Chap 5. The Relational Data Model and Relational Database Constraints.

Relational Model Concepts

- Domain, Tuples, Attributes, Relations
  - domain D : set of atomic values (atomic = indivisible)
    - name, data type, format, additional information
    - 주민등록번호 : set of 13 digit valid 주민등록번호
      - format : dddddd - ddddddd
    - 이름 : 사람 이름의 집합 (유효한 사람이름을 나타내는 문자열의 집합)
    - GPA : 0 <= G <= 4
      - format : d.dd
    - 직원나이 : 16 <= A <= 70
  - 학과 : 학과의 집합 (컴퓨터과학, 수학,....): 유효한 학과이름을 나타내는 문자열의 집합

Relational Model Concepts

- Relation Schema R
  : structure of a relation

  - $R(A_1, A_2, \ldots, A_n)$
    - R : relation name
    - $A_1, A_2, \ldots, A_n$: attribute names
    - 각 속성 domain이 relation에서 담당하는 역할 이름
    - domain of A_i = $dom(A_i)$
    - degree of a relation : # of attribute

Relational Model Concepts

- Example : relation schema STUDENT
  - STUDENT(Name, SSN, HomePhone, Address, OfficePhone, Age, GPA)
    - name of relation = STUDENT
    - degree = 7
    - dom(Name) = 학생 이름 집합
    - dom(SSN) = 주민등록번호 집합
    - dom(HomePhone) = 전화번호 집합
    - dom(OfficePhone) = 전화번호 집합
  - *STUDENT - HomePhone, OfficePhone*
    --> 동일 domain 다른 역할

Relational Model Concepts

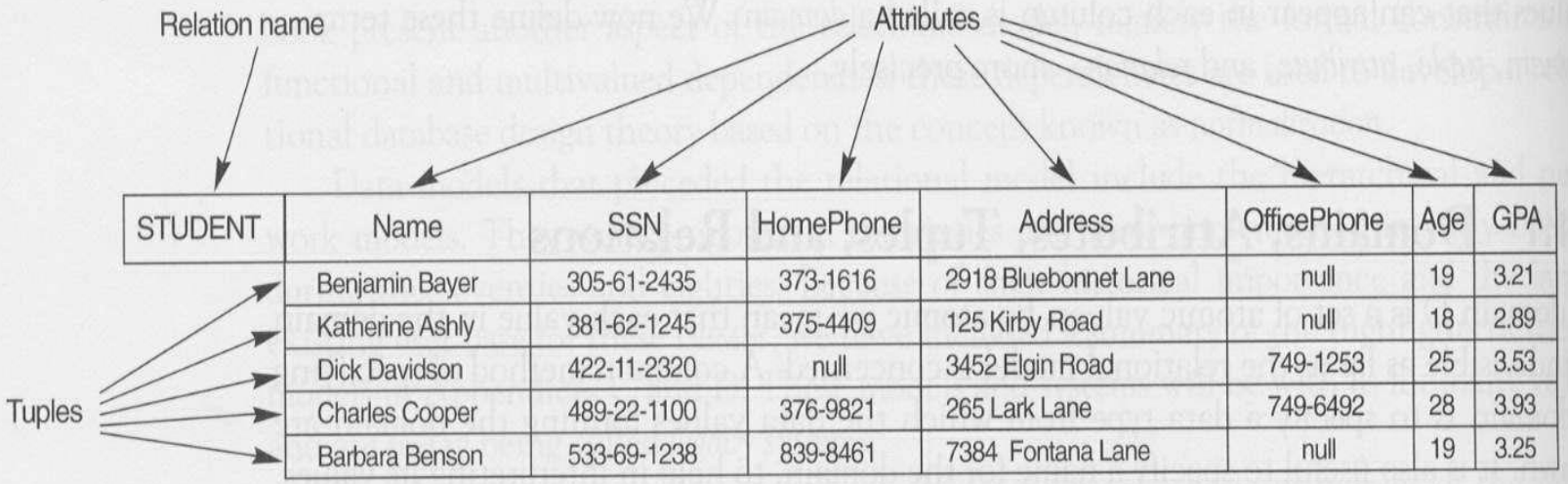
- relation r(R)
  - relation schema R(A_1,A_2,...,A_n)의 relation(or relation instance)
  - a set of n-tuples r = {t_1, t_2,...,t_m}
    - t_i = ordered list of n values <v_1,v_2,...,v_n>
    - $v_i \in$ {dom(A_i) U null}, 1 <= i <= n
  - relation intension = relation schema R
  - relation extension = relation instance r(R)
  - a relation r(R)은 relation schema R을 정의하는 모든 domain의 cartesian product의 부분집합.

