

11. NORMALISASI

Normalisasi merupakan teknik analisis data yang mengorganisasikan atau mengelompokkan atribut-atribut data sehingga terbentuk entitas yang *non-redundant*, stabil, dan fleksibel.

Normalisasi dilakukan sebagai uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi itu sudah baik, yaitu dapat dilakukan proses insert, update, delete, dan modifikasi pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut.

Dalam perancangan basis data, normalisasi berperan sebagai :

- Kerangka kerja formal untuk menganalisa skema relasi yang didasarkan pada primary keys dan *functional dependencies* antara atribut-atribut.
- Satu urutan test yang dapat dilakukan pada masing-masing skema relasi, sehingga basis data relasional dapat dinormalisasi ke suatu tingkat tertentu. Bilamana suatu test gagal, maka relasi yang menyalahi test harus didekomposisi menjadi sejumlah relasi yang masing-masing memenuhi kaidah normalisasi.

11.1 Bentuk Normal

Ada enam tingkatan dalam normalisasi yang masing-masing menghasilkan bentuk normal sebagai berikut :

- 1 NF
 - 2 NF
 - 3 NF
 - BCNF
- } di definisikan hanya dengan memperhatikan *functional dependencies* dan *key constrains*
- 4 NF → + constraint multivalued dependency
 - 5 NF → + constraint join dependency

Skema relasi dikatakan menjadi sebuah bentuk normal tertentu bila ia memenuhi beberapa kondisi tertentu. Misalnya, sebuah skema relasi dikatakan berada dalam bentuk normal kedua (2NF) jika dan hanya jika ia berada dalam 1NF dan juga memenuhi kondisi lain yang disyarkan untuk menjadi 2NF.

11.2 First Normal Form (1NF)

Secara histories, 1NF didefinisikan untuk tidak membolehkan adanya atribut yang berulang (*multivalued attribute*) atau yang bernilai ganda (*composite attribute*) atau kombinasi keduanya. Dalam 1NF disebutkan bahwa domains dari attribute harus hanya terdiri dari nilai yang berdiri sendiri (*atomic values*), dan nilai dari sembarang attribute dalam suatu tuple harus berupa nilai tunggal (*single values*) dari domain atribut tersebut.

Syarat suatu skema relasi bisa berada dalam bentuk 1NF jika :

1. Tidak ada set atribut yang berulang atau bernilai ganda.
2. Telah ditentukannya primary key untuk tabel atau relasi.
3. Tiap atribut hanya memiliki satu pengertian.

Jika ada atribut yang memiliki banyak nilai maka harus digambarkan dalam sebuah entitas atau relasi yang terpisah.

Contoh 1NF :

EMPPROJ (SSN, ENama, { PROJ (PNumber, Hours) })

↓ 1NF

EMP_PROJ1 (SSN, EName)

EMP_PROJ2 (SSN, PNumber, Hours)

11.3 Second Normal Form (2NF)

2NF didefinisikan pada skema relasi R jika setiap non_prime atributnya secara fungsional bergantung penuh pada key dari R.

2NF didasarkan pada konsep “*full function dependency*”. FD $X \rightarrow Y$ disebut *full function dependency*, jika penghapusan sembarang atribut A dari X menyebabkan sifat dependency tidak berlaku lagi; yaitu untuk sembarang atribut $A \in X$, maka $(X - \{A\}) \not\rightarrow Y$. Sebaliknya, FD $X \rightarrow Y$ disebut *partial dependency* jika beberapa atribut $A \in X$ dapat dihapus dari X dan tetap mempertahankan dependency yang ada; yaitu untuk $A \in X$, $(X - \{A\}) \rightarrow Y$.

