_input
1828421@sm1:~/coursework_2$ hdfs dfs -ls SR_DP_input
Found 4 items
-rw-r--r--   1 1828421 hadoop      42766 2019-04-18 11:21 SR_DP_input/female_
2020.csv
-rw-r--r--   1 1828421 hadoop      42771 2019-04-18 11:21 SR_DP_input/female_
2021.csv
-rw-r--r--   1 1828421 hadoop      42724 2019-04-18 11:21 SR_DP_input/male_20
20.csv
-rw-r--r--   1 1828421 hadoop      42757 2019-04-18 11:21 SR_DP_input/male_20
21.csv
1828421@sm1:~/coursework_2$
```

Compiling Java file, making jar file and running.

```
1828421@sm1:~/coursework_2$ javac -classpath $(hadoop classpath) SR_DP.java
1828421@sm1:~/coursework_2$ jar cf SR_DP.jar SR*.class
1828421@sm1:~/coursework_2$ hadoop jar SR_DP.jar SR_DP SR_DP_input/male_2020.csv SR_DP_output
2019-04-18 11:30:40,253 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0
.0.1:8050
2019-04-18 11:30:40,707 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsin
g not performed. Implement the Tool interface and execute your application with ToolRunner to
 remedy this.
2019-04-18 11:30:40,718 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path
: /tmp/hadoop-yarn/staging/1828421/.staging/job_1553250033923_0929
2019-04-18 11:30:40,897 INFO input.FileInputFormat: Total input files to process : 1
2019-04-18 11:30:40,944 INFO mapreduce.JobSubmitter: number of splits:1
2019-04-18 11:30:41,062 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_155325003
3923_0929
2019-04-18 11:30:41,063 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-04-18 11:30:41,197 INFO conf.Configuration: resource-types.xml not found
2019-04-18 11:30:41,197 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-04-18 11:30:41,241 INFO impl.YarnClientImpl: Submitted application application_155325003
3923_0929
2019-04-18 11:30:41,267 INFO mapreduce.Job: The url to track the job: http://sm1:8088/proxy/a
pplication_1553250033923_0929/
2019-04-18 11:30:41,267 INFO mapreduce.Job: Running job: job_1553250033923_0929
2019-04-18 11:30:47,341 INFO mapreduce.Job: Job job_1553250033923_0929 running in uber mode :
 false
2019-04-18 11:30:47,343 INFO mapreduce.Job:  map 0% reduce 0%
2019-04-18 11:30:51,405 INFO mapreduce.Job:  map 100% reduce 0%
2019-04-18 11:30:56,440 INFO mapreduce.Job:  map 100% reduce 100%
2019-04-18 11:30:57,458 INFO mapreduce.Job: Job job_1553250033923_0929 completed successfully
2019-04-18 11:30:57,562 INFO mapreduce.Job: Counters: 53
        File System Counters
```

Result of male 2020

1828421@sm1: ~/coursework_2

```
1828421@sm1:~/coursework_2$ hdfs dfs -cat SR_DP_output/part-r-00000
"Bristol,4,403195.00
"Herefordshire,4,152596.00
"Kingston upon Hull,4,218420.00
Adur,4,50863.00
Allerdale,4,76561.00
Amber Valley,4,99831.00
Arun,4,121612.00
Ashfield,4,101664.00
Ashford,4,102274.00
Aylesbury Vale,4,164163.00
Babergh,4,70045.00
Barking and Dagenham,4,179256.00
Barnet,4,331472.00
Barnsley,4,199870.00
Barrow-in-Furness,4,52388.00
Basildon,4,150221.00
Basingstoke and Deane,4,144431.00
Bassetlaw,4,93087.00
Bath and North East Somerset,4,157015.00
Bedford,4,141972.00
Bexley,4,200884.00
Birmingham,4,947527.00
Blaby,4,79091.00
Blackburn with Darwen,4,121468.00
Blackpool,4,111473.00
Bolsover,4,64040.00
```

Mongo DB:

Loading CSV file:

