

《数据库概论》实验一：用SQL进行数据操作 实验报告

姓名：田馥雯

学号：201220215

联系方式：1259067849@qq.com

实验环境

操作系统：Windows 11

软件版本：mysql-workbench-community-8.0.31-winx64

实验过程

10个SQL查询任务的SQL语句和查询结果截图如下：

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`)。

思路：在 `species` 表中筛查含单词 “this” 的数目。

```
1 SELECT count(id) AS speciesCount
2 FROM species
3 WHERE species.description LIKE '%this%';
```

	speciesCount
▶	90

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`)。

思路：在 `player` 和 `phonemon` 的自然连接表中找到username为 ‘Cook’ 或者 ‘Hughes’ 的元组，按照 `username` 分组，计算每个分组的总能量。

```

1 SELECT username, sum(phonemon.power) AS totalPhonemonPower
2 FROM player, phonemon
3 WHERE player.id = phonemon.player AND (player.username = 'Cook' OR player.username =
  'Hughes')
4 GROUP BY player.username;

```

	username	totalPhonemonPower
▶	Cook	1220
	Hughes	1170

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`)。

思路：在 `team` 和 `player` 的自然连接表中按照 `title` 分组，对每个分组的元组个数计数，结果按照人数降序排列。

```

1 SELECT team.title AS title, count(player.id) AS numberOfPlayers
2 FROM team, player
3 WHERE team.id = player.team
4 GROUP BY team.title
5 ORDER BY numberOfPlayers DESC;

```

	title	numberOfPlayers
▶	Mystic	8
	Valor	6
	Instinct	5

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

思路：在 `species` 和 `type` 的连接表中筛选出 `title` 为 ‘grass’ 的元组，这里的 `species` 具有两个 `type.id` 外键，连接条件应该是 (`species.type1 = type.id` OR `species.type2 = type.id`)。

```

1  SELECT species.id AS idSpecies, species.title AS title
2  FROM species, type
3  WHERE (species.type1 = type.id OR species.type2 = type.id) AND type.title = 'grass';

```

	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

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`)。

思路：在 `player` 表中查询不在购买过食物 `food` 的玩家中的元组，购买过食物 `food` 的玩家通过 `purchase` 和 `item` 的自然连接表且 `type` = 'F' 得到。

```

1  SELECT player.id AS idPlayer, player.username AS username
2  FROM player
3  WHERE player.id NOT IN (
4      SELECT purchase.player
5      FROM purchase, item
6      WHERE purchase.item = item.id AND item.type = 'F');

```

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

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`)。

思路：在 `player`，`purchase` 和 `item` 的连接表中按照 `level` 分组，通过金额降序排序。

注意：总金额是 `sum(item.price * purchase.quantity)`，而不是 `sum(item.price)`，记得乘商品数量。

```

1 SELECT player.level AS level, sum(item.price * purchase.quantity) AS
   totalAmountSpentByAllPlayersAtLevel
2 FROM player, purchase, item
3 WHERE player.id = purchase.player AND purchase.item = item.id
4 GROUP BY player.level
5 ORDER BY totalAmountSpentByAllPlayersAtLevel DESC;

```

	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

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`)。

思路：被购买次数最多也就是分组后 \geq 所有任意物品的被购买次数，被购买次数的计算是通过在 `purchase` 和 `item` 的自然连接表中按照 `id` 分组，计算每组的元组数量得到的。

```
1 SELECT item.id AS item, item.title AS title, count(purchase.id) AS numTimesPurchased
2 FROM purchase, item
3 WHERE purchase.item = item.id
4 GROUP BY item.id
5 HAVING count(purchase.id) >= ALL(
6     SELECT count(purchase.id)
7     FROM purchase, item
8     WHERE purchase.item = item.id
9     GROUP BY item.id);
```

	item	title	numTimesPurchased
►	1	Phoneball	10

8. 找到可获取的食物数量，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`)。

思路：从后面的 `numberDistinctFoodItemsPurchased` 的中文对照我理解这里的可获取食物的数量是指可购买的食物种数，我们要找的玩家即为购买食物种类数对于可购买的食物种数（`food` 表中元组个数），因此，在 `player`, `purchase` 和 `item` 的连接表中按照 `player.id` 分组，计算 `DISTINCT item.id` 的数量，让其等于 `food` 表中元组个数。

```
1 SELECT player.id AS playerID, player.username AS username, count(DISTINCT item.id) AS
   numberDistinctFoodItemsPurchased
2 FROM player, purchase, item
3 WHERE (player.id = purchase.player AND purchase.item = item.id) AND item.type = 'F'
4 GROUP BY player.id
5 HAVING numberDistinctFoodItemsPurchased = ALL(
6     SELECT count(*)
7     FROM food);
```

