Using SUM, Count, MAX, DISTINCT and ORDER BY.

Total world population

1.

Show the total **population** of the world.

```
world(name, continent, area, population, gdp)
```

```
SELECT SUM(population)
FROM world
```

Submit SQL restore default

Correct answer

SUM(populatio..
7830500887

List of continents

2.

.ist all the continents - just once each.

```
SELECT DISTINCT continent FROM world
```

Submit SQL restore default

Correct answer

continent
Asia
Europe
Africa
North America
South America
Insular Oceania

GDP of Africa

3.

Give the total GDP of Africa

```
SELECT SUM(gdp) FROM world
WHERE continent = 'Africa'
```

Submit SQL restore default

Correct answer

SUM(gdp)
2994342000000

Count the big countries

4.

How many countries have an **area** of at least 1000000

```
SELECT COUNT(name) FROM world
WHERE area > 1000000
```

Submit SQL restore default

Correct answer

COUNT(name)
28

Baltic states population

5.

What is the total **population** of ('Estonia', 'Latvia', 'Lithuania')

```
SELECT SUM(population) FROM world
where name IN ('Estonia', 'Latvia', 'Lithuania')
```

Submit SQL restore default

Correct answer

SUM(populatio..
6115449

Using GROUP BY and HAVING

You may want to look at these examples: Using GROUP BY and HAVING.

Counting the countries of each continent

6.

For each **continent** show the **continent** and number of countries.

```
SELECT continent, COUNT(name) FROM world
GROUP BY continent
```

Submit SQL restore default

Correct answer

continent	COUNT(name)
Africa	54
Asia	47
Europe	44
Insular Oceania	14
North America	23
South America	12

Counting big countries in each continent

7.

For each **continent** show the **continent** and number of countries with populations of at least 10 million.

```
SELECT continent, COUNT(name) FROM world
WHERE population >= 10000000
GROUP BY continent
```

Submit SQL restore default

Correct answer

continent	COUNT(name)
Africa	32
Asia	28
Europe	16
Insular Oceania	2
North America	7
South America	8

Counting big continents

8.

List the continents that **have** a total population of at least 100 million.

```
SELECT continent FROM world
GROUP BY continent
HAVING SUM(population) >= 100000000
```

Submit SQL restore default

Correct answer

continent
Africa
Asia
Europe
North America
South America

Summary

1.

How many **stops** are in the database.

```
SELECT COUNT(stops.id) FROM stops
```

Submit SQL restore default

Result:

num	company	pos	stop
124	SMT	9	1
100	MAC	6	2
106	SMT	8	3
29	LOW	8	4
16	SMT	8	5
142	SMJ	1	6
63	LRT	1	6

2.

Find the **id** value for the stop 'Craiglockhart'

```
SELECT id FROM stops
WHERE name = 'Craiglockhart'
```

Submit SQL restore default

Correct answer

id
53

3.

Give the **id** and the **name** for the **stops** on the '4' 'LRT' service.

```
SELECT id, name FROM stops
JOIN route ON id = stop
WHERE num = '4' AND company = 'LRT';
```

Submit SQL restore default

Correct answer

id	name
19	Bingham
177	Northfield
149	London Road
194	Princes Street
115	Haymarket
53	Craiglockhart
179	Oxgangs

Routes and stops

4.

The query shown gives the number of routes that visit either London Road (149) or Craiglockhart (53). Run the query and notice the two services that link these **stops** have a count of 2. Add a HAVING clause to restrict the output to these two routes.

```
SELECT company, num, COUNT(*)
FROM route WHERE stop=149 OR stop=53
GROUP BY company, num
HAVING COUNT(*) >= 2
```

Submit SQL restore default

Correct answer

company	num	COUNT(*)
LRT	4	2
LRT	45	2

5.

Execute the self join shown and observe that b.stop gives all the places you can get to from Craiglockhart, without changing routes. Change the query so that it shows the services from Craiglockhart to London Road.

```
SELECT a.company, a.num, a.stop, b.stop
FROM route a JOIN route b ON
  (a.company=b.company AND a.num=b.num)
WHERE a.stop=53 AND b.stop = 149
```

Submit SQL restore default

Correct answer

company	num	stop	stop
LRT	4	53	149
LRT	45	53	149

6.

The query shown is similar to the previous one, however by joining two copies of the **stops** table we can refer to **stops** by **name** rather than by number. Change the query so that the services between 'Craiglockhart' and 'London Road' are shown. If you are tired of these places try 'Fairmilehead' against 'Tollcross'

```
SELECT a.company, a.num, stopa.name, stopb.name
FROM route a JOIN route b ON
    (a.company=b.company AND a.num=b.num)
    JOIN stops stopa ON (a.stop=stopa.id)
    JOIN stops stopb ON (b.stop=stopb.id)
WHERE stopa.name='Craiglockhart' AND stopb.name = 'London Road'
```

Correct answer

company	num	name	name
LRT	4	Craiglockhart	London Road
LRT	45	Craiglockhart	London Road

Using a self join

7.

Give a list of all the services which connect stops 115 and 137 ('Haymarket' and 'Leith')

```
SELECT DISTINCT a.company, a.num FROM route a
JOIN route b ON a.num = b.num
WHERE a.stop = 115 AND b.stop = 137
```

Submit SQL restore default

Correct answer

company	num
LRT	12
LRT	2
LRT	22
LRT	25
LRT	2A
SMT	C5

8.

Give a list of the services which connect the **stops** 'Craiglockhart' and 'Tollcross'

```
SELECT a.company, a.num FROM route a
JOIN route b ON (a.num = b.num)
JOIN stops stopa ON (a.stop = stopa.id)
JOIN stops stopb ON (b.stop = stopb.id)
WHERE stopa.name = 'Craiglockhart' AND stopb.name = 'Tollcross'
```

Submit SQL restore default

Correct answer

company	num
LRT	10
LRT	27
LRT	45
LRT	47

9. 

Give a distinct list of the **stops** which may be reached from 'Craiglockhart' by taking one bus, including 'Craiglockhart' itself, offered by the LRT company. Include the company and bus no. of the relevant services.

```
SELECT DISTINCT stopb.name, b.company, b.num FROM route a
JOIN route b ON (a.num = b.num AND a.company = b.company)
JOIN stops stopa ON (a.stop = stopa.id)
JOIN stops stopb ON (b.stop = stopb.id)
WHERE stopa.name = 'Craiglockhart'
```

Submit SQL restore default

Correct answer

name	company	num
Silverknowes	LRT	10
Muirhouse	LRT	10
Newhaven	LRT	10
Leith	LRT	10
Leith Walk	LRT	10
Princes Street	LRT	10
Tollcross	LRT	10