BAB 9
ALJABAR RELASIONAL DAN KALKULUS RELASIONAL

Bahasa query formal basis data relasional adalah bahasa untuk meminta informasi dari basis data. Sebelum basis data relasional, query terhadap basis data merupakan tugas yang sangat sukar. Pemogram harus berususan dengan skema fisik internal dari basis data.

Bahasa query relasional misalkan SQL (sructured query language) berbeda dengan bahasa pemrograman konvensiona

l. Di SQL, kita menspesifikasikan properti – properti informasi yang hendak diambil tapi tidak mencantumkan rincian algoritma pengambilan. SQL adalah deklaratif, yaitu pada query dideklarasikan informasi yang merupakan jawaban yang dikehendaki bukan cara komputasi.

Query kompleks yag diperlukan pada kebanyakan aplikasi nyata memerlukan pengetahuan mendalam mengenai bahasa da semantiknya. Bahasa query relasional formal merupakan bahasa antara yang digunakan basis data, yaitu SQL dikonversi menjadi bahasa rlasional formal sehingga dapa diterapkan sekumpulan informasi untuk memperoleh query paling efisien. Terdapat dua jenis bahasa query relasional formal yang utama, yaitu:

1. Aljabar relasional.
2. Kalkulus relasional.

9.1. Aljabar relasional

Relational Algebra (aljabar relasional) merupakan kumpulan operasi terhadap relasi dimana setiap operasi menggunakan satu atau lebih relasi untuk menghasilkan satu relasi yang baru dan termasuk kategori prosedural dan juga menyediakan seperangkat operator untuk memanipulasi data.

Terdapat lima operasi dasar dalam aljabar relasional, yaitu:

1. Selection (σ)
2. Projection (π)
3. Cartesian – product (X, juga disebut sebagai cross product)
4. Union ($\cup$)
5. Set – difference (-)
6. Rename (ρ)

Operasi – operasi turunan dari operasi – operasi dasar tersebut adalah:

1. Set intersection ($\cap$)
2. Theta join (θ)
3. Natural-join ( )
4. Outer-join ($\supset\!\!\!\bowtie$)
5. Division ($\div$)

Semua operasi tersebut menghasilkan relasi baru. Bahasa disebut aljabar relasional karena bahasa berdasar sejumlah operator yang beroperasi pada relasi – relasi (tabel – tabel). Masing – masing operator beroperasi pada satu relasi atau lebih atau menghasilkan relasi – relasi lain sebagai hasil. Query adalah sekedar ekspresi yang melibatkan operator – operator itu. Hasil ekspresi adalah relasi yaitu jawaban terhadap query.

SQL adalah bahasa yang deklaratif yang berarti tidak menspesifikasikan algoritma yang digunakan untuk pengolahan query. Ekspresi relasional dapat dipandang sebagai spesifikasi algoritma (meskipun dalam level lebih tinggi dibanding algoritma untuk bahasa pemrograman konvensional).

Pemrogram menggunakan query SQL, DBMS menggunakan aljabar relasional sebagai bahasa antara dalam spesifikasi algoritma query. Langkah – langkah dalam DBMS untuk pengolahan query adalah:

1. DBMS melakukan parsing terhadap string dari query SQL dan menerjemahkannya menjadi ekspresi aljabar relasional yang dapat menuntun kedalam algoritma sederhana yang tidak efisien.
2. Setelah itu, bagian query optimizer mengkonversi ekspresi aljabar relasional ini menjadi ekspresi lain yang ekvivalen nemun lebih efisien untuk dieksekusi.
3. Berdasarkan ekpresi aljabar relasional yang telah dioptimasi, query optimizer mempesiapkan rencana eksekusi query (query execution plan) yang kemudian ditransformasikan menjadi kode yang dapat dieksekusi pembangkit kode di DBMS.
4. Karena ekspresi aljabar mempunyai semantik matematika yang presisi maka sistem dapat memferifikasi ekvivalensi ekspresi yang dioptimasi yang dihasilkan dari manipulasi ekpresi asal. Semantiks ini juga memungkinkan pembandingan rencana – rencana evaluasi query yang berbeda.

Aljabar relasional merupakan kunci pemahaman kerja internal DBMS relasional, pemahaman aljabar relasional merupakan hal yang esensi dlam merancang query SQL yang diolah secara efisien. Aljabar relasional banyak digunakan pada optimasi query dan

pengolahan query tersebar. Aljabar relasional mendefinisikan sekumpulan operator dan rumus untuk memanipulasi himpunan data.

9.1.1. Operasi aljabar relasional

Untuk mengimplementasikan kedalam operasi aljabar relasional, berikut ini diberikan relasi – relasi dari database Penjadwalan_mengajar_dosen pada STMIK 'Revanda Jaya' Bekasi. Relasi – relasi tersebut meliputi:

1. Dosen

   Dosen ={**nid**,nama_d,tempat_lhr,tgl_lahir,jkelamin,alamat,kota,kodepos,gpokok}

   Primary key relasi Dosen adalah nid, karena tidak ada seorang dosen yang nid (nomor induk dosen) sama dengan dosen yang lainnya.

