



Basis Data

Ketergantungan Fungsional

Ketergantungan Fungsional (KF)

- Functional Dependencies (FD) / Ketergantungan Fungsional (KF) digunakan untuk menggambarkan atau mendeskripsikan bentuk normal atas suatu relasi
- FD adalah batasan terhadap gugus relasi yang berlaku. Diperoleh berdasarkan hubungan antar atribut data.
- Kegunaan FD :
 1. Untuk memeriksa keabsahan apakah semua relasi sesuai dengan ketergantungan fungsional yang diberikan
 2. Untuk menetapkan batasan gugus relasi yang berlaku
 3. Untuk menentukan kunci relasi
 4. Untuk melakukan normalisasi atas suatu tabel relasional

Ketergantungan Fungsional (KF)

definisi

Misalkan R adalah suatu skema relasional, atribut $x \subseteq R$ dan $y \subseteq R$ maka x dikatakan secara fungsional menentukan y (atau y bergantung secara fungsional pada x), ditulis $x \rightarrow y$ pada R, jika :

1. Semua tupel $t_i[x]$, $1 \leq i \leq n$ adalah unik/tunggal
2. Semua pasangan tupel dimana $t_i[x] = t_j[x]$, $i \neq j$, terjadi juga $t_i[y] = t_j[y]$

dengan kata lain :

Untuk setiap nilai x terdapat hanya satu nilai y (x menentukan secara tunggal nilai y). Jadi apabila terdapat 2 tuple t_1 dan t_2 mempunyai nilai atribut x yang sama, maka juga akan mempunyai nilai atribut y yang sama.

$$t_1[x] = t_2[x] \Rightarrow t_1[y] = t_2[y] \text{ pada skema relasi R}$$

Ketergantungan Fungsional (KF)

contoh

R = (A, B, C)

A	B	C
1	4	C1
1	5	C1
2	7	C2

- $A \rightarrow B$?
 $t_1(A)=t_2(A)$, tetapi $t_1(B) \neq t_2(B)$
Maka $A \not\rightarrow B$
- $A \rightarrow C$?
 $t_1(A)=t_2(A)$ dan $t_1(C) = t_2(C)$
Maka $A \rightarrow C$

R = (A, B, C, D)

A	B	C	D
A1	B1	C1	D1
A1	B2	C1	D2
A2	B2	C2	D2
A2	B3	C2	D3
A3	B3	C2	D4

- $A \rightarrow C$? **Yes**
- $C \rightarrow A$? **No**
- $(A,B) \rightarrow C$? **Yes**
- $(A,B) \rightarrow D$? **Yes**

Ketergantungan Fungsional (KF) **contoh**

Film = (idfilm, title, year, length, filmType, idstudio, studioName, idstar, starName)

idfilm	title	year	length	filmType	idstudio	studioName	idstar	starName
F001	Star Wars	1977	124	color	STD01	Fox	STR01	Carrie Fisher
F001	Star Wars	1977	124	color	STD01	Fox	STR02	Mark Hamill
F001	Star Wars	1977	124	color	STD01	Fox	STR03	Harrison Ford
F002	Mighty Ducks	1991	104	color	STD02	Disney	STR04	Emilio Estevez
F003	Wayne's World	1992	95	color	STD03	Paramount	STR05	Dana Carvey
F003	Wayne's World	1992	95	color	STD03	Paramount	STR06	Mike Meyers
F004	My Hearts	1992	101	color	STD03	Paramount	STR01	Carrie Fisher
F004	My Hearts	1992	101	color	STD03	Paramount	STR01	Carrie Fisher

Apakah : idfilm \rightarrow title ?

idstar \rightarrow studioName ?

(idfilm,idstar) \rightarrow starName ?

Ketergantungan Fungsional (KF)

- FD dirumuskan berdasarkan batasan dari dunia nyata suatu atribut.

