《数据库概论》实验一

用SQL进行数据操作 实验报告

姓名：姜宁 学号：191840116 Email：191840116@smail.nju.edu.cn

实验环境

Window10+MySQL8.0.26.0

实验过程

Q1

```
select count(*) as speciesCount
from species
where description like '%this%' ;
```

查询结果如下：

	speciesCount
▶	90

使用like语句进行字符串匹配，%表示匹配任意长度的字符串

Q2

```
select p.username as username,sum(pm.power) as totalPhonemonPower
from player p,phonemon pm
where p.id=pm.player and (p.username='Cook' or p.username='Hughes')
group by p.id;
```

查询结果如下：

	username	totalPhonemonPower
▶	Cook	1220
	Hughes	1170

连接player表和phonemon表，按条件查询后根据player的id分组即可

Q3

```
select t.title as title,count(p.id) as numberOfPlayers
from team t,player p
where p.team=t.id
group by t.id
order by count(p.id) desc;
```

查询结果如下：

title	numberOfPlayers
Mystic	8
Valor	6
Instinct	5

连接team表和player表后找到player的team为特定team的id后按照队伍分组，并根据队伍人数排序

Q4

```
select sp.id as idSpecies,sp.title as title
from species sp,type ty
where (sp.type1=ty.id and ty.title='grass') or
(sp.type2=ty.id and ty.title='grass');
```

查询结果如下：

idSpecies	title
1	Bulbasaur
2	Ivysaur
3	Venusaur
43	Oddish
44	Gloom
45	Vileplume
69	Bellsprout
70	Weepinbell
71	Victreebel
102	Exeggcute
103	Exeggutor
114	Tangela

由于物种可能有两个属性，所以要考虑任一个属性为grass的情况，并用or连接

Q5

```
select p.id as idPlayer,p.username as username
from player p
where p.id not in (
    select pc.player
    from purchase pc,item i
    where pc.item=i.id and i.type='F'
    );
```

查询结果如下：

	idPlayer	username
▶	4	Reid
	7	Hughes
	8	Bruce
	10	Lyons
	11	Emily
	12	Darthy
	15	Huma

首先查询所有购买过food的玩家并返回玩家的id，再通过not in来筛选出没有购买过食物的玩家

Q6

```
select p.level as level,
    sum(pc.quantity*i.price) as totalAmountSpentByAllPlayersAtLevel
from player p,purchase pc,item i
where pc.item=i.id and pc.player=p.id
group by p.level
order by sum(pc.quantity*i.price) desc;
```

查询结果如下：

	level	totalAmountSpentByAllPlayersAtLevel
▶	2	130.68
	12	95.45
	6	62.37
	5	52.98
	3	51.75
	1	39.58
	4	33.74
	8	29.48
	11	26.97
	7	24.26
	10	17.22
	9	9.99

将player表，purchase表和item表做连接后根据player的level进行分组，花费的总金额为purchase中的quantity乘上其item的单价price，最后按照金额降序排序即可

Q7

```
select newtable.item as item,newtable.title as title,
    newtable.numTimesPurchased as numTimesPurchased
from (
    select i.id as item,i.title as title,count(pc.id) as numTimesPurchased
    from item i,purchase pc
    where i.id=pc.item
    group by i.id
    ) newtable
where newtable.numTimesPurchased=ALL(
```

```
    select max(_newtable.numTimesPurchased)
    from (
        select i.id as item,i.title as title,count(pc.id) as numTimesPurchased
        from item i,purchase pc
        where i.id=pc.item
        group by i.id
        ) _newtable
    );
```

查询结果如下：

	item	title	numTimesPurchased
▶	1	Phoneball	10

首先我们需要生成一个包含物品的id，物品的title和物品被购买次数numTimesPurchased的新表，代码如下：

```
select i.id as item,i.title as title,count(pc.id) as numTimesPurchased
from item i,purchase pc
where i.id=pc.item
group by i.id
```

将其命名为newtable，之后我们只需要找出numTimesPurchased最大的元组即可，我是通过先找出最大值再依次判等来找出所有并列项，代码如下：

```
where newtable.numTimesPurchased=ALL(
    select max(newtable.numTimesPurchased) from newtable
    );
```

ps：由于再from中创建的新表无法在where中使用，所以繁琐的又创建了一个一模一样的表_newtable

Q8

```
select p.id as playerID,p.username as username,count(newtable.itemID) as
numberDistinctFoodItemsPurchased
from player p,(
    select distinct i.id as itemID
    from purchase pc,item i
    where pc.item=i.id and i.type='F'
)newtable
where not exists(
    select * from item i  -- 没有被选择过的食物
    where not exists(
        select * from purchase pc where p.id=pc.player and i.id=pc.item
        ) and i.type='F'
    );
```

