



# NORMALISASI

## A. MATERI

Dalam merancang database harus dapat dijawab apabila kita diberikan data, maka bagaimana kita menentukan struktur logik yang tepat untuk data tersebut, atau bagaimana kita menentukan relation-relation yang diperlukan dan apa atributnya.

seperti telah dijelaskan di depan bahwa semua relation dalam relational database selalu sudah ternormalisasi, dalam arti bahwa semua relation sudah didefinisikan terhadap domain sederhana, yaitu domain yang hanya berisi nilai atomik. Dalam normalisasi lanjutan kita berusaha untuk menghilangkan/mengurangi data yang duplikasi atau mubazir agar supaya mendapatkan bentuk yang baik, hemat tempat, hemat waktu, hemat biaya dan yang memberikan respon yang baik dan cepat.

Suatu relation dikatakan sudah berada pada bentuk normalisasi tertentu bila memenuhi beberapa batasan tertentu pada tingkat tersebut. Tingkat normalisasi yang lebih tinggi dianggap lebih baik dari tingkat dibawahnya.

Tingkat-tingkat normalisasi :

1. Relation umum (yang belum dan yang sudah ternormalisasi)
2. 1NF (First Normal Form) relation yang sudah ternormalisasi.
3. 2NF (Second Normal Form) relation.
4. 3NF (Third Normal Form) relation.
5. BCNF (Boyce Codd Normal Form) relation.
6. 4NF (Fourth Normal Form) relation.
7. PJ/NF (Project Join Normal Form) atau 5NF (Fifth Normal Form) relation

Untuk 6NF dan 7NF dan seterusnya belum terpikirkan saat ini, walaupun begitu nanti akan kita bicarakan salah satu tingkat normalisasi lagi yaitu DKNF atau Domain-Key-Normal Form.

### **FUNCTIONALLY DEPENDENT DAN FUNCTIONALLY DETERMINES**

Suatu atribut Y disebut functionally dependent terhadap atribut X dalam suatu relation R bila setiap nilai X di R hanya ada satu hubungan ketergantungan dengan nilai Y di R pada suatu saat.

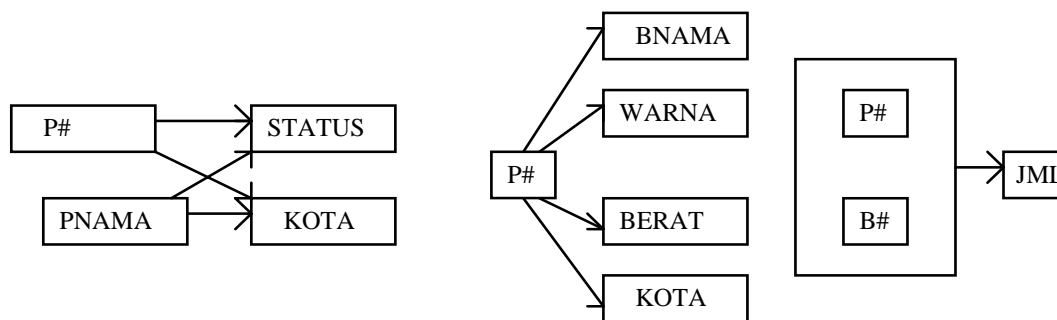
Contoh dalam database pemasok dan bahan APBM, atribut PNAMA, STATUS dan KOTA dari relation P semua tergantung pada (functionally dependent) terhadap atribut P# karena setiap nilai P# hanya ada satu hubungan dengan nilai-nilai di PNAMA, STATUS dan KOTA.

Dalam simbol dapat kita tulis sebagai :



P.P#  $\longrightarrow$  P.PNAMA  
 P.P#  $\longrightarrow$  P.STATUS  
 P.P#  $\longrightarrow$  P.KOTA  
 Atau P.P  $\longrightarrow$  P.(PNAMA,STATUS,KOTA)  
 P.P#  $\longrightarrow$  P.PNAMA menyatakan P.PNAMA functionally dependent terhadap P.P# atau sebaliknya P.P# functionally determines P.PNAMA.

Functional dependence (FD) adalah salah satu dari batasan integritas (Integrity constraints).



