

## EXERCISES

### TASK #1

#### Problem statement:

We need to create a table including all countries in the Middle East from the database. This table should be sorted by the 2004 population count and in descending order.

#### Solution:

By Data Definition Language scripts : we use SELECT command which is one of the most widely used SQL command. We have to choose which column and attribute we want to choose:

\*after **SELECT** we add the column name (in this case it is : name)

\*after **FROM** we add the table name ( in this case it is : country)

\*after **WHERE** we specify which region (middle east) and year (2004) And since we want the data(population) to appear in a descending order we add **order by population desc**.

#### Code:

```
use lab21;
select name from country where (region = 'Middle East' and year =
'2004') order by population desc;

+ CREATE TABLE [dbo].[country](...
```

## Reasoning:

Result :

	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

We have got table with different middle eastern countries ordered by their population.

## TASK #2

### Problem statement:

We need to create a table including the name, area and GDP of only european countries with a 2009 population of more than 10,000,000.

### Solution:

By Data Definition Language scripts : we use SELECT command which is one of the most widely used SQL command. We have to choose which column and attribute we want to choose:

\*after **SELECT** :

-we add the column name (in this case it is : name)

-we add the column name (in this case it is : area)

-we add the column name (in this case it is : gdp)

\*after **FROM** we add the table name ( in this caste it is : country)

\*after **WHERE** we specify which region (europe) and year (2004) and population (more than 10000000

### Code:

```
use lab21;
select name, area, gdp from country where (region like '%Europe%' )
and year = 2009 and population >10000000;
<
+ CREATE TABLE [dbo].[country](...
```

## Reasoning:

## Result:

	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

We have got table with different european countries which everyone has its own area and gdp.

### TASK #3

#### Problem statement:

We need to create a table including the name and region of countries with an area larger than 2,000,000 and smaller than 5,000,000, ordered in descending order by 2002 GDP.

#### Solution:

By Data Definition Language scripts : we use SELECT command which is one of the most widely used SQL command. We have to choose which column and attribute we want to choose:

\*after **SELECT** :

-we add the column name (in this case it is : name)

-we add the column name (in this case it is : region)

\*after **FROM** we add the table name ( in this case it is : country)

\*after **WHERE** we specify which year (2002) and population (between 2000000 and 5000000)

#### Code:

```
use lab21;
select name, region from country where (year =2002) and
area>2000000 and area <5000000 order by gdp desc;

+ CREATE TABLE [dbo].[country](...
```

#### Reasoning:

#### Result:

	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

We have got table with different countries which everyone has its own area and gdp.

### TASK #4

#### Problem statement:

We need to create a table with regions of all countries whose names start with 'S' (uppercase).

#### Solution:

By Data Definition Language scripts :

We use SELECT command and DISTINCT keyword to list only unique values (Region) in a table.

\*after **SELECT distinct** we add the column name (in this case it is : region)

\*after **FROM** we add the table name ( in this caste it is : country)

Then we add **Like operator** to search for a specified pattern(countries names starting with S in a column).

Code:

```
use lab21;  
select distinct region from country where name like 'S%'  
  
+ CREATE TABLE [dbo].[country](...
```

#### Reasoning:

#### Result:

	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

We have got table with different regions that includes countries starting with 'S'

### TASK #5

#### Problem statement:

We need to insert a new row in the table with country name "SQLvania", with year = 2004, area = 4707 and population = 65550.

#### Solution:

By Data Definition Language scripts : we use INSERT INTO statement with specifying both the column names which we want to fill and their corresponding values.

We have to add the table name ( country) which we are going to insert data in.

Then we have to choose which attributes are going to add.

After **values** we add the new data (sqlvania as name , 2004 as year, 4707 as area...).

Since we only need the row that includes informations about 'sqlvania', we mention **SELECT** command.

#### Code:

```
use lab21;
insert into country (name, year, area, population) values
('SQLvania' , 2004, 4707 , 65550);

SELECT * FROM country where name = 'SQLvania';

+ CREATE TABLE [dbo].[country](...
```

#### Reasoning:

#### Result:

	name	year	region	area	population	gdp
1	SQLvania	2004	NULL	4707	6500	NULL

We have got table with one single row(entity) but with different columns values(attributes).

TASK #6

### Problem statement:

We need to update data about the population of some countries with area less than 10000.

### Solution:

We need to use UPDATE command because the population exists already as one of the attributes.

According to 2007 population , we should add 15000 to the countries that has less than 10000.

The new population is the actual population + 15000 : **SET command**

\*after **WHERE** we specify which year (2004).

We should also specify which area (in this case less than 10000).

Code:

```
use lab21;
UPDATE country
SET population=population+15000
WHERE (year=2004);

select name, year , region, area, population, gdp from country where
area <10000

+ CREATE TABLE [dbo].[country](...
```

### Reasoning:

### Result:

	name	year	region	area	population	gdp
33	Andorra	1998	Europe	450	79716	NULL
34	Andorra	1999	Europe	450	80939	NULL
35	Andorra	2000	Europe	450	81824	NULL
36	Andorra	2001	Europe	450	82627	NULL
37	Andorra	2002	Europe	450	83403	0,038
38	Andorra	2003	Europe	450	84150	0,038
39	Andorra	2004	Europe	450	84865	0,038
40	Andorra	2005	Europe	450	85549	0,02
41	Andorra	2006	Europe	450	86201	0,04
42	Andorra	2007	Europe	450	86822	0,035
43	Andorra	2008	Europe	450	86822	0,035

We have got an updated table with new population number ( +15000)



### TASK #7

#### Problem statement:

We need to delete all rows that have a negative GDP. And we have to count the number of rows after deletion.

