UNIT 2

Lecture 32

Normalization

1NF & 2NF

Types of Functional Dependencies

1. Trivial FD
2. Non Trivial FD
3. Partial Dependency
4. Full Functional Dependency
5. Transitive Dependency
6. Multivalued Dependency
7. Join Dependency

Trivial and Non Trivial FD

- **Trivial Dependency** – A functional dependency X→Y is trivial if X ⊇ Y.(i.e X is a super set of Y or Y is a subset of X).

E.g. The FD AB→B and AB→A is a trivial dependency.

- **Non - Trivial Dependency** – A functional dependency X→Y is a non - trivial if X is a not a super set of Y or Y is not a subset of X.

E.g. The FD A→B and B→A is a trivial dependency.

GATE Question

Consider the relation X(P, Q, R, S, T, U) with the following set of functional dependencies

F = {

{P, R} → {S, T}

{P, S, U} → {Q, R}

}

Which of the following is the trivial functional dependency in F^+, where F^+ is the closure of F?

(a) {P, R} → {S, T}

(b) {P, R} → {R, T}

(c) {P, S} → {S}

(d) {P, S, U} → {Q}

[GATE 2015]

Partial Dependency

- **<u>Partial Dependency</u>** – A functional dependency X→Y is a partial dependency if some attribute A ϵ X can be removed from X and the dependency still hold; i.e. for some A ϵ X, (X–{A}) → Y.
- A partial dependency exists when an attribute B is functionally dependent on an attribute A, and A is a component of a multipart candidate key.

E.g. Consider a relation R (A, B, C, D, E) with a set of FDs

F = { A → B,

C → D,

D → E }

This relation has 2 partial dependencies A → B and C → D, because AC is a Candidate key of this relation.

Full Functional Dependency

- **<u>Full Functional Dependency</u>** – A functional dependency X→Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more; i.e. for any attribute A є X, (X – {A}) does not functionally determine Y.
- E.g. Consider a relation R (A, B, C, D, E) with a set of FDs

F = {AB → C,

A → D,

C → E }

This relation has 1 full functional dependency AB → C and 2 partial dependencies A → D and C → E.

Transitive Dependency

- **<u>Transitive Dependency</u>** - A functional dependency X→ Y in a relation schema R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key of R, and both X→ Z and Z→ Y hold.
- A function dependency X → Y is called transitive dependency if and only if both X and Y are non prime attributes.
- Consider attributes A, B, and C, and where A → B and B → C. FDs are transitive, which means that we also have the FD, A → C. We say that C is transitively dependent on A through B.
- E.g. Consider a relation R (A, B, C, D) with a set of FDs

F = { A → BC

B → D }

This relation has 1 transitive dependency B → D because both B and C are non prime attributes. This relation also has 1 full functional dependency A → BC.

First Normal Form (1NF)

- We say a relation is in 1NF if all values stored in the relation are single valued and atomic.
- 1NF places restrictions on the structure of relations. Values must be simple.
- 1NF disallows repeating values, set of values, relations within relations, nested relations.
- 1NF deals with the ‘shape’ of the record.
- To remove the repeating group, either :
  - Flatten the table and extend the key, or
  - Decompose the relation leading to 1NF.

First Normal Form (1NF)

Eg. 1. The following is not in 1NF.

EMP

EmpNum	EmpPhone	EmpDegrees
123	233-9876	
333	233-1231	BA, BCS, PhD
679	233-1231	BSC, MSC

EmpDegrees is a multi valued field:

Employee 679 has two degrees: BSC and MSC

Employee 333 has three degrees: BA, BSC, and PhD

First Normal Form (1NF)

To obtain 1NF relations we must, without loss of information, replace the above with two relations.

Employee

EmpNum	EmpPhone
123	233-9876
333	233-1231
679	233-1231

EmployeeDegree

EmpNum	EmpDegree
333	BA
333	BSC
333	PhD
679	BSC
679	MSC

An outer join between Employee and EmployeeDegree will produce the information that the original table hold.

