《数据库概论》实验一：用SQL进行数据操作 实验报告

姓名 XXX 学号 XXX 联系方式 XXX

实验环境

操作系统：Windows 10

数据库系统：MySQL 8.0.23

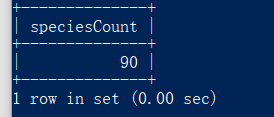
实验过程

首先用MySQL Workbench 导入数据库脚本，建立数据模式，然后用MySQL完成以下10个查询。查询代码和结果如下(使用PowerShell界面，经本地测试，完整sql脚本在MySQL Workbench中可以运行并得到10个相同的结果)

1、有多少物种species的描述description中含有单词“this”？查询以如下形式返回：(speciesCount)。

**SELECT** COUNT(\*) speciesCount **FROM** species

**WHERE** description like '%this%';



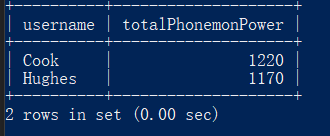
2、玩家player‘Cook’将与玩家player‘Hughes’作战。对于两个玩家，显示他们的用户名username和他们各自拥有的Phonemon的总能量。查询以如下形式返回：(username, totalPhonemonPower)。

**SELECT** a.username username,sum(b.power) totalPhonemonPower

**FROM** player a,phonemon b

**WHERE** a.id = b.player AND a.username IN ('Hughes','Cook')

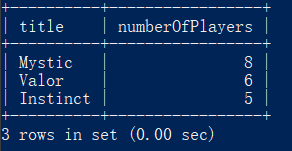
**GROUP** **BY** a.username;



3、每一个队伍team有多少名成员player？按照玩家数量降序列出队伍名称title和玩家数量。查询以如下形式返回：(title, numberOfPlayers)。

**SELECT** t.title,count(\*) numberOfPlayers **FROM** team t,player p

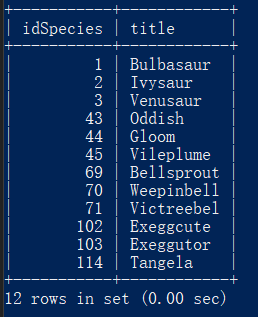
**WHERE** t.id = p.team **GROUP** **BY** t.title **ORDER** **BY** numberOfPlayers **desc**;



4、哪些物种species具有类型type‘grass’？查询以如下形式返回：(idSpecies, title)。

**SELECT** s.id idSpecies,s.title **FROM** species s, type t

**WHERE** t.title = 'grass' AND (t.id = s.type1 or t.id = s.type2);

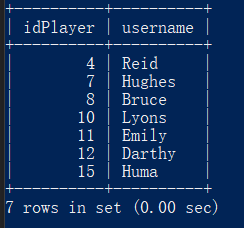


5、列出从未购买过食物food的玩家player。查询以如下形式返回：(idPlayer, username)。

**SELECT** p.id idPlayer,username **FROM** player p

**WHERE** p.id NOT IN (

**SELECT** pu.player **FROM** purchase pu,food f **WHERE** f.id = pu.item);



6、游戏中的每个玩家player具有特定的等级level。以金额降序列出每一特定等级以及该等级的所有玩家在购买上花费的总金额。查询以如下形式返回：(level, totalAmountSpentByAllPlayersAtLevel)。

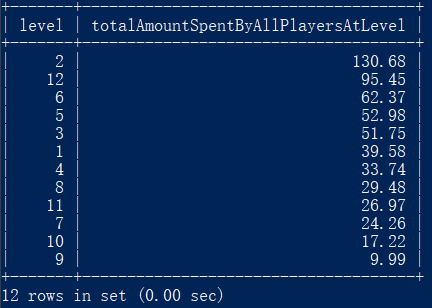
**SELECT** p.**level**,sum(it.price \* pu.quantity) totalAmountSpentByAllPlayersAtLevel

**FROM** player p,item it,purchase pu

**WHERE** pu.item = it.id AND pu.player = p.id

**GROUP** **BY** p.**level**

**ORDER** **BY** totalAmountSpentByAllPlayersAtLevel **desc**;



7、什么物品item被购买次数最多？如有并列，列出所有购买次数最多的物品。查询以如下形式返回：(item, title, numTimesPurchased)。

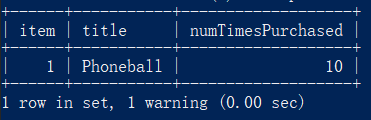
**SELECT** it.id item,it.title title,count(\*) numTimesPurchased

