

연습문제 2

(답안)

1. 다음 relation은 모두 4 개의 tuple들로만 이루어졌다고 가정한다.

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
a2	b3	c1	d1
a2	b3	c2	d2
a1	b3	c1	d3
a1	b3	c3	d4

(1) Super key가 될 수 있는 것을 (모두 찾아) 명시하라.

(2) Key가 될 수 있는 것을 (모두 찾아) 명시하라. 그리고 (적절한 가정하에서) primary key를 명시하라.

3) 위의 relation에 새로운 tuple을 insert 할 때 Key Integrity와 Entity Integrity를 각각 위반하는 예를 보여라. 그리고 Key Integrity와 Entity Integrity 모두를 위반하는 예를 보여라.

1. 1) Super Keys :

$\{A, C\}, \{D\}, \{A, B, C, D\}, \{A, C, D\}, \{A, B, C\}, \{A, C, D\}$

$\{A, B, D\}, \{A, C, D\}, \{A, D\}, \{B, D\}, \{B, C, D\}, \{C, D\}$

2) Keys : $\{A, C\}, \{D\}$

Primary Key : $\{D\}$ (Single key를 선택하는 것이 좋음. 그 이유는?)

3) Key Integrity 위반:

Insert $\langle a1, b2, c1, d5 \rangle$, Insert $\langle a3, b2, c1, d1 \rangle$, ... 등

Entity Integrity 위반:

Insert $\langle a4, b2, c1, \text{NULL} \rangle$, Insert $\langle a4, b2, c2, \text{NULL} \rangle$, .

Key/Entity Integrity 모두 위반 :

Insert $\langle a1, b2, c1, \text{NULL} \rangle$, Insert $\langle a2, b2, c1, \text{NULL} \rangle$, ... 등

2. 다음 relation들을 참조하라. 단, 밑줄은 Primary Key를 나타낸다.

학생(학번, 학생명, 전공)

수강(학번, 과목번호)

과목(과목번호, 과목명, 학점)

(1) Foreign key(FK) 들을 명시하라.

학생 : FK : 없음

수강 : FK : {학번}, {과목번호}

과목 : FK : 없음

(2) 과목 relation에서 어떤 과목의 과목번호가 변경된다면 어떤 일이 발생할 수 있는가?

어떤 과목의 번호가 100번에서 200번으로 변경되면, 수강에서 100번을 참조한 tuple들은 모두 참조 무결성 위반됨

(3) 학생 relation에서 어떤 학생이 삭제된다면 어떤 일이 발생할 수 있는가?

만약 어떤 학생이(예: 학번 123456)이 삭제된다면, 수강에서 이 학번을 참조한 tuple들은 모두 참조 무결성 위반됨

3. 다음의 relation들을 참조하여 Foreign Key를 나타내라. 단, 밑줄은 Primary Key를 나타낸다.

서적 (서적번호, 서적명, 가격)

저자 (저자명, 주소, 전화)

저서 (서적번호, 저자명)

서적사본 (서적번호, 도서관명, 사본갯수)

대여 (서적번호, 도서관명, 카드번호, 대여일, 반환일)

도서관 (도서관명, 주소)

대여자 (카드번호, 이름, 주소, 전화)

Foreign key들은 다음과 같음.

‘저서’에서 {서적번호}, {저자명} (2개)

‘서적사본’에서 {서적번호}, {도서관명} (2개)

‘대여’에서 {서적번호}, {도서관명}, {카드번호} (3개)

4. 다음의 relation들과 주어진 제약조건을 참조하라.

선수 (선수명, 소속팀명, 나이)

자동차 (번호판, 모델명, 소유자명, 직장명)

제약조건:

- 1) 자동차의 소유자명과 선수의 선수명은 서로 같은 data type이다.
- 2) 자동차의 직장명과 선수의 소속팀명은 서로 같은 data type이다.
- 3) 자동차는 {소유자명, 직장명} 값을 통해 선수들을 참조한다.
- 4) 선수의 유일한 key는 {선수명, 소속팀명} 이고, 자동차의 유일한 key는 번호판이다.
- 5) 각 선수는 최대 2 대의 자동차를 소유하고, 각 자동차의 소유자는 단 한 명이며 반드시 있어야 한다.

