SQL 进阶与性能优化(week5)

一、Thinking&Action:

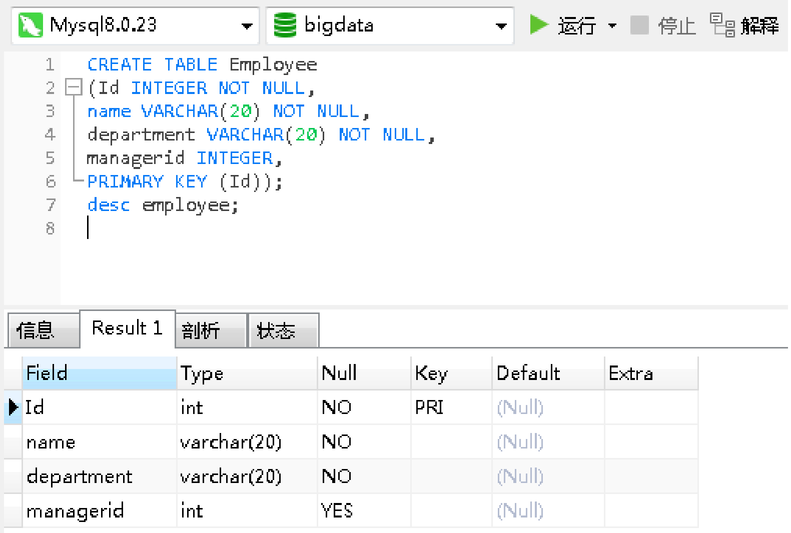
查找至少有 3 名直接下属的经理，employee 表中有 6 名员工数据，表中包含字段 id、name、department、managerid，先从 employee 表中查询；模拟 60 名新员工，添加到 employee 表中，其中 name 为 employee_names 数据表中任意，employee_names 数据表中，有 2 个字段，id 和 name，其中 name 当前数据只有 name_A,name_B,name_C,name_D,name_E,name_F，managerid 为 101-106 之间。

解答步骤如下：

在 navicat 中完成代码编写与调试，本机已安装 MYSQL8.0.23，并可通过 navicat 访问，在新建的 bigdata 数据库下新建相关表。

1、新建表 Employee

```
CREATE TABLE Employee
(Id INTEGER NOT NULL,
name VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
managerid INTEGER,
PRIMARY KEY (Id));
```



```
CREATE TABLE Employee
(Id INTEGER NOT NULL,
name VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
managerid INTEGER,
PRIMARY KEY (Id));
desc employee;
```

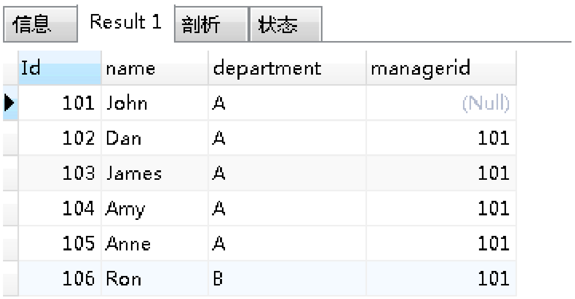
Field	Type	Null	Key	Default	Extra
Id	int	NO	PRI	(Null)	
name	varchar(20)	NO		(Null)	
department	varchar(20)	NO		(Null)	
managerid	int	YES		(Null)	

图 1 表 employee

2、插入数据

```
INSERT INTO `employee` VALUES (101, ’John’, ’A’, NULL);
INSERT INTO `employee` VALUES (102, ’Dan’, ’A’,101);
INSERT INTO `employee` VALUES (103, ’James’, ’A’, 101);
INSERT INTO `employee` VALUES (104, ’Amy’, ’A’,101);
INSERT INTO `employee` VALUES (105, ’Anne’, ’A’, 101);
INSERT INTO `employee` VALUES (106, ’Ron’, ’B’,101);
```

验证 select * from employee;

