数据库概论实验一:用SQL进行数据操作 实验报告

田永铭 221900180

tianym2022@smail.nju.edu.cn

一、实验环境

操作系统版本: Windows 11

软件版本: MySQL Workbench 8.0, MySQL Router 8.0, MySQL Server 8.0, MySQL Shell 8.0

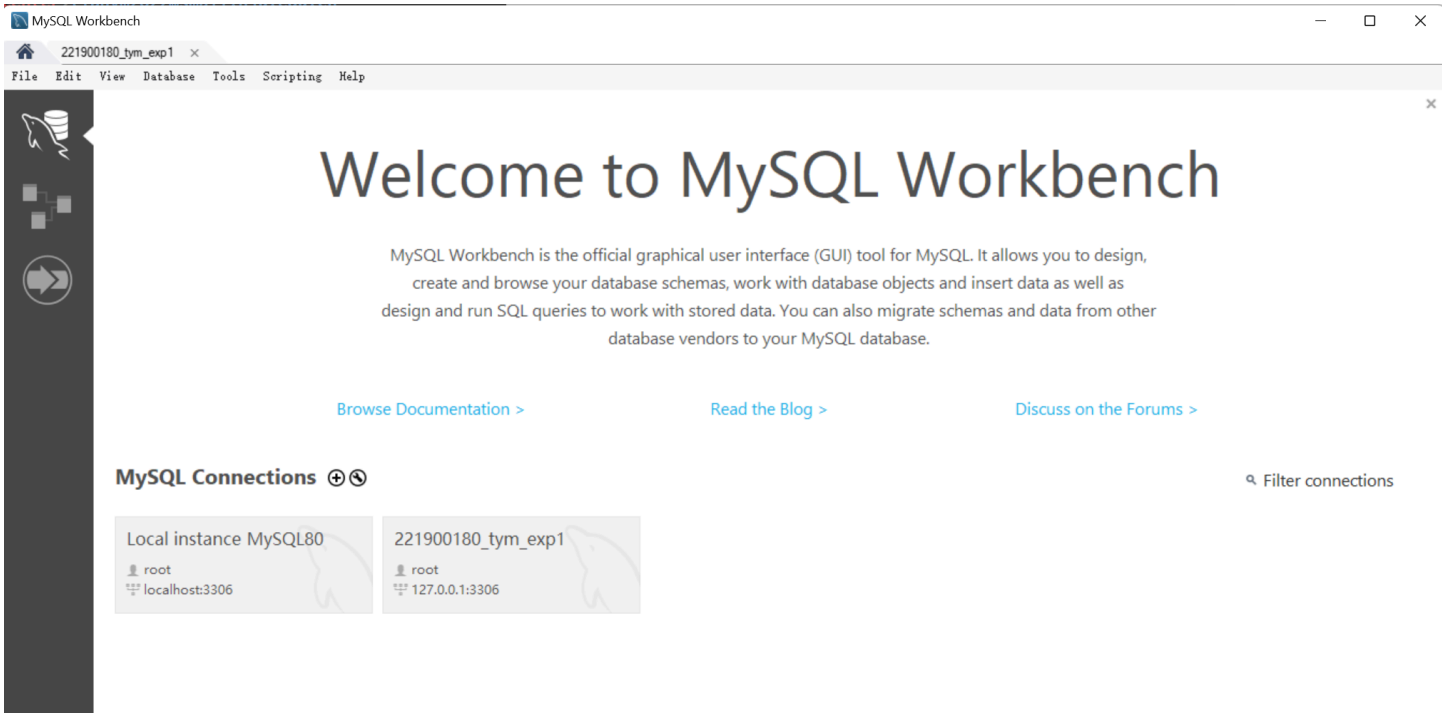
二、实验过程

1.下载和安装MySQL

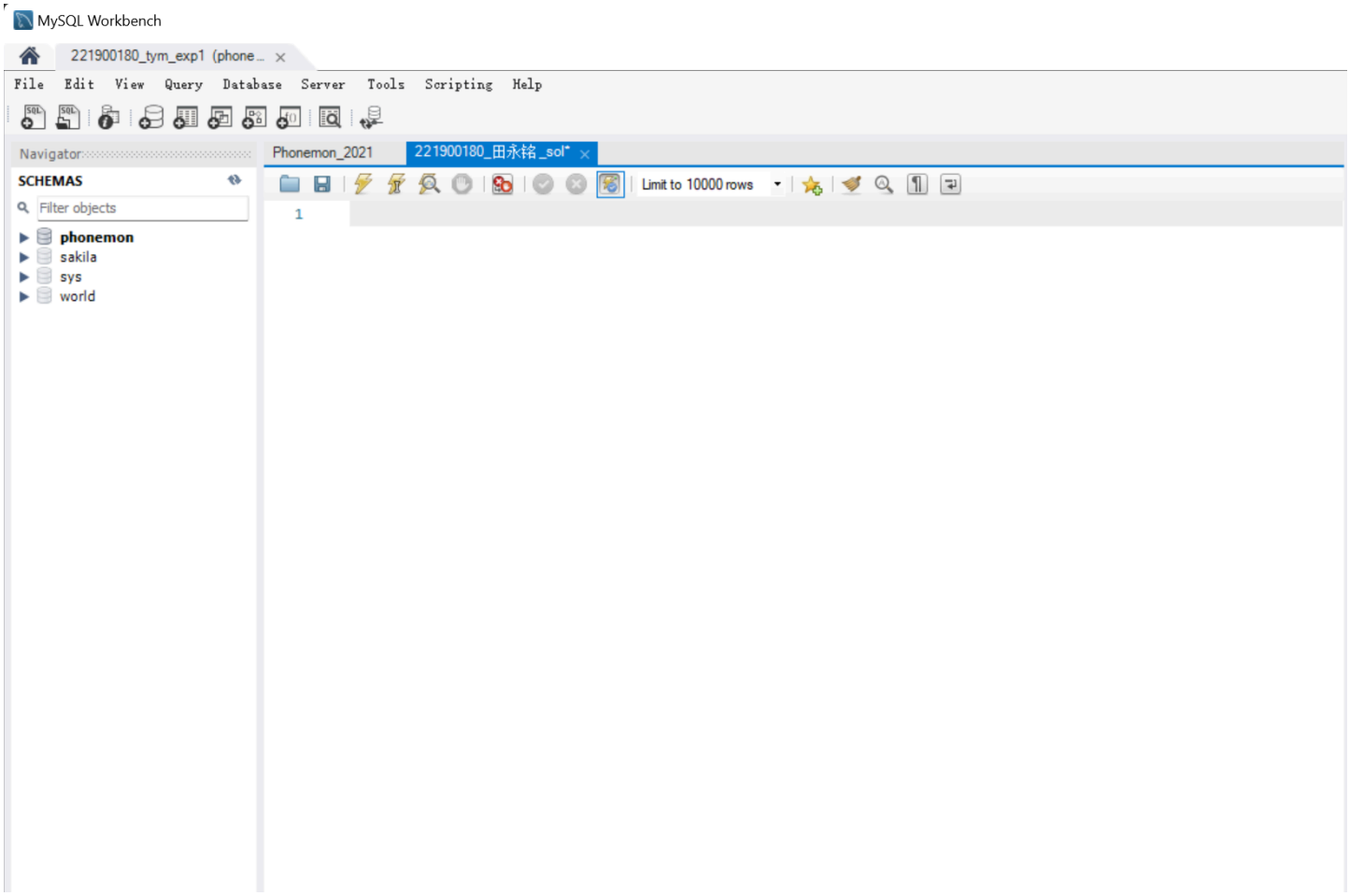
前往 MySQL官网 下载，参考文献 MySQL安装详解，可以较为简单地下载和安装好MySQL。

