



PROGRAM STUDI
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
Sistem Basis Data



Fungsi-fungsi pada SQL

TEKNIK INFORMATIKA S1
UNIVERSITAS DIAN NUSWANTORO

Capaian Pembelajaran

Mahasiswa mampu:

- Menerapkan fungsi AVG() untuk menghitung nilai rata-rata
- Menerapkan fungsi Count() untuk melakukan pencacahan data
- Menerapkan fungsi Max() untuk menghitung nilai maksimal
- Menerapkan fungsi Min() untuk menghitung nilai minimal
- Menerapkan fungsi SUM() untuk melakukan penjumlahan nilai pada field numerik

Pokok Bahasan

Fungsi AVG()

Fungsi COUNT()

Fungsi Max()

Fungsi Min()

Fungsi SUM()

Fungsi Agregasi

- Fungsi agregasi adalah fungsi yang mengambil kumpulan (satu set atau multiset) nilai sebagai input dan mengembalikan nilai tunggal.
- Terdapat lima fungsi agregat standar bawaan SQL yaitu:
 - Average: avg()
 - Minimum: min()
 - Maximum: max()
 - Total: sum()
 - Count: count()

AVG()

- **Fungsi avg()** digunakan untuk menghitung nilai rata-rata dari seluruh field yang bertipe numerik
- Hasil dari query ini adalah relasi dengan satu atribut yang berisi satu tuple dengan nilai numerik yang sesuai dengan atribut yang dihitung
- Bentuk umum Penulisan SQL:

```
SELECT AVG(attribute)  
FROM <table>  
WHERE <condition>
```

- **WHERE**: bersifat optional (digunakan untuk hasil yang spesifik)

Contoh - Menampilkan Rata-rata Gaji Staf

- Nama table: staf

```
MariaDB [abc]> select * from staf;
```

nip	nama	posisi	jns_kel	tgl_masuk	gaji	id_cabang
a12	joko	asisten	L	2012-09-20	3000000	0
a123	indah	asisten	P	2012-11-23	3000000	0
a124	anton	manajer	L	2009-04-05	5000000	0
a13	joni	supervisor	L	2002-06-23	7000000	1
c11	Doni	asisten	L	2011-07-15	3000000	1
c12	Dani	manajer	L	2008-08-10	5000000	1

- Untuk menampilkan rata-rata gaji staf
- Penulisan SQL:

```
SELECT AVG(gaji)
FROM STAF
```

Contoh - Menampilkan Rata-rata Gaji Staf

- Menampilkan rata-rata gaji staf

```
MariaDB [abc]> select avg(gaji) from staf;  
+-----+  
| avg(gaji) |  
+-----+  
| 4333333.3333 |  
+-----+
```

- Menampilkan rata-rata gaji staf dengan format *currency* dan alias

```
MariaDB [abc]> select format(avg(gaji),0) as rata_rata_gaji from staf;  
+-----+  
| rata_rata_gaji |  
+-----+  
| 4,333,333 |  
+-----+
```

Contoh Penggunaan Where pada Fungsi AVG()

- Contoh menampilkan rata-rata gaji staf dengan posisi **Asisten**
- `Select posisi,format(avg(gaji),0) as rata_rata_gaji
from staf where posisi='asisten';`

```
MariaDB [abc]> select posisi,format(avg(gaji),0) as rata_rata_gaji  
-> from staf  
-> where posisi='asisten';  
+-----+-----+  
| posisi | rata_rata_gaji |  
+-----+-----+  
| asisten | 3,000,000      |  
+-----+-----+
```


Penggunaan AVG() pada Group by

```
MariaDB [abc]> select posisi,format(avg(gaji),0) as rata_rata_gaji  
-> from staf  
-> group by posisi;
```

posisi	rata_rata_gaji
asisten	3,000,000
manajer	5,000,000
supervisor	7,000,000

COUNT()

- Fungsi **count()** digunakan untuk melakukan pencacahan terhadap data
- Tanda * digunakan untuk menghitung ada berapa data yang dimiliki pada suatu tabel

```
SELECT count(*)  
FROM <table>
```

- Contoh menghitung terdapat berapa data pada suatu tabel staf

```
SELECT count(*)  
FROM staf
```

```
MariaDB [abc]> select count(*) from staf;  
+-----+  
| count(*) |  
+-----+  
|         6 |  
+-----+
```

Penggunaan Distinct pada Count()

- **Distinct** digunakan untuk mengeliminasi data duplikat yang tidak diperlukan dan menghitungnya cukup sekali untuk nilai yang berbeda
- Pada table staf di bawah ini, apabila kita akan menghitung ada berapa nilai yang berbeda pada kolom jns_kel, maka jawabannya adalah 2

```
MariaDB [abc]> select * from staf;
```

nip	nama	posisi	jns_kel	tgl_masuk	gaji	id_cabang
a12	joko	asisten	L	2012-09-20	3000000	0
a123	indah	asisten	P	2012-11-23	3000000	0
a124	anton	manajer	L	2009-04-05	5000000	0
a13	joni	supervisor	L	2002-06-23	7000000	1
c11	Doni	asisten	L	2011-07-15	3000000	1
c12	Dani	manajer	L	2008-08-10	5000000	1

