

Pertemuan 4

Normalisasi dalam Database

Tujuan Intruksional :

Pokok Bahasan ini mempelajari tentang normalisasi pada model database

Kompetensi Yang Diharapkan :

Mahasiswa diharapkan memahami tentang normalisasi pada model database termasuk aturan *First Normal Form*, *Second Normal Form*, dan *Third Normal Form* dan dapat mengimplementasikan bentuk normal diatas pada permasalahan yang ada.

Waktu Pertemuan : 100 Menit

4.1 Pengertian Normalisasi

Normalisasi merupakan sebuah proses mengorganisir data dalam suatu database untuk menghindari redundansi data, anomali pada penyisipan (insertion), anomali pada perubahan (update), dan anomali pada penghapusan (deletion).

4.2 Anomali pada DBMS

Ada tiga jenis anomali yang terjadi ketika database tidak dinormalisasi. Anomaly yang terjadi adalah - *Insertion*, *update* dan *deletion anomaly*. Mari kita ambil contoh untuk memahami hal ini.

Misalkan sebuah perusahaan perabotan menyimpan rincian pegawai dalam tabel bernama pegawai yang memiliki empat atribut: **id_pegawai** untuk menyimpan id pegawai, **nama** untuk menyimpan nama pegawai, **alamat** untuk menyimpan alamat pegawai dan divisi untuk menyimpan rincian divisi di mana pegawai bekerja. Pada suatu waktu table pegawai terlihat seperti ini:

id_pegawai	nama	alamat	divisi
101	Agus	Bandar Lampung	D001
101	Agus	Bandar Lampung	D003

123	Budi	Pringsewu	D890
166	Susi	Pesawaran	D900
166	Susi	Pesawaran	D004

Tabel diatas tidaklah normal. Kita akan lihat masalah yang didapat pada tabel yang tidak normal.

Update Anomaly: Dalam tabel di atas kita memiliki dua baris untuk pegawai bernama Agus karena ia bekerja pada 2 divisi. Jika kita ingin memperbaharui (*update*) alamat dari Agus kita harus memperbaharui 2 baris data karena jika tidak, hal tersebut tidak konsisten. Jika entah bagaimana, alamat Agus yang benar berada pada 1 baris namu tidak pada baris yang lain, Agus akan mempunyai 2 alamat berbeda dan bisa menyebabkan data yang tidak konsisten.

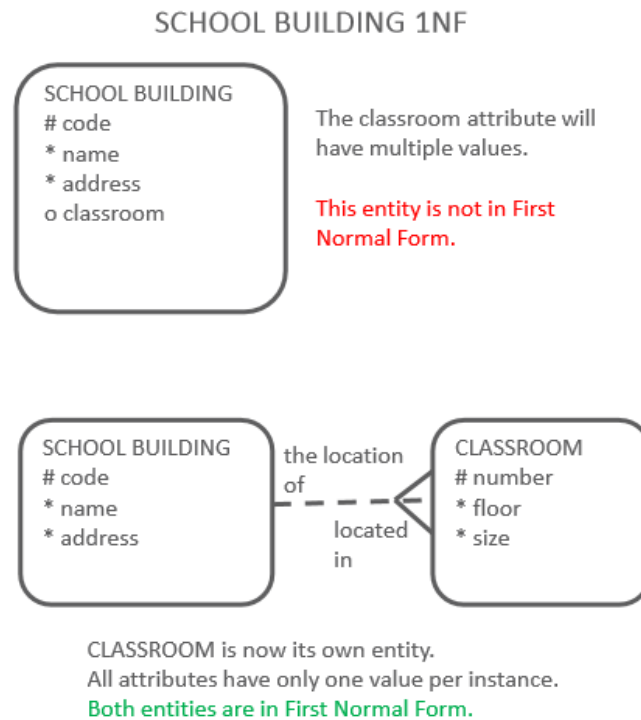
Insert Anomaly: Jika ada pegawai baru yang bergabung, namun masih dalam masa *training* dan belum terdaftar pada divisi manapun, maka data tidak akan bisa dimasukkan (*insert*) ke table jika atribut divisi tidak memperbolehkan nilai null.

Delete Anomaly: Jika pada suatu waktu perusahaan akan menutup divisi D890 lalu menghapus baris yang memiliki nilai pada atribut divisi yaitu D890 maka informasi mengenai pegawai Budi akan terhapus karena hanya ia terdaftar sebagai pegawai yang bekerja pada divisi D890.

