Chapter 4 연습문제 (총 5 문제 4.12, 4.13, 4.14, 4.16, 4.17)

employee (employee_name, street, city)
works (employee_name, company_name, salary)
company (company_name, city)
manages (employee_name, manager_name)

Figure 4.11. Employee database

4.12. For the database of Figure 4.11, write a query to find those employees with no manager. Note that an employee many simply have no manager listed or may have a *null* manager. Write your query using an outer join and then write it again using no outer join at all.

Answer:

1) **select** *employee name*
from *employee* **natural left outer join** *manages*
where *manager name* **is null**

2) **select** *employee name*
from *employee e*
where not exists
(**select** *employee name*
from *manages m*
where *e.employee name* = *m.employee name* **and**
m.manager name **is not null**)

4.13. Under what circumstances would the query

select *

from *student* **natural full outer join** *takes* **natural full outer join** *course*

include tuples with null values for the *title* attribute?

(교제 및 포탈에 올린 *student, takes, course* 스키마 이용)

Answer:

We first rewrite the expression with parentheses to make clear the order of the left outer join operations (the SQL standard specifies that the join operations are left associative).

select *
from (*student* **natural full outer join** *takes*) **natural full outer join** *course*

Given the above query, there are 2 cases for which the *title* attribute is null

a. Since *course id* is a foreign key in the *takes* table referencing the *course* table, the title attribute in any tuple obtained from the above query can be null if there is a course in *course* table that has a null title.
b. If a student has not taken any course, as it is a **natural full outer join**, such a student's entry would appear in the result with a **null** *title* entry.

4.14. Show how to define a view *tot_credits* (*year*, *num_credits*), giving the total number of credits taken by student in each year. (교제 및 포탈에 올린 *takes*, *course* 스키마 이용)

Answer:

create view *tot credits*(*year, tot credits*)
as
(**select** *year*, **sum**(*credits*)
from *takes* **natural join** *course* **group by** *year*)

Note that this solution assumes that there is no year where students didn't take any course, even though sections were offerred.

salaried_worker (name, office, phone, salary)
hourly_worker (name, hourly_wage)
address (name, street, city)

Figure 4.12. Employee database for Exercise 4.16

4.16. Referential-integrity constraints as defined in this chapter involve exactly two relations. Consider a database that includes the relations shown in Figure 4.12. Suppose that we wish to require that every name that appears in *address* appears in either *salaried_worker* or *hourly_worker*, but not necessarily in both.

a) Propose a syntax for expressing such constraints

b) Discuss the actions that the system must take to enforce a constraint of this form

Answer:

a. For simplicity, we present a variant of the SQL syntax. As part of the **create table** expression for *address* we include **foreign key** (*name*) **references** *salaried worker* **or** *hourly worker*

b. To enforce this constraint, whenever a tuple is inserted into the *address* relation, a lookup on the *name* value must be made on the *salaried worker* relation and (if that lookup failed) on the *hourly worker* relation (or vice-versa).

4.17. Explain why, when a manager, say Satoshi, grants an authorization, the grant should be done by the manager role, rather than by the user Satoshi.

Answer:

Consider the case where the authorization is provided by the user Satoshi and not the manager role. If we revoke the authorization from Satoshi, for example because Satoshi left the company, all authorizations that Satoshi had granted would also be revoked, even if the grant was to an employee whose job has not changed.

If the grant is done by the manager role, revoking authorizations from Satoshi will not result in such cascading revocation. In terms of the authorization graph, we can treat Satoshi and the role manager as nodes. When the grant is from the manager role, revoking the manager role from Satoshi has no effect on the grants from the manager role.