Contoh :

- Nomor Induk mahasiswa menentukan Nama Mahasiswa
 $NIM \rightarrow NamaMhs$
- Kode Matakuliah menentukan Nama Mata Kuliah dan SKS
 $KodeMK \rightarrow (NamaMK, SKS)$
- NIM dan Kode Mata Kuliah menentukan Nilai Matakuliah
 $(NIM, KodeMK) \rightarrow NilaiMK$

- Suatu FD : $x \rightarrow y$ disebut trivial jika $y \subseteq x$

Contoh :

$$\begin{array}{ll} X \rightarrow X & X, Y, Z \rightarrow X, Z \\ X, Y \rightarrow X & X, Y, Z \rightarrow Z \\ X, Y \rightarrow Y & X, Y, Z \rightarrow X, Y, Z \end{array}$$

Ketergantungan Fungsional (KF)

Armstrong's Rule

A1. Reflexive

Jika $y \subseteq x$ maka $x \rightarrow y$

A2. Augmentation

Jika $x \rightarrow y$ maka $(x,z) \rightarrow (y,z)$

A3. Transitive

Jika $x \rightarrow y$ dan $y \rightarrow z$ maka $x \rightarrow z$

A4. Decomposition

Jika $x \rightarrow (y,z)$ maka $x \rightarrow y$ dan $x \rightarrow z$

A5. Union

Jika $x \rightarrow y$ dan $x \rightarrow z$ maka $x \rightarrow (y,z)$

A6. Pseudotransitivity

Jika $x \rightarrow y$ dan $(z,y) \rightarrow w$ maka $(x,z) \rightarrow w$

Diketahui $x \rightarrow y$
Dari A2 $(x,z) \rightarrow (y,z)$
Diketahui $(z,y) \rightarrow w$
Dari A3 $(x,z) \rightarrow w$

KUNCI

atribut kunci (key)

- Kunci (key) adalah kolom/atribut atau kombinasi kolom/atribut yang dapat digunakan untuk mengidentifikasi baris dalam tabel (entitas) secara unik.
- Penentuan Key suatu tabel didasarkan pada sifat "determinasi".
- Determinan : gugus atribut dimana satu atau lebih atribut lain tergantung secara fungsional.
"A determinan B" artinya apabila nilai atribut A akan menentukan nilai-nilai atribut B.
"A determinan B" dapat dituliskan sebagai suatu ketergantungan fungsional $A \rightarrow B$. Jika A menentukan B,C dan D maka dituliskan $A \rightarrow B,C,D$.

Contoh : Relasi Mahasiswa=(NIM>Nama,Agama,TglLhr)

Bila nilai NIM seorang mahasiswa diketahui maka dapat digunakan untuk melihat nilai-nilai atribut Nama, Agama dan Tanggal Lahirnya.
Dituliskan $NIM \rightarrow Nama,Agama,TglLhr$

superkey

Superkey (key) :

- gugus atribut entitas yang dapat digunakan untuk mengidentifikasi entitas/obyek secara unik.
- satu atau lebih atribut yang membedakan setiap baris secara unik.

Misal R skema relasi, dan K adalah satu atau lebih atribut dari R dimana $K \subseteq R$ maka K disebut Superkey jika dan hanya jika $K \rightarrow R$.

Catatan : Suatu skema relasi dapat memiliki lebih dari 1 superkey.

Bila K adalah superkey maka semua atribut gabungan yang mengandung K juga merupakan superkey

Contoh : Relasi Sopir=(NoKTP,NoSIM>Nama,Alamat). Alternatif superkey :

- NoKTP superkey ; $\text{NoKTP} \rightarrow \text{Sopir}$
- NoSIM superkey ; $\text{NoSIM} \rightarrow \text{Sopir}$
- (NoKTP,NoSIM) superkey ; $(\text{NoKTP},\text{NoSIM}) \rightarrow \text{Sopir}$
- (NoKTP>Nama) superkey ; $(\text{NoKTP},\text{Nama}) \rightarrow \text{Sopir}$
- (NoKTP,NoSIM>Nama) superkey ; $(\text{NoKTP},\text{NoSIM},\text{Nama}) \rightarrow \text{Sopir}$
- (NoKTP,NoSIM>Nama,Alamat) dengan sendirinya juga superkey
- Nama bukan superkey. Demikian juga (Nama,Alamat) juga bukan superkey

candidate key

- Candidate Key :
 - Superkey dengan jumlah atribut minimal
 - Superkey tanpa redundansi (tidak memuat subset superkey yang lain)
- K adalah Candidate Key dari skema relasi R jika dan hanya jika :
 $K \rightarrow R$ dan tidak terdapat $\alpha \subset K$ dengan $\alpha \rightarrow R$

