JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA

MODUL 8
BAHASA QUERY

Membuat Agregat Data Menggunakan Fungsi Grup

A. TUJUAN

- Mahasiswa dapat mengidentifikasi fungsi grup yang tersedia.
- Mahasiswa dapat mengetahui penggunaan fungsi grup.
- Mahasiswa dapat menggelompokan data menggunakan klausa GROUP BY
- Mahasiswa dapat menyeleksi data yang dikelompokan menggunakan klausa HAVING

B. LANDASAN TEORI & LANGKAH PRAKTIKUM

Seperti yang telah dibahas pada modul 4, terdapat 2 jenis fungsi yaitu fungsi *single row* dan fungsi *multiple row*. Pada modul ini akan dibahas tentang fungsi *multiple row* (beberapa baris) yang akan memanfaatkan fungsi grup. Modul ini akan membahas bagaimana cara memperoleh ringkasan informasi untuk mengelompokan baris (seperti rata-rata). Akan membahas juga bagaimana mengelompokan baris dalam sebuah table ke dalam himpunan yang lebih kecil dan bagaimana menspesifikasikan kriteria pencarian untuk sekelompok baris.

B.1. Apa Yang Dimaksud Dengan Fungsi Grup

Tidak seperti fungsi *single-row,* fungsi grup beroperasi pada sekelompok baris untuk menghasilkan satu hasil per grup (kelompok). Kelompok ini mungkin seluruh tabel atau tabel yang dibagi ke dalam grup. Gambar 1 menunjukan gambar contoh fungsi grup.

Gambar 8.1. Contoh ilustrasi fungsi grup

B.2. Tipe-tipe dari Fungsi Grup

Setiap fungsi menerima satu argumen. Tabel 1 memperlihatkan berbagai jenis dari fungsi grup beserta dengan argumen yang dapat digunakan.

Tabel 8.1. Tipe-tipe dari Fungsi Grup

Fungsi	Keterangan
AVG([DISTINCT|ALL] *n*)	Rata-rata nilai dari *n,* mengabaikan nilai null
COUNT(*|{[DISTINCT|ALL] *expr*})	Jumlah dari baris, dimana *expr* mengevaluasi sesuatu yang tidak null (Menghitung seluruh baris menggunakan *, akan memasukan juga baris duplikasi dan baris dengan null)
MAX([DISTINCT|ALL] *expr*)	Nilai maksmium dari *expr* , mengabaikan null
MIN([DISTINCT|ALL] *expr*)	Nilai minimum dari *expr* , mengabaikan null
STDEV([DISTINCT|ALL] *x*)	Standar deviasi dari x, mengabaikan null
SUM([DISTINCT|ALL] *n*)	Menjumlahkan nilai *n*, mengabaikan null
VARIANCE([DISTINCT|ALL] *x*)	*Variance* n, mengabaikan null

B.3. Sintak Fungsi Grup

Sintak dari fungsi grup adalah sebagai berikut :

```
SELECT      [column,] group_function (column), ....
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```

Petunjuk penggunaan fungsi grup:

- DISTINCT membuat fungsi hanya memperhatikan nilai yang tidak duplikat, sedangkan ALL memperhatikan seluruh nilai termasuk yang memiliki nilai duplikat. Secara default fungsi grup adalah ALL.
- Tipe data untuk *expr* dapat berupa VARCHAR2, CHAR, NUMBER atau DATE.
- Seluruh fungsi grup mengabaikan nilai null. Untuk substitusi nilai null dapat menggunakan fungsi NVL, NVL2 atau COALESCE.

B.4. Penggunaan Fungsi Grup

B.4.1. Menggunakan fungsi AVG dan SUM

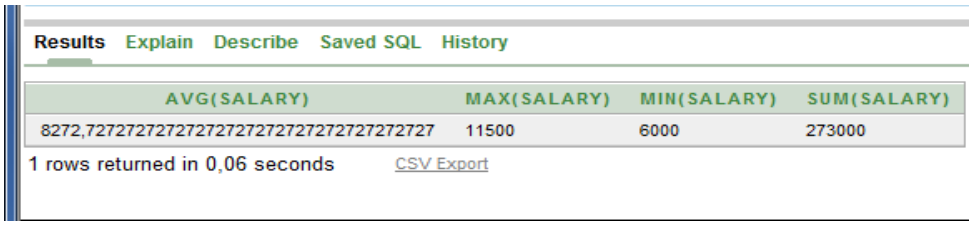
Fungsi AVG dan SUM dapat digunakan untuk kolom yang memiliki nilai numerik.

Contoh 1

- Menampilkan rata-rata gaji, gaji tertinggi, gaji terendah dan jumlah dari gaji bulanan untuk semua Sales Representative (REP)