   Dengan data value sebagai berikut:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
95001	Bambang Sutedjo,Ir,MMSi	Jakarta	2/23/58	Pria	Jl. Mawar 6 No.2	Jakarta Selatan	23234	1300000
95002	Asri Kasetyaningsih,M.Kom	Semarang	12/25/62	Wanita	Jl. Perjuangan 3 No.11	Bekasi Timur	54567	1200000
96001	Triyatno,Ir,MM,M.Kom	Bekasi	5/14/67	Pria	Jl. Mawar Indah 1 No.1	Bekasi Barat	54356	1100000
96002	Diastuti Pujiningsih,MM,MT	Surabaya	12/24/70	Wanita	Jl. Janur Kuning 2 No.56	Cibitung	54566	1000000
97001	Endang Junianti,Ir,MMSi	Bandung	5/20/67	Wanita	Jl. Kemukus 2 No.56	Cikarang	56555	1150000
97002	Djoko Pamungkas,M.Kom	Bogor	5/28/71	Pria	Jl. Anggrek 7 No.1	Bekasi Timur	45666	1100000
98001	Didik Atmadja,Ir,MMSi	Bandung	4/20/70	Pria	Jl. Cipete Raya No.6	Jakarta Selatan	24123	1250000
98002	Bagus Windarjo,M.Kom.	Tangerang	12/13/71	Pria	Jl. H. Ali No.21	Cibitung	54523	1150000
99001	Dewi Anjani,Ir,MM	Bekasi	12/14/75	Wanita	Jl.Kemang Raya 2 No.1	Bekasi Barat	54567	1200000
00001	Riswoko Sasono,MMSi	Bogor	12/24/74	Pria	Jl. Catur 3 No.23	Bekasi Utara	52123	1300000
00002	Hasta Riyanti,Ir,MM	Jakarta	5/27/69	Wanita	Jl. Delima 2 No.2	Bekasi Timur	45612	1150000
01001	Cokro Diningrat,Ir,MT	Bekasi	12/19/66	Pria	Jl. Dayang Raya 2 No.12	Cibitung	54456	1200000
01002	Sakib Aljaber,MT	Cikarang	5/20/70	Pria	Jl. Kemuning 1 No.1	Cikarang	56234	1150000

2. Matakuliah

   Matakuliah ={**kdmk**,nama_mk,sks,semester}

   Primary key relasi Matakuliah adalah kdmk, karena tidak ada kode suatu matakuliah yang kdmk (kode matakuliah) sama dengan matakuliah yang lainnya.

   Dengan data value sebagai berikut:

kdmk	nama_mk	sks	semester
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1
MKB331203	PRAKTIKUM PAKET PROGRAM APLIKASI I	1	1
MKK231201	DASAR MANAJEMEN DAN BISNIS I	2	1
MKK231203	MATEMATIKA DISKRIT I	2	1
MKK231205	AKUTANSI DASAR I	2	1
MKK231213	PENGANTAR EKONOMI	2	1
MPK131201	PENDIDIKAN AGAMA I	2	1
MPK131203	PENDIDIKAN PANCASILA	2	1
MPK131205	BAHASA INGGRIS I	2	1
MPK231207	SISTEM BASIS DATA	2	1

3. Jurusan

   Jurusan ={**kode_jur**,nama_jur,sjenjang,nama_kajur}

   Primary key relasi Jurusan adalah kode_jur, karena tidak ada kode_jur (kode jurusan) yang sama dengan jurusan yang lainnya.

   Dengan data value sebagai berikut:

kode_jur	nama_jur	jenjang	nama_kajur
KA	Komputerisasi Akuntansi	Diploma 3	Rini Wulandari,MM,MMSi
MI	Manajemen Informatika	Diploma 3	Wahono Diprodjo, MM, MKom.
SI	Sistem Informasi	Strata 1	Bagus Hermansyah, MM.Si,M.Kom.
TI	Teknik Informatika	Strata 1	Fadjar Sasongko, MT, M.Kom.
TK	Teknik Komputer	Diploma 3	Agus Budiyantara, T, MT.

4. Mengajar

   Mengajar ={**nid,thn_akademik,smt,hari,jam_ke**,kdmk,waktu,kelas,kode_jur}

   Primary key relasi Matakuliah adalah nid,thn_akademik,smt,hari,jam_ke, kondisi :

   - Bila primary key nid,thn_akademik, maka tidak bisa karena seorang dosen pada tahun akademik yang sama bisa mengajar lebih dari satu matakuliah.
   - Bila primary key nid,thn_akademik,smt, maka tidak bisa karena seorang dosen pada tahun akademik dan semester yang sama bisa mengajar lebih dari satu matakuliah.
   - Bila primary key nid,thn_akademik,smt,hari, maka tidak bisa karena seorang dosen pada tahun akademik, semester dan hari yang sama bisa mengajar lebih dari satu matakuliah.
   - Sehingga primary key pada relasi mengajar adalah nid,thn_akademik,smt,hari,jam_ke, maka unik dan dijadikan primary key pada relasi tersebut.

Dengan data value sebagai berikut:

nid	thn_akademik	smt	hari	jam_ke	kdmk	waktu	kelas	kode_jur
00001	2004	1	Rabu	1	MKB331205	8:00	T202	TI
00001	2004	1	Senin	1	MKB331201	8:00	M101	MI
00002	2004	2	Jumat	1	MPK131204	8:00	S201	SI
00002	2004	2	Jumat	2	MKP131204	10:00	S202	SI
95001	2004	1	Kamis	1	MKB331201	8:00	T101	TI
95001	2004	1	Senin	1	MKB331201	8:00	M101	MI
98002	2004	2	Rabu	1	MKB331204	8:00	S201	SI
98002	2004	2	Selasa	2	MPK131204	10:00	M201	MI
99001	2004	1	Senin	1	MKB331201	8:00	M102	MI
99001	2004	2	Selasa	1	MKB331205	8:00	T201	TI

9.1.2. Operasi Selection (σ)

Selection / Select (σ), adalah operasi untuk menyeleksi tupel – tupel yang memenuhi suatu predikat, kita dapat menggunakan operator perbandingan (<,>,>=,<=,=,#) pada predikat. Beberapa predikat dapat dikombinasikan menjadi predikat manjemuk menggunakan penghubung AND ($\wedge$) dan OR ($\vee$).