Contoh :

Skema relasi Sopir=(NoKTP,NoSIM>Nama,Alamat). Alternatif superkey :

- NoKTP superkey ; NoKTP \rightarrow Sopir
- NoSIM superkey ; NoSIM \rightarrow Sopir
- (NoKTP,NoSIM) superkey ; (NoKTP,NoSIM) \rightarrow Sopir
- (NoKTP>Nama) superkey ; (NoKTP>Nama) \rightarrow Sopir
- (NoKTP,NoSIM>Nama) superkey ; (NoSKTP,NoSIM>Nama) \rightarrow Sopir
- (NoKTP,NoSIM>Nama,Alamat) dengan sendirinya juga superkey

Sebagai Candidate Key nya adalah **NoKTP** atau **NoSIM**

secondary key

- Secondary Key adalah atribut (atau kombinasinya), yang digunakan sebagai perantara untuk mendapatkan kembali data asal.
- Biasanya dipakai pada pencarian data (data retrieval).

Contoh :

Skema relasi Sopir=(NoKTP,NoSIM,Nama,Alamat) dengan NoSIM sebagai Primary Key. Walaupun atribut ini lazim dipakai sebagai identitas seorang Sopir, tapi apakah seorang sopir dijamin hapal nomor SIM nya ketika misalnya ada transaksi yang berkaitan dengan penggunaan identitas No SIM ?.

Untuk memudahkan proses pencarian data sopir tersebut maka dapat digunakan atribut lain yang lebih mudah diingat misalnya "nama" dan/atau "alamat".

Penggunaan secondary key ini tentu saja tidak menjamin ditemukannya data yang unik, karena memang tidak ditujukan untuk kepentingan keunikan data. Tetapi sebagai alternatif atau fasilitas untuk membantu mengidentifikasi data.

Analogikan ketika kita lupa akan ID atau password account email kita.

Fasilitas apa yang bisa kita manfaatkan ?

foreign key

- Foreign Key adalah satu atau lebih atribut dalam satu tabel yang merupakan primary key tabel lain (kunci penghubung).

Produk

IDProd	NamaProduk	Harga	QtyStock
F001	TV 14"	1500000	12
F002	TV 21"	2100.000	4
F003	TV 21" Flatron	2700000	24

Tabel Name : Produk
Primary key : IDProd
Foreign Key : -

Sales

IDSIs	NmSIs	AlamatAsal	KotaAsal
S001	Anita	Jl. Nakula 9	Kendal
S002	Vicky	Jl. Arjuna I/6	Semarang
S003	Roni	Jl. Bima II/3	Semarang

Tabel Name : Sales
Primary key : IDSIs
Foreign Key : -

Order

NoOrder	Date	IDProd	QtyOrder	IDSIs
120301	12/11/04	P001	2	S001
120302	13/11/04	P001	2	S003
120303	22/11/04	P003	6	S001

Tabel Name : Order
Primary key : NoOrder
Foreign Key : IDProd,IDSIs

link

link

hubungan KF dengan key

Amstrong's rule dapat digunakan untuk menurunkan superkey tabel, berdasarkan 1 atau lebih superkey yang diketahui

Rule 1 :

Apabila diketahui FD yang memuat semua atribut pada tabel, maka atribut-atribut yang terdapat pada ruas kiri dari FD adalah superkey

Contoh :

Diketahui tabel $R = (W, X, Y, Z)$ dan FD : $XY \rightarrow WZ$ maka XY superkey

Sebab :

$XY \rightarrow WZ$ maka

$XY \rightarrow XY$ (refleksif)

$XY \rightarrow XYWZ$ (union)

$XY \rightarrow R$

Karena $XY \rightarrow R$ maka XY superkey.

