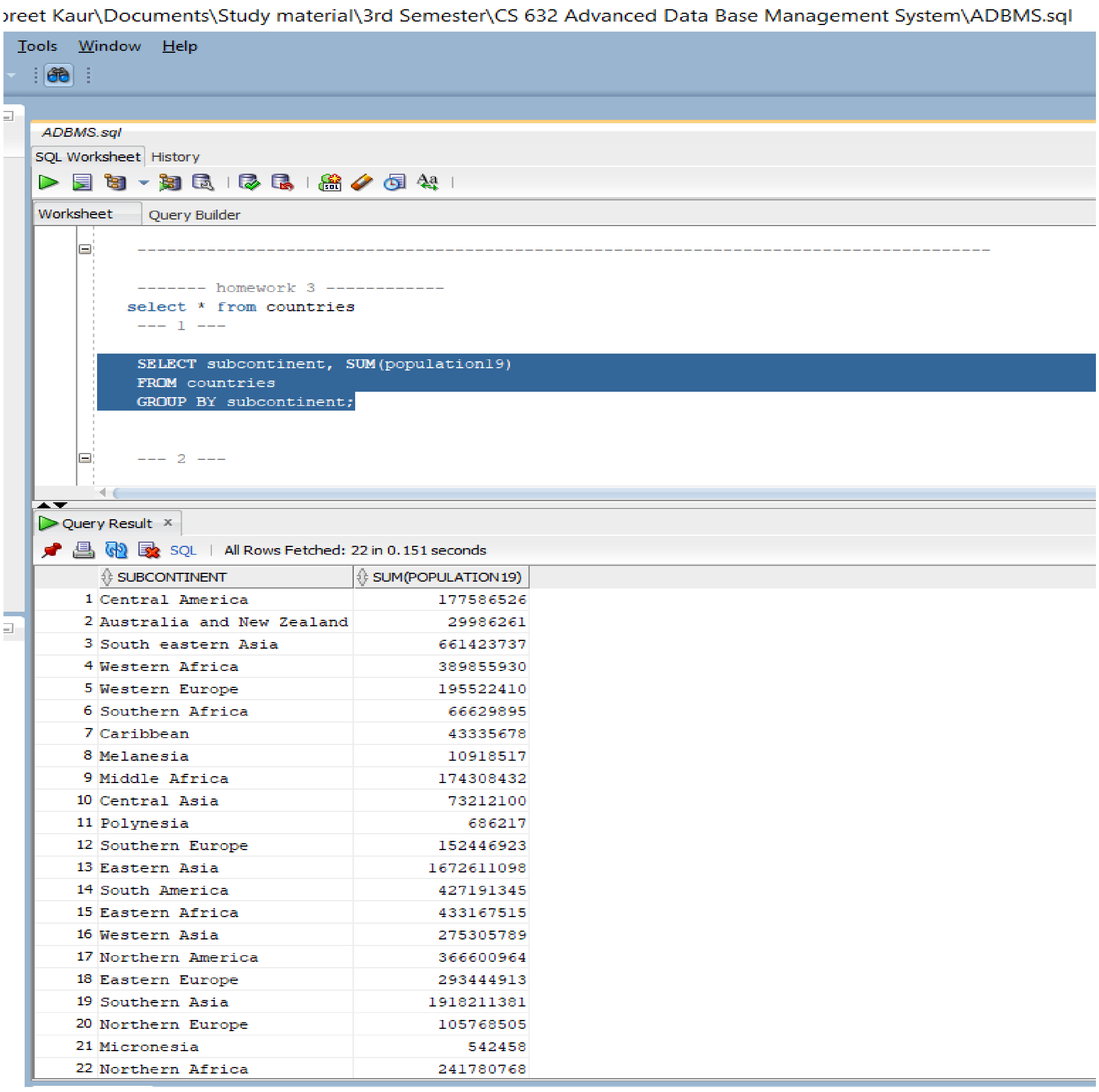
Homework 3

Question 1) Write an SQL Select statement against the table COUNTRIES that will display one line for each “UN statistical region” (e.g., Eastern Asia, South America) and the total population (sum) of all the countries in that region.

Use the 2019 population data. Hint: use group by. So, there will be only two columns in this result.

Answer 1)

SELECT subcontinent, SUM (population19)
FROM countries
GROUP BY subcontinent;



ɔreet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

```
  ------------------------------------------------------------------------------------

  ------- homework 3 ------------
select * from countries
  --- 1 ---

  SELECT subcontinent, SUM(population19)
  FROM countries
  GROUP BY subcontinent;


  --- 2 ---
```

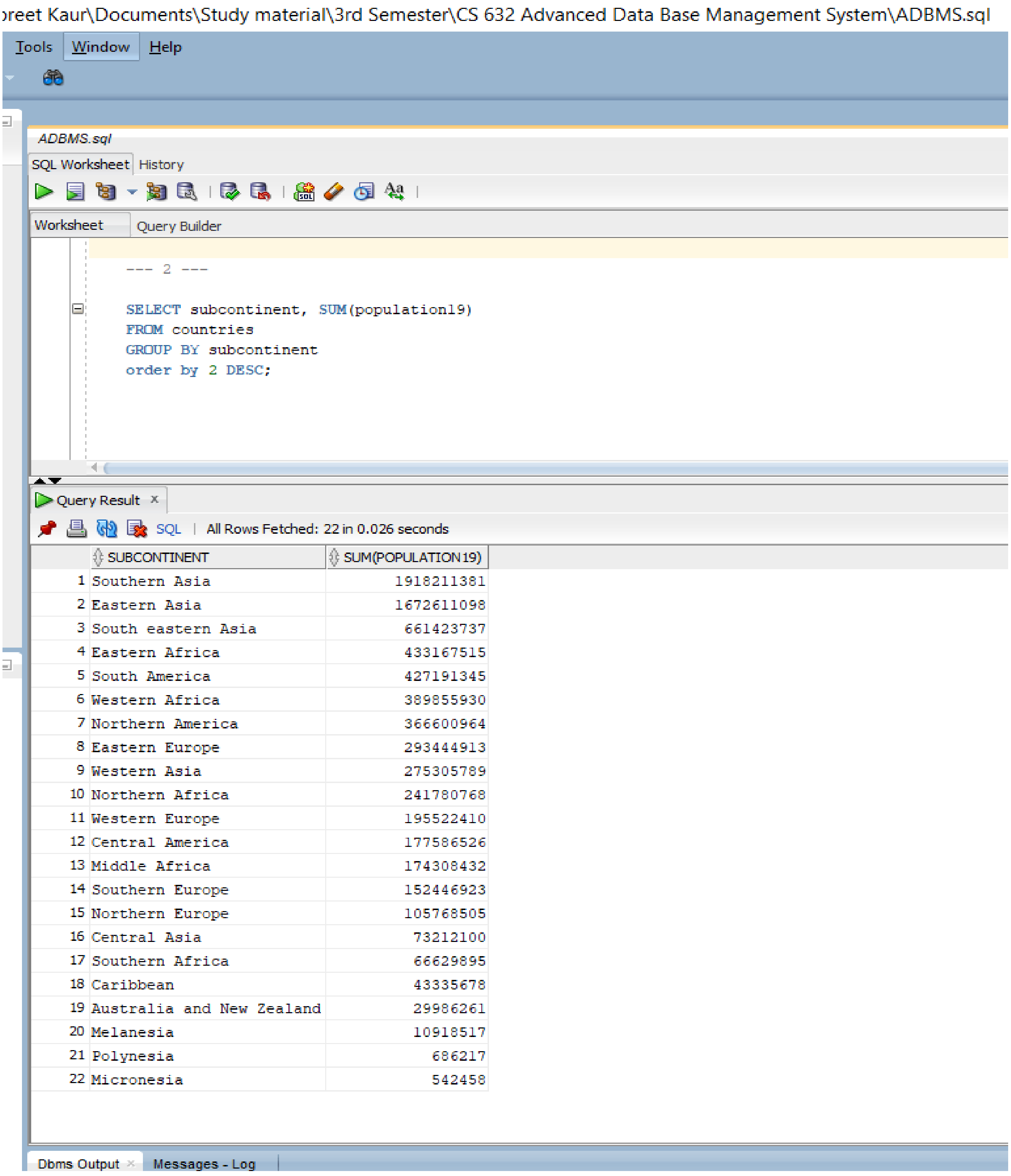
All Rows Fetched: 22 in 0.151 seconds

	SUBCONTINENT	SUM(POPULATION19)
1	Central America	177586526
2	Australia and New Zealand	29986261
3	South eastern Asia	661423737
4	Western Africa	389855930
5	Western Europe	195522410
6	Southern Africa	66629895
7	Caribbean	43335678
8	Melanesia	10918517
9	Middle Africa	174308432
10	Central Asia	73212100
11	Polynesia	686217
12	Southern Europe	152446923
13	Eastern Asia	1672611098
14	South America	427191345
15	Eastern Africa	433167515
16	Western Asia	275305789
17	Northern America	366600964
18	Eastern Europe	293444913
19	Southern Asia	1918211381
20	Northern Europe	105768505
21	Micronesia	542458
22	Northern Africa	241780768

Question 2) Extend question 1) to show the result in descending order by total population.

Answer 2)

SELECT subcontinent, SUM (population19)
FROM countries
GROUP BY subcontinent
order by 2 DESC;



```
--- 2 ---

SELECT subcontinent, SUM(population19)
FROM countries
GROUP BY subcontinent
order by 2 DESC;
```

All Rows Fetched: 22 in 0.026 seconds

	SUBCONTINENT	SUM(POPULATION19)
1	Southern Asia	1918211381
2	Eastern Asia	1672611098
3	South eastern Asia	661423737
4	Eastern Africa	433167515
5	South America	427191345
6	Western Africa	389855930
7	Northern America	366600964
8	Eastern Europe	293444913
9	Western Asia	275305789
10	Northern Africa	241780768
11	Western Europe	195522410
12	Central America	177586526
13	Middle Africa	174308432
14	Southern Europe	152446923
15	Northern Europe	105768505
16	Central Asia	73212100
17	Southern Africa	66629895
18	Caribbean	43335678
19	Australia and New Zealand	29986261
20	Melanesia	10918517
21	Polynesia	686217
22	Micronesia	542458

Question 3) Create a new table TEMPERATURES based on the web page https://en.wikipedia.org/wiki/List_of_countries_by_average_yearly_temperature

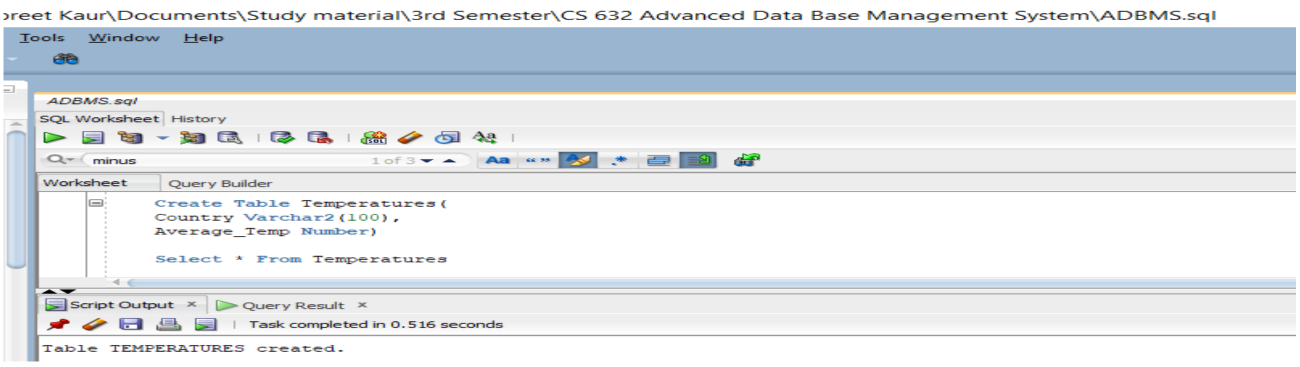
I see 191 countries here.

Do as much data cleaning as you can, so that the countries in this table appear the same as in the table COUNTRIES.

Answer 3)

Create Table Temperatures(
Country Varchar2(100),
Average_Temp Number)

Select * From Temperatures

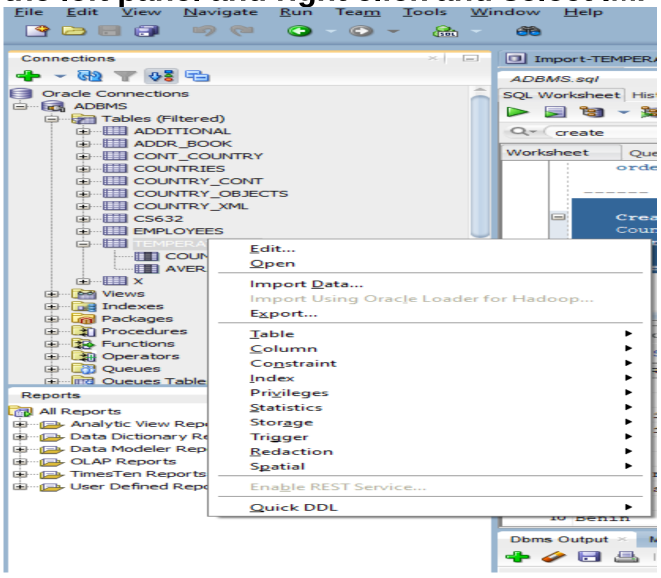


```
Create Table Temperatures(
Country Varchar2(100),
Average_Temp Number)

Select * From Temperatures
```

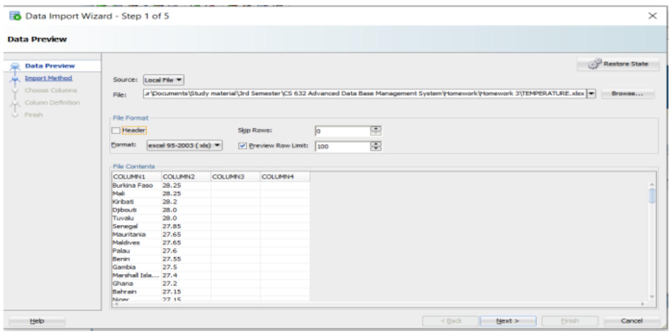
```
Task completed in 0.516 seconds

Table TEMPERATURES created.
```

Procedure :

Step 1) Copied the data from the given web page
Step 2) Pasted the data to new excel sheet
Step 3) Since the data contains the pictures and spaces in front of them. Therefore, cleaned it using trim method. To use trim, I inserted a new column and then used command =TRIM (clean(A1)) and then copied to all the 191 rows. That's how I cleaned my data.
Step 4) After that I saved the file with the name Temperatures.
Step 5) Now in SQL developer I created, Command shown above.
Step 6) Import the data from excel to my created table in SQL. Click on table Temperatures in the left panel and right click and select IMPORT DATA.



Step 7) Select your excel file in the option browse. Uncheck the header. And click next



Source: Local File

File: ..r\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\Homework\Homework 3\TEMPERATURE.xlsx

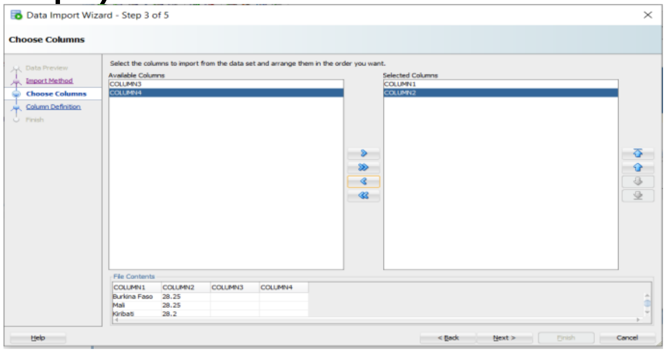
☐ Header Skip Rows: 0

Format: excel 95-2003 (xls) ☑ Preview Row Limit: 100

COLUMN1	COLUMN2	COLUMN3	COLUMN4
Burkina Faso	28.25		
Mali	28.25		
Kiribati	28.2		
Djibouti	28.0		
Tuvalu	28.0		
Senegal	27.85		
Mauritania	27.65		
Maldives	27.65		
Palau	27.6		
Benin	27.55		
Gambia	27.5		
Marshall Isla...	27.4		
Ghana	27.2		
Bahrain	27.15		
Niger	27.15		

Step 8) Click next for next step again

Step 9) Select the first two columns.



