

BAB III

MODEL DATA RELASIONAL

- Model data relasional diperkenalkan oleh “Codd” pada tahun 1970.
- Didasarkan pada suatu struktur data yang sederhana dan seragam (uniform), yaitu :
”Relasi” – yang sudah mempunyai dasar teori yang kuat.

3.1. KONSEP-KONSEP MODEL RELASIONAL

- Model relasional menyajikan database sebagai satu kumpulan relasi.
- Secara informal, setiap relasi dapat disamakan dengan sebuah tabel.
- Jika suatu relasi dipandang sebagai sebuah tabel dari sejumlah nilai, maka :
 - Setiap baris dalam tabel menyajikan sekumpulan nilai-nilai data yang berhubungan satu dengan yang lain.
 - ➔ Nilai-nilai ini dapat diinterpretasikan sebagai fakta-fakta yang menjelaskan ”entity” atau ”relationship”.
 - Nama tabel dan nama-nama kolom digunakan untuk membantu kita dalam melakukan interpretasi arti dari nilai-nilai dalam setiap baris dari tabel.
- Dalam terminologi model relasional :
 - ❖ Sebuah baris disebut sebuah tuple.
 - ❖ Suatu header dari kolom disebut attribute.
 - ❖ Tabel itu sendiri disebut relasi.
 - ❖ Tipe data yang menjelaskan jenis-jenis nilai yang dapat muncul dalam setiap kolom disebut domain.

3.1.1. Domain, Tuple, Attribute dan Relasi

- Suatu ”domain” D merupakan satu set dari nilai-nilai yang atomic.
 - ➔ Atomic mempunyai pengertian bahwa setiap nilai dalam domain harus utuh (indivisible).
 - ➔ Satu cara yang umum dalam menjelaskan suatu domain adalah dengan cara menjelaskan tipe data dimana nilai-nilai data yang ada membentuk domain yang dimaksud.

Contoh : domain “Nrp_mahasiswa” : satu set dari 10 digit yang valid untuk menyatakan nrp.mahasiswa.

→ Selain tipe data, suatu domain juga dijelaskan dengan memberikan : nama dan format data; plus informasi tambahan untuk memudahkan menginterpretasikan nilai-nilai domain.

- Suatu skema relasi R, yang dituliskan sebagai :

$$R (A1, A2, \dots, An)$$

terbentuk dari satu nama relasi R dan satu List attribute : A1, A2,, An.

→ Setiap attribute Ai merupakan nama dari peran yang dimainkan oleh beberapa domain D dalam skema relasi R.

→ Domain D dari suatu attribute Ai dinotasikan sebagai dom (Ai).

→ Derajat dari relasi sama dengan jumlah attribute dari skema relasi.

Contoh : skema relasi dengan derajat 4 :

MAHASISWA (Nama, Nrp, Tgl_lahir, Alamat)

➤ dom (Nama) = Satu set nama-nama mahasiswa dengan panjang karakter 30.

➤ dom (Nrp) = Satu set nrp mahasiswa berupa 10-digit nomor pokok mahasiswa yang valid.

➤ dom (Tgl_lahir) = Satu set tgl kelahiran mahasiswa dengan format : dd-mm-yyyy.

➤ dom (Alamat) = Satu set alamat mahasiswa berupa string alpha - numerik dengan panjang maks. 40.

- Satu nilai relasi (relation instance) r dari suatu skema relasi R (A1, A2,, An)

yang dituliskan sebagai : r (R) adalah satu set n-tuples $r = \{t1, t2, \dots, tm \}$, dimana :

→ Setiap n nilai-nilai $t = \langle V1, V2, \dots, Vn \rangle$, dimana setiap nilai $V1, 1 \leq i \leq n$ merupakan satu elemen dari dom (Ai), atau berupa nilai khusus null (belum diketahui atau nilainya memang tidak ada untuk beberapa tuple).

Definisi nilai relasi di atas dapat juga dinyatakan seperti berikut :

Suatu relasi r(R) adalah subset dari 'Cartesian Product' dari domains yang mendefinisikan R :

$$r(R) \subseteq (\text{dom} (A1) \times \text{dom} (A2) \times \dots \times \text{dom} (An))$$

→ Dari semua kemungkinan kombinasi yang ada, nilai relasi untuk satu waktu tertentu (current relation state) hanya mencerminkan tuples yang valid yang menyajikan suatu state tertentu dari dunia nyata.

3.1.2. Karakteristik-karakteristik Relasi

- Pengurutan Tuple dalam suatu Relasi
 - Relasi didefinisikan sebagai satu set tuple.
 - Secara matematis, elemen-elemen dari suatu set tidak mempunyai urutan.
Jadi, tuples dalam suatu relasi tidak mempunyai urutan-urutan tertentu.

