LABORATORY REPORT

LABORATORY REPORT

Common Data	
Student name	Das Anjan Kumar
Neptun code	D42DQA
Department	Dept. of Automation and Applied Informatics
Instructor name	AL-MAGSOOSI HUSAM
Laboratory place	IL206
Laboratory time	9th March,10.15am-11.45 pm
Title or Sequence number	2

Exercises	
Task 1	☐
Task 2	☐
Task 3	☐
Task 4	☐
Task 5	☐
Task 6	☐
Task 7	☐
Task 8	☐
Task 9	☐
Task 10	☐

EXERCISES

TASK #1

Problem statement: List the names of all countries in the Middle East, sorted by the 2004 population count in descending order

Solution:

```
select name from country /*only need name column from country table */
where year=2004 and region='Middle East' /*sorting by year and region*/
order by population desc /*as we want descending order by population */
```

	name
1	Iran
2	Turkey
3	Saudi Arabia
4	Iraq
5	Yemen
6	Israel
7	Jordan
8	Georgia
9	Lebanon
10	Oman
11	United Arab Emirates
12	Kuwait
13	Qatar
14	Cyprus
15	Bahrain

Reasoning: As we can see in the code, we have used specific year and region which is required by the problem. Also, we used the descending order as required. And finally, we got the expected result of Middle east countries.

TASK #2

Problem statement: List the names, area and GDP of any European countries with a 2009 population of more than 10,000,000.

Solution:

```
select name, area, gdp   from country  /* choosing name , area and gdp from the
country table*/
where year=2009 and region like '%Europe%' and population>10000000 /* specifying by
tear 2009, and region by %Europe% because there can be many type of european countries
but everyone will include Europe */
```

	name	area	gdp
1	Belgium	30510	0,013
2	Czech Republic	78703	0,039
3	France	547030	0,007
4	Germany	356910	0,013
5	Greece	131940	0,028
6	Italy	301230	-0,007
7	Netherlands	37330	0,018
8	Poland	312680	0,048
9	Portugal	92080	0,002
10	Romania	237500	0,076
11	Russia	17125187	0,054
12	Spain	504750	0,011
13	Ukraine	603700	0,021
14	United Kingdom	244820	0,007

Reasoning:

The main task here actually including all type of European countries, which I have done by including the "`region like '%Europe%'`" ,which lets us to choose all European countries. Also we have specified year and population, as a result we got the expected result.

TASK #3

Problem statement: List the names and regions of countries with an area larger than 2,000,000 but smaller than 5,000,000, sorted in descending order by 2002 GDP.

Solution:

```
select name, region   from country /* selecting only name and region from the country
table */
where   area BETWEEN 2000000 AND 5000000 AND year=2002 /* specifying the range of area
between two numbers*/
order by gdp desc /* descending order by gdp*/
```

	name	region
1	Kazakhstan	Commonwealth of Independent States - Central Asi...
2	Sudan	Africa
3	India	Asia
4	Algeria	Africa
5	Argentina	South America
6	Greenland	Arctic Region
7	Zaire	Africa

Reasoning:

Here, the amin task is to defining the area between two specific number, which I have done using "`area BETWEEN 2000000 AND 5000000`", `also specified the year and descending order of gdp. And we got the expected result.`

TASK #4

Problem statement: List the regions (without duplicates) of all countries whose names start with 'S' (uppercase).

Solution:

```
select distinct region /* using distinct to avoid duplicate*/
from country

where name like 'S%' /* finding similar region of countries who have S% like pattern*/
```

	region
1	Africa
2	Asia
3	Central America and the Caribbean
4	Ethnic Groups in Eastern Europe, Europe
5	Europe
6	Middle East
7	Oceania
8	South America
9	Southeast Asia

Reasoning:

We are here using "distinct" to avoid duplicity and to find regions of all countries whose names start with 'S' we have used "`where name like 'S%'`". That's why we have got the similar expected result.

TASK #5

Problem statement: nsert a new row with country name "SQLvania", with year = 2004, area = 4707 and population = 65550. Check with SELECT

Solution:

Inserting new record:

```
INSERT INTO country (name, year, area, population) VALUES ('SQLvania', '2004', '4707',
'65550') /* first writing the column names and the the values*/
```

Checking:

```
select name, year, region, area, population, gdp from country
where name='SQLvania'
```

	name	year	region	area	population	gdp
1	SQLvania	2004	NULL	4707	65550	NULL

Reasoning:

In this task, mainly we have to insert a new row or record, what we have specified by "INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...) " , where our column names are name, year, area and population.

