



Chapter 8: Relational Database Design (Part 2)

Database System Concepts, 6th Ed.

©Silberschatz, Korth and Sudarshan
(Modified by Hyo-Sang Lim)



Some Rules for Functional Dependencies

■ Armstrong's Axioms:

- if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ (**reflexivity**)
- if $\alpha \rightarrow \beta$, then $\gamma \alpha \rightarrow \gamma \beta$ (**augmentation**)
- if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ (**transitivity**)

■ Additional rules:

- If $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds, then $\alpha \rightarrow \beta\gamma$ holds (**union**)
- If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds (**decomposition**)
- If $\alpha \rightarrow \beta$ holds and $\gamma \beta \rightarrow \delta$ holds, then $\alpha \gamma \rightarrow \delta$ holds (**pseudotransitivity**)

The above rules can be inferred from Armstrong's axioms.



Full / Partial Functional Dependency

■ Full Functional Dependency (FFD)

- R에서 속성 집합 Y가 속성 집합 X에 함수적으로 종속되어 있지만, 속성 집합 X의 전체가 아닌 일부분에는 종속되지 않음을 의미
- 예) (고객아이디, 이벤트번호) → 당첨여부

■ Partial Functional Dependency (PFD)

- R에서 속성 집합 Y가 속성 집합 X의 전체가 아닌 일부분에도 함수적으로 종속됨을 의미
- 예) (고객아이디, 이벤트번호) → 고객이름



Repetition → Anomaly

■ 이상(anomaly) 현상

- 불필요한 데이터 중복으로 인해 테이블에 대한 데이터 삽입·수정·삭제 연산을 수행할 때 발생할 수 있는 부작용

■ Type of Anomalies

- Insert Anomaly: 새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제
- Update Anomaly: 중복 tuple 중 일부만 변경하면 데이터가 불일치하게 되는 문제
- Delete Anomaly: tuple을 삭제하면 다른 필요한 데이터까지 함께 삭제되는 데이터 손실 문제



An Example of Anomalies

고객아이디	이벤트번호	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver

- **Insert Anomaly:** 새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제



An Example of Anomalies (cont'd)

- **Insert Anomaly:** 새 데이터를 삽입하기 위해 불필요한 데이터도 함께 삽입해야 하는 문제
 - 아직 이벤트에 참여하지 않은 아이디가 “melon” 이고, 이름이 “성원웅”, 등급이 “gold”인 신규 고객의 데이터는 이벤트참여 릴레이션에 삽입할 수 없음
 - 삽입하려면 실제로 참여하지 않은 임시 이벤트번호(혹은 NULL)를 삽입해야 함

고객아이디	이벤트번호	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver
melon	NULL	NULL	성원웅	gold

← 삽입 불가!



An Example of Anomalies (cont'd)

- **Update Anomaly:** 중복 tuple 중 일부만 변경하면 데이터가 불일치하게 되는 문제
 - 아이디가 “apple”인 고객의 등급이 “gold”에서 “vip”로 변경되었는데, 일부 tuple에 대해서만 등급이 수정된다면 “apple” 고객이 서로 다른 등급을 가지는 모순이 발생

고객아이디	이벤트번호	당첨여부	고객이름	등급
apple	E001	Y	정소화	vip
apple	E005	N	정소화	vip
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver

← 데이터 불일치 발생!

8.7

데이터베이스 개론 (한빛아카데미)



An Example of Anomalies (cont'd)

- **Delete Anomaly:** tuple을 삭제하면 다른 필요한 데이터까지 함께 삭제되는 데이터 손실 문제
 - 아이디가 “orange”인 고객이 이벤트 참여를 취소해 관련 튜플을 삭제하게 되면 이벤트 참여와 관련이 없는 고객아이디, 고객이름, 등급 데이터까지 손실됨

고객아이디	이벤트번호	당첨여부	고객이름	등급
apple	E001	Y	정소화	gold
apple	E005	N	정소화	gold
apple	E010	Y	정소화	gold
banana	E002	N	김선우	vip
banana	E005	Y	김선우	vip
carrot	E003	Y	고명석	gold
carrot	E007	Y	고명석	gold
orange	E004	N	김용욱	silver

← 데이터 손실 발생!