```
SELECT AVG(salary),MAX(salary),MIN(salary),SUM(salary)
FROM employees
WHERE job_id like '%REP%';
```

Hasil :



Results Explain Describe Saved SQL History

AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
8272,727272727272727272727272727272727	11500	6000	273000

1 rows returned in 0,06 seconds CSV Export

B.4.2. Menggunakan fungsi MIN dan MAX

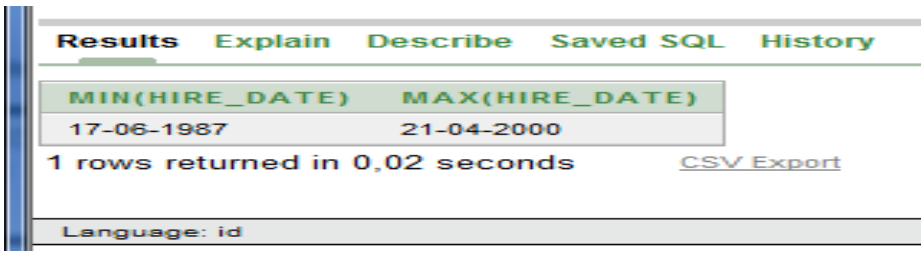
Fungsi MIN dan MAX dapat digunakan untuk tipe data numerik, karakter dan date.

Contoh 2

- Menampilkan employee yang paling junior dan yang paling senior

```
SELECT MIN(hire_date),MAX(hire_date)
FROM employees;
```

Hasil :



Results Explain Describe Saved SQL History

MIN(HIRE_DATE)	MAX(HIRE_DATE)
17-06-1987	21-04-2000

1 rows returned in 0,02 seconds CSV Export

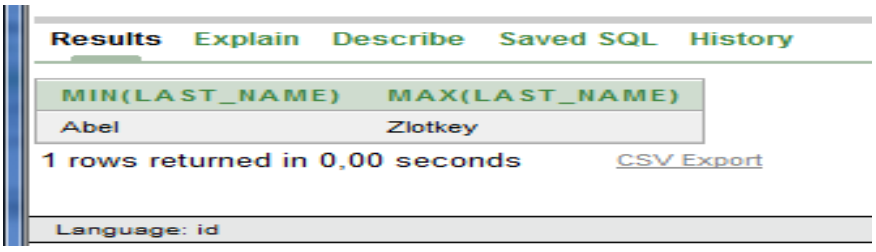
Language: id

Contoh 3:

- Menampilkan last_name dari employee yang pertama dalam urutan alphabet dan yang terakhir dalam urutan alphabet.

```
SELECT MIN(last_name),MAX(last_name)
FROM employees;
```

Hasil :



Results Explain Describe Saved SQL History

MIN(LAST_NAME)	MAX(LAST_NAME)
Abel	Zlotkey

1 rows returned in 0,00 seconds CSV Export

Language: id

B.4.3. Menggunakan Fungsi COUNT

Fungsi COUNT terdiri dari 3 format yaitu :

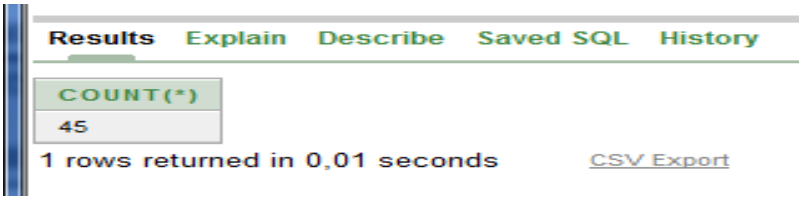
a. COUNT (*)
   Akan mengembalikan jumlah baris dalam tabel yang memenuhi kriteria perintah SELECT, termasuk baris yang duplikat, yang memiliki nilai null dalam kolom. Jika klausa WHERE diberikan bersama perintah COUNT(*) maka akan mengembalikan jumlah baris yang sesuai dengan kondisi Where.
b. COUNT(expr)
   Akan mengembalikan jumlah baris yang tidak null dari kolom yang diidentifikasi oleh expr
c. COUNT(DISTINCT expr)
   Akan mengembalikan jumlah baris yang unik (tidak duplikat), tidak null dari kolom yang diidentifikasi oleh expr.

Contoh 4

- Menampilkan jumlah karyawan yang ada di departement_id=50

```
SELECT COUNT(*)
FROM employees
WHERE department_id=50;
```

Hasil:

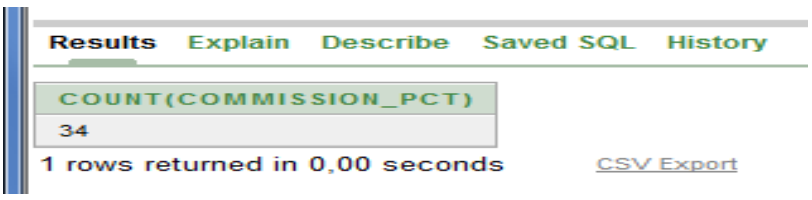


Contoh 5

- Menampilkan jumlah karyawan dalam departement=80 yang mendapatkan komisi