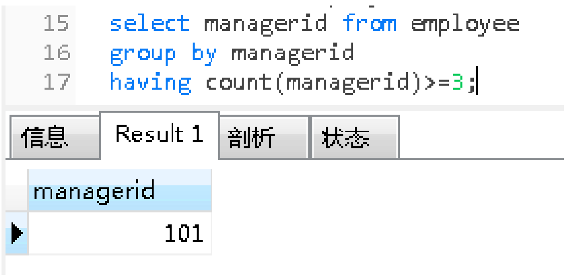


Id	name	department	managerid
101	John	A	(Null)
102	Dan	A	101
103	James	A	101
104	Amy	A	101
105	Anne	A	101
106	Ron	B	101

图 2 表 employee 数据

3、按照 managerid 进行分组，找出组内个数不小于 3 的分组，统计 managerid：

```
select managerid from employee
group by managerid
having count(managerid)>=3;
```



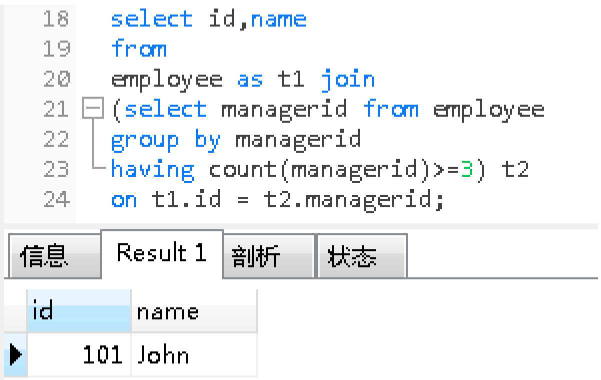
```
select managerid from employee
group by managerid
having count(managerid)>=3;
```

managerid
101

图 3 managerid

4、将查询结果作为临时表，进行联查

```
select id,name
from
employee as t1 join
(select managerid from employee
group by managerid
having count(managerid)>=3) t2
on t1.id = t2.managerid;
```



```
select id,name
from
employee as t1 join
(select managerid from employee
group by managerid
having count(managerid)>=3) t2
on t1.id = t2.managerid;
```

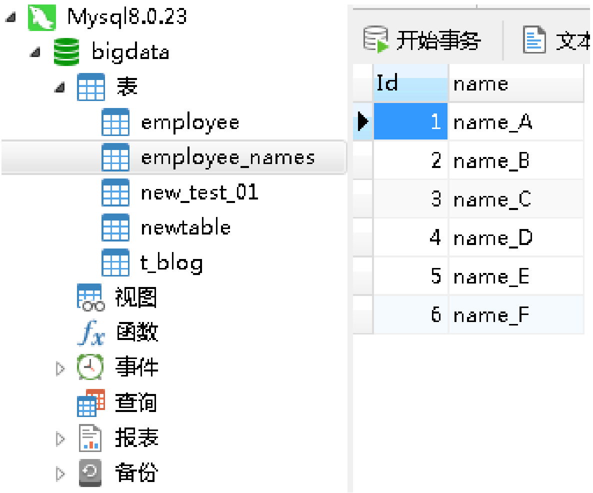
id	name
101	John

图 4 获取 name

得出至少有 3 名下属的经理是 John。

5、新建 employee_names 表

```
CREATE TABLE Employee_names
(Id INTEGER NOT NULL,
name VARCHAR(20) NOT NULL,
PRIMARY KEY (Id));
INSERT INTO `employee_names` VALUES (1, ’name_A’);
INSERT INTO `employee_names` VALUES (2, ’name_B’);
INSERT INTO `employee_names` VALUES (3, ’name_C’);
INSERT INTO `employee_names` VALUES (4, ’name_D’);
INSERT INTO `employee_names` VALUES (5, ’name_E’);
INSERT INTO `employee_names` VALUES (6, ’name_F’);
```



Id	name
1	name_A
2	name_B
3	name_C
4	name_D
5	name_E
6	name_F

图 5 表 employee_names

6、新建函数 add_employee

```
CREATE DEFINER=`root`@`%` PROCEDURE `add_employee`(IN num INT)—定义 num 为输入变量
BEGIN
DECLARE i INT DEFAULT 0;
DECLARE managerid INT;
DECLARE name2 VARCHAR(20);
REPEAT
-- 计算当前员工添加的数量
SET i = i + 1;
-- 随机产生一个 name
SELECT name INTO name2 FROM employee_names ORDER BY RAND() LIMIT 1;
-- 模拟 managerid，101-106 之间的随机数
SET managerid = ROUND(RAND() * (106-101) + 101);
select name2;-- 打印出随机产生的 name
-- 插入员工
INSERT INTO bigdata.employee(name,department,managerid) VALUES (name2,’B’,managerid);
UNTIL i = num
END REPEAT;
COMMIT;
END
```

