

# 《数据库概论》实验一：用 SQL 进行数据操作 实验报告

姓名 田逸凡 学号 201850076

联系方式 1577615968@qq.com

## 实验环境

Windows 10

MySQL workbench 8.0

## 实验过程

1.根据实验手册给出的信息分析每个结构之间的对应关系，分析题目并写出对应 SQL 语句进行调试。

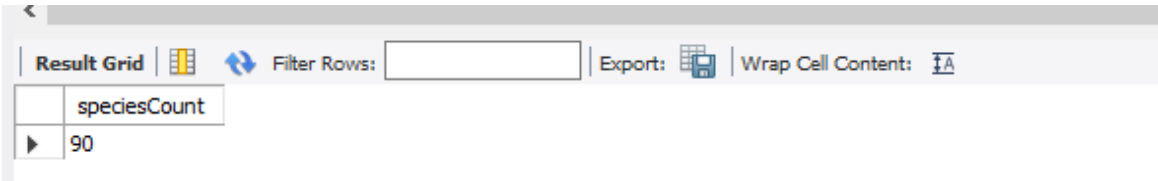
2.逐步完成实验并测试：

Q1：用 “%this%”表示表述中有 this 的单词，用 count 计数即可。

代码：

```
6 -- BEGIN Q1
7 • SELECT Count(*) as speciesCount FROM species WHERE species.description like "%this%";
8 -- END Q1
```

实验结果：



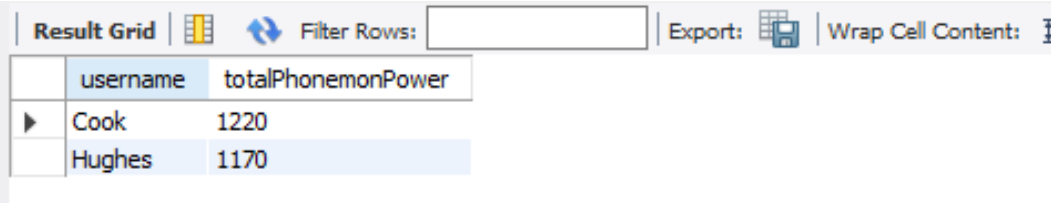
speciesCount
90

Q2：对不同人名对应的 Phonemon 能量分别求和，用 GROUP 分组

代码：

```
-- BEGIN Q2
SELECT username,sum(Power)as totalPhonemonPower FROM Player,Phonemon
WHERE (Player.username="Cook" or Player.username="Hughes") AND (Player.id=Phonemon.player)
GROUP BY Player.username;
-- END Q2
```

实验结果：



username	totalPhonemonPower
Cook	1220
Hughes	1170

Q3: 根据不同的队伍 id 进行分组计数，再降序排列即可  
代码:

```
-- BEGIN Q3
SELECT title,count(*) as numberOfPlayers From Player,Team
WHERE Player.team=Team.id Group by Team Order by count(*) DESC;
-- END Q3
```

实验结果:

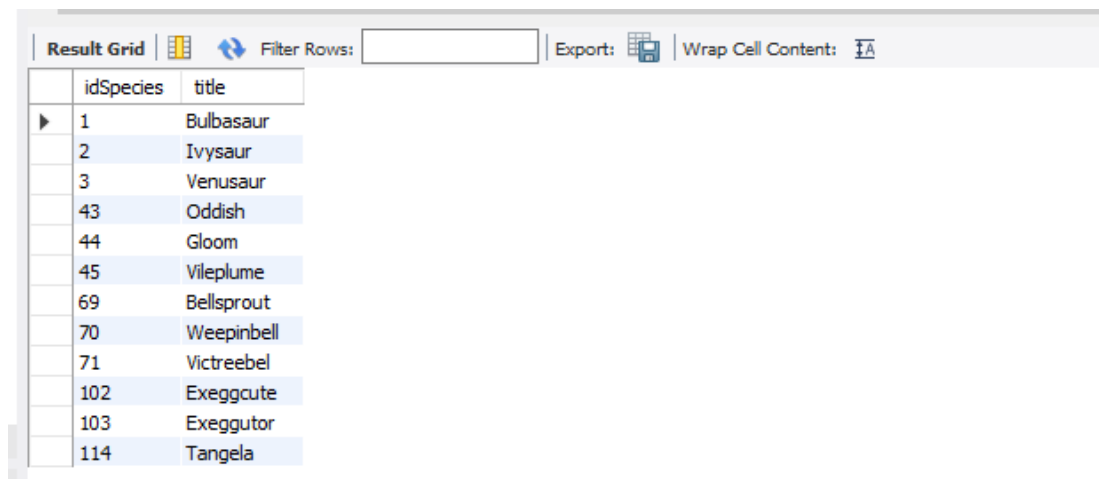


	title	numberOfPlayers
▶	Mystic	8
	Valor	6
	Instinct	5

Q4: 寻找类型为 Grass 的物种即可  
代码:

```
-- BEGIN Q4
• SELECT Species.id as idSpecies,Species.title FROM Species,Type
WHERE (Species.type1=Type.id or Species.type2=Type.id) AND Type.title="Grass";
-- END Q4
```

