[sqlzoo]

    https://github.com/codyloyd/sqlzoo-solutions/blob/master/SQLZOO\_solutions.md

    https://sqlzoo.net/wiki/Using\_Null

Select basics

15.

"Monaco-Ville"是合併國家名字 "Monaco" 和延伸詞"-Ville".

顯示國家名字，及其延伸詞，如首都是國家名字的延伸。

select substr(capital,1,length(name)),substr(capital,length(name)+1)

from world

where capital like concat(name,'\_','%')

知识点：substr

8.

美國、印度和中國(USA, India, China)是人口又大，同時面積又大的國家。排除這些國家。

顯示以人口或面積為大國的國家，但不能同時兩者。顯示國家名稱，人口和面積。

select name, population, area

from world

where population > 250000000 xor area > 3000000

知识点：xor

10.

顯示國家有至少一個萬億元國內生產總值（萬億，也就是12個零）的人均國內生產總值。四捨五入這個值到最接近1000。

顯示萬億元國家的人均國內生產總值，四捨五入到最近的$ 1000。

select name,round(gdp/population,-3)

from world

where gdp>1000000000000

知识点：round

13.

Oceania becomes Australasia

Countries in Eurasia and Turkey go to Europe/Asia

Caribbean islands starting with 'B' go to North America, other Caribbean islands go to South America

Show the name, the original continent and the new continent of all countries.

SELECT name, continent, CASE

    WHEN continent = 'Oceania' THEN 'Australasia'

    WHEN continent = 'Eurasia' THEN 'Europe/Asia'

    WHEN name = 'Turkey' THEN 'Europe/Asia'

    WHEN continent = 'Caribbean' AND name LIKE 'B%' then 'North America'

    WHEN continent = 'Caribbean' THEN 'South America'

    ELSE continent END

FROM world

知识点：case when

SELECT within SELECT Tutorial/zh

all:

通过all进行最值的筛选

example:

查詢找到世界上最大的國家(以人口計算)

SELECT name

FROM world

WHERE population >= ALL(SELECT population

    FROM world

    WHERE population>0

    )

null:

和null值进行比较，所有结果都为null

1>2——0

1<2——1

1<null or 1>null or 1=null——0

7. 在每一個州中找出最大面積的國家，列出洲份 continent, 國家名字 name 及面積 area。 (有些國家的記錄中，AREA是NULL，沒有填入資料的。)

SELECT continent, name, area FROM world x

WHERE area >= ALL

    (SELECT area FROM world y

        WHERE y.continent=x.continent

        AND area>0)

知识点: where y.continent=x.continent

        为表命名

8. 列出洲份名稱，和每個洲份中國家名字按子母順序是排首位的國家名。(即每洲只有列一國)

select distinct continent,(

    select name

    from world as b

    where a.continent=b.continent

    order by name ASC limit 1

)

from world as a

知识点：distinct去重

10. 有些國家的人口是同洲份的所有其他國的3倍或以上。列出 國家名字name 和 洲份 continent

select name,continent

from world as a

where population>=all(

    select population\*3

    from world as b

    where a.continent=b.continent

    and population>0

    and a.name<>b.name

    )

知识点：不可以直接3\*all()，只能通过修改all(select m\*3)的形式达到目的

The nobel table can be used to practice more SUM and COUNT functions./zh

5. 對每一個獎項(Subject),列出首次頒發的年份。

select subject,min(yr)

from nobel

group by subject

知识点：一对多的时候(subject对yr)，通过限定词是多的一方数量减少为1

7. 對每一個獎項(Subject),列出有多少個不同的得獎者。

select subject,count(distinct winner)

from nobel

group by subject

知识点：distinct

如果题目中有特定的"不同的"，则需要用distinct进行限制

11. 列出誰獲得多於一個獎項(Subject)

select winner

from nobel

group by winner

having count(distinct subject)>1

知识点: distinct

五、The JOIN operation/zh

11. 每一場波蘭'POL'有參與的賽事中，列出賽事編號 matchid, 日期date 和入球數字。

SELECT id,mdate,count(\*)

FROM game as ga

JOIN goal as g

ON g.matchid = ga.id

WHERE (team1 = 'POL' OR team2 = 'POL')

group by id,mdate

知识点：group by

如果两个相同列绑定(id,mdate)

进行分组的时候需要将两个相同列都写上

ex: group by id,mdate 而不是 group by id/mdate

13. List every match with the goals scored by each team as shown. This will use "CASE WHEN" which has not been explained in any previous exercises.

