MODUL

DATABASE

(CCS120)

MODUL SESI VII

BAHASA QUERY

DISUSUN OLEH

NOVIANDI, S.Kom, M.Kom

UNIVERSITAS ESA UNGGUL

2020

BAB VII
Bahasa Query

Tujuan

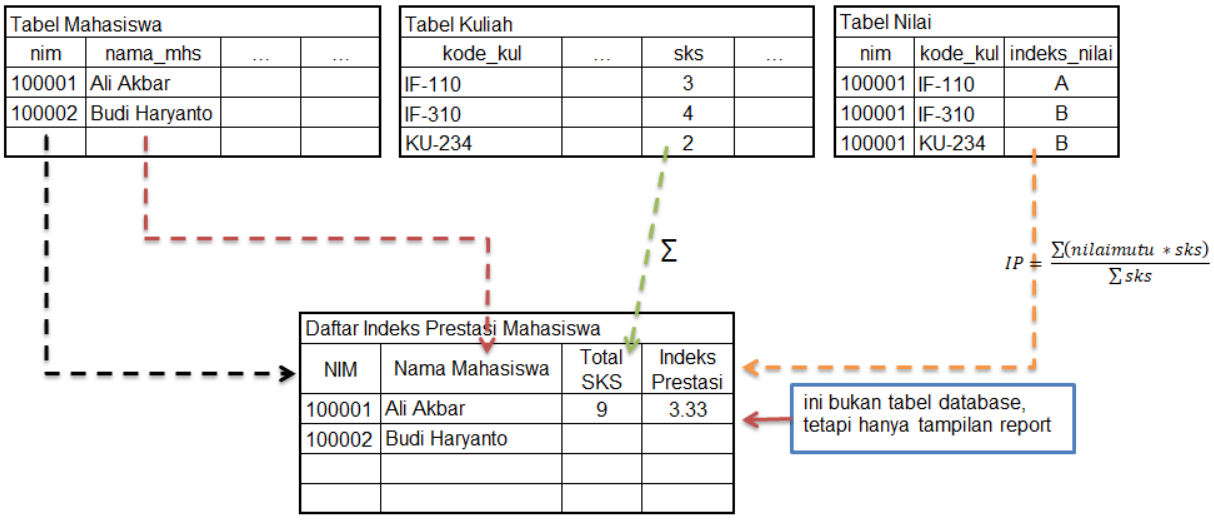
1. Menjelaskan Denormalisasi Database
2. Bahasa query berdasarkan aljabar relasional
3. Pengembangan operasi aljabar relasional
4. Bahasa query SQL (*Structure Query Language*), baik untuk proses query maupun manipulasi data.

Teori

Denormalisasi disebut juga dengan pelanggaran normalisasi. Satu-satunya alasan mengapa kita boleh melakukan denormalisasi ini adalah pertimbangan performansi. Jika performansi yang kita peroleh bisa menjadi jauh lebih baik, maka normalisasi basis data yang telah dilakukan cukup beralasan untuk dilanggar.

Bentuk-bentuk denormalisasi

1. Atribut yang terderivasi (atribut turunan)



Gambar 7.1 Denormalisasi dengan adanya Atribut Turunan

2. Atribut yang berlebihan

   Jenis-jenis atribut yang berlebihan adalah:

   a. Atribut terkodekan (*encoded attribute*)

      Atribut yang memiliki kode tambahan yang menunjukkan beberapa kondisi lainnya.

      Contoh:

      Atribut kode_kul di table kuliah yang didalamnya sudah terkandung data semester penyelenggaraan setiap mata kuliah.

   b. Atribut gabungan (*concatenated attribute*)

      Atribut dalam domain komposit

      Contoh:

      Atribut nim di table mahasiswa merupakan gabungan dari 2 dijit bilangan tahun masuk dan 4 dijit no.urut mahasiswa

   c. Atribut tumpeng tindih (*overlapping attribute*)

      Atribut dengan nilai yang tidak sepenuhnya eksklusif.

