

LPR WEEK 2

What we learned?

#1 : Introduction to Data & Database

#2 : Basic SQL

#3 : Advance SQL

Check it Out!



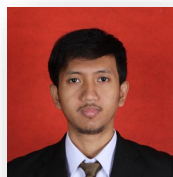
WHO WE ARE



The FAB Data Scientist

Kami adalah kelompok 9 dari Kelas Data Science **Digital Skola** batch ke-8.

Anggota The FAB Data Scientist:



Fajar Wijaya Pratama



Aji Kresna



Immanuel **Bona** Putra



#1 : Introduce to Data & Database



What is the Data ?

“Data adalah keterangan yang nyata dan benar”

What's Wrong With Data ?

- Nowadays everything can produce Data :
 - IOT (Internet Of Thing)
 - Transaction
 - Etc
- The data From Year to year is getting bigger
Pertumbuhan data, setiap tahun semakin besar dan meningkat.
- Data is a New Oil
- Data is a New Gold
- Data is a New Electricity

Artinya adalah data bisa menjadi bahan bakar baru, komoditi yang berharga, serta memecahkan sebuah permasalahan.



#1 : Introduce to Data & Database

1. Apa itu Database?

Kumpulan data yang terorganisir , yang umumnya disimpan dan diakses melalui sistem komputer.

2. Apa itu SQL?

Suatu bahasa yang digunakan untuk berkomunikasi dengan database, kegunaannya dapat sebagai alat untuk mendapatkan informasi yang dibutuhkan dari database.

- Database terdiri dari beberapa tabel, setiap tabel, terdiri dari baris dan kolom, setiap kolom berisi suatu field dari beberapa record, dan setiap row terdapat satu record.

kolom

baris

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							



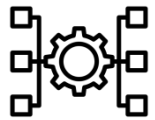
SQL tidak case sensitive,
yang artinya
MaMa = mama = MAMA



#1 : Introduce to Data & Database



SQL And Database



Organize

membuat database, dan mengorganisasikan database



Read

Data yang sudah tersedia di database kemudian sudah kita baca.



Write

Menulis script dan data baru.



Update

Mengubah database yang lama dan mengupdate yang baru.



Archive

kondisi database yang sudah selesai tetapi kita simpan.



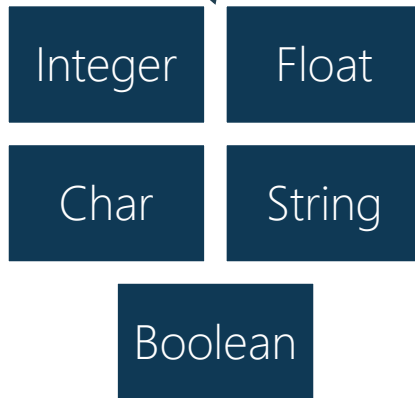
Delete

proses penghapusan database.



#1 : Introduce to Data & Database

Data Types



Data Types Based on the Structure

- Structured
Structured terdiri
dari baris dan
kolom

Contoh:
- Excel
- Sql, mysql, postgresql
- csv

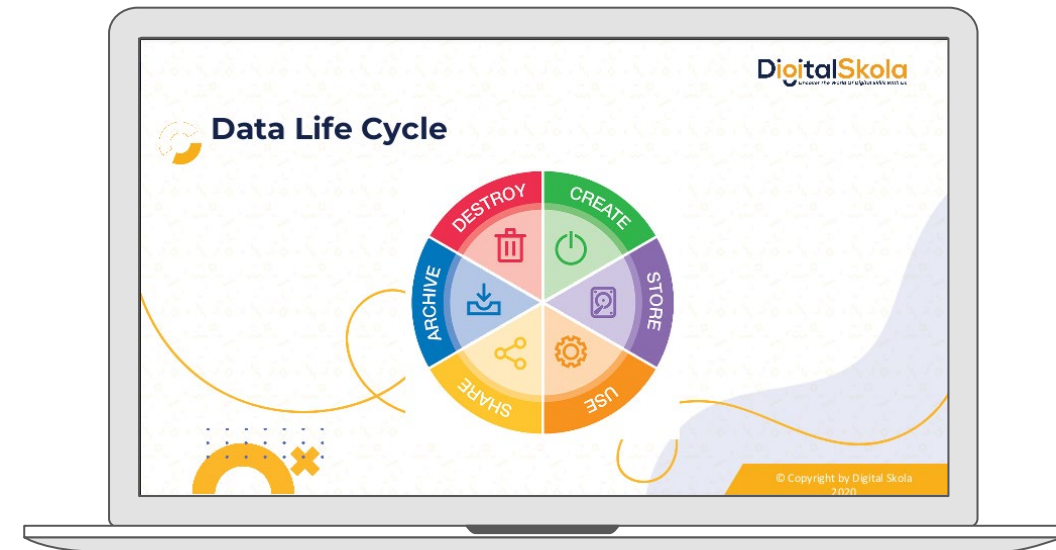
- Semi structured
data yang terbuat
dari key dan
value.