Gambar 1. Functional Dependence di relation P, B, dan PB

**FIRST NORMAL FORM (1NF)**

Gambar 2. Contoh normalisasi sederhana

PB1

P#	BJ	
	B#	JML
P1	B1	300
	B2	200
	B3	400
	B4	200
	B5	100
	B6	100
P2	B1	300
	B2	400
P3	B2	200
P4	B2	200
	B4	300
	B5	400

(Sebelum dinormalisasi) masih ada yang tidak terisi atau tidak diketahui (null)  
(1NF)

PB2

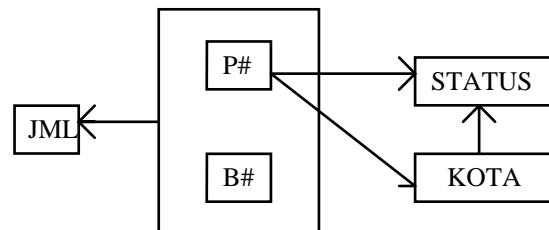
P#	BJ	
	B#	JML
P1	B1	300
P1	B2	200
P1	B3	400
P1	B4	200
P1	B5	100
P1	B6	100
P2	B1	300
P2	B2	400
P3	B2	200
P4	B2	200
P4	B4	300
P4	B5	400

(Setelah dinormalisasi) Semua komponen dari domain terisi atom.



Suatu relation dikatakan sudah berada pada 1NF bila semua domainnya sudah berisi atom semua.

Contoh relation FIRST(P#,STATUS,KOTA,B#, JML), dimana STATUS atau KOTA tidak secara penuh tergantung kepada primary key, STATUS dan KOTA tidak mutually independent. Primary key di sini adalah P#, B#



FIRST

P#	STATUS	KOTA	B#	JML
P1	20	London	B1	300
P1	20	London	B2	200
P1	20	London	B3	400
P1	20	London	B4	200
P1	20	London	B5	100
P1	20	London	B6	100
P2	10	Paris	B1	300
P2	10	Paris	B2	400
P3	10	Paris	B2	200
P4	20	London	B2	200
P4	20	London	B4	300
P4	20	London	B5	400

Gambar 3. Functional Dependence dan contoh data di relation FIRST

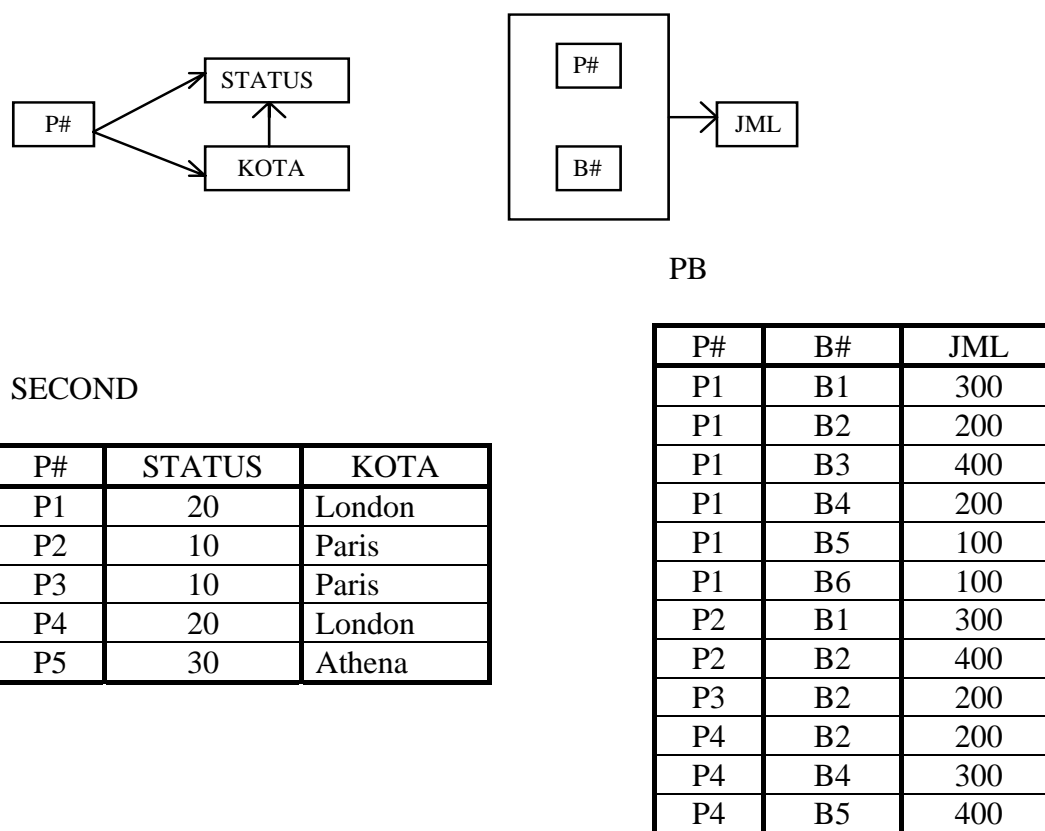
Di sini masih terdapat banyak kesulitan bila ingin melaksanakan operasi data misalnya :

