7 More JOIN

1. List the films where the yr is 1962 [Show id, title]

SELECT id, title

FROM movie

WHERE yr=1962

1. Give year of 'Citizen Kane'.

SELECT yr

FROM movie

WHERE title='Citizen Kane'

1. List all of the Star Trek movies, include the id, title and yr (all of these movies include the words Star Trek in the title). Order results by year.

SELECT id, title, yr

FROM movie

WHERE title LIKE '%Star Trek%'

1. What id number does the actor 'Glenn Close' have?

SELECT id

FROM actor

WHERE name='Glenn Close'

1. What is the **id** of the film 'Casablanca'

SELECT id

FROM movie

WHERE title='Casablanca'

1. Obtain the cast list for 'Casablanca'.

what is a cast list?

The cast list is the names of the actors who were in the movie.

SELECT name

FROM casting JOIN actor ON actorid=id

WHERE movieid=27

1. Obtain the cast list for the film 'Alien'

SELECT name

FROM ((casting JOIN actor ON actorid = actor.id)

JOIN movie ON movieid = movie.id)

WHERE title='Alien'

1. List the films in which 'Harrison Ford' has appeared

SELECT title

FROM ((casting JOIN actor ON actorid = actor.id)

JOIN movie ON movieid = movie.id)

WHERE name='Harrison Ford'

1. List the films where 'Harrison Ford' has appeared - but not in the starring role. [Note: the ord field of casting gives the position of the actor. If ord=1 then this actor is in the starring role]

SELECT title

FROM ((casting JOIN actor ON actorid = actor.id)

JOIN movie ON movieid = movie.id)

WHERE name='Harrison Ford' and ord<>1

1. List the films together with the leading star for all 1962 films.

SELECT title, name

FROM ((casting JOIN actor ON actorid = actor.id)

JOIN movie ON movieid = movie.id)

WHERE ord=1 and yr=1962

1. Which were the busiest years for 'Rock Hudson', show the year and the number of movies he made each year for any year in which he made more than 2 movies.

SELECT yr,COUNT(title) FROM

movie JOIN casting ON movie.id=movieid

JOIN actor ON actorid=actor.id

WHERE name='Rock Hudson'

GROUP BY yr

HAVING COUNT(title) > 1

1. List the film title and the leading actor for all of the films 'Julie Andrews' played in.

SELECT title, name

FROM movie JOIN casting ON movie.id=movieid AND ord=1

JOIN actor ON actorid=actor.id

WHERE movie.id IN (SELECT movieid

FROM casting

WHERE actorid IN (SELECT id

FROM actor

where name='Julie Andrews'))

1. Obtain a list, in alphabetical order, of actors who've had at least 15 starring roles.

SELECT name

FROM movie JOIN casting ON movie.id=movieid AND ord=1

JOIN actor ON actorid=actor.id

GROUP BY name

HAVING COUNT(movieid)>14

1. List the films released in the year 1978 ordered by the number of actors in the cast, then by title.

SELECT title, COUNT(actorid)

FROM movie JOIN casting ON movie.id=movieid

JOIN actor ON actorid=actor.id

WHERE yr = 1978

GROUP BY title

ORDER BY COUNT(name) DESC, title

1. List all the people who have worked with 'Art Garfunkel'.

SELECT name

FROM movie JOIN casting ON movie.id=movieid

JOIN actor ON actorid=actor.id

WHERE title in (SELECT title

FROM movie JOIN casting ON movie.id=movieid

JOIN actor ON actorid=actor.id

WHERE name='Art Garfunkel') AND name<>'Art Garfunkel'

GROUP BY name

8 Using NULL

1. List the teachers who have NULL for their department.

Why we cannot use =

You might think that the phrase dept=NULL would work here but it doesn't - you can use the phrase dept IS NULL

SELECT name

FROM teacher

WHERE dept IS NULL

1. Note the INNER JOIN misses the teachers with no department and the departments with no teacher.

SELECT teacher.name, dept.name

FROM teacher INNER JOIN dept

ON (teacher.dept=dept.id)

1. Use a different JOIN so that all teachers are listed.

SELECT teacher.name, dept.name

FROM teacher LEFT JOIN dept

ON (teacher.dept=dept.id)

1. Use a different JOIN so that all departments are listed.

SELECT teacher.name, dept.name

FROM teacher RIGHT JOIN dept

ON (teacher.dept=dept.id)

1. Use COALESCE to print the mobile number. Use the number '07986 444 2266' if there is no number given. Show teacher name and mobile number or '07986 444 2266'

SELECT teacher.name, COALESCE(teacher.mobile,'07986 444 2266')

FROM teacher

1. Use the COALESCE function and a LEFT JOIN to print the teacher name and department name. Use the string 'None' where there is no department.