      Contoh:

      Dapat menambahkan atribut baru prog_studi di table kuliah untuk menunjukkan program studi (D3 dan S1) mana yang mengajarkan matakuliah.

   d. Atribut bermakna ganda (*alternate attribute*)

      Atribut yang memiliki arti berbeda tergantung subentitasnya (kelompok entitasnya)

      Contoh:

      Di table dosen kita dapat menambah atribut baru, yaitu gaji.

3. Table rekapitulasi (*summary table*)

   Merupakan table hasil pengolahan dari semua/sekelompok data detail yang tersimpan di table-tabel tersebut.

4. Manajemen pengecualian

Bahasa query formal

Ada 2 dasar pembentukkan dan selanjutnya menentukan cara query dalam bahasa query, yaitu:

a. Aljabar relasional
b. Kalkulus relasional

Bahasa query yang di dasarkan pada operasi-operasi dalam aljabar relasional merupakan bahasa query procedural. Sejumlah operasi dasar yang dikenal dalam aljabar relasional, yaitu:

a. SELECT
b. Project
c. Rename

Merupakan operasi tunggal (unary operation), karena hanya beroperasi pada sebuah table/relasi basis data saja.

d. Cartesian-product
e. Union
f. Set-Difference

Merupakan operasi biner (*binary operation*), karena dapat beroperasi pada sejumlah table/relasi basis data.

Operasi Seleksi (*Select*)

Digunakan untuk mengambil sejumlah baris data yang memenuhi predikat yang diberikan. Predikat mengacu pada kondisi yang ingin dipenuhi dalam operasi seleksi. Sintak yang digunakan untuk menyatakan operasi ini adalah:

$$\sigma_p(E1)$$

Dimana p adalah predikat pada atribut-atribut di E_1.

$$\sigma_{kota='Bandung'}(mahasiswa)$$

Tanda (‘) atau (“) digunakan untuk mengapit sebuah konstanta teks (string). Pada dasarnya, predikat merupakan ekspresi lojik yang menyatakan pembandingan (komparasi) antara field dalam table dengan field lain atau dengan konstanta tertentu. Untuk itu, dapat memanfaatkan operator lojik, seperti =, ≠, <, >, ≤, ≥, ^ dan v yang masing-masing menyatakan kesamaan, ketidak samaan, lebih kecil, lebih besar, lebih kecil atau sama, lebih besar atau sama, relasi **and** dan **or** .

$\sigma_{semester<3}(kuliah)$ atau $\sigma_{semester\leq 2}(kuliah)$

Operasi Projeksi (Project)

Operasi ini memungkinkan untuk menentukan field-field data dari sebuah table atau hasil query yang akan kita tampilkan. Sintaks yang digunakan untuk menyatakan operasi ini adalah:

$$\pi_s(E1)$$

Dimana S adalah list yang berisi satu atau lebih field yang ada di E_1 dan ingin ditampilkan.

Jika pada table mahasiswa kita ingin menampilkan NIM dan Nama mahasiswa untuk semua baris data yang ada di table tersebut, maka operasi ini dapat digunakan sebagai berikut:

$$\pi_{nim,nama\_mahasiswa}(mahasiswa)$$

Jika ingin menampilkan NIM dan Nama mahasiswa yang bertempat tinggal di Cimahi saja, maka operasi seleksi dan projeksi harus kita gunakan secara bersamaan seperti berikut ini:

$$\pi_{nim,nama\_mahasiswa}(\sigma_{kota='Cimahi'}(mahasiswa))$$

Operasi Cartesian Product

Operasi ini memungkinkan untuk menggabungkan data dari dua buah table atau hasil query. Symbol yang digunakan untuk menyatakan operasi ini adalah “x” dan sintaks yang digunakan untuk operasi ini adalah:

E1 x E2

Yang berakibat semua record di E1 akan dipasangkan dengan semua record di E2 dan hasil dari operasi ini akan membuat semua field yang ada di E1 dan di E2.