Jadi ruas kiri dari FD merupakan superkey.

hubungan KF dengan key

Rule 2 :

Atribut yang secara fungsional menentukan superkey dari tabel maka atribut tersebut juga merupakan superkey

Contoh :

Diketahui W superkey dari tabel $R = (W, X, Y, Z)$ dan FD : $Z \rightarrow W$
maka Z superkey

Sebab :

$Z \rightarrow W$ dan

$W \rightarrow WXYZ$ (karena W superkey), maka

$Z \rightarrow WXYZ$ (transitif)

$Z \rightarrow R$

Karena $Z \rightarrow R$ maka Z superkey

hubungan KF dengan key

$R = (A, B, C, D)$

A	B	C	D
A1	B1	C1	D1
A1	B2	C1	D2
A2	B2	C2	D2
A2	B3	C2	D3
A3	B3	C2	D4

- Apakah (A, B) superkey dari R ?
Akan dibuktikan apakah $(A, B) \rightarrow R$.
Jika Ya maka (A, B) superkey dari R .

Karena semua tupel $t_i[A, B]$ untuk $1 \leq i \leq 5$ adalah unik,

$$t_1[A, B] = (A1, B1)$$

$$t_2[A, B] = (A1, B2)$$

$$t_3[A, B] = (A2, B2)$$

$$t_4[A, B] = (A2, B3)$$

$$t_5[A, B] = (A3, B3)$$

Maka $(A, B) \rightarrow (A, B, C, D)$ atau $(A, B) \rightarrow R$
Jadi (A, B) superkey dari R

- Apakah A superkey dari R ?
Bukan, sebab $A \not\rightarrow R$. Mengapa ?

hubungan KF dengan key

Diketahui $S = (A, B, C, D, E, F)$ dan

KF : $A \rightarrow BC$; $B \rightarrow D$; $C \rightarrow EF$; $BF \rightarrow A$

Carilah superkey dan candidate key dari S menggunakan KF

- $A \rightarrow BC$
 $A \rightarrow B$
 $A \rightarrow C$
- karena $A \rightarrow B$
dan $B \rightarrow D$ maka
 $A \rightarrow D$
- karena $A \rightarrow C$
dan $C \rightarrow EF$ maka
 $A \rightarrow EF$
- $A \rightarrow A$
Sehingga $A \rightarrow ABCDEF$
atau $A \rightarrow S$ (superkey)

- $B \rightarrow D$ maka
 $BC \rightarrow DC$
- $C \rightarrow EF$ maka
 $BC \rightarrow BEF$
- Jadi $BC \rightarrow BCDEF$
- $BC \rightarrow BCDEF$ dan
 $BC \rightarrow BF$ dan $BF \rightarrow A$
maka $BC \rightarrow A$
 $BC \rightarrow ABCDEF$
 $BC \rightarrow S$ (superkey)

- $BF \rightarrow A$
 $A \rightarrow ABCDEF$ maka
 $BF \rightarrow ABCDEF$
 $BF \rightarrow S$ (superkey)

- Superkey dari S
 A , BC , BF serta gabungan
atribut yang mengandung
 A , BC dan BF
- Candidate key dari S
 A , BC , BF

Tips !!
Fokuskan
perhatian Anda
pada atribut-2 di
ruas kiri dari FD
untuk mencari
superkey

hubungan KF dengan key

Latihan :

1. Diberikan $R(A,B,C,D)$ dengan FD : $A \rightarrow B, A \rightarrow C$ dan $A \rightarrow D$
Apakah A candidate key dari R ?
2. Diberikan $R(A,B,C,D)$ dengan FD : $A \rightarrow B$
 - a. Apakah ACD superkey dari R
 - b. Apakah A candidate key dari R
3. Diberikan $R(A,B,C,D,E,F)$ dengan FD : $C \rightarrow (AB); B \rightarrow (DE); E \rightarrow F; A \rightarrow BC$
 - a. Carilah superkey dari R
 - b. Carilah candidate key dari R
4. Diberikan $R(A,B,C,D,E)$ dengan FD : $A \rightarrow (BC); (CD) \rightarrow E; B \rightarrow D; E \rightarrow A$
 - a. Carilah superkey dari R
 - b. Carilah candidate key dari R
5. Diberikan $R(A,B,C)$ dengan FD : $A \rightarrow B; B \rightarrow C; C \rightarrow A$
Apakah A merupakan satu-satunya candidate key dari R