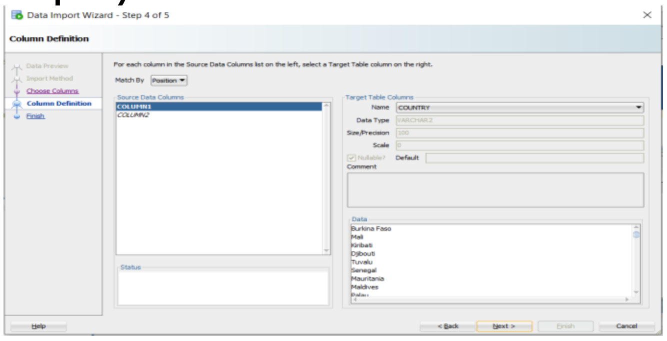
Select the columns to import from the data set and arrange them in the order you want.

Available Columns: COLUMN3, COLUMN4

Selected Columns: COLUMN1, COLUMN2

COLUMN1	COLUMN2	COLUMN3	COLUMN4
Burkina Faso	28.25		
Mali	28.25		
Kiribati	28.2		

Step 10) Check the column definitions. And click Next.



For each column in the Source Data Columns list on the left, select a Target Table column on the right.

Match By: Position

Source Data Columns: COLUMN1, COLUMN2

Target Table Columns

Name: COUNTRY

Data Type: VARCHAR2

Size/Precision: 100

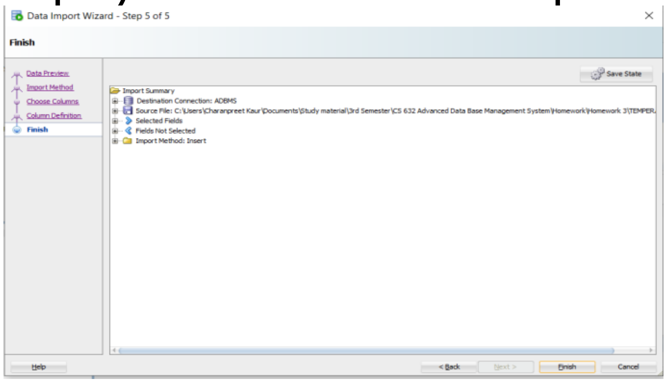
Scale: 0

☑ Nullable? Default:

Comment:

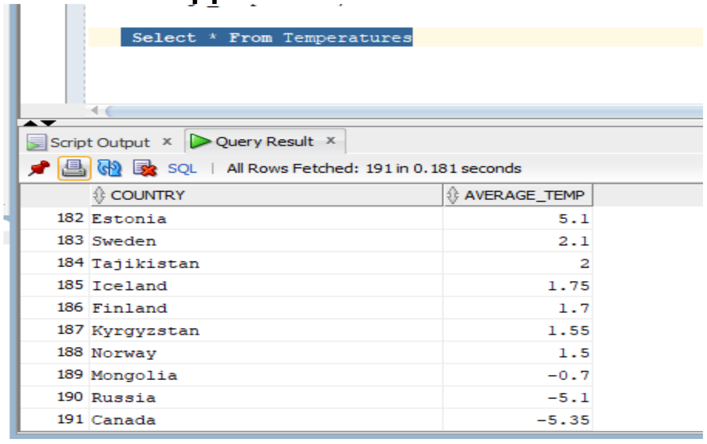
Data: Burkina Faso, Mali, Kiribati, Djibouti, Tuvalu, Senegal, Mauritania, Maldives, Palau

Step 11) Click Finish. And data is imported from excel to table in SQL developer.



```
Import Summary
  Destination Connection: ADBMS
  Source File: C:\Users\Charanpreet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\Homework\Homework 3\TEMPER
  Selected Fields
  Fields Not Selected
  Import Method: Insert
```

Step 12) Run the command select * from Temperatures. And the table with the imported values will appear.



```
Select * From Temperatures
```

All Rows Fetched: 191 in 0.181 seconds

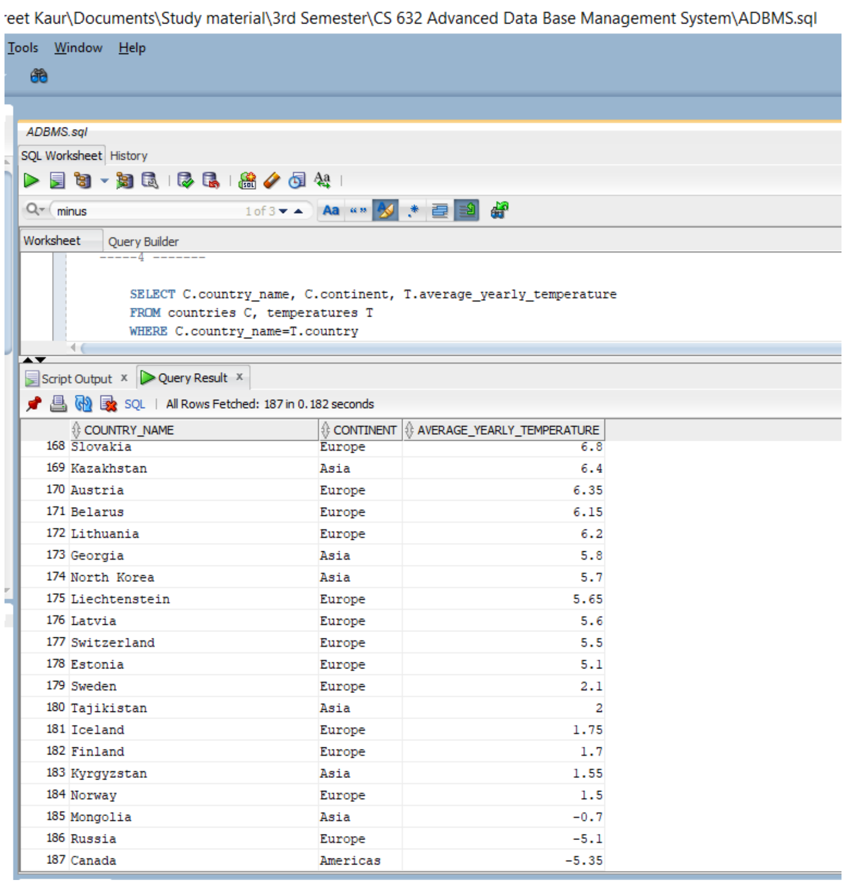
	COUNTRY	AVERAGE_TEMP
182	Estonia	5.1
183	Sweden	2.1
184	Tajikistan	2
185	Iceland	1.75
186	Finland	1.7
187	Kyrgyzstan	1.55
188	Norway	1.5
189	Mongolia	-0.7
190	Russia	-5.1
191	Canada	-5.35

Step 13) Next page contains all the values in table.

Question 4) Write an SQL Select statement that will show three columns: Country, continent and average temperature using the table COUNTRIES and TEMPERATURES. Either this table has 191 rows OR you have to clearly write down in your homework which countries are missing. See the previous homework how to do this with Set Differences.

Answer 4)

SELECT C.country_name, C.continent, T.average_yearly_temperature
FROM countries C, temperatures T
WHERE C.country_name=T.country;



```
------4 -------

SELECT C.country_name, C.continent, T.average_yearly_temperature
FROM countries C, temperatures T
WHERE C.country_name=T.country
```

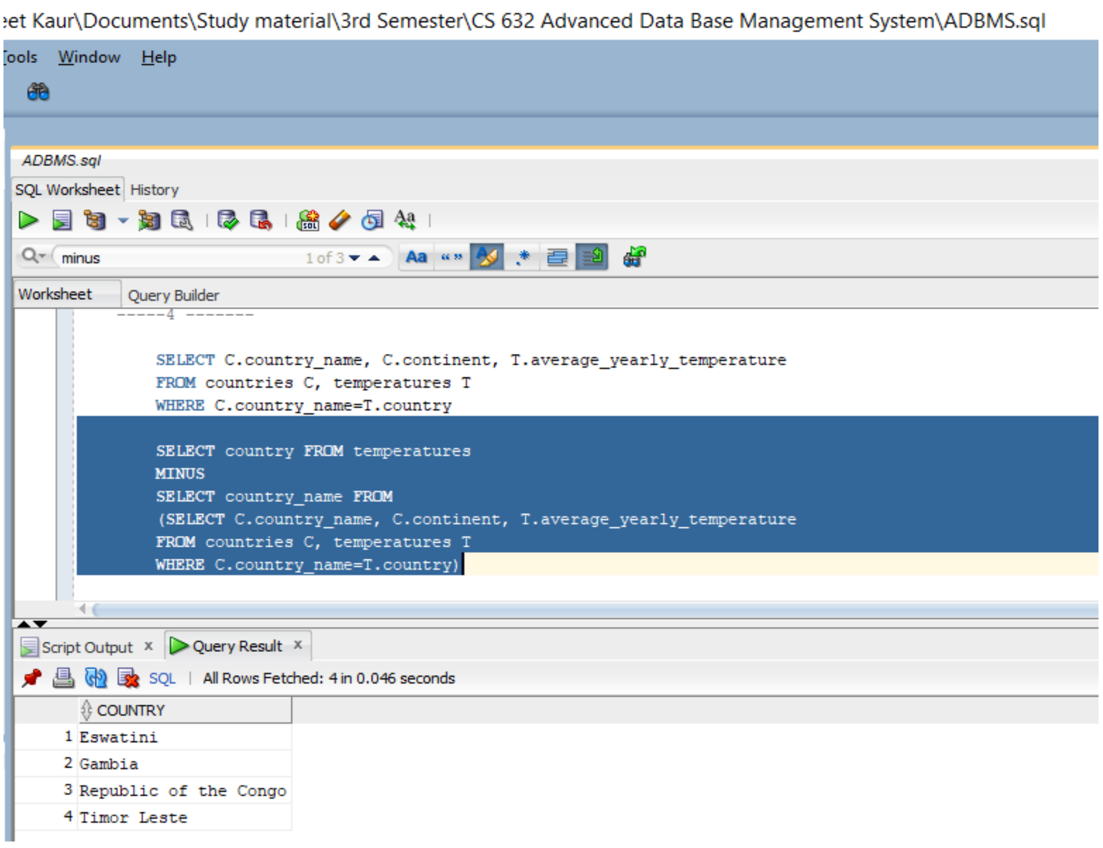
All Rows Fetched: 187 in 0.182 seconds

	COUNTRY_NAME	CONTINENT	AVERAGE_YEARLY_TEMPERATURE
168	Slovakia	Europe	6.8
169	Kazakhstan	Asia	6.4
170	Austria	Europe	6.35
171	Belarus	Europe	6.15
172	Lithuania	Europe	6.2
173	Georgia	Asia	5.8
174	North Korea	Asia	5.7
175	Liechtenstein	Europe	5.65
176	Latvia	Europe	5.6
177	Switzerland	Europe	5.5
178	Estonia	Europe	5.1
179	Sweden	Europe	2.1
180	Tajikistan	Asia	2
181	Iceland	Europe	1.75
182	Finland	Europe	1.7
183	Kyrgyzstan	Asia	1.55
184	Norway	Europe	1.5
185	Mongolia	Asia	-0.7
186	Russia	Europe	-5.1
187	Canada	Americas	-5.35

(Full file in next page)

After running this query only 187 countries appeared.
To see which countries are missing I ran the following command:

SELECT country FROM temperatures
MINUS
SELECT country_name FROM
(SELECT C.country_name, C.continent, T.average_yearly_temperature
FROM countries C, temperatures T
WHERE C.country_name=T.country)



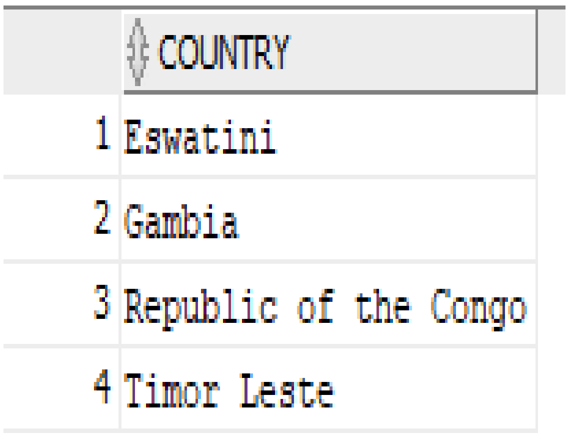
:et Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

```
-----4 --------

    SELECT C.country_name, C.continent, T.average_yearly_temperature
    FROM countries C, temperatures T
    WHERE C.country_name=T.country

    SELECT country FROM temperatures
    MINUS
    SELECT country_name FROM
    (SELECT C.country_name, C.continent, T.average_yearly_temperature
    FROM countries C, temperatures T
    WHERE C.country_name=T.country)
```

All Rows Fetched: 4 in 0.046 seconds

```
  COUNTRY
1 Eswatini
2 Gambia
3 Republic of the Congo
4 Timor Leste
```

The 4 countries that were missing are:



	COUNTRY
1	Eswatini
2	Gambia
3	Republic of the Congo
4	Timor Leste

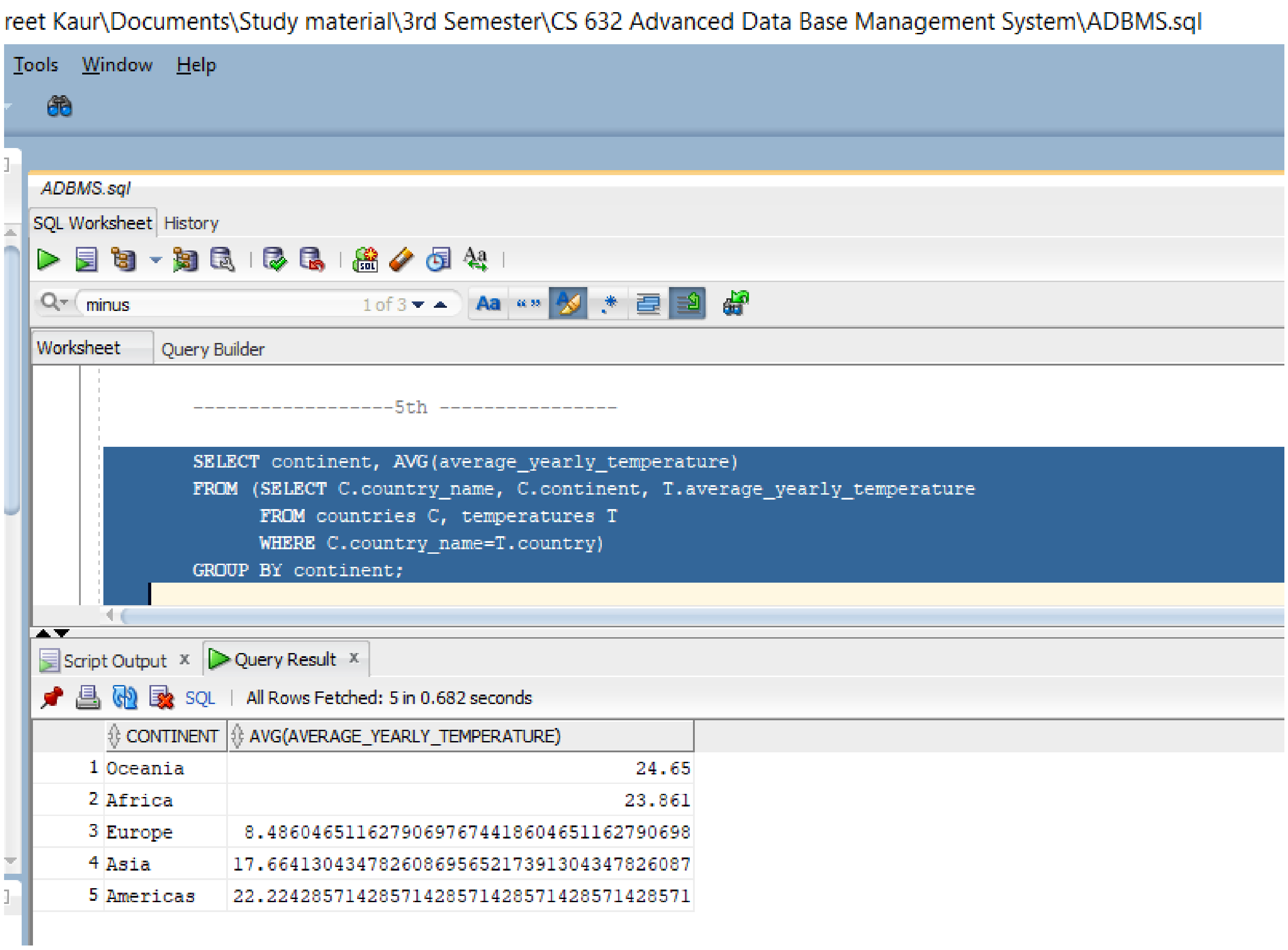
The name of these country in table COUNTRIES appeared as:

1) **Eswatini in table countries appeared as Eswatini Swaziland**
2) **Gambia in table countries appeared as The Gambia**
3) **Republic of the congo in table countries appeared as Democratic Republic of the Congo**
4) **Timor Leste in table countries appeared East Timor**

Question 5) Write an SQL Select statement that produces the following result: Two columns. The first column should contain all the continents according to the table COUNTRIES. The second column should contain the AVERAGE temperature on each continent.