- INSERT** : Kita tidak dapat memasukan data pemasok yang berdomisili di suatu kota sampai pemasok tersebut memasok paling tidak satu bahan. Di gambar 3 tidak terdapat P5 yang berdomisili di Athena, karena P# adalah merupakan primary key bersama B# dan belum pernah mengirim barang, maka tidak boleh muncul, sehingga tidak boleh B# kosong (null).
- DELETE** : Bila misalnya kita hapus salah satu tuple, kita tidak hanya kehilangan data tentang pengiriman barang, tetapi juga data bahwa suatu pemasok berdomisili di kota mana. Misalnya bila hapus tuple yang nilai P# = P3 dan B# = B2, maka kita akan kehilangan data bahwa P3 berdomisili di Paris



UPDATE : Suatu kota muncul berulang-ulang, sehingga apabila kita akan merubah bahwa S1 pindah dari London ke Amsterdam, maka akan mendapatkan persoalan untuk mencari semua S1 dan merubah London ke Amsterdam berikut statusnya yang berubah.

Pemecahan pertama adalah dengan memecah relation FIRST menjadi dua, yaitu SECOND(P#,STATUS,KOTA) dan PB(P#, B#, JML). Contohnya dapat dilihat pada gambar 4, yang menggambarkan functional dependence serta isi relation SECOND dan PB. disini sudah dapat ditambahkan pemasok P5 yang berdomisili di Athena di relation SECOND, tetapi tidak dapat di tambahkan di PB.



Gambar 4. Functional Dependence dan contoh data di relation SECOND dan PB

Dari gambaran pemecahan ini sudah akan mengurangi persoalan-persoalan operasi diatas, baik insert, delete maupun update. Dengan struktur ini kita sudah menghilangkan nonfull functional dependence. Sehingga dapat kita buat definisi dari tingkat normalisasi berikutnya.

### SECOND NORMAL FORM (2NF)

Suatu relation sudah berada pada 2NF, bila sudah berada pada 1NF dan setiap atribut yang bukan key fully functional dependence terhadap primary key.

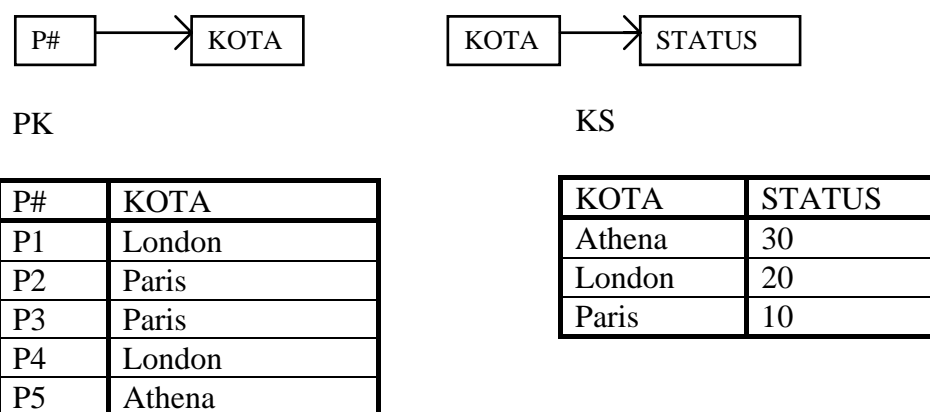


Reduksi relation FIRST ke SECOND dan PB adalah suatu contoh nonloss decomposition. Relation SECOND dan PB adalah hasil dari projection dari relation FIRST, dan sebaliknya FIRST adalah hasil natural join dari relation SECOND dan PB. Tetapi struktur SECOND di sini masih juga menimbulkan persoalan karena atribut-atribut yang bukan key belum mutual independence. Ketergantungan (dependency) STATUS terhadap P# adalah functional, tetapi masih tergantung juga terhadap kota (masih transitive dependence).