2.连接数据库，导入测试数据包，打开写代码的界面

参考 MySQL 数据库保姆级图解教程，创建一个数据库 **221900180_tym_exp1** ，建立好链接，双击打开链接到MySQL服务器，打开窗口。



将老师提供的测试数据包 **Phenemon_2021** 打开，再新建一个SQL查询界面，就可以在这里写代码并且运行查看结果了。

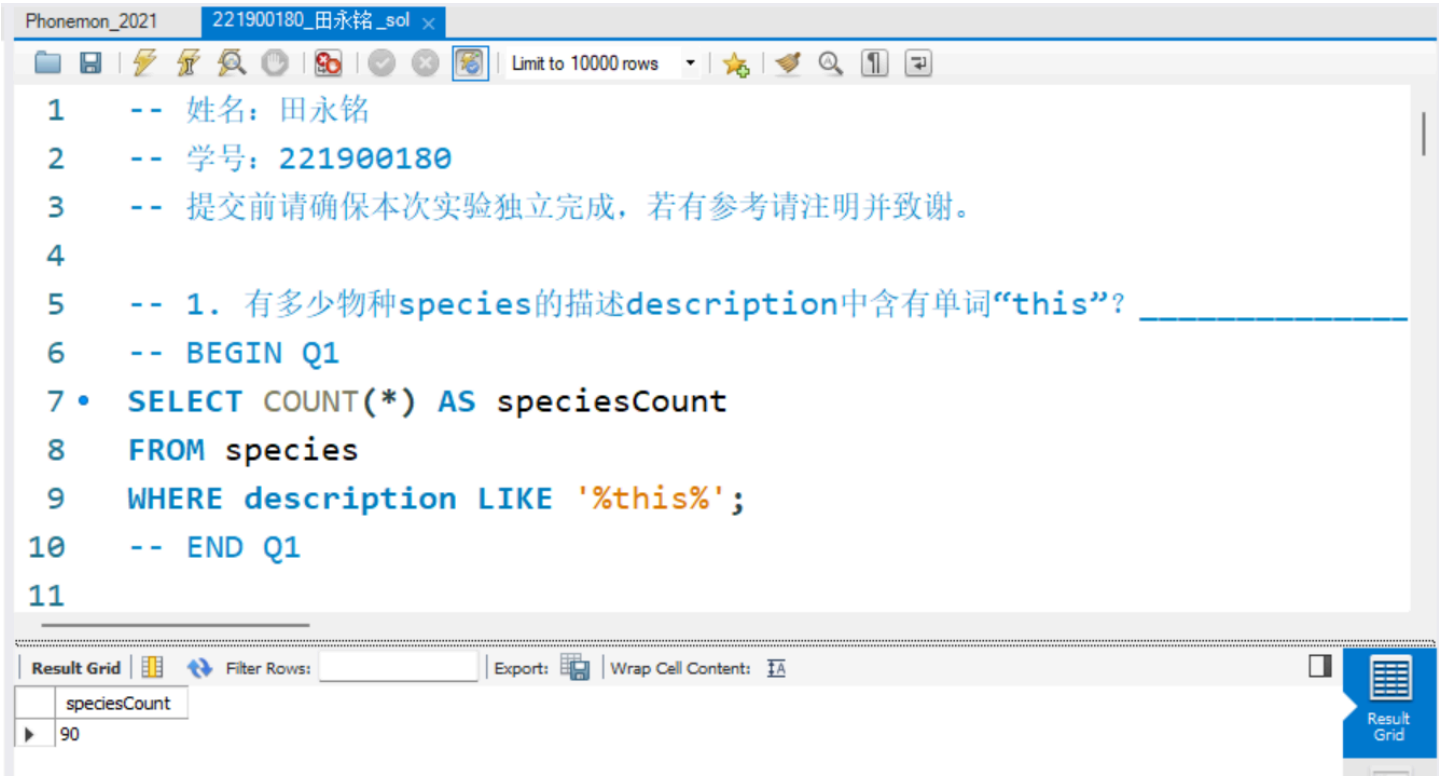


3.编写代码，运行查看结果

注意：由于有些代码一行太长了，所以我增加了换行，若要实际运行代码，请注意。同时，附件也有代码文件，能直接运行。

（1）有多少物种species的描述description中含有单词“this”?

```
SELECT COUNT(*) AS speciesCount
FROM species
WHERE description LIKE '%this%';
```