Jika kita hanya mengambil data dari hasil penggabungan table mahasiswa dan kuliah untuk mahasiswa yang bertempat tinggal di Cimahi dan untuk kuliah yang diselenggarakan di semester 2 saja, maka operasinya dapat ditulis sebagai berikut:

$$\sigma_{kota='Cimahi'\wedge semester=2}(mahasiswa\ x\ kuliah)$$

Jika ingin menampilkan tidak hanya sebatas NIM, tetapi juga namanya, maka harus melibatkan tiga buah table (mahasiswa, kuliah dan nilai sekaligus), karena NIM dan Nama bisa diperoleh dari table Mahasiswa, tetapi kriteria query nya hanya dapat diterapkan pada table kuliah dan nilai. Ekspresi operasinya adalah sebagai berikut:

$$\pi_{nim,namamahasiswa}(\sigma_{kuliah.kodekul-nilai.kodekul \wedge\ kuliah.kodekul='IF-221' \wedge nilai.nim=mahasiswa.nim}(kuliah\ x\ nilai\ x\ mahasiswa))$$

Operasi union

Operasi ini memungkinkan kita untuk menggabungkan data dari dua kelompok baris data (row) yang sejenis (memiliki hasil projeksi yang sama).

Contoh:

Tabel Mahasiswa

nim	nama_mhs	alamat_mhs	kota	tgl_lahir
100001	Ali Akbar	Jl. Dago Pojok 91	Bandung	02/01/1992
100002	Budi Haryanto	Jl. Pasantren	Cimahi	06/10/1991

Tabel Dosen

kode_dos	nama_dos	alamat_dos	kota
TI	Drs. Taufik Ismail	Kanayakan Baru No. 135	Bandung
AF	Ali Firdaus, MA	Jl. Tenteram No. 5	Sumedang

Maka hasil operasi union dari projeksi untuk field kota dari kedua table diatas adalah:

kota
Bandung
Cimahi
Bandung
Sumedang

- Dipakai untuk menggabungkan beberapa perintah SELECT yang menghasilkan sebuah gabungan.
- Fungsi MySQL UNION adalah Operator MySQL yang digunakan untuk menggabungkan kumpulan hasil dari 2 atau lebih pernyataan SELECT.
- Fungsi ini menghapus duplikat baris antara berbagai pernyataan SELECT. Setiap pernyataan SELECT di dalam operator UNION harus memiliki jumlah field yang sama pada set hasil dengan tipe data yang sama.

  Syntax

  SELECT expression1, expression2, ... expression_n
  FROM tables
  [WHERE conditions]
  UNION [DISTINCT]
  SELECT expression1, expression2, ... expression_n
  FROM tables
  [WHERE conditions];

Penjelasan Parameter

expression1, expression2, … expression_n

Kolom atau perhitungan yang ingin diambil kembali.

tables

Tabel yang ingin Anda ambil dari arsip. Paling tidak ada satu tabel yang tercantum dalam klausa FROM.

WHERE Conditions (Kondisi where)

Pilihan. Kondisi yang harus dipenuhi agar catatan bisa dipilih.

DISTINCT

Pilihan. Menghapus duplikat dari kumpulan hasil, namun penyertaan pengubah DISTINCT tidak memengaruhi set hasil operator UNION karena, secara default, operator UNION telah menghapus duplikat.

Catatan

- Harus ada jumlah ekspresi yang sama di kedua pernyataan SELECT.
- Karena operator UNION secara default menghapus semua baris duplikat dari kumpulan hasil, memberikan pengubah UNION DISTINCT tidak berpengaruh pada hasilnya.
- Nama kolom dari pernyataan SELECT pertama di operator UNION digunakan sebagai nama kolom untuk kumpulan hasil.