- (1) 위의 각 relation에 tuple들의 예를 각각 5 개 이상 기입하라.
- (2) 위의 relation에서 primary key와 foreign key를 명시하라.
- (3) 위의 relation들을 ER schema로 변환하여 그려라. 즉 원래의 ER schema가 무엇이었는지 그려라.

4. (1)

선수			자동차			
선수명	소속팀명	나이	번호판	모델	소유자명	직장명
김철수	삼성	25	123	모닝	김철수	삼성
홍길동	LG	30	234	모닝	홍길동	LG
김철수	LG	25	345	소나타	김철수	삼성
홍길동	삼성	28	456	소나타	홍길동	삼성
홍길동	SK	33	567	소나타	홍길동	삼성

(2)

선수 :

Primary Key: {선수명, 소속팀명}

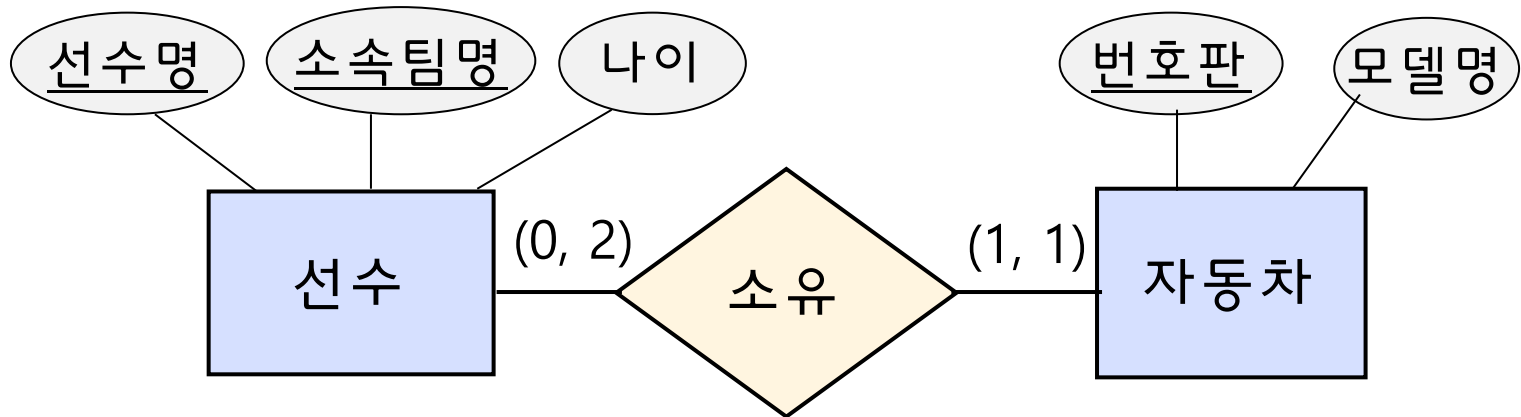
Foreign Key: 없음

자동차 :