DEKOMPOSISI RELASI

Manfaat KF pada dekomposisi

Untuk :

1. Lossless Join Decomposition

Mendapatkan dekomposisi yang tidak kehilangan data/informasi

2. No Redundancy

Mendapatkkan skema relasi yang tidak mengandung redundansi

3. Dependency Preservation

Terjaminnya pemeliharaan ketergantungan sehingga dapat mengatasi masalah update anomali

uji lossless-join decomposition

Misal diketahui skema relasi R didekomposisi menjadi gugus relasi $\{R_1, R_2, R_3, R_4, \dots, R_n\}$, maka dekomposisi ini disebut Lossless Join Decomposition jika kondisi $R_1 \cap R_2 \cap R_3 \cap \dots \cap R_n \rightarrow R_i$ dipenuhi sekurang-kurangnya untuk 1 nilai i , dimana $1 \leq i \leq n$.

Dengan kata lain, jika diketahui skema relasi R didekomposisi menjadi gugus relasi $\{R_1, R_2\}$, maka dekomposisi ini disebut Lossless Join Decomposition jika dipenuhi salah satu kondisi :

- $R_1 \cap R_2 \rightarrow R_1$ atau
- $R_1 \cap R_2 \rightarrow R_2$

Langkah-2 Uji Lossless-joint Decomposition :

1. Uji Dekomposisi

$$\mathbf{R_1 \cup R_2 \cup \dots \cup R_n = R}$$

2. Uji Lossless-join

Menggunakan sifat ketergantungan fungsional

uji lossless-join decomposition **contoh**

Diketahui skema relasi $R=(A,B,C,D,E,F,G,H)$ didekomposisi menjadi :
 $R1=(A,B,C,D,G)$ dan $R2=(B,D,E,F,H)$. FD pada R yang berlaku adalah :

(1) $B \rightarrow A,G$

(2) $E \rightarrow D,H$

(3) $A \rightarrow E,C$

(4) $D \rightarrow F$

Ujilah apakah dekomposisi $\{R1,R2\}$ tersebut lossless atau lossy ?

1. Uji Dekomposisi

$$\begin{aligned} R1 \cup R2 &= (A,B,C,D,G) \cup (B,D,E,F,H) \\ &= (A,B,C,D,E,F,G,H) \\ &= R \end{aligned}$$

\therefore Terbukti bahwa $\{R1,R2\}$ adalah dekomposisi dari R .

uji lossless-join decomposition

contoh

terbukti
{R1,R2}
Lossless

2. Uji Lossless

$$\begin{aligned} R1 \cap R2 &= (A,B,C,D,G) \cap (B,D,E,F,H) \\ &= (B,D) \end{aligned}$$

Akan dibuktikan bahwa paling sedikit satu kondisi berikut dipenuhi :

- $R1 \cap R2 \rightarrow R1$; $(B,D) \rightarrow (A,B,C,D,G)$ atau
- $R1 \cap R2 \rightarrow R2$; $(B,D) \rightarrow (B,D,E,F,H)$

Dari (1) $B \rightarrow A,G$ maka :
(5) $B,D \rightarrow A,G,D$ (augmentasi)
(6) $B,D \rightarrow B,D$ (refleksif)
Jadi (7) $B,D \rightarrow A,B,D,G$
(1) $B \rightarrow A,G$ maka
(8) $B \rightarrow A$ dan
(9) $B \rightarrow G$
Dari (3) $A \rightarrow E,C$ maka
(10) $A \rightarrow E$ dan
(11) $A \rightarrow C$ maka :

Dari (8) $B \rightarrow A$
dan (11) $A \rightarrow C$
Maka (12) $B \rightarrow C$ (transitif)
Dan (13) $B,D \rightarrow C,D$ (augmentasi)
Dari (7) dan (13) didapat :
 $B,D \rightarrow A,B,C,D,G$

