

Materi 3 : Klausu Join dan Case Statement

TUJUAN UMUM : Mahasiswa memahami dan terampil membuat Query dengan klausu Join, Inner Join, Left Join, Case Statement dan konversi data

Dengan klausu WHERE, kita dapat menghubungkan dua atau lebih tabel berdasarkan nilai dari dua field pada dua tabel.field/kolom yang digunakan untuk mengkombinasikan dua tabel biasanya melibatkan field primary key pada tab tabel le pertama dan field foreign key pada tabel lainnya. Namun usaha mengkombinasikan banyak tabel dengan klausu WHERE tidak akan bekerja jika dua field/kolom berisi nilai NULL.

Operasi menggabungkan banyak table disebut JOIN. Operasi Join dirancang khusus untuk menggabungkan banyak tabel. Dengan klausu WHERE, Kita dapat melakukan penggabungan juga dalam query yang sederhana, tetapi tidak selalu menghasilkan hasil yang benar.

a. INNER JOIN

Inner Join mengembalikan semua record yang nilainya sama pada dua tabel dan membuang record-record yang tidak samadari kedua tabel tersebut. Inner join merupakan nilai default jika type join tidak di tentukan.

Pada pembahasan GROUP BY, Kita mempunyai contoh query untuk menampilkan NIP, NamaPegawai dan jumlah anaknya (hanya yang mempunyai anak), dengan query sebagai berikut :

```
SELECT P.NIP, P. NAMA, COUNT(A.NIP) AS [JUMAH ANAK] FROM
PEGAWAI P, ANAK A WHERE P.NIP = A.NIP
GROUP BY P.NIP, P.NAMA
```

Kita dapat menuliskan query dengan inner join yang akan menghasilkan output yang sama, yaitu dengan query sebagaimana berikut :

```
SELECT P.NIP, P. NAMA, COUNT(A.NIP) AS [JUMAH ANAK] FROM
PEGAWAI P INNER JOIN ANAK A ON P.NIP = A.NIP
GROUP BY P.NIP, P.NAMA
```

b. LEFT JOIN

Left Join mengembalikan semua record dari tabel pertama (kiri) dari kedua tabel tersebut (yang digabungkan) meskipun nilai pada record pada tabel pertama tidak mempunyai kesamaan pada tabel lainnya.(tabel kedua/kanan)

Jika query INNER JOIN menjadi LEFT JOIN pada contoh diatas, seperti dibawah ini :

```
SELECT P.NIP, P. NAMA, COUNT(A.NIP) AS [JUMAH ANAK] FROM
PEGAWAI P LEFT JOIN ANAK A ON P.NIP = A.NIP
GROUP BY P.NIP, P.NAMA
```

Menggunakan Left Join, akan ditampilkan semua record pada tabel pertama (tabel pegawai) meskipun N01, N03, N06, dan N07 (kondisi field NIP) tidak terdapat pada tabel anak.

c. RIGHT JOIN

Right Join adalah kebalikan dari left join, yaitu inner join yaitu mengembalikan semua record yang ada pada tabel kedua(kanan) dari kedua tabel, meskipun nilai record pada tabel kedua tidak mempunyai kesamaan pada tabel lainnya (tabel pertama/kiri)

Jika pada contoh query LEFT JOIN kita ganti menjadi RIGT JOIN sebagai berikut :

```
SELECT P.NIP, P. NAMA, COUNT(A.NIP) AS [JUMAH ANAK] FROM
PEGAWAI P RIGT JOIN ANAK A ON P.NIP = A.NIP
GROUP BY P.NIP, P.NAMA
```

Ini akan menghasilkan semua record yang ada pada tabel kedua(kanan). Hasilnya akan sama dengan inner join, karena N02 dan N05 sama sama ada pada kedua tabel.

Sekarang misalkan saja kita tambahkan record pada tabel anak, masukkan NIP yang tidak ada pada tabel pegawai :

```
INSERT INTO ANAK (NIP,NAMA,JK,UMUR)
VALUES ('N15','Andre','L',12)
```

Kemudian jalankan Query RIGT JOIN diatas maka, query akan menampilkan nilai NULL untuk NIP dan NAMA karena memang NIP N15 tidak ada pada tabel pegawai, tetapi tetap ditampilkan pada hasil query. Karena kita menggunakan RIGT JOIN maka semua yang ada pada tabel kedua (sebelah kanan) akan ditampilkan meskipun N15 tidak ditemukan pada tabel pegawai.

d. Case Statement

Digunakan untuk menentukan sejumlah kondisi dan mengembalikan satu nilai dari banyak nilai ekspresi nilai kembalian.

Terdapat dua format CASE, yaitu :

- ❖ Simple CASE fuction, yaitu untuk mebandingkan suatu ekspresi pada ekspresi yang sederhana untuk menentukan hasilnya.

Contoh :

```
SELECT P.NAMA, J.JABATAN,
CASE J.KJ
    WHEN 'J01' THEN 'MARKETING'
    WHEN 'J02' THEN 'IT SOFTWARE'
    WHEN 'J03' THEN 'IT SOFTWARE'
    WHEN 'J04' THEN 'IT HARDWARE'
    WHEN 'J05' THEN 'IT SOFTWARE'
    WHEN 'J06' THEN 'ADMINISTRASI'
    ELSE 'STAFF'
END AS DEPARTEMEN
FROM PEGAWAI P INNER JOIN JABATAN J ON P.KJ = J.KJ
```

- ❖ Searched CASE function, yaitu untuk melakukan evaluasi dari suatu kumpulan ekspresi Boolean evaluates untuk menentukan hasilnya.

Contoh :

```
SELECT P.NAMA, J.JABATAN,
CASE J.KJ
    WHEN J.GAJI > 5000000 THEN 'TOP MANAGER'
    WHEN J.GAJI > 3000000 THEN 'MIDDLE MANAGER'
    ELSE 'STAFF'
END AS [LEVEL MANAGER]
FROM PEGAWAI P INNER JOIN JABATAN J ON P.KJ = J.KJ
```

e. Konversi Data

Untuk mengkonversi data (mengubah data dari suatu tipe data tertentu ke tipe data yang lainnya) kita akan menggunakan CONVERT

Sintaks :

```
CONVERT ( tipe_data [ (panjang) ], ekspresi [, style])
```

Contoh:

```
SELECT P.NAMA AS [NAMA PEGAWAI], A.NAMA AS [NAMA ANAK],  
CONVERT(VARCHAR, A.UMUR) + 'TAHUN' AS [UMUR ANAK]  
FROM PEGAWAI P INNER JOIN ANAK A ON P.NIP = A.NIP
```

Karena field umur pada tabel anak didefinisikan sebagai integer maka konversikan tipe data umur menjadi varchar, dan setelah menjadi varchar dapat kita gabungkan (concat) dengan string 'TAHUN'



PERCOBAAN PRAKTIKUM (Latihan-1)

1. Tampilkan nama pegawai dan tambahkan 'Mr' jika jenis kelaminya (JK) laki-laki (L) dan tambahkan 'Ms' jika jenis kelaminya (JK) adalah perempuan (P)
2. Tampilkan nama anak dan jenis umur anak, jika umur < 3 tahun 'BATITA', jika < 5 tahun 'BALITA', < 15 tahun 'ANAK-ANAK' dan >=15 tahun 'DEWASA'
3. Tampilkan Nama Jabatan dan Gaji ditambahkan kata 'Rupiah' setelah Gaji.