2016003581 서그림 데이터베이스 이론 과제1

**2.10**

- primary key 는 relation에서 tuple을 구별하기 위한 수단이다. 한 명의 학생이 한 명 이상의 지도교수를 가질 수 있다고 할 때, s\_id만으로 tuple을 구별할 수 없어지므로 s\_id는 advisor relation의 primary key가 될 수 없다. 따라서 advisor의 primary key로는 {s\_id, i\_id} 가 되어야 한다.

**2.12**

a. PROJECT person\_name (SELECT company\_name=’First Bank Corporation’ (works))

b. PROJECT person\_name, city ((SELECT company\_name=’First Bank Corporation’ (works)) NATURAL JOIN employee)

c. PROJECT person\_name, street, city (SELECT company\_name=’First Bank Corporation’ and salary>10000 (works NATURAL JOIN employee))

**2.13**

a. PROJECT loan\_number (SELECT amount>10000 (loan))

b. PROJECT customer\_name (SELECT balance>6000 (account NATURAL JOIN depositor))

c. PROJECT customer\_name (SELECT balance>6000 and branch\_name = ’Uptown’ (branch NATURAL JOIN account))

**3.11**

a.

**select distinct** name

**from** (course **natural join** takes) **join** student **using** (ID)

**where** course.dept\_name = ‘Comp.Sci.’;

b.

**select** ID,name

**from** student

**except**

**select** ID, name

**from** student **natural join** takes

**where** year <2009;

c.

**select** dept\_name, **max**(salary)

**from** instructor

**group by** dept\_name;

d.

**select** **min**(max\_salary)

**from** ( **select** dept\_name, **max**(salary) **as** max\_salary

**from** instructor

**group by** dept\_name);

**3.12**

a.

**insert into** course

**values**(‘CS-001’, ‘Weekly Seminar’, null, 0);

b.

**insert into** section

**values**(‘CS-001’, 1, ‘Autumn’, 2009, null, null, null);

c.

**insert into** takes

**select** ID, ‘CS-001’, 1, ‘Autumn’, 2009, null

**from** student

**where** dept\_name = ‘Comp.Sci.’;

d.

**delete from** takes

**where** course\_id = ‘CS-001’ **and** section\_id = 1 **and** semester = ‘Autumn’ **and**

year = 2009 **and**

ID **in** ( **select** ID

**from** student

**where** name = ‘Chavez’);

e.

**delete from** takes

**where** course\_id = ‘CS\_001’;

**delete from** section

**where** course\_id = ‘CS\_001’;

**delete from** course

**where** course\_id = ‘CS\_001’;

- 분반을 먼저 삭제하지 않고 이 삭제를 수행한다면 section relation에서 course relation 의 course\_id를 외래키로 불러올 때 해당 tuple이 없어 오류가 발생한다. takes relation 에서 section relation 의 키를 불러올 때도 마찬가지이다.

f.

**delete from** takes

**where** course\_id **in** ( **select** course\_id

**from** course

**where** title like ‘%database%’);

**3.14**

a.

**select** count (\*)

**from** accident

**where** exists ( **select** \*

**from** (person **natural join** owns) **join** participated **using** (licnese)

**where** name = ‘John Smith’ **and**

accident.report\_number = participated.report\_number);

b.

**update** participated

**set** damage\_amount = 3000

**where** license = ‘AABB2000’ **and** report\_number = ‘AR2197’;

**3.16**

a.

**select** employee\_name

**from** works

**where** company\_name = ‘First Bank Corporation’;

b.

**select** E.employee\_name

**from** employee E, works W, company C

**where** E.employee\_name = W.employee\_name **and** E.city = C.city **and**

W.company\_name = C.company\_name;

c.

**select** P.employee\_name

**from** employee P, employee R, manages M

**where** P.employee\_name = M.employee\_name **and**

M.manager\_name = R.employee\_name **and**

P.street = R.street **and** P.city = R.city

**3.20**

**create table** employee

(employee\_name varchar(20),

street char(30),

city varchar(20),

**primary key** (employee\_name));

**create table** works

(employee\_name person\_names,

company\_name varchar(20),

salary numeric(8, 2),

**primary key** (employee\_name));

**create table** company

(company\_name varchar(20),

manager\_name varchar(20),

**primary key** (company\_name));

**create table** manages

(employee\_name varchar(20),

manager\_name varchar(20),

**primary key** (employee\_name));

**3.21**

a.

**select** name

**from** member **natural join** borrowed **natural join** book

**where** publisher = ‘McGraw-Hill’;

b.

**select distinct** M.name

**from** member M

**where not exists** ((**select** isbn

**from** book

**where** publisher = ‘McGrawHill’)

**except**

(**select** isbn

**from** borrowed L

**where** L.memb\_no = M.memb\_no));

c.

**select** publisher, name

**from** (**select** publisher, name, count (isbn)

**from** member M, book B, borrowed L

**where** M.memb\_no = L.memb\_no **and** L.isbn = B.isbn

**group by** publisher, name) **as**

membpub(publisher, name, count\_books)

**where** count\_books > 5;

d.

**with** memcount **as**

(**select** count(\*)

**from** member)

**select** count(\*) / memcount

**from** borrowed

**3.24**

**select distinct** dept\_name

**from** instructor

**where** (**select** **sum**(salary)

**from** instructor

**where** department = dept\_name)

>= (**select** **avg**(s)

**from** (**select** **sum**(salary) **as** s

**from** instructor

**group by** department));