SELECT name, population FROM world (select \*代表所有)

SELECT DISTINCT 選出不重複的

WHERE name = 'Germany'

WHERE name IN ('Sweden','Norway','Denmark');  
 WHERE area BETWEEN 200000 AND 250000

WHERE name LIKE "Al%"

WHERE name LIKE '%a' OR name LIKE '%l'

特別用法

WHERE winner like 'PETER GR%NBERG' < 有特殊字

WHERE winner like 'EUGENE O' 'NEILL < 標點符號

**SELECT** name,**length**(name)**, Count(XXX) as YYY** (select 可以自由添加ex.area\*2) YYY為新名稱

**WHERE** **length**(name)=5 **and** region='Europe'

AND & OR

where (area > 3000000 and population < 250000000)

or (area < 3000000 and population > 250000000)

WHERE (subject = 'Physics' and yr = 1980) or (subject = 'Chemistry' and yr = '1984')

ROUND(xxx , 2) ROUND(xxx , -3)

LEFT & XOR

WHERE left(name,1) = left(capital,1) XOR name = capital

同時包含aeiou又不包含空白

WHERE name like '%a%' and name like '%e%' and name like '%i%'

and name like '%o%' and name like '%u%' and name not like '% %'

order by

order by yr DESC, winner

group by 分組

having 在 WHERE 关键字无法与合计函数一起使用，所以用having

SELECT continent FROM world

GROUP BY continent

HAVING sum(population) >= 100000000

符合條件為1，其餘為0 (可使符合條件至底)

ORDER BY subject in ('Chemistry','Physics'),subject, winner

CONCAT (‘a’,’b’) -> ab

ALL : 條件下所有的 ; ANY : 任意一個

MAX, DISTINCT, SUM, COUNT, MIN

COUNT(subject) gives 4; COUNT(DISTINCT subject) gives 3.

COUNT(\*) 函数返回在给定的选择中被选的列数。

難題

SELECT name, concat(round(population/(SELECT population FROM world WHERE name = 'Germany')\*100),'%')

FROM world

WHERE continent = 'Europe'

新用法 兩個表

SELECT continent, name, area FROM world x

WHERE area >= ALL(SELECT area FROM world y

WHERE x.continent = y.continent AND y.area > 0)

Name 也可以比大小 abc < acd

JOIN

FROM game JOIN goal ON (game.id = goal.matchid)

INNER JOIN = JOIN

LEFT JOIN 按照左邊的表來做匹配

RIGHT JOIN 按照右邊的表來做匹配

三張表的JOIN

SELECT actor.name

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

JOIN movie ON (casting.movieid = movie.id)

WHERE movie.title = 'Alien'

CASE WHEN

CASE WHEN condition1 THEN value1

WHEN condition2 THEN value2

ELSE def\_value

AS XXX

END

SELECT first.num, first.company, first.name, second.num, second.company

FROM

(SELECT bb.name, a.num, a.company

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

JOIN stops aa ON (a.stop = aa.id)

JOIN stops bb ON (b.stop = bb.id)

WHERE aa.name = 'Craiglockhart' AND bb.name != 'Lochend') as first

JOIN

(SELECT cc.name, c.num, c.company

FROM route c JOIN route d ON (c.company = d.company AND c.num = d.num)

JOIN stops cc ON (c.stop = cc.id)

JOIN stops dd ON (d.stop = dd.id)

WHERE cc.name != 'Craiglockhart' AND dd.name = 'Lochend') as second

ON (first.name = second.name)

ORDER BY first.num, first.company, first.name, second.num, second.company

IS NULL , IS NOT NULL

ISNULL(XXX, YYY) = COALESCE(XXX,YYY ) 如果XXX是NULL回傳YYY

PRIMARY KEY 约束唯一标识数据库表中的每条记录。

主键必须包含唯一的值。

主键列不能包含 NULL 值。

每个表都应该有一个主键，并且每个表只能有一个主键。

主鍵可以包含一或多個欄位。當主鍵包含多個欄位時，稱為組合鍵 (Composite Key)。

好例子 SELF JOIN