然后保存此函数，进入 MYSQL 命令行模式下：

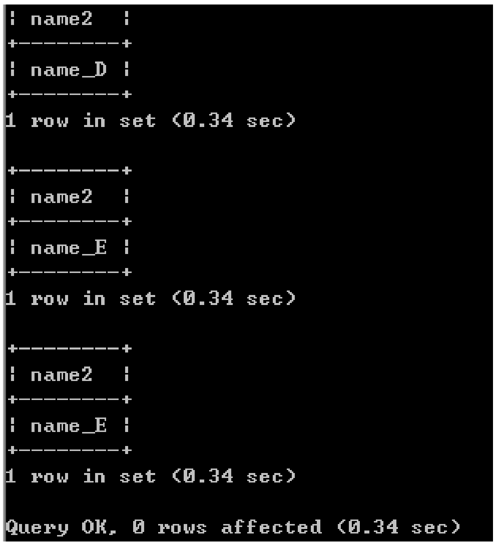
```
use bigdata;
call add_employee(60);
```

报错：fileid ‘Id’ doesn’t have a default value.

没有对表 employee 的字段 id 设置自动递增，运行：

alter table `employee` modify id int auto_increment;

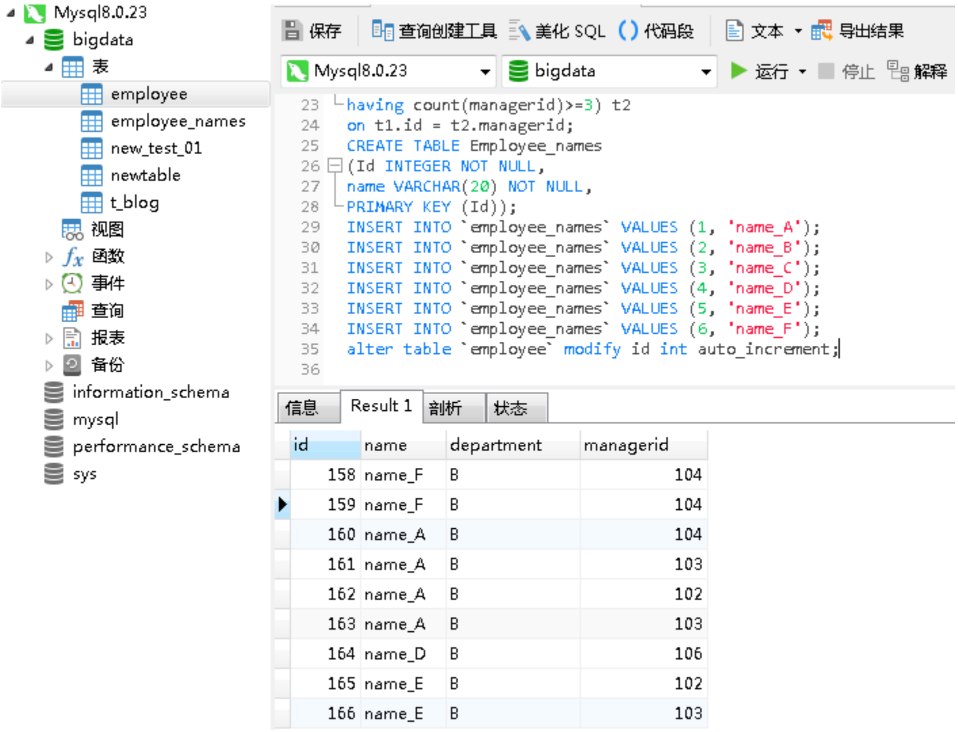
再次运行 call add_employee(60);



```
¦ name2  ¦
+--------+
¦ name_D ¦
+--------+
1 row in set (0.34 sec)

+--------+
¦ name2  ¦
+--------+
¦ name_E ¦
+--------+
1 row in set (0.34 sec)

+--------+
¦ name2  ¦
+--------+
¦ name_E ¦
+--------+
1 row in set (0.34 sec)

Query OK, 0 rows affected (0.34 sec)
```