Answer 5)

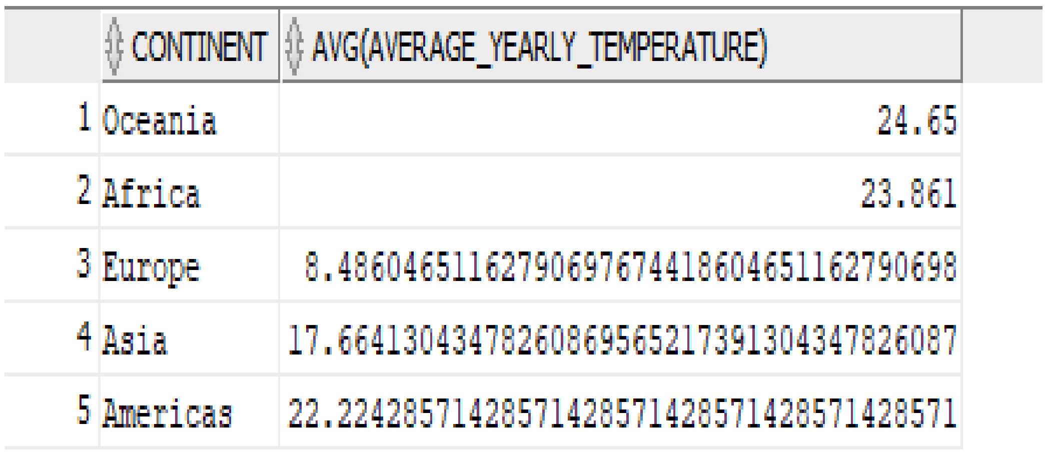
```
                SELECT continent, AVG(average_yearly_temperature)
                 FROM (SELECT C.country_name, C.continent, T.average_yearly_temperature
                          FROM countries C, temperatures T
                           WHERE C.country_name=T.country)
                GROUP BY continent;
```


reet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

```
        ------------------5th ----------------

        SELECT continent, AVG(average_yearly_temperature)
        FROM (SELECT C.country_name, C.continent, T.average_yearly_temperature
              FROM countries C, temperatures T
              WHERE C.country_name=T.country)
        GROUP BY continent;
```

All Rows Fetched: 5 in 0.682 seconds

	CONTINENT	AVG(AVERAGE_YEARLY_TEMPERATURE)
1	Oceania	24.65
2	Africa	23.861
3	Europe	8.48604651162790697674418604651162790698
4	Asia	17.66413043478260869565217391304347826087
5	Americas	22.22428571428571428571428571428571428571

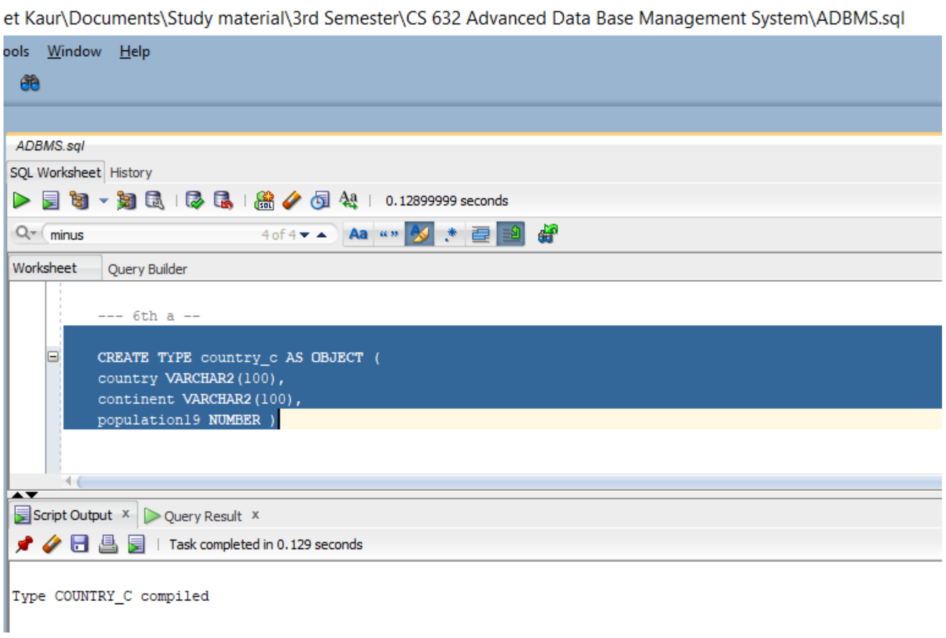


	CONTINENT	AVG(AVERAGE_YEARLY_TEMPERATURE)
1	Oceania	24.65
2	Africa	23.861
3	Europe	8.48604651162790697674418604651162790698
4	Asia	17.66413043478260869565217391304347826087
5	Americas	22.22428571428571428571428571428571428571

Question 6) a)
Create a class in SQL called COUNTRY_C. It should have data attributes Country, Continent, Population2019.

Answer 6 a)

```
CREATE TYPE country_c AS OBJECT (
        country VARCHAR2(100),
        continent VARCHAR2(100),
        population19 NUMBER );
```


et Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

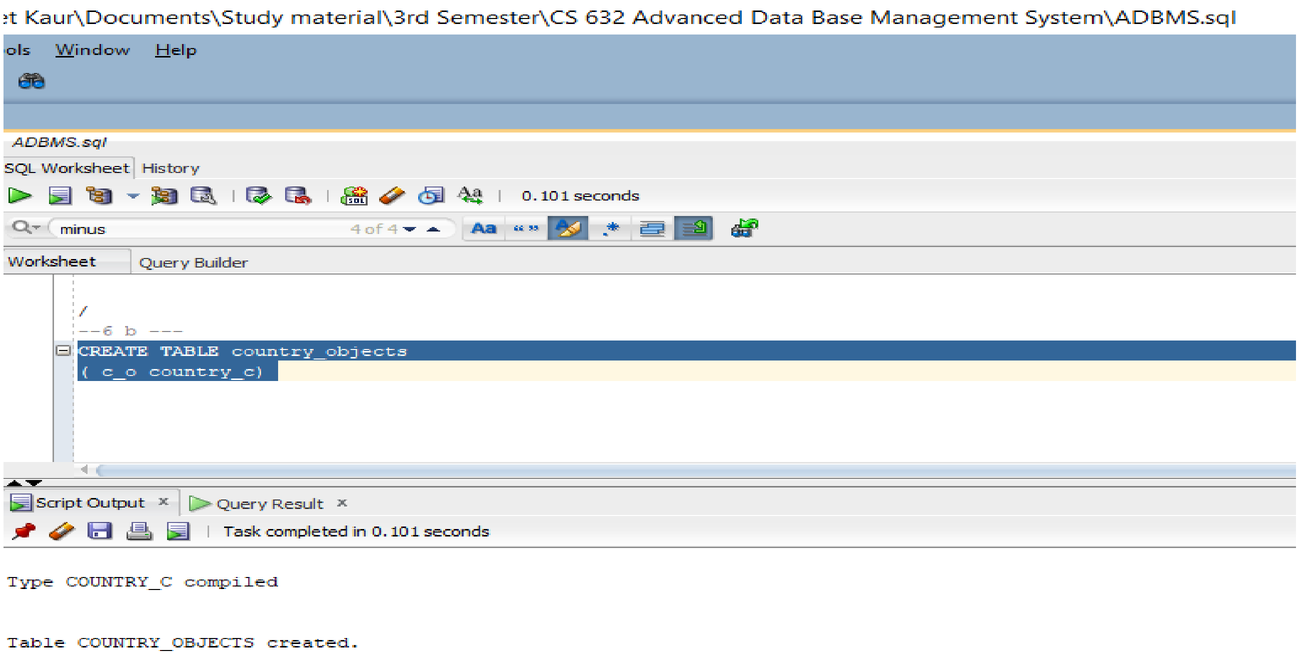
```
--- 6th a --

CREATE TYPE country_c AS OBJECT (
country VARCHAR2(100),
continent VARCHAR2(100),
population19 NUMBER )
```

```
Type COUNTRY_C compiled
```

b) Create a table COUNTRY_OBJECTS that contains **only one column** of type COUNTRY_C. The column should be named C_O.

Answer b)

```
CREATE TABLE country_objects ( c_o country_c)
```


ɛt Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

```
/
--6 b ---
CREATE TABLE country_objects
( c_o country_c)
```

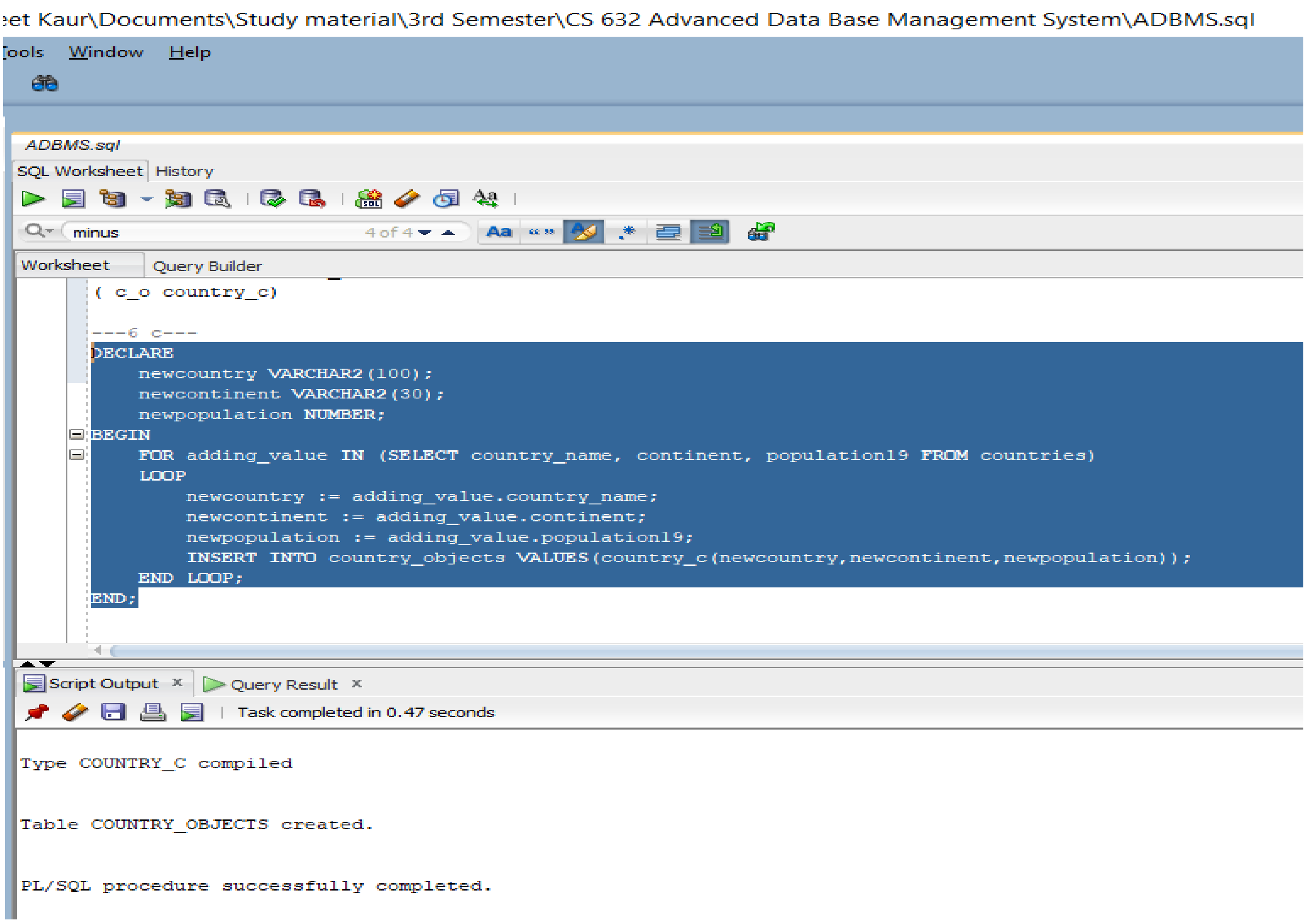
```
Type COUNTRY_C compiled


Table COUNTRY_OBJECTS created.
```

c) Write a PL/SQL program, using a cursor that will insert all rows from COUNTRIES from the columns Country, Continent, and Population2019 into COUNTRY_OBJECTS. One row of data from COUNTRIES should appear as one object in COUNTRY_OBJECTS.

In other words, the first row of COUNTRY_OBJECTS will contain one object that contains the values China, Asia, and 1433783686. The second row will contain one object with the attribute values India, Asia, 1366417754, and so on. The last row will contain an object with Vatican City, Europe, 799.