查询结果如下：

	playerID	username	numberDistinctFoodItemsPurchased
▶	20	Zihan	6

首先查询所有食物的数量，则创建一个包含所有食物的新表，代码如下：

```
select distinct i.id as itemID
from purchase pc,item i
where pc.item=i.id and i.type='F'
```

通过distinct来去重，最后利用count函数即可统计所有食物的数量。

接着查询购买了所有食物的玩家，where条件中最内部的

```
select * from purchase pc where p.id=pc.player and i.id=pc.item
```

是查询所有特定玩家购买特定物品的信息，那么

```
select * from item i
where not exists(
    select * from purchase pc where p.id=pc.player and i.id=pc.item
    ) and i.type='F'
```

则是查询所有没有被特定玩家购买过的食物，那么最外层的where条件则是选取所有没有“未购买的食物”的学生，即购买了所有种类食物的学生

Q9

```
select finaltable.numberOfPhonemonPairs as numberOfPhonemonPairs,
    finaltable.distance as distanceX
from(
    select count(newtable.distance) as numberOfPhonemonPairs,
        newtable.distance as distance
    from (
        select round(sqrt(power(pm1.latitude-pm2.latitude,2)
            +power(pm1.longitude-pm2.longitude,2))*100,5) as distance
        from phonemon pm1,phonemon pm2
        where pm1.id>pm2.id
        ) newtable
    group by newtable.distance
) finaltable
where finaltable.distance=ALL(
    select min(_finaltable.distance)
    from (
        select count(newtable.distance) as numberOfPhonemonPairs,
            newtable.distance as distance
        from (
            select round(sqrt(power(pm1.latitude-pm2.latitude,2)
                +power(pm1.longitude-pm2.longitude,2))*100,5) as distance
            from phonemon pm1,phonemon pm2
            where pm1.id>pm2.id
            ) newtable
        group by newtable.distance
        ) _finaltable
);
```

查询结果如下：

	numberOfPhonemonPairs	distanceX
▶	54	0.19096

首先我们需要生成一个包含所有phonemon之间距离的表，代码如下：

```
select round(sqrt(power(pm1.latitude-pm2.latitude,2)
    +power(pm1.longitude-pm2.longitude,2))*100,5) as distance
from phonemon pm1,phonemon pm2
where pm1.id>pm2.id
```

这里首先将phonemon表自连接，通过比较id大小来去除重复计算。将新表命名为newtable后我们需要利用newtable表创建一个新表用来统计相同distance的数量，代码如下：

```
select count(newtable.distance) as numberOfPhonemonPairs,
    newtable.distance as distance
from newtable
group by newtable.distance
```

将其命名为finaltable后只需要选出finaltable中最小的distanceX即可，处理方法同Q7（where里的_finaltable也同Q7）

ps：round(x,i) 的作用是将x保留i位小数，power(x,i)的结果是x的i次方

Q10

```
select p.username as username,tp.title as typeTitle
from player p,type tp
where not exists(
    select * from species sp where not exists(      -- 所有没有被选择的物种
        select * from phonemon pm where pm.player=p.id and pm.species=sp.id
    ) and (sp.type1=tp.id or sp.type2=tp.id)
);
```

查询结果如下：

	username	typeTitle
▶	Lyons	Fairy
	Lyons	Bug

最内部的

```
select * from phonemon pm where pm.player=p.id and pm.species=sp.id
```

是查询所有属于特定玩家和特定species的phonemon信息，那么

```
select * from species sp where not exists(
        select * from phonemon pm where pm.player=p.id and pm.species=sp.id
    ) and (sp.type1=tp.id or sp.type2=tp.id)
```

则是查询某特定类型的所有没有被选择过的物种（species），在这个基础上最外层的where not exist则是找出某个“不存在所有没有被选择过的species”的type，即每一物种species都被捕捉的某特定类型type，这样就成功完成了查询要求

实验中遇到的困难及解决办法

1.在Q7中首次遇到from中创建新表的情况，之后在where中直接使用后报错该表not exist，解决办法是在where中再重新创建一遍；

2.Q8和Q10的类似“查找选修了所有课程的学生”的查询，难以直接处理“所有”，解决办法是利用双重否定句和exist语句间接求出

参考文献及致谢

独立完成