4.3 Bentuk Normal

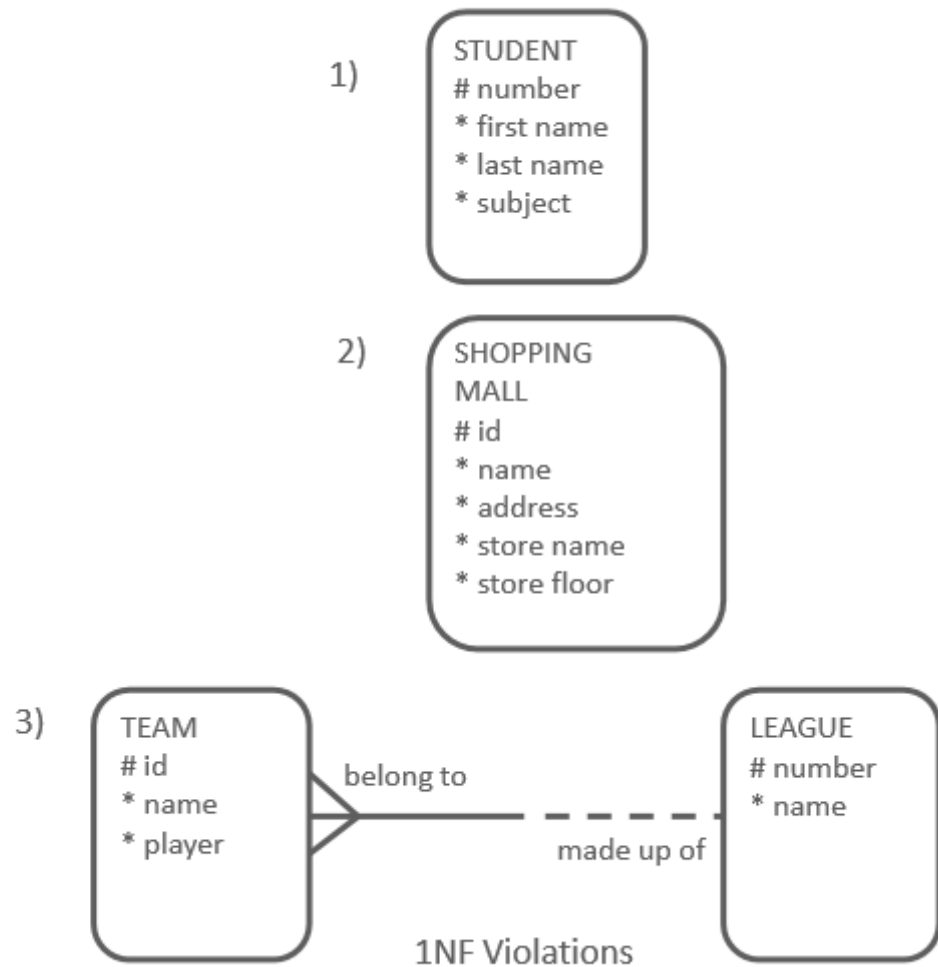
Berikut adalah bentuk-bentuk normal yang paling umum digunakan:

1. First Normal Form (1NF)



- Setiap data dibentuk dalam flat file, data dibentuk dalam satu record demi satu record nilai dari field berupa “atomic value”.
- Untuk memeriksa 1NF, setiap atribut harus memiliki nilai tunggal untuk setiap instance dari sebuah entitas. (Tidak ada set atribut yang berulang atau bernilai ganda).
- Satu kode, satu nama, dan satu alamat ada untuk gedung sekolah, tetapi tidak satu kelas. Satu sekolah akan memiliki banyak kelas, oleh karena itu banyak data yang akan mengalami redundansi.
- Karena banyak ruang kelas yang ada di gedung sekolah, ruang kelas adalah *multi-valued* dan melanggar 1NF.
- Jika ada atribut *multi-valued*, buatlah suatu entitas tambahan dan hubungkan dengan entitas asli dengan jenis relasi 1: M (*one to many*).