Answer c)

```
DECLARE
   newcountry VARCHAR2(100);
   newcontinent VARCHAR2(30);
   newpopulation NUMBER;
BEGIN
   FOR adding_value IN (SELECT country_name, continent, population19 FROM countries)
   LOOP
      newcountry := adding_value.country_name;
      newcontinent := adding_value.continent;
      newpopulation := adding_value.population19;
      INSERT INTO country_objects
VALUES(country_c(newcountry,newcontinent,newpopulation));
   END LOOP;
END;
```

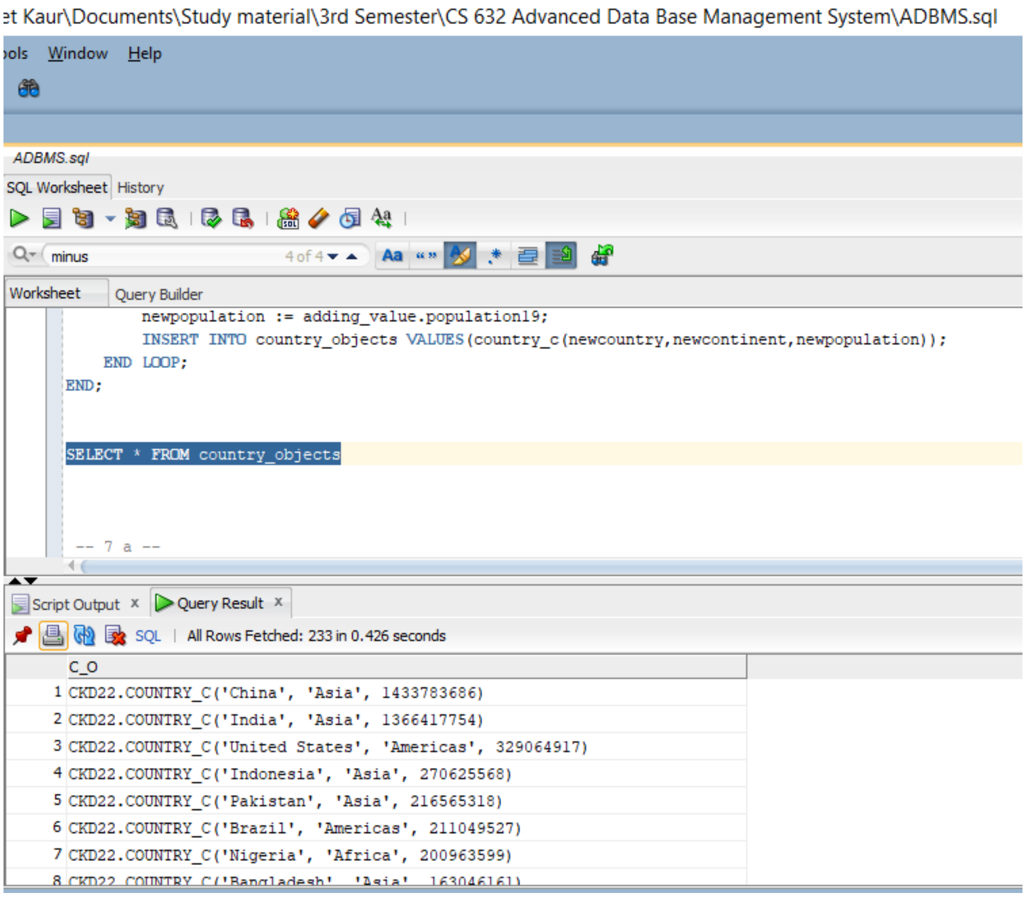

eet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql
Tools Window Help
ADBMS.sql
SQL Worksheet History
minus 4 of 4
Worksheet Query Builder
Script Output Query Result

```
( c_o country_c)

---6 c---
DECLARE
    newcountry VARCHAR2(100);
    newcontinent VARCHAR2(30);
    newpopulation NUMBER;
BEGIN
    FOR adding_value IN (SELECT country_name, continent, population19 FROM countries)
    LOOP
        newcountry := adding_value.country_name;
        newcontinent := adding_value.continent;
        newpopulation := adding_value.population19;
        INSERT INTO country_objects VALUES(country_c(newcountry,newcontinent,newpopulation));
    END LOOP;
END;
```

Task completed in 0.47 seconds

```
Type COUNTRY_C compiled


Table COUNTRY_OBJECTS created.


PL/SQL procedure successfully completed.
```

SELECT * FROM country_objects (full result in next page)



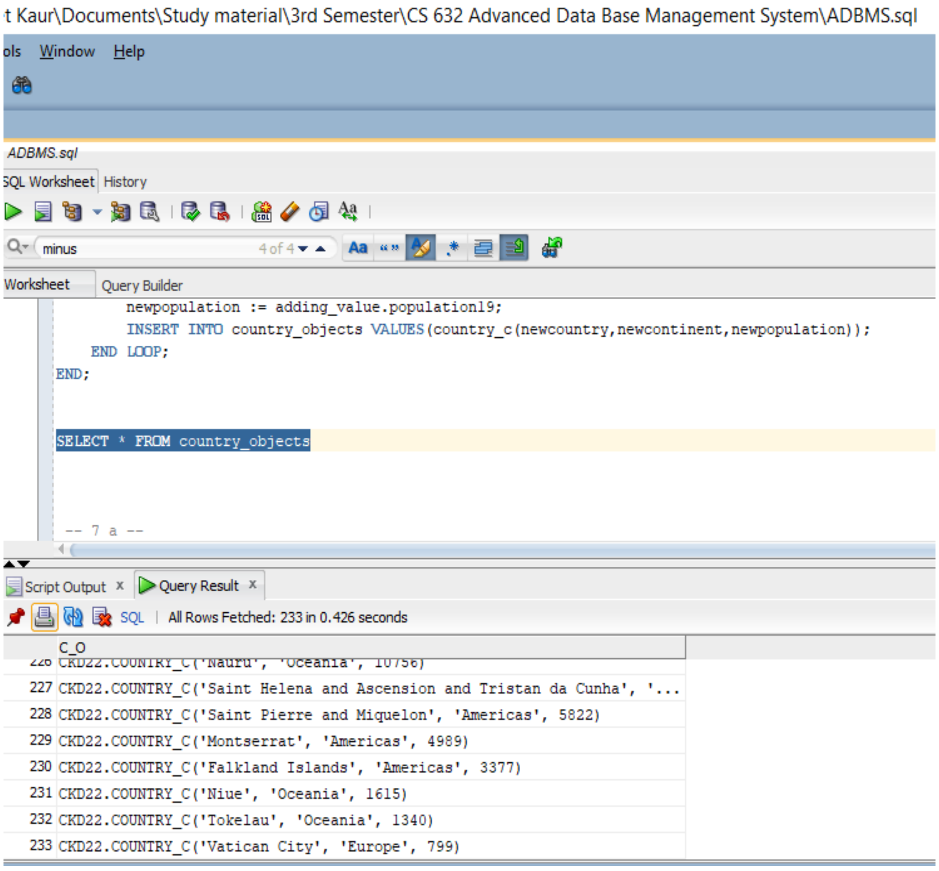
```
        newpopulation := adding_value.population19;
        INSERT INTO country_objects VALUES(country_c(newcountry,newcontinent,newpopulation));
    END LOOP;
END;

SELECT * FROM country_objects

-- 7 a --
```

All Rows Fetched: 233 in 0.426 seconds

```
  C_O
1 CKD22.COUNTRY_C('China', 'Asia', 1433783686)
2 CKD22.COUNTRY_C('India', 'Asia', 1366417754)
3 CKD22.COUNTRY_C('United States', 'Americas', 329064917)
4 CKD22.COUNTRY_C('Indonesia', 'Asia', 270625568)
5 CKD22.COUNTRY_C('Pakistan', 'Asia', 216565318)
6 CKD22.COUNTRY_C('Brazil', 'Americas', 211049527)
7 CKD22.COUNTRY_C('Nigeria', 'Africa', 200963599)
8 CKD22.COUNTRY_C('Bangladesh', 'Asia', 163046161)
```

……

……



```
        newpopulation := adding_value.population19;
        INSERT INTO country_objects VALUES(country_c(newcountry,newcontinent,newpopulation));
    END LOOP;
END;

SELECT * FROM country_objects

-- 7 a --
```

All Rows Fetched: 233 in 0.426 seconds

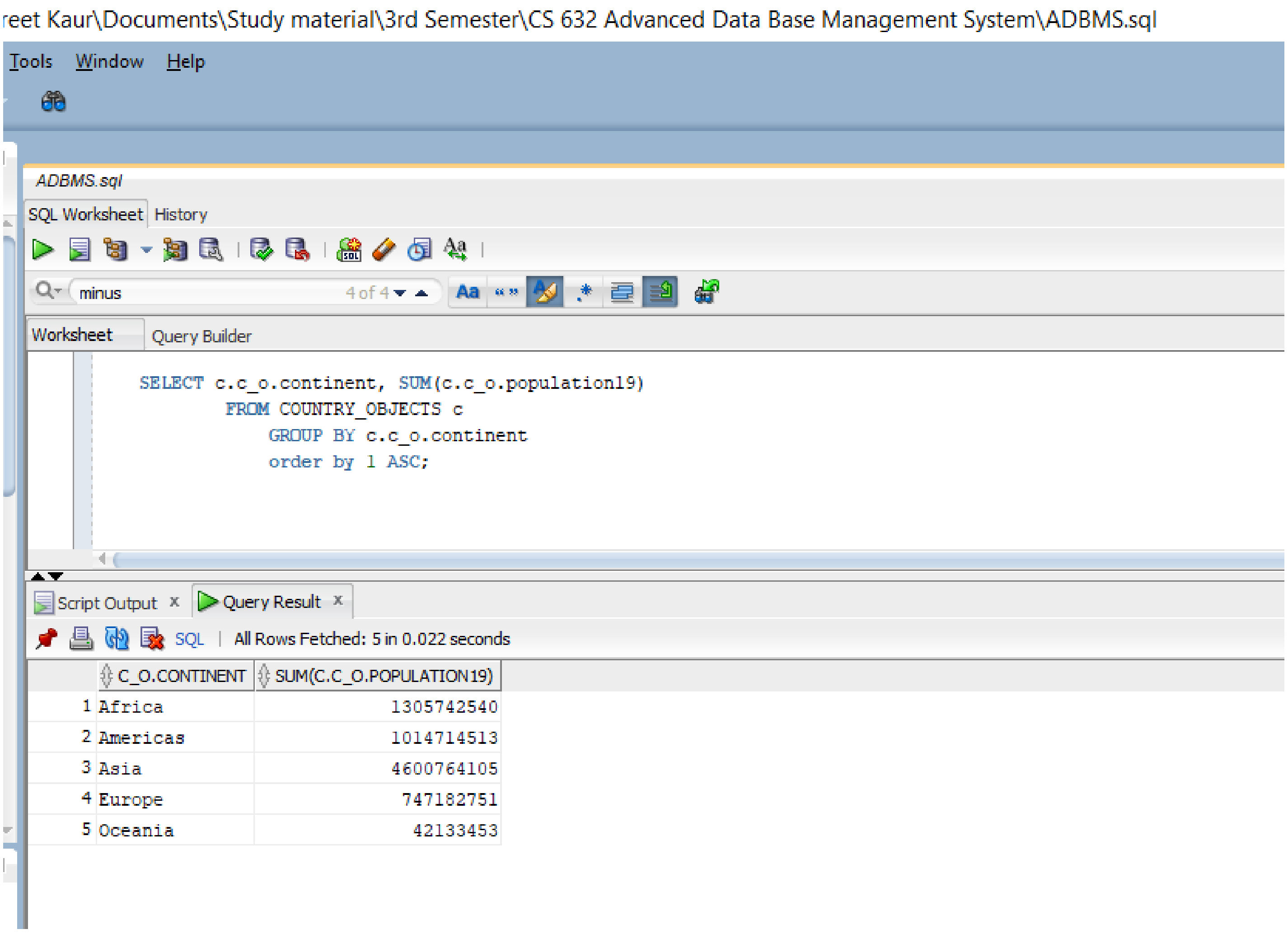
```
    C_O
226 CKD22.COUNTRY_C('Nauru', 'Oceania', 10756)
227 CKD22.COUNTRY_C('Saint Helena and Ascension and Tristan da Cunha', '...
228 CKD22.COUNTRY_C('Saint Pierre and Miquelon', 'Americas', 5822)
229 CKD22.COUNTRY_C('Montserrat', 'Americas', 4989)
230 CKD22.COUNTRY_C('Falkland Islands', 'Americas', 3377)
231 CKD22.COUNTRY_C('Niue', 'Oceania', 1615)
232 CKD22.COUNTRY_C('Tokelau', 'Oceania', 1340)
233 CKD22.COUNTRY_C('Vatican City', 'Europe', 799)
```

d) Write an SQL Select statement using **ONLY** the table COUNTRY_OBJECTS that will display one line for each Continent and the total 2019 population (sum) of all the countries in that Continent.

Hint: use group by. So there will be only two columns in this result. Show the result in alphabetical order by continent.

Answer d)

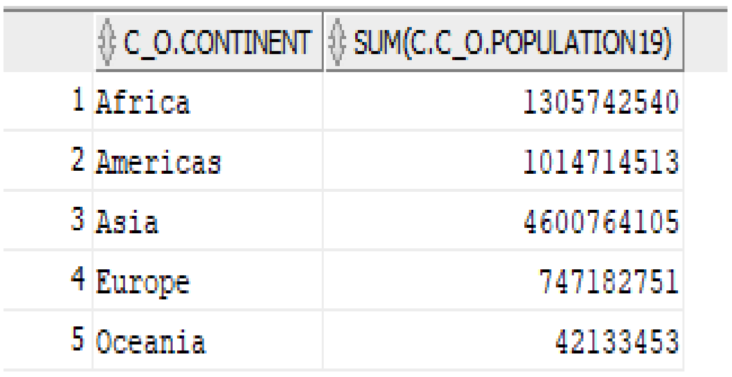
SELECT c.c_o.continent, SUM(c.c_o.population19)
FROM COUNTRY_OBJECTS c
GROUP BY c.c_o.continent
order by 1 ASC;



```
SELECT c.c_o.continent, SUM(c.c_o.population19)
        FROM COUNTRY_OBJECTS c
            GROUP BY c.c_o.continent
            order by 1 ASC;
```

All Rows Fetched: 5 in 0.022 seconds

	C_O.CONTINENT	SUM(C.C_O.POPULATION19)
1	Africa	1305742540
2	Americas	1014714513
3	Asia	4600764105
4	Europe	747182751
5	Oceania	42133453



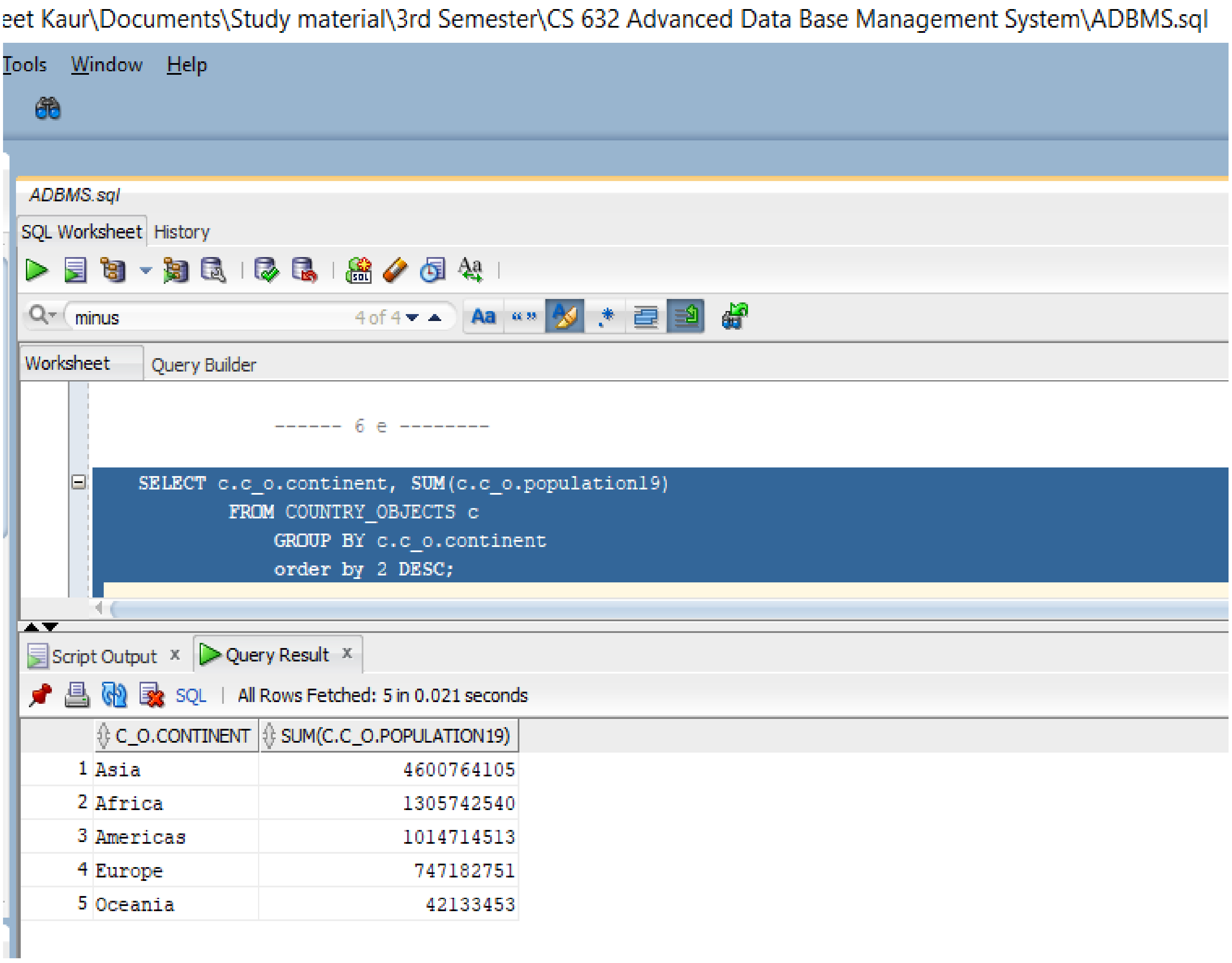
	C_O.CONTINENT	SUM(C.C_O.POPULATION19)
1	Africa	1305742540
2	Americas	1014714513
3	Asia	4600764105
4	Europe	747182751
5	Oceania	42133453

e) Write an SQL Select statement using **ONLY** the table COUNTRY_OBJECTS that will display one line for each Continent and the total 2019 population (sum) of all the countries in that Continent.