Bila setiap P# menentukan nilai KOTA dan STATUS, dan kemudian KOTA menentukan juga nilai STATUS Transitive dependence ini juga akan menimbulkan persoalan terhadap operasi update antara STATUS dan KOTA.

- INSERT : Kita tidak dapat memasukkan data bahwa suatu kota mempunyai nilai STATUS tertentu, misalnya kita tidak dapat menyatakan bahwa pemasok barang di ROMA harus mempunyai STATUS 50 sampai kita mempunyai pemasok yang berada di kota tersebut. Alasannya adalah sebelum pemasok tersebut muncul kita tidak punya nilai primary key.
- DELETE : Bila kita hanya menghapus tuple suatu kota tertentu, kita menghapus tidak data mengenai pemasok tersebut, tetapi juga kota yang mempunyai nilai status tersebut. Misalnya bila kita menghapus tuple yang berisi P5, kita akan kehilangan nilai status untuk kota Athena yang berisi 30.
- UPDATE : Nilai status suatu kota di relation SECOND muncul berkali-kali. Bila kita ingin merubah nilai status kota London dari 20 menjadi 30, maka kita akan diberikan persoalan baik untuk pencarian semua kota London dan juga mungkin salah penulisan.

Cara pemecahannya adalah dengan memecah relation SECOND menjadi dua dengan operasi projection, yaitu PK(P#,KOTA) dan KS(KOTA,STATUS), seperti dapat dilihat di gambar 5. Dalam gambar 5 terlihat bahwa kita dapat mengeliminasi transitive dependence dari STATUS terhadap P#. Sehingga dapat kita definisikan tingkat normalisasi berikutnya yaitu 3NF.



Gambar 5. Functional Dependence dan contoh data di relation PK dan KS



### THIRD NORMAL FORM (3NF)

Suatu relation sudah berada pada 3NF bila sudah berada dalam 2NF dan setiap atribut yang bukan key tidak dependent terhadap atribut lain kecuali terhadap primary key (nontransitively dependent terhadap primary key).

Dalam relation PK, Atribut P# sebagai primary key, dan di dalam relation KS atribut KOTA sebagai primary key.

### BOYCE/CODD NORMAL FORM (BCNF)

Definisi 3NF diperkuat lagi dan disederhanakan dengan definisi dari BCNF, yaitu tidak menunjuk ke 1NF, 2NF, maupun konsep full dan transitive dependence. Dalam definisi ini kita menggunakan istilah (functional) determinant sebagai atribut dimana semua atribut lainnya fully functionally dependent terhadapnya.

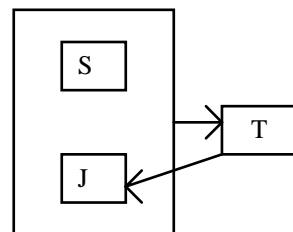
Sehingga definisi dari suatu relation sudah berada pada BCNF, bila setiap determinant adalah merupakan candidate key.

Suatu relation yang sudah pada 3NF belum tentu pada BCNF, tetapi relation yang sudah pada BCNF pasti pada 3NF.

Contoh : Dalam gambar 6 relation SJT berada pada 3NF tetapi tidak pada BCNF, karena terjadi overlap candidate key (S, J) dan (S, T)

SJT

S	J	T
Smith	Math	Prof. White
Smith	Physics	Prof. Green
Jones	Math	Prof. White
Jones	Physics	Prof. Brown



Gambar 6. Functional Dependence dan contoh data di relation SJT

### FOURTH NORMAL FORM (4NF)

Dalam normalisasi ini kita mengenal adanya tipe baru dari ketergantungan (dependency) yaitu Multivalued Dependency (MVD), yang merupakan generalisasi dari functional dependence. Atau sebaliknya, FD adalah peristiwa khusus dari MVD di mana himpunan nilai dependent berisi satu nilai saja. Sebagai contoh relation di gambar 7 ini.



MDT

MTKUL	DOSEN	TEXT
Physics	Prof. Green	Basic Mechanics
Physics	Prof. Green	Principles of Optics
Physics	Prof. Brown	Basic Mechanics
Physics	Prof. Brown	Principles of Optics
Physics	Prof. Black	Basic Mechanics
Physics	Prof. Black	Principles of Optics
Math	Prof. White	Modern Algebra
Math	Prof. White	Projective Geometry

Gambar 7. Relation MDT

Mata kuliah physics diberikan oleh tiga dosen yaitu Prof. Green, Prof. Brown, Prof. White, sedangkan mata kuliah Math diberikan oleh satu dosen yaitu Prof. White. Untuk mata kuliah Physics mempergunakan dua text yaitu Basic Mechanics dan principles of Optics sedangkan mata kuliah Math mempergunakan dua text yaitu Modern Algebra dan Projective Geometry.