    Notice in the query given every goal is listed. If it was a team1 goal then a 1 appears in score1, otherwise there is a 0. You could SUM this column to get a count of the goals scored by team1. Sort your result by mdate, matchid, team1 and team2.

select distinct mdate,team1,

sum(CASE WHEN teamid=team1 THEN 1 ELSE 0 END) score1,

team2,

sum(CASE WHEN teamid=team2 THEN 1 ELSE 0 END) score2

from game as ga

join goal as g

on ga.id=g.matchid

group by mdate,team1,team2

order by mdate,matchid,team1,team2

知识点：内连接inner join 与外连接left/right join

More JOIN operations/zh

10. 列出演員夏里遜福 'Harrison Ford' 曾演出的電影，但他不是第1主角。

多表连接：

SELECT title FROM casting

JOIN movie ON (movie.id = movieid)

JOIN actor ON (actor.id = actorid)

WHERE name = 'Harrison Ford'  AND ord > 1

子查询：

select title

from movie

where id in (

select movieid

from casting

where actorid=(

select id

from actor

where name='Harrison Ford') and ord <>1)

More JOIN operations/zh

13. 列出演員茱莉·安德絲'Julie Andrews'曾參與的電影名稱及其第1主角。

select title,name

from movie

join casting on movieid=movie.id

join actor on actor.id=actorid

where ord=1 and movie.id in(

select movie.id

from movie

join casting on movieid=movie.id

join actor on actor.id=actorid

where name='Julie Andrews'

)

知识点：使用名称查询可能会出现同名的情况，需要考虑

错误版本：

select title,name

from movie

join casting on movieid=movie.id

join actor on actor.id=actorid

where ord=1 and title in(

select title

from movie

join casting on movieid=movie.id

join actor on actor.id=actorid

where name='Julie Andrews'

)

join quiz 2

知识点：order

order by 2=order by 第二列

Using Null

5. 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 name,

coalesce(mobile, '07986 444 2266')

from teacher

知识点：coalesce

coalesce(mobile,a)

如果mobile is null则返回a，不为null则返回正常值

Self join

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

select distinct r1.company,r1.num

from route as r1

join route as r2

on r1.num=r2.num

where r1.stop=115 and r2.stop=137

知识点：distinct

distinct 只能放在句子开头，不能放在第二个显示段落之前

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 name, a.company, a.num

FROM route a

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

JOIN stops ON a.stop = stops.id

WHERE b.stop = 53;

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.

SELECT a.num, a.company, stopb.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))

JOIN stops stopa ON a.stop = stopa.id

JOIN stops stopb ON b.stop = stopb.id

JOIN stops stopc ON c.stop = stopc.id

JOIN stops stopd ON d.stop = stopd.id

WHERE stopa.name = 'Craiglockhart'

    AND stopd.name = 'Sighthill'

    AND stopb.name = stopc.name

ORDER BY LENGTH(a.num), b.num, stopb.name, LENGTH(c.num), d.num;

xuesql.cn

1. 按电影名字母顺序升序排列，列出排名第6-10 的电影

select \*

from movies

order by title ASC

limit 5,5

知识点：limit 1-排名第一的数据,limit 2-排名前2的数据

limit 1,2-排名2，3 名的数据

同理：

example：如果按照片场排列，John Lasseter导演导过片长第3长的电影是哪部，列出名字即可

select title

from movies

where director='John Lasseter'

order by length\_minutes desc

limit 2,1

2. 列出所有的电影id,名字和销售总额(以百万美元为单位计算)

select id,title,(b.domestic\_sales+international\_sales)/1000000 as total

from movies as m

left join boxoffice as b

on m.id=b.movie\_id

知识点：

    如果针对的是每一行数据，对每一行原始数据进行计算(行数不变)，可以直接用+/-，比如abs(A-B)