Contoh :

{ SSN, PNumber } \rightarrow Hours

Merupakan full dependency, karena SSN $\not\rightarrow$ Hours

PNumber $\not\rightarrow$ Hours

{ SSN, PNumber } \rightarrow EName

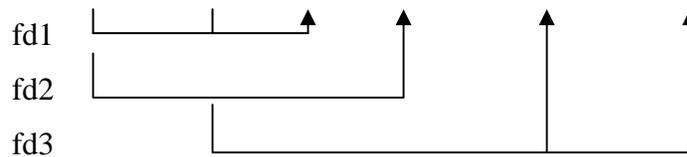
Merupakan partial dependency, karena SSN \rightarrow EName

Syarat suatu skema relasi bisa berada dalam bentuk 2NF jika :

- 1.. Bentuk data telah memenuhi kriteria bentuk normal ke satu.
2. Atribut bukan kunci (*non-key attribute*) haruslah memiliki ketergantungan fungsional sepenuhnya pada primary key

Contoh 2NF :

EMPPROJ (SSN, PNumber, Hours, EName, PName, PLocation)



Normalisasi 2NF

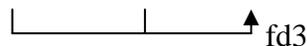
EP1 (SSN, PNumber, Hours)



EP2 (SSN, EName)



EP3 (PNumber, PName, PLocation)



Kesimpulan :

Jika suatu skema relasi tidak berada dalam 2NF, ia dapat dinormalisasi lanjut menjadi sejumlah relasi 2NF dengan cara mengasosiasikan non-key atribut yang ada hanya dengan sebagian dari primary key dimana atribut-atribut tersebut secara fungsional bergantung penuh.

11.4 Third Normal Form (3NF)

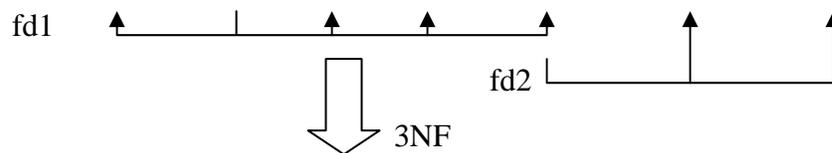
3NF didasarkan pada konsep “*transitive dependency*”. Dimana suatu FD $X \rightarrow Y$ dalam suatu relasi R merupakan *transitive dependency* jika terdapat satu set atribut Z yang tidak merupakan subset dari sembarang key dari R, dan berlaku $X \rightarrow Z$ dan $Z \rightarrow Y$.

Syarat suatu skema relasi bisa berada dalam bentuk 3NF jika :

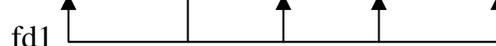
- 1.. Bentuk data telah memenuhi kriteria bentuk normal ke dua (2NF).
2. Atribut bukan kunci (*non-key attribute*) tidak boleh memiliki ketergantungan secara transitif pada primary key.

Contoh 3NF :

EMP_DEPT (EName, SSN, BDate, Address, DNumber, DName, DMGRSSN)



ED1 (EName, SSN, BDate, Address, DNumber)



ED2 (DNumber, DName, DMGRSSN)



➔ Operasi Natural JOIN terhadap ED1 dan ED2 akan menghasilkan relasi EMP_DEPT semula (tanpa ada spurious tuples)

Definisi Umum 3NF :

Suatu skema relasi R ada dalam 3NF bilamana berlaku suatu FD $X \rightarrow A$ dalam R, dan memenuhi kondisi :

- (a) X adalah superkey dari R, atau
- (b) A adalah prime atribut dari R

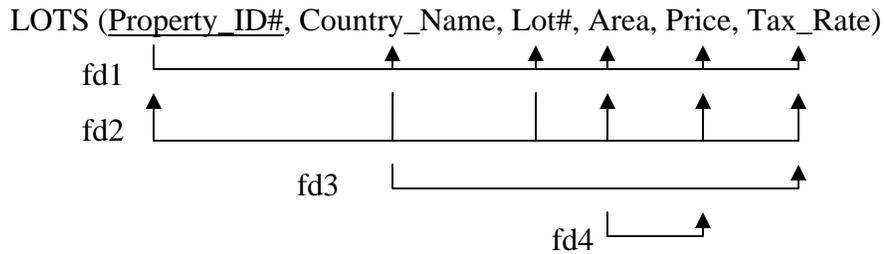
Implikasinya :

- Suatu skema relasi R menyalahi definisi umum 3 NF, jika berlaku FD $X \rightarrow A$ dalam R yang menyalahi baik kondisi (a) ataupun (b).
- Suatu skema relasi R berada dakam 3NF, jika setiap non-prime atribut dari R :
 - Secara fungsional bergantung penuh pada setiap key dari R, dan
 - Secara non-transitive bergantung pada setiap key dari R.

