

데이터베이스 및 설계

Chap 4. 관계 데이터베이스 (#2/2)

Key 및 integrity constraints



2012.04.09.

오 병 우

컴퓨터공학과

Introduction

● Integrity rule

- ◆ To inform the DBMS of certain constraints in the real world, so that it can prevent impossible configuration of values from occurring

- ◆ (ex)

- (1) supplier numbers must be of the form Snnnn
- (2) part numbers must be of the form Pnnnn
- (3) supplier status values must be in the range 1-100
- (4) part weights must be greater than zero
- (5) shipment quantities must be a multiple of 100

● Two general integrity rules (apply to every database)

- ◆ Entity integrity rule ... primary key
- ◆ Referential integrity rule ... foreign key

무결성, 완전한
상태,
soundness

제약

배치, 형상

Key

● 일반적인 의미

- ◆ 자물쇠를 푸는 열쇠
- ◆ 중요하고 핵심적인 부분 (key man)

● Relation에서의 의미

- ◆ 튜플을 유일하게 식별할 수 있는 애트리뷰트 집합
- ◆ 애트리뷰트 전부를 사용해도 식별 가능하기는 함
- ◆ 일반적으로는 애트리뷰트의 일부를 사용함

● 흔히 혼용해서 사용

- ◆ Primary Key 또는 Candidate Key로 혼용하므로 문맥의 의미를 파악하고 유의해야 함

Candidate key

후보키

◆ 무엇에 대한 후보? (정답) Primary Key

Attribute K (possible composite) of R: candidate key

◆ it satisfies the following two time-independent properties:

– 릴레이션 $R(A_1, A_2, \dots, A_n)$ 에 대한 애트리뷰트 집합 $K = \{ A_i, A_j, \dots, A_k \}$ 로서 아래 성질을 만족하면 후보키

◆ 1. uniqueness: (s#, CITY)

– At any given time, no two tuples of R have same value for K

◆ 2. minimality: (s#)

– If K is composite, then no component of K can be eliminated without destroying the uniqueness property

– 필요없는 attribute가 있으면 candidate key가 아님 (super key라고 부름)

* every relation has at least one candidate key

(\because no duplicate tuples)

Primary key

Primary key

◆ 후보 키(candidate key) 중에서 데이터베이스 설계자가 지정한 하나의 키

◆ One (arbitrarily) chosen key from the set of candidate keys

임의로

– Every relation has a primary key

– Tuple-level addressing mechanism (→some specific tuple)

P
릴레이션에서
P#가
Primary
Key일 경우

– (ex) P(P#, PNAME, COLOR, WEIGHT, CITY)

```
SELECT *
FROM P
WHERE P# = 'P3';
```



(at most) one tuple

```
SELECT *
FROM P
WHERE CITY = 'Paris';
```



unpredictable number of tuples

Alternate key

Alternate key

- ◆ A candidate key that is not the primary key
- ◆ Example: S(S#, SNAME, STATUS, CITY)
 - 1. candidate key: S#, SNAME
 - 2. primary key: S#
 - 3. alternate key: SNAME

대체

후보

Relational model requires “associative addressing”

- ◆ 주소가 아닌 값으로 tuple을 select 가능
- ◆ All relational addressing clearly is associative
(i.e., value-based, not position-based)
- ◆ At logical level
 - no associative hardware in physical level

Key 값을
주소로 사용

정리하면: Candidate Key = Primary Key + Alternate Key

The Entity Integrity Rule

Entity integrity rule

- ◆ No component of the primary key of a base relation is allowed to accept nulls
- ◆ Primary key는 null 일 수 없음

Null (not exist, unknown) (\neq blank or zero)

Null Tag	Value
-------------	-------

- ◆ Information is missing for some reason
 - (e.g., the property does not apply, or the value is unknown, etc.)
- ◆ A value or representation that is understood by convention not to stand for any real value of the applicable attribute
- ◆ 정보 부재를 명시적으로 표현하는 특수한 데이터 값
 - ① 알려지지 않은 값 (unknown value)
 - ② 해당 없음 (inapplicable)

관습, 관례

Foreign key

Foreign key (no converse: 반대로는 안됨)

- ◆ An attribute (possibly composite) of one relation R2 whose values are required to match those of the primary key of some relation R1
 - R1 and R2 not necessarily distinct

Referencing relation

Referenced relation (target relation)

Attribute FK (possible composite) of base relation R2: foreign key

- ◆ it satisfies the following two time-independent properties:
 - 1. each value of FK is either wholly null or wholly nonnull
 - 2. there exists a base relation R1 with primary key PK such that each nonnull value of FK is identical to the value of PK in some tuple of R1

◆ Referential diagram

R1 for S#

S# R2

P# R1 for P#

$S(S\#, SNAME, STATUS, CITY) \leftarrow SP(S\#, P\#, QTY) \rightarrow P(P\#, PNAME, COLOR, WEIGHT, CITY)$

