

# 作业4-10185102142-李泽浩

(1) 统计所有部门的员工数目，输出部门名称和其员工数目。注：使用外连接，保证输出所有部门。

```
SELECT d.DEPT_ID, COUNT(e.EMP_ID) as numbers
FROM department as d LEFT JOIN employee as e on d.DEPT_ID = e.DEPT_ID
GROUP BY d.DEPT_ID;
```

DEPT_ID	numbers
1	14
2	1
3	3
4	0

(2) 查询交易历史表(acc\_transaction)里所有交易对应执行交易的出纳员编号(TELLER\_EMP\_ID)和交易对应账户的开户员工编号

```
SELECT t.TELLER_EMP_ID, a.OPEN_EMP_ID
FROM account as a,acc_transaction as t
WHERE a.ACCOUNT_ID = t.ACCOUNT_ID;
```

TELLER_EMP_ID	OPEN_EMP_ID
1	1
1	1
1	1
1	1
(NULL)	1
1	1
1	1
(NULL)	1
10	10
10	10
10	10
10	10
(NULL)	10
10	10
10	10
10	10

TELLER_EMP_ID	OPEN_EMP_ID
10	10
(NULL)	10
10	10
10	10
10	10
(NULL)	13
13	13
13	13
(NULL)	13
16	16
16	16
16	16
16	16
16	16
16	16

(3) 查询位于“上海市”的客户的单位联系人信息，输出其姓名、职位和在职时间（使用START\_DATE字段计算）。

```
SELECT CONCAT(o.LAST_NAME,o.FIRST_NAME) as name, o.TITLE,
DATEDIFF(CURDATE(),o.START_DATE)/365 as "time 单位:年"
FROM customer as c, officer as o
WHERE c.ADDRESS LIKE "上海市%"
and c.CUST_ID = o.CUST_ID;
```

	name	TITLE	time 单位:年
	华哨	校长	15.9589
	魏俊杰	董事长	18.7973

导入course.sql并完成以下两个题目（第4题和第5题），C\_ID和P\_ID分别为课程号和前期必修课程号。

(4) 利用函数，实现用迭代输出某个课程（作为函数输入）的所有前期必修课程号（包含本课程的课程号），用'&'隔开，按照先父后子的顺序输出，请勿输出多余符号。输入以下指令以查看结果。

```
CREATE DEFINER = CURRENT_USER FUNCTION class(id varchar(255)) RETURNS
varchar(255)
BEGIN
DECLARE res VARCHAR(255) DEFAULT('');
DECLARE temp VARCHAR(255) DEFAULT(id);
WHILE temp IS NOT NULL DO
IF(res == '') THEN
set res = CONCAT(temp,res);
ELSE
set res = CONCAT(temp,'&',res);
END IF;
SELECT GROUP_CONCAT(P_ID) INTO temp
FROM course
```

```

        WHERE FIND_IN_SET(C_ID, temp) > 0;
    END WHILE;
    RETURN res;
END;

```

(5) 利用函数，实现用迭代输出某个课程（作为函数输入）的所有后续可修课程号（包含本课程的课程号），用','隔开，请勿输出多余符号。输入以下指令以查看结果

```

CREATE DEFINER = CURRENT_USER FUNCTION getNext(id VARCHAR(255)) RETURNS
varchar(255)
BEGIN
    DECLARE res VARCHAR(255) DEFAULT('');
    DECLARE temp VARCHAR(255) DEFAULT(id);
    WHILE temp IS NOT NULL DO
        IF res='' THEN
            SET res = CONCAT(res, temp);
        ELSE
            SET res = CONCAT(res, ', ', temp);
        END IF;
        SELECT GROUP_CONCAT(C_ID) INTO temp
        FROM course
        WHERE FIND_IN_SET(P_ID, temp) > 0;
    END WHILE;
    RETURN res;
END

```

(6) 定义视图v\_getEmpHeadOffice，定义时候使用with check option子句，从employee表中找出ASSIGNED\_BRANCH\_ID为1（即所在分支机构为“上海市总行”）的员工信息，输出名、姓、开始/就职日期、职位名称、所在分支结构编号、所在部门编号、上机领导编号。输入下面的指令往该视图中插入数据，能否成功？为什么？

insert into v\_getempheadoffice values('志文','王','2021-01-01','出纳员',2,1,4)

```

#创建视图
CREATE VIEW v_getEmpHeadOffice AS
SELECT first_name, last_name, start_date, title, ASSIGNED_BRANCH_ID
       ,superior_emp_id, dept_id
FROM employee
WHERE ASSIGNED_BRANCH_ID=1
WITH CHECK OPTION;

```

执行插入操作后报错如下：

信息 状态	
sql	message
insert into v_getempheadoffice values('志文','王... 1369 - CHECK OPTION failed 'bank_2.v_getempheadoffice', Time: 0.002000s ...	1369 - CHECK OPTION failed 'bank_2.v_getempheadoffice', Time: 0.002000s