```
1828421@csl-student: ~
1828421@csl-student:~$ mongoimport --db db1828421 --username 1828421 --password
1828421MDB --type CSV --headerline --file ./male_2020.csv --collection male_2020
2019-04-18T13:04:06.218+0100    connected to: localhost
2019-04-18T13:04:06.354+0100    imported 1483 documents
1828421@csl-student:~$ mongoimport --db db1828421 --username 1828421 --password
1828421MDB --type CSV --headerline --file ./male_2021.csv --collection male_2021
2019-04-18T13:04:34.117+0100    connected to: localhost
2019-04-18T13:04:34.285+0100    imported 1484 documents
1828421@csl-student:~$ mongoimport --db db1828421 --username 1828421 --password
1828421MDB --type CSV --headerline --file ./female_2021.csv --collection female_
2021
2019-04-18T13:04:47.273+0100    connected to: localhost
2019-04-18T13:04:47.431+0100    imported 1484 documents
1828421@csl-student:~$ mongoimport --db db1828421 --username 1828421 --password
1828421MDB --type CSV --headerline --file ./female_2020.csv --collection female_
2020
2019-04-18T13:04:57.568+0100    connected to: localhost
2019-04-18T13:04:57.701+0100    imported 1484 documents
```

```
1828421@csl-student: ~
> show collections
-bash: syntax error near unexpected token `show'
1828421@csl-student:~$ runMongo
Hello: 1828421
MongoDB shell version: 3.2.12
connecting to: 127.0.0.1:27017/db1828421
> show collection
2019-04-18T13:06:20.470+0100 E QUERY    [thread1] Error: don't know how to show
[collection] :
shellHelper.show@src/mongo/shell/utils.js:865:11
shellHelper@src/mongo/shell/utils.js:651:15
@(shellhelp2):1:1

> show collections
female_2020
female_2021
katpost
male_2020
male_2021
myCollection
new_katpost
student
weather
>
```

1828421@csl-student: ~

```
> db.male_2021.find({age_group:'all_ages'}).pretty().limit(5)
{
        "_id" : ObjectId("5cb867d2d01cf55993e74817"),
        "area" : "Darlington",
        "age_group" : "all_ages",
        "population" : 51886
}
{
        "_id" : ObjectId("5cb867d2d01cf55993e7481b"),
        "area" : "County Durham",
        "age_group" : "all_ages",
        "population" : 260956
}
{
        "_id" : ObjectId("5cb867d2d01cf55993e7481f"),
        "area" : "Hartlepool",
        "age_group" : "all_ages",
        "population" : 45624
}
{
        "_id" : ObjectId("5cb867d2d01cf55993e74823"),
        "area" : "Middlesbrough",
        "age_group" : "all_ages",
        "population" : 69971
}
{
        "_id" : ObjectId("5cb867d2d01cf55993e74827"),
        "area" : "Northumberland",
        "age_group" : "all_ages",
        "population" : 155234
}
>
```

Loading JSON:

```
1828421@csl-student: ~
1828421@csl-student:~$ ls
female_2020.csv  katpost.json   male_2021.csv  svn  uow.json
female_2021.csv  male_2020.csv  public_html    tmp  weather.json
1828421@csl-student:~$ [3~
```

Making collection of json uow

```
1828421@csl-student: ~
1828421@csl-student:~$ mongoimport --db db1828421 --username 1828421 --password
1828421MDB --type JSON --file ./uow.json --collection uow
2019-04-18T15:34:09.068+0100    connected to: localhost
2019-04-18T15:34:12.041+0100    [###############.........] db1828421.uow       9
.05MB/14.3MB (63.2%)
2019-04-18T15:34:14.984+0100    [########################] db1828421.uow       1
4.3MB/14.3MB (100.0%)
2019-04-18T15:34:14.984+0100    imported 3211 documents
1828421@csl-student:~$
```

```
1828421@csl-student: ~
1828421@csl-student:~$ runMongo
Hello: 1828421
MongoDB shell version: 3.2.12
connecting to: 127.0.0.1:27017/db1828421
> show collections
female_2020
female_2021
katpost
male_2020
male_2021
myCollection
new_katpost
student
uow
weather
>
```

Displaying first two uow data

