

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

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(*) = 2;
```

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

#### 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'
```

Submit SQL

Restore default

#### 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.company=b.company AND 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 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 stopa.id=a.stop
JOIN stops stopb ON stopb.id=b.stop
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 DISTINCT stopb.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 stopa.id = a.stop
JOIN stops stopb ON stopb.id = b.stop
WHERE stopa.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, trans1.name , c.num, c.company
FROM route a JOIN route b ON (a.company = b.company AND a.num = b.num)
JOIN ( route c JOIN route d ON (c.company = d.company AND c.num= d.num))
ON stops start ON (a.stop = start.id)
JOIN stops trans1 ON (b.stop = trans1.id)
JOIN stops trans2 ON (c.stop = trans2.id)
```

Submit SQL

Restore default

result

Clear your results  
Self join Quiz