（2）玩家player‘Cook’将与玩家player‘Hughes’作战。显示他们的用户名username和他们各自拥有的Phonemon的总能量。

```
SELECT p.username,(SELECT SUM(power) FROM phonemon WHERE phonemon.player = p.id)
  AS totalPhonemonPower
FROM player p
WHERE p.username IN ('Cook', 'Hughes');
```

（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。

```
SELECT t.title, (SELECT COUNT(*) FROM player WHERE player.team = t.id) AS numberOfPlayers
FROM team t
ORDER BY numberOfPlayers DESC;
```

(4) 哪些物种species具有类型type‘grass’?

```
SELECT s.id AS 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。

```
SELECT p.id AS idPlayer, p.username
FROM player p
WHERE p.id NOT IN (
        SELECT p2.id
    FROM player p2, purchase pur, item i
    WHERE pur.player = p2.id and pur.item = i.id and i.type = 'F'
);
```

```
33    -- 5. 列出从未购买过食物food的玩家player。
34    -- BEGIN Q5
35 •  SELECT p.id AS idPlayer, p.username
36    FROM player p
37    WHERE p.id NOT IN (
38        SELECT p2.id
39        FROM player p2, purchase pur, item i
40        WHERE pur.player = p2.id and pur.item = i.id and i.type = 'F'
41    );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Result Grid | Form Editor | Field

（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。

```
SELECT p.level, SUM(i.price * pur.quantity) AS totalAmountSpentByAllPlayersAtLevel
FROM player p, purchase pur, item i
WHERE p.id = pur.player and pur.item = i.id
GROUP BY p.level
ORDER BY totalAmountSpentByAllPlayersAtLevel DESC;
```

```
43
44    -- 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
45    -- BEGIN Q6
46 •  SELECT p.level, SUM(i.price * pur.quantity) AS totalAmountSpentByAllPlayersAtLevel
47    FROM player p, purchase pur, item i
48    WHERE p.id = pur.player and pur.item = i.id
49    GROUP BY p.level
50    ORDER BY totalAmountSpentByAllPlayersAtLevel DESC;
51    -- END Q6
52
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Result Grid / Form Editor / Field Types / Query Stats

（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。

```
SELECT i.id AS item, i.title, COUNT(pur.id) AS numTimesPurchased
FROM purchase pur, item i
WHERE pur.item = i.id
GROUP BY i.id, i.title
HAVING  COUNT(pur.id) = (
        SELECT MAX(times_count)
        FROM (
            SELECT COUNT(pur2.id) AS times_count
            FROM purchase pur2
            GROUP BY pur2.item
        ) AS max_times
    );
```

```
53    -- 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。
54    -- BEGIN Q7
55    SELECT i.id AS item, i.title, COUNT(pur.id) AS numTimesPurchased
56    FROM purchase pur, item i
57    WHERE pur.item = i.id
58    GROUP BY i.id, i.title
59    HAVING  COUNT(pur.id) = (
60            SELECT MAX(times_count)
61            FROM (
62                SELECT COUNT(pur2.id) AS times_count
63                FROM purchase pur2
64                GROUP BY pur2.item
65            ) AS max_times
66        );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

