

# Assignment 3 - SQL Zoo - Self join

☰ Tags

## 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

1.

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

```
select count(*)
from stops;
```

## 2.

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

```
select id
from stops
where name = 'Craiglockhart';
```

## 3.

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

```
select s.id, s.name
from stops s
join route r on r.stop = s.id
where r.num = '4' and r.company = 'LRT';
```

# 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;
```

## 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)
JOIN stops s on s.id = b.stop
WHERE s.name = 'London Road' and a.stop = 53;
```

## 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';
```

## 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';
```

## 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';
```

## 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
    (a.stop=stopa.id)
JOIN stops stopb ON
    (b.stop=stopb.id)
WHERE stopa.name='Craiglockhart';

```

## 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.

Self-join twice to find buses that visit Craiglockhart and Lochend, then join those on matching stops.

```

SELECT a.num, a.company, stopc.name, c.num, c.company
FROM route a
JOIN stops stopa ON stopa.id = a.stop and stopa.name='Craiglockhart'
JOIN route b ON b.company=a.company AND b.num=a.num
JOIN stops stopb ON stopb.id = b.stop
JOIN route c ON c.stop = b.stop
JOIN stops stopc ON stopc.id = c.stop
JOIN route d ON d.company=c.company AND d.num=c.num
JOIN stops stopd ON stopd.id = d.stop and stopd.name='Lochend'
group by 1, 2, 3, 4, 5

```