And after that when we check with select, we can find the expected record from the country table.

TASK #6

Problem statement: Add 15,000 to the 2007 population of all countries with an area less than 10,000. Check with SELECT before and after your changes

Solution:

Before:

```
select name, year, region, area, population, gdp  from country
where year=2007 and area<10000
```

	name	year	region	area	population	gdp
1	American Samoa	2007	Oceania	199	57663	0.03
2	Andorra	2007	Europe	450	71822	0.035
3	Anguilla	2007	Central America and the Caribbean	91	13677	0.102
4	Aruba	2007	Central America and the Caribbean	193	100018	0.024
5	Bahrain	2007	Middle East	620	708573	0.078
6	Barbados	2007	Central America and the Caribbean	430	280946	0.035
7	Bermuda	2007	North America	50	66163	0.046
8	Cape Verde	2007	World	4030	423613	0.055
9	Cayman Islands	2007	Central America and the Caribbean	260	46600	0.009
10	Chad	2007	Africa	1	9885661	0.01
11	Cocos (Keeling) Islands	2007	Southeast Asia	14	596	NULL
12	Comoros	2007	Africa	2170	711417	0.03
13	Cook Islands	2007	Oceania	240	21750	0.001
14	Cyprus	2007	Middle East	9250	788457	0.106
15	Dominica	2007	Central America and the Caribbean	750	72386	0.031
16	Faroe Islands	2007	Europe	1400	47511	0.024
17	French Polynesia	2007	Oceania	3941	278963	0.051

Query executed successfully.

Updating:

```
UPDATE country
SET population=population+15000 /* adding 15000 to the population of the countries
belonging to 2007 year and have area less than 10000*/
where year=2007 and area<10000
```

```
(43 rows affected)

Completion time: 2022-03-14T13:55:09.9146595+01:00
```

After:

```
select name, year, region, area, population, gdp  from country
where year=2007 and area<10000
```

	name	year	region	area	population	gdp
1	American Samoa	2007	Oceania	199	72663	0.03
2	Andorra	2007	Europe	450	86822	0.035
3	Anguilla	2007	Central America and the Caribbean	91	28677	0.102
4	Aruba	2007	Central America and the Caribbean	193	115018	0.024
5	Bahrain	2007	Middle East	620	723573	0.078
6	Barbados	2007	Central America and the Caribbean	430	295946	0.035
7	Bermuda	2007	North America	50	81163	0.046
8	Cape Verde	2007	World	4030	438613	0.055
9	Cayman Islands	2007	Central America and the Caribbean	260	61600	0.009
10	Chad	2007	Africa	1	9900661	0.01
11	Cocos (Keeling) Islands	2007	Southeast Asia	14	15596	NULL
12	Comoros	2007	Africa	2170	726417	0.03
13	Cook Islands	2007	Oceania	240	36750	0.001
14	Cyprus	2007	Middle East	9250	803457	0.106
15	Dominica	2007	Central America and the Caribbean	750	87386	0.031
16	Faroe Islands	2007	Europe	1400	62511	0.024
17	French Polynesia	2007	Oceania	3941	293963	0.051

Query executed successfully.

Reasoning:

First we checked all the countries with the constraint of year 2007 and and having population less than 10,000. Then we updated the population of those record using " `SET population=population+15000`" and as a result we can see that 43 rows were changed.

And, finally we can see the change in the last picture above.

TASK #7

Problem statement: Delete all rows that have a negative GDP.