Dari contoh di atas, tunjukkan pula bahwa $(B,D) \rightarrow (B,D,E,F,H)$

Ketergantungan Fungsional (KF) Closure FD (F^+)

- Misal F adalah gugus ketergantungan fungsional pada skema relasi R , maka semua FD yang mungkin dapat diturunkan dari F dengan hukum-hukum FD disebut : **Closure dari F** , ditulis F^+ .
- Armstrong's rule dapat dimanfaatkan untuk menentukan F^+
Contoh : Diketahui $R = (A, B, C, D)$
 $F = \{ A \rightarrow B, B \rightarrow C, A \rightarrow C, C \rightarrow D \}$ maka :
 - $A \rightarrow D$
sebab $A \rightarrow C$ dan $C \rightarrow D$, dari sifat transitif (A3) didapat $A \rightarrow D$
 - $B \rightarrow D$
sebab $B \rightarrow C$ dan $C \rightarrow D$, dari sifat transitif (A3) didapat $B \rightarrow D$Sehingga $\{A \rightarrow B, B \rightarrow C, A \rightarrow C, C \rightarrow D, A \rightarrow D, B \rightarrow D\} \in F^+$.

Kita dapat menurunkan anggota-anggota F^+ yang lain berdasarkan FD yang diketahui menggunakan Armstrong's rule.

Closure FD (F^+) berguna untuk Uji Dependency Preservation

uji dependency preservation

Misal skema relasi R dengan himpunan ketergantungan fungsional F didekomposisi menjadi $R_1, R_2, R_3, \dots, R_n$. Dan $F_1, F_2, F_3, \dots, F_n$ adalah himpunan ketergantungan fungsional yang berlaku di $R_1, R_2, R_3, \dots, R_n$ maka dekomposisi tersebut dikatakan memenuhi sifat Dependency Preservation apabila berlaku :

$$(F_1 \cup F_2 \cup F_3 \cup \dots \cup F_n)^+ = F^+$$

Dependency Preservation (Pemeliharaan Ketergantungan) merupakan kriteria yang menjamin keutuhan relasi ketika suatu relasi didekomposisi menjadi beberapa tabel. Sehingga diharapkan tidak terjadi inkonsistensi atau anomali ketika dilakukan update data.

uji dependency preservation

Contoh :

Diketahui skema relasi $R=(A,B,C)$ dengan FD : $A \rightarrow B$; $B \rightarrow C$

Didekomposisi menjadi $R1=(A,B)$ dan $R2=(B,C)$

- a. Apakah dekomposisi tsb Lossless-Joint ?
- b. Apakah dekomposisi tsb memenuhi Dependency Preservation ?

a. $R1 \cup R2 = (A,B) \cup (B,C) = (A,B,C) = R$

$R1 \cap R2 = (A,B) \cap (B,C) = B$

Lossless jika $B \rightarrow (A,B)$ atau $B \rightarrow (B,C)$.

Karena diketahui $B \rightarrow C$ maka $BB \rightarrow BC$ atau $B \rightarrow BC$ (Augmentasi).

Jadi dekomposisi tsb Lossless.

uji dependency preservation

- b. $R=(A,B,C)$ dan $F = \{A \rightarrow B, B \rightarrow C\}$.
Karena $A \rightarrow B$ dan $B \rightarrow C$ maka $A \rightarrow C$. Maka dapat dibentuk closure
 $F^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$.

$R_1=(A,B)$ dan $F_1=\{A \rightarrow B\}$. Karena hanya $A \rightarrow B$ yang berlaku di R_1 .
 $R_2=(B,C)$ dan $F_2=\{B \rightarrow C\}$. Karena hanya $B \rightarrow C$ yang berlaku di R_2 .

$F_1 \cup F_2 = \{A \rightarrow B, B \rightarrow C\}$. Karena $A \rightarrow B$ dan $B \rightarrow C$ maka $A \rightarrow C$.
Sehingga $(F_1 \cup F_2)^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\} = F^+$

Jadi dekomposisi tsb memenuhi Dependency Preservation.