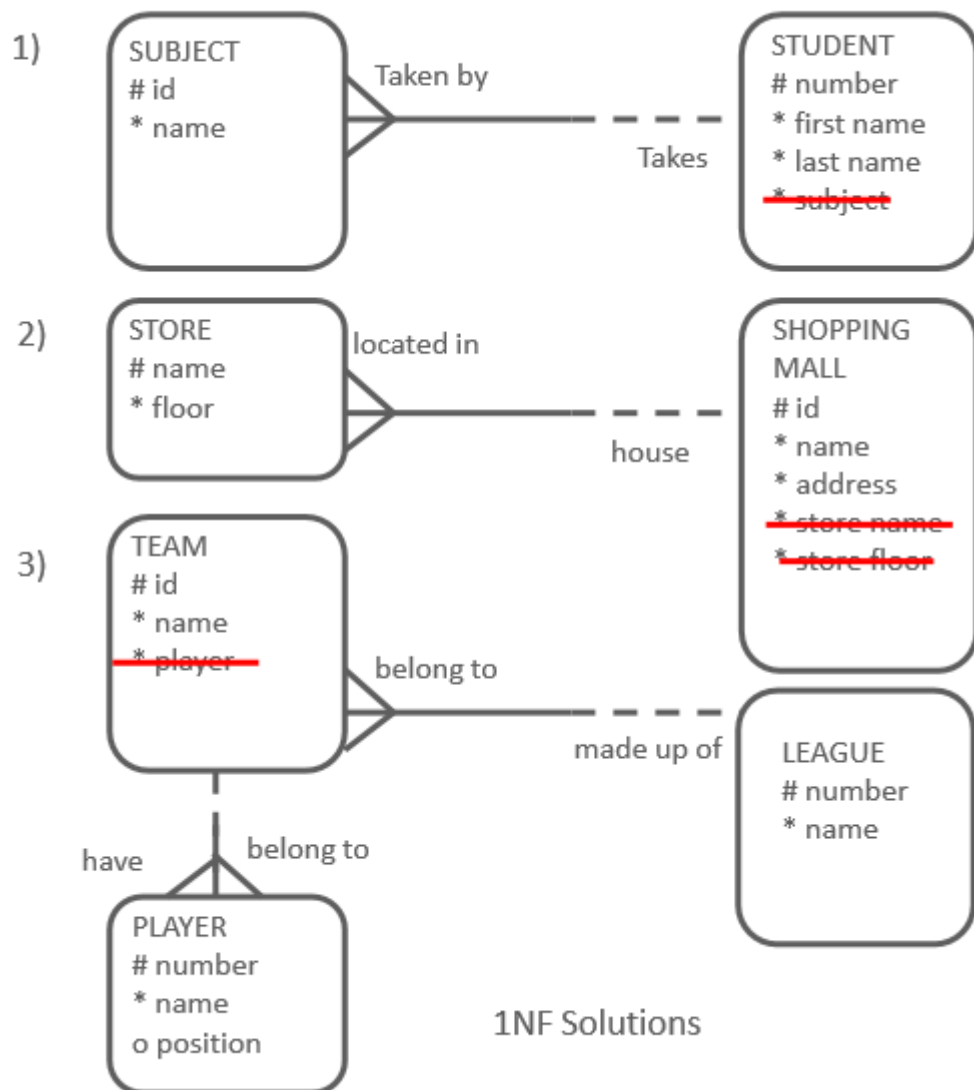
Pelanggaran Bentuk Normal Pertama (First Normal Form)



- Coba perhatikan entitas-entitas diatas.
- Apakah ada atribut yang bersifat *multi-valued*?

Solusi Bentuk Normal Pertama (First Normal Form)

“Ketika semua atribut dalam suatu entitas yang bernilai tunggal, entitas yang dikatakan dalam Bentuk Normal Pertama.”



2. Second Normal Form (2NF)

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.

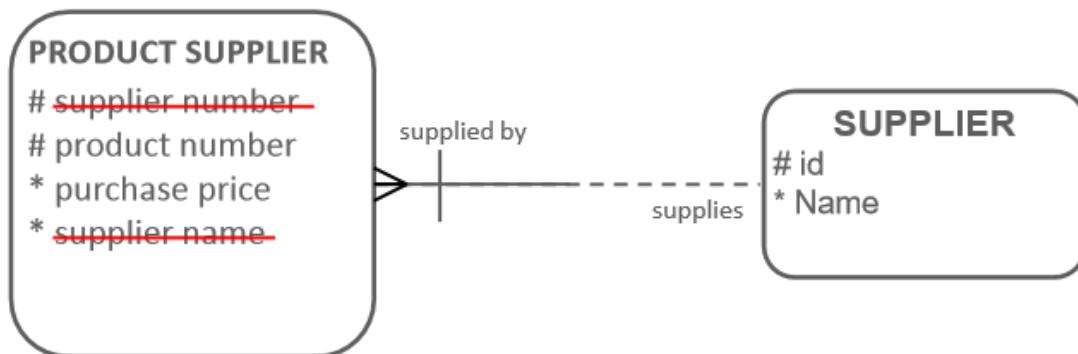
- Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
- Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.
- Tidak ada atribut dalam tabel yang secara fungsional bergantung pada candidate key tabel tersebut.