```
SELECT COUNT(commission_pct)
FROM employees
WHERE department_id=80;
```

Hasil:

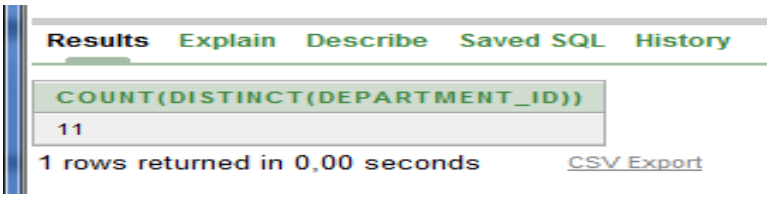


Contoh 6

- Menampilkan jumlah department tanpa duplikasi dari tabel EMPLOYEES

```
SELECT COUNT(DISTINCT(department_id))
FROM employees;
```

Hasil:



Contoh 7

- Bandingkan dengan contoh 6.

```
SELECT COUNT(department_id)
FROM employees;
```

Hasil:

B.4.5. Fungsi Grup dan Nilai Null

Fungsi grup secara keseluruhan mengabaikan nilai null pada kolom. Akan tetapi jika nilai null ingin digunakan sebagai nilai yang hendak dipaksakan untuk dihitung maka dapat menggunakan fungsi NVL.

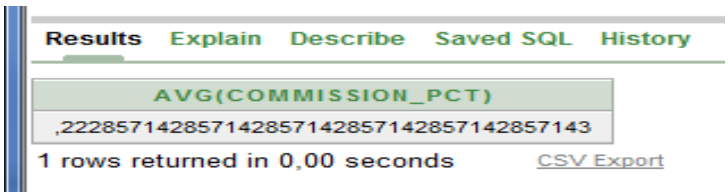
Perhatikan contoh 8 yang mengabaikan nilai null dan contoh 9 yang memasukan nilai null dalam pencarian nilai rata-rata (AVG) dari commission_pct.

Contoh 8

Menampilkan nilai rata-rata kolom commission_pct yang memiliki nilai valid dari tabel EMPLOYEES. Nilai rata-rata dihitung dari jumlah commission_pct yang dibayarkan kepada karyawan (yang berarti mengabaikan nilai null, karena null berarti tidak mendapatkan commission_pct) dibagi dengan karyawan yang menerima komisi.

```
SELECT AVG(commission_pct)
FROM employees;
```

Hasil:



Contoh 9

Menampilkan nilai rata-rata seluruh kolom commission_pct baik yang memiliki nilai null maupun yang tidak. Nilai rata-rata dihitung dari jumlah commission_pct yang dibayarkan kepada seluruh karyawan dibagi dengan seluruh jumlah karyawan

```
SELECT AVG(NVL(commission_pct,0))
FROM employees;
```

Hasil:

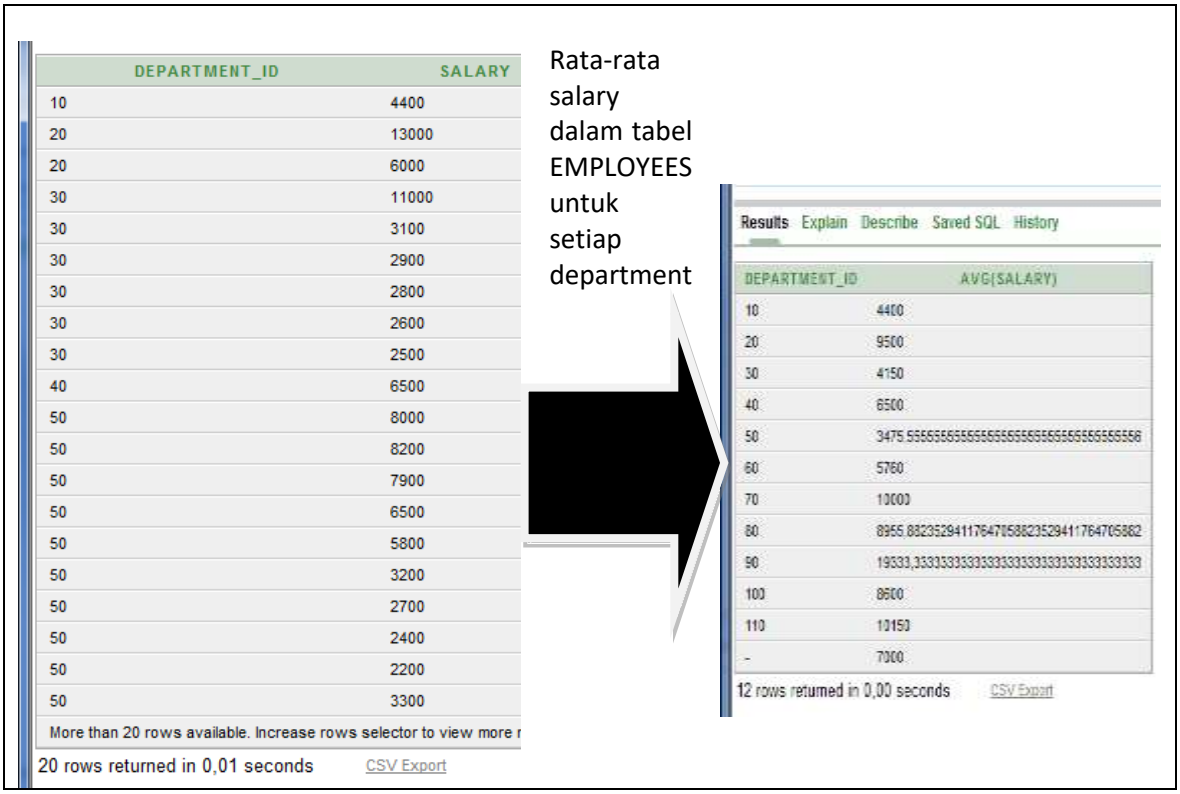
Results Explain Describe Saved SQL History

AVG(NVL(COMMISSION_PCT,0))
,072897196261682242990654205607476635514

1 rows returned in 0,00 seconds CSV Export

B.5. Membuat Grup dari Data

Pada penjelasan sebelumnya, fungsi grup memperlakukan tabel sebagai sebuah grup besar dari data. Klausa GROUP BY dapat digunakan untuk membagi informasi dalam tabel ke dalam grup yang lebih kecil. Ilustrasi dapat dilihat pada gambar 2.



Gambar 8.2. Contoh ilustrasi GROUP BY

Sintak klausa GROUP BY