```
1828421@csl-student: ~
student
uow
weather
> db.uow.find(2)
2019-04-18T15:37:37.408+0100 E QUERY    [thread1] Error: don't know how to massa
ge : number :
DBCollection.prototype._massageObject@src/mongo/shell/collection.js:218:11
DBCollection.prototype.find@src/mongo/shell/collection.js:266:1
@(shell):1:1

> db.uow.find().pretty().limit(2)
{
        "_id" : ObjectId("5c53295defc5690091f1182c"),
        "created_at" : "Thu Jan 31 15:42:29 +0000 2019",
        "id" : NumberLong("1090998762619711494"),
        "id_str" : "1090998762619711494",
        "text" : "The heating in the Performance Hub (WD), Walsall Campus, is no
w working. The building will remain shut today and re… https://t.co/0l9DtvbQmb",
        "truncated" : true,
        "entities" : {
                "hashtags" : [ ],
                "symbols" : [ ],
                "user_mentions" : [ ],
                "urls" : [
```

1828421@csl-student: ~

```
> db.uow.find({'lang':'en'},{'lang':1}).pretty().count()
3164
> db.uow.find({'lang':'en'},{'lang':1}).pretty()
{ "_id" : ObjectId("5c53295defc5690091f1182c"), "lang" : "en" }
{ "_id" : ObjectId("5c53295defc5690091f1182d"), "lang" : "en" }
{ "_id" : ObjectId("5c53295defc5690091f1182e"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f1182f"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11830"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11831"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11832"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11833"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11834"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11835"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11836"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11837"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11838"), "lang" : "en" }
{ "_id" : ObjectId("5c53295eefc5690091f11839"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183a"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183b"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183c"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183d"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183e"), "lang" : "en" }
{ "_id" : ObjectId("5c53295fefc5690091f1183f"), "lang" : "en" }
Type "it" for more
```

Py-Spark:

Loading csv and converting into view:

```
1828421@sm1: ~/coursework_2
1828421@sm1:~/coursework_2$ pyspark
Python 2.7.15+ (default, Oct  2 2018, 22:12:08)
[GCC 8.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0
      /_/

Using Python version 2.7.15+ (default, Oct  2 2018 22:12:08)
SparkSession available as 'spark'.
>>> Male2020 = spark.read.format("csv").load("hdfs://localhost:9000/user/1828421/SR_DP_input/
male_2020.csv")
>>> Male2020.createOrReplaceTempView("Male2020")
>>> Male2021 = spark.read.format("csv").load("hdfs://localhost:9000/user/1828421/SR_DP_input/
male_2021.csv")
>>> Male2021.createOrReplaceTempView("Male2021")
>>> Female2020 = spark.read.format("csv").load("hdfs://localhost:9000/user/1828421/SR_DP_inpu
t/female_2020.csv")
>>> Female2020.createOrReplaceTempView("Female2020")
>>> Female2021 = spark.read.format("csv").load("hdfs://localhost:9000/user/1828421/SR_DP_inpu
t/female_2021.csv")
>>> Female2021.createOrReplaceTempView("Female2021")
>>>
```

Joining two tables of male and female of 2020

1828421@sm1: ~/coursework_2

```
>>> df2020 = spark.sql("SELECT m.*,f._c2 no_of_female FROM Male2020 m, Female2020 f WHERE m._
c0=f._c0").show(20, False)
+-------------+---------+------+------------+
|_c0          |_c1      |_c2   |no_of_female|
+-------------+---------+------+------------+
|Darlington   |all_ages |51885 |11400       |
|Darlington   |all_ages |51885 |16845       |
|Darlington   |all_ages |51885 |4687        |
|Darlington   |all_ages |51885 |54643       |
|Darlington   |age_16_24|4824  |11400       |
|Darlington   |age_16_24|4824  |16845       |
|Darlington   |age_16_24|4824  |4687        |
|Darlington   |age_16_24|4824  |54643       |
|Darlington   |age_25_49|15878 |11400       |
|Darlington   |age_25_49|15878 |16845       |
|Darlington   |age_25_49|15878 |4687        |
|Darlington   |age_25_49|15878 |54643       |
|Darlington   |age_50_64|10752 |11400       |
|Darlington   |age_50_64|10752 |16845       |
|Darlington   |age_50_64|10752 |4687        |
|Darlington   |age_50_64|10752 |54643       |
|County Durham|all_ages |260159|56984       |
|County Durham|all_ages |260159|78448       |
|County Durham|all_ages |260159|28735       |
|County Durham|all_ages |260159|268344      |
+-------------+---------+------+------------+
only showing top 20 rows
```

Female of 2021

