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

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

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

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

Routes and stops

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.

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.

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'

Using a self join

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

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

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.

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

Clear your results

Self join Quiz

Category: Pages using deprecated source tags