- Pengurutan nilai dalam suatu tuple
 - Berdasarkan definisi sebelumnya bahwa suatu n-tuple merupakan ordered-list (Satu list terurut) dari n nilai; sehingga urutan-urutan nilai dalam suatu tuple (demikian juga urutan attribute dalam suatu relasi) menjadi penting.

- Nilai-nilai dalam Tuple
 - Setiap nilai dalam suatu tuple merupakan satu nilai yang atomic (tidak dapat dipecah menjadi komponen-komponen yang lebih kecil dalam konteks model relasional).
 - Sehingga Composite dan Multivalued attribute tidak diperbolehkan dalam model relasional.
 - Attribute-attribute multivalued harus dipisahkan menjadi relasi-relasi yang terpisah.
 - Attribute-attribute composite harus disajikan oleh simple attributes yang membentuknya.

3.2. CONSTRAINTS UNTUK MODEL DATA RELASIONAL

1. DOMAIN CONSTRAINTS

Nilai dari setiap attribute suatu relasi harus atomic dari domain attribute tersebut.

2. KEY CONSTRAINTS

Suatu relasi adalah satu set tuple, dan semua elemen dalam satu set harus berbeda (tidak boleh terdapat dua tuple yang mempunyai nilai yang sama untuk semua attributenya).

3. ENTITY INTEGRITY CONSTRAINTS

Dalam suatu relasi, tidak boleh terdapat “Primary Key” yang bernilai kosong (null).

4. REFERENTIAL INTEGRITY CONSTRAINT

Constraint ini berkaitan dengan konsep FOREIGN KEY (Kunci Tamu) antara dua buah relasi yang mempunyai ”relationship”.

KONSEP FOREIGN KEY :

Untuk dua relasi R1 dan R2, satu set attribute FK dalam skema R1 disebut sebagai kunci tamu dari R2, jika memenuhi 2 attribute sbb :

- (i) Attribute-attribute dalam FK mempunyai domain yang sama dengan attribute yang menjadi primary key dalam skema relasi R2.
 - Attribute-attribute FK dikatakan mengacu pada relasi R2.
- (ii) Suatu nilai FK dalam suatu tuple t1 dalam R1 mungkin :
 - a) Berupa suatu nilai PK dalam sembarang tuple t2 dalam R2

Atau

- b) Bernilai kosong (tidak ada).

Dalam kasus (a), berlaku $t1 [FK] = t2 [PK]$ dan dikatakan bahwa tuple t1 mengacu ke tuple t2.

3.3. OPERASI-OPERASI UPDATE PADA RELASI

- Ada 3 operasi dasar :
 - Insert
 - Delete
 - Modify
- Penggunaan dari operasi update harus tidak melanggar integrity constraint yang telah ditentukan pada skema basis data relasional.

1. OPERASI INSERT

- Menyisipkan satu tuple baru ke dalam suatu relasi R.
- Operasi ini dapat melanggar 4 jenis constraint :
 - a) DOMAIN constraint dapat dilanggar jika suatu nilai attribute yang diberikan tidak ada dalam domain yang berkorespondensi dengan attribute tadi.

- b) KEY constraint dapat dilanggar jika nilai key dalam tuple baru t sudah ada dalam tuple lain dalam relasi r(R).
- c) ENTITY INTEGRITY dapat dilanggar jika primary key dari tuple baru t adalah NULL.
- d) REFERENTIAL INTEGRITY dapat dilanggar jika nilai dari suatu foreign key dalam t mengacu ke suatu tuple yang tidak ada dalam relasi yang diacu.

1.1. Manakah relasi operasi insert berikut yang valid?

- 1. Insert <'kolonsky', '6776', '05-APR-58', '6357 Windy Lane Texas', F, 28000, null, 4 > into EMPLOYEE.
- 2. Insert <'Zelaya', '9998', 05-MAY-50', '6357 Windy Lane Texas ', F, 28000, '9876', 4> into EMPLOYEE.
- 3. Insert <'kolonsky', null, '05-APR-58', '6357 Windy Lane Texas', F, 28000, null, 4> into EMPLOYEE.
- 4. Insert <'kolonsky', '6776', '05-APR-58', '6357 Windy Lane Texas', F, 28000, '9876', 7> into EMPLOYEE

1.2. Pilihan tindakan yang dapat dilakukan jika satu atau lebih constraint dilanggar akibat operasi insert :

- 1. Menolak (reject) operasi insertion.
Akan berguna jika DBMS menjelaskan mengapa proses insertion ditolak !
- 2. Berupa memperbaiki alasan penolakan proses insertion.
Insertion diterima jika kemudian user melakukan perubahan nilai-nilai atribut sehingga insertion dapat diterima.