- Unstructured
data yang tidak
terstruktur

Data Life Cycle

Siklus hidup ini mencakup semua tahapan yang dilalui data, dimulai dari pembuatan data hingga data akan dihancurkan.

Data Types, Data Types Based on Structured, And Data Life Cycle



#2 : Basic SQL



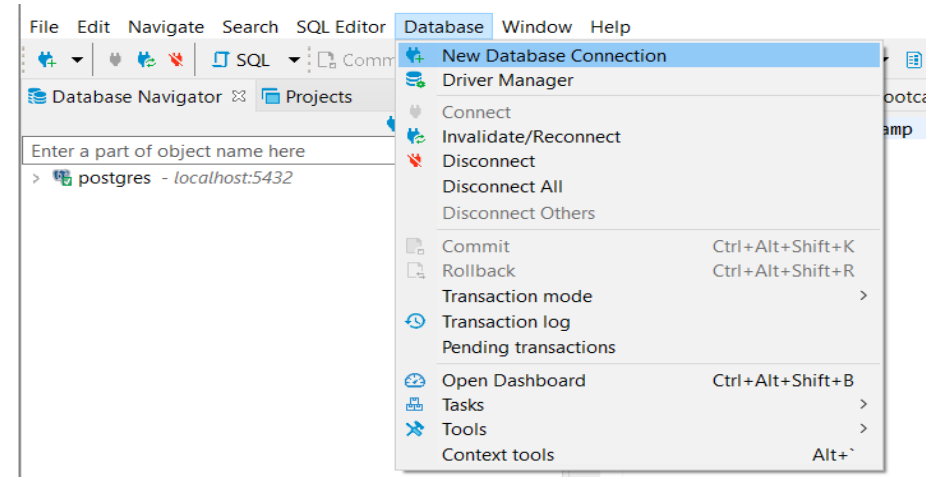
1. Create and connecting to database

1. Kita akan menggunakan postgresSQL sebagai database, database dapat di download secara gratis di <https://www.postgresql.org/download/>, pilih sesuai system operasi yang digunakan lalu install

Pada saat penginstallan jangan lupa untuk mengingat password yang sudah ditetapkan

2. Selanjutnya kita akan menggunakan Dbeaver sebagai DBMS, di asumsikan sudah dilakukan penginstallan Dbeaver, setelah Dbeaver dibuka, pilih menu :

3. Database → New Database Connection

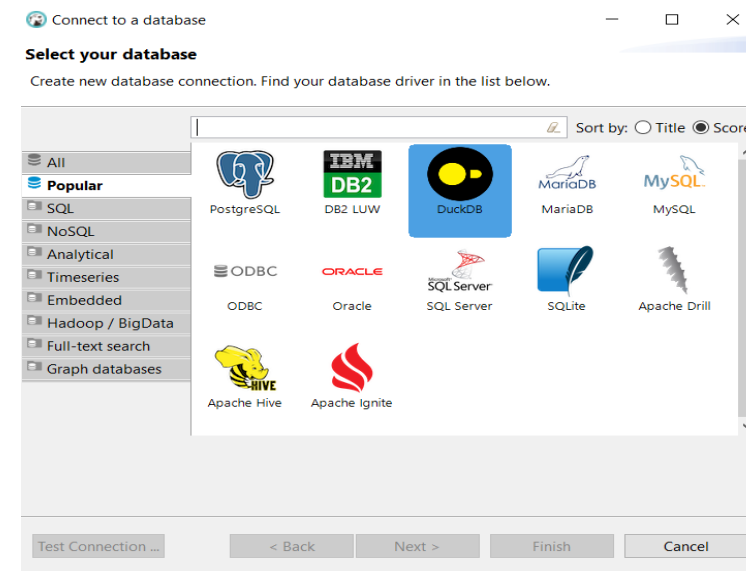


#2 : Basic SQL

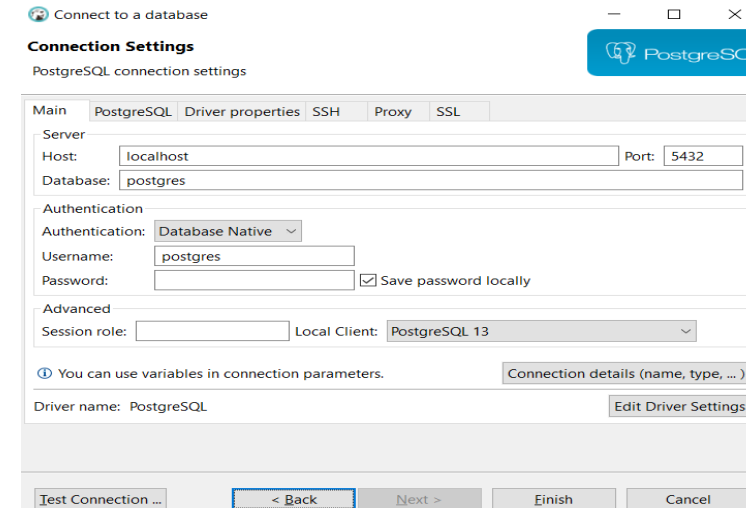


1. Create and connecting to database

- Setelah mengklik new database connection, akan muncul windows seperti disamping, pilih postgresSQL untuk database (gambar gajah biru)