#### Solution:

We need to use DELETE command to delete rows from the table: those rows with a negative gdp.(gdp<0)

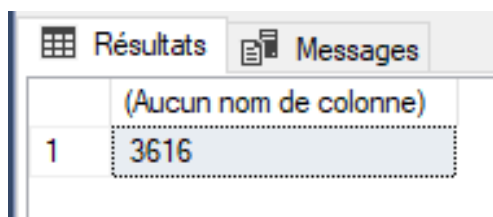
We should add count function to calculate the number of rows left after deletion.

Code:

```
use lab21;  
delete from country where gdp<0  
SELECT count(name)from country  
  
+ CREATE TABLE [dbo].[country](...
```

#### Reasoning:

#### Result:



(Aucun nom de colonne)	
1	3616

\*After the deletion of the negative gdp rows

We have got a table with the number of the rest of rows.

TASK #8

### Problem statement:

We need to create a table country with the biggest population in 2010

### Solution:

We use SELECT and we have to mention 3 columns :name, region and population after the command )

\*after FROM we add the table name ( in this caste it is : country)

\*after WHERE we specify which year (2010) And since we want the data(population) to appear in a descending order we add order by population desc.

Code:

```
use lab21;  
select name,region, population  
from country where ( year = '2010') order by population desc;  
  
+ CREATE TABLE [dbo].[country](...
```

### Reasoning:

### Result:

	name	region	population
1	China	Asia	1330141295
2	India	Asia	1173108018
3	Indonesia	Southeast Asia	242968342
4	Pakistan	Asia	184404791
5	Bangladesh	Asia	156118464
6	Nigeria	Africa	152217341
7	Russia	Europe, Asia	142856536
8	Philippines	Southeast Asia	99900177
9	Ethiopia	Africa	88013491
10	Egypt	Africa	80471869
11	Iran	Middle East	76923300

We have got a table with the highest population according to to the year 2010.

TASK #9

### Problem statement:

We need to list three Asian countries with the smallest total GDP growth for all years.

### Solution:

We need to use SELECT command and the TOP clause to specify the number of records to return.

Using TOP(3) since we need only to get the 1<sup>st</sup> 3 rows from the asian table with the smallest GDP growth for all years.

Also we've used SUM function to get the total gdp in the 2<sup>nd</sup> column of the table.

\*after FROM we add the table name ( in this caste it is : country)

\*after WHERE we specify which region (asia)

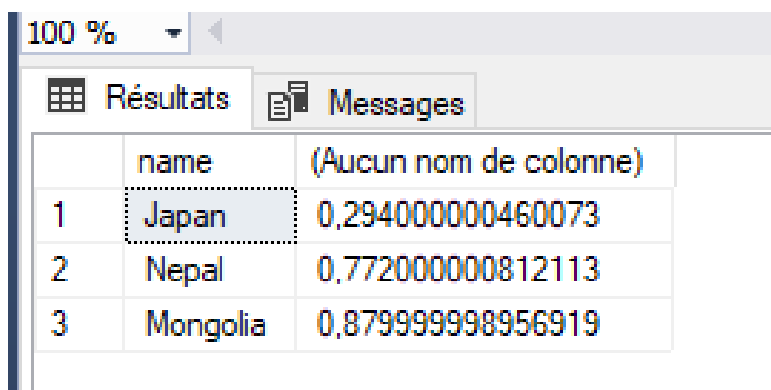
Code:

```
use lab21;
select TOP(3) name, sum(gdp)
from country where (region = 'asia')group by name order by
sum(gdp);

+ CREATE TABLE [dbo].[country](...
```

### Reasoning:

#### Result:



	name	(Aucun nom de colonne)
1	Japan	0,294000000460073
2	Nepal	0,772000000812113
3	Mongolia	0,879999998956919

We have got a table including the 3 asian countries with the smallest total(gdp) growth for all years.

## TASK #10

**Problem statement:**

We need to list the countries with total GDP growth greater than 1.4

**Solution:**

We use SELECT and we have to mention 3 columns :name, region and total(gdp) after the command.

Here we've used **SUM function** to get the total gdp in the 3<sup>rd</sup> column of the table.

Using **group statement** to assemble the result-set by 3 columns.

Also here we've used **the HAVING clause** because the **WHERE** keyword cannot be used with aggregate functions.

```
use lab21;

select name, region, sum(gdp)
from country group by name,region having sum(gdp)>1.4 ;

+ CREATE TABLE [dbo].[country](...
```

**Reasoning:****Result:**

	name	region	(Aucun nom de colonne)
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

We have got a table including the countries with the total(gdp) greater than 1.4.