```
SELECT      column,group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```

Keterangan:

- Group_by_expression : menspesifikasikan kolom yang nilainya digunakan sebagai dasar dari pengelompokan baris

Klausa GROUP BY dapat digunakan untuk membagi baris dalam sebuah table ke dalam grup-grup. Selanjutnya penggunaan fungsi grup untuk mengembalikan informasi summary dari setiap grup.

Petunjuk penggunaan klausa GROUP BY:

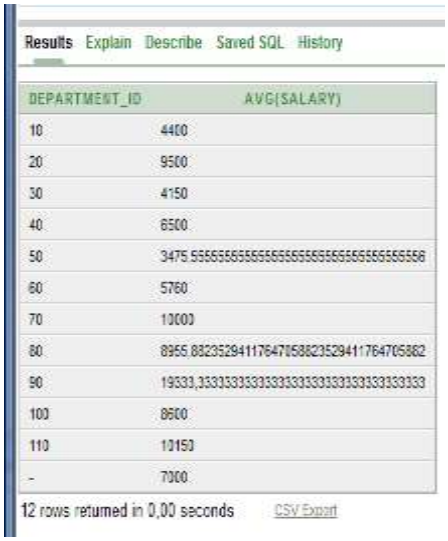
- Dalam menggunakan klausa GROUP BY, pastikan bahwa semua kolom dalam daftar SELECT yang bukan bagian dari fungsi grup, dimasukan dalam klausa GROUP BY.
- Gunakan klausa WHERE jika baris dalam tabel tidak akan dimasukan dalam grup.
- Kolom digunakan dalam klausa GROUP BY
- Kolom alias tidak dapat digunakan dalam klausa GROUP BY

Contoh 10

- Penggunaan klausa GROUP BY. Seluruh kolom dalam daftar SELECT yang tidak termasuk dalam fungsi grup yang digunakan dimasukan ke dalam klausa GROUP BY.

```
SELECT department_id,AVG(salary)
FROM employees
GROUP BY department_id;
```

Hasil



Results Explain Describe Saved SQL History

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
30	4150
40	6500
50	3475.5555555555555555555555555555555555556
60	5760
70	10000
80	8955.882352941176470588235294117647058824
90	19333.333333333333333333333333333333333
100	8600
110	10150
-	7000

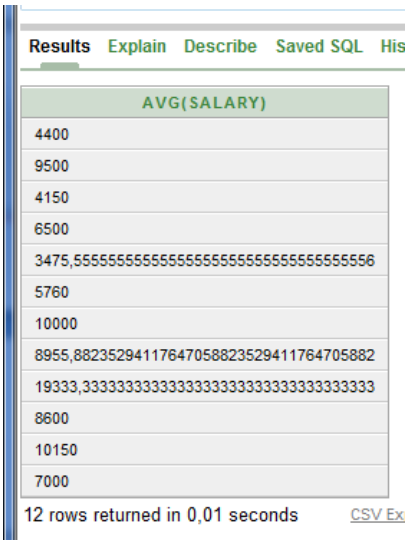
12 rows returned in 0.00 seconds CSV Export

Contoh 11

- Penggunaan klausa GROUP BY tanpa ada kolom dalam perintah SELECT.

```
SELECT AVG(salary)
FROM employees
GROUP BY department_id;
```

Hasil



Results Explain Describe Saved SQL His

AVG(SALARY)
4400
9500
4150
6500
3475,5555555555555555555555555555555556
5760
10000
8955,882352941176470588235294117647058882
19333,333333333333333333333333333333333
8600
10150
7000