Contoh operasi Select:

a. Contoh 1

1. Query : Tampilkan daftar dosen yang tempat lahirnya di 'Bekasi'.
2. Aljabar relasional:

   σtempat_lhr='Bekasi' (Dosen)

3. Hasilnya adalah:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
96001	Triyatno,Ir,MM,M.Kom	Bekasi	5/14/67	Pria	Jl. Mawar Indah 1 No.1	Bekasi Barat	54356	1100000
99001	Dewi Anjani,Ir,MM	Bekasi	12/14/75	Wanita	Jl.Kemang Raya 2 No.1	Bekasi Barat	54567	1200000
01001	Cokro Diningrat,Ir,MT	Bekasi	12/19/66	Pria	Jl. Dayang Raya 2 No.12	Cibitung	54456	1200000

b. Contoh 2

1. Query : Tampilkan daftar dosen yang tempat lahirnya di 'Jakarta' atau 'Bogor'
2. Aljabar relasional:

   σtempat_lhr='Jakarta' $\vee$ tempat_lhr='Bogor' (Dosen)

3. Hasilnya adalah:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
95001	Bambang Sutedjo,Ir,MMSi	Jakarta	2/23/58	Pria	Jl. Mawar 6 No.2	Jakarta Selatan	23234	1300000
97002	Djoko Pamungkas,M.Kom	Bogor	5/28/71	Pria	Jl. Anggrek 7 No.1	Bekasi Timur	45666	1100000
00001	Riswoko Sasono,MMSi	Bogor	12/24/74	Pria	Jl. Catur 3 No.23	Bekasi Utara	52123	1300000
00002	Hasta Riyanti,Ir,MM	Jakarta	5/27/69	Wanita	Jl. Delima 2 No.2	Bekasi Timur	45612	1150000

c. Contoh 3

1. Query : Tampilkan daftar dosen yang tempat lahirnya di 'Bogor' dan jenis kelaminnya 'Pria'
2. Aljabar relasional:

σtempat_lhr='Bogor' $\wedge$ jkelamin='Pria' (Dosen)

3. Hasilnya adalah:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
97002	Djoko Pamungkas,M.Kom	Bogor	5/28/71	Pria	Jl. Anggrek 7 No.1	Bekasi Timur	45666	1100000
00001	Riswoko Sasono,MMSi	Bogor	12/24/74	Pria	Jl. Catur 3 No.23	Bekasi Utara	52123	1300000

9.1.3. Operasi Projection (π)

Projection / Project (π), adalah operasi untuk memperoleh kolom – kolom tertentu. Operasi project adalah operasi unary yang mengirim relasi argumen dengan kolom – kolom tertentu. Karena relasi adalah himpunan, maka baris – baris duplikasi dihilangkan.

Sintaks yang digunakan dalam operasi proyeksi ini adalah sebagai berikut :

π colum1,…,column (tabel)

Contoh operasi Project:

a. Contoh 1

1. Query : Tampilkan nid,nama_d,alamat,kota dari relasi Dosen
2. Aljabar relasional:

π nid,nama_d,alamat,kota(Dosen)

3. Hasilnya adalah:

nid	nama_d	alamat	kota
00001	Riswoko Sasono,MMSi	Jl. Catur 3 No.23	Bekasi Utara
00002	Hasta Riyanti,Ir,MM	Jl. Delima 2 No.2	Bekasi Timur
01001	Cokro Diningrat,Ir,MT	Jl. Dayang Raya 2 No.12	Cibitung
01002	Sakib Aljaber,MT	Jl. Kemuning 1 No.1	Cikarang
95001	Bambang Sutedjo,Ir,MMSi	Jl. Mawar 6 No.2	Jakarta Selatan
95002	Asri Kasetyaningsih,M.Kom	Jl. Perjuangan 3 No.11	Bekasi Timur
96001	Triyatno,Ir,MM,M.Kom	Jl. Mawar Indah 1 No.1	Bekasi Barat
96002	Diastuti Pujiningsih,MM,MT	Jl. Janur Kuning 2 No.56	Cibitung
97001	Endang Junianti,Ir,MMSi	Jl. Kemukus 2 No.56	Cikarang
97002	Djoko Pamungkas,M.Kom	Jl. Anggrek 7 No.1	Bekasi Timur

b. Contoh 2

1. Query : Tampilkan nid,nama_d,alamat,kota,gajipokok dari relasi Dosen, dimana gaji pokoknya lebih besar dari Rp.1200000
2. Aljabar relasional:

   π nid,nama_d,alamat,kota,gajipokok(σgajipokok>1200000 (Dosen))