Primary Key: {번호판}

Foreign Key: {소유자명, 직장명}

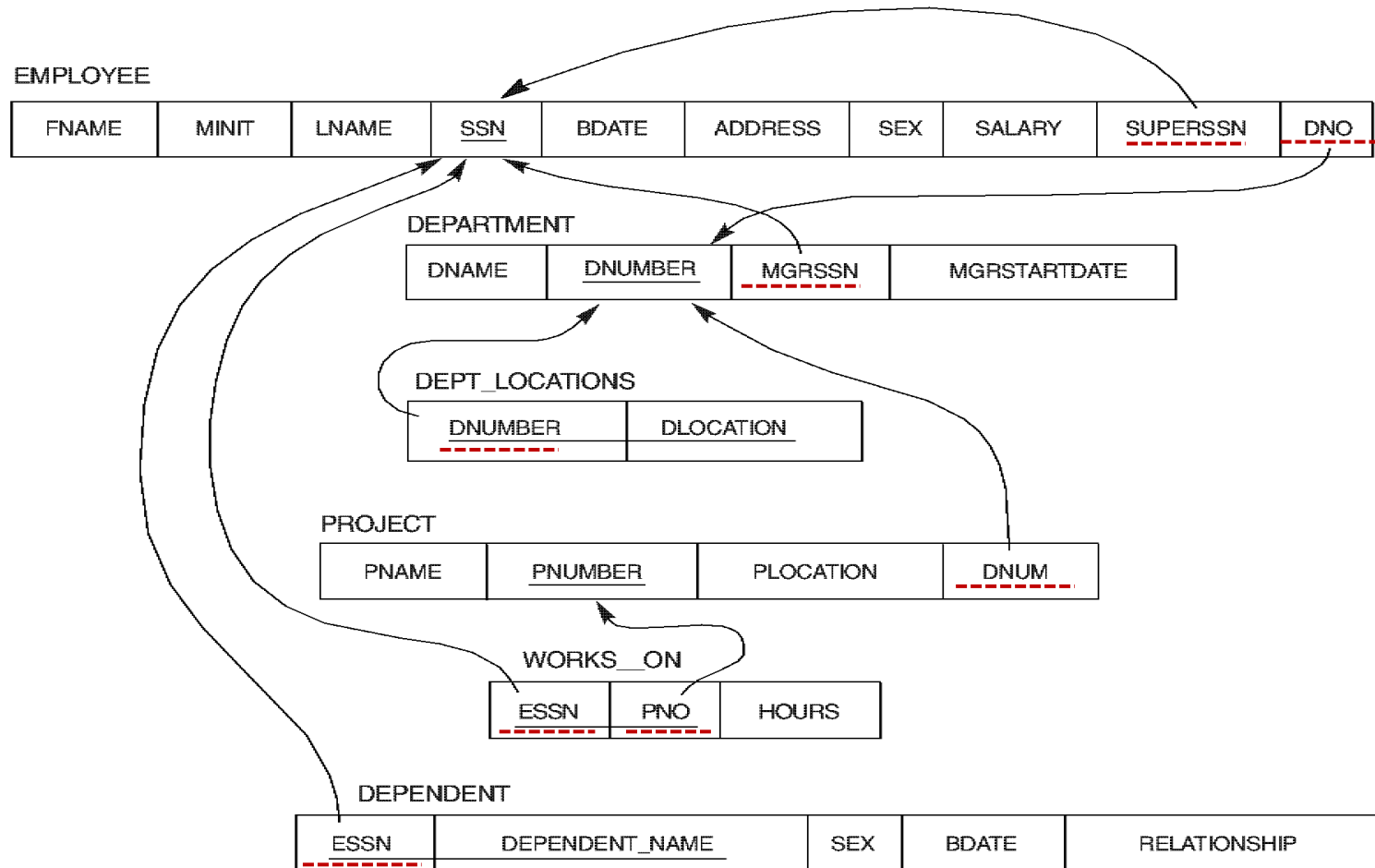
(3) ER Schema



- 선수의 PK는 {선수명, 소속팀명}, 자동차의 PK는 {번호판};
- ER schema에는 FK를 명시하지 않음. 즉 선수와 자동차의 relationship만 도출함. FK는 이 ER schema를 relation 구조로 변환하는 과정에서 도출됨.

다음의 COMPANY 데이터베이스를 참조하라. 실선 밑줄은 Primary Key를, 점선 밑줄은 Foreign Key를 나타낸다..

Figure 7.7 Referential integrity constraints displayed on the COMPANY relational database schema diagram.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

5. 앞의 COMPANY 데이터베이스 예제를 참조하여, 다음 각 연산에서 참조 무결성(referential integrity)의 위반 유무를 판단하라. 만약 위반시에 어느 relation들이 영향을 받아 위반했는지 적어라.

1) Delete <Jennifer, . . 987654321, 1941-06-20,. . . > from EMPLOYEE
: 위반했음. EMPLOYEE, DEPARTMENT, WORKS_ON, DEPENDENT
모두 영향 받아 위반 했음.

2) Insert a new employee <David, . 35678890. ., 3> into EMPLOYEE
: 위반 했음. DEPARTMENT 영향 받아 위반했음.

3) Delete department with Dname = 'Research' from DEPARTMENT
: 위반 했음. EMPLOYEE, PROJECT, DEPT_LOCATIONS 모두
영향 받아 위반했음.

4) Delete all tuple(s) with Dnumber = 5 from DEPT_LOCATIONS
: 위반 안 했음. 영향 받는 relation들 없음.

5) Update Dnumber 5 by 7 from DEPARTMENT

: 위반 했음. EMPLOYEE, DEPT-LOCATION, PROJECT 모두 영향 받음.