Foreign key

Points

- ◆ 1. foreign key and corresponding primary key: the same underlying domain
- ◆ 2. foreign key need not be a component of the primary key in the referencing relation
 - DEPT (DEPT#, ..., BUDGET, ...) : 여기에서는 primary key
 - EMP (EMP#, ..., **DEPT#**, ..., SALARY, ...) : 여기에서 DEPT#는 primary key가 아님
- ◆ 3. a given relation can be both a referenced relation and a referencing relation
 - $R_n \rightarrow R_{(n-1)} \rightarrow R_{(n-2)} \rightarrow \dots \rightarrow R_2 \rightarrow R_1$: referential path
- ◆ 4. relations R1 and R2 are not necessarily distinct
 - EMP(EMP#, ..., SALARY, ..., MGR_EMP#, ...) : MGR_EMP#는 팀장 (6번 참조)
- ◆ 5. referential cycle
 - $R_n \rightarrow R_{(n-1)} \rightarrow R_{(n-2)} \rightarrow \dots \rightarrow R_2 \rightarrow R_1 \rightarrow R_n$
- ◆ 6. foreign keys sometimes have to accept null (not exist)
 - Value of MGR_EMP# for the president (사장은 null)
- ◆ 7. foreign-to-primary-key match
 - Certain relationships between tuples

Unknown
이 아니라
not exist

확실한,
틀림없는

The Referential Integrity Rule

Referential integrity rule

Referential integrity rule

- The database must not contain any unmatched foreign key value
(if B references A, then A must exist)

S, P, SP에서 S의 S# 변경 시에 SP를 어떻게 처리?

특정 tuple의 S#를 S1에서 S6로 변경할 때 처리 방법

- 허락하지 않음
- S#에서 변경후 SP에서도 변경
- S#에서 변경후 SP에서 대응값은 NULL로 처리

Foreign Key Rules

Three questions for foreign key

- ◆ 1. can that foreign key accept nulls ? → possible
- ◆ 2. an attempt to delete the target of a foreign key reference
 - 1) RESTRICTED: if no matching referencing tuples (otherwise, rejected)
 - 2) CASCADES : cascades to delete the matching referencing tuples
 - (ex) $R3 \rightarrow R2 \rightarrow R1$ with CASCADES DELETE on a given tuple of relation $R1$
 - 3) NULLIFIES : set to null in the foreign keys of the matching referencing tuples
- ◆ 3. an attempt to update the primary key of the target of a foreign key reference
 - 1) RESTRICTED: if no matching referencing tuples (otherwise, rejected)
 - 2) CASCADES : cascades to update the matching referencing tuples
 - 3) NULLIFIES : set to null in the foreign keys of the matching referencing tuples

연쇄, 폭포처럼 흐른다

example

● 대학(University) 관계 데이터베이스

학생 (STUDENT)	학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
		100	나 수 영	4
	200	이 찬 수	3	전기
	300	정 기 태	1	컴퓨터
	400	송 병 길	4	컴퓨터
	500	박 종 화	2	산공

과목 (COURSE)	과목번호 (Cno)	과목이름 (Cname)	학점 (Credit)	학과 (Dept)	담당교수 (PRname)
		C123	프로그래밍	3	컴퓨터
	C312	자료 구조	3	컴퓨터	황수관
	C324	화일 구조	3	컴퓨터	이규찬
	C413	데이터베이스	3	컴퓨터	이일로
	E412	반 도 체	3	전자	홍봉진

example

● 대학(University) 관계 데이터베이스(cont'd)

등록
(ENROL)

학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75
300	C413	A	95	90
400	C312	A	90	95
400	C324	A	95	90
400	C413	B	80	85
400	E412	C	65	75
500	C312	B	85	80

예제

● R1 ≠ R2인 경우

교수 (교수번호, 교수이름, 학과번호, 직급)

학과 (학과번호, 학과이름, **학과장교수번호**, 학생수)

PK

FK

학생 (학번, 이름, 학년, 학과)

과목 (과목번호, 과목이름, 학점, 학과, 담당교수)

등록 (**학번**, **과목번호**, 성적)

FK

FK

● R1 = R2인 경우

교수1 (교수번호, 교수이름, 학과번호, **학장교수번호**)

PK

FK

Summary : Integrity Constraint

(1) 개체 무결성 (entity integrity)

- ◆ Primary key 값은 언제 어느 때고 null값을 가질 수 없다.

Garbage
(쓰레기)
개념과 비슷

(2) 참조 무결성 (referential integrity)

- ◆ Foreign key 값은 참조된 relation의 primary key 값이거나 null이다.

Dangling
Reference
개념과 비슷

● 데이터베이스 상태(database state)가 항상 만족되어야 함

- ◆ DBMS는 데이터베이스 상태의 변화에도 항상 무결성 제약을 만족시키도록 해야 함
- ◆ 어느 한 시점에 데이터베이스에 저장된 데이터 값
- ◆ Database instance
- ◆ 데이터베이스 상태 변화: insert, delete, update 연산