- Ujilah dekomposisinya apakah Lossless dan Dependency Preservation Apabila R di atas didekomposisi menjadi $R_1=(A,B)$ dan $R_2=(A,C)$.
- Bagaimana bila $R_1=(A,B)$ dan $R_2=(B,C)$ tetapi FD : $B \rightarrow C, AC \rightarrow B$

Dekomposisi relasi

- **Dekomposisi** : memecah relasi/tabel menjadi relasi/tabel yang lebih kecil untuk mendapatkan skema yang tidak mengandung anomali dan redundansi
- Diketahui skema relasi R. Gugus relasi $\{R_1, R_2, \dots, R_n\}$ disebut **Dekomposisi dari R** jika :

$$R_1 \cup R_2 \cup \dots \cup R_n = R$$

- **Artinya** $\{R_1, R_2, \dots, R_n\} \rightarrow$ dekomposisi dari R jika setiap atribut dalam R muncul paling sedikit di salah satu R_i untuk $1 \leq i \leq n$

Dekomposisi relasi

original

Film

idfilm	title	year	length	filmType	idstudio	studioName	idstar	starName
F001	Star Wars	1977	124	color	STD01	Fox	STR01	Carrie Fisher
F001	Star Wars	1977	124	color	STD01	Fox	STR02	Mark Hamill
F001	Star Wars	1977	124	color	STD01	Fox	STR03	Harrison Ford
F002	Mighty Ducks	1991	104	color	STD02	Disney	STR04	Emilio Estevez
F003	Wayne's World	1992	95	color	STD03	Paramount	STR05	Dana Carvey
F003	Wayne's World	1992	95	color	STD03	Paramount	STR06	Mike Meyers

DaftarFilm

idfilm	title	year	length	filmType
F001	Star Wars	1977	124	color
F002	Mighty Ducks	1991	104	color
F003	Wayne's World	1992	95	color

StudioFilm

idstudio	studioName
STD01	Fox
STD02	Disney
STD03	Paramount

BintangFilm

idstar	starName
STR01	Carrie Fisher
STR02	Mark Hamill
STR03	Harrison Ford
STR04	Emilio Estevez
STR05	Dana Carvey
STR06	Mike Meyers

Decomposition result

Dekomposisi relasi

- Dekomposisi relasi R menjadi gugus relasi $\{R_1, R_2, \dots, R_n\}$ yang tidak menyebabkan hilangnya informasi disebut **Lossless-Join Decomposition**.

Jadi, jika $r \in R$ dan $r_i = \Pi_{R_i}(R)$ dimana $1 \leq i \leq n$ maka akan selalu memenuhi kondisi berikut :

$$\text{atau} \quad r \subseteq \bigtimes_{i=1}^n r_i$$
$$r \subset \bigtimes_{i=1}^n r_i \text{ atau } r = \bigtimes_{i=1}^n r_i$$

- Dekomposisi relasi R menjadi gugus relasi $\{R_1, R_2, \dots, R_n\}$ yang menyebabkan hilangnya informasi disebut **Lossy-Join Decomposition**.

Lossless Join digunakan untuk menjamin keutuhan data untuk operasi gabungan (join) dan merupakan fokus dalam desain basis data relasional

Dekomposisi relasi

Film = (idfilm, title, year, length, filmType, idstudio, studioName, idstar, starName)

idfilm	title	year	length	filmType	idstudio	studioName	idstar	starName
F001	Star Wars	1977	124	color	STD01	Fox	STR01	Carrie Fisher
F001	Star Wars	1977	124	color	STD01	Fox	STR02	Mark Hamill
F001	Star Wars	1977	124	color	STD01	Fox	STR03	Harrison Ford
F002	Mighty Ducks	1991	104	color	STD02	Disney	STR04	Emilio Estevez
F003	Wayne's World	1992	95	color	STD03	Paramount	STR05	Dana Carvey
F003	Wayne's World	1992	95	color	STD03	Paramount	STR06	Mike Meyers