6) Delete tuple(s) with Pno = 10 from WORKS-ON

: 위반 안 했음. 영향 받는 relation들 없음.

7) Update John Smith's Super-SSN by 234567890

: 위반 했음. EMPLOYEE 영향 받음.

8) Delete tuple(s) with Pname = Product Y from PROJECT

: 위반 했음. WORKS_ON 영향 받음.

6. 다음의 각 연산을 수행한 결과에 생성될 수 있는 tuple의 최소 개수와 최대 개수를 각각 계산하라. Relation R1 과 R2의 tuple 개수는 각각 N1 과 N2이고, $N1 > N2 > 0$ 이다.

(1) $R1 - R2$: 최소 $N1 - N2$, 최대 $N1$

(2) $R1 \cup R2$: 최소 $N1$, 최대 $N1 + N2$

(3) $R1 \bowtie R2$: 최소 0, 최대 $N1 * N2$

7. 다음 각 연산을 Relation R에 수행한 결과에 생성되는 tuple들의 최소 개수와 최대 개수를 각각 계산하라. 단 R의 tuple들의 수는 N이다.

(1) $\sigma_{A=5 \text{ AND } B=5}(R)$ 최소 0, 최대 N

(2) $\pi_{A, B}(R)$ 최소 1, 최대 N

(단 A, B는 R에 있는 임의의 attribute 이름이다.)

8. 다음의 Relational Schema를 참조하여 각 Query에 대해 Relational Algebra 식으로 표현하라.

사원(사원명, 나이, 도시명)

근무(사원명, 회사명, 봉급)

회사(회사명, 도시명)

관리(사원명, 상관사원명)

- 1) 사원들 가운데 자기들이 근무하는 회사와 같은 도시에 거주하는 사원들의 이름을 찾아라.
- 2) '현대자동차'에 근무하지 않는 사원들의 이름을 찾아라.
- 3) 각 회사는 여러 도시에 분산되어 있을 수 있다고 가정하자.
'현대자동차' 회사가 위치해 있는 모든 도시들에 위치한 회사들의 이름을 찾아라.

1) $\text{Result} \leftarrow \pi_{\text{사원명}} ((\text{근무} \bowtie_{\text{회사명=회사명}} \text{회사}) \bowtie_{\text{도시명=도시명}} \text{사원})$

/* 즉 근무, 사원, 회사 의 세 개의 relation들을 Equi-Join한다.
여기서 Join 연산의 순서에 대해서는 어느 relation을 먼저 해도
상관없이 같은 결과가 나온다. 즉 교환 법칙이 성립한다. */

2) $\text{Result} \leftarrow \pi_{\text{사원명}} (\sigma_{\text{회사명} \neq \text{'현대자동차'}} (\text{근무}))$

3) $T1 \leftarrow \pi_{\text{도시명}} (\sigma_{\text{회사명} = \text{'현대자동차'}} (\text{회사}))$

/* T1에는 현대자동차가 있는 모든 도시들이 있다. */

$\text{Result} \leftarrow \text{회사} \div T1$

/* 회사를 T1로 나누면 현대자동차가 있는 모든 도시(예: 울산,
수원, 창원)들에 위치한 회사들을 찾을 수 있다. */

9. 다음 relation 들을 참조하여, 각 query를 relational algebra 식으로 표현하라.

교수 (교수명, 나이)
수강 (학번, 과목번호)
학생 (학번, 학생명)
강의 (교수명, 과목번호)

- (1) '김응모'가 강의하는 과목들을 강의하는 교수들의 이름을 검색하라.
- (2) '김응모'가 강의하지 않는 과목들만을 모두 듣는 학생들의 이름을 검색하라.