    如果是需要将多行数据进行合并(将某几行数据进行求和，行数变化)，需要使用sum函数

3. 按角色分组算出每个角色按有办公室和没办公室的统计人数(列出角色，数量，有无办公室)，注意一个角色如果部分有办公室，部分没有则需要进行分开统计

select role,case

when building is not null then 1

else 0

end as have,count(\*)

from employees as e

group by role,have

知识点：case when

    当原始数据需要统计多栏数目，且这些数据原表没有时，考虑使用case对数据进行归纳和分类

leetcode-cn.com

176. 获取 Employee 表中第二高的薪水（Salary）,如果不存在第二高的薪水，那么查询应返回 null

    1.

    select (

        select distinct salary

        from employee

        order by salary

        desc limit 1,1)

        as SecondHighestSalary

    2.

    select IFNULL((

        select distinct(Salary)

        from Employee

        order by Salary desc

        limit 1,1),null)

        as SecondHighestSalary

知识点：

    1. 第二高的薪水用到distinct

    2. 返回null值不可以用case when函数，可以使用ifnull(不为null返回值，为null返回值)

177.

如果两个分数相同，则两个分数排名（Rank）相同。请注意，平分后的下一个名次应该是下一个连续的整数值。换句话说，名次之间不应该有“间隔”。

    1.

    新方法：

    select (

        select score

        from scores

        order by score desc

    ) as a,

    dense\_rank() over(order by a) as Rank

    from scores

    2.

    经典方法：

    知识点：

    排序函数：row\_number,rank,dense\_rank(只生成序号)

    公式：row\_number() over(order by '')as ''

    区别：

        row\_number：顺序产生序号

        rank：相同值序号相同，会跳号

        dense\_rank：相同值序号相同，序号顺序递增

        example: select rank() over(order by field1),\* from t\_table order by field1

626. 小美是一所中学的信息科技老师，她有一张 seat 座位表，平时用来储存学生名字和与他们相对应的座位 id。

    其中纵列的 id 是连续递增的

    小美想改变相邻俩学生的座位。

    如果学生人数是奇数，则不需要改变最后一个同学的座位。

    select case

    when id%2=1 and id=(select max(id) from seat) then id

    when id%2=1 then id+1

    when id%2=0 then id-1

    end as id,student

    from seat

    order by id asc

    知识点：case when

    case可以对多多条件进行筛选，越精细的条件放在越前面

197. 给定一个 Weather 表，编写一个 SQL 查询，来查找与之前（昨天的）日期相比温度更高的所有日期的 Id。

select id

from weather as w1

where temperature>(

    select temperature

    from weather as w2

    where datediff(w1.recorddate,recorddate)=1

)

知识点：日期之间可以用>或者<，如果是日期差值需要用datediff

datediff(A,B)=date(A)-date(B)

184. 编写一个 SQL 查询，找出每个部门工资最高的员工。对于上述表，您的 SQL 查询应返回以下行（行的顺序无关紧要）

方法1：

select d.name Department,b.name Employee,Salary

from (

    select e.Name,e.Salary,e.DepartmentId

    from Employee e,(

        select max(Salary) m,DepartmentId

        from Employee

        group by DepartmentId) a

    where e.Salary =a.m and e.DepartmentId =a.DepartmentId) as b , Department as d

where b.DepartmentId =d.id;

方法2：

SELECT S.NAME, S.EMPLOYEE, S.SALARY  FROM (

    SELECT D.NAME,

    T.NAME EMPLOYEE,T.SALARY,

    ROW\_NUMBER() OVER(PARTITION BY T.DEPARTMENTID ORDER BY T.SALARY DESC) as RN

    FROM EMPLOYEE T

    LEFT JOIN DEPARTMENT D

    ON T.DEPARTMENTID = D.ID) as S

WHERE S.RN = 1

知识点：窗函数(row\_number,rank,dense\_rank)

区别：

row\_number：顺序产生序列，不考虑值相等的情况

rank：考虑到值相等的情况，有跳值的情况(1,1,1,4)

dense\_rank：考虑到值相等的情况，无跳值的情况(1,1,1,2)

注意：

    1. 窗口函数不能直接用having，只可以使用select,order by，所以一般使窗口函数成表，再调用表使用where筛选

example1:

    185. 编写一个 SQL 查询，找出每个部门获得前三高工资的所有员工.

    IT 部门中，Max 获得了最高的工资，Randy 和 Joe 都拿到了第二高的工资，Will 的工资排第三

    销售部门（Sales）只有两名员工，Henry 的工资最高，Sam 的工资排第二

    select w.name as Department,q.name as employee,q.salary

    from (

        select departmentid,name,salary,

        dense\_rank() over(partition by departmentid order by salary desc) as r

        from employee

    ) as q,department as w

    where q.departmentid=w.id and q.r<=3

example2：

    请你找出每个岗位分数排名前2的用户，得到的结果先按照language的name升序排序，再按照积分降序排序，最后按照grade的id升序排序

    select id,name,score

    from (

    select g.id,l.name,g.score,

    dense\_rank() over(partition by g.language\_id order by score desc) as n

        from grade as g

        join language as l

        on l.id=g.language\_id

    )

    where n<=2

    order by name asc,score desc,id asc

example3：

    找出每个部门工资最高的员工

    Max 和 Jim 在 IT 部门的工资都是最高的，Henry 在销售部的工资最高

    select m.name as department,m.employee,m.salary

    from (

        SELECT D.NAME,

        T.NAME EMPLOYEE,

        T.SALARY,

        dense\_rank() OVER(partition by t.departmentid ORDER BY T.SALARY DESC) RN

        FROM EMPLOYEE T

        JOIN DEPARTMENT D

        ON T.DEPARTMENTID = D.ID

    ) as  m

    where m.rn=1

example4：

    找出每个岗位分数排名前2的用户，得到的结果先按照language的name升序排序，再按照积分降序排序，最后按照grade的id升序排序

    select id,name,score

    from(

        select g.id,l.name,score,

        dense\_rank() over(partition by language\_id order by score desc) as rn

        from grade as g

        join language as l

        on g.language\_id=l.id) as a

    where a.rn<=2

    order by name,score desc,id

3. 查找所有至少连续出现三次的数字

select distinct num as ConsecutiveNums

from logs as l1

where num=(

    select num

    from logs

    where id=l1.id+1)

and num=(

    select num

    from logs

    where id=l1.id+2)

知识点：连续出现这类问题的时候要记得给原表命名，不能直接id=id+1

4. 查找当前薪水(to\_date='9999-01-01')排名第二多的员工编号emp\_no、薪水salary、last\_name以及first\_name，你可以不使用order by完成吗

通用型可以求任意第几高，并且可以求多个形同工资

select e.emp\_no,s.salary,e.last\_name,e.first\_name

from employees as e

join salaries as s

on e.emp\_no=s.emp\_no

where s.to\_date='9999-01-01'

and s.salary =

(

     select s1.salary

     from salaries s1

     join salaries s2

     on s1.salary<=s2.salary

     where s1.to\_date='9999-01-01' and s2.to\_date='9999-01-01'

     group by s1.salary

     having count(distinct s2.salary)=2

 )

牛客网

66. 每一个日期里面，正常用户发送给正常用户邮件失败的概率是多少，结果最多保留到小数点后面3位(3位之后的四舍五入)，并且按照日期升序排序

select b1.date as m,round(1\*round(ifnull(num2,0),3)/num1,3)

    from(select date,count(\*) as num1

    from email as e

    where send\_id in(

        select id

        from user

        where is\_blacklist=0

    ) and receive\_id in(

        select id

        from user

        where is\_blacklist=0)

    group by date

    )as b1

    left join (select date,count(\*) as num2

    from email as e

    where send\_id in(

        select id

        from user

        where is\_blacklist=0

    ) and receive\_id in(

        select id

        from user

        where is\_blacklist=0) and type='no\_completed'

    group by date

    )as b2

    on b1.date=b2.date

    order by m asc

知识点：

    1. 这种涉及到除法/比例的题目需要考虑到结果为1.0或者0的情况，如果出现0的情况，很有可能有一个部分返回null值

    2. 由于当值返回null值时，前面的目录也不会有查询结果(如当时间为1-12时，num2为numm，b2的结果只有时间为1-11的那一行)

    3. 所以判断ifnull的函数必须加在第一行的筛选中

    4. 由于会出现null值，为了保证数据的完整性，必须要使用left/right join函数

    5. 结果进行order by的时候，选取的列名不能和分表的列名重合(比如本题不能使用date作为order的列名，只能使用m)

example: 查出 2013年10月1日 至 2013年10月3日 期间非禁止用户的取消率。基于上表，你的 SQL 语句应返回如下结果，取消率（Cancellation Rate）保留两位小数

        取消率的计算方式如下：(被司机或乘客取消的非禁止用户生成的订单数量) / (非禁止用户生成的订单总数)

    select t1.request\_at as Day,ifnull(round(((t1.num1-t2.num1)/t1.num1),2),1.0) as 'Cancellation Rate'

    from (

        select request\_at,count(\*) as num1

    from trips

    where client\_id in (

        select users\_id

        from users

        where banned='No'

    ) and driver\_id  in(

        select users\_id

        from users

        where banned='No'

    )

    and request\_at in ('2013-10-01','2013-10-02','2013-10-03')

    group by request\_at

    ) as t1

    left join

    (

        select request\_at,case

    when count(\*) is not null then count(\*)

    else 0

    end as num1

    from trips

    where client\_id in (

        select users\_id

        from users

        where banned='No'

    ) and driver\_id  in(

        select users\_id

        from users

        where banned='No'

    ) and status='completed'

    group by request\_at

    ) as t2

    on t1.request\_at=t2.request\_at

68. 查询每个用户最近一天登录的日子，用户的名字，以及用户用的设备的名字，并且查询结果按照user的name升序排序

    方法1：

    select u.name as u\_n,c.name as c\_n,l.date as d

    from login as l

    join user as u

    on l.user\_id=u.id

    join client as c

    on l.client\_id=c.id

    where l.id in (

        select m

        from

        (select id as m,user\_id,max(date)

        from login

        group by user\_id

        )

    )

    order by u\_n

    方法2：

    select u.name u\_n,c.name c\_n, max(l.date) d

    from login l

    join user u on u.id=l.user\_id

    join client c on c.id=l.client\_id

    group by l.user\_id

    order by u.name asc;

14. 从titles表获取按照title进行分组，每组个数大于等于2，给出title以及对应的数目t。

注意对于重复的emp\_no进行忽略(即emp\_no重复的title不计算，title对应的数目t不增加)。

select ti,count(\*) as t

    from (

        select distinct emp\_no,title as ti

        from titles

    )

group by ti

having t>=2

知识点：

    1. distinct 多个字段的时候可以将所有字段进行筛选，只要不是所有字段都一样就可以进行记录

    2. 碰到多个字段去重的问题可以将所有字段先distinct再进行表筛选

    3. 碰到去重和进行筛选的问题，需要明确是先去重再筛选还是先筛选再去重(关系到having 在小表中还是大表中)

53. 按照dept\_no进行汇总，属于同一个部门的emp\_no按照逗号进行连接，结果给出dept\_no以及连接出的结果employees

select dept\_no,group\_concat(emp\_no,',')

from dept\_emp

group by dept\_no

知识点：group\_concat

    1. 可以将group后的每一个字段进行拼接，如果不用','，进行设置，默认都为逗号连接

    2. 可以进行去重，group\_concat(distinct score)

    3. 可以进行排序，group\_concat(distinct score order by score desc)

54. 查找排除最大、最小salary之后的当前(to\_date = '9999-01-01' )员工的平均工资avg\_salary。

错误答案：

select avg(salary) as avg\_salary

from salaries as s1

where salary not in(

    select max(salary),min(salary)

    from salaries as s

    where to\_date='9999-01-01'

) and s1.to\_date='9999-01-01'

正确答案：

select avg(salary) as avg\_salary

from salaries as s1

where salary <>(

    select max(salary)

    from salaries as s

    where to\_date='9999-01-01')

and salary<>(

    select min(salary)

    from salaries as s

    where to\_date='9999-01-01')

and s1.to\_date='9999-01-01'

知识点：不可以用not in (select max(salary),min(salary))

原因是in/not in 后面只能跟一列多行数据(一个字段),题目中跟了两列数据(两个字段)，会报错

61. 对于employees表中，输出first\_name排名(按first\_name升序排序)为奇数的first\_name

窗口函数：

select first\_name

from(

    select first\_name,

    row\_number() over(order by first\_name asc) as rk

    from employees

    order by first\_name

)

where rk%2=1

order by first\_name asc

普通方法：

SELECT e1.first\_name

FROM employees e1

WHERE

(SELECT COUNT(\*)

FROM employees e2

WHERE e1.first\_name >= e2.first\_name)%2 =1;

知识点：名字(字符串)也可以比较大小