3. Hasilnya adalah:

nid	nama_d	alamat	kota	gajipokok
00001	Riswoko Sasono,MMSi	Jl. Catur 3 No.23	Bekasi Utara	1300000
95001	Bambang Sutedjo,Ir,MMSi	Jl. Mawar 6 No.2	Jakarta Selatan	1300000
98001	Didik Atmadja,Ir,MMSi	Jl. Cipete Raya No.6	Jakarta Selatan	1250000

c. Contoh 3

4. Query : Tampilkan nid,nama_d,alamat,kota,gajipokok dari relasi Dosen, dimana kota alamatnya 'Cibitung' dan gaji pokoknya lebih besar dari Rp.1000000
5. Aljabar relasional:

   π nid,nama_d,alamat,kota,gajipokok(σkota='Cibitung' $\wedge$ gajipokok>1000000 (Dosen))

6. Hasilnya adalah:

nid	nama_d	alamat	kota	gajipokok
01001	Cokro Diningrat,Ir,MT	Jl. Dayang Raya 2 No.12	Cibitung	1200000
98002	Bagus Windarjo,M.Kom.	Jl. H. Ali No.21	Cibitung	1150000

9.1.4. Operasi Cartesian-product (X)

Cartesian-product (X), adalah operasi untuk menghasilkan table hasil perkalian kartesian. Sintaks yang digunakan dalam operasi proyeksi ini adalah sebagai berikut :

$$R \times S = \{(x,y) \mid x \in R \text{ dan } y \in S\}$$

Operasi cartesian-product memungkinkan kita mengkombinasikan informasi beberapa relasi, operasi ini adalah operasi biner. Sebagaimana telah dinyatakan bahwa relasi adalah subset hasil cartesian-product dan himpunan domain relasi – relasi tersebut. Kita harus memilih atribut – atribut untuk relasi yang dihasilkan dari cartesian-product.

Contoh operasi Cartesian-product:

a. Contoh 1

1. Query : Tampilkan nid,nama_d (dari relasi Dosen), nama_mk (dari relasi Matakuliah), thn_akademik,smt,hari,jam_ke,waktu,kelas (dari relasi Mengajar) dimana semester mengajar adalah pada semester '1'.
2. Aljabar relasional:

   π nid,nama_d,nama_mk, thn_akademik,smt,hari,jam_ke,waktu,kelas (σ smt=1 $\wedge$ Dosen.nid=Mengajar.nid $\wedge$ Mengjar.kdmk=Matakuliah.kdmk$^{(DosenxMatakuliahxMengajar)}$)

   atau:

   σ (Mengajar.nid=Dosen.nid $\wedge$ Mengajar.kdmk=Matakuliah.kdmk) $\wedge$ smt=1 (((π nid,nama_d $^{(Dosen)}$) X (π nama_mk $^{(Matakuliah)}$) X (π thn_akademik,smt,hari,jam_ke,waktu,kelas $^{(Mengajar)}$))))

3. Hasilnya adalah:

nid	nama_d	nama_mk	thn_akademik	smt	hari	jam_ke	waktu	kelas
00001	Riswoko Sasono,MMSi	PERANCANGAN BASIS DATA	2004	1	Rabu	1	8:00	T202
00001	Riswoko Sasono,MMSi	PENGANTAR TEKNOLOGI INFORMASI	2004	1	Senin	1	8:00	M101
95001	Bambang Sutedjo,Ir,MMSi	PENGANTAR TEKNOLOGI INFORMASI	2004	1	Kamis	1	8:00	T101
95001	Bambang Sutedjo,Ir,MMSi	PENGANTAR TEKNOLOGI INFORMASI	2004	1	Senin	1	8:00	M101
99001	Dewi Anjani,Ir,MM	PENGANTAR TEKNOLOGI INFORMASI	2004	1	Senin	1	8:00	M102

b. Contoh 2

1. Query : Tampilkan nama_d (dari relasi Dosen), nama_mk,sks (dari relasi Matakuliah), hari,jam_ke,waktu (dari relasi Mengajar) dimana sks matakuliah >3 atau hari mengajar = 'Jumat'.

2. Aljabar relasional:

   π nama_d,nama_mk,sks,hari,jam_ke,waktu (σ sks>3 $\vee$ hari='Jumat' $\wedge$ Mengajar.nid=Dosen.nid $\wedge$ Mengajar.kdmk=Matakuliah.kdmk$^{(MengajarxDosenxMatakuliah)}$)

   atau:

   σ (Mengajar.nid=Dosen.nid $\wedge$ Mengajar.kdmk=Matakuliah.kdmk) $\wedge$ (sks>3 $\vee$ hari='Jumat' (((π nama_d $^{(Dosen)}$) X (π nama_mk,sks $^{(Matakuliah)}$) X (π hari,jam_ke,waktu $^{(Mengajar)}$))))

3. Hasilnya adalah:

nama_d	nama_mk	sks	hari	jam_ke	waktu
Riswoko Sasono,MMSi	PERANCANGAN BASIS DATA	4	Rabu	1	8:00
Hasta Riyanti,Ir,MM	PENDIDIKAN KEWARGANEGARAAN	2	Jumat	1	8:00
Hasta Riyanti,Ir,MM	PENDIDIKAN KEWARGANEGARAAN	2	Jumat	2	10:00
Bagus Windarjo,M.Kom.	PERANCANGAN BASIS DATA	4	Rabu	1	8:00
Dewi Anjani,Ir,MM	PERANCANGAN BASIS DATA	4	Selasa	1	8:00

b. Contoh 3

1. Query : Tampilkan kdmk,nama_mk,sks (dari relasi Matakuliah), smt,hari,jam_ke,waktu (dari relasi Mengajar) dimana semester (smt) yang diajar dosen pada semester '1' dan jam_ke '1'
2. Aljabar relasional:

   π kdmk,nama_mk,sks,smt,hari,jam_ke,waktu (σ smt=1 $\wedge$ jam_ke='1' $\wedge$ Mengajar.kdmk=Matakuliah.kdmk$^{(MengajarxMatakuliah)}$)

   atau:

   σ Mengajar.kdmk=Matakuliah.kdmk $\wedge$ smt=1 $\wedge$ jam_ke='1' ((π kdmk,nama_mk,sks $^{(Matakuliah)}$) X (π hari,jam_ke,waktu $^{(Mengajar)}$))

3. Hasilnya adalah:

kdmk	nama_mk	sks	smt	hari	jam_ke	waktu
MKB331205	PERANCANGAN BASIS DATA	4	1	Rabu	1	8:00
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	Senin	1	8:00
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	Kamis	1	8:00
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	Senin	1	8:00
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	Senin	1	8:00

9.1.5. Operasi Union ($\cup$)

Union ($\cup$), adalah operasi untuk menghasilkan gabungan table degan syarat kedua table memiliki atribut yangsama, yaitu domain atribut ke-i masing – masing table harus sama. Sintaks yang digunakan dalam operasi union ini adalah sebagai berikut :

$$R \cup S = \{x \mid x \in R \text{ atau } X \in S\}$$

Operasi ini dapat dilaksanakan apabila R dan S mempunyai atribut yang sama sehingga jumlah komponennya sama.

R

A	**B**
D	A
C	F
G	H

S

A	**B**
D	A
H	T
G	H

R $\cup$ S adalah:

A	**B**
D	A
C	F
G	H
H	T

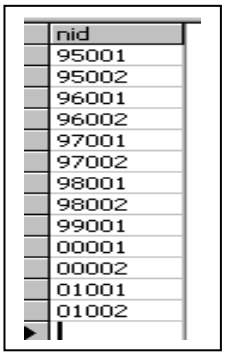
Contoh operasi Union:

a. Contoh 1

1. Query : Tampilkan nid (dari relasi Dosen) Union dari nid (dari relasi Mengajar) .
2. Aljabar relasional:

   π nid (Dosen) $\cup$ nid (Mengajar)

3. Hasilnya adalah:



nid
95001
95002
96001
96002
97001
97002
98001
98002
99001
00001
00002
01001
01002

9.1.6. Set-difference (---)

Set-difference (--), adalah operasi untuk mendapatkan table pada suatu relasi, tapi tidak ada pada relasi yang lainnya. Sintaks yang digunakan dalam operasi union ini adalah sebagai berikut :

$$R - S = \{ x \mid x \in R \text{ dan } X \notin S\}$$

Operasi ini dapat dilaksanakan apabila R dan S mempunyai atribut yang tidak sama yang akan ditampilkan, artinya adalah atribut R yang tidak ada di S akan ditampilkan, sedangkan atribut yang sama tidak ditampilkan.

a. Contoh 1

1. Query : Tampilkan nid (dari relasi Dosen) Set-difference dari nid (dari relasi Mengajar).
2. Aljabar relasional:

   π nid $^{(Dosen)}$ -- nid $^{(Mengajar)}$

3. Hasilnya adalah:

nid
95002
96001
96002
97001
97002
98001
01001
01002

9.1.7. Rename (ρ)

Rename (ρ), adalah operasi untuk menyalin table lama kedalam table yang baru. Sintaks yang digunakan dalam operasi union ini adalah sebagai berikut :

ρ [nama_table] $^{(table_lama)}$

a. Contoh 1

1. Query : Salinlah table baru dengan nama DosenNew dari table Dosen, dimana jenis kelaminnya adalah 'Pria'.
2. Aljabar relasional:

   ρ DosenNew (σ jkelamin='Pria') $^{(Dosen)}$)

3. Hasilnya adalah:

nid	nama_d	tempat_lhr	tgl_lahir	jkelamin	alamat	kota	kodepos	gajipokok
00001	Riswoko Sasono,MMSi	Bogor	12/24/74	Pria	Jl. Catur 3 No.23	Bekasi Utara	52123	1300000
01001	Cokro Diningrat,Ir,MT	Bekasi	12/19/66	Pria	Jl. Dayang Raya 2 No.12	Cibitung	54456	1200000
01002	Sakib Aljaber,MT	Cikarang	5/20/70	Pria	Jl. Kemuning 1 No.1	Cikarang	56234	1150000
95001	Bambang Sutedjo,Ir,MMSi	Jakarta	2/23/58	Pria	Jl. Mawar 6 No.2	Jakarta Selatan	23234	1300000
96001	Triyatno,Ir,MM,M.Kom	Bekasi	5/14/67	Pria	Jl. Mawar Indah 1 No.1	Bekasi Barat	54356	1100000
97002	Djoko Pamungkas,M.Kom	Bogor	5/28/71	Pria	Jl. Anggrek 7 No.1	Bekasi Timur	45666	1100000
98001	Didik Atmadja,Ir,MMSi	Bandung	4/20/70	Pria	Jl. Cipete Raya No.6	Jakarta Selatan	24123	1250000
98002	Bagus Windarjo,M.Kom.	Tangerang	12/13/71	Pria	Jl. H. Ali No.21	Cibitung	54523	1150000