Hint: use group by. So there will be only two columns in this result. Show the result in descending order by population.

Answer e)

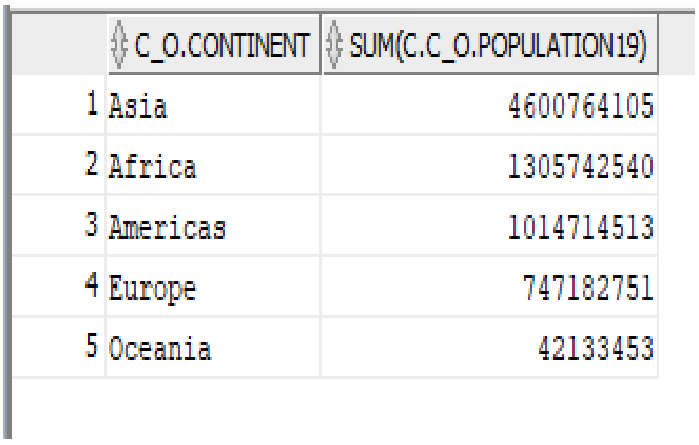
SELECT c.c_o.continent, SUM(c.c_o.population19)
FROM COUNTRY_OBJECTS c
GROUP BY c.c_o.continent
order by 2 DESC;



```
------ 6 e --------

SELECT c.c_o.continent, SUM(c.c_o.population19)
        FROM COUNTRY_OBJECTS c
            GROUP BY c.c_o.continent
            order by 2 DESC;
```

All Rows Fetched: 5 in 0.021 seconds

	C_O.CONTINENT	SUM(C.C_O.POPULATION19)
1	Asia	4600764105
2	Africa	1305742540
3	Americas	1014714513
4	Europe	747182751
5	Oceania	42133453

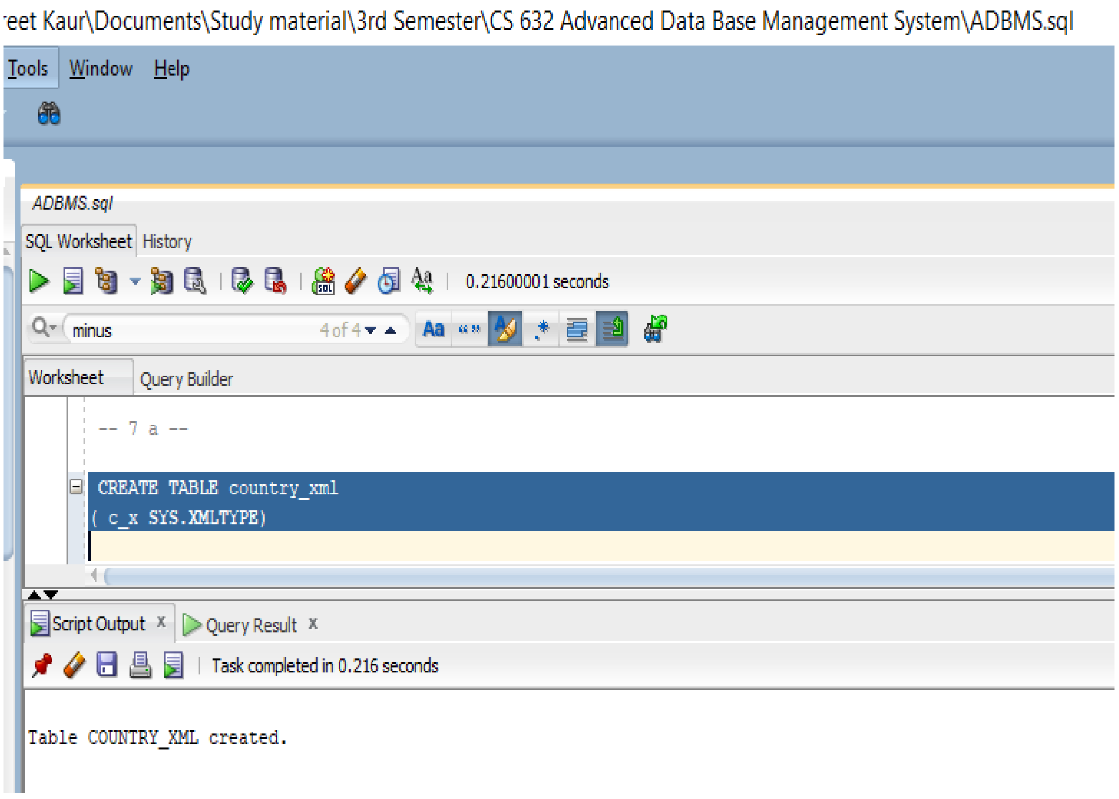


	C_O.CONTINENT	SUM(C.C_O.POPULATION19)
1	Asia	4600764105
2	Africa	1305742540
3	Americas	1014714513
4	Europe	747182751
5	Oceania	42133453

Question 7) a)
Create a table COUNTRY_XML that contains **only one column** of type sys.xmltype. The column should be named C_X.

Answer 7 a)

CREATE TABLE country_xml (c_x SYS.XMLTYPE)



```
eet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql
```

```
-- 7 a --

CREATE TABLE country_xml
( c_x SYS.XMLTYPE)
```

```
Task completed in 0.216 seconds

Table COUNTRY_XML created.
```

b) Write a PL/SQL program, using a cursor that will insert all rows from COUNTRIES from the columns Country, Continent, and Population2019 into COUNTRY_XML. One row of data from COUNTRIES should appear as one XML tree in COUNTRY_XML. Use the tags as given in the example below.

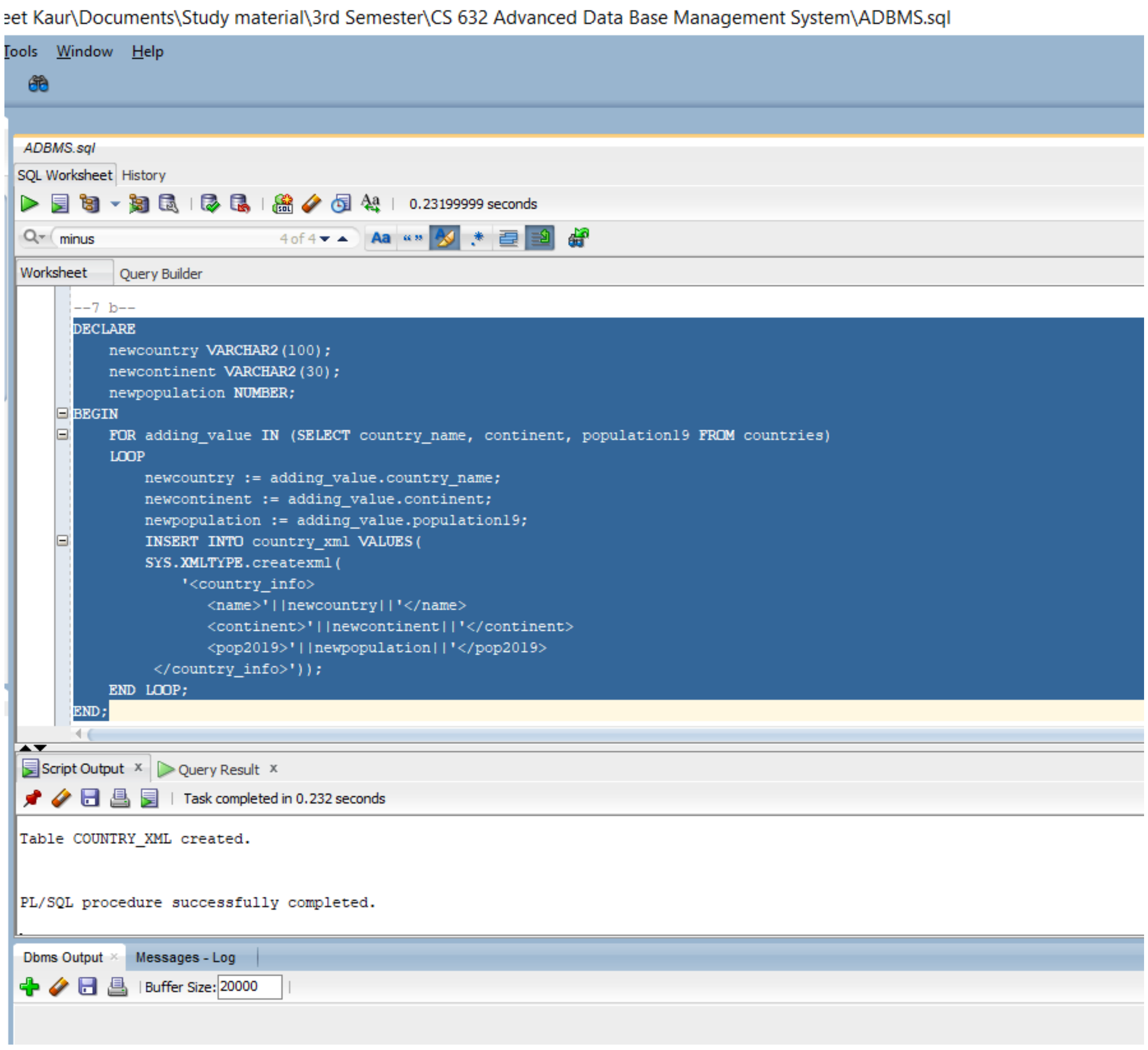
The first row should appear as:

```
<country_info>
   <name>China</name>
   <continent>Asia</continent>
   <pop2019>1433783686</pop2019>
</country_info>
```

HINT: Traditionally this question is usually difficult for students. You need to extract the values like “India” using a cursor and then with LOTS of concatenations construct the whole XML expressions.
    <continent>’ || a_variable_that_contains_Asia || ‘</continent>

Answer b)

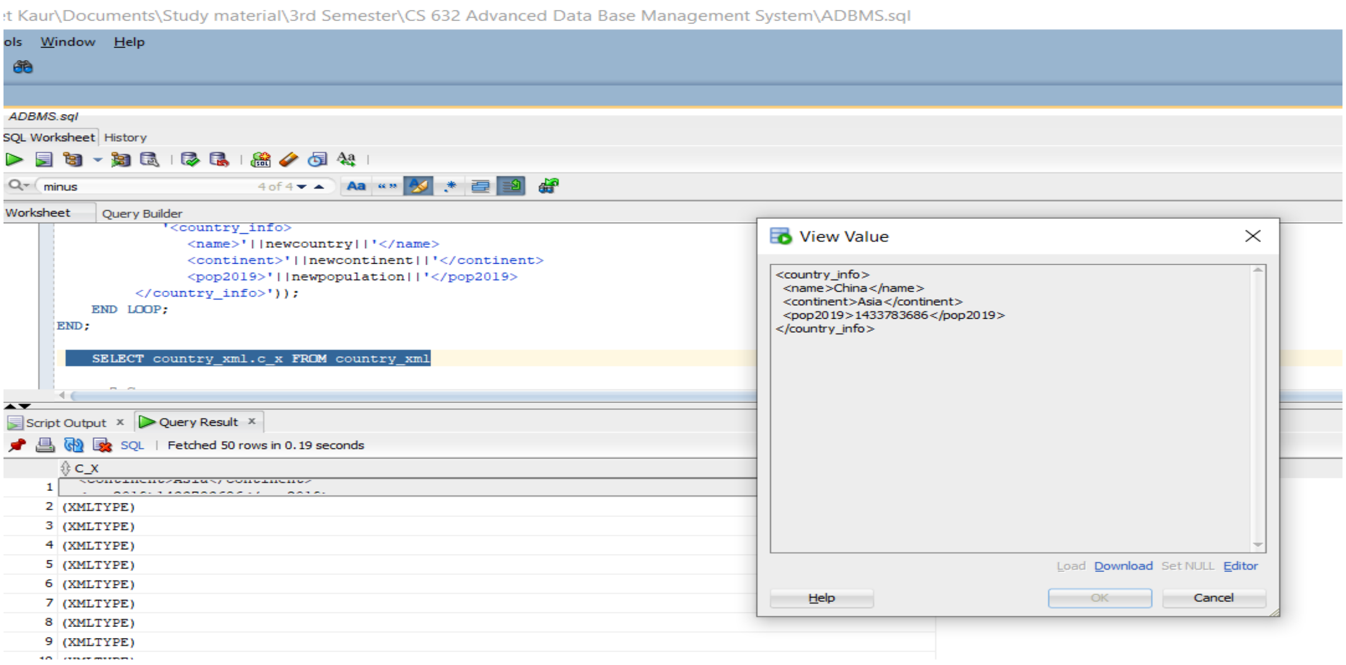
```
 DECLARE
    newcountry VARCHAR2(100);
    newcontinent VARCHAR2(30);
    newpopulation NUMBER;
BEGIN
    FOR adding_value IN (SELECT country_name, continent, population19 FROM countries)
    LOOP
        newcountry := adding_value.country_name;
        newcontinent := adding_value.continent;
        newpopulation := adding_value.population19;
        INSERT INTO country_xml VALUES(
        SYS.XMLTYPE.createxml(
            '<country_info>
               <name>'||newcountry||'</name>
               <continent>'||newcontinent||'</continent>
               <pop2019>'||newpopulation||'</pop2019>
         </country_info>'));
    END LOOP;
END;
```



```
--7 b--
DECLARE
    newcountry VARCHAR2(100);
    newcontinent VARCHAR2(30);
    newpopulation NUMBER;
BEGIN
    FOR adding_value IN (SELECT country_name, continent, population19 FROM countries)
    LOOP
        newcountry := adding_value.country_name;
        newcontinent := adding_value.continent;
        newpopulation := adding_value.population19;
        INSERT INTO country_xml VALUES(
        SYS.XMLTYPE.createxml(
            '<country_info>
               <name>'||newcountry||'</name>
               <continent>'||newcontinent||'</continent>
               <pop2019>'||newpopulation||'</pop2019>
         </country_info>'));
    END LOOP;
END;
```

```
Table COUNTRY_XML created.


PL/SQL procedure successfully completed.
```

SELECT country_xml.c_x FROM country_xml



```
'<country_info>
    <name>'||newcountry||'</name>
    <continent>'||newcontinent||'</continent>
    <pop2019>'||newpopulation||'</pop2019>
</country_info>'));
    END LOOP;
END;

SELECT country_xml.c_x FROM country_xml
```