2. OPERASI DELETE

- 1. Operasi ini hanya dapat melanggar referential integrity, jika tuple yang akan dihapus diacu oleh kunci-kunci tamu dari tuple yang lain dalam database.
- 2. Manakah dari contoh operasi delete berikut yang valid?
 - 1. Delete tuple dalam Works-on dengan Essn = '9998' dan PNO = 10
 - 2. Delete tuple dalam EMPLOYEE dimana SSN = '9998'
 - 3. Delete tuple dalam EMPLOYEE dengan SSN = '3334'
- 3. Pilihan tindakan jika suatu deletion melanggar constraint yang telah ditentukan :
 - 1. Menolak (reject) proses deletion.

2. Berusaha untuk melakukan "cascade (propagate deletion)", yaitu dengan menghapus sejumlah tuple yang mengacu pada tuple yang akan dihapus.
3. Melakukan modifikasi nilai-nilai attribute yang mengacu pada tuple yang dihapus, yaitu setiap nilai diset NULL atau diganti dengan nilai dari tuple lain yang valid sebagai acuan baru.
 - Ingat bahwa jika attribute yang mengacu V8 menyebabkan pelanggaran adalah bagian dari primary key, maka ia tidak dapat diset dengan NULL (melanggar entity integrity).
4. Kombinasi 2 dan 3

3. OPERASI MODIFY

- Digunakan untuk merubah nilai-nilai satu atau lebih attribute dalam satu atau lebih tuple dalam sejumlah relasi R.
- Melakukan modifikasi suatu attribut yang bukan primary key atau bukan foreign key biasanya tidak akan menimbulkan masalah.
(DBMS hanya perlu untuk mengecek apakah nilai-nilai baru yang diberikan mempunyai tipe data dan domain yang valid).

Melakukan modifikasi nilai suatu primary key serupa dengan proses deletion satu tuple dan inserting yang lain pada tempat yang sama (karena menggunakan primary key untuk mengidentifikasi tuple).

↳ Akibatnya, pilihan-pilihan seperti yang dilakukan pada operasi INSERT dan DELETE dapat dipakai agar modifikasi tidak melanggar constrain.

Jika yang dimodifikasikan berupa foreign keys, maka DBMS harus melakukan pengecekan bahwa nilai-nilai baru yang diberikan mengacu pada tuple yang ada dalam relasi-relasi yang diajukan acuan.

3.4 MENDEFINISIKAN RELASI

- Skema Dasis Data harus didefinisikan pada saat diimplementasikan untuk digunakan dalam sejumlah aplikasi yang meliputi :
 - keputusan mengenai attribute-attribute yang akan dikelompokkan dalam satu relasi
 - menentukan nama-nama yang sesuai untuk relasi dan attribute

- menentukan domain dan tipe data untuk semua attribute yang ada
 - mengidentifikasi candidate keys, dan memilih satu primary key untuk setiap relasi serta menspesifikasikan semua foreign keys.
- ↳ Fasilitas pendefinisian sangat tergantung dari DBMS yang digunakan.

Contoh beberapa definisi relasi dari database COMPANY dalam SQL (SQLq2):

CREATE TABLE EMPLOYEE

FNAME	VARCHAR(15)	NOT NULL
MINIT	CHAR,	
LNAME	VARCHAR(15)	NOT NULL
SSN	CHAR(4)	NOT NULL
BDATE	DATE,	
ADDRESS	VARCHAR(30),	
SEX	CHAR,	
SALARY	DECIMAL(10,2),	
SUPERSSN	CHAR(9),	
DNO	INT	NOT NULL DEFAULT

CONSTRAINT EMPPK

PRIMARY KEY(SSN)

CONSTRAIN EMPSUPERFK

FOREIGN KEY(SUPERSSN) REFERENCES EMPLOYEE (SSN)

ON DELETE SET NULL

ON UPDATE CASCADE,

CONSTRAIN EMPDEPTPK

FOREIGN KEY (DNO) REFERENCE DEPARTMENT (DNUMBER,

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

3.5 ALGORITMA PEMETAAN KE SKEMA ER RELASIONAL

1. untuk setiap entity type E yang regular (non-weak) dalam skema ER, buat satu relasi R yang atributenya meliputi semua simple attribute dari E. Untuk suatu composite attribute, hanya komponen simple atributenya yang dipetakan. Kemudian pilih satu key attribute dari E sebagai primary key dari R. Jika key terpilih adalah composite, maka primary key dari R dibentuk dari satu set simple attribute dari composite

Note : Dalam langkah pertama ini, keterlibatan kunci tamu dan relationship attribute jangan dipikirkan dulu.

2. Untuk setiap weak entity type W dalam skema ER dengan owner entity type E, buat satu relasi R yang atributenya meliputi semua simple attribute (atau simple component dari composite attribute).

