

## 数据库原理第 4 次作业

42233099 曾慧鑫

### 1. 下列三个 SQL 语句查询均不合法：

```
13
14 SELECT dept_name, min(salary)
15 FROM instructor;
16
17
```

[42803] ERROR: column "instructor.dept\_name" must appear in the GROUP BY clause or be used in an aggregate function  
位置: 9

图 1-1

原因：在 SELECT 语句中，dept\_name 不是聚合函数的一部分，且 GROUP BY 子句缺失。SQL 规则要求，如果使用了聚合函数（如 MIN(salary)），那么 SELECT 语句中出现的非聚合列（dept\_name）必须出现在 GROUP BY 子句中。

```
30
31 SELECT dept_name, min(salary)
32 FROM instructor
33 GROUP BY dept_name
34 HAVING name LIKE '%at%';
```

[42803] ERROR: column "instructor.name" must appear in the GROUP BY clause or be used in an aggregate function  
位置: 72

图 1-2

原因：HAVING 子句用于筛选聚合结果，但 HAVING 后面的 name 并没有出现在 SELECT 或 GROUP BY 中，也没有聚合计算，SQL 语法不允许这样使用。name 不是 GROUP BY 字段，也没有聚合运算，因此不能在 HAVING 子句中使用。

```
22
23 SELECT dept_name
24 FROM instructor
25 WHERE AVG(salary)>20000;
26
27
28
29
```

[42803] ERROR: aggregate functions are not allowed in WHERE  
位置: 40

图 1-3

原因：AVG(salary) 是一个聚合函数，而 WHERE 不能用于聚合函数（聚合函数必须在 HAVING 里）。AVG(salary) 需要结合 GROUP BY 来计算每个 dept\_name 的平均薪资。

#### 2.1 查询工资最高的员工，假设工资最高员工只有一个：

ORDER BY salary DESC 按 salary 降序排列，使得最高工资的员工排在第一位，LIMIT 1 只取第一行数据。输出结果如下：

```
1 ✓ SELECT name
2   FROM instructor
3   ORDER BY salary DESC
4   LIMIT 1;
5
```

输出 university.public.instructor ×

name
Wieand

图 2-1

## 2.2 查询工资最高的员工，假设工资最高员工不只一个：

方法一：SELECT MAX(salary) FROM instructor 计算最高工资。WHERE salary = (...) 选择所有工资等于最高工资的员工。

```
5
6 ✓ SELECT name
7   FROM instructor
8   WHERE salary = (SELECT MAX(salary) FROM instructor);
9
```

图 2-2

方法二：RANK() OVER (ORDER BY salary DESC)给工资最高的员工排名 1。WHERE r = 1 选择所有排名第一的员工，即工资最高的员工。

```
9
10 ✓ SELECT name
11   FROM (
12     SELECT name, salary, RANK() OVER (ORDER BY salary DESC) AS r
13     FROM instructor
14   ) ranked
15   WHERE r = 1;
16
```

图 2-3

方法三：使用 DENSE\_RANK 和 RANK()类似

```
17 ✓ SELECT name
18   FROM (
19     SELECT name, salary, DENSE_RANK() OVER (ORDER BY salary DESC) AS r
20     FROM instructor
21   ) ranked
22   WHERE r = 1;
```

图 2-4

## 2.3 解释 SQL 语句：

```
24 ✓ SELECT 1 IN (1);
25
26 ✓ SELECT 1 = (1);
27
28 ✓ SELECT (1, 2) = (1, 2);
29
30 ✓ SELECT (1) IN (1, 2);
```

图 2-5

SELECT 1 IN (1): 1 IN(1)相当于  $1=1$ , 因为 1 在列表(1)中, 因此返回 TRUE

SELECT 1 = (1): $1 = (1)$ 实际上是  $1 = 1$ , 因此返回 TRUE。

SELECT (1, 2) = (1, 2):(1, 2) = (1, 2)表示两个元组的所有对应元素都相等, 因此返回 TRUE。

SELECT (1) IN (1, 2):SQL 解析(1)为 1, 等同于  $1 \text{ IN } (1, 2)$ 。因为 1 在(1, 2)中, 所以返回 TRUE。