Untuk operasi rename ini hasil dari perintah tersebut adalah membentuk table baru dengan nama DosenNew beserta datanya dimana jenis kelaminnya adalah 'Pria'

9.1.8. Set-intersection ($\cap$)

Set-intersection / Intersection ($\cap$) termasuk kedalam operator tambahan, karena operator ini dapat diderivikasi dari operator dasar seperti berikut:

$A \cap B = A - (A - B)$, atau $A \cap B = B - (B - A)$

Operasi ini merupakan operasi binary, yang digunakan untuk membentuk sebuah relasi baru dengan tuple yang berasal dari kedua relasi yang dihubungkan, misalkan:

R1

X	**Y**
A	C
B	F

R2

X	**Y**
D	F
A	C
H	I

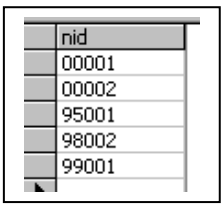
R1 $\cap$ R2 adalah:

A	**B**
A	C

a. Contoh 1

1. Query : Tampilkan nid (dari relasi Dosen) Set-intersection dari nid (dari relasi Mengajar).
2. Aljabar relasional:

   π nid $^{(Dosen)}$ $\cap$ nid $^{(Mengajar)}$



nid
00001
00002
95001
98002
99001

3. Hasilnya adalah:

9.1.9. Theta-join (θ) / Equi-join ($\bowtie$)

Theta-join ($\bowtie$) dan equi-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria

a. Contoh 1

1. Query : Tampilkan seluruh data yang ada pada relasi Matakuliah dan relasi Mengajar
2. Aljabar relasional:

Matakuliah $\bowtie$ Mengajar.kdmk=Matakuliah.kdmk Mengajar

3. Hasilnya adalah:

kdmk	nama_mk	sks	semester	nid	thn_aka	smt	hari	jam_ke	kdmk	waktu	kelas	kode_jur
MKB331205	PERANCANGAN BASIS DATA	4	2	00001	2004	1	Rabu	1	MKB331205	8:00	T202	TI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	00001	2004	1	Senin	1	MKB331201	8:00	M101	MI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	00002	2004	2	Jumat	1	MPK131204	8:00	S201	SI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	00002	2004	2	Jumat	2	MPK131204	10:00	S202	SI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	95001	2004	1	Kamis	1	MKB331201	8:00	T101	TI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	95001	2004	1	Senin	1	MKB331201	8:00	M101	MI
MKB331205	PERANCANGAN BASIS DATA	4	2	98002	2004	2	Rabu	1	MKB331205	8:00	S201	SI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	98002	2004	2	Selasa	2	MPK131204	10:00	M201	MI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	99001	2004	1	Senin	1	MKB331201	8:00	M102	MI
MKB331205	PERANCANGAN BASIS DATA	4	2	99001	2004	2	Selasa	1	MKB331205	8:00	T201	TI

9.1.10. Natural-join ($\bowtie$)

Natural-join ($\bowtie$) sama seperti operasi equi-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria pada kolom yang sama.

a. Contoh 1

1. Query : Tampilkan seluruh data yang ada pada relasi Matakuliah dan relasi Mengajar
2. Aljabar relasional:

Matakuliah $\bowtie$ Mengajar.kdmk=Matakuliah.kdmk Mengajar

3. Hasilnya adalah:

kdmk	nama_mk	sks	semester	nid	thn_akademik	smt	hari	jam_ke	waktu	kelas	kode_jur
MKB331205	PERANCANGAN BASIS DATA	4	2	00001	2004	1	Rabu	1	8:00	T202	TI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	00001	2004	1	Senin	1	8:00	M101	MI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	00002	2004	2	Jumat	1	8:00	S201	SI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	00002	2004	2	Jumat	2	10:00	S202	SI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	95001	2004	1	Kamis	1	8:00	T101	TI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	95001	2004	1	Senin	1	8:00	M101	MI
MKB331205	PERANCANGAN BASIS DATA	4	2	98002	2004	2	Rabu	1	8:00	S201	SI
MPK131204	PENDIDIKAN KEWARGANEGARAAN	2	2	98002	2004	2	Selasa	2	10:00	M201	MI
MKB331201	PENGANTAR TEKNOLOGI INFORMASI	3	1	99001	2004	1	Senin	1	8:00	M102	MI
MKB331205	PERANCANGAN BASIS DATA	4	2	99001	2004	2	Selasa	1	8:00	T201	TI

9.1.11. Outer-join ($\supset\!\bowtie$)

Outer-join adalah operasi untuk menggabungkan operasi selection dan cartesian-product dengan suatu kriteria pada kolom yang sama.

a. Contoh 1

1. Query : Tampilkan nid_nama_d (dari relasi Dosen) dan thn_akademik,smt,hari,jam_ke,waktu (dari relasi Mengajar) dengan outer join, artinya adalah pada kolom nid,nama_d pada relasi Dosen akan ditampilkan walaupun dosen tersebut tidak melakukan transaksi mengajar.
2. Aljabar relasional:

π nid,nama_d (Dosen) $\supset\!\bowtie$ π thn_akademik,smt,hari,jam_ke,waktu (Mengajar)