Relational Model Concepts

- $r(R) \subset (dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n))$
  - $|D|$ = cardinality of a domain D
  - if all domains are finite, total number of tuples in cartesian product
    $= |dom(A_1)| \times |dom(A_2)| \times \ldots \times |dom(A_n)|$
- 특정시간의 relation instance
  =current relation state
  - mini world의 특정 상태를 나타내는 유효한 tuples 모임
    - relation – dynamic
    - relation schema – almost static

Relation의 특성

- order of tuples in relation
  - no ordering (since relation = set of tuples)
    - table as a file -> physical ordering 존재
- order of values within a tuple
  - 속성과 속성값의 관계



STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Figure 7.1 The attributes and tuples of a relation STUDENT.

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21

Figure 7.2 The relation STUDENT from Figure 7.1, with a different order of tuples.

Relation의 특성

- Alternative definition of a Relation
  - relation schema R = {A_1,A_2,...,A_n}
    - relation r(R) = {t_1, t_2,...,t_m}
      - t_i: tuple
    - mapping r from R to D
      - D=dom(A_1) U dom(A_2) U ... U dom(A_n)
      - t(A_i) ∈ dom(A_i) for 1<=i<=n
    - tuple : a set of (<attribute>,<value>)pairs --> order is not important
    - Values in tuples : atomic value (indivisible)
      - no composite/multi-valued attribute ->first normal form
      - relaxation : nested relation or nonfirst normal form

Relation의 특성

- Interpretation of relation
  - uniform representation of relationship and facts
    - relation schema : declaration
    - facts about entities
    - relationship among entities

Relational Model Notation

- relation schema R of degree n : $R(A_1, A_2,\ldots,A_n)$
- n-tuple t in relation r(R)
  - $t = \langle v_1, v_2,\ldots,v_n \rangle$
  - value v_i in t for attribute A_i : $t[A_i]$
  - subtuple of value $\langle v_u, v_w,\ldots,v_z \rangle$ from tuple t : $t[A_u, A_w,\ldots,A_z]$, A_i: attribute
  - qualified attribute name : relation_name.attribute_name
  - *Example*
    - *STUDENT : current relation state*
    - *STUDENT(Name, SSN,...) : relation schema*
    - *STUDENT.Name, STUDENT.Age*

Relational Model Notation

- Example
  - t = <‘Barbara Benson’, ‘533-69-1238’, ‘839-8461’, ‘7348 Fontana Lane’, null,19, 3.25>
    - t[Name]=<‘Barbara Benson’>
    - t[SSN,GPA,AGE]= <‘533-69-1238’,3.25,19>

Domain Constraints

- value of attribute A – atomic value from dom(A)
  - data type
- standard type
  - integer(short, long)
  - real(single, double)
  - character
  - fixed-length string
  - variable-length string
  - date
  - time
  - money data type

Key Constraints

- relation : a set of tuples

  => all tuples must be distinct
- Superkey of relation schema R : SK
  - subsets of attributes of relation schema R such that $t_1[SK] =! t_2[SK]$ where $t_1, t_2 \in r(R)$
  - 모든 relation schema는 적어도 한개이상의 superkey

  - Key K of relation schema R : minimal superkey
    - superkey 면서 K에서 어느 속성 한 개를 제거했을때 더 이상 superkey가 안되는 superkey

Key Constraints

- key의 값은 relation의 한 tuple을 찾는데 사용함
- key는 mini-world에 의해 결정되며 relation schema에서 정의됨
  - time-invariant
- relation schema에는 한 개 이상의 key가 존재
  - candidate key (그림 7.4)
  - primary key (by DB designer) (그림 7.5)
    - candidate key 중에서 선택
    - 단순속성 or 적은 수의 속성을 갖는 복합속성 선택

CAR	LicenseNumber	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

Figure 7.4 The CAR relation with two candidate keys: LicenseNumber and EngineSerialNumber.

