Functional Dependency & Normalization in Relational Data Models

Kudang B. Seminar

Functional Dependency

Functional dependency (FD) defines a constraint between two attributes (possibly groups of attributes) of a table.

Let α and β be sets of attributes of any table R. There exists a functional dependency $\alpha \rightarrow \beta$ if only if each value of attribute α in the rows of *R* has associated with it at any one time precisely one value of β in *R*

The notation $\alpha \rightarrow \beta$ can be read as

(a) α *functionally determines* β, or

(B) β *functionally dependent on* α

Functional Dependency

R

A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

Based on the table R, the following functional dependencies apply:

- $A \rightarrow C$
- $D \rightarrow B$
- $(A,B) \rightarrow D$
- $(A,B) \rightarrow C$
- $(A,B) \rightarrow A$
- $(A,B) \rightarrow B$
- $(A,B) \rightarrow (A,B,C,D)$

AXIOMAS OF FD

Axiom of Reflexivity
If $\beta \subseteq \alpha$ then $\alpha \rightarrow \beta$

Axiom of Augmentation
If $\alpha \rightarrow \beta$ then $(\alpha,\gamma) \rightarrow (\beta,\gamma)$

Axiom of Transitivity
If $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$ then $\alpha \rightarrow \gamma$

Axiom of Union
If $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$ then $\alpha \rightarrow (\beta,\gamma)$

Axiom of Decomposition
If $\alpha \rightarrow (\beta,\gamma)$ then $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$

DEFINITION OF SUPERKEY (SK)

If *R* is table and *att(R)* is a set of all attributes of *R* then *λ*, where *λ* ⊆ *att(R)*, is a superkey of *R* if and only if *λ* → *att(R)*.

DEFINITION OF CANDIDATE KEY (CK)

Candidate Key (CK) is SK which is not a superset of another SK.

DEFINITION OF PRIMARY KEY (PK)

Primary Key (PK) is CK which is chosen to be main identification key.

Introduction

- Normalization is a design technique that is widely used as a guide in designing relational databases.
- A relational table is said to be a particular normal form if it satisfied a certain set of constraints.
- There are currently five normal forms that have been defined. In this section, we will cover the first three normal forms that were defined by E. F. Codd.

Basic Concepts

- **The goal of normalization is to create a set of relational tables that are free of redundant data and that can be consistently and correctly modified**
- **This means that all tables in a relational database should be in the third normal form (3NF)**
- **A relational table is in 3NF if and only if all non-key columns are (a) mutually independent and (b) fully dependent upon the primary key**
- **The first two normal forms (1NF & 2NF) are intermediate steps to achieve the goal of having all tables in 3NF**

Functional Dependency (FD)

- **A column, Y, of the relational table R is said to be functionally dependent upon column X of R if and only if each value of X in R is associated with precisely one value of Y at any given time.**
- **If column X is a primary key, then all columns in the relational table R must be functionally dependent upon X.**
- **Notation R.x —> R.y can be read as either *colom y of table R is functionally dependent upon column x of table R.;* or colom y *of table R is functionally determines colom x of table R***

Full Functional Dependance

- **Full functional dependence** applies to tables with composite keys.
- Column Y in relational table R is fully functional on X of R if it is functionally dependent on X and not functionally dependent upon any subset of X.

Example of Unormalized Form & 1-NF

Student	*StudentID*	*CourseID*
Jones	61521	MAT231, ECO220, HST211
Dewi	61300	HST211
Tony	61425	ENG202, MAT231
Paula	61230	MAT231, ENG202

Table 1. Non-Normal Form (Unnormalized Form)

Student	*StudentID*	*CourseID*
Jones	61521	MAT231
Jones	61521	ECO220
Jones	61521	HST211
Dewi	61300	HST211
Tony	61425	ENG202
Tony	61425	MAT231
Paula	61230	MAT231
Paula	61230	ENG202

every column in every row can contain only a single value & simple (atomic value)

Table 2. 1NF (1st Normal Form)

2nd Normal Form

A table satisfies the 2-NF if and only if it satisfies 1-NF and every non-key column is fully dependent on the primary key. A 2-NF cannot have any column that is functionally dependent on only part (subset) of the primary key