Relation ini berada pada BCNF karena tidak ada functional determinants lain kecuali MTKUL. Tetapi kita lihat banyak duplikasi data, sehingga struktur ini dapat kita pecah menjadi dua relation MD dan MT seperti pada gambar 8.

MD

MTKUL	DOSEN
Physics	Prof. Green
Physics	Prof. Brown
Physics	Prof. Black
Math	Prof. White

MT

MTKUL	TEXT
Physics	Basic Mechanics
Physics	Principles of Optics
Math	Modern Algebra
Math	Projective Geometry

Gambar 8. Relation MD dan MT

Terdapat dua MVD di relation MDT yaitu :

$$\begin{aligned} \text{MDT.MTKUL} &\twoheadrightarrow \text{MDT.DOSEN} \\ \text{MDT.MTKUL} &\twoheadrightarrow \text{MDT.TEXT} \end{aligned}$$

Artinya atribut MDT.DOSEN multydependent terhadap atribut MDT.MTKUL atau atribut MDT.MTKUL multidetermines atribut MDT.DOSEN. Begitu juga untuk yang kedua. Sehingga definisi dari MVD adalah :

Bila dalam suatu relation R terdapat atribut A, B, dan C multi ..... dependence  $R.A \twoheadrightarrow R.B$  terjadi di R, bila himpunan dari nilai B match (cocok) dengan pasangan nilai A dan C di R, hanya tergantung pada nilai A dan bebas terhadap nilai C.

$$R.A \twoheadrightarrow R.B \mid R.C$$



Contoh :

$MDT.MTKUL \twoheadrightarrow \twoheadrightarrow MDT.DOSEN \mid MDT.TEXT$

MVD terjadi hanya bila dalam relation mempunyai paling sedikit tiga atribut.

Definisi 4NF :

Suatu relation berada pada 4NF bila terjadi MVD di R yaitu  $A \twoheadrightarrow \twoheadrightarrow B$ . kemudian semua atribut di R juga functionally dependent pada A.

4NF lebih baik dari BCNF yaitu setiap relation 4NF berada pada BCNF. Semua relation dapat nonloss decomposed ke dalam kumpulan 4NF relation.

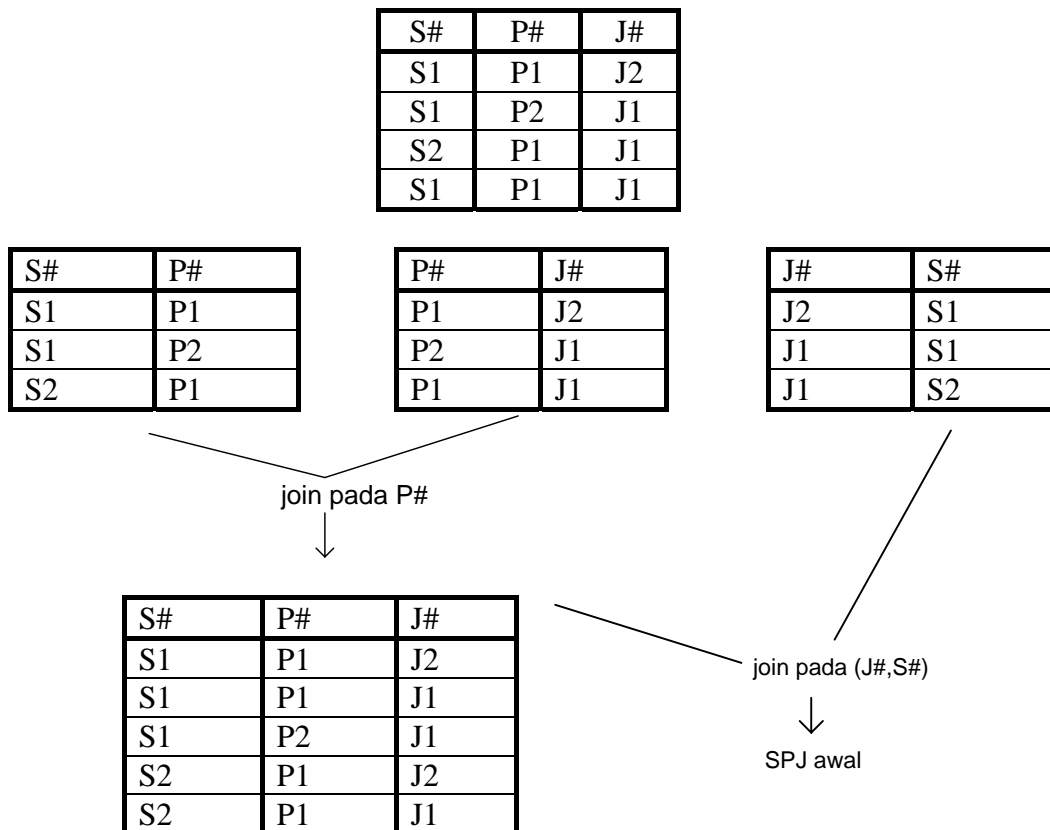
**FIFTH NORMAL FORM (5NF) / PROJECT-JOIN NORMAL FORM (PJNF)**