```
1828421@sm1: ~/coursework_2
>>> df2021 = spark.sql("SELECT f.* FROM Female2021 f WHERE f._c2<5000").show(20, False)
+-----------------+---------+----+
|_c0              |_c1      |_c2 |
+-----------------+---------+----+
|Darlington       |age_16_24|4590|
|Hartlepool       |age_16_24|4383|
|Rutland          |age_16_24|1373|
|Rutland          |age_25_49|4774|
|Rutland          |age_50_64|4252|
|City of London   |all_ages |2960|
|City of London   |age_16_24|222 |
|City of London   |age_25_49|658 |
|City of London   |age_50_64|744 |
|Isles of Scilly  |all_ages |1090|
|Isles of Scilly  |age_16_24|89  |
|Isles of Scilly  |age_25_49|308 |
|Isles of Scilly  |age_50_64|222 |
|Allerdale        |age_16_24|3722|
|Barrow-in-Furness|age_16_24|2845|
|Copeland         |age_16_24|2592|
|Eden             |age_16_24|1865|
|South Lakeland   |age_16_24|3488|
|Burnley          |age_16_24|3976|
|Chorley          |age_16_24|4727|
+-----------------+---------+----+
only showing top 20 rows
```

JSON with py-spark

```
1828421@sm1: ~

      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.0
      /_/

Using Python version 2.7.15+ (default, Oct  2 2018 22:12:08)
SparkSession available as 'spark'.
>>> df = spark.read.json("uow.json")
19/04/18 17:35:03 WARN Utils: Truncated the string representation of a plan sinc
e it was too large. This behavior can be adjusted by setting 'spark.debug.maxToS
tringFields' in SparkEnv.conf.
>>> df.show()
+--------------------+------------+-----------+--------------------+------------
--------+--------------------+--------------+---------+----+--------------------
+--------------------+-----------------------+---------------------+-------------
------------+--------------------+-----------------------+---------------+----+-
----+------------------+--------------------+--------------------+--------------
------+-------------+---------+--------------------+--------------------+-------
-------------+---------+--------------------+
|                 _id|contributors|coordinates|          created_at|
entities|   extended_entities|favorite_count|favorited| geo|                  id
|             id_str|in_reply_to_screen_name|in_reply_to_status_id|in_reply_to_s
tatus_id_str| in_reply_to_user_id|in_reply_to_user_id_str|is_quote_status|lang|p
```

1828421@sm1: ~

```
>>> df.select("text","lang").show()
+--------------------+----+
|                text|lang|
+--------------------+----+
|The heating in th...|  en|
|We're delighted t...|  en|
|BTW... The Univer...|  en|
|RT @wlv_alumni: F...|  en|
|RT @wlv_teaching:...|  en|
|@holyhead6thform ...|  en|
|STUDENTS - The Li...|  en|
|STUDENTS/STAFF - ...|  en|
|@bluenoze63 We'd ...|  en|
|RT @WLV_Arts: Joi...|  en|
|We're all set to ...|  en|
|A Computer Scienc...|  en|
|STUDENTS : The Na...|  en|
|RT @BCCCmembers: ...|  en|
|@fungaindem @geth...|  en|
|We're holding a f...|  en|
|Congratulations t...|  en|
|RT @WolveUniCop: ...|  en|
|RT @Jamesd1307: F...|  en|
|RT @stonesthrowlt...|  en|
```

1828421@sm1: ~