Contoh – Mengembalikan Satu Field

Berikut ini adalah contoh operator MySQL UNION yang mengembalikan satu field dari beberapa pernyataan SELECT (dan kedua field memiliki tipe data yang sama):

```
SELECT supplier_id
FROM suppliers
UNION
SELECT supplier_id
FROM order_details;
```

Dalam contoh operator UNION MySQL ini, jika supplier_id muncul di tabel pemasok dan order_details, itu akan muncul sekali dalam kumpulan hasil Anda. Operator UNI MySQL menghapus duplikat. Jika Anda tidak ingin menghapus duplikat, coba gunakan operator MySQL UNION ALL.

Contoh – Menggunakan ORDER BY

Operator UNION MySQL dapat menggunakan klausa ORDER BY untuk memesan hasil query.

Sebagai contoh:

```
SELECT supplier_id, supplier_name
FROM suppliers
WHERE supplier_id <= 500
UNION
SELECT company_id, company_name
FROM companies
WHERE company_name = 'Apple'
```

ORDER BY 2;

Dalam operator UNION MySQL ini, karena nama kolomnya berbeda antara dua pernyataan SELECT, lebih menguntungkan untuk mereferensikan kolom dalam klausa ORDER BY berdasarkan posisinya di set hasil. Dalam contoh ini, kami telah mengurutkan hasilnya dengan supplier_name / company_name dalam urutan menaik, seperti yang dilambangkan dengan ORDER BY 2.

Operasi Set-Difference

Operasi ini merupakan kebalikan dari operasi union,yaitu pengurangan data di table/hasil proyeksi pertama (E1) oleh data di table/hasil proyeksi yang kedua (E2). Symbol dari operasi ini adalah:

E1 – E2

Tabel Kuliah_S1			
kode_kul	nama_kul	sks	semester
IF-110	Struktur Data	3	1
IF-221	Pemrograman 1	3	2
IF-310	Basis Data	4	3
IF-320	Pemrograman II	3	3
IF-423	Sistem Pakar	2	4

Tabel Kuliah_D3			
kode_kul	nama_kul	sks	semester
IF-110	Struktur Data	3	1
IF-120	Aplikasi Akuntansi	2	1
IF-221	Pemrograman I	3	2
IF-310	Basis Data	4	3

Maka hasil operasi $\pi_{namakul}(kuliahS1) - \pi_{namakul}(kuliahD3)$ adalah:

nama_kul
Pemrograman II
Sistem Pakar

Operasi Rename (ρ)

- Operasi rename dibutuhkan untuk melakukan penamaan kembali pada suatu table atau hasil proyeksi agar kita dapat menunjukkan acuan yang jelas dalam sebuah operasi yang lengkap, khususnya yang melibatkan dua/lebih sumber data yang sama.
- Operasi untuk menyalin table lama ke dalam tab yang baru.
- Query: buatlah table baru dengan nama tb dari table r dimana b=5
- Aljabar relasionalnya: $\rho_{tb}(\sigma_{b=5}(r))$

Contoh:

kode_kul	nim	indeks_nilai
IF-221	100001	B
IF-221	100002	C
IF-221	100003	D

Hasil dari operasi terdalam, yaitu (nilai x ρ_n (nilai)) adalah

kode_kul	nim	indeks_nilai	kode_kul	nim	indeks_nilai
IF-221	100001	B	IF-221	100001	B
IF-221	100001	B	IF-221	100002	C
IF-221	100001	B	IF-221	100003	D
IF-221	100002	C	IF-221	100001	B
IF-221	100002	C	IF-221	100002	C
IF-221	100002	C	IF-221	100003	D
IF-221	100003	D	IF-221	100001	B
IF-221	100003	D	IF-221	100002	C
IF-221	100003	D	IF-221	100003	D