(1) $T \leftarrow \pi_{\text{과목번호}} (\sigma_{\text{교수명} = \text{김응모}} (\text{강의}))$
/* 김응모가 강의하는 과목들의 과목번호들 */
 $\text{Result} \leftarrow \pi_{\text{교수명}} (T \bowtie \pi_{\text{과목번호} = \text{과목번호}} (\text{강의}))$
/* T에 있는 과목들을 강의하는 교수명들 */

(2) $T1 \leftarrow \pi_{\text{과목번호}}(\text{강의})$

/* 현재 개설된 모든 과목들의 과목번호들 */

$T2 \leftarrow \pi_{\text{과목번호}}(\sigma_{\text{교수명} = \text{김응모}}(\text{강의}))$

/* 김응모가 강의하는 과목들의 과목 번호들 */

$T3 \leftarrow T1 - T2$

/* 김응모가 강의하지 않는 과목들의 과목번호들 */

$T4 \leftarrow (\text{수강} \bowtie_{\text{과목번호} = \text{과목번호}} T2)$

/* 김응모가 강의하는 과목들을 듣는 학생들 */

$T5 \leftarrow \text{수강} - T4$

/* 김응모가 강의하지 않는 과목들을 듣는 (모든) 학생들 */

$T6 \leftarrow T5 \div T2$

/* 김응모가 강의하지 않는 과목들만을 모두 듣는 학생들 */

$\text{Result} \leftarrow \pi_{\text{학생명}}(T6 \bowtie_{\text{학번} = \text{학번}} \text{학생})$

/* 김응모가 강의하지 않는 과목들만을 모두 듣는 학생들 이름 */

10. 다음의 relational algebra 표현은 무엇을 구하는 query인지를 설명하라.
단, 밑줄은 Primary Key(PK)를 나타낸다.

영화 (영화명, 상영년도, 필름색깔)

영화1(영화명1, 상영년도1) $\leftarrow \pi_{\text{영화명}, \text{상영년도}}$ (영화)

영화2 (영화명2, 상영년도2) $\leftarrow \pi_{\text{영화명}, \text{상영년도}}$ (영화)

결과(영화명) $\leftarrow \pi_{\text{영화명}_1}$ (영화1 \bowtie (상영년도1 \neq 상영년도2) AND (영화명1 = 영화명2) 영화2)

여기서 동일한 relation을 서로 join하려면, 같은 이름의 attribute를 rename을 먼저 하여야 한다. 또한 {영화명, 상영년도}가 PK이므로, 같은 해에 동일한 이름의 영화들이 존재하지 않음을 유의하자.

정답은 "동일한 이름의 영화가 두 번 이상 상영된 영화들의 영화명들을 구하라." (예: 킹콩, 1957, 킹콩 1988, 킹콩, 2002, 벤허 1957, 벤허 2022, ..)

Figure 3.6

One possible database state for the COMPANY relational database schema.

EMPLOYEE

<u>Fname</u>	<u>Minit</u>	<u>Lname</u>	<u>Ssn</u>	<u>Bdate</u>	<u>Address</u>	<u>Sex</u>	<u>Salary</u>	<u>Super_ssn</u>	<u>Dno</u>
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

<u>Dname</u>	<u>Dnumber</u>	<u>Mgr_ssn</u>	<u>Mgr_start_date</u>
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

<u>Essn</u>	<u>Pno</u>	<u>Hours</u>
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

<u>Pname</u>	<u>Pnumber</u>	<u>Plocation</u>	<u>Dnum</u>
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	<u>Sex</u>	<u>Bdate</u>	<u>Relationship</u>
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

11. 앞의 그림 3.6의 COMPANY 데이터베이스를 참조하여, 다음 각 query에 대해 Relational Algebra 명령문을 작성하고, 각 query의 결과값이 무엇인지 적어라.

- a) Retrieve the names of employees in department 5 who work more than 10 hours per week on the 'ProductX' project.

$TEMP1 \leftarrow (\sigma_{Pname = 'ProductX'}(PROJECT)) \bowtie_{Pnumber = Pno} (WORKS_ON)$

$TEMP2 \leftarrow EMPLOYEE \bowtie_{SSN = ESSN} (\sigma_{Hours > 10}(TEMP1))$

$RESULT \leftarrow \pi_{Lname, Fname} (\sigma_{DNO = 5}(TEMP2))$

Result:

<u>Lname</u>	<u>Fname</u>
Smith	John
English	Joyce

(b) Retrieve the names of employees that are directly supervised by 'Franklin Wong'.

$TEMP1 \leftarrow \pi_{SSN} (\sigma_{Fname = 'Franklin' \text{ AND } Lname = 'Wong'} (EMPLOYEE))$

$TEMP2 \leftarrow (EMPLOYEE) \bowtie_{SuperSSN = SSN} (TEMP1)$

$RESULT \leftarrow \pi_{Lname, Fname} (TEMP2)$

Result:

<u>Lname</u>	<u>Fname</u>
Smith	John
Narayan	Ramesh
English	Joyce

c) Retrieve the names of employees who have a dependent with the same sex as themselves.