item	title	numTimesPurchased
1	Phoneball	10

(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。

```
SELECT p.id AS playerID, p.username, COUNT(DISTINCT i.id) AS
   numberDistinctFoodItemsPurchased
FROM player p, purchase pur, item i
WHERE p.id = pur.player and pur.item = i.id and i.type = 'F'
GROUP BY p.id, p.username
HAVING COUNT(DISTINCT i.id) = (
    SELECT COUNT(DISTINCT i2.id)
    FROM item i2
    WHERE type = 'F'
);
```

```
68
69    -- 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。
70    -- BEGIN Q8
71    SELECT p.id AS playerID, p.username, COUNT(DISTINCT i.id) AS numberDistinctFoodItemsPurchased
72    FROM player p, purchase pur, item i
73    WHERE p.id = pur.player and pur.item = i.id and i.type = 'F'
74    GROUP BY p.id, p.username
75    HAVING COUNT(DISTINCT i.id) = (
76        SELECT COUNT(DISTINCT i2.id)
77        FROM item i2
78        WHERE type = 'F'
79    );
80    -- END Q8
81
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

playerID	username	numberDistinctFoodItemsPurchased
20	Zihan	6

（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。

```
SELECT COUNT(*)  AS numberOfPhonemonPairs,(
          SELECT MIN(ROUND(SQRT(POW(p1.latitude - p2.latitude, 2) +
          POW(p1.longitude - p2.longitude, 2)) * 100,2))
          FROM phonemon p1, phonemon p2
          WHERE p1.id < p2.id
        ) AS distanceX
FROM phonemon p3, phonemon p4
WHERE p3.id < p4.id and (ROUND(SQRT(POW(p3.latitude - p4.latitude, 2)
+ POW(p3.longitude - p4.longitude, 2)) * 100,2)) = (
         SELECT MIN(ROUND(SQRT(POW(p1.latitude - p2.latitude, 2) +
      POW(p1.longitude - p2.longitude, 2)) * 100,2))
         FROM phonemon p1, phonemon p2
         WHERE p1.id < p2.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。

```
SELECT p.username, t.title as typeTitle
FROM player p, type t, phonemon pho, species s
WHERE (t.id = s.type1 or t.id = s.type2) and pho.player = p.id and pho.species = s.id
GROUP BY username, typeTitle
HAVING COUNT(distinct s.id) = (
         SELECT COUNT(*)
    FROM species s2,type t2
         WHERE (s2.type1 = t2.id or s2.type2 = t2.id) and t2.title = t.title
    );
```

```
96
97      -- 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
98      -- BEGIN Q10
99      SELECT p.username, t.title as typeTitle
100     FROM player p, type t, phonemon pho, species s
101     WHERE (t.id = s.type1 or t.id = s.type2) and pho.player = p.id and pho.species = s.id
102     GROUP BY username, typeTitle
103     HAVING COUNT(distinct s.id) = (
104         SELECT COUNT(*)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

username	typeTitle
Lyons	Bug
Lyons	Fairy

Result Grid

三、实验中遇到的困难及解决办法

（1）困难1：安装和布置好写代码的界面

MySQL对于我来说是相对比较陌生的一个软件，从安装和布置好写代码的界面我进展的较慢，不过在参考文献的基础上，我解决了这个非技术性的困难。

（2）困难2：对于题意的理解

这是一个相对比较大的项目作业，建立在一个比较完整的场景设置之中，理解起来需要一定时间。我选择结合E-R图片，自己也画一画、理一理关系，最终理解了作业的整个框架。

建议：在这里，我建议下次出题的时候可以更加注重细节和语言的歧义句。

例如第8题中“最多可获得的食物数量”这种表达我认为就没有“最多可获得的食物的**种类的**数量”好。

又如在第9题中，只提到使用ROUND()函数，却没有规定保留的小数位位数，这样会导致答案的不同。最终在**与老师和助教的交流**中，我得知应该保留两位小时。

衷心希望下次作业题意可以更清楚一点！

（3）困难3：嵌套多了之后SQL语句的编写困难

由于上课讲的内容我还没有完全消化，在编写复杂一点的SQL语句的时候，尤其是嵌套比较多的时候，显得有些吃力。解决方案：认真复习PPT知识点，融会贯通。

四、参考文献及致谢

本人所有作业独立完成，所有有关SQL使用的非技术性的参考文献如下，其余均无参考：

MySQL官网

MySQL安装详解

MySQL 数据库保姆级图解教程