3. Hasilnya adalah:

nid	nama_d	thn_akademik	smt	hari	jam_ke	waktu
00001	Riswoko Sasono,MMSi	2004	1	Rabu	1	8:00
00001	Riswoko Sasono,MMSi	2004	1	Senin	1	8:00
00002	Hasta Riyanti,Ir,MM	2004	2	Jumat	1	8:00
00002	Hasta Riyanti,Ir,MM	2004	2	Jumat	2	10:00
01001	Cokro Diningrat,Ir,MT	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
01002	Sakib Aljaber,MT	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
95001	Bambang Sutedjo,Ir,MMSi	2004	1	Kamis	1	8:00
95001	Bambang Sutedjo,Ir,MMSi	2004	1	Senin	1	8:00
95002	Asri Kasetyaningsih,M.Kom	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
96001	Triyatno,Ir,MM,M.Kom	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
96002	Diastuti Pujiningsih,MM,MT	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
97001	Endang Junianti,Ir,MMSi	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
97002	Djoko Pamungkas,M.Kom	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
98001	Didik Atmadja,Ir,MMSi	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
98002	Bagus Windarjo,M.Kom.	2004	2	Rabu	1	8:00
98002	Bagus Windarjo,M.Kom.	2004	2	Selasa	2	10:00
99001	Dewi Anjani,Ir,MM	2004	1	Senin	1	8:00
99001	Dewi Anjani,Ir,MM	2004	2	Selasa	1	8:00

9.1.12 Devision (÷)

Devision (÷) adalah operasi yang banyak digunakan dlam query yang mencakup frase “setiap” atau “untuk semua”, operasi ini juga merupakan pembagian atas tuple – tuple dari dua relasi.

a. Contoh 1

1. Query : Tampilkan nid,hari, waktu (dari relasi Mengajar) dan nid (dari relasi Dosen) dimana dosen yang jenis kelaminnya ‘Pria’ dan lakukan devision pada kedua relasi tersebut.

2. Aljabar relasional:

π nid,hari,waktu $^{(Mengajar)}$) $\div$ (π nid (σ jkelamin='Pria' $^{(Dosen)}$))

4. Hasilnya adalah:

π nid,hari,waktu $^{(Mengajar)}$ π nid (σ gajipokok>1300000 $^{(Dosen)}$)

nid	hari	waktu
00001	Rabu	8:00
00001	Senin	8:00
00002	Jumat	8:00
00002	Jumat	10:00
95001	Kamis	8:00
95001	Senin	8:00
98002	Rabu	8:00
98002	Selasa	10:00
99001	Senin	8:00
99001	Selasa	8:00

nid
00001
95001

Hasil akhir adalah:

nid
98002
99001
99001

9.2. Kalkulus relasional

Pemakai mendiskripsikan informasi yang dikehendaki tanpa memberikan prosedur (deret operasi) spesifik untuk memperoleh informasi. Pada model relasional, bahasa formal non prosedural adalah bahasa kalkulus (predikat(relasional yaitu diekspresikan dengan menspesifikasikan predikat terhadap tuple atau domain yang harus dipenuhi. Kalkulus relasional dibagi menjadi 2 (dua) yaitu:

1. Kalkulus relasional tupel (tuple relational calculus).
2. Kalkulus relasional domain (domain relational calculus).

9.2.1. Kalkulus Relasional Tupel

Kalkulus relasional tupel mendiskripsikan informasi tanpa perlu memberikan prosedur / cara spesifik untuk memperoleh informasi tersebut. Konsep dasar kalkulus relasional tupel adalah konsep

variable tupel. Variable ini merepresentasikan tupel – tupel pada relasi dan digunakan untuk mengekstrak data dari relasi. Komponen - komponen lain rumus kalkulus tupel adalah kualifikasi data dengan membatasi nilai – nilai dari atribut – atribut yang dispesifikasikan.

Query pada kalkulus relasional tupel dapat diekspresikan dengan:

$\{ t \mid P(t) \}$

yaitu himpunan semua tupel t sehingga predikat P bernilai True untuk t, notasi t[A] untuk menunjukkan nilai tuple t pada atribut A. dan menggunakan $t \in r$ untuk menunjukkan nilai tupel t di relasi r. predikat P adalah berupa rumus, beberapa variable tupel dapat muncul di rumus. Variable tupel dikatakan variable bebas kecuali dikuantifikasi $\exists$ atau $\forall$.

Maka:

$r \in R \wedge \exists s \in S\ (r[a]=S[a])$

t adalah variable bebas, variable tupel s dikatakan sebagai variable terikat.

Berikut ini contoh kalkulus relasional tupel yang diterapkan pada SQL:

```
SELECT Dosen.nid,Dosen.nama_d,Dosen.gajipokok
FROM Dosen
WHERE
        Dosen.jkelamin='Pria' AND Dosen.gajipokok>1000000
```

Pada query diatas sebenarnya menyatakan dua hal yaitu:

1. Kita berkehandak mengekstrak tupel – tupel pada relasi Dosen yang mempunyai atribut jkelamin adalah 'Pria' dan memiliki atribut gajipokok adalah lebih besar dari 1000000.
2. Dari tupel – tupel ini, kita berkehendak menampilkan atribut tertentu yaitu nid,nama_d,gajipokok.

Dengan demikian atribut Dosen.nid,Dosen.nama_d,Dosen.gajipokok adalah variable – variable tupel. Bentuk umum dari kalkulus relasional tupel adalah:

TupleVariable1 operator [TupleVariable2 | constant]

Dimana :

TupleVariablen adalah variable tupel dimana i=1 sampai n variable tupel.