Definisi umum dari 3NF dapat diaplikasikan secara langsung untuk memeriksa apakah suatu skema relasi berada dalam 3NF (tanpa perlu melalui pemeriksaan 2NF terlebih dahulu).

Contoh lain penurunan relasi-relasi 2NF dan 3NF :

Pada basis data “Parcel of land for sale” dalam berbagai negara bagian di USA, terdapat relasi sebagai berikut :

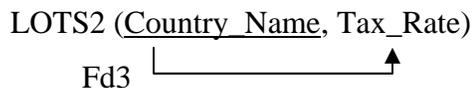
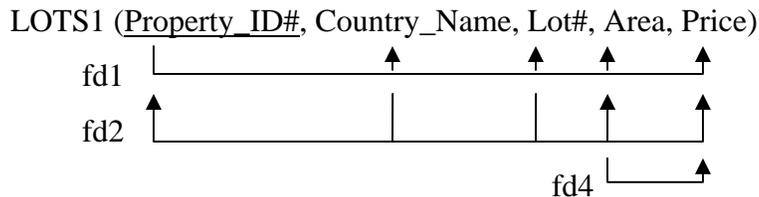


Ada dua candidate keys, yaitu :

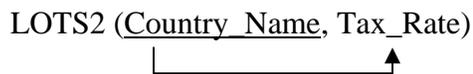
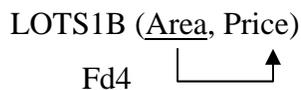
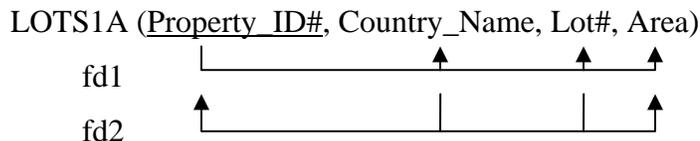
- Property_ID#
- { Country_Name, Lot# }

Penurunan melalui normalisasi 2NF :

Pada contoh diatas, fd3 menyalahi aturan normalisasi 2NF, sehingga menjadi :

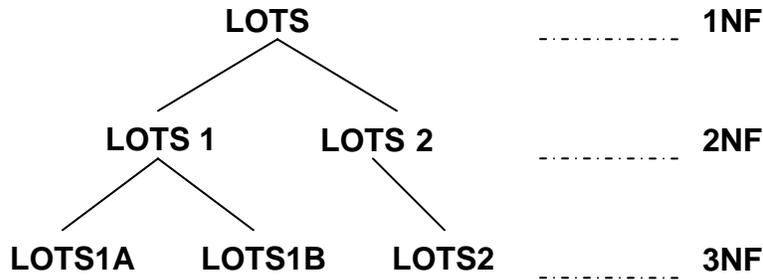


Pada contoh diatas, fd4 masih menyalahi aturan normalisasi 3NF, sehingga menjadi :



Fd3

Penurunan melalui normalisasi 2NF diatas dapat digambarkan sebagai berikut :



Penurunan tanpa melalui normalisasi 2NF :

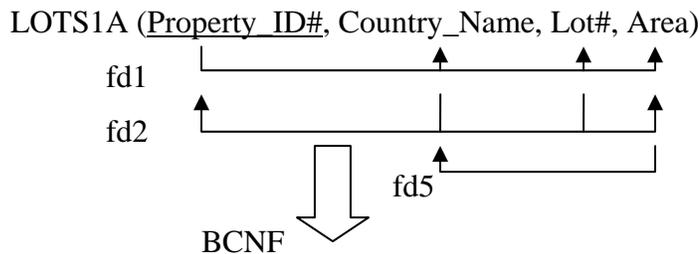
Dari keempat FD yang ada (fd1 s/d fd 4) dalam skema relasi LOTS, terlihat bahwa fd3 dan fd4 menyalahi 3NF, sehingga dapat langsung didekomposisi menjadi LOTS1A, LOTS1B, dan LOTS2.

11.5 Boyce/Codd Normal Form (BCNF)

BCNF merupakan bentuk normal yang lebih ketat dibandingkan 3NF. Dan ini berarti bahwa setiap relasi dalam BCNF juga berada dalam 3NF, tetapi tidak sebaliknya.