**FROM** item it,purchase p **WHERE** p.item = it.id

**GROUP** **BY** item

**havINg** numTimesPurchased >=all(

**SELECT** count(\*) **FROM** purchase **GROUP** **BY** item);



8、找到可获取的食物的数量，和购买所有种类食物至少各一次的玩家。查询以如下形式返回：(playerID, username, numberDistinctFoodItemsPurchased)。

**SELECT** **distinct** p.id playerID,p.username username,count(**distinct** pu.item) numberDistinctFoodItemsPurchased

**FROM** player p,purchase pu,item it

**WHERE** pu.player = p.id AND it.id = pu.item AND it.type = 'F'

    AND p.id IN

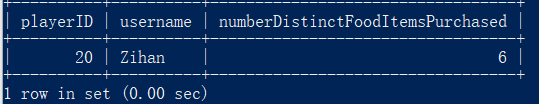
    (**SELECT** id **FROM** player **WHERE**

    NOT exists(**SELECT** \* **FROM** food f

**WHERE** NOT exists(**SELECT** \* **FROM** purchase pur **WHERE** pur.item = f.id AND player.id = pur.player))

    )

**GROUP** **BY** p.id;



9、将距离最近的两个Phonemon之间的欧氏距离称为X。计算相互之间距离为X的Phonemon对的数量。查询以如下形式返回：(numberOfPhonemonPairs,distanceX)。提示1：使用ROUND()函数；提示2：请确保每对仅被计数一次，避免重复计数。

**SELECT** count(\*) numberOfPhonemonPairs,round(100 \* **mIN**(sqrt((p1.latitude - p2.latitude)\*(p1.latitude - p2.latitude)

+ (p1.longitude - p2.longitude) \* (p1.longitude - p2.longitude))),2) **as** distanceX

**FROM** phonemon p1,phonemon p2 **WHERE** p1.id < p2.id AND

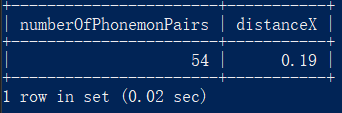
 (p1.latitude - p2.latitude)\*(p1.latitude - p2.latitude) +

 (p1.longitude - p2.longitude) \* (p1.longitude - p2.longitude) <= all(

**SELECT** (p3.latitude - p4.latitude)\*(p3.latitude - p4.latitude) +

     (p3.longitude - p4.longitude) \* (p3.longitude - p4.longitude)

**FROM** phonemon p3,phonemon p4 **WHERE** p3.id < p4.id);



10、一些玩家player热衷于某种特定类型type的Phonemon。列出捕捉了任一特定类型type中每一物种species至少各一个Phonemon的玩家的名称以及该类型的名称。查询以如下形式返回：(username, typeTitle)。例如，假设类型‘bug’共有三种物种species的Phonemon，而玩家‘Greg’拥有三种Phonemon各一个，那么结果中包含(Greg, bug)。

**SELECT** p.username,t.title **FROM** player p,type t,

(

**SELECT** player pid,type tid **FROM**

    (

**SELECT** count(**distINct** species) scount,player,type **FROM** (

**SELECT** a.type,p.species,p.player **FROM** (

**SELECT** s1.id id, s1.type1 type **FROM** species s1

**union** all

**SELECT** s2.id id, s2.type2 type **FROM** species s2

**WHERE** s2.type2 **is** NOT null)

**as** a,

            phonemon p **WHERE** p.species = a.id AND p.player **is** NOT null

**ORDER** **BY** a.type,p.species,p.player

        ) x

**GROUP** **BY** player,type

    )y

**WHERE** (scount,type) IN

(**SELECT** count(**distINct** sid) scount,type **FROM**(

**SELECT** s1.id sid ,s1.type1 type **FROM** species s1

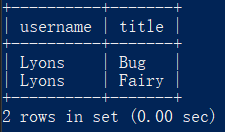
**union** all

**SELECT** s2.id sid,s2.type2 type **FROM** species s2 **WHERE** s2.type2 **is** NOT null

) x **GROUP** **BY** type)

)c

**WHERE** p.id = c.pid AND t.id = c.tid;



实验中遇到的困难及解决办法

[详细说明你认为本次实验中比较困难的地方，也可以对实验设计提出建议]

比较困难的地方：复杂多表查询，多个子查询

解决方法：用terminal一步一步调，比如第10题，species有两个type，先把(type,species)整合到一个子查询,再按type分组整合为(type,count(type))利用一种类型的数量类判断是否包含一种type的全部species。再把这张表与player连结，筛选满足条件的(player,type),最后把这些筛选出的id和player，type连结，选出player name 和type title。

参考文献及致谢

《数据库系统使用教程》

《数据库原理、编程与性能》