Penggunaan Distinct pada Count()

- Namun apabila kita langsung menghitung tanpa menyertakan **Distinct**. Maka akan menghasilkan nilai 6, padahal yang diinginkan adalah 2

```
MariaDB [abc]> select count(jns_kel) from staf;  
+-----+  
| count(jns_kel) |  
+-----+  
|              6 |  
+-----+
```

- Bentuk penulisan SQL

```
SELECT COUNT(distinct column)  
FROM <table>
```

```
MariaDB [abc]> select count(distinct jns_kel) from staf;  
+-----+  
| count(distinct jns_kel) |  
+-----+  
|              2 |  
+-----+
```

MAX()

- Fungsi **Max()** digunakan untuk mencari nilai yang terbesar pada field yang bertipe numerik
- Contoh menampilkan gaji tertinggi pada table staf

```
MariaDB [abc]> select max(gaji) from staf;  
+-----+  
| max(gaji) |  
+-----+  
| 7000000 |  
+-----+
```

- Penggunaan format currency dan alias

```
MariaDB [abc]> select format(max(gaji),0) as gaji_tertinggi from staf;  
+-----+  
| gaji_tertinggi |  
+-----+  
| 7,000,000 |  
+-----+
```

MIN()

- Fungsi **Min()** digunakan untuk mencari nilai yang terkecil pada field yang bertipe numerik
- Contoh menampilkan gaji terkecil pada table staf

```
MariaDB [abc]> select min(gaji) from staf;  
+-----+  
| min(gaji) |  
+-----+  
| 3000000 |  
+-----+
```

- Penggunaan format currency dan alias

```
MariaDB [abc]> select format(min(gaji),0) as gaji_terendah from staf;  
+-----+  
| gaji_terendah |  
+-----+  
| 3,000,000 |  
+-----+
```

SUM()

- Fungsi **SUM()** digunakan untuk melakukan penjumlahan isi dari suatu field numerik
- Contoh menampilkan total gaji pada table staf

```
MariaDB [abc]> select sum(gaji) from staf;  
+-----+  
| sum(gaji) |  
+-----+  
| 26000000 |  
+-----+
```

- Penggunaan format currency dan alias

```
MariaDB [abc]> select format(sum(gaji),0) as total_gaji from staf;  
+-----+  
| total_gaji |  
+-----+  
| 26,000,000 |  
+-----+
```

Penggunaan SUM() dengan Group by

```
MariaDB [abc]> select posisi,format(sum(gaji),0) as total_gsji  
-> from staf  
-> group by posisi;
```

posisi	total_gsji
asisten	9,000,000
manajer	10,000,000
supervisor	7,000,000

Having Clause

- Klausa `Having` digunakan untuk menyatakan kondisi yang dimiliki setelah pengelompokan diterapkan
- Misalkan kita tertarik untuk menampilkan gaji rata-rata tiap posisi yang rata-rata gajinya lebih dari 3jt. Maka kita akan menerapkan fungsi agregasi, dan mengelompokkan posisi setelah itu memfilter gaji yang lebih dari 3jt. Pada case ini, kita tidak bisa menggunakan `where` namun kita perlu menerapkan klausa `having`

Having Clause

Select id_cabang, format(avg(gaji),0) as rata_rata_gaji
from staf
group by id_cabang
having avg(gaji)>3000000;

```
MariaDB [abc]> select id_cabang, format(avg(gaji),0) as rata_rata_gaji from staf group by id_cabang having avg(gaji)>3000000;
```

id_cabang	rata_rata_gaji
0	3,666,667
1	5,000,000

```
2 rows in set (0.052 sec)
```

Latihan

1. Tampilkan rata-rata gaji staf tiap cabang
2. Tampilkan total gaji staf tiap cabang
3. Tampilkan rata-rata gaji staf tiap cabang yang rata-ratanya diatas 5jt
4. Tampilkan total gaji staf tiap cabang yang total gajinya diatas 10jt

Referensi

UTAMA

1. Silberschatz, A., Korth, H. F. & Sudarshan, S., 2022. Database System Concepts. 7th ed. New York: McGraw-Hill Education
2. Connolly, T. & Begg, C., 2015. Database Systems A practical Approach to Design, Implementation, and Management. Sixth Edition ed. s.l.:Pearson.
3. Elmasri, R. & Navathe, S. B., 2016. Fundamentals of Database Systems. 7th ed. s.l.:Pearson

PENDUKUNG

Aripin., 2005. *Praktikum Basis Data Dengan Database Server MySQL*. Semarang: Fakultas Ilmu Komputer



TERIMA KASIH

ANY QUESTIONS?