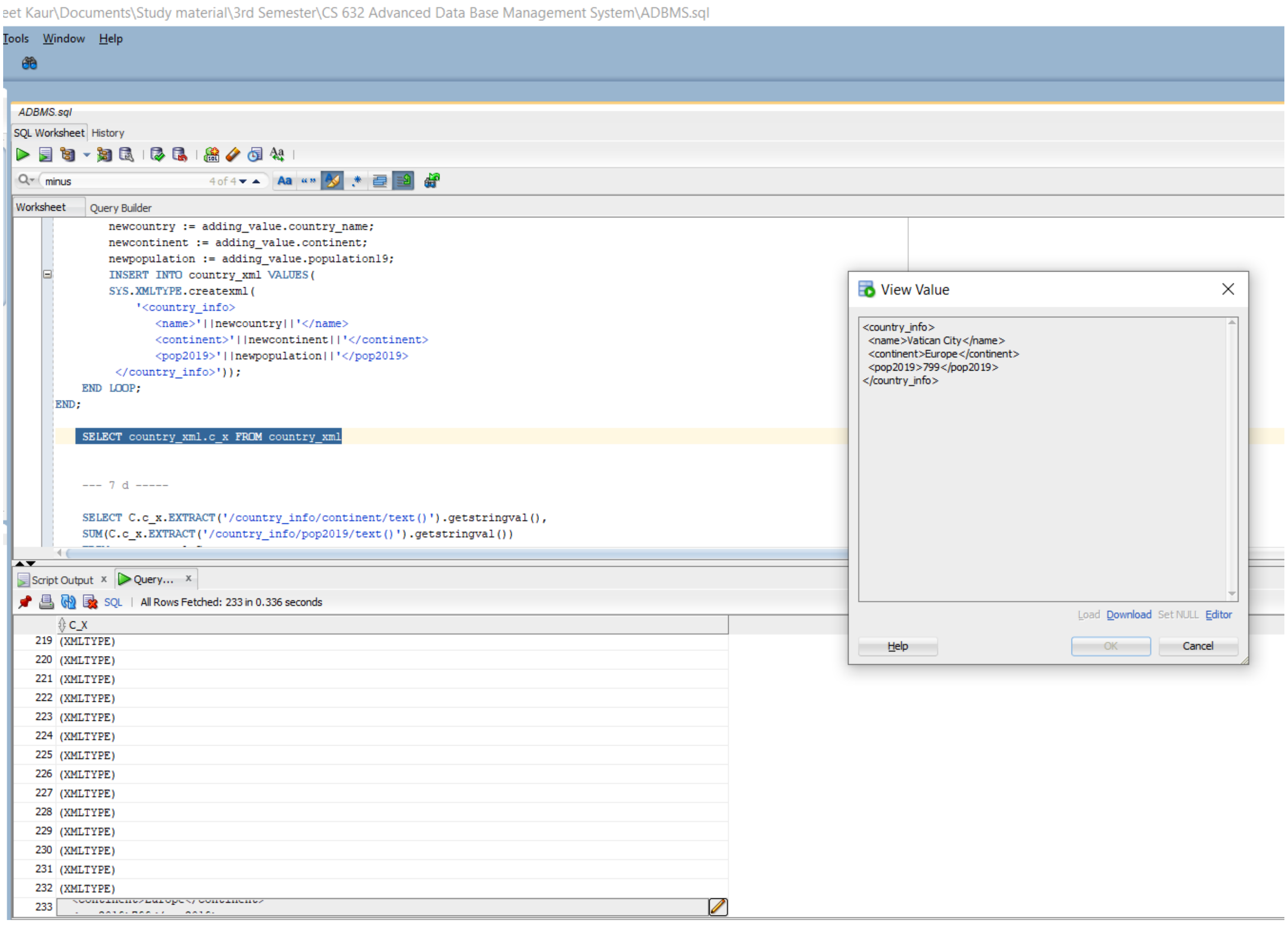
Fetched 50 rows in 0.19 seconds

```
    C_X
 1
 2  (XMLTYPE)
 3  (XMLTYPE)
 4  (XMLTYPE)
 5  (XMLTYPE)
 6  (XMLTYPE)
 7  (XMLTYPE)
 8  (XMLTYPE)
 9  (XMLTYPE)
```

View Value

```
<country_info>
  <name>China</name>
  <continent>Asia</continent>
  <pop2019>1433783686</pop2019>
</country_info>
```

t Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql
ols Window Help
ADBMS.sql
SQL Worksheet History
minus 4 of 4
Worksheet Query Builder
Script Output
Query Result
SQL
Load Download Set NULL Editor
Help OK Cancel



```
        newcountry := adding_value.country_name;
        newcontinent := adding_value.continent;
        newpopulation := adding_value.population19;
        INSERT INTO country_xml VALUES(
        SYS.XMLTYPE.createxml(
            '<country_info>
                <name>'||newcountry||'</name>
                <continent>'||newcontinent||'</continent>
                <pop2019>'||newpopulation||'</pop2019>
         </country_info>'));
    END LOOP;
END;

    SELECT country_xml.c_x FROM country_xml


    --- 7 d -----

    SELECT C.c_x.EXTRACT('/country_info/continent/text()').getstringval(),
    SUM(C.c_x.EXTRACT('/country_info/pop2019/text()').getstringval())
```

All Rows Fetched: 233 in 0.336 seconds

```
    C_X
219 (XMLTYPE)
220 (XMLTYPE)
221 (XMLTYPE)
222 (XMLTYPE)
223 (XMLTYPE)
224 (XMLTYPE)
225 (XMLTYPE)
226 (XMLTYPE)
227 (XMLTYPE)
228 (XMLTYPE)
229 (XMLTYPE)
230 (XMLTYPE)
231 (XMLTYPE)
232 (XMLTYPE)
233
```

View Value

```
<country_info>
  <name>Vatican City</name>
  <continent>Europe</continent>
  <pop2019>799</pop2019>
</country_info>
```

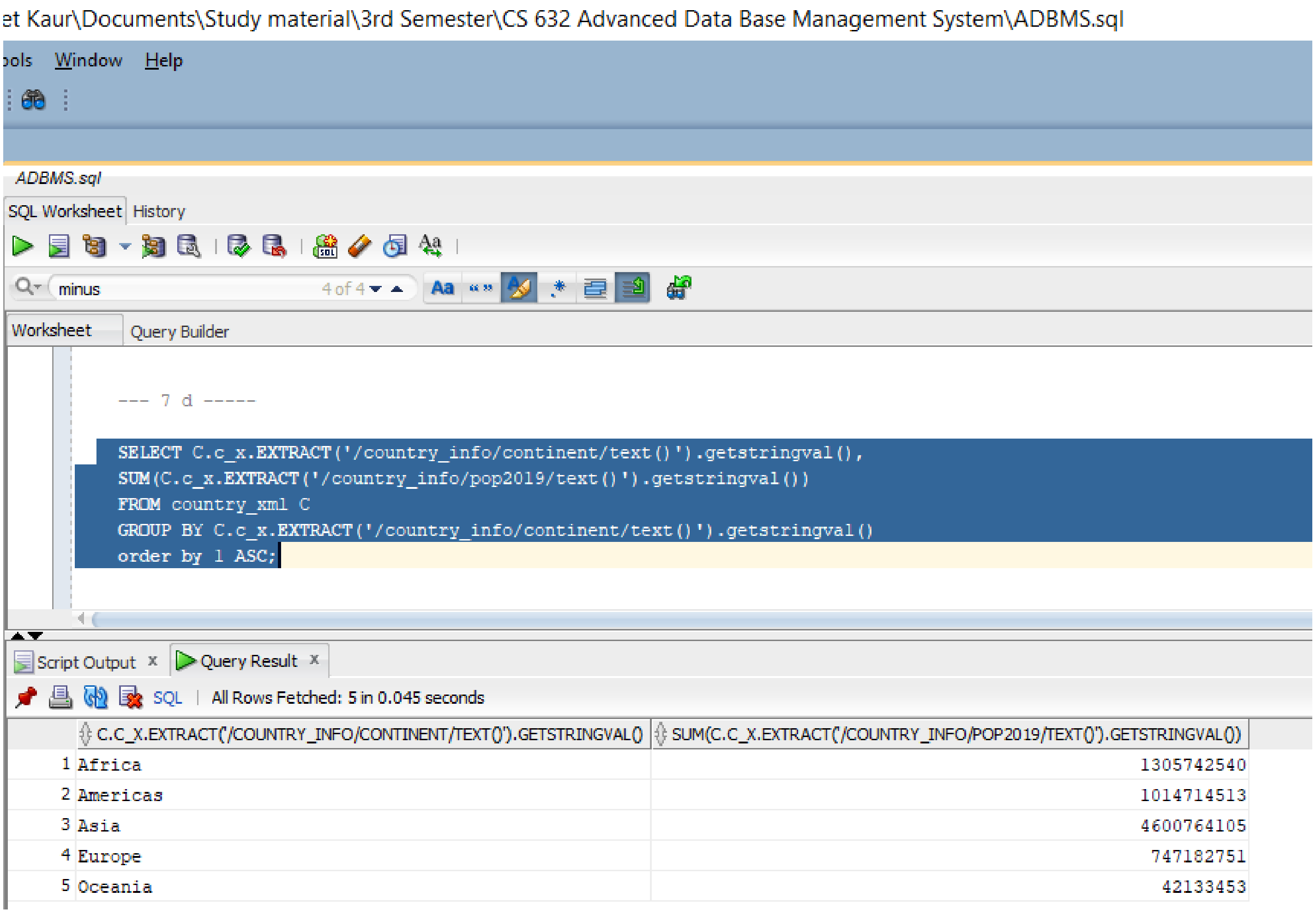
eet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql
Tools Window Help
ADBMS.sql
SQL Worksheet History
minus 4 of 4
Worksheet Query Builder
Script Output
Query...
SQL
Load Download Set NULL Editor
Help OK Cancel

d) Write an SQL Select statement using **ONLY** the table COUNTRY_XML that will display one line for each Continent and the total 2019 population (sum) of all the countries in that Continent.

Hint: use group by. So there will be only two columns in this result. Show the result in alphabetical order by continent.

Answer d)

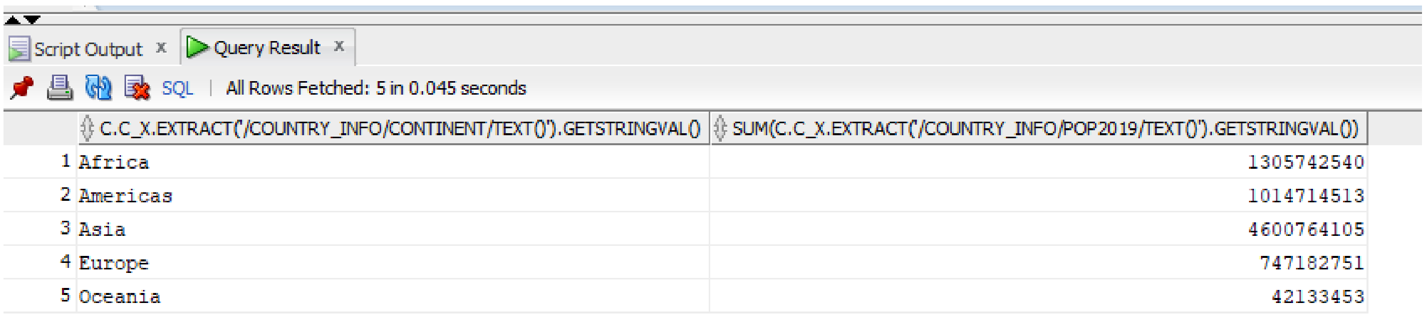
```
SELECT C.c_x.EXTRACT('/country_info/continent/text()').getstringval(),
        SUM(C.c_x.EXTRACT('/country_info/pop2019/text()').getstringval())
FROM country_xml C
GROUP BY C.c_x.EXTRACT('/country_info/continent/text()').getstringval()
order by 1 ASC;
```


et Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql
ools Window Help
ADBMS.sql
SQL Worksheet History
minus 4 of 4
Worksheet Query Builder
Script Output Query Result
SQL

```
--- 7 d -----

SELECT C.c_x.EXTRACT('/country_info/continent/text()').getstringval(),
SUM(C.c_x.EXTRACT('/country_info/pop2019/text()').getstringval())
FROM country_xml C
GROUP BY C.c_x.EXTRACT('/country_info/continent/text()').getstringval()
order by 1 ASC;
```

All Rows Fetched: 5 in 0.045 seconds

	C.C_X.EXTRACT('/COUNTRY_INFO/CONTINENT/TEXT()').GETSTRINGVAL()	SUM(C.C_X.EXTRACT('/COUNTRY_INFO/POP2019/TEXT()').GETSTRINGVAL())
1	Africa	1305742540
2	Americas	1014714513
3	Asia	4600764105
4	Europe	747182751
5	Oceania	42133453



All Rows Fetched: 5 in 0.045 seconds

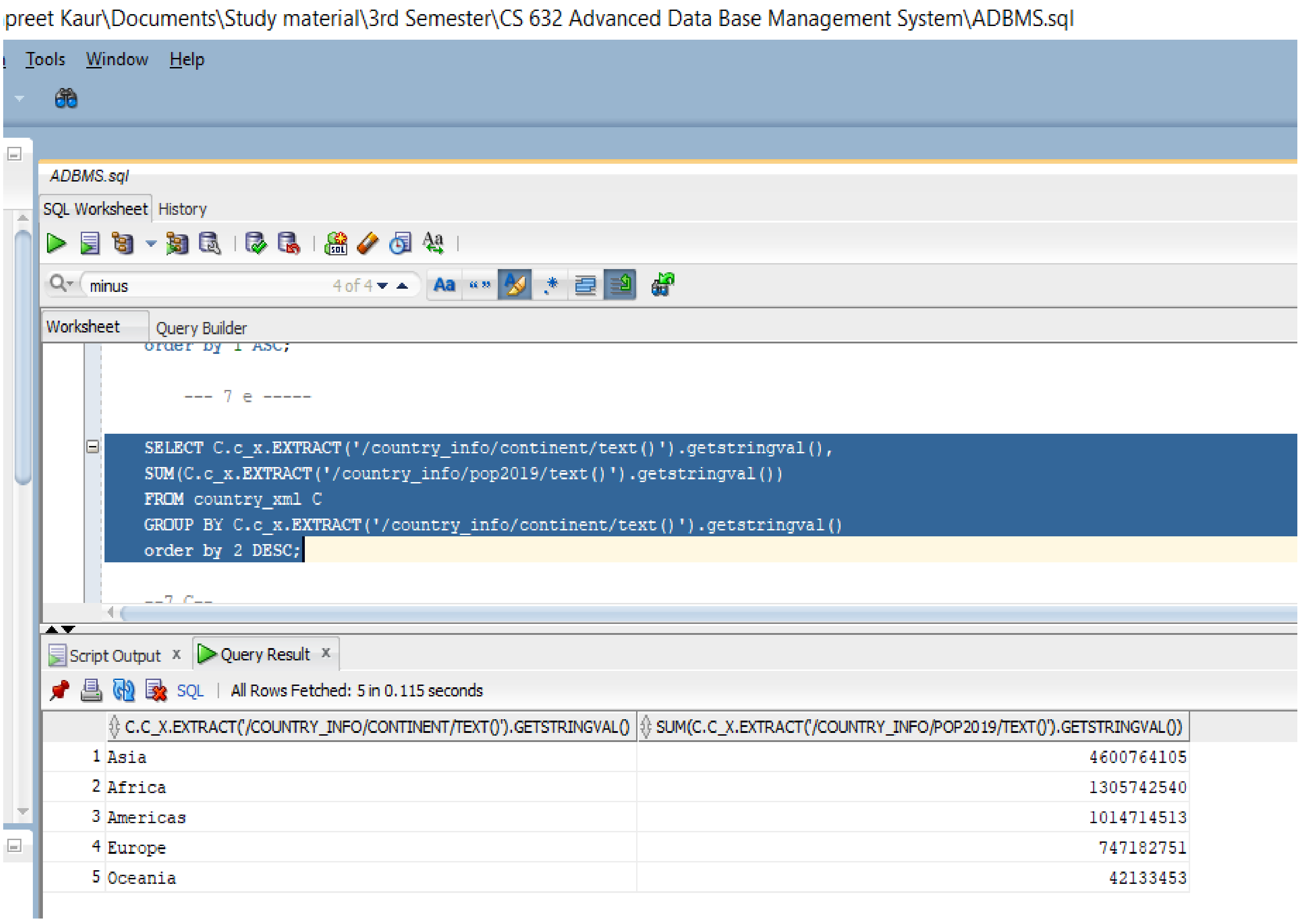
	C.C_X.EXTRACT('/COUNTRY_INFO/CONTINENT/TEXT()').GETSTRINGVAL()	SUM(C.C_X.EXTRACT('/COUNTRY_INFO/POP2019/TEXT()').GETSTRINGVAL())
1	Africa	1305742540
2	Americas	1014714513
3	Asia	4600764105
4	Europe	747182751
5	Oceania	42133453

e) Write an SQL Select statement using **ONLY** the table COUNTRY_XML that will display one line for each Continent and the total 2019 population (sum) of all the countries in that Continent.

Hint: use group by. So there will be only two columns in this result. Show the result in descending order by population.