8.8

데이터베이스 개론 (한빛아카데미)



Goals of Normalization

- Let R be a relation scheme with a set F of functional dependencies.
- Decide whether a relation scheme R is in “good” form.
- In the case that a relation scheme R is not in “good” form, decompose it into a set of relation scheme $\{R_1, R_2, \dots, R_n\}$ such that
 - each relation scheme is in good form (e.g., no repetition)
 - FD를 이용해 릴레이션을 연관성이 있는 속성들로만 구성되도록 분해해서 이상 현상이 발생하지 않는 바람직한 릴레이션으로 만들어 감
 - the decomposition is a lossless-join decomposition



Normal Form(NF, 정규형)

- 릴레이션이 normalization된 정도
- 각 정규형마다 제약조건이 존재
 - 정규형의 차수가 높아질수록 요구되는 제약조건이 많아지고 엄격해짐
- 릴레이션의 특성을 고려하여 적합한 정규형을 선택





제 1 정규형(1NF, First Normal Form)

- 릴레이션의 모든 속성이 더는 분해되지 않는 원자 값(atomic value)만 가지면 제 1 정규형을 만족함
- 제 1 정규형을 만족해야 관계 데이터베이스의 릴레이션이 될 자격이 있음

제 1정규형을 만족하지 않는 릴레이션

고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001, E005, E010	Y, N, Y	gold	10%
banana	E002, E005	N, Y	vip	20%
carrot	E003, E007	Y, Y	gold	10%
orange	E004	N	silver	5%

8.11

데이터베이스 개론 (한빛아카데미)



제 1정규형을 만족하는 릴레이션

고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

→ 제1정규형을 만족하지만, 데이터 중복으로 인한 **anomaly** 발생

8.12

데이터베이스 개론 (한빛아카데미)



고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001	Y	vip	10%
apple	E005	N	vip	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%
grape	NULL	NULL	silver	5%

← 데이터 불일치로 인한 갱신 이상

← 데이터 손실로 인한 삭제 이상

← 삽입 불가로 인한 삽입 이상

→ 이상 현상의 발생 이유는?

Primary key에 FFD되지 못한 등급과 할인율 때문

→ 해결 방법은?

PFD가 제거되도록 이벤트참여 릴레이션을 분해 → 제 2 정규형

8.13

데이터베이스 개론 (한빛아카데미)



제 2 정규형(2NF, Second Normal Form)

- 릴레이션이 제 1 정규형에 속하고, primary key가 아닌 모든 속성이 primary key에 FFD되면 제 2 정규형을 만족함

분해 전의 이벤트참여 릴레이션

고객아이디	이벤트번호	당첨여부	등급	할인율
apple	E001	Y	gold	10%
apple	E005	N	gold	10%
apple	E010	Y	gold	10%
banana	E002	N	vip	20%
banana	E005	Y	vip	20%
carrot	E003	Y	gold	10%
carrot	E007	Y	gold	10%
orange	E004	N	silver	5%

부분 함수 종속을 제거하려고 분해

고객 릴레이션

고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

이벤트참여 릴레이션

고객아이디	이벤트번호	당첨여부
apple	E001	Y
apple	E005	N
apple	E010	Y
banana	E002	N
banana	E005	Y
carrot	E003	Y
carrot	E007	Y
orange	E004	N

고객 릴레이션과 이벤트참여 릴레이션은 모두 제 2 정규형에 속함

8.14

데이터베이스 개론 (한빛아카데미)



■ 그렇다면 모든 anomaly가 사라졌을까?

고객 릴레이션

고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

함수 종속성을 여러 개 포함하여
이상 현상이 발생할 수 있음

이벤트참여 릴레이션

고객아이디	이벤트번호	당첨여부
apple	E001	Y
apple	E005	N
apple	E010	Y
banana	E002	N
banana	E005	Y
carrot	E003	Y
carrot	E007	Y
orange	E004	N

함수 종속성을 하나만 포함하여
이상 현상 발생하지 않음

부분 함수 종속 해소

8.15

데이터베이스 개론 (한빛아카데미)



이상 현상의 발생 이유는? **FD들 간의 transitivity가 존재하기 때문**

고객아이디	등급	할인율
apple	gold	15%
banana	vip	20%
carrot	gold	10%
orange	silver	5%
NULL	bronze	1%

← 데이터 불일치로 인한 갱신 이상

← 데이터 손실로 인한 삭제 이상

← 삽입 불가로 인한 삽입 이상

→ 해결 방법은?