SELECT teacher.name, COALESCE(dept.name,'None')

FROM teacher LEFT JOIN dept

ON (teacher.dept=dept.id)

1. Use COUNT to show the number of teachers and the number of mobile phones.

SELECT COUNT(teacher.name), COUNT(teacher.mobile)

FROM teacher

1. Use COUNT and GROUP BY dept.name to show each department and the number of staff. Use a RIGHT JOIN to ensure that the Engineering department is listed.

SELECT dept.name, COUNT(teacher.name)

FROM teacher RIGHT JOIN dept

ON (teacher.dept=dept.id)

GROUP BY dept.name

1. Use CASE to show the name of each teacher followed by 'Sci' if the teacher is in dept 1 or 2 and 'Art' otherwise.

SELECT name, CASE WHEN (dept=1 OR dept=2) THEN 'Sci' ELSE 'Art' END

FROM teacher

1. Use CASE to show the name of each teacher followed by 'Sci' if the teacher is in dept 1 or 2, show 'Art' if the teacher's dept is 3 and 'None' otherwise.

SELECT name, CASE WHEN (dept=1 OR dept=2) THEN 'Sci'

WHEN dept=3 THEN 'Art'

ELSE 'None' END

FROM teacher

9 Self JOIN

1. How many stops are in the database.

SELECT COUNT(id)

FROM stops

1. Find the id value for the stop 'Craiglockhart'

SELECT id

FROM stops

WHERE name='Craiglockhart'

1. 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 company='LRT' AND num='4'

1. 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

1. 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

1. 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'

1. 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

1. 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.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'

1. 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 stopb.name, b.company, b.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'

1. 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.

SELECT DISTINCT start.num, start.company, stops.name, finish.num, finish.company

FROM

(SELECT a.company, a.num, b.stop

FROM route a JOIN route b ON (a.company=b.company AND a.num=b.num)

WHERE a.stop=(SELECT id FROM stops WHERE name= 'Craiglockhart')

)start

JOIN

(SELECT a.company, a.num, b.stop

FROM route a JOIN route b ON (a.company=b.company AND a.num=b.num)

WHERE a.stop=(SELECT id FROM stops WHERE name= 'Lochend')

)finish

ON (start.stop = finish.stop)

JOIN stops ON(stops.id = start.stop)

8+ NSS Tutorial

1. The example shows the number who responded for:

question 1

at 'Edinburgh Napier University'

studying '(8) Computer Science'

Show the the percentage who STRONGLY AGREE

SELECT A\_STRONGLY\_AGREE

FROM nss

WHERE question='Q01'

AND institution='Edinburgh Napier University'

AND subject='(8) Computer Science'

1. Show the institution and subject where the score is at least 100 for question 15.

SELECT institution, subject

FROM nss

WHERE score>=100 AND question='Q15'

1. Show the institution and score where the score for '(8) Computer Science' is less than 50 for question 'Q15'

SELECT institution,score

FROM nss

WHERE question='Q15'

AND subject='(8) Computer Science'

AND score<50

1. Show the subject and total number of students who responded to question 22 for each of the subjects '(8) Computer Science' and '(H) Creative Arts and Design'.

SELECT DISTINCT subject, SUM(response)

FROM nss

WHERE question='Q22'

AND subject IN('(8) Computer Science','(H) Creative Arts and Design')

GROUP BY subject

1. Show the subject and total number of students who A\_STRONGLY\_AGREE to question 22 for each of the subjects '(8) Computer Science' and '(H) Creative Arts and Design'.

SELECT subject, SUM(A\_STRONGLY\_AGREE\*response/100)

FROM nss

WHERE question='Q22'

AND subject IN('(8) Computer Science','(H) Creative Arts and Design')

GROUP BY subject

1. Show the percentage of students who A\_STRONGLY\_AGREE to question 22 for the subject '(8) Computer Science' show the same figure for the subject '(H) Creative Arts and Design'.

Use the ROUND function to show the percentage without decimal places.

SELECT subject, SUM(A\_STRONGLY\_AGREE/response)

FROM nss

WHERE question='Q22'

AND subject IN('(8) Computer Science','(H) Creative Arts and Design')

GROUP BY subject

1. Show the average scores for question 'Q22' for each institution that include 'Manchester' in the name.

The column score is a percentage - you must use the method outlined above to multiply the percentage by the response and divide by the total response. Give your answer rounded to the nearest whole number.

SELECT institution, ROUND(SUM(score)/COUNT(institution),0) as score

FROM nss

WHERE question='Q22'

AND (institution LIKE '%Manchester%')

GROUP BY institution

ORDER BY institution

1. Show the institution, the total sample size and the number of computing students for institutions in Manchester for 'Q01'.

SELECT institution, SUM(sample), COUNT(ukprn) as comp

FROM nss

WHERE question='Q01'

AND (institution LIKE '%Manchester%')

GROUP BY institution