***there is Anomaly !!
Decompose it***

Dekomposisi relasi

Film

idfilm	title	year	length	filmType	idstudio	studioName	idstar	starName
F001	Star Wars	1977	124	color	STD01	Fox	STR01	Carrie Fisher
F001	Star Wars	1977	124	color	STD01	Fox	STR02	Mark Hamill
F001	Star Wars	1977	124	color	STD01	Fox	STR03	Harrison Ford
F002	Mighty Ducks	1991	104	color	STD02	Disney	STR04	Emilio Estevez
F003	Wayne's World	1992	95	color	STD03	Paramount	STR05	Dana Carvey
F003	Wayne's World	1992	95	color	STD03	Paramount	STR06	Mike Meyers

DaftarFilm

idfilm	title	year	length	filmType
F001	Star Wars	1977	124	color
F002	Mighty Ducks	1991	104	color
F003	Wayne's World	1992	95	color

StudioFilm

idstudio	studioName
STD01	Fox
STD02	Disney
STD03	Paramount

BintangFilm

idstar	starName
STR01	Carrie Fisher
STR02	Mark Hamill
STR03	Harrison Ford
STR04	Emilio Estevez
STR05	Dana Carvey
STR06	Mike Meyers

Andaikan di dekomposisi
Menjadi 3 tabel tsb

Lossless
or
Lossy ?

KF dan Normalisasi

Dekomposisi relasi

Dengan ke-3 tabel hasil dekomposisi, misal ditanyakan informasi :

“Di studio manakah Star Wars dibuat ?”

Pasti kita akan membutuhkan tabel DaftarFilm dan StudioFilm.

Tapi dapatkah kita diperoleh informasi yang kita inginkan dari kedua skema relasi tersebut ?

Tampaknya : TIDAK.

Karena kita harus melakukan operasi gabungan terlebih dahulu dari Ke-2 tabel. Misal kita lakukan operasi “cross product” antara Daftar_Film dan StudioFilm.

DaftarFilm × StudioFilm

KF dan Normalisasi

Dekomposisi relasi

Kehilangan Informasi !!

DaftarFilm × **StudioFilm**

idfilm	title	year	length	filmType	idstudio	studioName
F001	Star Wars	1977	124	color	STD01	Fox
F001	Star Wars	1977	124	color	STD02	Disney
F001	Star Wars	1977	124	color	STD03	Paramount
F002	Mighty Ducks	1991	104	color	STD01	Fox
F002	Mighty Ducks	1991	104	color	STD02	Disney
F002	Mighty Ducks	1991	104	color	STD03	Paramount
F003	Wayne's World	1992	95	color	STD01	Fox
F003	Wayne's World	1992	95	color	STD02	Disney
F003	Wayne's World	1992	95	color	STD03	Paramount

Ternyata kita tidak mendapatkan informasi yang dibutuhkan, karena film Star Wars dibuat oleh 3 studio (Fox, Disney, Paramount)

KF dan Normalisasi

soal latihan

Ujilah dekomposisi dari skema relasi R, apakah lossless atau lossy ?

1. $R = (A, B, C, D, E, F, G, H)$ didekomposisi menjadi :
 $R_1 = (A, B, C, D, E)$ dan $R_2 = (C, D, F, G, H)$ dengan FD :
 $C \rightarrow (A, B, D)$; $F \rightarrow (G, H)$; $D \rightarrow (E, F)$
2. $R = (A, B, C, D, E)$ didekomposisi menjadi :
 $R_1 = (A, B, C, D)$ dan $R_2 = (C, D, E)$ dengan FD :
 $A \rightarrow B$; $(C, D) \rightarrow E$; $B \rightarrow D$; $E \rightarrow A$
3. $R = (X, Y, Z, W, U, V)$ didekomposisi menjadi :
 $R_1 = (X, Y, Z, W)$ dan $R_2 = (W, U, V)$ dengan FD :
 $W \rightarrow X$; $X \rightarrow Z$
4. $R = (A, B, C, D, E, F)$ didekomposisi menjadi :
 $R_1 = (A, B, C)$, $R_2 = (A, D, F)$ dan $R_3 = (E, D)$ dengan FD :
 $A \rightarrow (B, C)$; $D \rightarrow (F, A)$

Ujilah pula dependency preservation nya untuk masing-masing soal tsb.