Answer e)

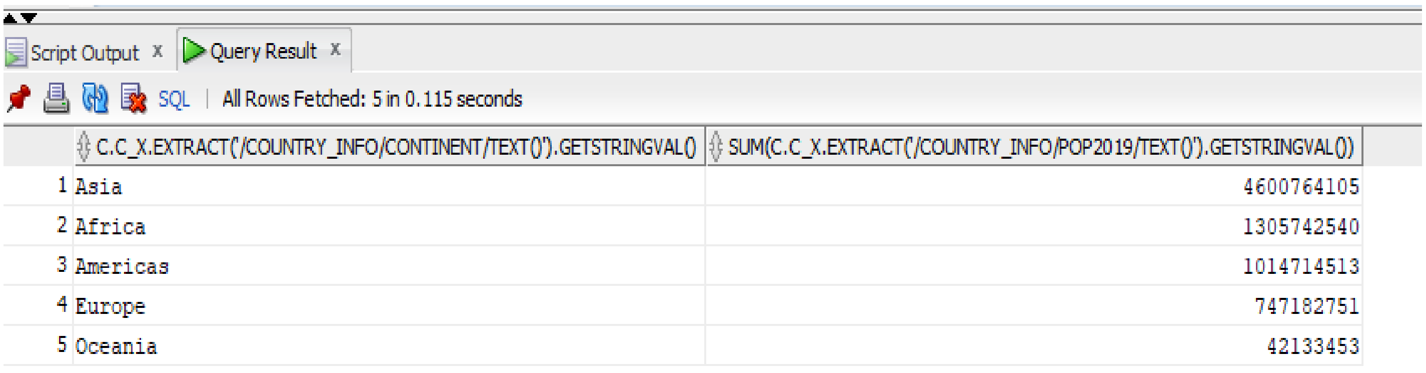
```
SELECT C.c_x.EXTRACT('/country_info/continent/text()').getstringval(),
           SUM(C.c_x.EXTRACT('/country_info/pop2019/text()').getstringval())
FROM country_xml C
GROUP BY C.c_x.EXTRACT('/country_info/continent/text()').getstringval()
order by 2 DESC;
```


preet Kaur\Documents\Study material\3rd Semester\CS 632 Advanced Data Base Management System\ADBMS.sql

```
order by 1 ASC;

    --- 7 e -----

SELECT C.c_x.EXTRACT('/country_info/continent/text()').getstringval(),
SUM(C.c_x.EXTRACT('/country_info/pop2019/text()').getstringval())
FROM country_xml C
GROUP BY C.c_x.EXTRACT('/country_info/continent/text()').getstringval()
order by 2 DESC;

--7 C--
```

Script Output × Query Result ×
SQL | All Rows Fetched: 5 in 0.115 seconds

```
  C.C_X.EXTRACT('/COUNTRY_INFO/CONTINENT/TEXT()').GETSTRINGVAL()  SUM(C.C_X.EXTRACT('/COUNTRY_INFO/POP2019/TEXT()').GETSTRINGVAL())
1 Asia                                                                                                          4600764105
2 Africa                                                                                                        1305742540
3 Americas                                                                                                      1014714513
4 Europe                                                                                                         747182751
5 Oceania                                                                                                         42133453
```

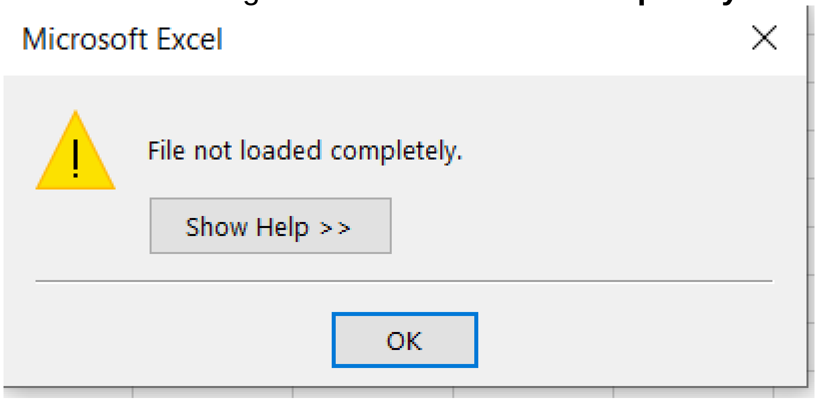

Script Output × Query Result ×
SQL | All Rows Fetched: 5 in 0.115 seconds

```
  C.C_X.EXTRACT('/COUNTRY_INFO/CONTINENT/TEXT()').GETSTRINGVAL()  SUM(C.C_X.EXTRACT('/COUNTRY_INFO/POP2019/TEXT()').GETSTRINGVAL())
1 Asia                                                                                                          4600764105
2 Africa                                                                                                        1305742540
3 Americas                                                                                                      1014714513
4 Europe                                                                                                         747182751
5 Oceania                                                                                                         42133453
```

Question 8) a) Go to: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
Go to 2019>January>Yellow Taxi Trip Records
Download the file.
Try to open it in EXCEL. For me it opens, but I get an error message.
What is the error message? What is the number of the last row you could load? [1]

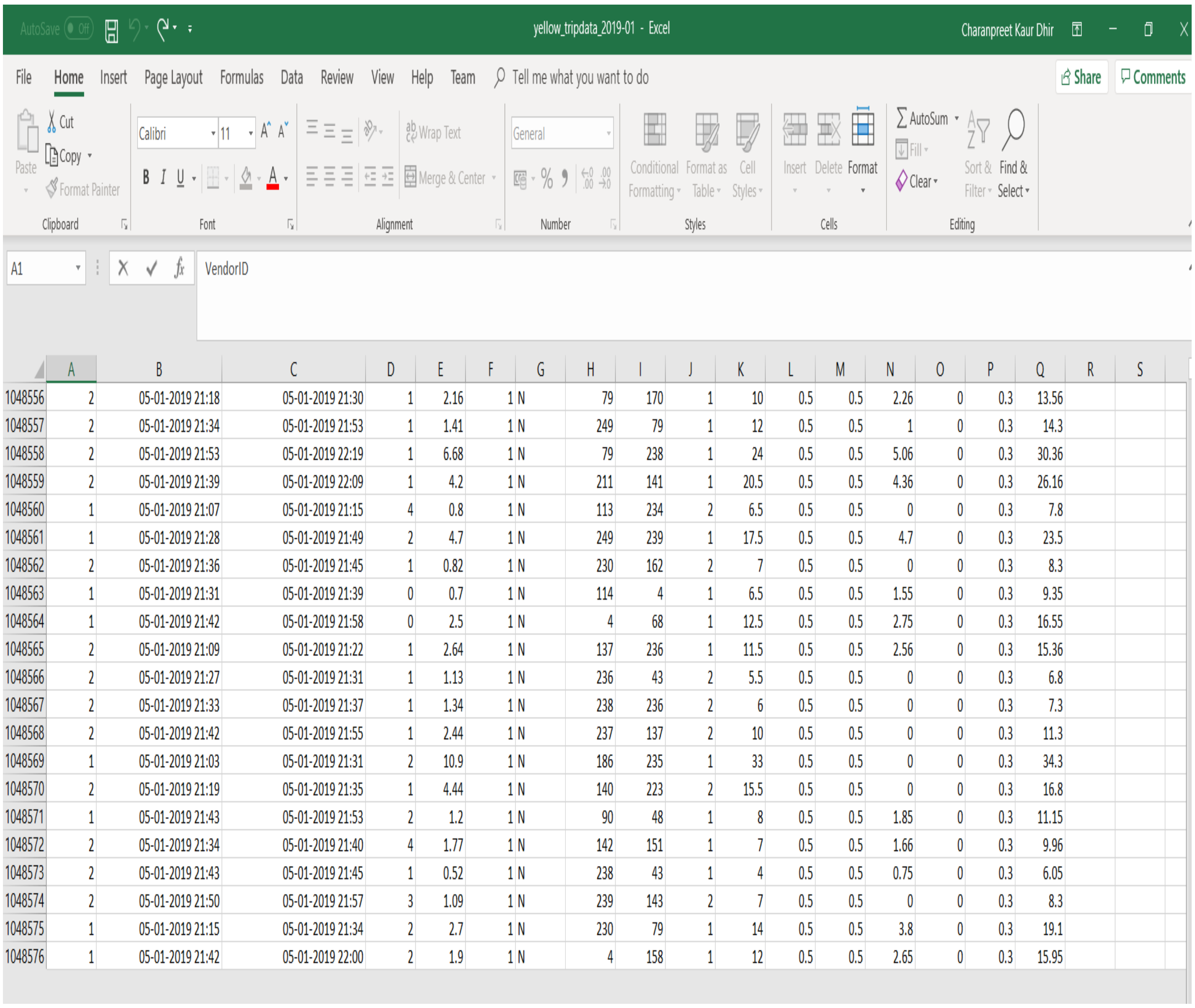
Answer 8 a)

When I opened the file in EXCEL,
The Error message is: **File not loaded completely**


Microsoft Excel

File not loaded completely.

Number of the last row I could load is: **1048576**



A1 VendorID

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1048556	2	05-01-2019 21:18	05-01-2019 21:30	1	2.16	1	N	79	170	1	10	0.5	0.5	2.26	0	0.3	13.56		
1048557	2	05-01-2019 21:34	05-01-2019 21:53	1	1.41	1	N	249	79	1	12	0.5	0.5	1	0	0.3	14.3		
1048558	2	05-01-2019 21:53	05-01-2019 22:19	1	6.68	1	N	79	238	1	24	0.5	0.5	5.06	0	0.3	30.36		
1048559	2	05-01-2019 21:39	05-01-2019 22:09	1	4.2	1	N	211	141	1	20.5	0.5	0.5	4.36	0	0.3	26.16		
1048560	1	05-01-2019 21:07	05-01-2019 21:15	4	0.8	1	N	113	234	2	6.5	0.5	0.5	0	0	0.3	7.8		
1048561	1	05-01-2019 21:28	05-01-2019 21:49	2	4.7	1	N	249	239	1	17.5	0.5	0.5	4.7	0	0.3	23.5		
1048562	2	05-01-2019 21:36	05-01-2019 21:45	1	0.82	1	N	230	162	2	7	0.5	0.5	0	0	0.3	8.3		
1048563	1	05-01-2019 21:31	05-01-2019 21:39	0	0.7	1	N	114	4	1	6.5	0.5	0.5	1.55	0	0.3	9.35		
1048564	1	05-01-2019 21:42	05-01-2019 21:58	0	2.5	1	N	4	68	1	12.5	0.5	0.5	2.75	0	0.3	16.55		
1048565	2	05-01-2019 21:09	05-01-2019 21:22	1	2.64	1	N	137	236	1	11.5	0.5	0.5	2.56	0	0.3	15.36		
1048566	2	05-01-2019 21:27	05-01-2019 21:31	1	1.13	1	N	236	43	2	5.5	0.5	0.5	0	0	0.3	6.8		
1048567	2	05-01-2019 21:33	05-01-2019 21:37	1	1.34	1	N	238	236	2	6	0.5	0.5	0	0	0.3	7.3		
1048568	2	05-01-2019 21:42	05-01-2019 21:55	1	2.44	1	N	237	137	2	10	0.5	0.5	0	0	0.3	11.3		
1048569	1	05-01-2019 21:03	05-01-2019 21:31	2	10.9	1	N	186	235	1	33	0.5	0.5	0	0	0.3	34.3		
1048570	2	05-01-2019 21:19	05-01-2019 21:35	1	4.44	1	N	140	223	2	15.5	0.5	0.5	0	0	0.3	16.8		
1048571	1	05-01-2019 21:43	05-01-2019 21:53	2	1.2	1	N	90	48	1	8	0.5	0.5	1.85	0	0.3	11.15		
1048572	2	05-01-2019 21:34	05-01-2019 21:40	4	1.77	1	N	142	151	1	7	0.5	0.5	1.66	0	0.3	9.96		
1048573	2	05-01-2019 21:43	05-01-2019 21:45	1	0.52	1	N	238	43	1	4	0.5	0.5	0.75	0	0.3	6.05		
1048574	2	05-01-2019 21:50	05-01-2019 21:57	3	1.09	1	N	239	143	2	7	0.5	0.5	0	0	0.3	8.3		
1048575	1	05-01-2019 21:15	05-01-2019 21:34	2	2.7	1	N	230	79	1	14	0.5	0.5	3.8	0	0.3	19.1		
1048576	1	05-01-2019 21:42	05-01-2019 22:00	2	1.9	1	N	4	158	1	12	0.5	0.5	2.65	0	0.3	15.95		

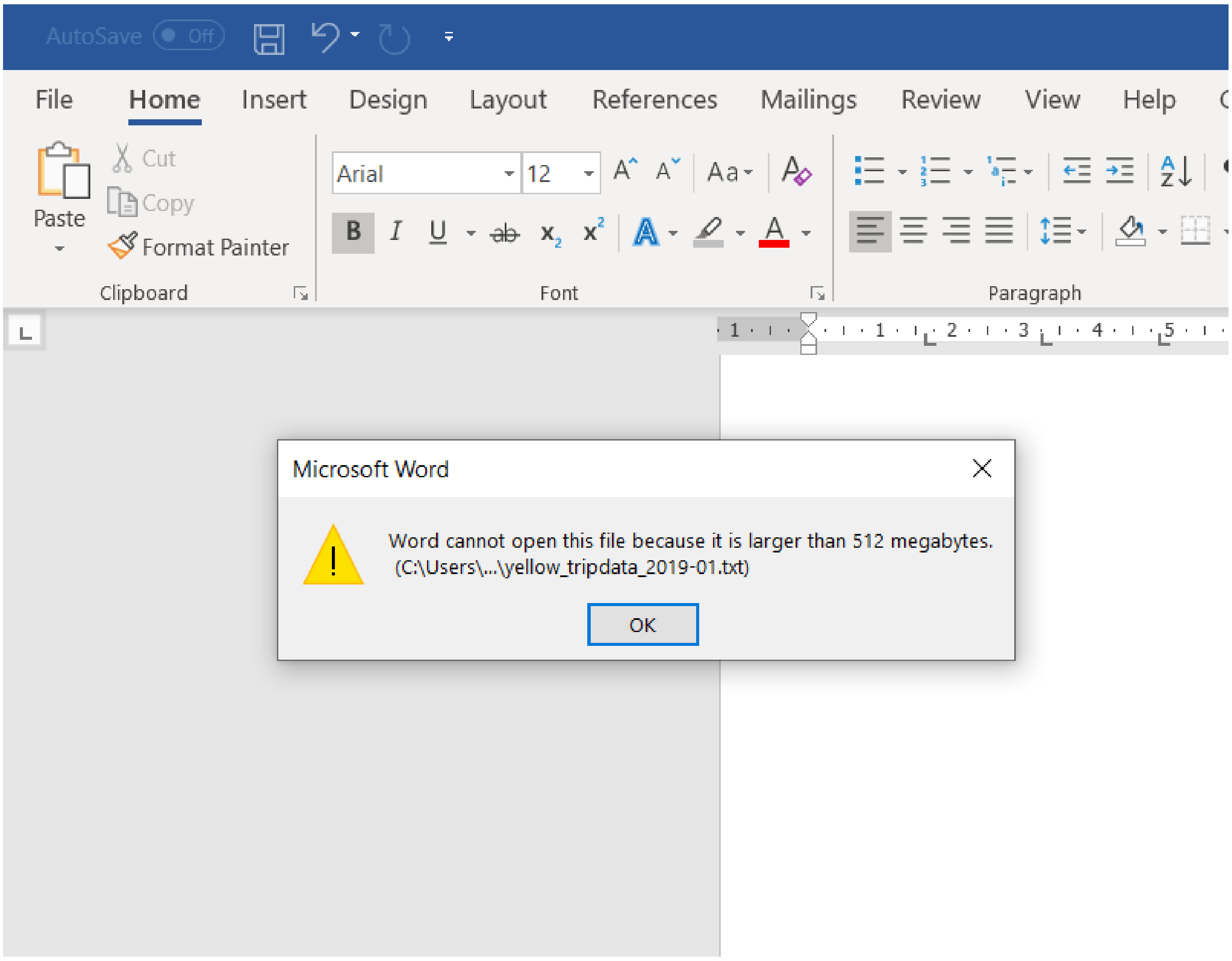
b) Go into file explorer and change the file extension of the downloaded file from .csv to .txt. Try to open this .txt file in MS WORD. What happens? What message do you get?