12 rows returned in 0,01 seconds CSV Ex

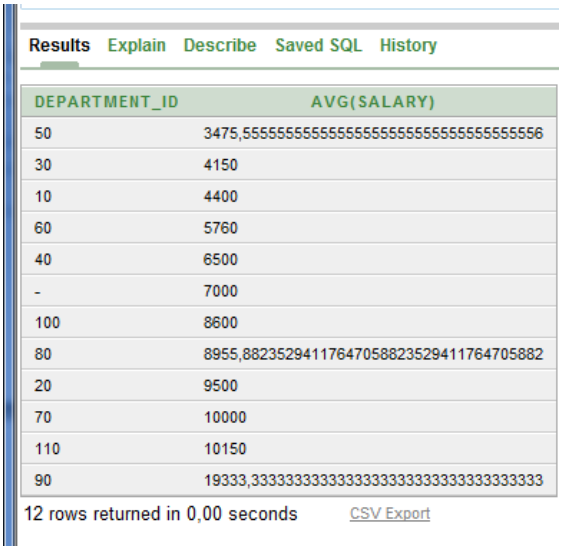
Penjelasan:
Karena tidak memasukan department_id dalam perintah SELECT mengakibatkan hasil dari contoh 11 tidak memiliki arti, karena tidak diketahui rata-rata salary tersebut dari department_id yang mana .

Contoh 12

- Penggunaan klausa GROUP BY dan fungsi grup dalam klausa ORDER BY

```
SELECT department_id,AVG(salary)
FROM employees
GROUP BY department_id
ORDER BY AVG(salary);
```

Hasil:



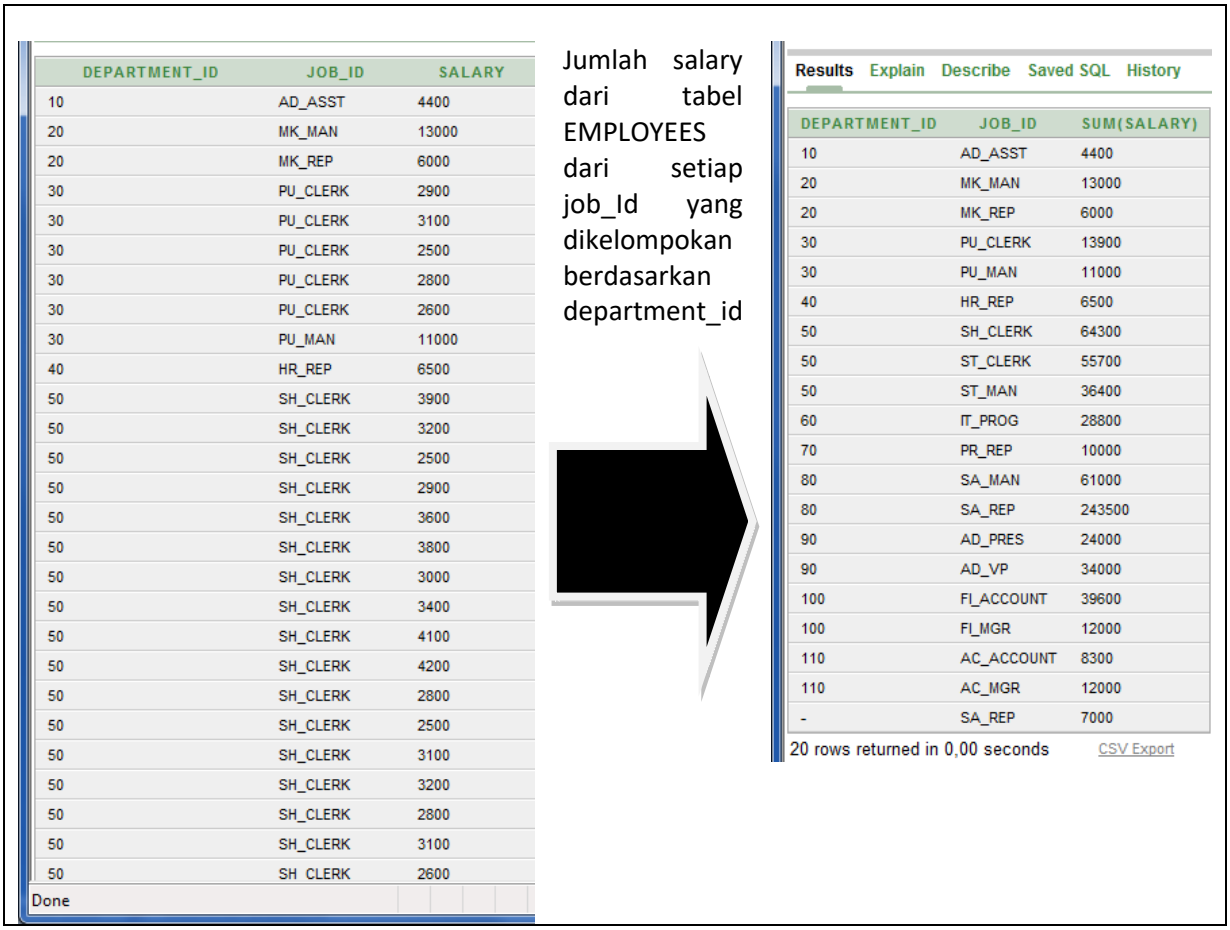
Results Explain Describe Saved SQL History