Student	StudentID	BirtDate	Course	CourseID	Credit	Grade	Weight
Jones	61521	12/05/77	Math	MAT231	3	B	3
Jones	61521	12/05/77	Economics	ECO220	3	A	4
Jones	61521	12/05/77	History	HST211	2	B	3
Dewi	61300	04/28/78	History	HST211	2	A	4
Tony	61425	11/01/76	English	ENG202	2	C	2
Tony	61425	11/01/76	Math	MAT231	3	B	3
Paula	61230	06/14/77	Math	MAT231	3	B	3
Paula	61230	06/14/77	English	ENG202	2	C	2

Table 2. 1NF (1st Normal Form) and non 2NF

Known Functional Dependencies

- *StudentID → Student*
- *StudentID → BirthDate*
- *CourseID → Course*
- *CourseID → Credit*
- *StudentID, CourseID → Grade*
- *Grade → Weight*

Table *SC1*:

Student	*StudentID*	*BirtDate*
Jones	61521	12/05/77
Dewi	61300	04/28/78
Tony	61425	11/01/76
Paula	61230	06/14/77

Table *SC2*:

Course	*CourseID*	*Credit*
Math	MAT231	3
Economics	ECO220	3
History	HST211	2
English	ENG202	2

Table *SC3*:

StudentID	*CourseID*	*Grade*	*Weight*
61521	MAT231	B	3
61521	ECO220	A	4
61521	HST211	B	3
61300	HST211	A	4
61425	ENG202	C	2
61425	MAT231	B	3
61230	MAT231	B	3
61230	ENG202	C	2

All tables are 2NF

3rd Normal Form (3NF)

A table satisfies the 3-NF if and only if it satisfies 2-NF and every non-key column is not dependent on other non-key column.

Table SC3 is 2NF but not 3NF!

Table *SC3A*:

StudentID	*CourseID*	*Grade*
61521	MAT231	B
61521	ECO220	A
61521	HST211	B
61300	HST211	A
61425	ENG202	C
61425	MAT231	B
61230	MAT231	B
61230	ENG202	C

Table *SC3B*:

Grade	*Weight*
A	4
B	3
C	2
D	1
E	0

Boyce-Codd Normal Form

Boyce-Codd normal form (BCNF) is a more rigorous version of the 3NF deal with relational tables that had (a) multiple candidate keys, (b) composite candidate keys, and (c) candidate keys that overlapped .

BCNF is based on the concept of determinants. A determinant is a set of attributes on which some of attributes are fully functionally dependent.

A relational table is in BCNF if and only if every determinant is a candidate key.

Example of NON-BCNF

Table *CarDealer*

City	*Car*	*Dealer*
Jakarta	Mercedez	German Auto
Surabaya	Mercedez	Astra
Jakarta	Volvo	German Auto
Surabaya	Volvo	Bertha Auto

It is also known the following dependencies:

Dealer -> City **(1) Meaning:** *Each dealer is in only one city*

Car, City -> Dealer **(2) Meaning:** *For each city that sells that car has only one approved dealer*

SK = {(***Car, City***), (***Car, Dealer***), and (***Car, City, Dealer***)}

CK = {(***Car, City***), (***Car, Dealer***)}

Determinant = {(***Dealer***), (***Car, City***)}

Example of Non-BCNF

Table *CarDealer*

City	*Car*	*Dealer*
Jakarta	Mercedez	German Auto
Surabaya	Mercedez	Astra
Jakarta	Volvo	German Auto
Surabaya	Volvo	Bertha Auto

Table *CarDealer is not BCNF*

Not in CK

It is also known the following dependencies:

Dealer -> City **(1) Meaning:** *Each dealer is in only one city*

Car, City -> Dealer **(2) Meaning:** *For each city that sells that car has only one approved dealer*

SK = {(***Car, City***), (***Car, Dealer***), and (***Car, City, Dealer***)}

CK = {(***Car, City***), (***Car, Dealer***)}

Determinant = {(***Dealer***), (***Car, City***)}

Converting to BCNF

Table *CarDealer* must be split into two tables as follows:

Table *CDealer*

Car	*Dealer*
Mercedez	German Auto
Mercedez	Astra
Volvo	German Auto
Volvo	Bertha Auto

SK = CK = PK = (***Car, Dealer***)

Determinant = { }

Table *CityDealer*

City	*Dealer*
Jakarta	German Auto
Surabaya	Astra
Surabaya	Bertha Auto

SK= {(***Dealer***), (***Dealer, City***)}

CK = PK = (***Dealer***)

Determinant = (***Dealer***)

Therefore, both tables are in BCNF

Conclusions

- A 3-NF table with one candidate key must always be BCNF
- A 3-NF table which has non-overlapping candidate keys must always be BCNF
- A BCNF table must always be 3-NF