First Normal Form (1NF)

Eg. 2. The following is not in 1NF.

EMP

EmpNum	EmpPhone	EmpAddress		
		Street	City	Pincode
123	233-9876	Ram Nagar	Raipur	492001
333	233-1231	Shankar Nagar	Durg	491001

To obtain 1NF relations we must, without loss of information, replace the above with simple components of composite attribute EmpAddress.

EMP

EmpNum	EmpPhone	Street	City	Pincode
123	233-9876	Ram Nagar	Raipur	492001
333	233-1231	Shankar Nagar	Durg	491001

First Normal Form (1NF)

Eg. 3. The following is not in 1NF.

EMP

EmpNum	EmpName
123	RAM
333	SHYAM
333	SHYAM
444	GOPAL

EMP

EmpNum	EmpName
123	RAM
333	SHYAM
444	GOPAL

To obtain 1NF relations we must, without loss of information, replace the above with an added primary key constraint that means we replace above relation with a relation in which we make EmpNum as primary key that will remove data redundancy.

Second Normal Form (2NF)

- Second normal form (2NF) is based on the concept of **full functional dependency**.
- A relation schema R is in 2NF if it is in 1NF, and every nonprime attribute is fully functionally dependent on each candidate key of R.
- **General Def. Of 2NF** – A relation schema R is in 2NF if every nonprime attribute A in R is not partially dependent on any key of R.
- A relation is 2NF will not have any partial dependencies.
- If a relation schema is not in 2NF, it can be "second normalized" or "2NF normalized" into a number of 2NF relations in which nonprime attributes are associated only with the part of the primary key on which they are fully functionally dependent.

Second Normal Form (2NF)

E.g. 1 Consider a relation R (A, B, C, D) with a set of FDs

F = { A → B,

C → D }

is not in 2NF, because it has 2 partial dependencies A → B and C → D, with candidate key as AC.

2NF Decomposition Algorithm

1. Compute the closure of a set of FDs F^+.
2. For every partial dependency $X \rightarrow Y$ in F^+, create 2 relations Ri (X ,Y) and Rj (R – Y).
3. If possible merge 2 or more relations on the basis of common candidate keys.

Second Normal Form (2NF)

E.g.1 Consider a relation R (A, B, C, D) with a set of FDs

F = { A → B,

C → D }

Sol : First we compute F^+.

So. F^+ = { A → B,

C → D }

From F^+, we have 2 partial dependencies A → B, and C → D.

R (A, B, C, D)

A → B

R1 (A, B) R2(A, C, D)

C → D

R21 (C, D) R22 (A, C)

So, relations R1 ($\underline{A}$, B), R21 ($\underline{C}$, D) and R22 ($\underline{A, C}$) are the required 2NF decompositions of R.

Second Normal Form (2NF)

E.g.2 Consider a relation R (A, B, C, D, E) with a set of FDs

F = { AB → C,

A → D,

D → E }

is not in 2NF, because it has 2 partial dependencies A → D and A → E, with candidate key as AB.

Second Normal Form (2NF)

E.g.2 Consider a relation R (A, B, C, D) with a set of FDs

F = { $AB \rightarrow C$,

$A \rightarrow D$,

$D \rightarrow E$ }

Sol : First we compute F^+.

So. F^+ = { $AB \rightarrow CDE$,

$A \rightarrow DE$,

$D \rightarrow E$ }

From F^+, we have 2 partial dependencies $A \rightarrow D$, and $A \rightarrow E$.