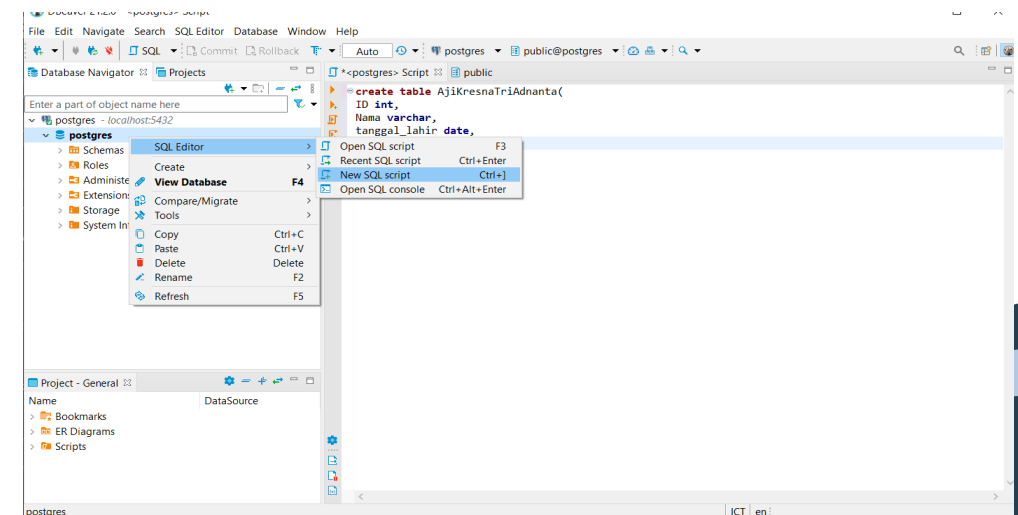
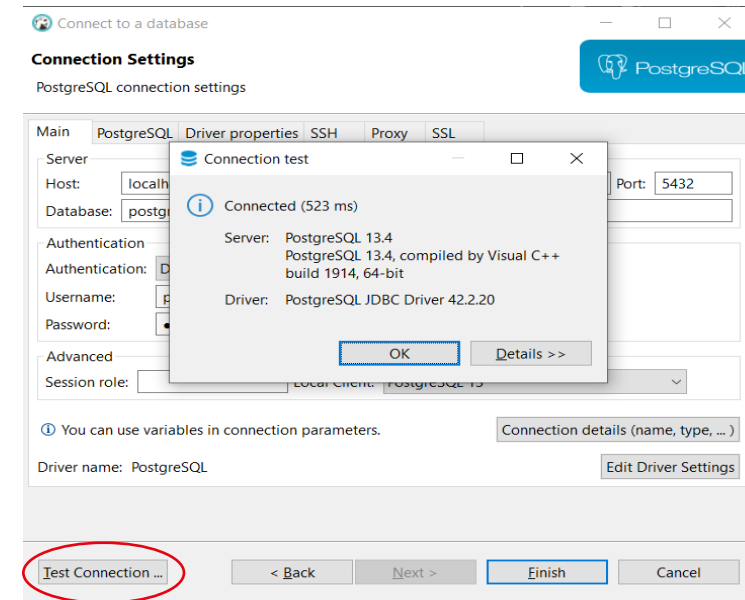
- Lalu akan muncul windows baru lagi seperti berikut, biarkan kolom kolom yang sudah terisi dan isi kolom password dengan password yang anda gunakan sewaktu penginstallan postgresSQL



#2 : Basic SQL

1. Create and connecting to database

6. Selanjutnya sebelum mengklik finish, baiknya kita lakukan pengetestan koneksi melalui button test connection(dilingkari merah)
7. Jika sukses maka akan muncul windows seperti disamping, dan kita bisa mengklik finish
8. Jika sudah berhasil maka akan muncul list database di menu sebelah kiri seperti gambar disamping, klik kanan pada database, lalu pilih SQL editor, dan kemudian pilih New SQL Script pada sub menunya untuk materi berikutnya.

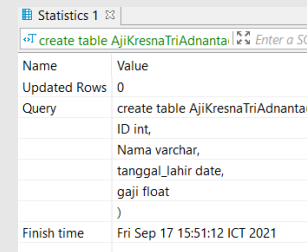
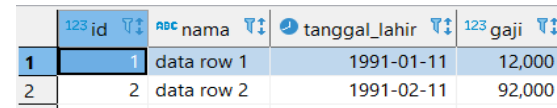
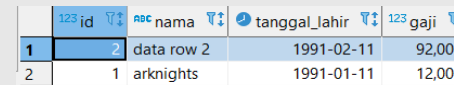
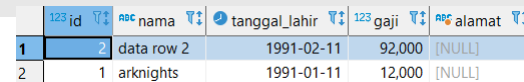


#2 : