# 理论课homework2

## 20337251伍建霖

### 4.12

SELECT \* FROM employee;

SELECT \* FROM managers;

SELECT employee\_name

FROM employee NATURAL LEFT OUTER JOIN managers

WHERE manager\_name IS NULL;

SELECT employee\_name FROM employee

WHERE employee\_name NOT IN (SELECT employee\_name FROM managers)

OR

employee\_name IN (

SELECT employee\_name FROM managers

WHERE manager\_name IS NULL );

### 4.14

DROP TABLE IF EXISTS grade\_points;

CREATE TABLE grade\_points(

grade VARCHAR(2),

points REAL);

INSERT INTO grade\_points

VALUES ('A+', 4.33), ('A', 4.0), ('A-', 3.67), ('B+', 3.33),

('B', 3.0), ('B-',2.67), ('C+', 2.33), ('C', 2.0), ('C-', 1.67),

('D+', 1.33), ('D', 1.0), ('D-', 0.67), ('F', 0.0);

-- 创建视图

DROP VIEW IF EXISTS tot\_credits;

CREATE VIEW tot\_credits(year, num\_credits) AS (

SELECT year, SUM(points \* credits)

FROM course NATURAL JOIN takes NATURAL JOIN grade\_points

GROUP BY(YEAR)

ORDER BY (year));

-- 测试

SELECT \* FROM tot\_credits;

### 4.16

a. foreign key (name) references salaried\_worker or hourly\_worker

b. 在 salaried\_worker 中查找name的值，若找不到则在hourly\_worker中查找

### 4.18

public匹配数据库当前与将来用户。授权后B也获得了相应权利，PSQL只支持把授权选项授予给角色，不允许授权给public。

将向public授权更改为直接向B授权后，经实验会导致授权图中的环。另外，起码在SQL中，删除已赋予其他用户权限的用户,往往要借用REASSIGN OWNED和DROP OWNED命令。

### 5.12

程序print了 dog的经理，然后print dog的经理的经理，以此类推，直到我们找到一个没有经理的经理。

### 5.16

with instr count (dept name, number) as (select dept name, count (ID) from instructor group by dept name) select dept name, budget from department, instr count where department.dept name = instr count.dept name and number > 12

### 5.19

with recursive total part(name) as (select part.name from subpart, part where subpart.part id = "P-100" and subpart.part id = part.part id union select p2.name from subpart s, part p1, part p2 where s.part id = p1.part id and p1.name total part.name and s.subpart id = p2.part id) select \* from total par

### 5.28

(select 1, count(∗) from account where 3∗ balance <= (select max(balance) from account) ) union (select 2, count(∗) from account where 3∗ balance > (select max(balance) from account) and 1.5∗ balance <= (select max(balance) from account) ) union (select 3, count(∗) from account where 1.5∗ balance > (select max(balance) from account))