DEPARTMENT_ID	AVG(SALARY)
50	3475,5555555555555555555555555555555556
30	4150
10	4400
60	5760
40	6500
-	7000
100	8600
80	8955,882352941176470588235294117647058882
20	9500
70	10000
110	10150
90	19333,333333333333333333333333333333333

12 rows returned in 0,00 seconds CSV Export

B.6. Menggunakan Grouping Dengan Lebih Dari Satu Kolom

Kadangkala kita memerlukan untuk melihat hasil dari grup dalam grup. Seperti pada Gambar 3 yang menampilkan jumlah salary yang harus dibayarkan untuk setiap job title dalam setiap department. Pertama kali tabel EMPLOYEES akan dikelompokan berdasarkan department number dan kemudian dikelompokan dalam job title. Perhatikan untuk department_id =20, terdapat 2 jenis job_id yaitu 'PU_CLERK','PU_MAN', setiap job_id tersebut dijumlahkan salarynya.



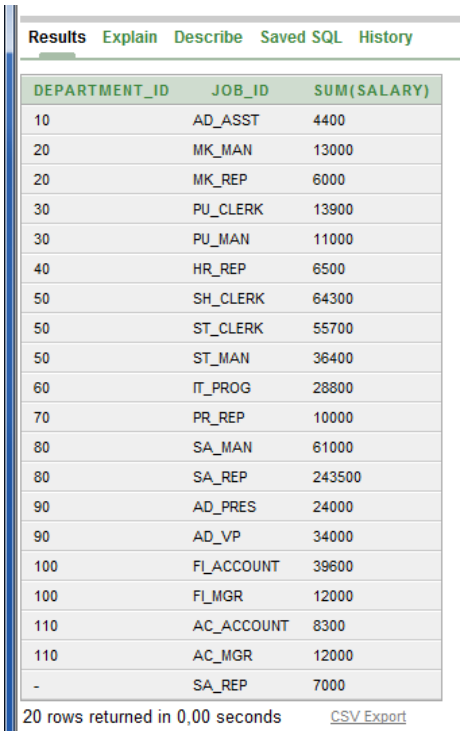
Gambar 8.3. Ilustrasi GROUP BY Lebih dari Satu Kolom

Contoh 13

- Menampilkan department_id,job_id dan jumlah salary untuk setiap department_id dan job_id.

```
SELECT department_id,job_id,sum(salary)
FROM employees
GROUP BY department_id,job_Id
ORDER BY department_id,job_Id
```

Hasil :



Results Explain Describe Saved SQL History

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
30	PU_CLERK	13900
30	PU_MAN	11000
40	HR_REP	6500
50	SH_CLERK	64300
50	ST_CLERK	55700
50	ST_MAN	36400
60	IT_PROG	28800
70	PR_REP	10000
80	SA_MAN	61000
80	SA_REP	243500
90	AD_PRES	24000
90	AD_VP	34000
100	FI_ACCOUNT	39600
100	FI_MGR	12000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
-	SA_REP	7000

20 rows returned in 0,00 seconds CSV Export

Keterangan:

- Hasil summary untuk setiap grup dan sub grup dapat dilakukan dengan menambahkan kolom lebih dari satu setelah GROUP BY.
- Pengurutan hasil dapat dilakukan berdasarkan kriteria tertentu. Dalam contoh 13 pengurutan dilakukan berdasarkan urutan pengelompokan yaitu department_id dan job_id.
- Klausa GROUP BY akan mengevaluasi dua langkah yaitu:
    - Pertama mengelompokan berdasarkan department_id.
    - Kedua mengelompokan berdasarkan job_id dari kelompok department_id (Setiap kelompok department_id akan memiliki beberapa job_id)

  Kemudian fungsi SUM digunakan pada kolom salary untuk setiap job_id yang menjadi anggota dari department_id

B.7. Kueri yang Ilegal dalam Fungsi Grup

Beberapa kondisi yang menyebabkan kueri ilegal dalam fungsi grup

1. Setiap kolom atau expresi dalam daftar SELECT yang tidak menggunakan fungsi aggregat maka kolom atau expresi tersebut harus terdapat dalam klausa GROUP BY, jika hal tersebut tidak ditepati maka akan terjadi error.