Answer b)

When I tried to open the file in MS word after changing it to .txt,

Thing that happened next is: **It didn't opened the file.**

Message I received is: **Word cannot open thi file because it is larger than 512 Megabytes.**



Word cannot open this file because it is larger than 512 megabytes.
(C:\Users\...\yellow_tripdata_2019-01.txt)

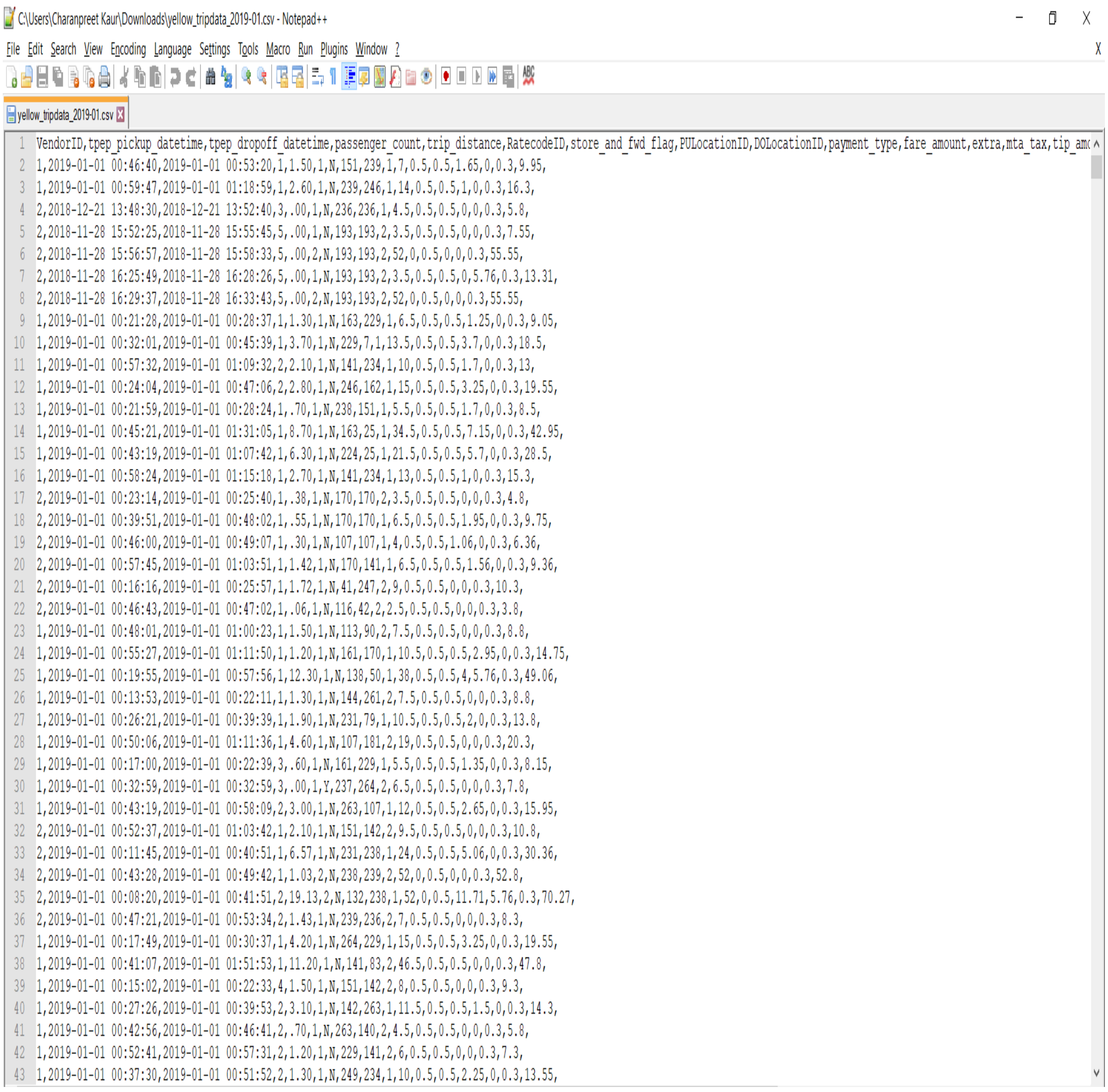
c) Try to open the file with Notepad++.
Does it open? If so, what is the line number of the last row you can see? How does this compare to what you saw in EXCEL?

If you don't have Notepad++ then download it first.

Generally a file like this is NOT considered big data. However, in our class we will consider any file of this size as "big."

Answer c)

When I tried to open the file in Notepad++,
Yes, It opened
Number of the last row I could see is: **7667792 (7667793-1 (As it includes header line also)) in** notepad ++ but in excel I could see only: **1048576**

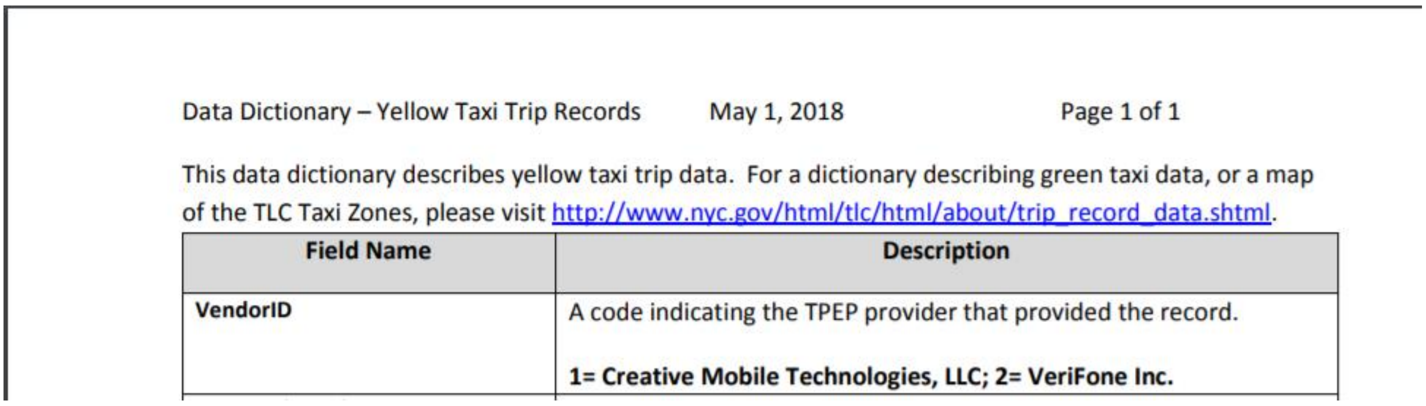

C:\Users\Charanpreet Kaur\Downloads\yellow_tripdata_2019-01.csv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
yellow_tripdata_2019-01.csv

```
VendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,RatecodeID,store_and_fwd_flag,PULocationID,DOLocationID,payment_type,fare_amount,extra,mta_tax,tip_am
1,2019-01-01 00:46:40,2019-01-01 00:53:20,1,1.50,1,N,151,239,1,7,0.5,0.5,1.65,0,0.3,9.95,
1,2019-01-01 00:59:47,2019-01-01 01:18:59,1,2.60,1,N,239,246,1,14,0.5,0.5,1,0,0.3,16.3,
2,2018-12-21 13:48:30,2018-12-21 13:52:40,3,.00,1,N,236,236,1,4.5,0.5,0.5,0,0,0.3,5.8,
2,2018-11-28 15:52:25,2018-11-28 15:55:45,5,.00,1,N,193,193,2,3.5,0.5,0.5,0,0,0.3,7.55,
2,2018-11-28 15:56:57,2018-11-28 15:58:33,5,.00,2,N,193,193,2,52,0,0.5,0,0,0.3,55.55,
2,2018-11-28 16:25:49,2018-11-28 16:28:26,5,.00,1,N,193,193,2,3.5,0.5,0.5,0,5.76,0.3,13.31,
2,2018-11-28 16:29:37,2018-11-28 16:33:43,5,.00,2,N,193,193,2,52,0,0.5,0,0,0.3,55.55,
1,2019-01-01 00:21:28,2019-01-01 00:28:37,1,1.30,1,N,163,229,1,6.5,0.5,0.5,1.25,0,0.3,9.05,
1,2019-01-01 00:32:01,2019-01-01 00:45:39,1,3.70,1,N,229,7,1,13.5,0.5,0.5,3.7,0,0.3,18.5,
1,2019-01-01 00:57:32,2019-01-01 01:09:32,2,2.10,1,N,141,234,1,10,0.5,0.5,1.7,0,0.3,13,
1,2019-01-01 00:24:04,2019-01-01 00:47:06,2,2.80,1,N,246,162,1,15,0.5,0.5,3.25,0,0.3,19.55,
1,2019-01-01 00:21:59,2019-01-01 00:28:24,1,.70,1,N,238,151,1,5.5,0.5,0.5,1.7,0,0.3,8.5,
1,2019-01-01 00:45:21,2019-01-01 01:31:05,1,8.70,1,N,163,25,1,34.5,0.5,0.5,7.15,0,0.3,42.95,
1,2019-01-01 00:43:19,2019-01-01 01:07:42,1,6.30,1,N,224,25,1,21.5,0.5,0.5,5.7,0,0.3,28.5,
1,2019-01-01 00:58:24,2019-01-01 01:15:18,1,2.70,1,N,141,234,1,13,0.5,0.5,1,0,0.3,15.3,
2,2019-01-01 00:23:14,2019-01-01 00:25:40,1,.38,1,N,170,170,2,3.5,0.5,0.5,0,0,0.3,4.8,
2,2019-01-01 00:39:51,2019-01-01 00:48:02,1,.55,1,N,170,170,1,6.5,0.5,0.5,1.95,0,0.3,9.75,
2,2019-01-01 00:46:00,2019-01-01 00:49:07,1,.30,1,N,107,107,1,4,0.5,0.5,1.06,0,0.3,6.36,
2,2019-01-01 00:57:45,2019-01-01 01:03:51,1,1.42,1,N,170,141,1,6.5,0.5,0.5,1.56,0,0.3,9.36,
2,2019-01-01 00:16:16,2019-01-01 00:25:57,1,1.72,1,N,41,247,2,9,0.5,0.5,0,0,0.3,10.3,
2,2019-01-01 00:46:43,2019-01-01 00:47:02,1,.06,1,N,116,42,2,2.5,0.5,0.5,0,0,0.3,3.8,
1,2019-01-01 00:48:01,2019-01-01 01:00:23,1,1.50,1,N,113,90,2,7.5,0.5,0.5,0,0,0.3,8.8,
1,2019-01-01 00:55:27,2019-01-01 01:11:50,1,1.20,1,N,161,170,1,10.5,0.5,0.5,2.95,0,0.3,14.75,
1,2019-01-01 00:19:55,2019-01-01 00:57:56,1,12.30,1,N,138,50,1,38,0.5,0.5,4,5.76,0.3,49.06,
1,2019-01-01 00:13:53,2019-01-01 00:22:11,1,1.30,1,N,144,261,2,7.5,0.5,0.5,0,0,0.3,8.8,
1,2019-01-01 00:26:21,2019-01-01 00:39:39,1,1.90,1,N,231,79,1,10.5,0.5,0.5,2,0,0.3,13.8,
1,2019-01-01 00:50:06,2019-01-01 01:11:36,1,4.60,1,N,107,181,2,19,0.5,0.5,0,0,0.3,20.3,
1,2019-01-01 00:17:00,2019-01-01 00:22:39,3,.60,1,N,161,229,1,5.5,0.5,0.5,1.35,0,0.3,8.15,
1,2019-01-01 00:32:59,2019-01-01 00:32:59,3,.00,1,Y,237,264,2,6.5,0.5,0.5,0,0,0.3,7.8,
1,2019-01-01 00:43:19,2019-01-01 00:58:09,2,3.00,1,N,263,107,1,12,0.5,0.5,2.65,0,0.3,15.95,
2,2019-01-01 00:52:37,2019-01-01 01:03:42,1,2.10,1,N,151,142,2,9.5,0.5,0.5,0,0,0.3,10.8,
2,2019-01-01 00:11:45,2019-01-01 00:40:51,1,6.57,1,N,231,238,1,24,0.5,0.5,5.06,0,0.3,30.36,
2,2019-01-01 00:43:28,2019-01-01 00:49:42,1,1.03,2,N,238,239,2,52,0,0.5,0,0,0.3,52.8,
2,2019-01-01 00:08:20,2019-01-01 00:41:51,2,19.13,2,N,132,238,1,52,0,0.5,11.71,5.76,0.3,70.27,
2,2019-01-01 00:47:21,2019-01-01 00:53:34,2,1.43,1,N,239,236,2,7,0.5,0.5,0,0,0.3,8.3,
1,2019-01-01 00:17:49,2019-01-01 00:30:37,1,4.20,1,N,264,229,1,15,0.5,0.5,3.25,0,0.3,19.55,
1,2019-01-01 00:41:07,2019-01-01 01:51:53,1,11.20,1,N,141,83,2,46.5,0.5,0.5,0,0,0.3,47.8,
1,2019-01-01 00:15:02,2019-01-01 00:22:33,4,1.50,1,N,151,142,2,8,0.5,0.5,0,0,0.3,9.3,
1,2019-01-01 00:27:26,2019-01-01 00:39:53,2,3.10,1,N,142,263,1,11.5,0.5,0.5,1.5,0,0.3,14.3,
1,2019-01-01 00:42:56,2019-01-01 00:46:41,2,.70,1,N,263,140,2,4.5,0.5,0.5,0,0,0.3,5.8,
1,2019-01-01 00:52:41,2019-01-01 00:57:31,2,1.20,1,N,229,141,2,6,0.5,0.5,0,0,0.3,7.3,
1,2019-01-01 00:37:30,2019-01-01 00:51:52,2,1.30,1,N,249,234,1,10,0.5,0.5,2.25,0,0.3,13.55,
```

d) Look at the columns of the data. What values are in the first column? What do these values mean?
HINT: I did not find the answer. I don't know. But maybe you will find it.
REMEMBER WE ARE DEALING WITH REAL DATA. NOT WITH MADE-UP HOMEWORK DATA.

Answer d)

On the same link provided by the professor, I went to the Data Dictionary and Metadata link that was just below the datasets and in that we had a link for yellow trips data dictionary. When I clicked that link it showed me the information about first column of datasets i.e. Vendor ID.
First column either had the values 1 or 2.
Where 1 is for Creative Mobile Technologies
And 2 is Verifone Inc.



Data Dictionary – Yellow Taxi Trip Records May 1, 2018 Page 1 of 1

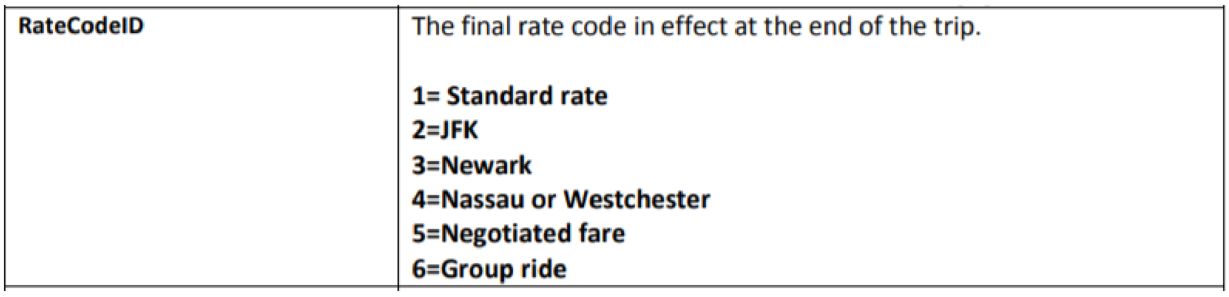
This data dictionary describes yellow taxi trip data. For a dictionary describing green taxi data, or a map of the TLC Taxi Zones, please visit http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

Field Name	Description
VendorID	A code indicating the TPEP provider that provided the record. **1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.**

e) Same question about RatecodeID. What does it mean?
Answer e)

I got the information about RatecodeID in the same way.

Ratecode ID is The final rate code in effect at the end of the trip.. It has value upto 6.



RateCodeID	The final rate code in effect at the end of the trip. **1= Standard rate** **2=JFK** **3=Newark** **4=Nassau or Westchester** **5=Negotiated fare** **6=Group ride**

The information about all the columns are in next page.