```
>>> df.select("text","lang").where(df.lang!="en").where(df.lang!="und").show()
+--------------------+----+
|                text|lang|
+--------------------+----+
|RT @LisaClaireBB:...|  no|
|RT @WlvWomenFC: F...|  de|
|@toridancingdrew ...|  ro|
|RT @FineArtWSA: B...|  de|
|RT @UoW_Catering:...|  da|
|Lá Fhéile Pádraig...|  cs|
|RT @CGI_Bghm: Dr....|  ro|
|RT @Laura_Ford97:...|  fr|
+--------------------+----+

>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

c. Analysis of the data:

Data analysis is most important part to understand about the data and extract information out of it. Two different years is selected 2020 and 2021. From these two years, male and female are selected so that the comparison of male of both years can be analyzed. So, with the female. Total population can be extracted adding total population of male and female.

Oracle Query displays the result of all aged group population and selecting female of 2020 and 2021 and joining by area and only displaying which population is less than 1000.

```
SELECT unique area AREA, f21.age_group AGE_2021,f21.population POPULATION_2021, f20.population POPULATION_2020
FROM female_2021 f21 JOIN  female_2020 f20
USING (area)
WHERE f20.population<1000 and f21.age_group = f20.age_group
ORDER BY area;
```

Script Output | Query Result

All Rows Fetched: 6 in 0.301 seconds

	AREA	AGE_2021	POPULATION_2021	POPULATION_2020
1	City of London	Aged 16 to 24	222	222
2	City of London	Aged 25 to 49	658	694
3	City of London	Aged 50 to 64	744	733
4	Isles of Scilly	Aged 16 to 24	89	93
5	Isles of Scilly	Aged 25 to 49	308	315
6	Isles of Scilly	Aged 50 to 64	222	220

In Map-Reduce, population is added as per the area and gives number of area and total population of the respective area.

1828421@sm1: ~/coursework_2

```
1828421@sm1:~/coursework_2$ hdfs dfs -cat SR_DP_output/part-r-00000
"Bristol,4,403195.00
"Herefordshire,4,152596.00
"Kingston upon Hull,4,218420.00
Adur,4,50863.00
Allerdale,4,76561.00
Amber Valley,4,99831.00
Arun,4,121612.00
Ashfield,4,101664.00
Ashford,4,102274.00
Aylesbury Vale,4,164163.00
Babergh,4,70045.00
```

Loading and displaying result from json file in mongo and pyspark

```
1828421@csl-student: ~
student
uow
weather
> db.uow.find(2)
2019-04-18T15:37:37.408+0100 E QUERY    [thread1] Error: don't know how to massa
ge : number :
DBCollection.prototype._massageObject@src/mongo/shell/collection.js:218:11
DBCollection.prototype.find@src/mongo/shell/collection.js:266:1
@(shell):1:1
```

In spark, also displaying text and language except English and undefined.