Dari entitas diatas

1. Apakah purchase price milik dari supplier number, product number, atau keduanya?
2. Apakah supplier name milik dari supplier number, product number, atau keduanya?

Solusi Bentuk Normal Kedua (Second Normal Form)



Bahwasanya entitas sebelumnya merupakan pelanggaran dalam 2NF, jika menggunakan entitas tersebut maka terjadinya data rangkap. maka perlu dibuat entitas baru untuk mendefinisikan Supplier dan Product Supplier.

3. Third Normal For (3NF)

Normalisasi database dalam bentuk 3NF bertujuan untuk menghilangkan seluruh atribut atau field yang tidak berhubungan dengan primary key. Dengan demikian tidak ada ketergantungan transitif pada setiap kandidat key. Syarat dari bentuk normal ketiga atau 3NF adalah :

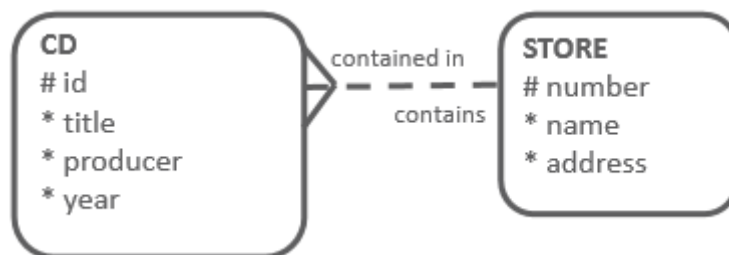
1. Memenuhi semua persyaratan dari bentuk normal kedua.
2. Menghapus kolom yang tidak tergantung pada primary key.



Perhatikan entitas diatas.

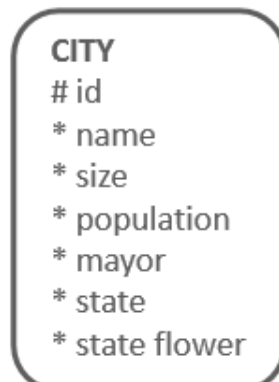
1. Apakah store name berhubungan dengan UID dari CD ?
2. Apakah store address berhubungan dengan UID dari CD ?

Tidak, dan ini merupakan pelanggaran pada 3NF.



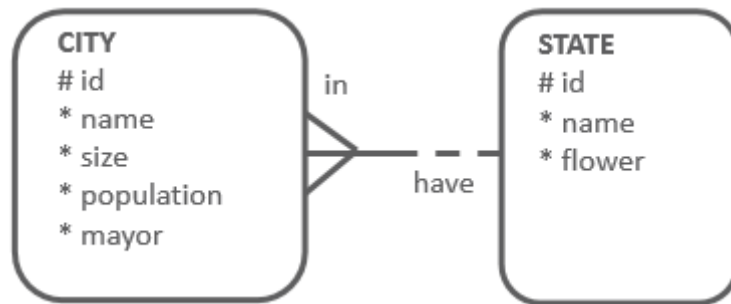
Model 3NF menunjukkan bahwasanya, perlunya dibuat entitas baru dengan nama STORE yang terdiri dari number, name, dan address yang terhubung dengan entitas CD.

Contoh Aturan 2



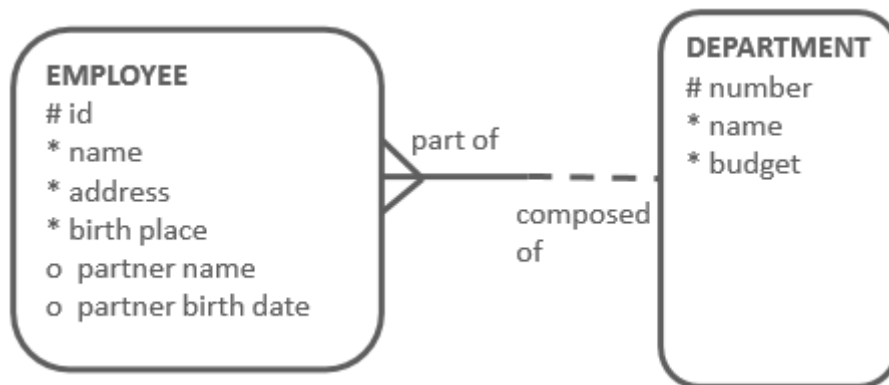
Entitas diatas merupakan entitas CITY yang didalamnya, Juga terdapat atribut state. Dalam 3NF itu merupakan pelanggaran, karena pada entitas diatas termasuk dalam