实验结果:



	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

Q5: 用 not in 排除所有买过 food 的玩家，剩下的即满足要求。  
代码:

```
-- BEGIN Q5
• SELECT Player.id as idPlayer,username From Player
WHERE Player.id NOT IN
(SELECT Player.id FROM Player,Purchase,Item
WHERE Player.id=Purchase.player AND Purchase.item=Item.id AND Item.type="F");
-- END Q5
```

实验结果:

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

Q6: 根据 level 分组, 并用单价乘购买数量统计总金额并累加。

代码:

```
-- BEGIN Q6
SELECT Player.level,sum(Item.price*Purchase.quantity) as totalAmountSpentByAllPlayersAtLevel
FROM Player,Purchase,Item
WHERE Player.id=Purchase.player and Item.id=Purchase.item
GROUP BY Player.level ORDER BY sum(Item.price*Purchase.quantity) DESC;
-- END Q6
```

实验结果:



	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

Q7: 选出物品 ID 和购买次数, 用 having 和 all 的筛选条件找到最大值。

代码:

```
-- BEGIN Q7
SELECT Item.id as item,Item.title,count(Item.id) as numTimesPurchased
FROM Item,Purchase WHERE Purchase.item=Item.id GROUP BY Item.id
HAVING count(Item.id)>=all(SELECT count(*) FROM Purchase Group by Item);
-- END Q7
```



实验结果:

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	item	title	numTimesPurchased
▶	1	Phoneball	10

Q8: 套用除法, 看成所有购买除以 Food 的形式, 得到购买所有 food 的 userid  
代码:

```
-- BEGIN Q8
SELECT Player.id,Player.username,count(distinct f1.id) as numberDistinctFoodItemsPurchased
FROM Player,Food f1 WHERE NOT EXISTS(SELECT * FROM Food f2 WHERE NOT EXISTS
(SELECT * FROM Purchase WHERE Purchase.item=f2.id AND Purchase.player=Player.id));
-- END Q8
```

实验结果:



Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	id	username	numberDistinctFoodItemsPurchased
▶	20	Zihan	6

Q9: 将所有点对之间的距离保存在 table 中, 并选择出其中最小的 disX, 再在 table 中对距离值等于这个数的点对进行计数, 最后除以 2 排除相同点对。

代码:

```
-- BEGIN Q9
• SELECT (count(*)/2) as numberOfPhonemonParis,table1.dis as distanceX
FROM(SELECT round((sqrt(pow((p1.latitude-p2.latitude),2)+pow((p1.longitude-p2.longitude),2))*100),4) as dis
FROM Phonemon p1,Phonemon p2
WHERE p1.id<>p2.id) as table1
WHERE table1.dis=(SELECT min(table1.dis)
FROM (SELECT round((sqrt(pow((p1.latitude-p2.latitude),2)+pow((p1.longitude-p2.longitude),2))*100),4) as dis
from Phonemon p1,Phonemon p2
where p1.id<>p2.id) as table1)
GROUP BY table1.dis;
-- END Q9
```

实验结果:

Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	numberOfPhonemonParis	distanceX
▶	98.0000	0.191

Q10: 和 Q8 类似，套用除法，看成特定类型不存在一个物种没有被玩家抓住即可，输出 id 和 type

代码：

```
73      -- BEGIN Q10
74 •    SELECT Player.username,Type.title as typeTitle
75      FROM Player,Type
76      WHERE not exists(SELECT* FROM Species
77      WHERE Type.id in(Species.type1,Species.type2) AND not exists(SELECT*FROM phonemon
78      WHERE Player.id=Phonemon.player AND Phonemon.species=Species.id));
79      -- END Q10
```

实验结果：

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	username	typeTitle			
▶	Lyons	Fairy			
	Lyons	Bug			

## 实验中遇到的困难及解决办法

1. 在代码书写的格式方面容易弄错导致运行出错，通过看报错提示以及几道题的熟悉之后慢慢掌握方法。
2. Q8 和 Q10 的除法运用不够熟练，一开始被卡住了，后来通过反复阅读 ppt 除法课件和不断调试逐渐学会写除法语句。
- 3.Q9 的距离计算一开始没有头绪，后来想到可以通过 table 来保存数据后顺利做完。

## 参考文献及致谢

教学立方 SQL 部分的各个 PPT