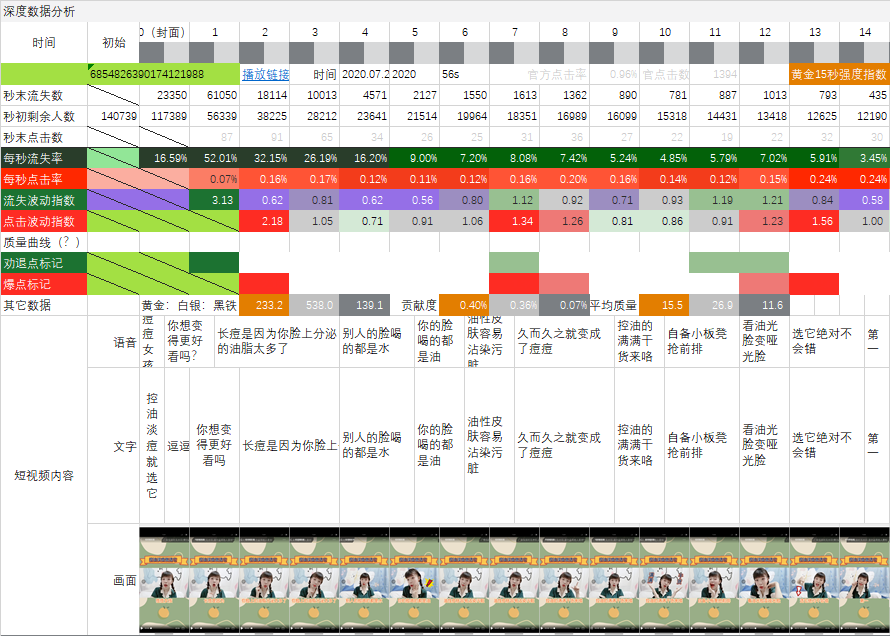
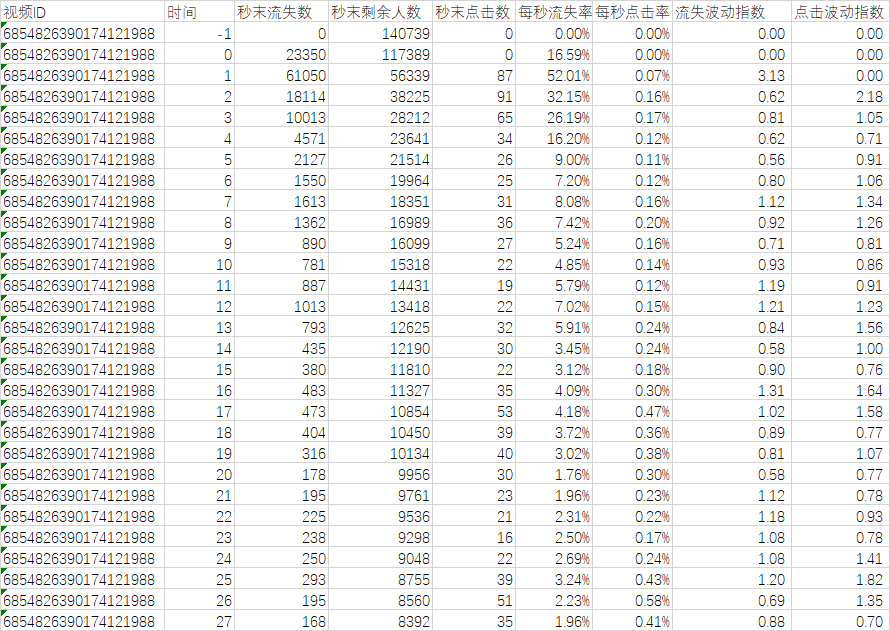
# 戴师兄SQL零基础入门教程

1. 第一天
   1. SQL是什么？
      1. **SQL是一种能指挥数据库自动&批量处理大量Excel表格的超级简单的语言**
         1. 语法标准由ANSI美国国家标准化组织统一制定，主流数据库的关键语法都是相同的
            1. 如select、update、delete、insert、where等
         2. 只针对结构化数据进行处理，不处理乱糟糟非结构化数据
            1. 非结构化数据



* + - * 1. 结构化数据



* + - 1. 有严格的语法结构和运行顺序（所以学起来非常简单）
         1. 语法结构：select--from--where--group by--having--order by--limit
         2. 运行顺序：from--where--group by--having--order by--limit--select
    1. **SQL一点也不难，难的是那些本身就很难的表格处理逻辑**
  1. 免费教程学完后能到什么程度？
     1. 在Mysql等标准数据库实现80%以上场景的数据查询
     2. 对于一般的运营、产品经理和数据分析实习岗位完全够用
  2. 免费教程如何使用？
     1. 直播的小伙伴重点在于听师兄介绍语法，然后跟上和理解师兄的解题逻辑
     2. 直播后一定要跟着录播认认真真地完成每一题的代码
     3. 录播的小伙伴看到哪练到哪，边暂停边学就好
  3. 主知识点一：select&from
     1. 【知识点引入】
        1. 首先我们先认识两句子句，select和from
           1. 【标准语法】

select 字段名

from 表名称

* + - * 1. 【语法解释】

select 字段名 表明选择查询表格中的哪几列进行查看

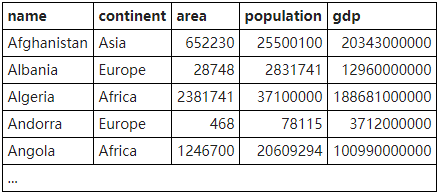
from 表格名 表明你查询的数据来自哪一个数据库表

* + - 1. 但凡大家要从数据库中查询数据，必不可少的就是select和from语句
      2. 现在给大家5秒的时间记忆这两个单词，5，4，3，2，1
      3. 好啦，大家现在已经学会最基础的sql查询语句啦
      4. 接下来让我们一起来学习如何应用select和from语句进行查询吧
    1. 【例题讲解】
       1. 【基础查询select单列&多列&所有列&别名应用】
          1. 我们点开链接<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第一题

【题目】

阅读关于表格的信息，观察SQL代码并运行

我们先观察我们要查询的表格中的表头，知道每个列名称，再查看表格前几行数据知道每一行和每一列数据的大致内容



观察题目给我们的代码



由刚才我们所学的select 和from组成，select后有多个字段名，表明这段代码查询的是多列数据，字段顺序按照select后所写的字段顺序依次展现，注意逗号是在英文输入法下输入的

select name, continent, population from world表明要查询名为 "name" 、 "continent" 和"population"的列的内容从名为world的数据库表中，同时记住SQL对大小写不敏感SELECT和select完全等效，但是为了书写效率，成熟的分析师一般都用小写

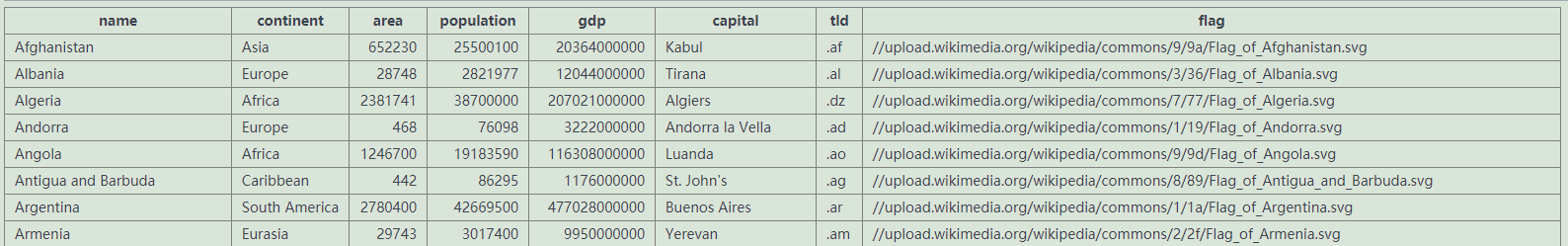
点击运行代码

得到结果



将所有列名称变更为单一的\*：SELECT \* FROM world

运行代码，得到表格的所有列，所以通配符\*是选取所有列的快捷方式



在select中还可以为字段名指定别名

在select的字段名或者from的表格名后加 [as 别名]，即可为字段名起个别名，也可以省略as，但是记得字段名和表格名后的别名要用空格隔开

select name as 国家名, continent 大洲, population 人口 from world

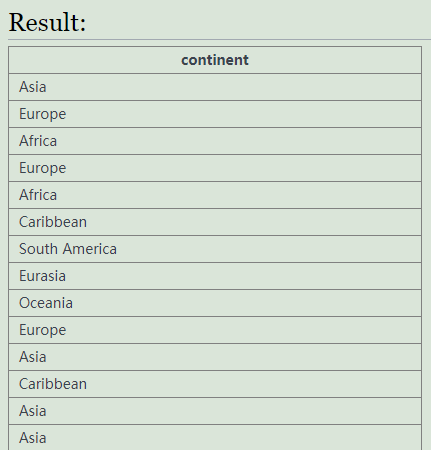


* + - 1. 【select中distinct去重】
         1. select中还可以加distinct对重复的行数据进行去重，我们点开链接（distinct去重）<https://sqlzoo.net/wiki/SUM_and_COUNT>第二题

【阅读题目】

列出所有大洲（continent），且仅出现一次

如果运行代码select continent from world，就会发现大洲多次出现在查询结果里，这不是我们所要的答案



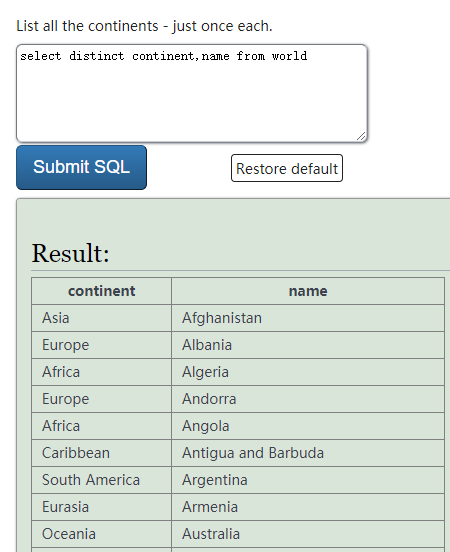
在select后加入distinct对重复的数据去重

运行代码select distinct continent from world得到去重后的数据



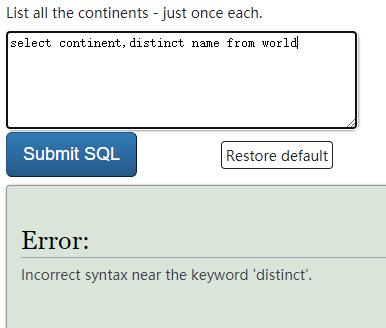
如果select distinct后有多个字段名时，是对重复的行数据进行去重

运行select distinct continent,name from world



注意distinct在select中使用时，只能加在select后而不是字段名前

所以大家如果这样写：select continent,distinct name from world，就会出现语法错误



* + - 1. 【select中计算字段的运用】
         1. select中还可以直接进行计算，我们点击链接<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第三题

【题目】

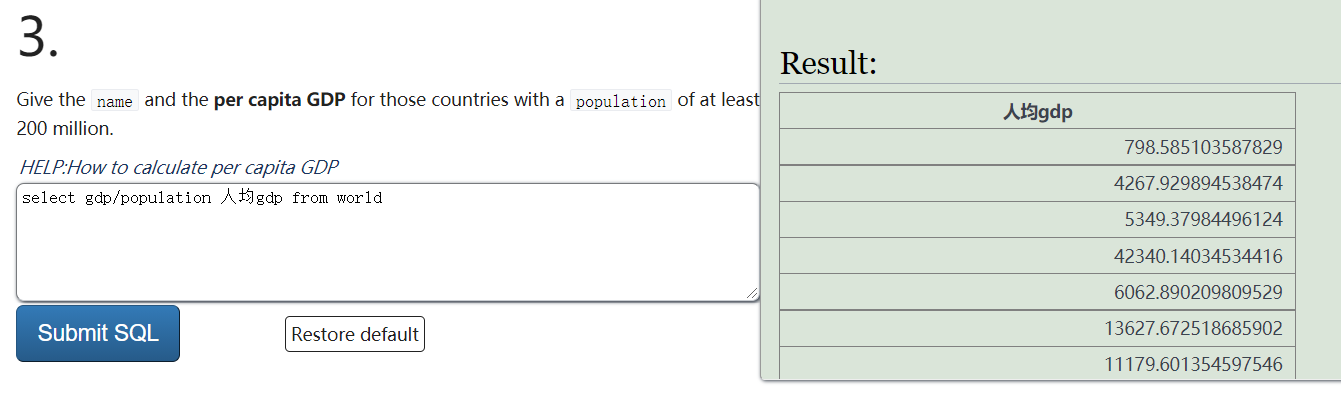
查询人口数至少2亿的国家名和人均GDP

人均GDP该如何计算呢？这里有给出公式：人均GDP = GDP/population

可见人均GDP无法从数据库表world中直接得到，需要进行计算，而这个新的字段人均GDP由数据库表中的两个字段GDP和population计算得到，因此我们在select后写算式GDP/population构建计算字段

运行代码

select name,GDP/population 人均gdp from world



此时每行数据的GDP和population相除，生成我们所需的新字段人均gdp，我们还可以对计算字段添加别名人均gdp

* + 1. 【总结】
       1. 基础语法
          1. select 字段名 from 表名称
       2. 别名语法
          1. select 字段名 as 别名 from 表名称
          2. 注意：as可以省略
       3. 查询多列
          1. select 字段名1, 字段名2, 字段名3 from 表名称
       4. 查询所有列
          1. select \* from 表名称
       5. 数据去重
          1. select distinct 字段名 from 表名称
       6. select中的计算字段
          1. select 字段名,计算字段 from 表名称
          2. 注意：计算字段中的算式所涉及的 字段名必须是表格中包含的，或者算式本身可以独立运算
    2. 【练习题】
       1. 自己完整敲一遍知识点中出现的代码吧~
  1. 主知识点二：where
     1. 【知识点引入】
        1. 我们来认识第三个子句where，where子句写在from之后
           1. 【标准语法】

select 字段名

from 表格名

where 条件代码

* + - * 1. 【语法解释】

where 条件代码 表明从数据库表中选取满足条件的数据

* + - 1. where子句主要用来进行数据的筛选，从表格中筛选出符合条件的行数据
      2. 接下来让我们来快速应用where子句吧~
    1. 【例题讲解】
       1. 【运算符】
          1. 条件代码中最常用的是运算符。

运算符查询标准语法

select 字段名

from 表名称

where 字段名 运算符 值

* + - * 1. 先来看这个表格，比较运算符，用于判断表中的哪些数据符合条件，and、or和not为逻辑运算符，not一般和与其他连用例如not in等。



* + - * 1. and 和 or 可以把两个或多个条件结合起来。若and连接的两个条件对于该数据都成立，就显示该数据，若or连接的两个条件对于该数据只要有一个成立，就显示该数据，否则就筛除该数据。
        2. 点击链接<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第三题

【题目】

查询人口数至少2亿的国家名和人均GDP

人均GDP如何查询我们已经在前面讲过了，这里就不再赘述了。这里着重看的是“人口数至少2亿”这个条件，说人话就是人口population大于等于两亿，写成代码语言就是population>=200000000,并写在where子句后，与select和from子句组合在一起得到正确代码

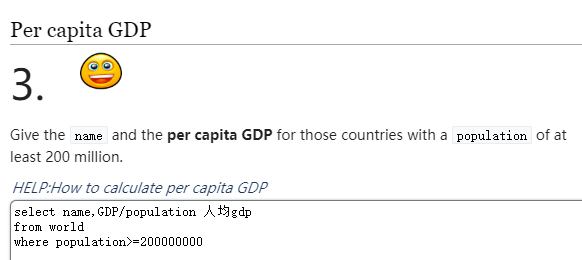
【运行代码】

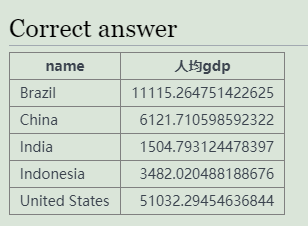
select name,GDP/population 人均gdp

from world

where population>=200000000

【运行结果】





* + - * 1. 点击链接<https://sqlzoo.net/wiki/SELECT_basics>

看第一题

【题目】

修改代码查询德国（Germany）人口数量

我们把题目给的代码中的France改成Germany

注意在France外的单引号不要删除，所有的字符串都要用单引号包裹，与不用单引号包裹的字段名、表名和语法作区分，如果字符串仅有数字可以省略单引号

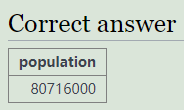
同时也要注意单引号是在英文输入法下输入的和逗号一样，不然就会出现语法错误

运行代码

SELECT population

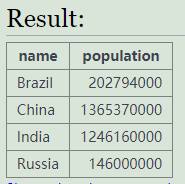
FROM world

WHERE name = 'Germany'



看第二题，了解in的用法



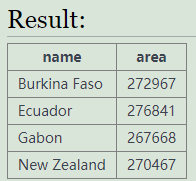


in会筛选出字段中所有与括号内数据相等的行

在这题中in筛选的是名为name的这列字段中值等于Brazil, Russia, India, China的行

看第三题，了解between...and...的用法





between and选取介于两个值之间且包含这两个值的数据范围，这些值可以是数值或者日期

在这题中between and筛选出面积area字段的值在25万和30万之间的行

* + - 1. 【模糊查询like】
         1. 条件代码中除了使用运算符来进行条件判断，还可以使用like操作符组合通配符进行模糊查询

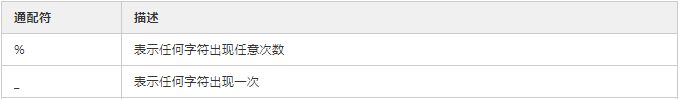
模糊查询标准语法

select 字段名

from 表格名

where 字段名 like '通配符+字符'

通配符用来匹配值的一部分，跟在LIKE关键字后面进行数据过滤常用的通配符有%和\_，%用来匹配多个字符可以是零个、一个也可以是多个字符，\_仅能用来匹配单个字符



点击链接<https://sqlzoo.net/wiki/SELECT_names>

看第五题

【题目】

查询国家名name字段中以C开头ia结尾的国家

由于C和ia之间没有限定字符数，所以我们使用%通配符

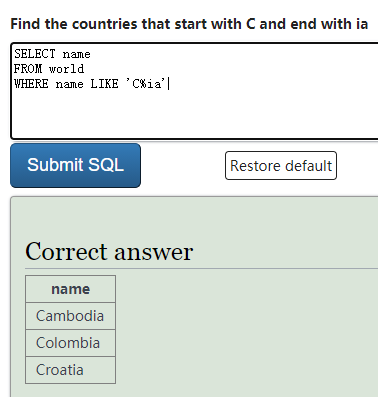
写成代码就是where name like 'C%ia'

运行正确代码

SELECT name

FROM world

WHERE name LIKE 'C%ia'



看第八题

【题目】

查询国家名name中第二个字符为't'的国家

指定第二个字符为t，此时使用\_通配符，但未指定字符t后的字符数因此用%

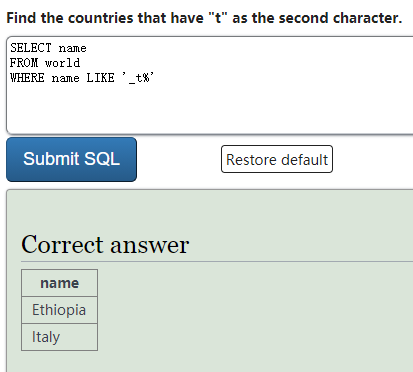
写成代码 where name like '\_t%'

运行代码

SELECT name

FROM world

WHERE name LIKE '\_t%'



看第九题

【题目】

查询国家名name中含有两个o且被两个字符隔开的国家名

题目指定有两个o且被两个字符隔开，这两个字符用两个\_进行匹配而不能用%，因为\_仅指代一个字符，不能没有也不能是两个，而%匹配数量不定的字符，可能没有也可能有二三四五六个

这里题目只指明两个o之间有两个字符但没有指明两个o前或者后有多少个字符

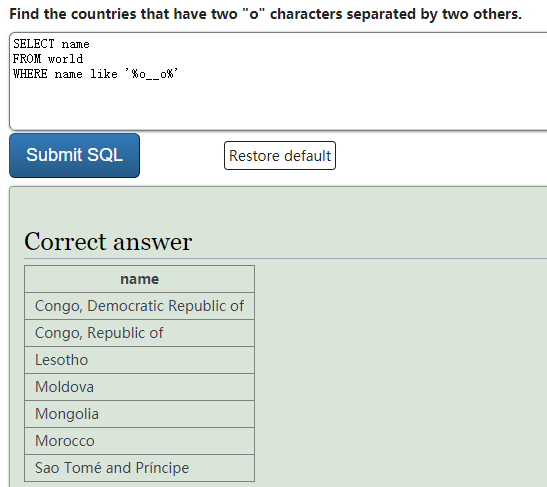
因此写成代码where name like '%o\_\_o%'

运行代码

SELECT name

FROM world

WHERE name like '%o\_\_o%'



* + - 1. 【多条件查询】
         1. 前面我们已经学会使用运算符和like加通配符进行查询啦，我们还可以使用and或者or逻辑运算符对多个条件进行组合筛选我们想要的数据
         2. 点击链接<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第八题

【题目】

从nobel表中查询1980年物理奖获得者和1984年化学奖获得者的年份（yr），学科（subject）和获奖者姓名（winner）

这一题是从一张名为nobel的数据库表中查询数据，输入代码select \* from nobel可以查看一下这张表

这张表有三个字段，分别是年份yr，学科subject，获奖者姓名winner，记录的是获得诺贝尔奖的获奖相关信息

再看题目，查询1980年物理奖获得者和1984年化学奖获得者，翻译成可以写成代码的话就是同时满足年份yr是1980年且学科subject是物理physics或者同时满足年份yr是1984年且学科subject是化学chemistry的获奖者，"同时"满足用逻辑运算符and，"或者"用逻辑运算符or

写出代码where (subject = 'Physics' and yr = 1980) or (subject = 'Chemistry' and yr = 1984)

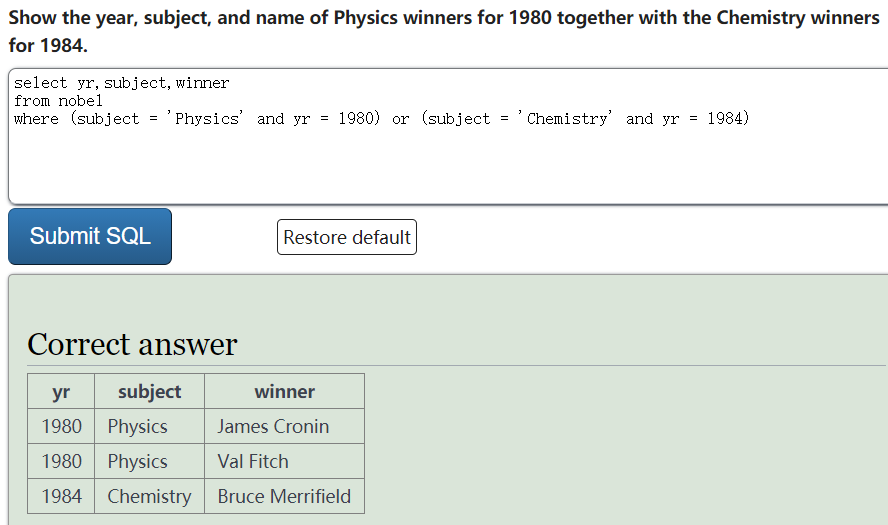
逻辑是同时满足两个条件或者同时满足另两个条件所以or前后两个条件部分需要加括号

运行代码，得到正确结果

select yr,subject,winner

from nobel

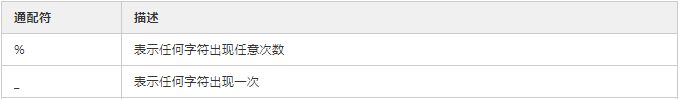
where (subject = 'Physics' and yr = 1980) or (subject = 'Chemistry' and yr = 1984)



* + 1. 【总结】
       1. 标准语法
          1. select 字段名
          2. from 表格名
          3. where 条件代码
       2. 运算符查询语法
          1. select 字段名
          2. from 表名称
          3. where 字段名 运算符 值
       3. 模糊查询语法
          1. select 字段名
          2. from 表名称
          3. where 字段名 like '通配符+字符'
       4. 使用多条件查询
          1. select 字段名
          2. from 表名称
          3. where 条件代码1 and|or 条件代码2
       5. 运算符

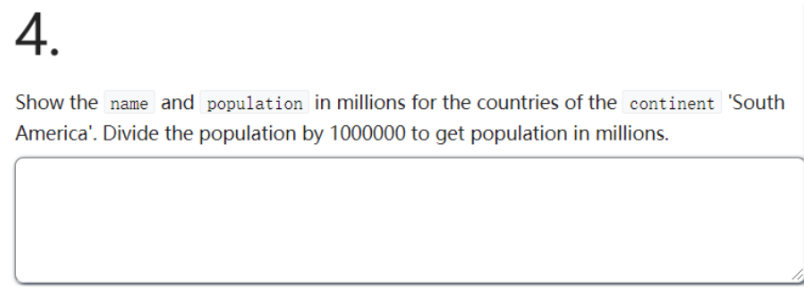


* + - 1. 通配符



* + 1. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第四题
          1. 【题目】

查询南美洲（south america）所有国家的名称以及它们以百万（1000000 ）为单位的人口数量



* + - * 1. 【正确答案】

select

name

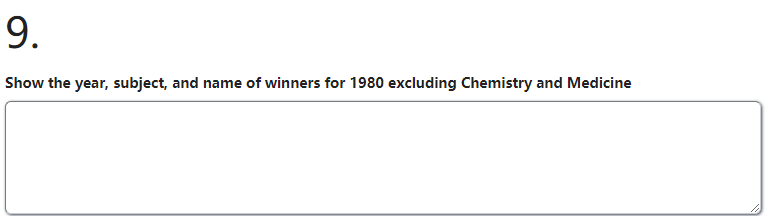
, population/1000000 population\_in\_millions

from world

where continent = 'South America';

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第九题
         1. 【题目】

查询1980年除诺贝尔化学奖和诺贝尔医学奖外其余奖项获奖者的所有信息。



* + - * 1. 【正确答案】

select \*

from nobel

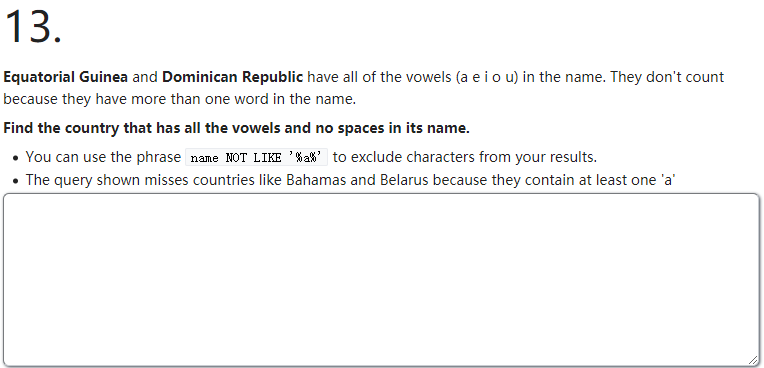
where yr = 1980

and subject not in ('Chemistry','Medicine')

* + - 1. 【3】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第十三题
         1. 【题目】

查询既包含有所有元音字母（a,e,i,o,u），同时国家名中没有空格的国家，最后显示他们的名字

例如，赤道几内亚**Equatorial Guinea** 和 多米尼加共和国**Dominican Republic** ，都包括有五个元音字母（a,e,i,o,u)，但这些国家不会被记录，因为国家名中，由两个单词构成（有空格）



* + - * 1. 【正确答案】

select name

from world

where name like '%a%'

and name like '%e%'

and name like '%i%'

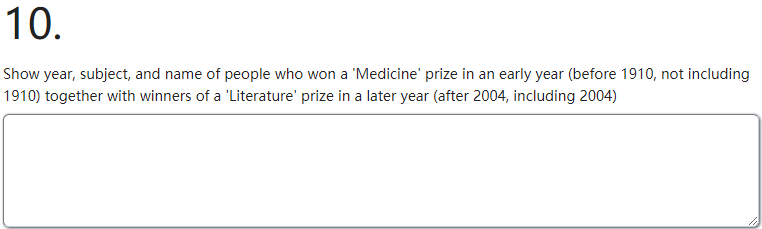
and name like '%o%'

and name like '%u%'

and name not like '% %';

* + - 1. 【4】<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第十题
         1. 【题目】

查询1910年以前（不含1910）诺贝尔医学奖获得者和2004年及以后诺贝尔文学奖获得者的所有信息



* + - * 1. 【正确答案】

select \*

from nobel

where (subject = 'Medicine' and yr < 1910)

or (subject = 'Literature' and yr >= 2004)

* 1. 主知识点三：聚合函数、group by&having
     1. 【知识点引入】
        1. 我们先来认识最常见的聚合函数



* + - * 1. 有时候我们只是需要获取数据的汇总信息，比如说行数啊、平均值啊这种，并不需要吧所有数据都检索出来，只需要使用聚合函数即可
        2. 【函数说明】

AVG() 返回某列的均值

COUNT() 返回某列的行数

MAX() 返回某列的最大值

MIN() 返回某列的最小值

SUM() 返回某列的和

* + - * 1. 注意聚合函数都会忽略列中的NULL值，但是COUNT(\*)也就是统计全部数据的行数时，不会忽略NULL值
      1. 我们再来认识两个子句group by 和having
         1. 【标准语法】

select 字段名1

from 表格名

[where 条件代码]

group by 字段名1

having 条件代码

* + - * 1. 【语法解释】

之前学到的筛选操作都是基于整个表去进行的，那如果想要依据某列中的不同类别（比如说不同品牌/不同性别等等）进行分类统计时，就要用到数据分组，在SQL中数据分组是使用GROUP BY子句建立

GROUP BY子句可以包含任意数量的列，因而可以对分组进行多重嵌套，如按照班级和性别进行分组的话，结果中班级A包含男生组和女生组，班级B也包含男生组和女生组

having 子句和where 子句一样用于条件筛选，两者的操作符一致，但不同的是where是在指定分组前对数据进行筛选过滤，而having可以对分组后的数据进行筛选过滤

注意写的子句的顺序不可颠倒，需要严格按照语法的顺序！

* + 1. 【例题讲解】
       1. 【聚合函数】
          1. 点击链接<https://sqlzoo.net/wiki/SUM_and_COUNT>

第三题

【题目】

从world表中查询非洲（Africa）的总GDP

要求查询的是所有大洲（continent）为非洲（Africa）的GDP总和

求和用聚合函数sum，对gdp求和就写select sum(GDP)，可以写上别名gdp总和

从world表中查询，写from world

筛选表中大洲（continent）为非洲（Africa）的数据，写where continent = 'Africa'

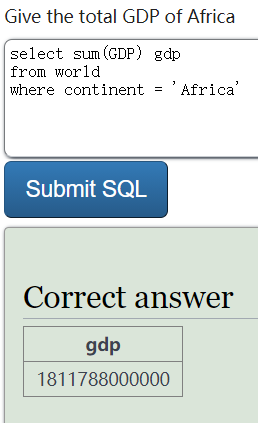
【运行代码】

select sum(GDP) gdp总和

from world

where continent = 'Africa'

【运行结果】



注意只使用聚合函数不使用group by时是对整张表作聚合运算

此时select后只能写聚合函数不能添加其他字段名

第四题

【题目】

从world表中查询有多少国家（name）的面积（area）至少有100万

查询有多少个国家需要进行计数，这里使用count函数

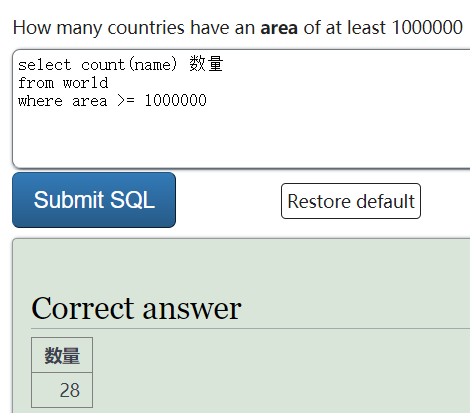
【运行代码】

select count(name) 数量

from world

where area >= 1000000

【运行结果】



* + - 1. 【group by数据分组】
         1. 点击链接<https://sqlzoo.net/wiki/SUM_and_COUNT>第六题

【题目】

查询每个大洲（continent）和大洲内的国家（name）数量

这题需要对数据按照大洲字段分组，我们写出代码group by continent

并对每组的数据进行计数聚合运算，最后显示每个大洲和大洲中的国家数量，写出代码select continent,count(name)

对代码进行组合

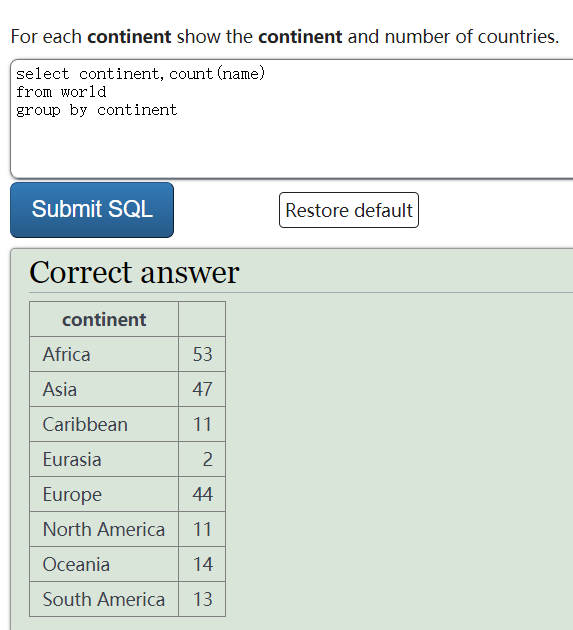
【运行代码】

select continent,count(name)

from world

group by continent

【运行结果】



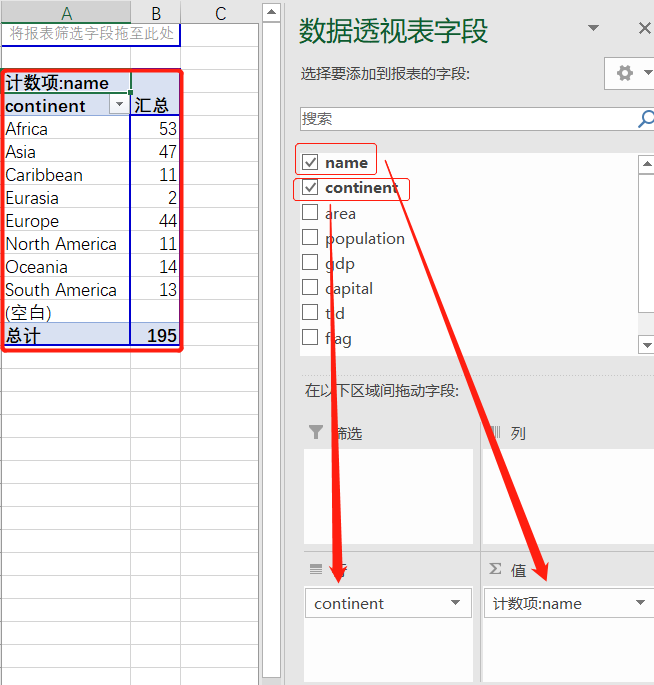
group by 数据分组类似于excel中的数据透视表功能

【excel数据透视表演示】

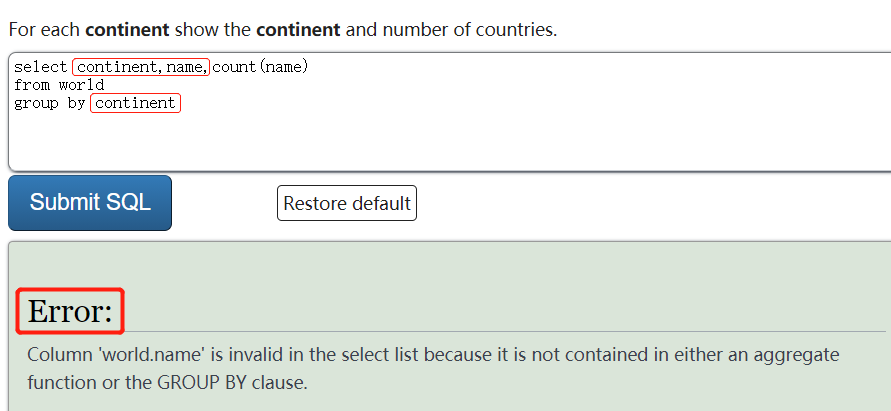
1.



2.插入数据透视表



需要注意的是使用group by时，select后的字段只能写group by中出现过的字段和聚合函数，否则就会出现错误



* + - 1. 【having基于聚合运算结果进行筛选】
         1. 点击链接<https://sqlzoo.net/wiki/SUM_and_COUNT>第八题

【题目】

查询总人口数量至少为1亿（100000000）的大洲

题目要求对大洲人口数量进行筛选

首先要对数据进行分组写出代码group by continent

然后对分组后的数据进行筛选此时使用having

要对组内的数据人口数量（population）求和进行筛选，写出代码 having sum(population) >100000000

对代码按正确顺序组合

【运行代码】

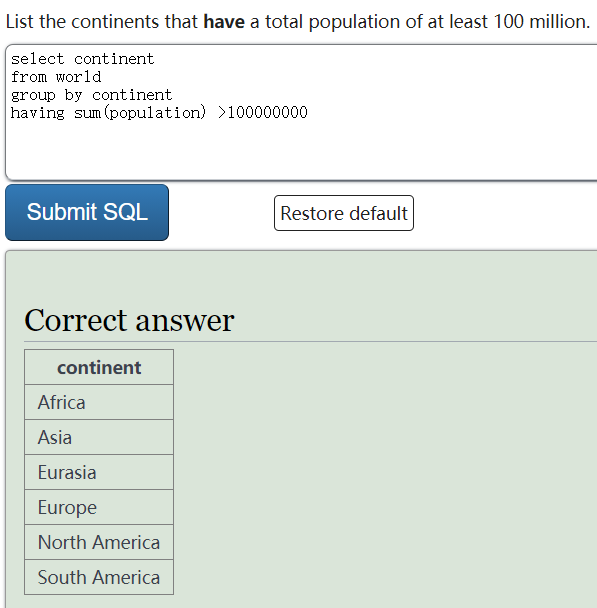
select continent

from world

group by continent

having sum(population) >100000000

【运行结果】

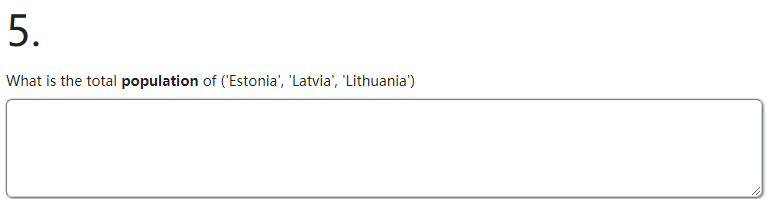


* + 1. 【总结】
       1. 常用聚合函数



* + - 1. 标准语法
         1. select 字段名1
         2. from 表格名
         3. [where 条件代码]
         4. group by 字段名1
         5. having 条件代码
    1. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/SUM_and_COUNT>第五题
          1. 【题目】

计算Estonia, Latvia, Lithuania这几个国家的总人口数



* + - * 1. 【正确答案】

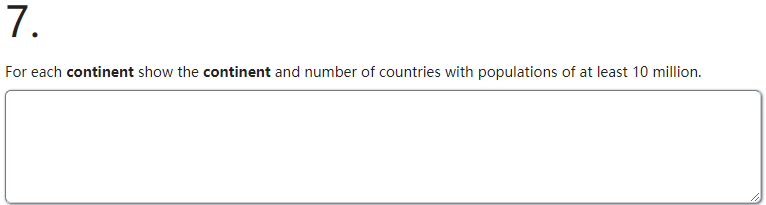
select sum(population)

from world

where name in ('Estonia', 'Latvia', 'Lithuania')

* + - 1. 【2】<https://sqlzoo.net/wiki/SUM_and_COUNT>第七题
         1. 【题目】

查询每个大洲和该大洲里人口数超过1千万的国家的数量



* + - * 1. 【正确答案】

select

continent

,count(name) 'number of countries'

from world

where population >= 10000000

group by continent

* 1. 主知识点四：order by
     1. 【知识点引入】
        1. 我们再来认识第六个子句order by
           1. 【标准语法】

select 字段名1

from 表格名

[where 条件代码]

[group by 字段名1]

[having 条件代码]

order by 字段名 asc|desc

* + - * 1. 【语法解释】

order by 子句会对最后查询出的结果集进行排序

order by 字段名，表明根据指定的字段进行排序

asc指定该字段升序排序，desc为降序排序，不写则默认为升序排序

order by 可以对多个字段按照主字段和次字段排序，每个字段都可以指定升序还是降序排序

* + 1. 【例题讲解】
       1. 点击链接<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第十三题
          1. 【题目】

查询姓名以Sir开头的获奖者的姓名，获奖年份和奖项

查询结果按照年份从近到远排序，再按照姓名字母顺序升序排序

* + - * 1. 查询获奖者的姓名，获奖年份和奖项，写出代码select winner, yr, subject
        2. 从数据库表nobel中查询，写出代码from nobel
        3. 姓名以Sir开头的获奖者，写出代码where winner like 'Sir%'
        4. 按年份降序排序，姓名字母顺序升序排序，写出代码order by yr desc,winner
        5. 【运行代码】

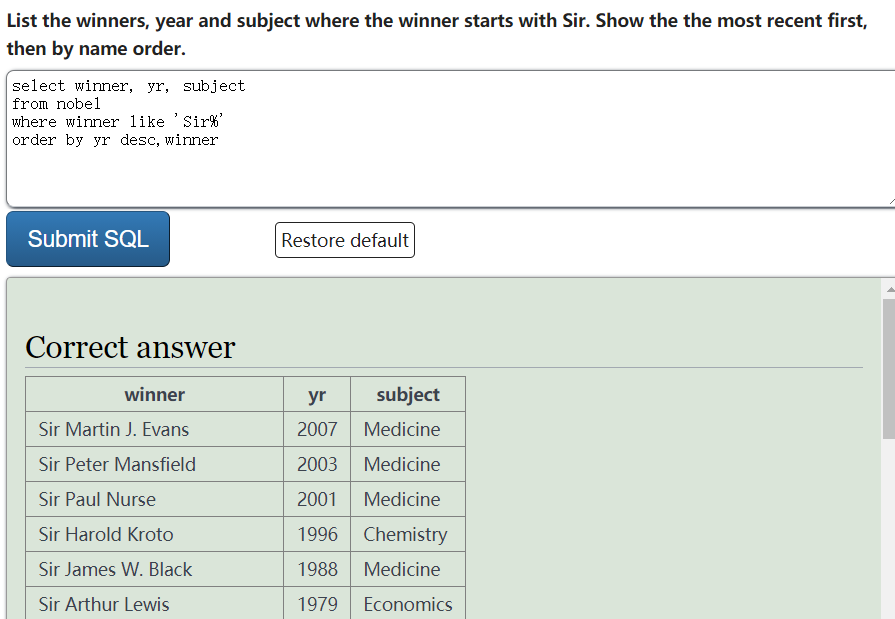
select winner, yr, subject

from nobel

where winner like 'Sir%'

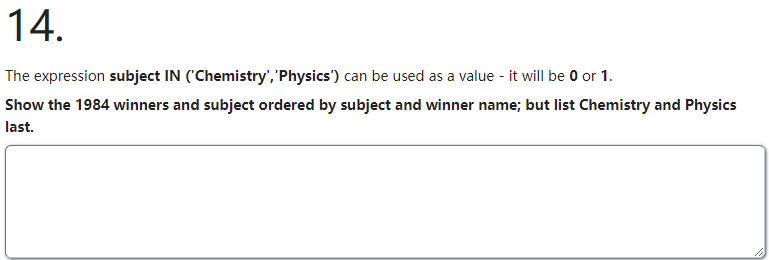
order by yr desc,winner

* + - * 1. 【运行结果】



* + 1. 【总结】
       1. 【标准语法】
          1. select 字段名1
          2. from 表格名
          3. [where 条件代码]
          4. [group by 字段名1]
          5. [having 条件代码]
          6. order by 字段名 asc|desc
    2. 【练习题】
       1. 【1】点击链接<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第十四题
          1. 【题目】

查询1984年所有获奖者的姓名和奖项。结果将诺贝尔化学奖和物理学奖排在最后，然后按照奖项排序，再按照获奖者姓名排序



* + - * 1. 【正确答案】

select

winner

,subject

from nobel

where yr = 1984

order by subject in ('chemistry','physics') , subject, winner

* 1. 主知识点五：limit
     1. 【知识点引入】
        1. select 字段名1
        2. from 表格名
        3. [where 条件代码]
        4. [group by 字段名1]
        5. [having 条件代码]
        6. [order by 字段名 asc|desc]
        7. limit n
     2. 【例题讲解】
        1. 【查询结果返回某几行】
           1. 点击链接<https://www.nowcoder.com/practice/f24966e0cb8a49c192b5e65339bc8c03?tpId=82&&tqId=29823&rp=1&ru=/ta/sql&qru=/ta/sql/question-ranking>

【题目】

分页查询employees表，每5行一页，返回第2页的数据

翻译成人话就是查询第6行到第10行的数据，共5行

写出代码limit 5,5

【运行代码】

select \*

from employees

limit 5,5

* + 1. 【总结】
       1. 【查询结果返回前n行】
          1. select 字段名1
          2. from 表格名
          3. [where 条件代码]
          4. [group by 字段名1]
          5. [having 条件代码]
          6. [order by 字段名 asc|desc]
          7. **limit n**
       2. 【查询结果返回x+1行到x+y行】
          1. select 字段名1
          2. from 表格名
          3. [where 条件代码]
          4. [group by 字段名1]
          5. [having 条件代码]
          6. [order by 字段名 asc|desc]
          7. **limit x,y**
  1. 主知识点六：子查询
     1. 【知识点引入】
        1. 到这里我们已经把sql的主要子句学完啦，接下来就是sql语法的进阶学习
        2. 首先是造就了sql语句千变万化的子查询
        3. 子查询本身就是一个完整的查询语句，然后用括号（）包裹嵌套在主查询语句中，子查询可以多层嵌套
        4. 之前所涉及到的都是从数据库中检索数据的单条语句，但当我们想要检索的数据并不能直接从数据库表中获取，而是需要从筛选后的表格中再度去查询时，就要用到子查询，相当于我们无法直达时，需要进行换乘
        5. 子查询的执行优先于主查询执行，因为主查询的条件用到了子查询的结果
        6. 子查询主要分为三类
           1. 标量子查询

查询结果只有一行一列

* + - * 1. 行子查询

查询结果只有一行

* + - * 1. 列子查询

查询结果只有一列

* + - * 1. 表子查询

查询结果有多行多列

* + 1. 【例题讲解】
       1. 【where基于子查询条件筛选】
          1. 点击链接<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第六题

【题目】

查询出gdp高于欧洲每个国家的所有国家名称，有一些国家gdp值可能为NULL，请排除这些国家

将题目翻译成人话就是查询国家名称，这些国家的gdp大于欧洲国家中最大的gdp，其中国家的gdp不为空值

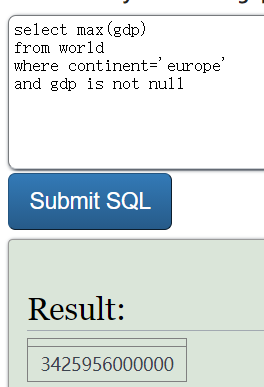
由于欧洲国家中最大的gdp未知，我们先需要写子查询语句

select max(gdp)

from world

where continent='europe'

and gdp is not null



这样我们就得到欧洲国家中最大的GDP了

接着我们查询gdp大于这个值的国家名

写出代码where gdp>子查询

将代码进行组合

【运行代码】

select name

from world

where gdp>(

select max(gdp)

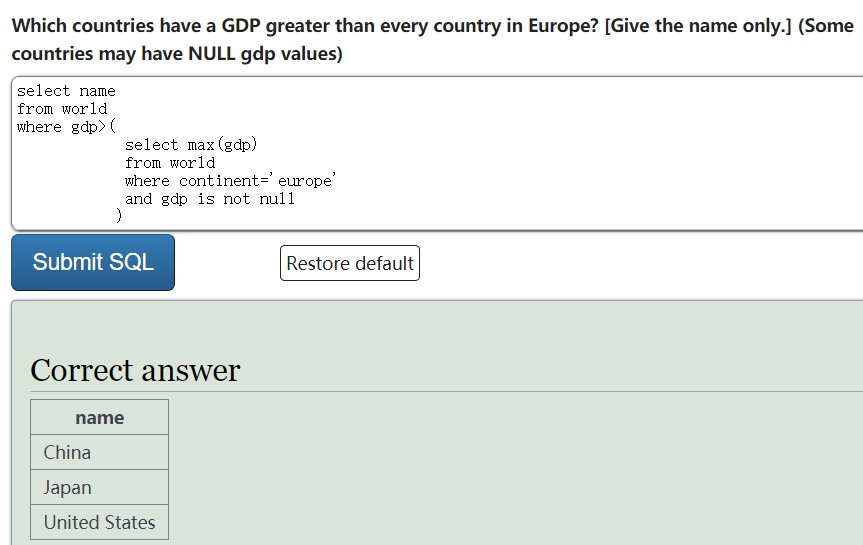
from world

where continent='europe'

and gdp is not null

)

【运行结果】



这题代码中的子查询结果是单行单列的标量子查询

等同于where gdp > 3425956000000

只是欧洲国家中最大的GDP数值我们并不知道，因此需要先构建子查询作为中转，具体的数值3425956000000就由子查询来代替了

* + - * 1. 点击链接<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第三题

【题目】

查询跟阿尔及尼亚（Argentina）和澳大利亚（Australia）在同一大洲的所有国家名称， 并按照国家名称进行排序展示

由于阿尔及尼亚（Argentina）和澳大利亚（Australia）所属大洲未知，所以我们先写子查询得到这两个国家所属的大洲

【运行代码】

select continent

from world

where name='Argentina'

or name='Australia'

【运行结果】



接下来查询属于这两个大洲的国家名称，并按照名字排序，这里没有指定排序降序还是升序，所以使用默认的升序

由于子查询是列子查询，有多个值，所以条件筛选中使用in操作符

【运行代码】

select name,continent

from world

where continent in (

select continent

from world

where name='Argentina'

or name='Australia'

)

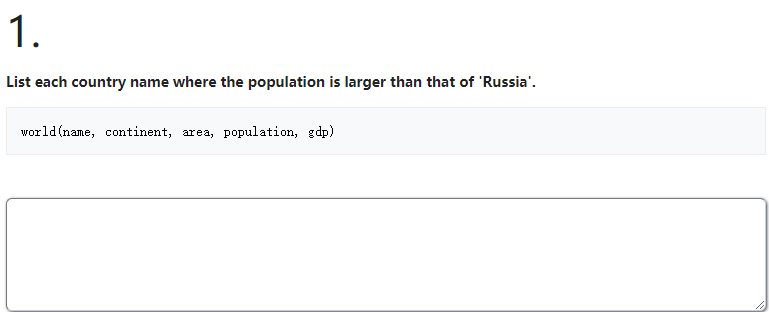
order by name

【运行结果】



* + 1. 【总结】
       1. 子查询本身是一个完整的查询，由括号包裹嵌套在主查询中
       2. 子查询最后返回查询出的结果给主查询
       3. 子查询可以在select，from，where，having子句中使用，但要注意不同子句能接受的子查询种类有差别
       4. 子查询可以多重嵌套
    2. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第一题
          1. 【题目】

查询 人口（population）比俄罗斯Russia更多的国家名（name）



* + - * 1. 【正确答案】

select

name

from world

where population > (

select population

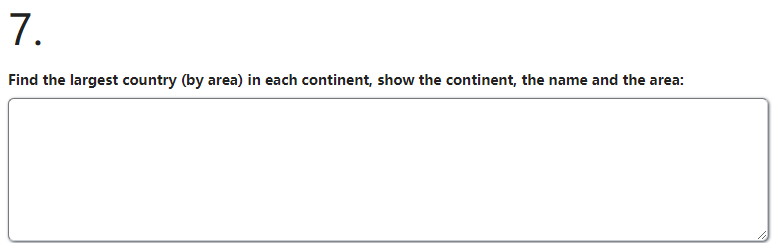
from world

where name = 'Russia'

)

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第七题
         1. 【题目】

查找每个大陆（continent）中最大的国家（按区域area），显示该大陆（continent），名称（name）和区域（area）



* + - * 1. 【正确答案】

select

continent

,name

,area

from world

where (continent,area) in (

select

continent

,max(area)

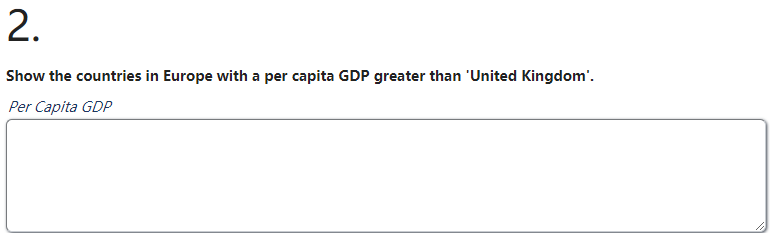
from world

group by continent

)

* + - 1. 【3】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第二题
         1. 【题目】

查询在欧洲Europe中人均gdp大于 英国 'United Kingdom' 的国家名



* + - * 1. 【正确答案】

select

name

from world

where continent = 'Europe'

and gdp / population > (

select

gdp / population

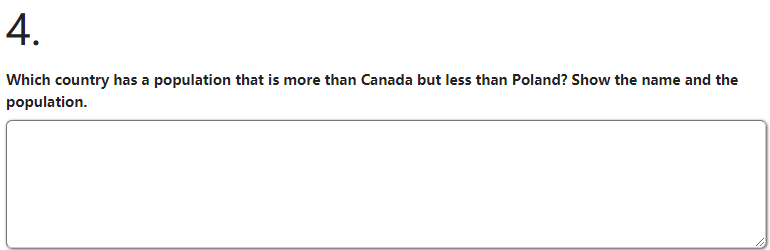
from world

where name = 'United Kingdom'

)

* + - 1. 【4】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第四题
         1. 【题目】

查询国家的人口数（population）超过加拿大（Canada）但少于波兰（Poland）的国家的名称（name）和人口数（population）



* + - * 1. 【正确答案】

select

name

, population

from world

where population > (

select

population

from world

where name = 'Canada'

)

and population < (

select

population

from world

where name = 'Poland'

)

1. 第二天
   1. 主知识点七：表连接
      1. 【知识点引入】
         1. 接下来让我们一起学习表连接join，通过两个或多个表中的列之间的关系连接，再从连接的表中跨表查询数据
            1. 【基础语法】

内连接

select 字段名

from 表格名1 inner join 表格名2 on 表格名1.字段名 = 表格名2.字段名

注意内连接inner可以省略，直接使用join默认为内连接

左连接

select 字段名

from 表格名1 left join 表格名2 on 表格名1.字段名 = 表格名2.字段名

右连接

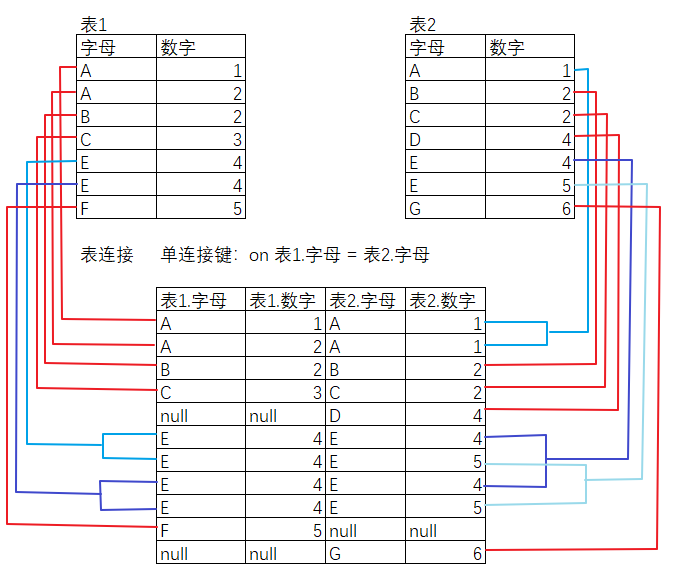
select 字段名

from 表格名1 right join 表格名2 on 表格名1.字段名 = 表格名2.字段名

* + - * 1. 【语法解释】

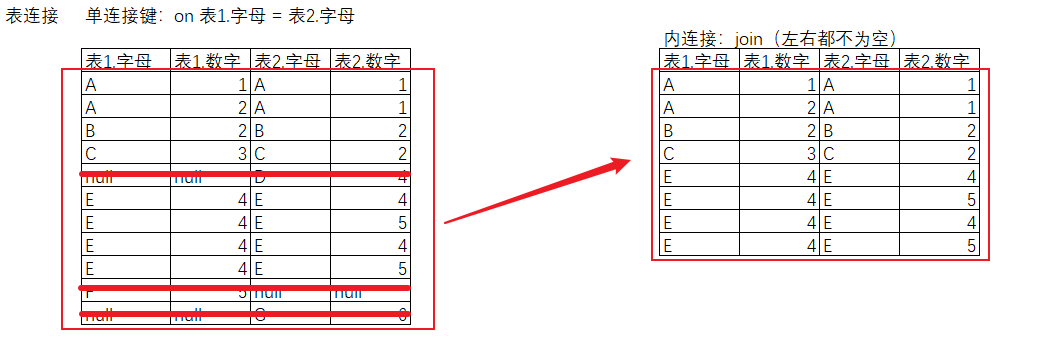
join表连接通过on 表格名1.字段名 = 表格名2.字段名，将两个表格各自的字段等值连接来匹配连接，无法匹配的填充null值

【Excel演示】



inner join内连接，连接两个表留下同时互相匹配上的行得到新表

【Excel演示】



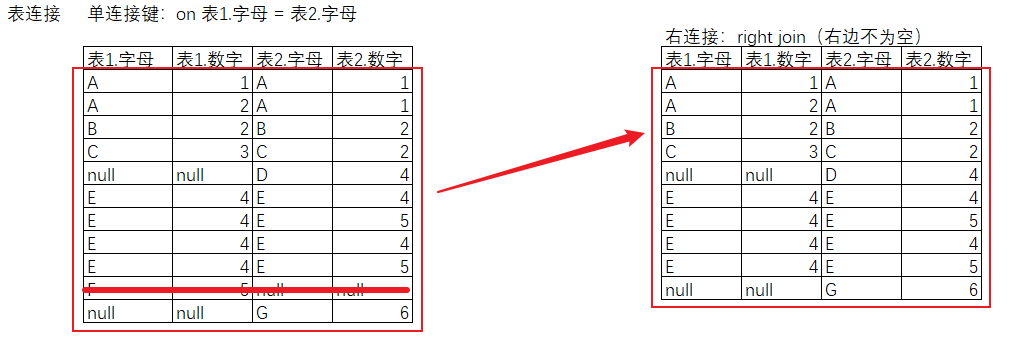
left join左连接，左边的表返回所有行，右边的表只留下匹配上的行得到新表

【Excel演示】



right join右连接，右边的表返回所有行，左边的表只留下匹配上的行得到新表

【Excel演示】



from **表1 left join 表2** on 表1.字段A = 表2.字段B 等价于 from **表2 right join** **表1** on 表1.字段A = 表2.字段B

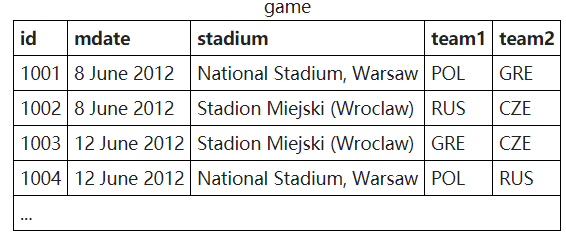
* + 1. 【例题讲解】
       1. 【表与表连接】
          1. 点击链接<https://sqlzoo.net/wiki/The_JOIN_operation>第四题

【题目】

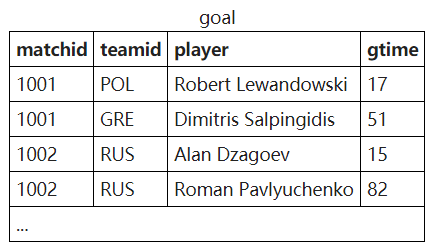
查询有球员名叫Mario进球的队伍1（team1），队伍2（team2）及球员姓名

先来看给的三张表示例

表一：game赛事表。记录的每一场赛事的编号（id），赛事的举办时间（mdate），赛事举办的场地（stadium），参加本场比赛的两个队伍（team1，team2）



表二：goal得分表。记录了球员进球的赛事编号（matchid），队伍编号（teamid），球员名字（player）及进球时间（gtime）



表三：eteam队伍表。记录了每个队伍的编号（id）,队伍名字（teamname）和队伍的教练名字（coach）



题目要查询队伍1，队伍2和球员姓名，队伍1和队伍2来自game表，球员姓名来自goal表，来自不同的表但是要同时查询此时对两张表进行连接，连接键为赛事编号在game表中为id列，在goal表中为matchid列，连接表时可以为表格起别名，如果select中查询的字段名在联查的表中都有一摸一样的名字，就必须要指定该字段来源于哪一张表，如果查询的字段在联查的表中名字唯一，可以不指定字段来源于哪一张表格。

写出代码

select

a.team1

,a.team2

,b.player

from game a

join goal b

on a.id = b.matchid

同时还要查询球员名字为Mario的，添加代码where b.player like 'Mario%';

【运行代码】

select

a.team1

,a.team2

,b.player

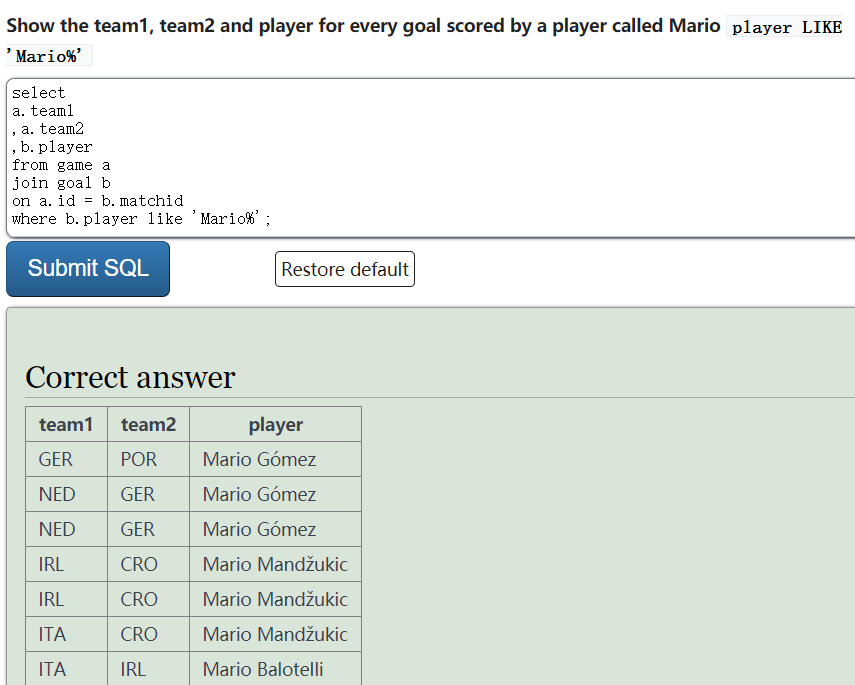
from game a

join goal b

on a.id = b.matchid

where b.player like 'Mario%'

【运行结果】



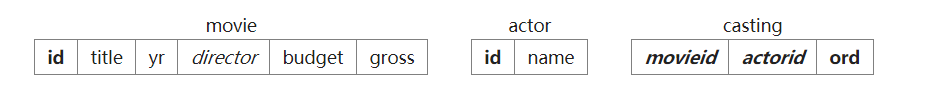
* + - 1. 【多表连结】
         1. 点击链接<https://sqlzoo.net/wiki/More_JOIN_operations>第九题

【题目】

查询演员Harrison Ford出演过的他不是第1主角的电影

首先看这道题涉及的几张表

分别是记录了编号id，电影名title，首影年份yr，导演director，制作费budget，票房收入gross的电影表。记录了演员编号id，演员名字name的演员表。记录了电影编号movieid，演员编号actorid，角色次序ord的角色表。角色次序为1，代表第1主角，角色次序为2，代表第2主角，依次类推。



这道题最终查询电影名title，字段来源于movie表，条件是演员名为Harrison Ford，且出演的角色不是第一主角，涉及到字段name和ord分别来自actor和casting表

movie中的id字段和casting表中的movieid可以匹配连接，actor表中的id字段和casting表中的actorid可以匹配连接，因此需要对三张表进行连接

写出相对应的代码

【运行代码】

select title

from actor a

join casting b

on a.id = b.actorid

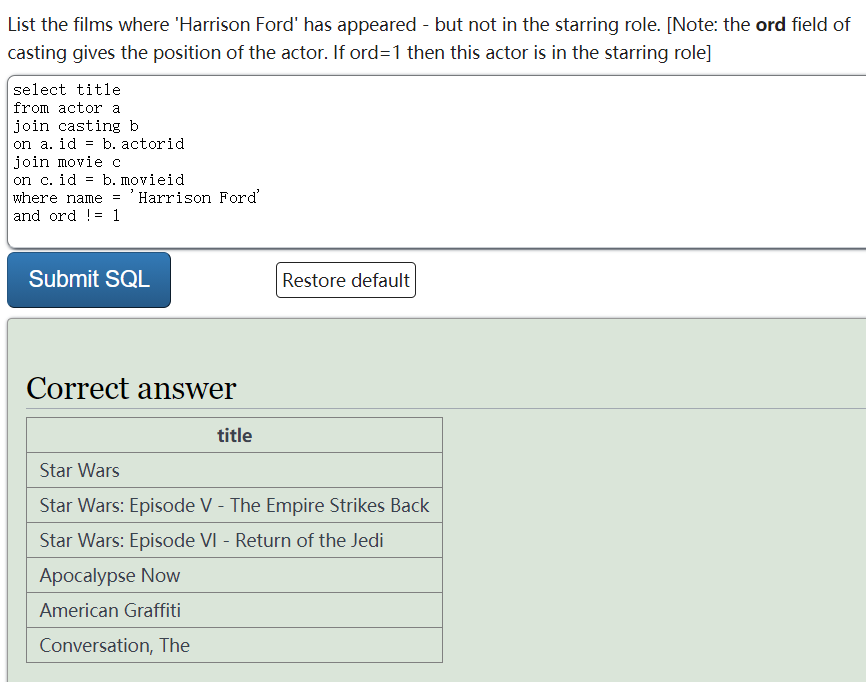
join movie c

on c.id = b.movieid

where name = 'Harrison Ford'

and ord != 1

【运行结果】



* + - 1. 【子查询与表连接】
         1. 点击链接<https://sqlzoo.net/wiki/More_JOIN_operations>第十三题

【题目】

查询至少出演过第1主角15次的演员名

首先演员名，第1主角涉及到表actor和casting

先通过where ord = 1筛选出casting表中所有第1主演的记录

再对casing表根据字段actorid分组计数，对聚合后的数据筛选数量大于等于15的数据

组合写出代码

【运行代码】

select

actorid

,count(movieid) mid

from casting

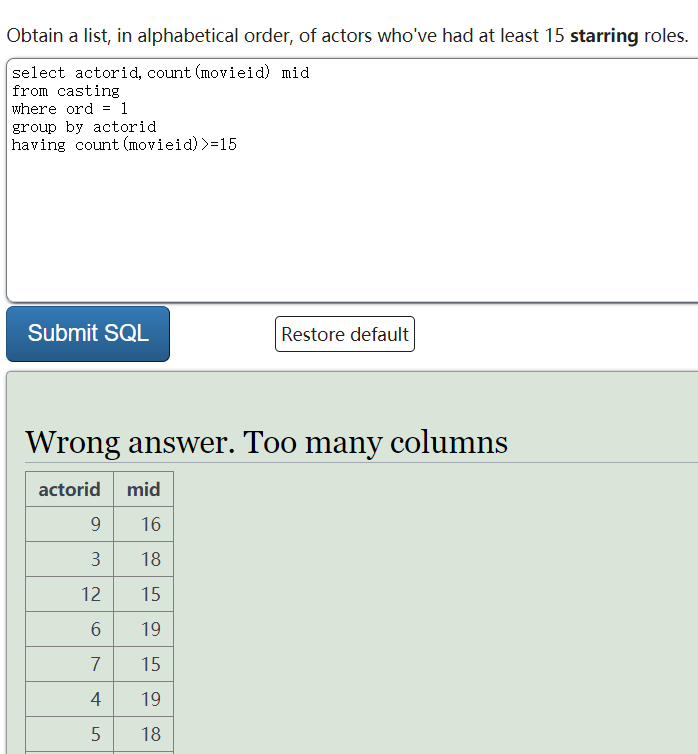
where ord = 1

group by actorid

having count(movieid)>=15

【运行结果】

得到多行多列的查询结果



我们最后查询的是演员名，可以把该子查询作为处理后的表与actor表连接得到与actorid一一对应的演员名称name

【运行代码】

select name

from

(

select

actorid

,count(movieid) mid

from casting

where ord = 1

group by actorid

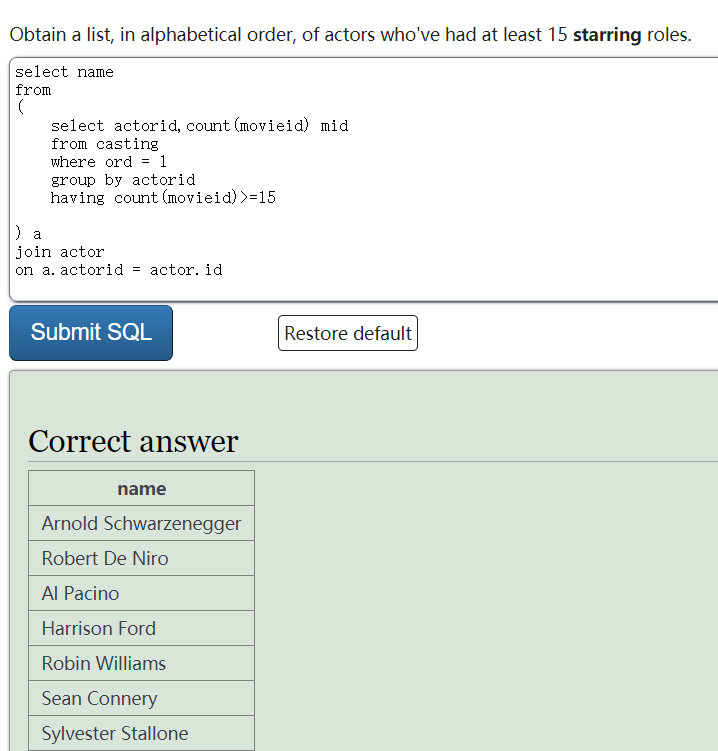
having count(movieid)>=15

) a

join actor

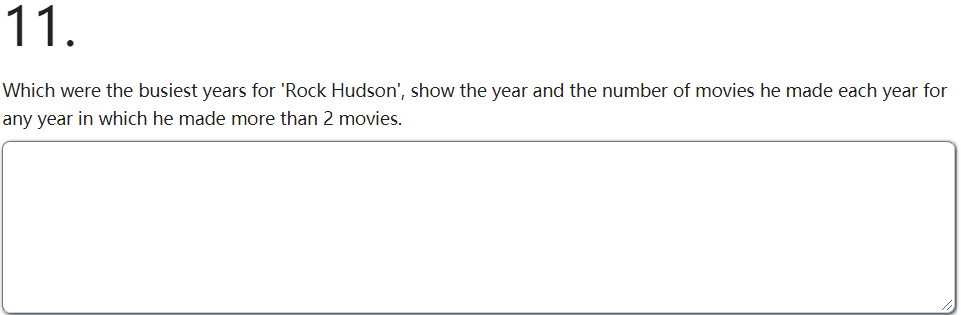
on a.actorid = actor.id

【运行结果】



* + 1. 【总结】
       1. 【内连接inner join语法】
          1. select 字段名
          2. from 表格名1 inner join 表格名2 on 表格名1.字段名 = 表格名2.字段名
          3. 注意内连接inner可以省略，直接使用join默认为内连接
       2. 【左连接left join语法】
          1. select 字段名
          2. from 表格名1 left join 表格名2 on 表格名1.字段名 = 表格名2.字段名
       3. 【右连接right join语法】
          1. select 字段名
          2. from 表格名1 right join 表格名2 on 表格名1.字段名 = 表格名2.字段名
    2. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/More_JOIN_operations>第十一题
          1. 【题目】

查询演员Rock Hudson每年出演超过2部电影的年份和这些年他出演电影的数量。



* + - * 1. 【参考答案】

select

m.yr

,count(m.title)

from movie m

join casting c

on m.id=c.movieid

join actor a

on a.id=c.actorid

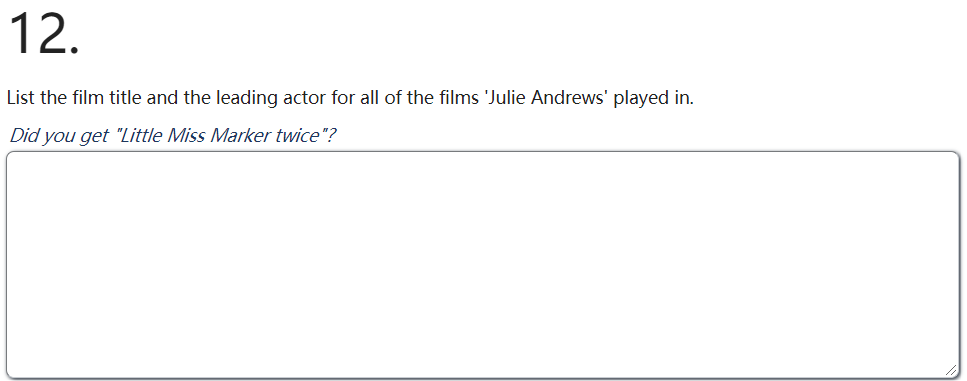
where a.name = 'Rock Hudson'

group by m.yr

having count(m.title)>2

* + - 1. 【2】<https://sqlzoo.net/wiki/More_JOIN_operations>第十二题
         1. 【题目】

查询Julie Andrews出演过的所有电影的电影名和该电影的主演



* + - * 1. 【参考答案】

select

m.title

,a.name

from movie m

join casting c

on m.id = c.movieid

join actor a

on a.id = c.actorid

where c.ord = 1

and c.movieid in (

select

c.movieid

from actor a

join casting c

on a.id = c.actorid

join movie m

on c.movieid = m.id

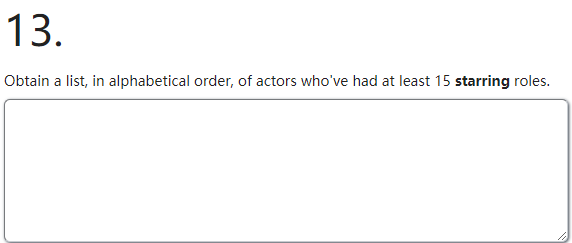
where [a.name](http://a.name/) = 'Julie Andrews'

)

* + - 1. 【3】<https://sqlzoo.net/wiki/More_JOIN_operations>第十三题
         1. 【题目】

查询至少出演过第1主角15次的演员名

和例题讲解的题目一样，**尝试不使用子查询完成**



* + - * 1. 【参考答案】

select name

from casting c

join actor a

on c.actorid = a.id

where ord = 1

group by name

having count(movieid)>=15

* 1. 主知识点八：常用函数
     1. 【知识点引入】
        1. 接着我们再来补充一些mysql中常用函数
           1. 【条件函数语法】

case when

case 字段名 when 值1 then 值2 else 值3 end

【解释】

创建新的一列字段，当某字段等于值1在新列中就等于值2，除此之外的数据在新列中就等于值3

或

case when 字段名 运算符 值1 then 值2 else 值3 end

【解释】

创建新的一列字段，字段名 运算符 值1这一部分可以看成是条件代码，当某字段符合条件时，在新列中就等于值2，除此之外的数据在新列中就等于值3

if

if(字段名 运算符 值1,值2,值3)

【解释】

创建新的一列字段，如果字段满足条件，在新列中就等于值2，否则就等于值3

* + - * 1. 【数据类型转换函数语法】

cast

cast(字段名 as type)

type:date或int等

【解释】

将字段从原来的数据类型转换成指定的数据类型，比如日期date，整数型int等，在type的位置填写指定的参数即可

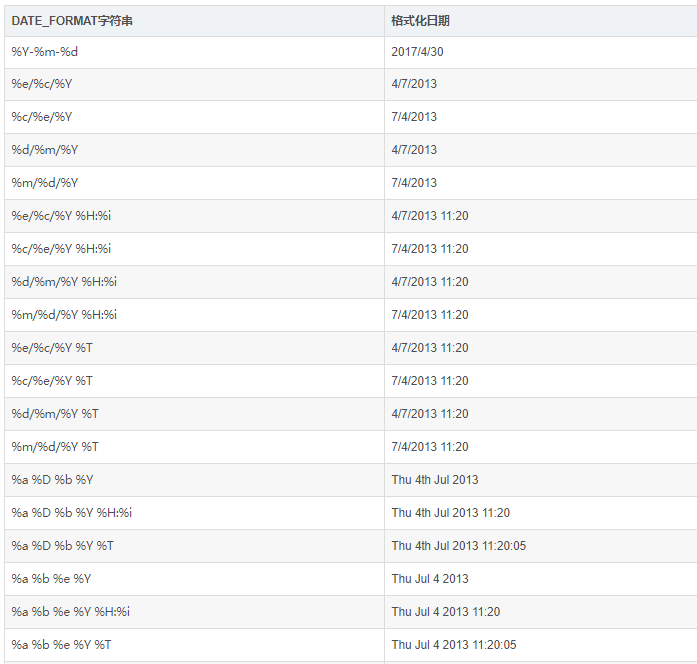
date\_formate

date\_formate(date,formate)

格式



常用日期格式



【解释】

date\_formate用于以不同格式显示日期/时间数据。date 指合法的日期字段，format 规定日期/时间的输出格式，记得格式外加单引号

例如：date\_formate('2021-06-01','%m/%d/%Y')返回数据06/01/2021

* + - * 1. 【日期函数语法】

year、month、day

year(date)

month(date)

day(date)

【解释】

year函数提取数据格式为日期或时间字段的年份

month提取月份，day提取几号

hour, minute, second等函数同理

例如：year('2021-06-01')返回数据2021

now

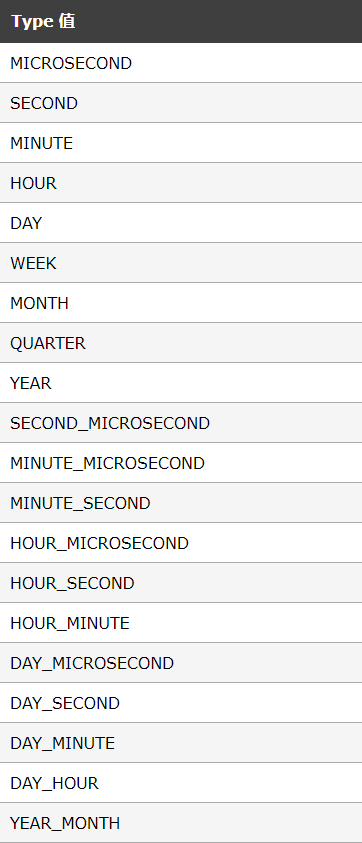
now()

【解释】没有其他参数，返回当前的日期和时间

date\_add

date\_add(date,interval 值 type)

type



【解释】

向日期添加指定的时间间隔

例如：date\_add('2021-06-01',interval 2 day)，向日期函数2021-06-01加了2天，返回值2021-06-03

date\_sub

date\_sub(date,interval 值 type)

【解释】

从日期减去指定的时间间隔

type参数与date\_add相同，使用方法也相同

datediff

datediff(date1,date2)

【解释】

返回两个日期之间的天数，date1-date2

*date1* 和 *date2* 代表是日期或日期加时间的表达式

注意只有值的日期部分参与计算

例如：datediff('2021-06-08','2021-06-01')返回7，datediff('2021-06-01','2021-06-08')返回-7

* + - * 1. 【字符处理函数语法】

round

round(字段名,小数指定位数)

【解释】

round函数将数值字段舍入到指定的小数位数

例如：round(3.15,1)返回数据3.2

concat

concat(字段名1,字段名2,'字符串')

【解释】

concat是字符串拼接函数，用于将多个字符串连接在一起

例如：concat('123',' ','ABC')返回数据123 ABC

left、right

left(字符串,n)

right(字符串,n)

【解释】

left函数返回字符串从左到右n个字符

right函数返回字符串从右到左n个字符

例如：left('123456',2)返回12，right('123456',3)返回456

substring

substring(字符串,x,y)

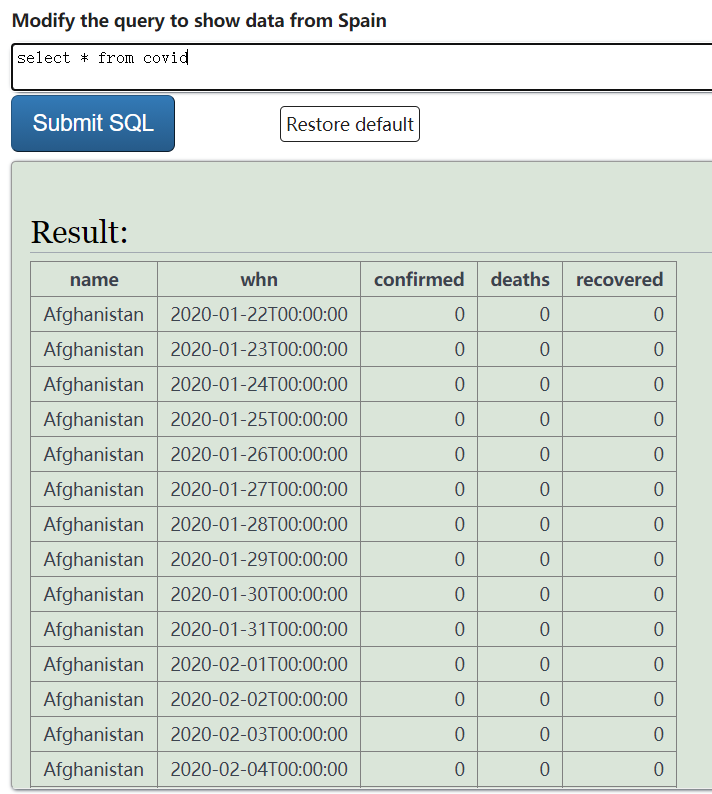
【解释】

substring函数用来截取字符串从x位开始取y位，如果没有y则取从x位开始到最后一位

例如：substring('New York',3,4)返回w Yo，substring('New York',3)返回w York

* + 1. 【例题讲解】
       1. 点击链接<https://sqlzoo.net/wiki/Window_LAG> 第一题
          1. 查看本题数据库表covid

观察本表数据，记录了国家（name）截至时间（whn）的累计确诊（confirmed），累计死亡（deaths），累计治愈（recovered）



* + - * 1. 我们以本题为例试验一下我们刚学习完的函数
        2. case when和if函数

【运行代码】

select

recovered 累计治愈人数

,case when recovered = 1 then 'one' when recovered > 1 then 'more' else '0' end

from covid

where recovered > 0

【运行结果】



* + - * 1. year、month、day函数

【运行代码】

select

whn 更新时间

,year(whn) 年

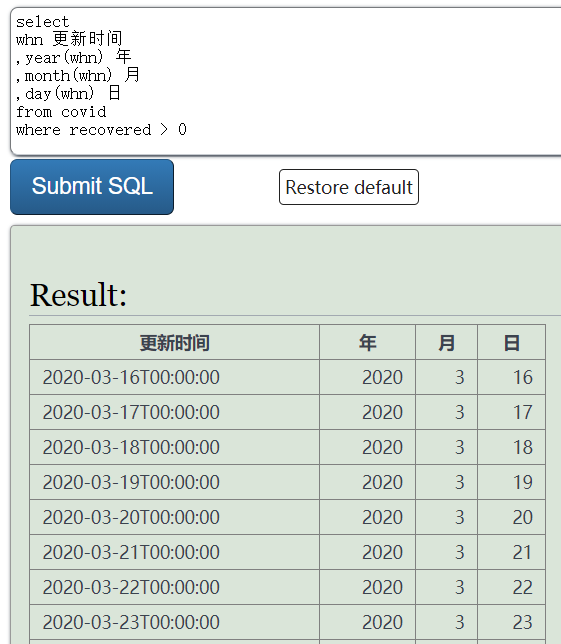
,month(whn) 月

,day(whn) 日

from covid

where recovered > 0

【运行结果】



* + - * 1. date\_add

【运行代码】

select

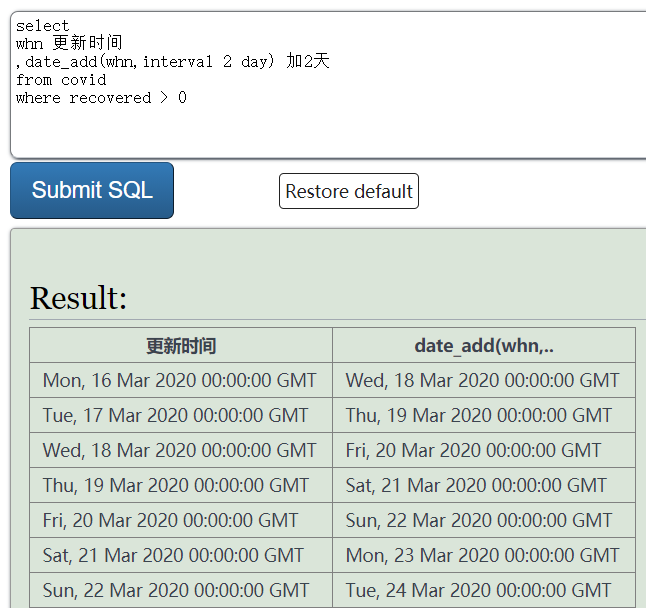
whn 更新时间

,date\_add(whn,interval 2 day) 加2天

from covid

where recovered > 0

【运行结果】



* + - * 1. round和concat嵌套得到百分比数据

【运行代码】

select

confirmed

,deaths

,recovered

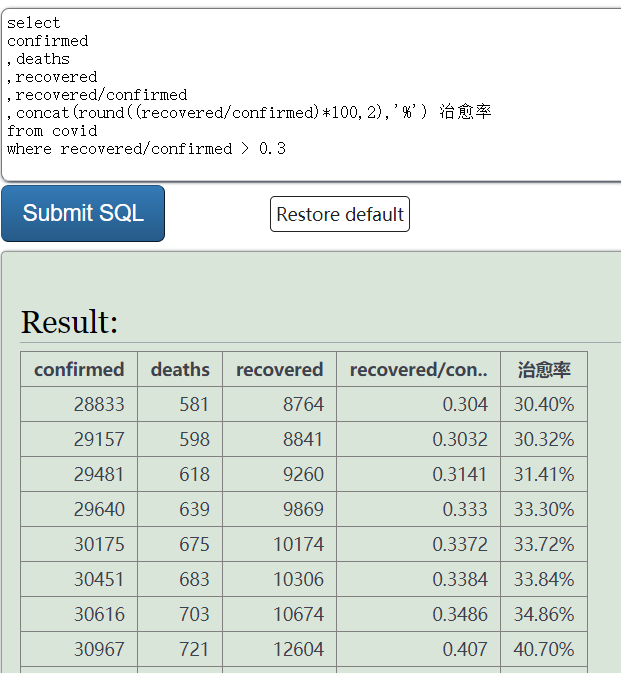
,recovered/confirmed

,concat(round((recovered/confirmed)\*100,2),'%') 治愈率

from covid

where recovered/confirmed > 0.3

【运行结果】



* + - * 1. substring

【运行代码】

select distinct

name

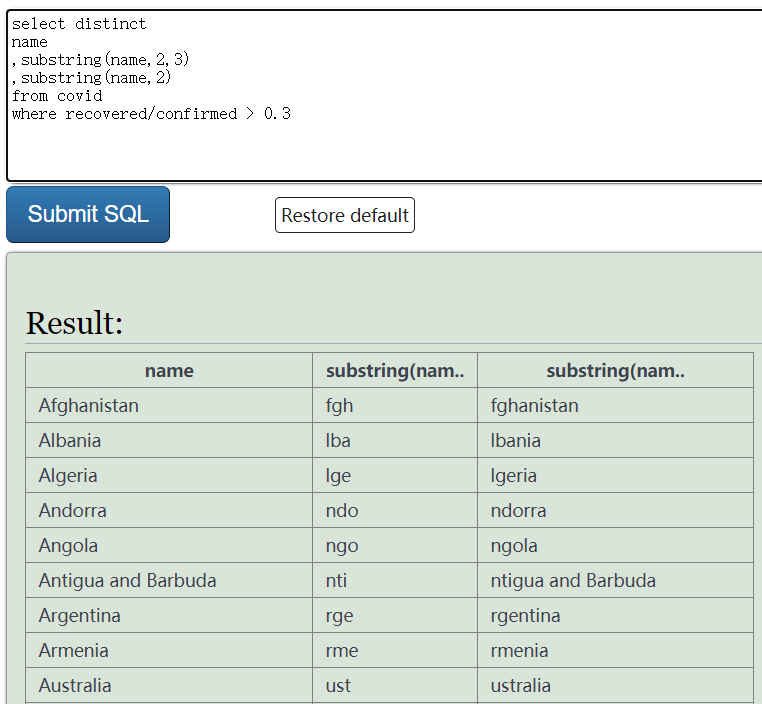
,substring(name,2,3)

,substring(name,2)

from covid

where recovered/confirmed > 0.3

【运行结果】



* + 1. 【总结】
       1. 【条件函数语法】
          1. case when

case 字段名 when 值1 then 值2 else 值3 end

或

case when 字段名 运算符 值1 then 值2 else 值3 end

* + - * 1. if

if(字段名 运算符 值1,值2,值3)

* + - 1. 【数据类型转换函数语法】
         1. cast

cast(字段名 as type)

type:date或int等

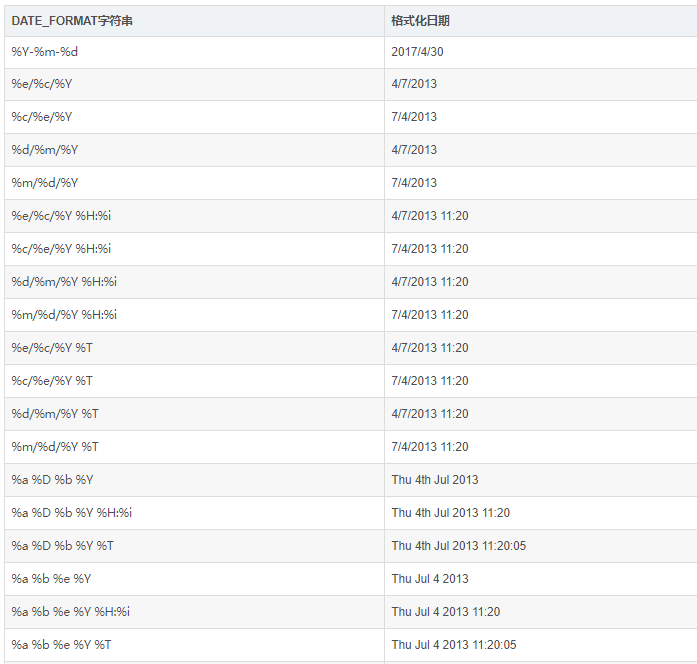
* + - * 1. date\_formate

date\_formate(date,formate)

格式



常用日期格式



* + - 1. 【日期函数语法】
         1. year、month、day

year(date)

month(date)

day(date)

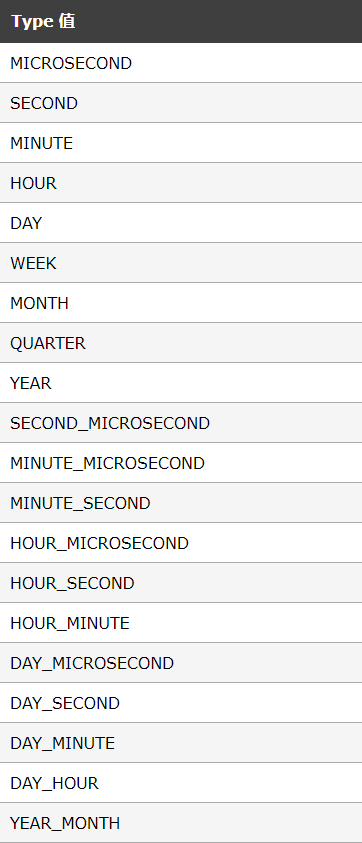
* + - * 1. now

now()

* + - * 1. date\_add

date\_add(date,interval 值 type)

type



* + - * 1. date\_sub

date\_sub(date,interval 值 type)

* + - * 1. datediff

datediff(date1,date2)

* + - 1. 【字符处理函数语法】
         1. round

round(字段名,小数指定位数)

* + - * 1. concat

concat(字段名1,字段名2,'字符串')

字符串和字段都支持哦~

* + - * 1. left、right

left(字符串,n)

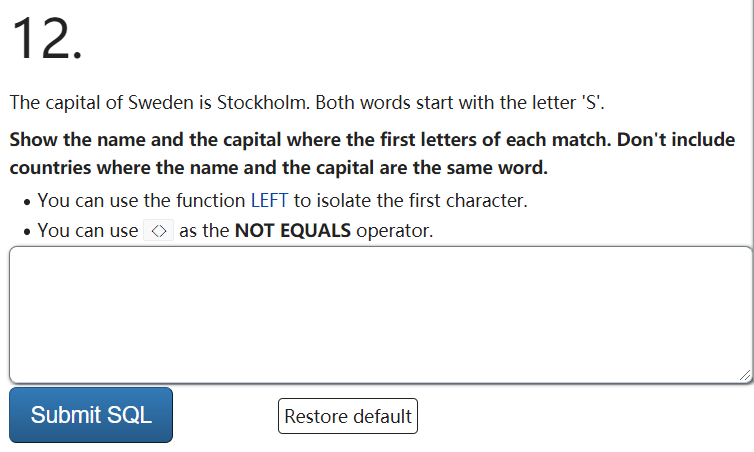
right(字符串,n)

* + - * 1. substring

substring(字符串,x,y)

* + 1. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第十二题
          1. 【题目】

查询国家名称及其首都名称都以相同的字母开头的国家名称及其首都，且不能包括国家名称和首都名称完全相同的情况



* + - * 1. 【参考答案】

select

name

,capital

from world

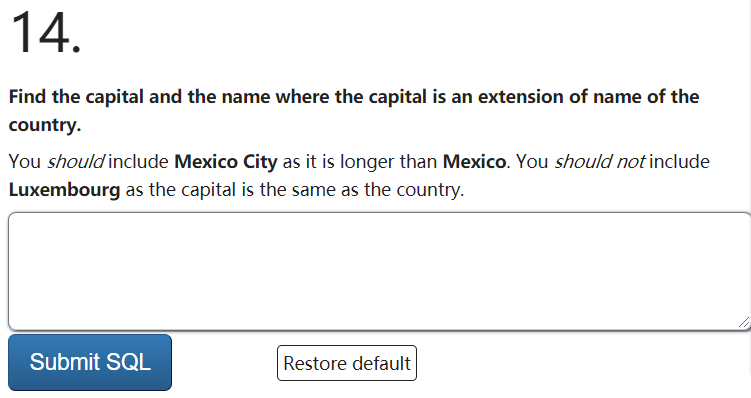
where left(name, 1) = left(capital, 1)

and name != capital;

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_names>第十四题
         1. 【题目】

查询首都和名称，其中首都需是国家名称的扩展

例如：答案中应该包括**墨西哥城（Mexico City），**因为它比**墨西哥（Mexico）**更长，而不应该将**卢森堡（Luxembourg）**包括在内，因为首都名与国家名相同



* + - * 1. 【参考答案】

select

capital

,name

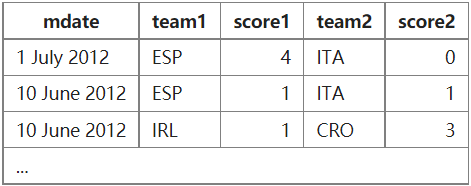
from world

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

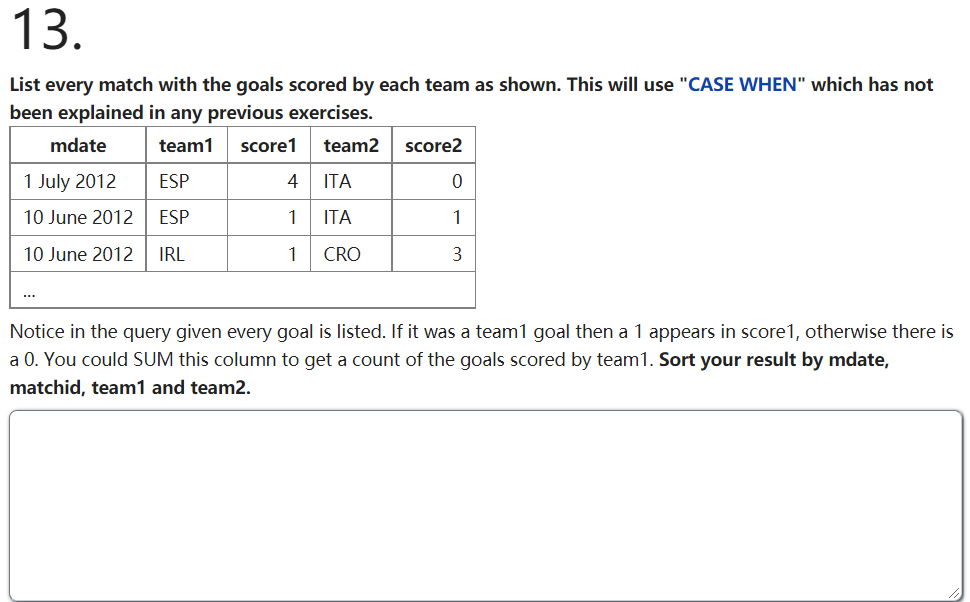
and capital != name;

* + - 1. 【3】<https://sqlzoo.net/wiki/The_JOIN_operation>第十三题
         1. 【题目】

查询每场比赛，每个球队的得分情况，按照以下格式显示



最后按照举办时间（mdate）、赛事编号（matchid）、队伍1（team1）和队伍2（team2）排序



* + - * 1. 【参考答案】

select

ga.mdate

,ga.team1

,sum(case when ga.team1=go.teamid then 1 else 0 end) score1

,ga.team2

,sum(case when ga.team2=go.teamid then 1 else 0 end) score2

from game ga

left join goal go

on ga.id = go.matchid

group by ga.mdate,ga.team1,ga.team2

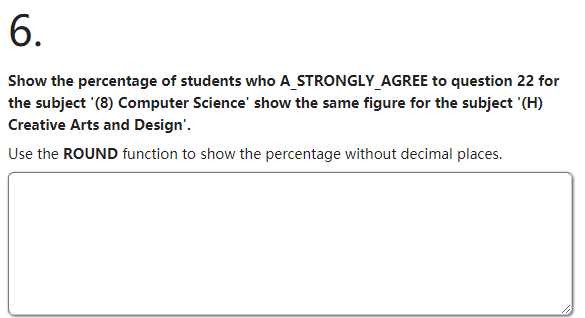
order by ga.mdate, go.matchid, ga.team1, ga.team2

* + - 1. 【4】<https://sqlzoo.net/wiki/NSS_Tutorial>第六题
         1. 【题目】

查询 来自'(8) Computer Science' 和'(H) Creative Arts and Design' 专业的学生，在回答问题22（Q22）时，表示强烈同意（A\_STRONGLY\_AGREE）的各专业人数占比

精确到个位数，不需要百分号

自行阅读题目涉及的表格哦~



* + - * 1. 【参考答案】

select

subject

, round(sum(A\_STRONGLY\_AGREE \* response) / sum(response),0)

from nss

WHERE (question='Q22' and subject='(8) Computer Science' )

or (question='Q22' and subject = '(H) Creative Arts and Design')

group by subject

* 1. 主知识点九：窗口函数
     1. 【知识点引入】
        1. 最后我们要来学习窗口函数
        2. 【标准语法】
           1. 窗口函数 over (partition by 用于分组的字段名 order by 用于排序的字段名)
           2. 常用窗口函数



* + 1. 【例题讲解】
       1. 我们以排序窗口函数rank()over()为例
       2. 窗口函数rank()over()是在指定分区(partition by)对指定字段排序(order by)然后依次赋予排名的函数
       3. 【指定分区partition by和排序order by】
          1. 点击链接<https://sqlzoo.net/wiki/Window_functions> 第三题

【题目】

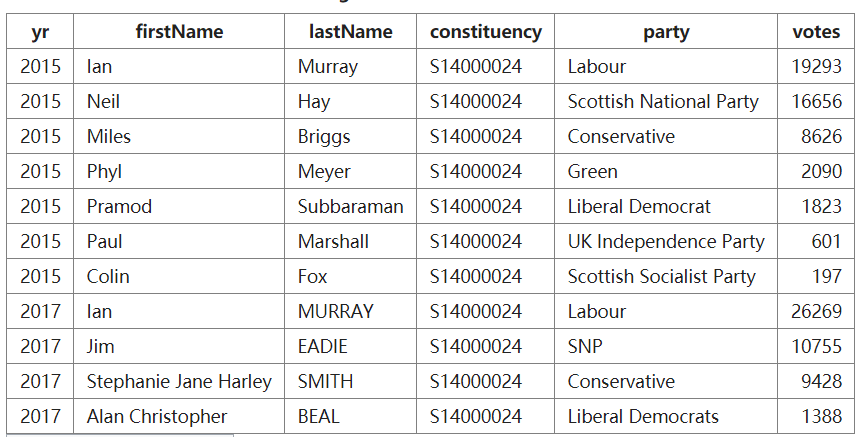
查询每一年S14000021选区中所有候选人所在的团体party和选票数votes

并且对每一年中的所有候选人根据选票数的高低进行排名posn，选票数最高则为第一名，后续以此类推

最后根据团体party和年份yr排序

先查看题目表格

记录了每一年yr，候选人名字firstname和姓氏lastname,所属选区constituency和团体party，以及选票数votes



查询每一年S14000021选区中所有候选人所在的团体party和选票数votes，写出代码

select

yr

,party

, votes

from ge

where constituency = 'S14000021'

order by party,yr

还需要对筛选出的S14000021选区中的每一年的数据分区然后根据票数高到低排序,用desc降序

使用窗口函数

rank()over(partition by yr order by votes desc) as posn

将窗口函数写在select子句后

【运行代码】

select

yr

,party

, votes

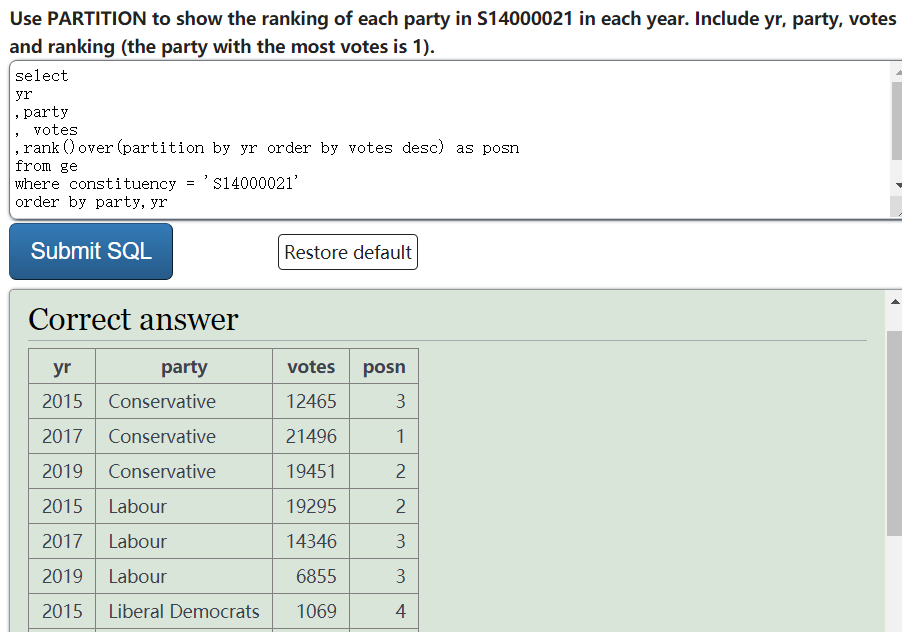
,rank()over(partition by yr order by votes desc) as posn

from ge

where constituency = 'S14000021'

order by party,yr

【运行结果】



【excel演示】

通过以下代码得到表格

select

yr

,party

, votes

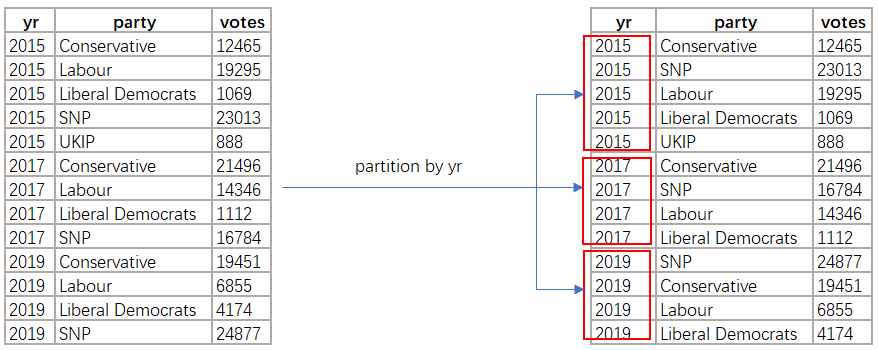
from ge

where constituency = 'S14000021'

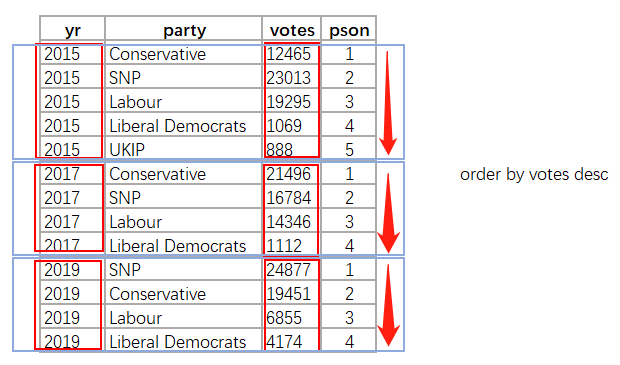
order by party,yr

对窗口函数rank()over(partition by yr order by votes desc) as posn进行拆解

partition by yr对年份数据分区



order by votes desc对分区内的选票数从高到低排序并给它一个排名的编号，最后得到pson列



* + - 1. 【from 子查询和窗口函数】
         1. 点击链接<https://sqlzoo.net/wiki/Window_functions> 第五题

【题目】

查询2017年所有在爱丁堡的选区当选议员所在的选区即对应的团队party

已知爱丁堡选区编号为S14000021至S14000026

当选议员，即各选区得票数最高的候选人

先对2017年选区编号为S14000021至S14000026的各选区分区，然后对每个选区的得票数从高到底排名

查询涉及到选区constituency、团队patry、选票数votes

写出子查询

select

constituency

,party

,votes

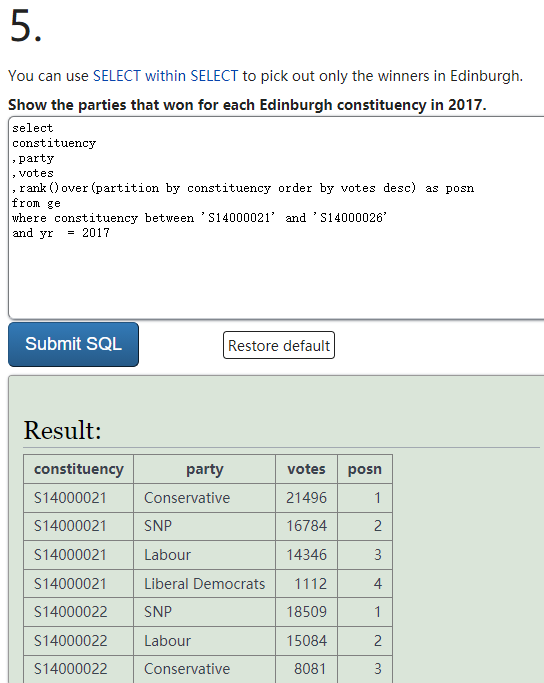
,rank()over(partition by constituency order by votes desc) as posn

from ge

where constituency between 'S14000021' and 'S14000026'

and yr = 2017

【运行结果】



各选区得票数最高的候选人，即筛选条件posn = 1

题目要求选区constituency即团队party

【运行代码】

select

constituency

,party

from

(

select

constituency

,party

,votes

,rank()over(partition by constituency order by votes desc) as posn

from ge

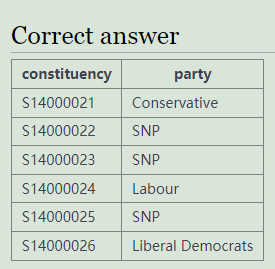
where constituency between 'S14000021' and 'S14000026'

and yr = 2017

) RK

where RK.posn=1

【运行结果】



注意from后的子查询一定要有别名

对窗口函数结果筛选一定要再嵌套一个查询，否则会出现语法错误

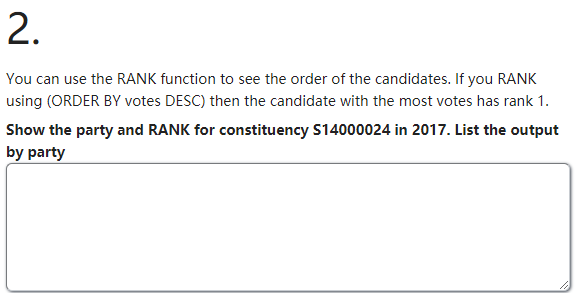
* + 1. 【总结】
       1. 【窗口函数语法】
          1. 窗口函数 over (partition by 用于分区的字段名 order by 用于排序的字段名)
       2. 常用窗口函数



* + 1. 【练习题】
       1. 【1】<https://sqlzoo.net/wiki/Window_functions>第二题
          1. 【题目】

查询2017年选区为 'S14000024' 的所有候选人所在团体（party）和其得到选票数（votes）、还有候选人得票数在选区内对应的的排名

结果按团队party排序



* + - * 1. 【参考答案】

select

party

,votes

,rank() over (order by votes desc) as posn

from ge

where constituency = 'S14000024'

and yr = 2017

order by party

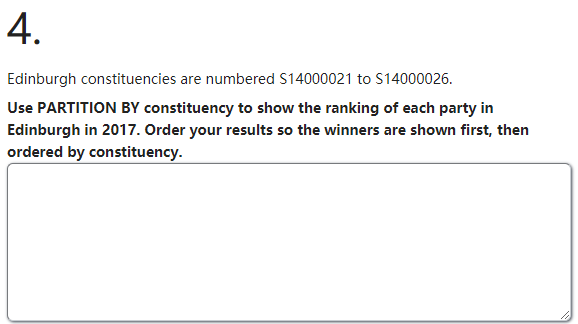
* + - 1. 【2】<https://sqlzoo.net/wiki/Window_functions>第四题
         1. 【题目】

查询2017年爱丁堡各个选区内各团队的排名情况

按照排名、选区排序

排名情况包括选区constituency，团队party，选票数votes，排名posn

已知爱丁堡选区编号为S14000021至S14000026



* + - * 1. 【参考答案】

select

constituency

,party

,votes

,rank()over(partition by constituency order by votes desc) as posn

from ge

where constituency between 'S14000021' and 'S14000026'

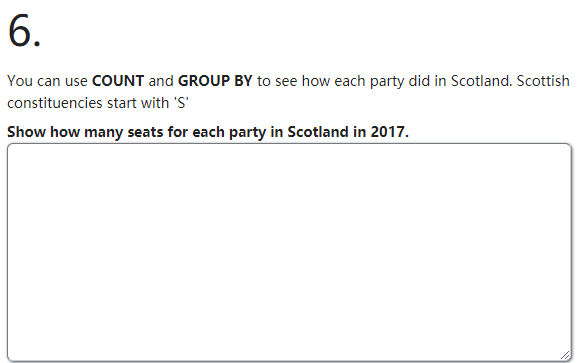
and yr = 2017

order by posn,constituency

* + - 1. 【3】<https://sqlzoo.net/wiki/Window_functions>第六题
         1. 【题目】

查询2017年在苏格兰的选区内（选区编号以S开头的）每个团体获得的席位数

选区候选人得票数为第1名，即获得该选区席位



* + - * 1. 【参考答案】

select

party

,count(\*)

from

(

select

constituency

,party

,votes

,rank()over(partition by constituency order by votes desc) as posn

from ge

where constituency like 'S%'

and yr = 2017

) rk

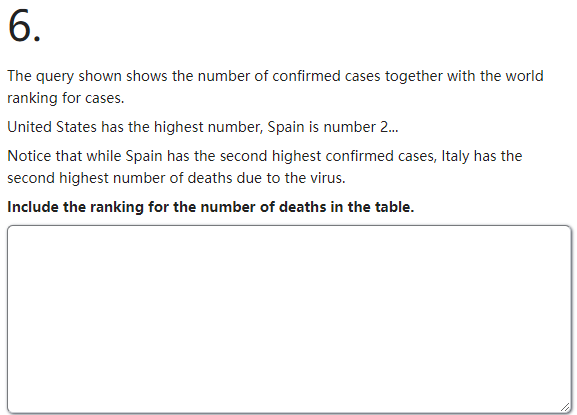
where rk.posn=1

group by rk.party

* + - 1. 【4】<https://sqlzoo.net/wiki/Window_LAG>第六题
         1. 【题目】

查询更新时间为2020年4月20日的国家名，确诊人数，确诊人数排名，死亡人数，死亡人数排名

按照确证人数降序排名



* + - * 1. 【参考答案】

select

name

,confirmed

,rank()over(order by confirmed desc) rc

,deaths

,rank()over(order by deaths desc) deathrc

from covid

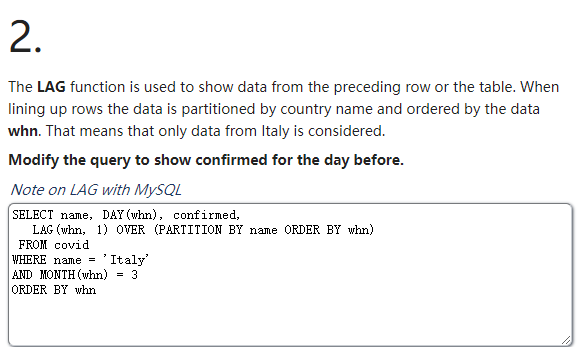
where whn = '2020-04-20'

order by confirmed desc

* + - 1. 【5】<https://sqlzoo.net/wiki/Window_LAG>第二题
         1. 【题目】

修改给出的代码，查询三月份意大利每天的确诊人数和前一天的确诊人数

并按更新时间排序



* + - * 1. 【窗口函数LAG简介】

**LAG(列名,n,m)**: 当前记录前面第n行记录的<列名>的值，没有则默认值为m；如果不带参数n,m，则查找当前记录前面第一行的记录<列名>的值，没有则默认值为null

* + - * 1. 【参考答案】

select

name

, day(whn)

, confirmed

,lag(confirmed,1) over (partition by name order by whn) lag

from covid

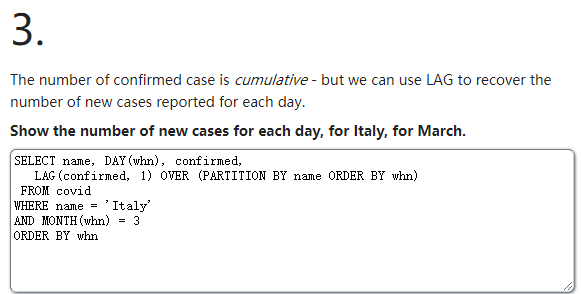
where name = 'italy'

and month(whn) = 3

order by whn

* + - 1. 【6】<https://sqlzoo.net/wiki/Window_LAG>第三题
         1. 【题目】

修改给出的代码，使用LAG函数查询三月份意大利每日新增确诊数



* + - * 1. 【参考答案】

select

name

, day(whn)

, (confirmed -lag(confirmed, 1) over (partition by name order by whn)) new

from covid

where name = 'italy'

and month(whn) = 3

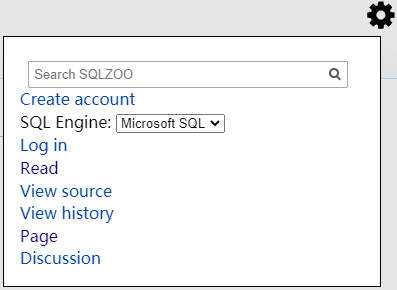
order by whn

* + - 1. 【7】<https://sqlzoo.net/wiki/Window_LAG>第四题 **这题务必认真复习！！！**
         1. 【切换数据库操作】

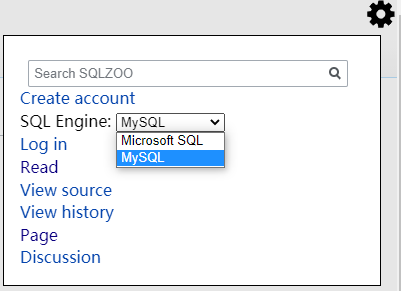
找到页面最右上角的齿轮图标



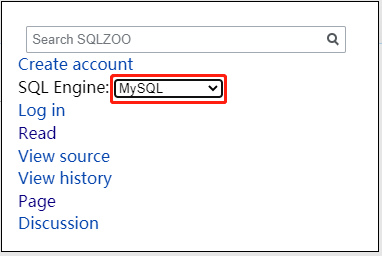
点击图标



切换数据库引擎为MySQL



mysql支持函数date\_format()和weekday()，但Microsoft SQl不支持

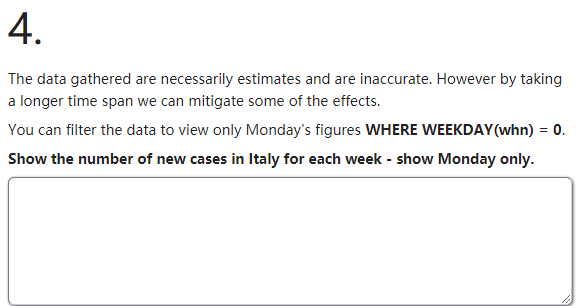


* + - * 1. 【题目】

查询意大利每周新增确诊数（显示每周一的数值 weekday(whn) = 0）

最后显示国家名，标准日期（2020-01-27），每周新增人数

按照更新时间排序



* + - * 1. 【参考答案】

select

name

, date\_format(whn,'%Y-%m-%d') date

, (confirmed - lag(confirmed,1)over(partition by name order by whn)) New

from covid

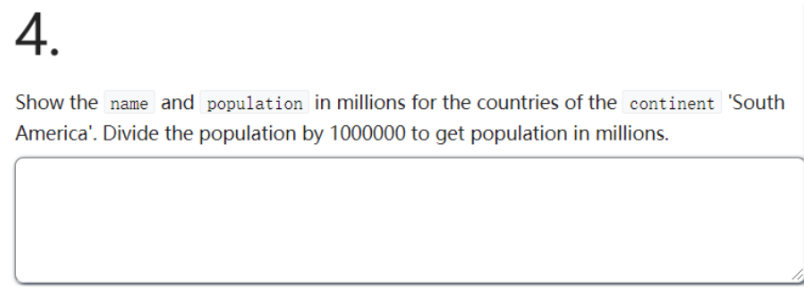
where name = 'Italy'

and weekday(whn) = 0

order by whn

1. 练习题合集
   1. 0604第一天
      1. 主知识点一：select&from
         1. 自己完整敲一遍知识点中出现的代码吧~
      2. 主知识点二：where
         1. 【1】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第四题
            1. 【题目】

查询南美洲（south america）所有国家的名称以及它们以百万（1000000 ）为单位的人口数量



* + - * 1. 【正确答案】

select

name

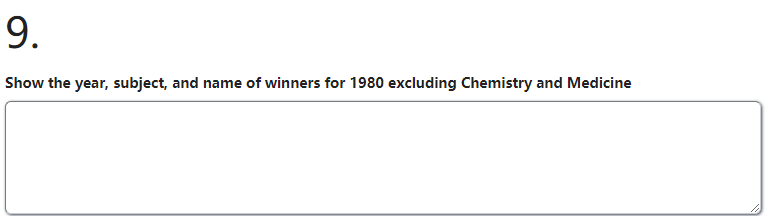
, population/1000000 population\_in\_millions

from world

where continent = 'South America';

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第九题
         1. 【题目】

查询1980年除诺贝尔化学奖和诺贝尔医学奖外其余奖项获奖者的所有信息。



* + - * 1. 【正确答案】

select \*

from nobel

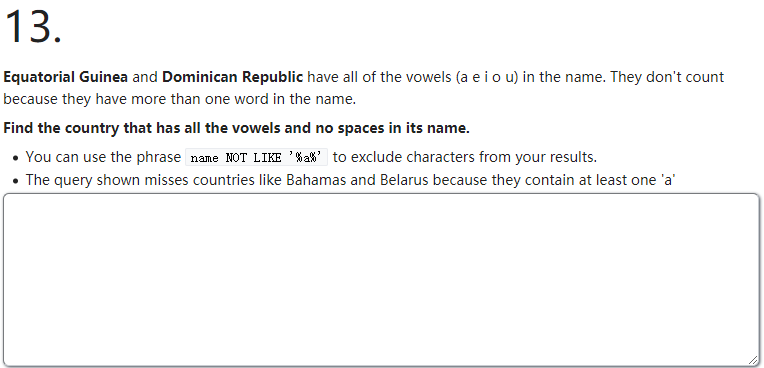
where yr = 1980

and subject not in ('Chemistry','Medicine')

* + - 1. 【3】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第十三题
         1. 【题目】

查询既包含有所有元音字母（a,e,i,o,u），同时国家名中没有空格的国家，最后显示他们的名字

例如，赤道几内亚**Equatorial Guinea** 和 多米尼加共和国**Dominican Republic** ，都包括有五个元音字母（a,e,i,o,u)，但这些国家不会被记录，因为国家名中，由两个单词构成（有空格）



* + - * 1. 【正确答案】

select name

from world

where name like '%a%'

and name like '%e%'

and name like '%i%'

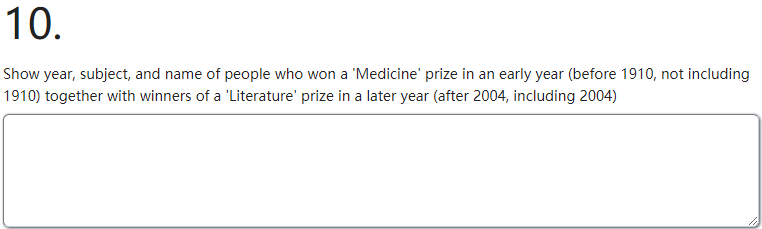
and name like '%o%'

and name like '%u%'

and name not like '% %';

* + - 1. 【4】<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第十题
         1. 【题目】

查询1910年以前（不含1910）诺贝尔医学奖获得者和2004年及以后诺贝尔文学奖获得者的所有信息



* + - * 1. 【正确答案】

select \*

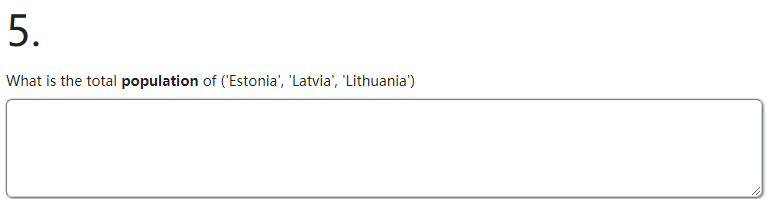
from nobel

where (subject = 'Medicine' and yr < 1910)

or (subject = 'Literature' and yr >= 2004)

* + 1. 主知识点三：聚合函数、group by&having
       1. 【1】<https://sqlzoo.net/wiki/SUM_and_COUNT>第五题
          1. 【题目】

计算Estonia, Latvia, Lithuania这几个国家的总人口数



* + - * 1. 【正确答案】

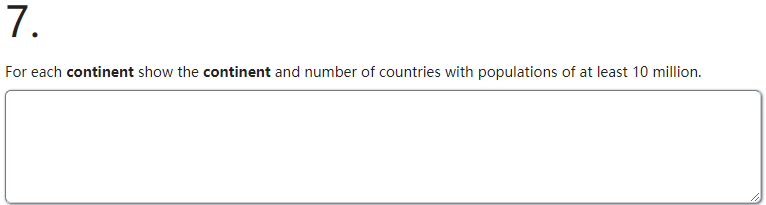
select sum(population)

from world

where name in ('Estonia', 'Latvia', 'Lithuania')

* + - 1. 【2】<https://sqlzoo.net/wiki/SUM_and_COUNT>第七题
         1. 【题目】

查询每个大洲和该大洲里人口数超过1千万的国家的数量



* + - * 1. 【正确答案】

select

continent

,count(name) 'number of countries'

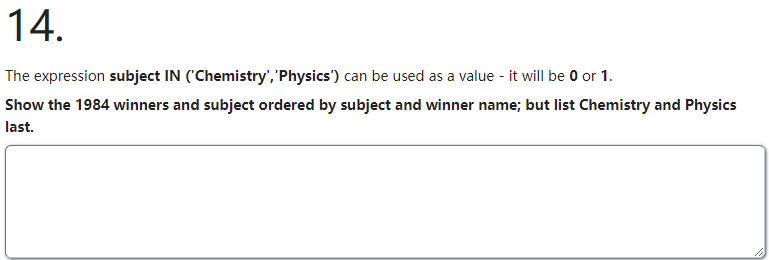
from world

where population >= 10000000

group by continent

* + 1. 主知识点四：order by
       1. 【1】点击链接<https://sqlzoo.net/wiki/SELECT_from_Nobel_Tutorial>第十四题
          1. 【题目】

查询1984年所有获奖者的姓名和奖项。结果将诺贝尔化学奖和物理学奖排在最后，然后按照奖项排序，再按照获奖者姓名排序



* + - * 1. 【正确答案】

select

winner

,subject

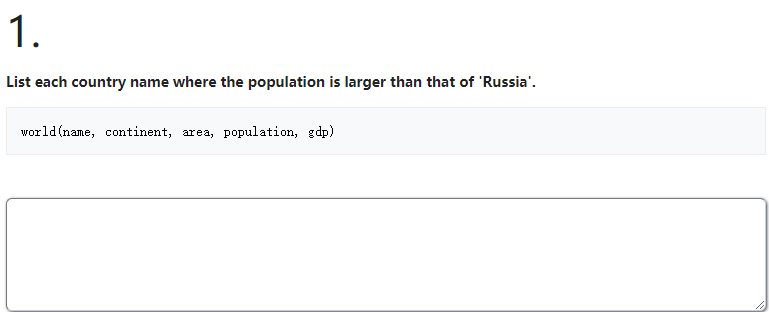
from nobel

where yr = 1984

order by subject in ('chemistry','physics') , subject, winner

* + 1. 主知识点五：limit
    2. 主知识点六：子查询
       1. 【1】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第一题
          1. 【题目】

查询 人口（population）比俄罗斯Russia更多的国家名（name）



* + - * 1. 【正确答案】

select

name

from world

where population > (

select population

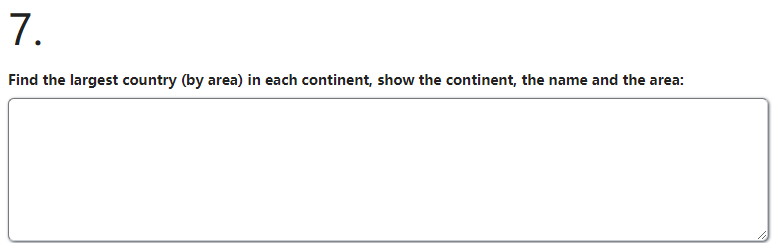
from world

where name = 'Russia'

)

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第七题
         1. 【题目】

查找每个大陆（continent）中最大的国家（按区域area），显示该大陆（continent），名称（name）和区域（area）



* + - * 1. 【正确答案】

select

continent

,name

,area

from world

where (continent,area) in (

select

continent

,max(area)

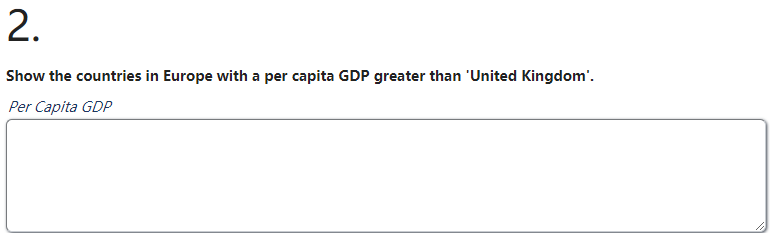
from world

group by continent

)

* + - 1. 【3】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第二题
         1. 【题目】

查询在欧洲Europe中人均gdp大于 英国 'United Kingdom' 的国家名



* + - * 1. 【正确答案】

select

name

from world

where continent = 'Europe'

and gdp / population > (

select

gdp / population

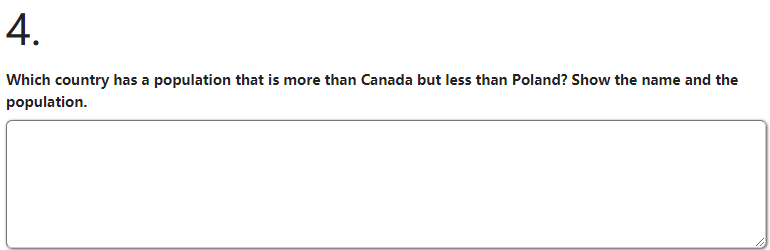
from world

where name = 'United Kingdom'

)

* + - 1. 【4】<https://sqlzoo.net/wiki/SELECT_within_SELECT_Tutorial>第四题
         1. 【题目】

查询国家的人口数（population）超过加拿大（Canada）但少于波兰（Poland）的国家的名称（name）和人口数（population）



* + - * 1. 【正确答案】

select

name

, population

from world

where population > (

select

population

from world

where name = 'Canada'

)

and population < (

select

population

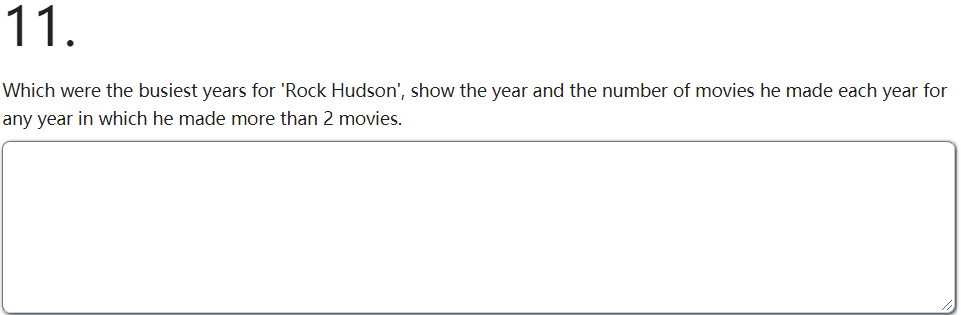
from world

where name = 'Poland'

)

* 1. 0605第二天
     1. 主知识点七：表连接
        1. 【1】<https://sqlzoo.net/wiki/More_JOIN_operations>第十一题
           1. 【题目】

查询演员Rock Hudson每年出演超过2部电影的年份和这些年他出演电影的数量。



* + - * 1. 【参考答案】

select

m.yr

,count(m.title)

from movie m

join casting c

on m.id=c.movieid

join actor a

on a.id=c.actorid

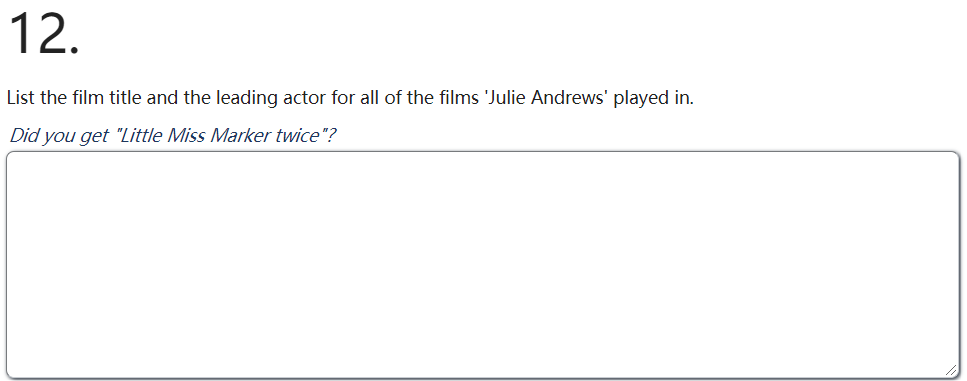
where a.name = 'Rock Hudson'

group by m.yr

having count(m.title)>2

* + - 1. 【2】<https://sqlzoo.net/wiki/More_JOIN_operations>第十二题
         1. 【题目】

查询Julie Andrews出演过的所有电影的电影名和该电影的主演



* + - * 1. 【参考答案】

select

m.title

,a.name

from movie m

join casting c

on m.id = c.movieid

join actor a

on a.id = c.actorid

where c.ord = 1

and c.movieid in (

select

c.movieid

from actor a

join casting c

on a.id = c.actorid

join movie m

on c.movieid = m.id

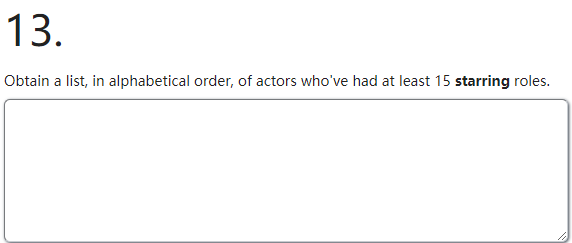
where [a.name](http://a.name/) = 'Julie Andrews'

)

* + - 1. 【3】<https://sqlzoo.net/wiki/More_JOIN_operations>第十三题
         1. 【题目】

查询至少出演过第1主角15次的演员名

和例题讲解的题目一样，**尝试不使用子查询完成**



* + - * 1. 【参考答案】

select name

from casting c

join actor a

on c.actorid = a.id

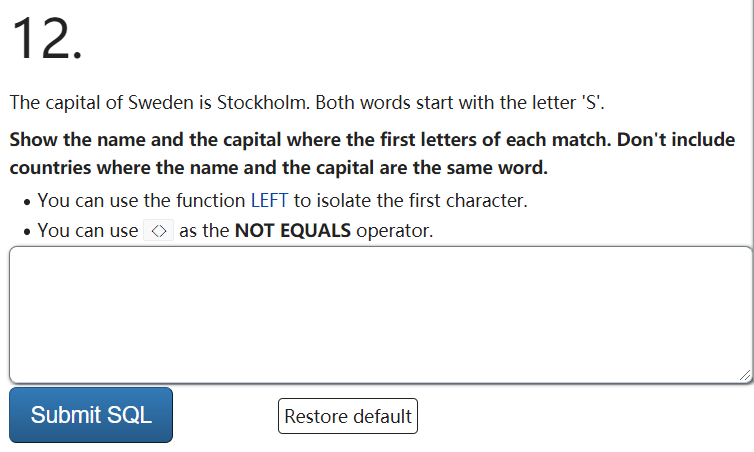
where ord = 1

group by name

having count(movieid)>=15

* + 1. 主知识点八：常用函数
       1. 【1】<https://sqlzoo.net/wiki/SELECT_from_WORLD_Tutorial>第十二题
          1. 【题目】

查询国家名称及其首都名称都以相同的字母开头的国家名称及其首都，且不能包括国家名称和首都名称完全相同的情况



* + - * 1. 【参考答案】

select

name

,capital

from world

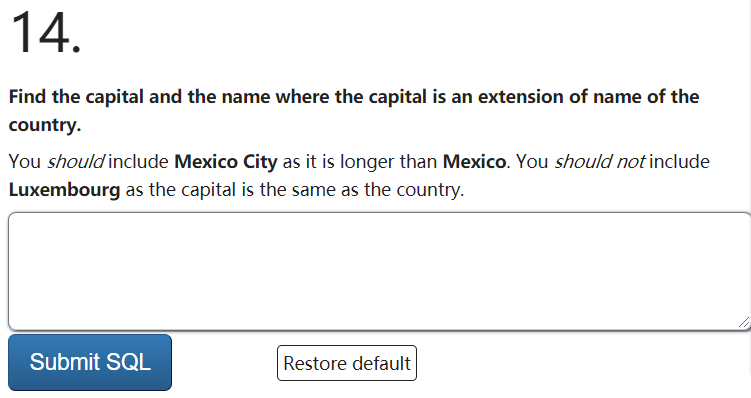
where left(name, 1) = left(capital, 1)

and name != capital;

* + - 1. 【2】<https://sqlzoo.net/wiki/SELECT_names>第十四题
         1. 【题目】

查询首都和名称，其中首都需是国家名称的扩展

例如：答案中应该包括**墨西哥城（Mexico City），**因为它比**墨西哥（Mexico）**更长，而不应该将**卢森堡（Luxembourg）**包括在内，因为首都名与国家名相同



* + - * 1. 【参考答案】

select

capital

,name

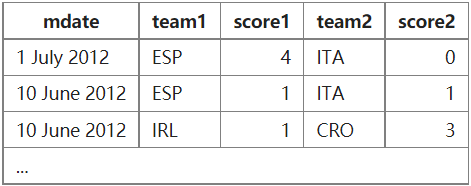
from world

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

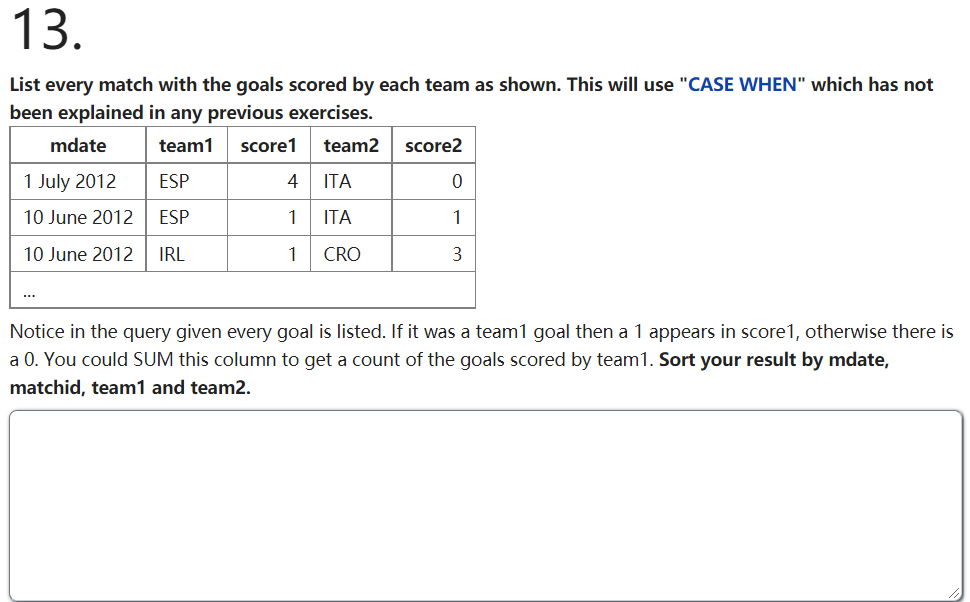
and capital != name;

* + - 1. 【3】<https://sqlzoo.net/wiki/The_JOIN_operation>第十三题
         1. 【题目】

查询每场比赛，每个球队的得分情况，按照以下格式显示



最后按照举办时间（mdate）、赛事编号（matchid）、队伍1（team1）和队伍2（team2）排序



* + - * 1. 【参考答案】

select

ga.mdate

,ga.team1

,sum(case when ga.team1=go.teamid then 1 else 0 end) score1

,ga.team2

,sum(case when ga.team2=go.teamid then 1 else 0 end) score2

from game ga

left join goal go

on ga.id = go.matchid

group by ga.mdate,ga.team1,ga.team2

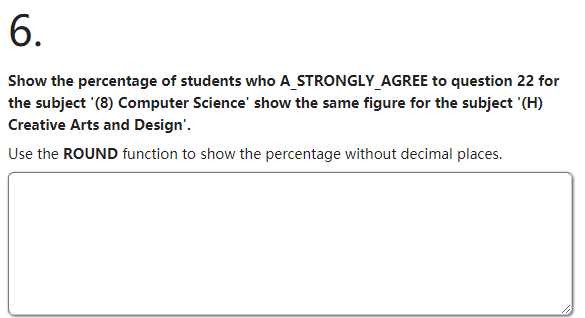
order by ga.mdate, go.matchid, ga.team1, ga.team2

* + - 1. 【4】<https://sqlzoo.net/wiki/NSS_Tutorial>第六题
         1. 【题目】

查询 来自'(8) Computer Science' 和'(H) Creative Arts and Design' 专业的学生，在回答问题22（Q22）时，表示强烈同意（A\_STRONGLY\_AGREE）的各专业人数占比

精确到个位数，不需要百分号

自行阅读题目涉及的表格哦~



* + - * 1. 【参考答案】

select

subject

, round(sum(A\_STRONGLY\_AGREE \* response) / sum(response),0)

from nss

WHERE (question='Q22' and subject='(8) Computer Science' )

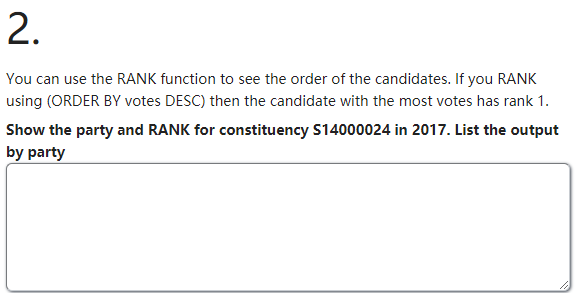
or (question='Q22' and subject = '(H) Creative Arts and Design')

group by subject

* + 1. 主知识点九：窗口函数
       1. 【1】<https://sqlzoo.net/wiki/Window_functions>第二题
          1. 【题目】

查询2017年选区为 'S14000024' 的所有候选人所在团体（party）和其得到选票数（votes）、还有候选人得票数在选区内对应的的排名

结果按团队party排序



* + - * 1. 【参考答案】

select

party

,votes

,rank() over (order by votes desc) as posn

from ge

where constituency = 'S14000024'

and yr = 2017

order by party

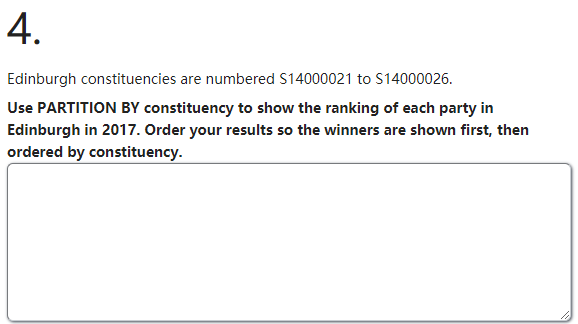
* + - 1. 【2】<https://sqlzoo.net/wiki/Window_functions>第四题
         1. 【题目】

查询2017年爱丁堡各个选区内各团队的排名情况

按照排名、选区排序

排名情况包括选区constituency，团队party，选票数votes，排名posn

已知爱丁堡选区编号为S14000021至S14000026



* + - * 1. 【参考答案】

select

constituency

,party

,votes

,rank()over(partition by constituency order by votes desc) as posn

from ge

where constituency between 'S14000021' and 'S14000026'

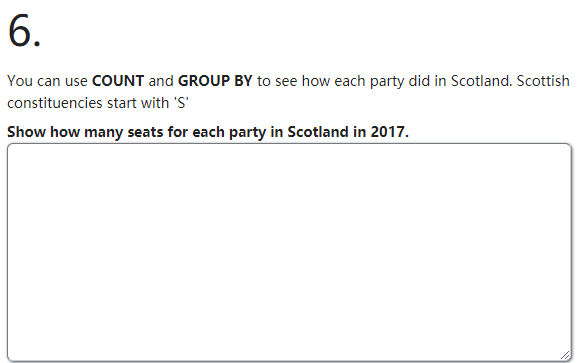
and yr = 2017

order by posn,constituency

* + - 1. 【3】<https://sqlzoo.net/wiki/Window_functions>第六题
         1. 【题目】

查询2017年在苏格兰的选区内（选区编号以S开头的）每个团体获得的席位数

选区候选人得票数为第1名，即获得该选区席位



* + - * 1. 【参考答案】

select

party

,count(\*)

from

(

select

constituency

,party

,votes

,rank()over(partition by constituency order by votes desc) as posn

from ge

where constituency like 'S%'

and yr = 2017

) rk

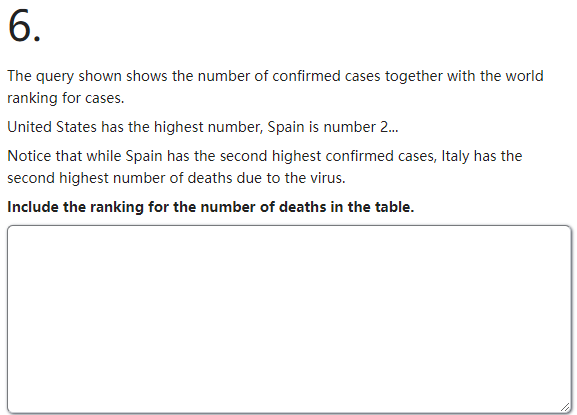
where rk.posn=1

group by rk.party

* + - 1. 【4】<https://sqlzoo.net/wiki/Window_LAG>第六题
         1. 【题目】

查询更新时间为2020年4月20日的国家名，确诊人数，确诊人数排名，死亡人数，死亡人数排名

按照确证人数降序排名



* + - * 1. 【参考答案】

select

name

,confirmed

,rank()over(order by confirmed desc) rc

,deaths

,rank()over(order by deaths desc) deathrc

from covid

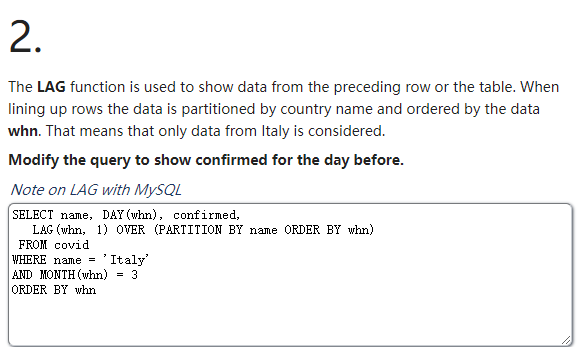
where whn = '2020-04-20'

order by confirmed desc

* + - 1. 【5】<https://sqlzoo.net/wiki/Window_LAG>第二题
         1. 【题目】

修改给出的代码，查询三月份意大利每天的确诊人数和前一天的确诊人数

并按更新时间排序



* + - * 1. 【窗口函数LAG简介】

**LAG(列名,n,m)**: 当前记录前面第n行记录的<列名>的值，没有则默认值为m；如果不带参数n,m，则查找当前记录前面第一行的记录<列名>的值，没有则默认值为null

* + - * 1. 【参考答案】

select

name

, day(whn)

, confirmed

,lag(confirmed,1) over (partition by name order by whn) lag

from covid

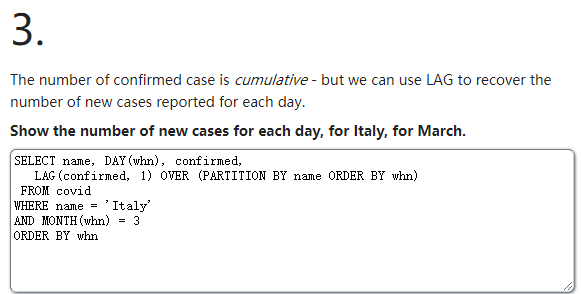
where name = 'italy'

and month(whn) = 3

order by whn

* + - 1. 【6】<https://sqlzoo.net/wiki/Window_LAG>第三题
         1. 【题目】

修改给出的代码，使用LAG函数查询三月份意大利每日新增确诊数



* + - * 1. 【参考答案】

select

name

, day(whn)

, (confirmed -lag(confirmed, 1) over (partition by name order by whn)) new

from covid

where name = 'italy'

and month(whn) = 3

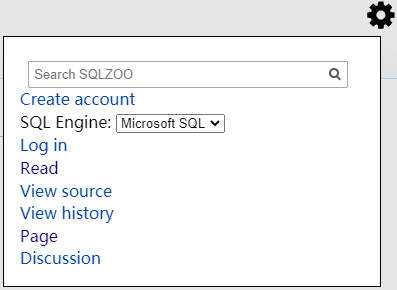
order by whn

* + - 1. 【7】<https://sqlzoo.net/wiki/Window_LAG>第四题 **这题务必认真复习！！！**
         1. 【切换数据库操作】

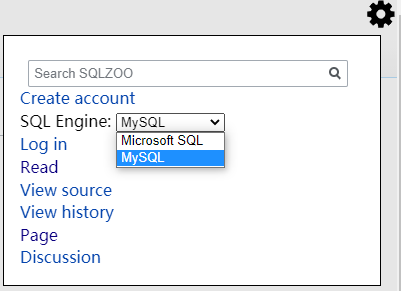
找到页面最右上角的齿轮图标



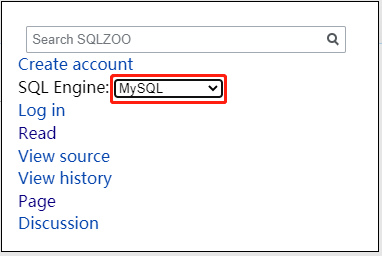
点击图标



切换数据库引擎为MySQL



mysql支持函数date\_format()和weekday()，但Microsoft SQl不支持

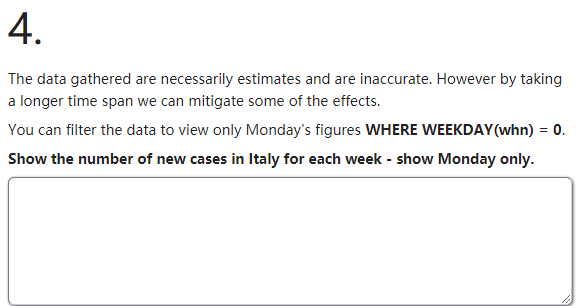


* + - * 1. 【题目】

查询意大利每周新增确诊数（显示每周一的数值 weekday(whn) = 0）

最后显示国家名，标准日期（2020-01-27），每周新增人数

按照更新时间排序



* + - * 1. 【参考答案】

select

name

, date\_format(whn,'%Y-%m-%d') date

, (confirmed - lag(confirmed,1)over(partition by name order by whn)) New

from covid

where name = 'Italy'

and weekday(whn) = 0

order by whn

1. 学完这个再学习运行原理，效果更佳哦~