Contoh 14

- Kueri yang salah ketika akan menampilkan jumlah employees dalam setiap departement.

```
SQL> SELECT department_id,count(last_name)
  2  FROM employees;
select department_id,count(last_name)
       *
ERROR at line 1:
ORA-00937: not a single-group group function
```

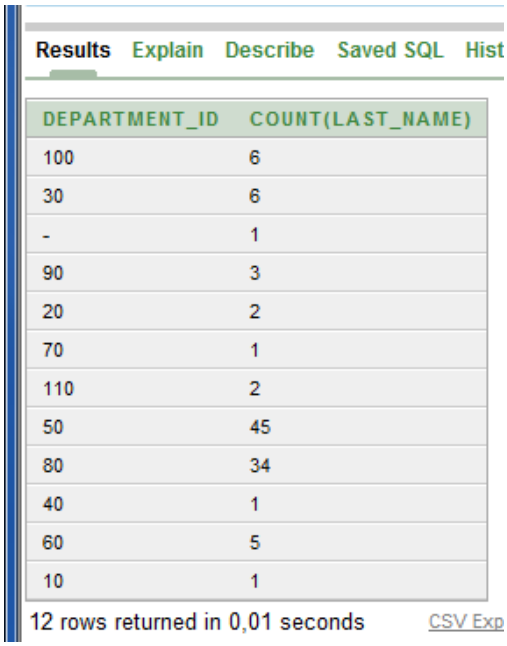
Ketika kita menggabungkan antara item individual(DEPARTEMENT_ID) dan fungsi grup (COUNT) dalam statemetn SELECT yang sama, maka individual item (dalam kasus ini DEPARTEMENT_ID) harus harus dimasukan ke dalam klaus GROUP_BY. Jika tidak maka akan terjadi error seperti pada contoh 14. Koreksi dari kueri pada contoh 14, dapat dilihat pada contoh 15.

Contoh 15

- Menampilkan jumlah employees dalam setiap departement.

```
SELECT department_id,count(last_name)
FROM employees GROUP BY department_id;
```

Hasil:



Results | Explain | Describe | Saved SQL | Hist

DEPARTMENT_ID	COUNT(LAST_NAME)
100	6
30	6
-	1
90	3
20	2
70	1
110	2
50	45
80	34
40	1
60	5
10	1

12 rows returned in 0,01 seconds CSV Exp

2. Tidak dapat menggunakan klausa WHERE untuk membatasi grup, tetapi menggunakan klausa HAVING.
3. Tidak dapat menggunakan fungsi grup dalam klausa WHERE

Contoh 16

- Kueri yang salah ketika menggunakan klausa WHERE dalam melakukan pembatasan grup, untuk menampilkan rata-rata salary diatas 8000;

```
SQL> SELECT department_id,AVG(salary)
  2  FROM employees
  3  WHERE AVG(salary) > 8000
  4  GROUP BY department_id;
where avg(salary) > 8000
      *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Contoh 17

- Perbaikan kueri contoh 16 yang benar, untuk menampilkan rata-rata salary diatas 8000.

```
SELECT department_id,AVG(salary)
FROM employees
HAVING AVG(salary) > 8000
GROUP BY department id;
```

Hasil :

```
 DEPARTMENT_ID AVG(SALARY)
------------- -----------
          100        8600
           90  19333,3333
           20        9500
           70       10000
          110       10150
           80  8955,88235

6 rows selected.
```

B.7. Membatasi Hasil Grup

Dengan cara yang sama ketika kita membatasi baris dengan klausa WHERE, maka untuk membatasi grup digunakan klausa HAVING.

Sintak untuk penggunaan klausa HAVING

```
SELECT      column,group_function
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group _condition]
[ORDER BY   column];
```

group_condition akan membatasi grup baris yang dikembalikan oleh group tersebut jika kondisi yang diberikan bernilai true.

Yang dilakukan oleh server ketika memproses klausa HAVING adalah :

1. Baris dikelompokan
2. Fungsi grup diterapkan pada grup
3. Grup-grup yang sesuai dengan kriteria dalam klausa HAVING ditampilkan.

Contoh 18

- Menampilkan department_id, maksimum salary dari setiap department_id yang memiliki maksimum salary diatas 10000.

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id
HAVING max(salary) > 10000;
```

Hasil :

```
DEPARTMENT_ID MAX(SALARY)
------------- -----------
          100       12000
           30       11000
           90       24000
           20       13000
          110       12000
           80       14000

6 rows selected.
```

Contoh 19

- Menampilkan job_id, jumlah salary setiap bulan alias PAYROLL untuk yang jumlah salarynya diatas 13000 dan tidak menampilkan yang job idnya adalah REP diurutkan berdasarkan jumlah salarynya.

```
SELECT job_id, SUM(salary) PAYROLL
FROM employees
WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING SUM(salary) > 13000
ORDER BY SUM(salary);
```

Hasil :

```
JOB_ID        PAYROLL
---------- ----------
PU_CLERK        13900
AD_PRES         24000
IT_PROG         28800
AD_VP           34000
ST_MAN          36400
FI_ACCOUNT      39600
ST_CLERK        55700
SA_MAN          61000
SH_CLERK        64300
```

B.8. Fungsi Grup Bersarang

Fungsi grup dapat dibuat bersarang dengan kedalaman 2.