R (A, B, C, D, E)

$A \rightarrow DE$

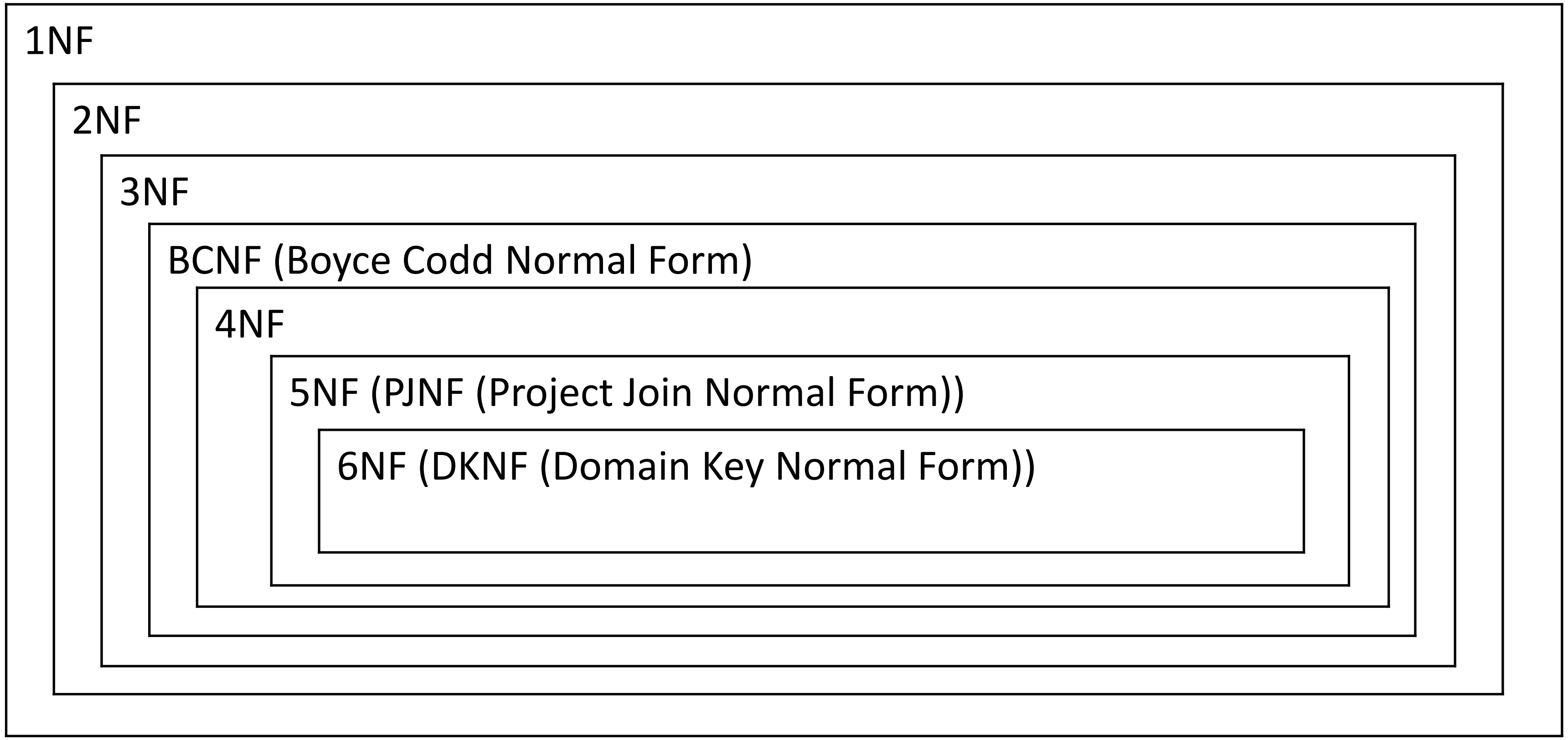
R1 (A, D, E)   R2(A, B, C)

So, relations R1 (<u>A</u>, D, E) and R2 (<u>A, B</u>, C) are the required 2NF decompositions of R.

Objective of Normalization

- *“to create relations where every dependency is on the key, the whole key, and nothing but the key”.*

Levels of Normalization



For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JlTp1iim6aOTRuw?sub_confirmation=1