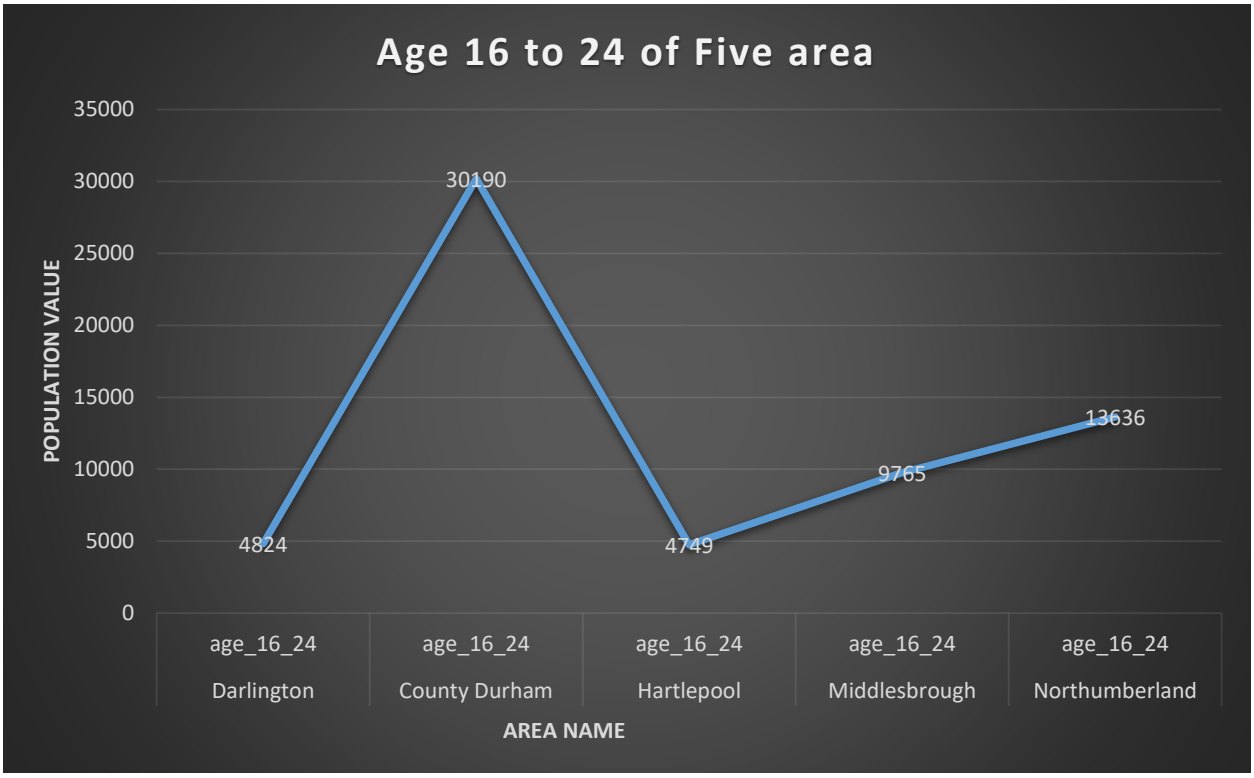
```
1828421@sm1: ~

>>> df.select("text","lang").where(df.lang!="en").where(df.lang!="und").show()
+--------------------+----+
|                text|lang|
+--------------------+----+
|RT @LisaClaireBB:...|  no|
|RT @WlvWomenFC: F...|  de|
|@toridancingdrew ...|  ro|
|RT @FineArtWSA: B...|  de|
|RT @UoW_Catering:...|  da|
|Lá Fhéile Pádraig...|  cs|
|RT @CGI_Bghm: Dr....|  ro|
|RT @Laura_Ford97:...|  fr|
+--------------------+----+
```

Similarly, CSV file is loaded and manipulated in both mongo db. and py spark. Instead of header option, default column is used. It displays, male2020 all col and female total population.