Suatu relation dapat terus dilakukan proses nonloss-decomposition sampai semua projection berada pada 5NF. Tetapi hal ini juga ada keterbatasan yaitu join dependency (JD), yaitu bila projection dapat dilakukan sehingga tidak ada nilai yang hilang, apabila di join kembali. JD adalah generalisasi dari MVD.

Suatu relation R berada pada 5NF bila setiap join dependency di R dibuat dengan candidate key di R.

Contoh :

SPJ



Gambar 9. Relation SPJ dengan operasi join dari projection-projectionnya





Beberapa persoalan update di SPJ adalah seperti contoh pada gambar 10

SPJ

S#	P#	J#
S1	P1	J2
S1	P2	J1

Bila (S2,P1,J1) diinsert (S1,P1,J1)  
harus juga diinsert.

Tetapi tidak sebaliknya

SPJ

S#	P#	J#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

Dapat menghapus (S2,P1,J1)  
tanpa ada pengaruh sampingan.  
Bila (S1,P1,J1) dihapus, harus  
ada tuple lain yang dihapus.

Gambar 10. Problem pada proses update di relation SPJ

## KESIMPULAN

Tahap-tahap proses reduksi pada normalisasi, yaitu :

1. Ambil projection pada relation 1NF awal untuk mengeliminasi semua nonfull functional dependence, sehingga menghasilkan relation 2NF.
2. Ambil projection pada relation 2NF untuk mengeliminasi semua transitive dependence, sehingga menghasilkan kumpulan relation 3NF.
3. Ambil projection pada relation 3NF untuk mengeliminasi semua sisa functional dependence di mana determinannya bukan candidate key, sehingga menghasilkan relation BCNF.

Catatan : Tahap 1 sampai dengan 3 dapat dipersingkat sebagai berikut :

- Ambil projection dari relation awal untuk mengeliminasi semua FD di mana determinannya bukan candidate key.
4. Ambil projection pada relation BCNF untuk mengeliminasi MVD yang bukan FD juga, sehingga menghasilkan kumpulan relation 4NF.
5. Ambil projection pada relation 4NF untuk mengeliminasi semua JD yang bukan implementasi dari candidate key.

Seorang ahli (Fagin) menambahkan suatu tingkat normalisasi yaitu (3.3)NF sebagai peningkatan dari 3NF, serupa dengan BCNF, bukan merupakan 4NF. Selain itu juga menambahkan satu tingkat normalisasi lagi yaitu DK/NF (Domain Key Normal Form) yang tidak menyinggung sama sekali tentang FD, MVD maupun JD. DK/NF terjadi bila setiap relation yang terjadi memenuhi batasan-batasan (constraints) tertentu yaitu sebagai akibat dari batasan key (key constraints) dan batasan domain (domain constraints).

Key constraints adalah ketentuan/batasan tentang atribut atau kombinasinya yang merupakan candidate key.



Domain Constraints adalah ketentuan/batasan dimana nilai atribut tertentu berada pada himpunan nilai yang sudah ditentukan batas-batasnya. Dinyatakan juga bahwa DK/NF juga otomatis 5NF, sehingga juga 4NF dan (3.3)NF. DK/NF tidak harus selalu tercapai dan tidak juga terjawabnya pertanyaan secara pasti kapan hal tersebut dapat dicapai.