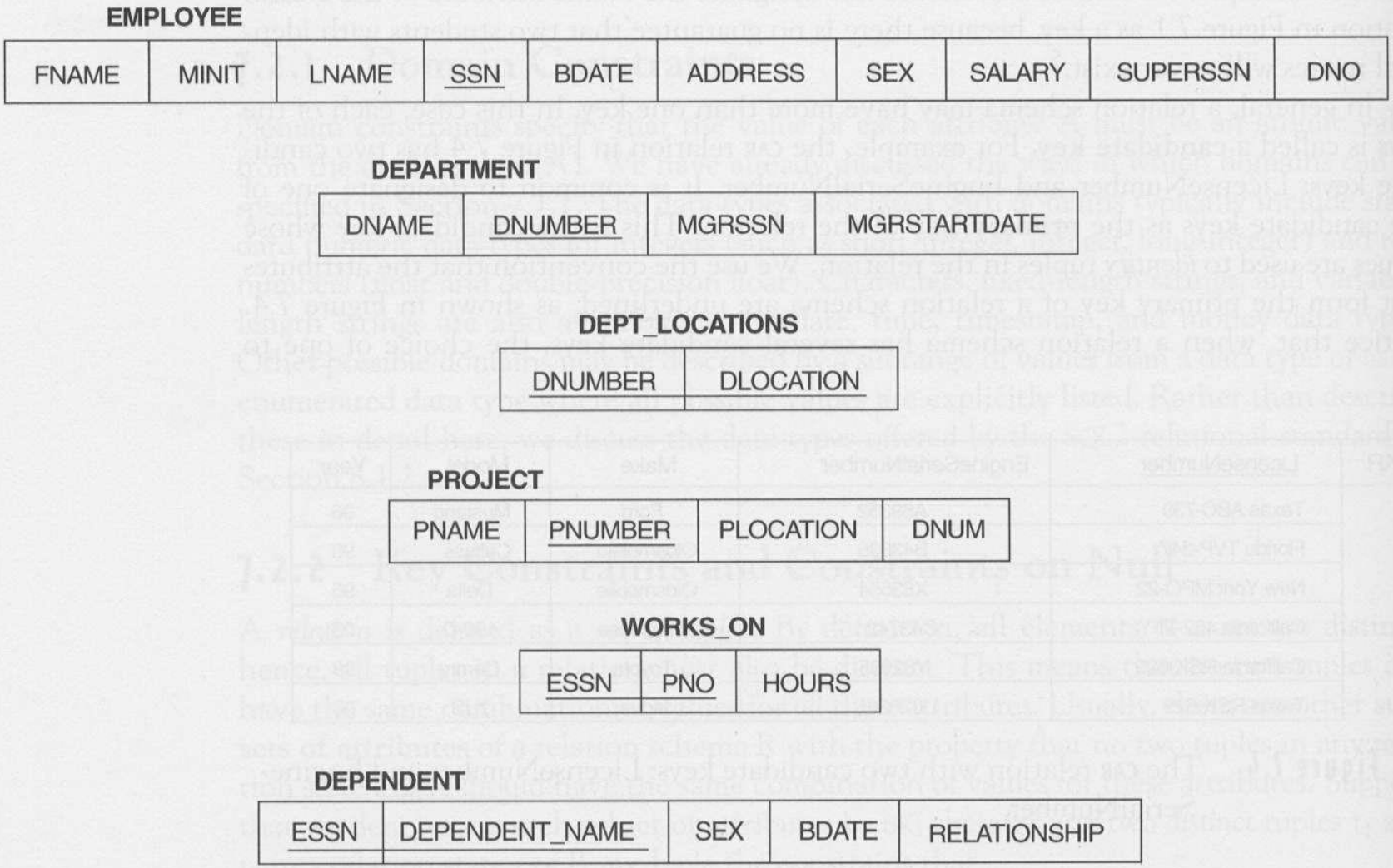


Figure 7.5 Schema diagram for the COMPANY relational database schema; the primary keys are underlined.

Relational Database Schema

- relational database는 여러 개의 relation으로 구성
- relational database schema S = $\{R_1, R_2, \ldots, R_m\}$
  - R_i : relation schema
  - a set of relation schemas and a set of integrity constraints
- relational database instance DB
  - integrity constraint 를 만족하는 relation instance의 집합

- relational database = relational database schema + relational database instance

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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	1968-07-19	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

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

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

Figure 7.6 One possible relational database state corresponding to the COMPANY schema.