$\text{TEMP1} \leftarrow (\text{EMPLOYEE}) \bowtie_{(\text{SSN} = \text{ESSN}) \text{ AND } (\text{sex} = \text{sex})} (\text{DEPENDENT})$

$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}} (\text{TEMP1})$

Result:

<u>Lname</u>	<u>Fname</u>
Smith	John
Franklin	Wong

d) Retrieve the names of employees who work on every project whose location is in 'Houston'.

$TEMP1 \leftarrow \pi_{ESSN, Pno} (WORKS_ON)$

$TEMP2 \leftarrow \pi_{Pnumber} (\sigma_{Plocation = 'Houston'} (PROJECT))$

$TEMP3 \leftarrow TEMP1 \div TEMP2$

$RESULT \leftarrow \pi_{Lname, Fname} (EMPLOYEE \bowtie_{SSN = ESSN} TEMP3)$

Result:

<u>Lname</u>	<u>Fname</u>
Franklin	Wong

e) List the names of department managers who have no dependents.

$TEMP1 \leftarrow \pi_{Mgr-SSN} (DEPARTMENT)$

$TEMP2 \leftarrow \pi_{ESSN} (DEPENDENT)$

$TEMP3 \leftarrow TEMP1 - TEMP2$

$RESULT \leftarrow \pi_{Lname, Fname} (EMPLOYEE \bowtie_{SSN = Mgr-SSN} TEMP3)$

Result:

<u>Lname</u>	<u>Fname</u>
Borg	James

- f) Retrieve the names and addresses of employees who work on at least one project located in Houston, but whose department has no location in Houston.

$TEMP1 \leftarrow \pi_{ESSN} (WORKS_ON \bowtie_{Pno=Pnumber} (\sigma_{Plocation='Houston'} (PROJECT)))$

$TEMP2 \leftarrow$

$\pi_{Dnumber} (DEPARTMENT) - \pi_{Dnumber} (\sigma_{Dlocation='Houston'} (DEPARTMENT))$

$TEMP3 \leftarrow \pi_{SSN} (EMPLOYEE \bowtie_{Dno = Dnumber} (TEMP2))$

$TEMP4 \leftarrow TEMP1 \cap TEMP3$

$RESULT \leftarrow \pi_{Lname, Fname, Address} (EMPLOYEE \bowtie_{SSN = ESSN} TEMP4)$

Result:

<u>Lname</u>	<u>Fname</u>	<u>Address</u>
Wallace	Jennifer	291 Berry, Bellaire, TX

g) Retrieve the last names of employees who do not work on any project.