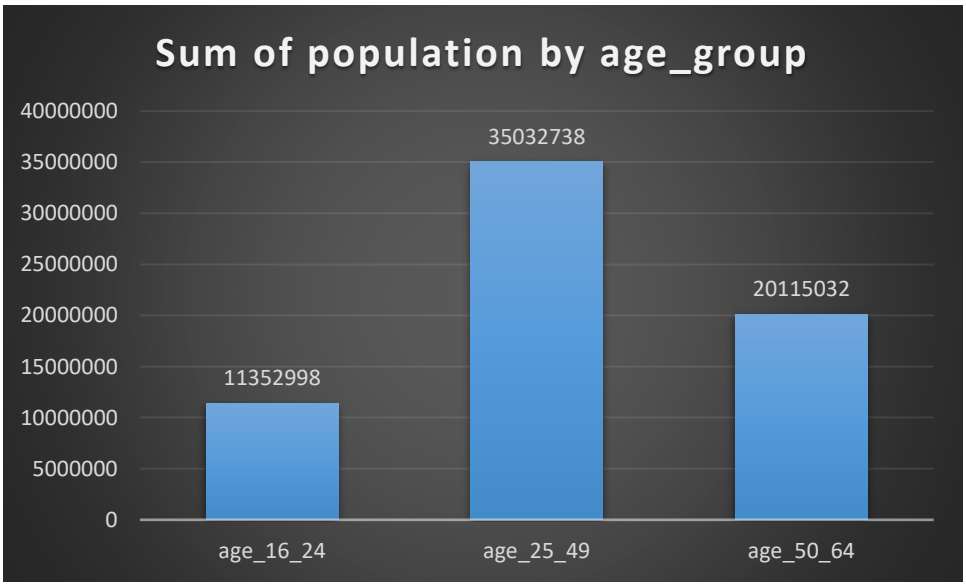
```
1828421@sm1: ~/coursework_2

>>> df2020 = spark.sql("SELECT m.*,f._c2 no_of_female FROM Male2020 m, Female2020 f WHERE m._
c0=f._c0").show(20, False)
+-------------+---------+------+------------+
|_c0          |_c1      |_c2   |no_of_female|
+-------------+---------+------+------------+
|Darlington   |all_ages |51885 |11400       |
|Darlington   |all_ages |51885 |16845       |
|Darlington   |all_ages |51885 |4687        |
|Darlington   |all_ages |51885 |54643       |
|Darlington   |age_16_24|4824  |11400       |
|Darlington   |age_16_24|4824  |16845       |
|Darlington   |age_16_24|4824  |4687        |
|Darlington   |age_16_24|4824  |54643       |
|Darlington   |age_25_49|15878 |11400       |
|Darlington   |age_25_49|15878 |16845       |
|Darlington   |age_25_49|15878 |4687        |
|Darlington   |age_25_49|15878 |54643       |
|Darlington   |age_50_64|10752 |11400       |
|Darlington   |age_50_64|10752 |16845       |
```

d. Data Visualization:

Taking five different cities of male 2020 and displaying graph of age 16 to 24. Conty Durham has more population than usual.



Adding all male2020 population and analyzing age group, age from 25 to 49 has more population.



C. Contribution

For this report, both of us worked together discussing on different topic and ideas. Both of us selected different research papers, sites and discussed to complete this report. Practical is important in real world so practical was done individually except oracle. Though everything was discussed, some of the contribution is mentioned below:

Name	Prakash Dahal	Rajan Sapkota (Sharma)
Report	Different references were discussed and finally, introduction, evaluation, matrix and conclusion were written with the agreement of both.	
Investigation: Oracle	Data cleaning, Pivot tables	Importing and SQL Query
Investigation: Mongo DB	Individual	Individual
Investigation: Hadoop	Individual	Individual

References

Chaokui Li, W. Y., 2014. *The distributed storage strategy research of remote sensing image based on Mongo DB.* Changsha, IEEE.

Chitresh Verma, R. P., 2016. *Cloud System and Big Data Engineering (Confluence).* Noida, India, IEEE.

Cyran, M., 2005. *Oracle Database Concepts,* s.l.: Oracle.

Gurjit singh Bhathal, A. S. D., 2018. *Big Data Solution: Improvised Distributions Framework of Hadoop.* Madurai, IEEE.

Huadong Dai, S. Z. L. W. Y. D., 2016. *Research and implementation of big data preprocessing system based on Hadoop.* Hangzhou, IEEE.

M. Sowmya, N. S., 2017. Big Data: An Overview of Features, Tools, Techniques and Applications. *International Journal of Engineering Science and Computing,* pp. 1364-13647.

Prabagaren, G., 2014. *Systematic approach for validating Java-MongoDB Schema.* Chennai, IEEE.

Richard L. Villars, D. V., 2014. *Building a Datacenter Infrastructure to Support Your,* s.l.: IDC.

Rick Greenwald, D. C. K., 2003. *Oracle in a Nutshell: A Desktop Quick Reference.* Sebastopol: O'Reilly Media, Inc.

Ruchi Bhardwaj, N. M. R. K., 2014. *Data analyzing using Map-Join-Reduce in cloud storage.* Solan, IEEE.

Sriramoju, S. B., 2017. *INTRODUCTION TO BIG DATA: INFRASTRUCTURE AND NETWORKING CONSIDERATIONS.* Warangal, India: Horizion Books.