不能成功，因为插入的数据中ASSIGNED\_BRANCH\_ID为2，不满足限制条件，插入后不能通过视图看到修改后的数据。

(7) 定义视图v\_getCustbyProCount，从account表中找出购买产品数量超过2个的客户，输出属性包括账户编号（CUST\_ID）、该账户购买产品个数（命名为Count\_Acc）、该账户各产品可用余额总数（命名为Sum\_Avail\_Balance）、该账户各产品可用余额平均数（命名为 Avg\_Avail\_Balance）。

```
CREATE VIEW v_getCustbyProCount AS
SELECT CUST_ID,
       avg(Avail_Balance) as "Avg_Avail_Balance",
       sum(Avail_Balance) as "Sum_Avail_Balance",
       count(DISTINCT PRODUCT_CD) as "Count_Acc"
FROM account
GROUP BY CUST_ID
HAVING Count_Acc > 2;
```

信息 剖析 状态	
sql	message
CREATE VIEW v_getCustbyProCount AS	OK, Time: 0.001000s

视图

函数

工具

用户

查询

备份

自动运行

模型

对象

无标题@bank\_2 (root)

v\_getcustbyprocount@b...

CUST_ID	Avg_Avail_Balance	Sum_Avail_Balance	Count_Acc
1	267019.25000000	801057.7500	3
4	259444.70666667	778334.1200	3
9	8156.89000000	24470.6700	3

(8) 定义存储过程getEmpInfobyDept, 接收输入参数DEPT\_ID, 列出属于指定部门编号的员工信息, 输出属性包括员工编号EMP\_ID、姓名 (LAST\_NAME与FIRST\_NAME拼接)、开始/就职日期 (START\_DATE)、职位名称 (TITLE)、所在分支机构名称 (NAME)、上级领导姓名 (LAST\_NAME与FIRST\_NAME拼接)

```
DROP PROCEDURE getEmpInfobyDept;

CREATE PROCEDURE getEmpInfobyDept(IN d_id INT)
BEGIN

    SELECT e1.EMP_ID, CONCAT(e1.LAST_NAME,e1.FIRST_NAME) as all_name,
           DATEDIFF(CURRENT_DATE,e1.START_DATE)/365 as time,
           e1.TITLE,b.`NAME`,
           CONCAT(e2.LAST_NAME,e2.FIRST_NAME) as sup_name
    FROM   employee as e1, employee as e2, branch as b
    WHERE  e1.DEPT_ID = d_id
           AND e1.ASSIGNED_BRANCH_ID = b.BRANCH_ID
           AND e1.SUPERIOR_EMP_ID = e2.EMP_ID
    UNION
    SELECT e1.EMP_ID, CONCAT(e1.LAST_NAME,e1.FIRST_NAME) as all_name,
           DATEDIFF(CURRENT_DATE,e1.START_DATE)/365 as time,
           e1.TITLE,b.`NAME`, NULL
    FROM   employee as e1, employee as e2, branch as b
    WHERE  e1.DEPT_ID = d_id
           AND e1.ASSIGNED_BRANCH_ID = b.BRANCH_ID
           AND ISNULL(e1.SUPERIOR_EMP_ID);

END;

#以下用于检验
CALL getEmpInfobyDept (3);
CALL getEmpInfobyDept (2);
CALL getEmpInfobyDept (1);
```

							信息	结果 1
	EMP_ID	all_name	time	TITLE	NAME	sup_name		
	2	钱学冬	18.5945	副行长	上海市总行	赵元源		
	3	孙家雨	21.1863	财务主管	上海市总行	赵元源		
	1	赵元源	19.8192	行长	上海市总行	(NULL)		

信息 结果 1 结果 2						
EMP_ID	all_name	time	TITLE	NAME	sup_name	
5	周一维	17.4219	信贷部主管	上海市总行	李易枫	

信息 结果 1 结果 2 结果 3						
EMP_ID	all_name	time	TITLE	NAME	sup_name	
4	李易枫	18.9808	营业部主管	上海市总行	孙家雨	
6	吴新	17.0822	出纳主任	上海市总行	李易枫	
7	郑楷	16.5836	出纳员	上海市总行	吴新	
8	王瓯	18.3726	出纳员	上海市总行	吴新	
9	冯龚	18.9562	出纳员	上海市总行	吴新	
10	陈易	18.7233	出纳主任	建国支行	李易枫	
11	诸健超	20.4822	出纳员	建国支行	陈易	
12	卫振	18.2712	出纳员	建国支行	陈易	
13	蒋琴琴	20.9342	出纳主任	南京分行	李易枫	
14	沈藤	18.6877	出纳员	南京分行	蒋琴琴	
15	韩虹	18.0438	出纳员	南京分行	蒋琴琴	
16	杨天宝	20.0904	出纳主任	杭州分行	李易枫	
17	朱欣	18.8000	出纳员	杭州分行	杨天宝	
18	秦海陆	18.3452	出纳员	杭州分行	杨天宝	