Operator adalah +,>,<,>=,<=,<>

Constant adalah sembarang nilai numerik atau string.

Konstrain yag berlaku adalah variable – variable tupel dan konstanta harus mempunyai domain yang sama antara bagian kiri operator dan bagian kanan operator.

Rumus dapat dikoneksikan operator boolean AND, OR, dan NOT sehingga bentuk umumnya adalah:

TupleVariable1 operator [TupleVariable2 | constant3]
AND
TupleVariable4 operator [TupleVariable5 | constant6]

AND
.
.
OR
TupleVariablem operator [TupleVariablen | constantp]

Bentukan tersebut digunakan SQL pada klausa SELECT serta WHERE. Dengan demikian dapat disimpulkan bahwa SQL juga dikembangkan berbasis pada kalkulus relasional tupel.

Rumus kalkulus relasional tupel dibangun dari atom – atom, atom mempunyai salah satu bentuk berikut:

- s $\in$ R, dimana s adalah variable tupel dan R adalah relasi, kita tidak mengijinkan operasi $\notin$.
- S[x] u[y], dimana s dan u adalah variable tupel, x adalah atribut yang didefinisikan di s, y adalah atribut didefinisikan di u. Θ adalah operator perbandingan (<, <=,=, >=, $\neq$). Kita perlu atribut x dan y yang mempunyai domain – domain yang anggotanya dapat dibandingkan dengan Θ.
- s[x] Θ c, dimana s adalah variable tupel , x adalah atribut yang didefinisikan di s. Θ adalah operator pembanding dan c adalah konstanta dari domain atribut x.

Rumus dapat dibangun menggunakan aturan – aturan berikut:

- atom adalah rumus.

- Jika P1 adalah rumus, maka $\neg$P1 dan (P1)
- Jika P1 ddddan P2 adalah rumus, maka P1 $\vee$ P2, P1 $\wedge$ P2, dan P1 $\Rightarrow$ P2.
- Jika P1(x) adalah rumus di x, dimana x adalah vaariable tupel x, maka r $\in$ R $\wedge$ $\exists$s $\in$ S (rr[a]=s[a])

Contoh :

- Cari semua nid,nama_d,gajipokok dari semua dari semua dosen yang gaji pokoknya > 1000000:

  { D.nid \ D.nama_d \ D.gajipokok>1000000 } dimana D $\in$ Dosen.

- Cari semua nid,nama_d dari semua dosen yang mengajar pada ttahun akademik 2004 dan semester 1:

  { D.nid \ D.nama_d \ M.thn_akademik=2004 $\wedge$ semester=1 $\wedge$ Dosen.nid = Mengajar.nid }

9.2.2. Kalkulus Relasional Domain

Kalkulus relasional domain menggunakan variable – variable pada nilai - nilai domain atribut, bukan nilai – nilai untuk sebuah tupel. Ekspresi pada kalkulus relasional domain berbentuk:

{ < X1, X2, …, Xn > | P (X1, X2, …, Xn)}

Dimana : X1, X2, …,Xn menyatakan variable – variable domain.
P menyatakan rumus – rumus yang disusun dari atom – atom sebagaimana pada kalkulus relasional tupel.

Atom pada kalkulus relasional domain adalah salah satu dari:

- < X1, X2, …, Xn > $\in$ r, dimana r adalah relasi dengan n atribut dan X1, X2, …, Xn adalah variable – variable domain atau konstanta – konstanta domain.

- x Θ y, dimana x dan y adalah variable domain, Θ adalah operasi pembandingan (<, <=,=, >=, ≠). Variable x dan y harus merupakan domain – domain yang dapat dibandingkan dengan Θ.
- x Θ c, dimana x dalah variable domain. Θ adalah operator pembandingan (<, <=,=, >=, ≠) dan c adalah konstanta.

Contoh :

- Cari nip,nama_d,gajipokok daridosen, dimana jenis kelaminnya adalah Pria dan gaji pokoknya lebih besar 1200000

{ nip | ∃ nam_d | ∃ gajipokok
(Dosen(nid,nama_d,gajipokok) AND jkelamin='Pria' AND gajipokok > 1200000) }

Berdasarkan acuan model relasional, ada 2 bahasa query komersial yang tersedia,yaitu SQL dan QBE.

9.2.3. QBE (Query By Example)

QBE masing-masing mewakili bahasa query prosedural dan non-prosedural. SQL dibangun dengan basis aljabar relasional yang dijelaskan bab sebelumnya. SQL memberikan bahasa query tingkat tinggi (a high level query language) dengan struktur sederhana dengan kosakata dan gramatika yang sederhana pula, seperti berikut :

Select $A_1, A_2, \ldots, A_n$
From $T_1, T_2, \ldots, T_n$
Where P

Dimana :

$A_1,A_2,\ldots,A_n$: himpunan dari semua atribut yang hendak ditampilkan.

$T_1,T_2,\ldots,T_n$: himpunan dari semua tabel yang terlibat (diperlukan) dalam query.

P : predikat / kriteria yg diinginkan tentang informasi yg dicari.

Struktur dasar SQL tersebut equvalen dengan operasi pada aljabar relasional berikut:

π A1,A2,…An (σ P(T1 X …X Tn))

Beberapa operator SQL yang berkorespondensi dengan operator aljabar relasional :

Aljabar Relasional	SQL
$\wedge$	AND
$\vee$	OR
$\neq$	<> or ><
$\geq$	>=
$\leq$	<=