Transitivity가 제거되도록 고객 릴레이션을 분해 → 제 3 정규형

8.16

데이터베이스 개론 (한빛아카데미)



제 3 정규형(3NF, Third Normal Form)

- 릴레이션이 제 2 정규형에 속하고, primary key가 아닌 모든 속성이 primary key에 transitivity 성질이 없으면 제 3 정규형을 만족함

분해 전의 고객 릴레이션

고객아이디	등급	할인율
apple	gold	10%
banana	vip	20%
carrot	gold	10%
orange	silver	5%

이행적 함수 종속을 제거하려고 분해

고객 릴레이션

고객아이디	등급
apple	gold
banana	vip
carrot	gold
orange	silver

고객등급 릴레이션

등급	할인율
gold	10%
vip	20%
silver	5%

고객 릴레이션과 고객등급 릴레이션은 모두 제 3 정규형에 속함

8.17

데이터베이스 개론 (한빛아카데미)



보이스/코드 정규형(BCNF, Boyce/Codd Normal Form)

■ 필요성

- 하나의 릴레이션에 여러 개의 candidate key가 존재하는 경우, 제 3 정규형까지 모두 만족해도 이상 현상이 발생할 수 있음

■ 의미: 강한 제 3 정규형(strong 3NF)

- Candidate key를 여러 개 가지고 있는 릴레이션에 발생할 수 있는 이상 현상을 해결하기 위해 제 3 정규형보다 좀 더 엄격한 제약조건을 제시
- 보이스/코드 정규형에 속하는 모든 릴레이션은 제 3 정규형에 속하지만, 제 3 정규형에 속하는 모든 릴레이션이 보이스/코드 정규형에 속하는 것은 아님

8.18

데이터베이스 개론 (한빛아카데미)



제 3 정규형을 만족하지만 보이스/코드 정규형은 만족하지 않는 릴레이션
(FD 분석에 따라 2개 이상의 candidate key가 존재하기 때문)

고객아이디	인터넷강좌	담당강사번호
apple	영어회화	P001
banana	기초토익	P002
carrot	영어회화	P001
carrot	기초토익	P004
orange	영어회화	P003
orange	기초토익	P004

[강좌신청 릴레이션의 candidate key]

- (고객아이디, 인터넷강좌) : primary key
- (고객아이디, 담당강사번호)



이상 현상의 발생 이유는?
담당강사번호가 candidate key가 아님에도 인터넷강좌 속성을 결정하기 때문

고객아이디	인터넷강좌	담당강사번호	
apple	영어회화	P001	
banana	기초토익	P002	← 데이터 손실로 인한 삭제 이상
carrot	영어회화	P001	
carrot	중급토익	P004	← 데이터 불일치로 인한 갱신 이상
orange	영어회화	P003	
orange	기초토익	P004	
NULL	중급토익	P005	← 삽입 불가로 인한 삽입 이상



강좌신청 릴레이션

고객아이디	인터넷강좌	담당강사번호
apple	영어회화	P001
banana	기초토익	P002
carrot	영어회화	P001
carrot	기초토익	P004
orange	영어회화	P003
orange	기초토익	P004

후보키가 아닌 결정자를
제거하려고 분해

고객담당강사 릴레이션과
강좌담당 릴레이션은
모두 BCNF에 속함

고객담당강사 릴레이션

고객아이디	담당강사번호
apple	P001
banana	P002
carrot	P001
carrot	P004
orange	P003
orange	P004

강좌담당 릴레이션

담당강사번호	인터넷강좌
P001	영어회화
P002	기초토익
P003	영어회화
P004	기초토익

8.21

데이터베이스 개론 (한빛아카데미)



Lossless Decomposition

- For the case of $R = (R_1, R_2)$, we require that for all possible relations r on schema R

$$r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$$

- A decomposition of R into R_1 and R_2 is lossless join if at least one of the following dependencies is in FDs:

- $R_1 \cap R_2 \rightarrow R_1$
- $R_1 \cap R_2 \rightarrow R_2$

- In other words, if $R_1 \cap R_2$ forms a superkey of either R_1 or R_2 , the decomposition of R is a lossless decomposition.

- Example:

- $inst_dept (ID, name, salary, dept_name, building, budget)$

can be lossless decomposed into two relations

- $instructor (ID, name, dept_name, salary)$
- $department (dept_name, building, budget)$



Design Goals

- Goal for a relational database design is:
 - 3NF or BCNF
 - Lossless join decomposition
- Interestingly, SQL does not provide a direct way of specifying functional dependencies other than superkeys.

Can specify FDs using assertions, but they are expensive to test, (and currently not supported by any of the widely used databases!)



Overall Database Design Process

- We have assumed schema R is given
 - R could have been generated when converting E-R diagram to a set of tables.
 - R could have been a single relation containing *all* attributes that are of interest (called **universal relation**).
 - Normalization breaks R into smaller relations.
 - R could have been the result of some ad hoc design of relations, which we then test/convert to normal form.