Contoh 20

- Menampilkan maksimum dari rata-rata gaji.

```
SELECT MAX(AVG(salary))
FROM employees
GROUP BY department id;
```

Hasil :

```
MAX(AVG(SALARY))
----------------
      19333,3333
```

C. TUGAS

(No. 1 sd 3, pilihlah True jika pernyataan benar dan False jika pernyataan Salah)

1. Fungsi grup bekerja pada beberapa row untuk menghasilkan satu hasil untuk setiap grupnya (True/False).
2. Fungsi grup melibatkan null dalam perhitungan (True/False).
3. Klausa WHERE membatasi baris sebelum dimasukan ke dalam perhitungan grup (True/False).

4. Tampilkan salary tertinggi, terendah, jumlah salary dan rata-rata salary untuk seluruh karyawan bulatkan hasilnya. Beri nama kolom secara berurutan dengan Maximum, Minimum, Sum, dan Average. Simpan dengan nama lab8_4.sql

```
 MAXIMUM    MINIMUM        SUM    AVERAGE
---------- ---------- ---------- ----------
     24000       2100     691400       6462
```

5. Modifikasi file lab8_4.sql untuk menampilkan minimum, maksimum, jumlah salary dan rata-rata salary untuk setiap tipe job. Simpan dengan nama lab8_5.sql.

```
JOB_ID        MAXIMUM    MINIMUM        SUM    AVERAGE
---------- ---------- ---------- ---------- ----------
IT_PROG          9000       4200      28800       5760
AC_MGR          12000      12000      12000      12000
AC_ACCOUNT       8300       8300       8300       8300
ST_MAN           8200       5800      36400       7280
PU_MAN          11000      11000      11000      11000
AD_ASST          4400       4400       4400       4400
AD_VP           17000      17000      34000      17000
SH_CLERK         4200       2500      64300       3215
FI_ACCOUNT       9000       6900      39600       7920
FI_MGR          12000      12000      12000      12000
PU_CLERK         3100       2500      13900       2780
SA_MAN          14000      10500      61000      12200
MK_MAN          13000      13000      13000      13000
PR_REP          10000      10000      10000      10000
AD_PRES         24000      24000      24000      24000
SA_REP          11500       6100     250500       8350
MK_REP           6000       6000       6000       6000
ST_CLERK         3600       2100      55700       2785
HR_REP           6500       6500       6500       6500

19 rows selected.
```

6. Tampilkan jumlah orang yang mempunyai job yang sama, beri label JUMLAH

```
JOB_ID         JUMLAH
---------- ----------
AC_ACCOUNT          1
AC_MGR              1
AD_ASST             1
AD_PRES             1
AD_VP               2
FI_ACCOUNT          5
FI_MGR              1
HR_REP              1
IT_PROG             5
MK_MAN              1
MK_REP              1
PR_REP              1
PU_CLERK            5
PU_MAN              1
SA_MAN              5
SA_REP             30
SH_CLERK           20
ST_CLERK           20
ST_MAN              5

19 rows selected.
```

7. Tentukan jumlah manager tanpa melist manager dalam hasilnya. Beri label Jumlah Manager.

```
Jumlah Manager
--------------
            18
```

8. Tampilkan perbedaan antara gaji tertinggi dan terendah. Beri label PERBEDAAN GAJI

```
Perbedaan Gaji
--------------
         21900
```

9. Tampilkan nomor manager dan minimum salary dari employee yang dibawahi oleh manager tersebut, dengan minimum salary lebih besar dari 6000. Urutkan hasilnya berdasarkan salary.

```
MANAGER_ID MIN(SALARY)
---------- -----------
       148        6100
       147        6200
       149        6200
       108        6900
       146        7000
       145        7000
       205        8300
       102        9000
                 24000
```

10. Buatlah kueri untuk menampilkan jumlah employee yang disewa berdasarkan kelompok tahun sewa yaitu 1995, 1996,1997 dan 1998, beserta dengan total untuk tahun tersebut. Tampilan yang diinginkan adalah :

TOTAL	1995	1996	1997	1998
65	4	10	28	23

11. Buatlah kueri untuk menampilkan setiap job, salary dari setiap job berdasarkan department number dan total salary untuk setiap job yang memiliki department 20,50,80 dan 90. Hasil yang diinginkan adalah sebagai berikut:

Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
AC_ACCOUNT					8300
AC_MGR					12000
AD_ASST					4400
AD_PRES				24000	24000
AD_VP				34000	34000
FI_ACCOUNT					39600
FI_MGR					12000
HR_REP					6500
IT_PROG					28800
MK_MAN	13000				13000
MK_REP	6000				6000
PR_REP					10000
PU_CLERK					13900
PU_MAN					11000
SA_MAN			61000		61000
SA_REP			243500		250500
SH_CLERK		64300			64300
ST_CLERK		55700			55700
ST_MAN		36400			36400

19 rows selected.

D. DAFTAR PUSTAKA

1. *Oracle Database 10g : SQL Fundamental I,* Oracle Inc. 2004

☺ Setia sampai Akhir ☺