Self join

Edinburgh Buses

Details of the database Looking at the data

```
stops(id, name)
route(num, company, pos, stop)
```

stops
id
name

route
num
company
pos
stop

Summary

1.

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

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

Submit SQL Restore default

result

2.

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

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

Submit SQL Restore default

result

3.

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

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

Submit SQL Restore default

result

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(*)>1
```

Submit SQL Restore default

result

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

result

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'
```

Submit SQL Restore default

result

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.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='Haymarket' AND stopb.name='Leith'
```

Submit SQL Restore default

result

8.

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

```
SELECT DISTINCT a.company, a.num
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='Tollcross'
```

Submit SQL Restore default

result

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 stopa.name,a.company, a.num
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 stopb.name='Craiglockhart'
```

Submit SQL Restore default

result

10.

Find the routes involving two buses that can go from **Craiglockhart** to **Lochend**.
Show the bus no. and company for the first bus, the name of the stop for the transfer,
and the bus no. and company for the second bus.

Hint

```
SELECT A.num, A.company, A.name, B.num, B.company
FROM
(SELECT a.num, a.company, 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)
```

Submit SQL Restore default

result

Clear your results

Self join Quiz