	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。查询以如下形式返回：(`numberOfPhonemonPairs` , `distanceX`)。

思路：距离最近的对也就是欧式距离<=所有任意两个 `phonemon` 的距离，在 `phonemon` 和 `phonemon` 的笛卡尔乘积中选择 `P1.id < P2.id` 的部分（去重！），这就是去重后的任意两个 `phonemon` 对，通过欧氏距离进行分组，选择分组的欧氏距离<=所有任意两个 `phonemon` 的距离的那一组输出该组的元组个数。

关于欧式距离的计算：`ROUND(sqrt(pow(P1.latitude - P2.latitude, 2) + pow(P1.longitude - P2.longitude, 2)) * 100, 6)`，通过`round()`函数该处所设置的精度为6。

```

1 SELECT count(*) AS numberOfPhonemonPairs, ROUND(sqrt(pow(P1.latitude - P2.latitude, 2)
  + pow(P1.longitude - P2.longitude, 2)) * 100, 6) AS distanceX
2 FROM phonemon AS P1, phonemon AS P2
3 WHERE P1.id < P2.id
4 GROUP BY distanceX
5 HAVING distanceX <= ALL(
6     SELECT ROUND(sqrt(pow(P1.latitude - P2.latitude, 2) + pow(P1.longitude -
7     P2.longitude, 2)) * 100, 6)
8     FROM phonemon AS P1, phonemon AS P2
9     WHERE P1.id < P2.id);

```

	numberOfPhonemonPairs	distanceX
▶	54	0.190957

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`)。

思路：将 `player`，`phonemon`，`species` 和 `type` 表做连接操作后按照 `player.id` 和 `type.id` 分组后记录每一组的 `species.id` 数量，这也就是某一位玩家在某种特定类型上所拥有的物种数量，该物种数量若等于这一类型上的物种数量，就说明这个玩家在这一类型上每一个物种至少有一个 `Phonemon`。

计算该类型的物种数量：将 `species` 和 `type` 表连接，限制 `type.title = typeTitle` 即可。

```

1  SELECT player.username AS username, type.title AS typeTitle
2  FROM player, phonemon, species, type
3  WHERE player.id = phonemon.player AND phonemon.species = species.id AND (species.type1
   = type.id OR species.type2 = type.id)
4  GROUP BY player.id, type.id
5  HAVING count(DISTINCT species.id) = ALL(
6      SELECT count(*)
7      FROM species, type
8      WHERE (species.type1 = type.id OR species.type2 = type.id) AND type.title =
   typeTitle);

```

	username	typeTitle
▶	Lyons	Bug
	Lyons	Fairy

实验中遇到的困难及解决办法

Default 1: 第六题计算总金额时没有乘商品数量

错误SQL语句:

```

1  SELECT player.level AS level, sum(item.price) AS totalAmountSpentByAllPlayersAtLevel
2  FROM player, purchase, item
3  WHERE player.id = purchase.player AND purchase.item = item.id
4  GROUP BY player.level
5  ORDER BY totalAmountSpentByAllPlayersAtLevel DESC;

```

修改后:

```

1  SELECT player.level AS level, sum(item.price * purchase.quantity) AS
   totalAmountSpentByAllPlayersAtLevel
2  FROM player, purchase, item
3  WHERE player.id = purchase.player AND purchase.item = item.id
4  GROUP BY player.level
5  ORDER BY totalAmountSpentByAllPlayersAtLevel DESC;

```

Question 1: 第八题可获取的食物的数量如何理解?

从后面要返回的的 `numberDistinctFoodItemsPurchased` 的中文对照猜测是该 `player` 购买的食物的种类数? 不是很理解, 希望以后的表述可以清晰一点~

Question 2: 第九题 `distanceX` 的数据精度是多少?

查看 `phonemon` 表中 `latitude` 和 `longitude` 发现其精度为小数点后六位, 因此在这里我也保留了小数点后六位, 但是不太确定数据的估读位数。