Relational Database Schema

- 속성이름
  - 동일 개념 : 다른 이름 사용가능
    - [DNO in EMPLOYEE, DNUM in PROJECT]
    - 동일 개념에 동일이름 사용은 역할 구분 불가능함.
    - 예) 직원(직원 주민등록번호, 관리자 주민등록번호)
      - 직원(SSN, SSN) : SSN 개념 두 번 나옴 (X)
      - 직원(SSN, SUPERSSN)사용 (O)
  - 다른 개념 : 동일이름

    NAME in PROJECT, NAME in DEPARTMENT 사용 가능

Entity & Referential Integrity

- entity integrity constraint
  - primary key ≠ null

- referential integrity constraint
  - 한 relation의 tuple에서 다른 relation을 참조할 때는 항상 참조되는 tuple이 존재함.
  - EMPLOYEE의 DNO=> DEPARTMENT의 DNUMBER 참조
  - entity 간의 relationship.
  - referential integrity constraint

    relation schema에서 속성의 의미 및 역할에 대한 명확한 이해 필요.

Foreign Key

- Foreign key FK (R_1 <--> R_2)
  - FK : a set of attributes FK in relation schema R_1 is a foreign key of R_1
  - relation schema R_1의 속성 집합 FK:
    relation R_2에 대한 R_1의 foreign key
    - dom(FK) = dom(PK), PK=primary key of R_1
    - $t_1[FK]=t_2[FK]$ or null, $t_1 \in r(R_1)$, $t_2 \in r(R_2)$
    - *DNO in EMPLOYEE:foreign key to DEPARTMENT(그림7.6)*
- semantic integrity constraint
  - general constraint on mini-world
  - 대부분 DBMS에서 지원하지 않는다
    - 예) *직원의 월급은 그 직원의 관리자 보다 많으면 안됨*
    - *최대 노동시간은 일주일에 56시간을 초과하지 않는다*