图 6 name2

返回到 navicat 中，运行 select * from employee;可以看到表中数据已更新。



```
having count(managerid)>=3) t2
on t1.id = t2.managerid;
CREATE TABLE Employee_names
(Id INTEGER NOT NULL,
name VARCHAR(20) NOT NULL,
PRIMARY KEY (Id));
INSERT INTO `employee_names` VALUES (1, 'name_A');
INSERT INTO `employee_names` VALUES (2, 'name_B');
INSERT INTO `employee_names` VALUES (3, 'name_C');
INSERT INTO `employee_names` VALUES (4, 'name_D');
INSERT INTO `employee_names` VALUES (5, 'name_E');
INSERT INTO `employee_names` VALUES (6, 'name_F');
alter table `employee` modify id int auto_increment;
```

id	name	department	managerid
158	name_F	B	104
159	name_F	B	104
160	name_A	B	104
161	name_A	B	103
162	name_A	B	102
163	name_A	B	103
164	name_D	B	106
165	name_E	B	102
166	name_E	B	103

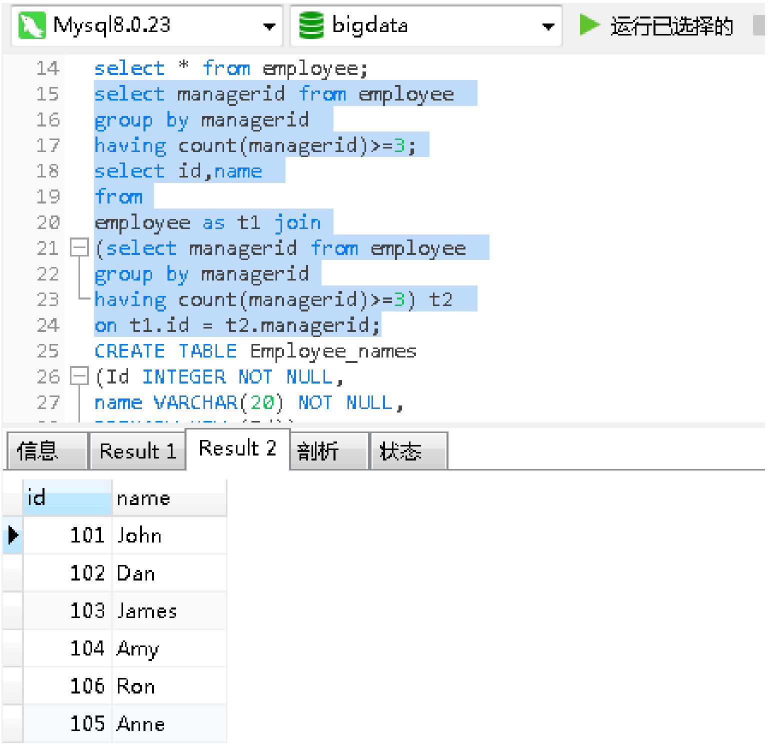
图 7 查询 employee

7、查询下有 3 名直接下属的经理名字，运行：

```
select managerid from employee
group by managerid
having count(managerid)>=3;
select id,name
from
```

```
employee as t1 join
(select managerid from employee
group by managerid
having count(managerid)>=3) t2
on t1.id = t2.managerid;
```

运行结果如下图：



```
select * from employee;
select managerid from employee
group by managerid
having count(managerid)>=3;
select id,name
from
employee as t1 join
(select managerid from employee
group by managerid
having count(managerid)>=3) t2
on t1.id = t2.managerid;
CREATE TABLE Employee_names
(Id INTEGER NOT NULL,
name VARCHAR(20) NOT NULL,
```

id	name
101	John
102	Dan
103	James
104	Amy
106	Ron
105	Anne

图 8 result

可以看出直接有 3 名下属的经理名字是：John、Dan、James、Amy、Ron、Anne。