Transitive Dependency -> Ketergantungan Transitif



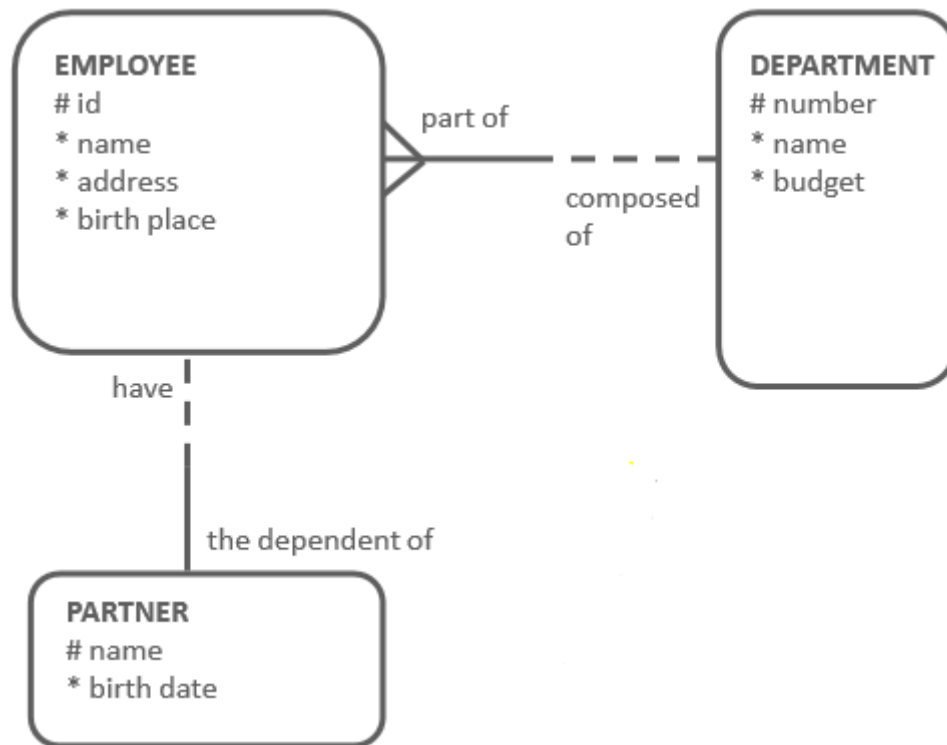
Model 3NF menunjukkan bahwasanya, perlunya dibuat entitas baru dengan nama STATE yang terdiri dari id, name, dan flower yang terhubung dengan entitas CITY.

Contoh 3



- Pada entitas diatas, diasumsikan bahwasanya setiap employee memiliki *partner*.
- Mengingat bahwasanya pada 3NF tidak diperbolehkan adanya **Transitive Dependency -> Ketergantungan Transitif**.
- Pada entitas diatas apakah partner name dan birth date berhubungan dengan UID dari EMPLOYEE ?

Tentu tidak, maka diperlukan entitas baru yang mendefinisikan partner yang terhubung dengan Employee.



Latihan :

ID Buku	Judul_Buku	TglTerbit	ID_Penerbit	Nama_Penerbit	Alamat_Penerbit	IDPeminjaman	NamaPeminjaman	Alamat_Peminjaman
B12	Sistemku	20-jan-12	P102	Gramed	Teluk	PJ65	Ateng	Jakarta
A67	Bukuku	12-feb-17	P454	PencetPin	Karang	PJ77	Deri	Palembang
H63	ORACLE	8-des-10	P54	ChuMart	Metro	PJ34	Iman	Medan
J53	MySQL	9-feb-13	P87	Indomei	Rajabasa	PJ2	Saddik	Surabaya

Buatlah *Conseptual Model* yang sudah di-normalisasi berdasarkan tabel diatas!