Suatu skema relasi berada dalam BCNF bilamana berlaku suatu FD $X \rightarrow A$ dalam R, maka X merupakan superkey dari R. Jadi perbedaan antara BCNF dan 3NF adalah kondisi (b) dari 3NF yang membolehkan A untuk prime jika X bukan superkey **TIDAK BOLEH** ada dalam BCNF.

Jika dalam relasi LOTS1A berlaku FD tambahan (fd5) :



LOTS1AX (Property_ID#, Area, Lot#)

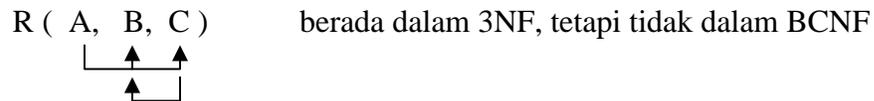
LOTS1AY (Area, Country_Name)

→ fd2 hilang dari hasil dekomposisi BCNF

Kesimpulan Akhir :

Dalam praktik, kebanyakan skema-skema relasi yang berada dalam 3NF juga berada dalam BCNF. Hanya jika ada dependency $X \rightarrow A$ dalam skema relasi R dengan X bukan superkey dan A adalah prime atribut, akan menjadikan R berada dalam 3NF tetapi tidak berada dalam BCNF

Kasus umum :



Jadi, dalam proses normalisasi : upayakan untuk membentuk BCNF, dan jika tidak memungkinkan baru biarkan berada dalam 3NF.

11.6 Multivalued Dependencies Dan Fourth Normal Form (4NF)

Multivalued dependencies timbul sebagai konsekuensi dari bentuk normal pertama yang tidak membolehkan sebuah atribut dalam suatu tuple untuk memiliki satu set nilai (*multivalue*).

Bilamana dua atau lebih atribut-atribut multivalued yang satu sama lain independent terdapat dalam satu skema yang sama, maka akan muncul persoalan “*repeating group*” dari setiap nilai dari salah satu atribut dengan setiap nilai atribut yang lain untuk menjaga konsistensi dari relation instances. Constraint ini disebut multivalued dependency

Contoh :

Relasi EMP :

EName	PName	DName
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

Multivalued : pasangan { EName, PName } dan { EName, DName }

Definisi Formal Multivalued Dependency

Secara formal, suatu multivalued dependency (MVD) $X \twoheadrightarrow Y$ yang berlaku dalam skema relasi R, dimana X dan Y keduanya merupakan subset dari R, menjelaskan constraint berikut pada setiap sembarang relasi r dari R :

Jika dua tuple t_1 dan t_2 ada dalam r, $t_1[x] = t_2[x]$, maka sembarang dua tuple t_3 dan t_4 harus juga ada dalam r yang memenuhi sifat-sifat :

- $t_3[x] = t_4[x] = t_1[x] = t_2[x]$
- $t_3[Y] = t_1[Y]$ dan $t_4[Y] = t_2[Y]$
- $t_3[R - (XY)] = t_2[R - (XY)]$ dan $t_4[R - (XY)] = t_1[R - (XY)]$

Suatu MVD dalam R disebut **TRIVIAL** jika :

- (a) Y merupakan subset dari X, atau
- (b) $X \cup Y = R$

MVD selain TRIVIAL disebut **NON-TRIVIAL**. Contoh Non-Trivial adalah relasi EMP diatas, dimana hasil dekomposisi menjadi 2 relasi berikut disebut sebagai MVD Trivial :

EMP_PROJ (EName, PName)

EMP_DEP (EName, DName)

Definisi 4NF

Suatu skema relasi R berada dalam 4NF berdasarkan pada satu set dependency F jika, untuk setiap nontrivial multivalued dependency $X \twoheadrightarrow Y$ dalam F^+ , x adalah superkey dari R.

Contoh :

Relasi EMP (EName, PName, DName) diatas tidak berada dalam 4NF, karena terdapat nontrivial MVD :

EName \twoheadrightarrow PName

EName \twoheadrightarrow DName

dan EName bukan superkey

Tetapi relasi hasil dekomposisi :

EMP_PROJ (EName, PName)

EMP_DEP (EName, DName)

Keduanya berada dalam 4NF karena :

EName $\rightarrow\rightarrow$ PName dalam EMP_PROJ

EName $\rightarrow\rightarrow$ DName dalam EMP_DEP

Keduanya merupakan trivial MVD