$\text{TEMP1} \leftarrow \pi_{\text{SSN}} (\text{EMPLOYEE})$

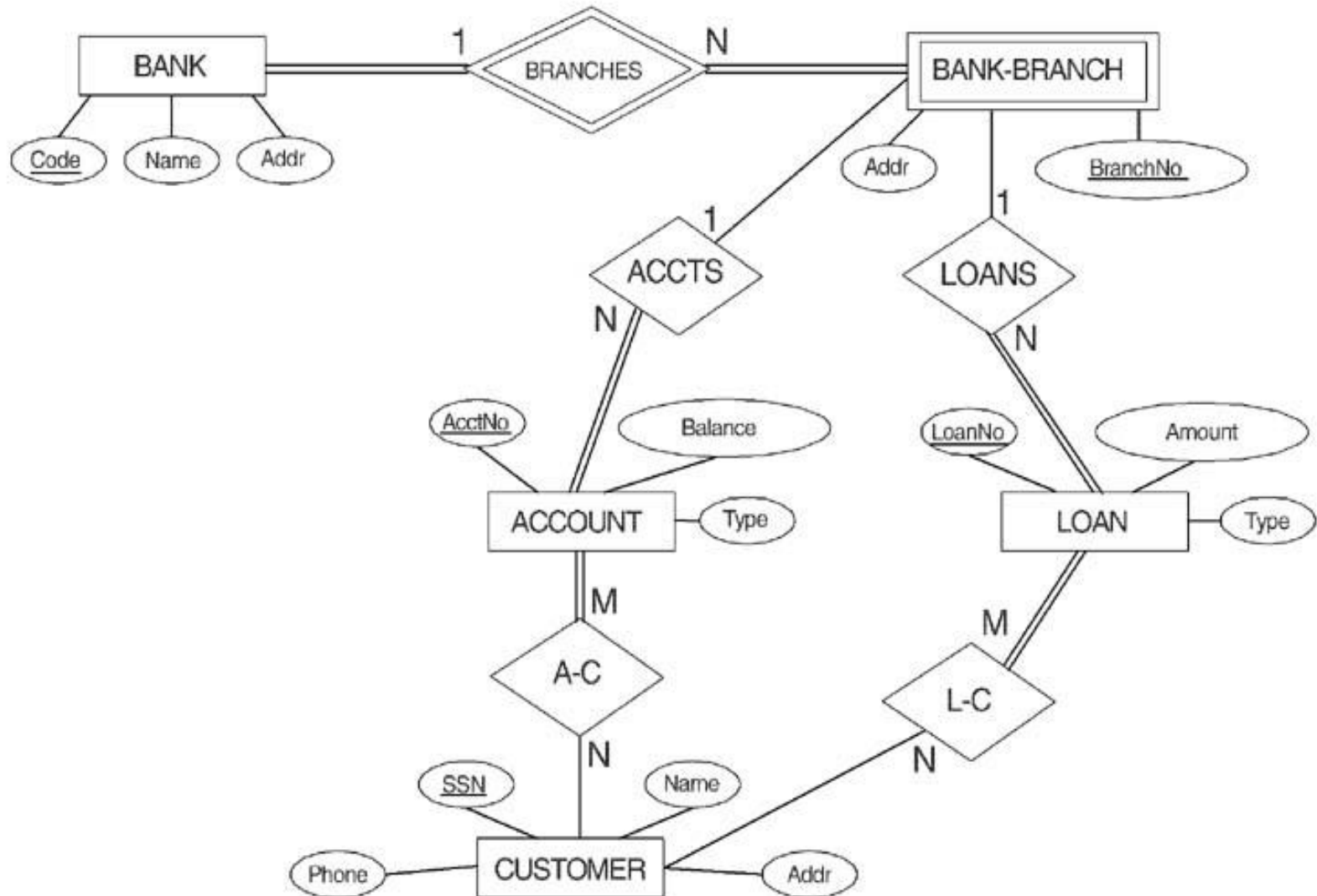
$\text{TEMP2} \leftarrow \pi_{\text{ESSN}} (\text{WORKS_ON})$

$\text{TEMP3} \leftarrow \text{TEMP1} - \text{TEMP2}$

$\text{RESULT} \leftarrow \pi_{\text{Lname}} (\text{EMPLOYEE} \bowtie_{\text{SSN} = \text{SSN}} (\text{TEMP3}))$

Result:
Empty

12. 다음의 ER schema를 relational schema로 변환하라. 각 relation schema에서 반드시 PK를 명시하고, FK (혹시 있는 경우만)를 또한 명시할 것.



12.

BANK (Code, Name, Addr) FK: 없음

BANK_BRANCH (Branch-No, Code, Addr) FK: {Code}

ACCOUNT (AccNo, Balance, Type, Branch-No, Code)

FK: {BranchNo, Code}

CUSTOMER (SSN, Name, Phone, Age) FK: 없음

A-C (Acc-No, SSN) FK: {AccNo}, {SSN}

LOAN (Loan-No, Branch-No, Code, Amount, Type)

FK: {BranchNo, Code}

L-C (LoanNo, SSN) FK: {LoanNo}, {SSN}

위에서 밑줄 그은 부분이 PK임.

13. 다음의 relation들을 참조하여, 아래의 각 relational algebra 표현식과 동등한 SQL 표현식으로 변환하라.

R (A, B, C)

S (D, E, F)

(1) $\pi_A (R)$

(2) $\sigma_{A=17} ((\sigma_{C=17} (R)))$

(3) $\pi_{A, F} (\sigma_{C=D} (R \times S))$

(1) SELECT DISTINCT A
FROM R

(2) SELECT *
FROM R
WHERE (C = 17) AND (A = 17)

(3) SELECT DISTINCT (A, F)
FROM R, S
WHERE R.C = S.D