Solution:

Before:

```
select count (*) as Rows /* checking the total number of rows */
from country
```

	Rows
1	4140

```
select name, gdp from country  /* Checking how many rows are in our constraint*/
where gdp<0
```

	name	gdp
1	Albania	-0.1
2	Albania	-0.08
3	Algeria	-0.018
4	Angola	-0.226
5	Angola	-0.01
6	Angola	-0.009
7	Anguilla	-0.043
8	Anguilla	-0.085
9	Anguilla	-0.085
10	Argentina	-0.044
11	Argentina	-0.03
12	Argentina	-0.046
13	Argentina	-0.109
14	Armenia	-0.34
15	Armenia	-0.099
16	Armenia	-0.02
17	Armenia	-0.142

	name	gdp
510	Zambia	-0.02
511	Zambia	-0.03
512	Zambia	-0.028
513	Zambia	-0.02
514	Zimbab...	-0.1
515	Zimbab...	-0.024
516	Zimbab...	-0.061
517	Zimbab...	-0.065
518	Zimbab...	-0.13
519	Zimbab...	-0.136
520	Zimbab...	-0.082
521	Zimbab...	-0.077
522	Zimbab...	-0.044
523	Zimbab...	-0.06
524	Zimbab...	-0.126
525	Zimbab...	-0.013

We can see that there are 525 rows who have gdp less than zero

Deleting:

```
DELETE FROM country
WHERE gdp<0
```

```
(525 rows affected)

Completion time: 2022-03-14T14:12:29.8772664+01:00
```

After:

	Rows
1	3615

Reasoning:

First we check the total number of total rows. Also we checked the number of records which are in our constraint of gdp less than zero, and the number was 525. After creting the delete command we also saw that 525 rows were affected. So we did that task correctly. Also, check the number of total rows later, which was 3615, and 4140-3615= 525

TASK #8

Problem statement: List a country with the biggest population in 2010. The resulting table should contain name, region, and population.

Solution:

```
SELECT name,region, population /* these three column required by the task*/
FROM country
WHERE population = (SELECT MAX(population) /* using the MAX aggregate function to find
the maximum value from our constraint */
                FROM country
                WHERE year=2010)
```

	name	region	population
1	China	Asia	1330141295

Reasoning: Here we had to use a nested SELECT. This nested select will find the max population from the constraint and return to the main select. We have to follow this method because otherwise all the rows of 2010 having the maximum population of that country at that time will be shown.

TASK #9

Problem statement: List three Asian countries with the smallest total GDP growth for all years. The resulting table should contain name and total GDP

Solution:

```
SELECT TOP 3 Name, sum(gdp) as [Total GDP] from country  /* using TOP 3 to find only
the expeceted three rocords*/
WHERE region='asia' /* specifying region*/
GROUP by name /* using group by because we used an aggregate function, sum*/

order by [Total GDP] asc /* using ascenfing order as we want smallest three*/
```

	Name	Total GDP
1	Japan	0.294000000460073
2	Nepal	0.772000000812113
3	Mongolia	0.879999998956919

Reasoning:

We have used the syntax TOP 3 to see only the three records we want, also used alias to name total sum of gdp for a country as Total GDP. Used specified constraint of region and group by as a consequence of using aggregate function, and finally order by to see in ascending order. As a result, we found our expected result.

TASK #10

Problem statement: List any countries with total GDP growth greater than 1.4. The resulting table should contain name, region and total GDP

Solution:

```
SELECT  Name, Region,sum(gdp) as [Total GDP ]from country
GROUP by name, region

having sum(gdp)>1.4 /* only the countries who have total GDP growth greater than 1.4
*/
```

	Name	Region	Total GDP
1	Angola	Africa	1.42000001110137
2	Equatorial Guinea	Africa	1.92400003038347
3	Bhutan	Asia	1.48999999091029
4	China	Asia	1.84600000083447
5	Maldives	Asia	1.43700000457466
6	Turkmenistan	Commonwealth of Independent States - Central Asia...	1.43900000851136
7	Azerbaijan	Commonwealth of Independent States - European St...	1.95399999897927
8	Kuwait	Middle East	1.83300002105534
9	Malaysia	Southeast Asia	1.42900000652298
10	Singapore	Southeast Asia	1.47900000028312

Reasoning:

We selected name , region from the table, and as we used aggregate function sum , we had to use group by. Renamed sum(gdp) as Total GDP using alias. And finally used having to follow our constraint of the countries who have total GDP growth greater than 1.4. And we found the expected result.

INSTRUCTIONS

1. **Problem statement is mandatory.**
2. **A solution without explanation is NOT accepted.**
3. **If you need to copy the source code, you can do it with copy/paste commands. Please do not use screenshots for code listings.**
4. **Other screenshots (figures, graphs, etc.) should be scaled appropriately. Please cut off unnecessary elements on the images.**