Masukkan sebagai foreign key dalam R, Primary key attribute dari relasi-relasi yang merupakan owner/pemilik dari W (dapat diperoleh dengan memperhatikan identifying relationship type dari W).

Primary key dari R dibentuk dari kombinasi antara primary key dari relasi-relasi pemilik dan partial key dari weak entity type W (jika ada).

3. Untuk setiap binary 1:1 relationship type R dalam skema ER, perhatikan relasi-relasi S dan T yang berkorespondensi dengan entity type yang berpartisipasi dalam R. Kemudian, ambil salah satu relasi (misalkan S) dan cantumkan primary key dari T sebagai foreign key dalam S. Untuk kasus ini, S sebaiknya dipilih dari entity type yang berpartisipasi total dalam R (untuk mengurangi null value dari foreign key yang diambil dari T).

Notes :

Sebagai alternative untuk pemetaan dari 1:1 relationship type ini adalah dengan menggabungkan kedua entity types menjadi satu relasi tunggal. Cara ini terutama sangat sesuai bilamana kedua entity type berpartisipasi total dan tidak berpartisipasi dalam relationship type yang lain.

4. Untuk setiap regular (non-weak relasi) binary 1 : N relationship type R, perhatikan relasi S yang mewakili entity type yang berpartisipasi pada sisi-N dari relationship type.

Cantumkan primary key dari relasi yang berperan pada sisi-1 sebagai foreign key dalam S, karena setiap entity instance pada sisi-N terkait dengan paling banyak satu entity instance pada sisi-1 dari relationship type. Kemudian, cantumkan semua simple attribute (atau simple component dari composite attribute) dari 1 relationship type sebagai attribute dalam S.

5. Untuk setiap binary M : N relationship type R, buat satu relasi baru S untuk mewakili R. kemudian cantumkan sebagai foreign key dalam S. Primary key dari relasi-relasi yang mewakili entity type yang berpartisipasi dalam R; dimana kombinasi dari primary keys ini akan membentuk primary key dari S juga, cantumkan semua simple attribute (semua simple component dari composite attribute) dari M:N relationship type sebagai attribute dari S.

Notes:

Binary relationship 1 : 1 atau 1 : N selalu dapat dipetakan seperti pada M:N relationship. Alternatife ini terutama sangat berguna bilamana relationship instances yang ada sangat sedikit (menghindari null values dalam foreign key).

Untuk kasus ini, primary key dari relasi "relationship" hanya akan menjadi foreign key dari satu "entity" relasi yang berpartisipasi. Jadi, untuk 1:N relationship, hanya pada sisi-N, sedang untuk 1:1 adalah entity relasi yang berpartisipasi total (jika ada) yang dipilih.

6. Untuk setiap multivalued attribute A, buat satu relasi baru R yang mencantumkan satu attribute yang mewakili A ditambah dengan primary key K (sebagai foreign key dalam R) dari relasi yang mewakili entity type atau relationship type yang mempunyai A sebagai atributenya. Primary key dari R adalah kombinasi A dan K. jika multivalued attribute adalah composite, maka hanya simple componentnya yang dicantumkan.
7. Untuk setiap n-ary relationship type R ($n > 2$), buat satu relasi baru S yang mewakili R. cantumkan sebagai foreign key attribute dalam S, primary key dari relasi-relasi yang mewakili entity types yang berpartisipasi juga cantumkan sebagai attribute dalam S. Semua simple attribute (simple component dari composite attribute) dalam n-ary relationship type.

Primary key dari S biasanya berupa kombinasi dari foreign key yang mengacu pada relasi-relasi yang mewakili entity types yang berpartisipasi. Namun, jika ‘participation constraint’ (min, max) dari satu entity type E yang berpartisipasi dalam R mempunyai $\underline{\text{max}}=1$, maka primary key dari S berupa satu-satu foreign key attribute yang mengacu pada relasi E’ (yang berkorespondensi dengan E). ini dilakukan karena setiap entity instance e dalam E hanya akan berpartisipasi dalam paling banyak satu relationship instance dalam R sehingga dapat secara unik mengidentifikasi relationship instance dari R.

KORESPONDENSI ANTARA MODEL ER DENGAN MODEL RELASIONAL

<u>ER MODELS</u>	→	<u>RELATIONAL MODEL</u>
○ Entity type	→	○ "entity" relation
○ 1:1 or 1:N relationship type	→	○ foreign key ("relationship" relation)
○ M : N relationship type	→	○ "relationship" relation and two Foreign keys
○ n-ary relationship	→	○ "relationship" relation and n foreign keys
○ simple attribute	→	○ attribute
○ composite attribute	→	○ set of simple component attributes
○ multivalued attribute	→	○ relation and foreign key
○ value set	→	○ domain
○ Key attribute	→	○ Primary (or secondary) key