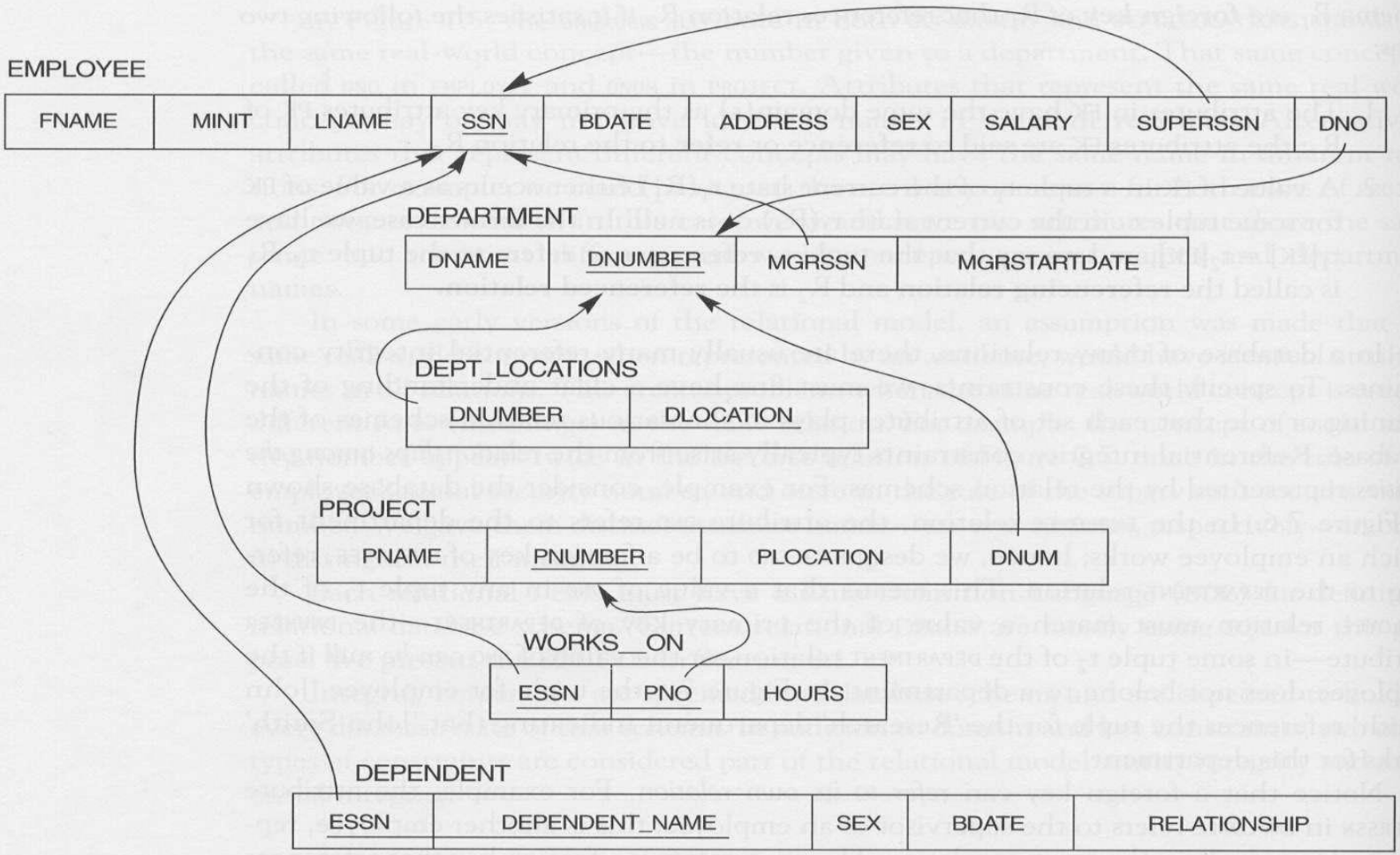


Figure 7.7 Referential integrity constraints displayed on the COMPANY relational database schema diagram.

Insert Operation

- 새로운 tuple을 relation에 첨가
- constraint checking
  - domain constraint
    - 첨가되는 tuple에 있는 속성의 값이 속성의 domain에 없을 때
  - key constraint
    - 첨가되는 tuple의 key값이 relation의 다른 tuple의 key 값으로 존재할 때
  - entity constraint
    - 첨가되는 tuple의 key 값이 null일 때
  - referential constraint
    - 참조되는 relation에 첨가되는 tuple의 foreign key값을 갖는 tuple이 존재하지 않을 때

Insert Operation

- Example
  - *insert ('Cecilia','F', 'Kolonsky', '677678989','05-APR-50', '6357 Windy Lane, Katy, TX', F, 28000, null, 4) into EMPLOYEE*
    - acceptable – satisfy all constraints
    - if SEX = T => domain constraint 어김
    - SSN = 99988777 => key constraint 어김
    - SSN = null => entity constraint 어김
    - DNO = 7 => referential constraint 어김
  - constraint를 어기면 : reject insertion or correct insertion with user interactively

Delete Operation

- Constraint checking
  - Referential integrity constraint
    - 삭제될 tuple이 다른 relation에 의해 참조될 때
  - 대처방안
    - reject deletion operation
    - propagate deletion
      - 삭제된 tuple을 참조하는 모든 tuple 삭제
    - modify referencing attribute values
      - 참조하는 값을 null 또는 다른 값으로 대치
      - 만약 참조하는 속성이 key이면 null로 대치 못함
    - combination of three options

Delete Operation

- Example
  - *delete SSN=999887777 from EMPLOYEE* => not accepted
    - (a) reject deletion
    - (b) delete two tuples in WORKS_ON
  - *delete SSN=333445555 from EMPLOYEE* => not accepted
    - (c) delete tuples in WORKS_ON, DEPENDENT
    - modify or make null in DEPARTMENT(..., MGRSSN,...) EMPLOYEE(..., SUPERSSN,...)

Modify Operation

- Delete + insert
- Example
  - employee에서 SSN=999887777인 tuple의 월급을 28000으로 갱신
    - acceptable
  - employee에서 SSN=999887777인 tuple의 DNO를 7로 갱신
    - unacceptable : referential integrity (department에 7인 부서 없음)
  - employee에서 SSN=999887777인 tuple의 SSN을 987654321로 갱신
    - unacceptable, primary key and referential constraint 어김
    - (SSN=987654321인 다른 tuple 존재, works_on의 참조 조건, foreign key임)

Defining Relation

- 설계된 DB를 DBMS에서 정의하는 법
- Relational DBMS
  - DDL(Data Definition Language) => SQL
- Relational database schema 정의
  - Declare schema COMPANY
- Domain 정의

Defining Relation

- Relation 정의
  - Relation name, attribute name & domains, primary key and other keys, foreign key