Hasil dari seleksi ($\sigma_{nilai.indeks\_nilai < n.indeks\_nilai}(nilai\ x\ \rho_n(nilai))$ adalah:

kode_kul	nim	indeks_nilai	kode_kul	nim	indeks_nilai
IF-221	100002	C	IF-221	100001	B
IF-221	100003	D	IF-221	100001	B
IF-221	100003	D	IF-221	100002	C

Projeksi untuk field indeks_nilai dari hasil seleksi diatas adalah:

indeks_nilai
C
D
D

Sementara projeksi untuk field indeks_nilai dari table Nilai adalah:

indeks_nilai
B
C
D

Maka pengurangan dari kedua projeksi tersebut adalah (yang merupakan nilai indeks_nilai tertinggi untuk mata kuliah IF-221)

indeks_nilai
B

Fungsi Agregasi

Fungsi agregasi adalah fungsi-fungsi yang melakukan pengumpulan/penggabungan nilai dari atribut-atribut (umumnya yang bertipe numerik) dari suatu query.

Ada 5 buah fungsi agregasi yang umum diketahui, yaitu:

1. Fungsi SUM, yang akan menyajikan total nilai dari suatu atribut tersebut yang terdapat dalam hasil query.
   SELECT SUM (nama field yang ingin di total) FROM (nama table)

   Contoh:
   SELECT SUM (SKS)
   FROM Mahasiswa

   Hasil menunjukkan jumlah SKS dari table maahasiswa.
2. Fungsi AVG, yang akan menyajikan nilai rata-rata dari suatu atribut tersebut yang terdapat dalam hasil query.
   Select avg (Nama_field_yang_ingin_dirata_ratakan) from nama_tabel

Contoh:

SELECT AVG (SKS) FROM Mahasiswa

Hasilnya menunjukkan rata-rata SKS dari table mahasiswa.

3. Fungsi Count, yang akan menyajikan banyaknya kemunculan suatu atribut tersebut yang terdapat dalam hasil query.

   SELECT count (*) FROM (nama tabel)

   SELECT COUNT (*) FROM MAHASISWA
   WHERE Jenis_Kelamin Like 'L'

   Hasilnya menampilkan jumlah jenis kelamin 'L' yang terdapat pada table Mahasiswa.

4. Fungsi Max, yang akan menyajikan nilai terbesar dari suatu atribut tersebut yang terdapat dalam hasil query.
   Select max(nama_field) from nama_table

   Contoh
   SELECT MAX (SKS) FROM Mahasiswa

   Hasilnya Nilai SKS maksimum dari table mahasiswa

5. Fungsi Min, yang akan menyajikan nilai terkecil dari suatu atibut tersebut yang terdapat dalam hasil query.
   Select min (nama_field) from nama_table

   Contoh:
   SELECT MIN (SKS) FROM Mahasiswa

   Hasilnya menunjukkan nilai minimum dari table Mahasiswa.

Struktur Dasar

Sebuah ekspresi SQL dasar sebenarnya hanya terdiri atas 3 klausa, yaitu: SELECT, FROM dan WHERE.

- Klausa SELECT digunakan untuk menetapkan daftar atribut (field) yang diinginkan ssebagai hasil query. Klausa ini mengacu pada operasi projeksi dalam bahasa query formal
- Klausa FROM digunakan untuk menetapkan relasi atau table (atau gabungan tabel) yang akan ditelusuri selama query data dilakukan
- Klausa WHERE yang sifatnya opsional, digunakan sebagai predikat (kriteria) yang harus dipenuhi dalam memperoleh hasil query. Klausa ini mengacu pada operasi seleksi dalam bahasa query formal.

Latihan

1. Jelaskan fase query processing.
2. Buatlah sebuah algoritma sorting yang dapat menghilangkan duplikasi data.
3. Jelaskan definisi semantic query optimization
4. Jelaskan definisi:
   a. Query decomposition
   b. Query optimization
   c. Heuristics rules
   d. Cost estimation
   e. Dynamic programming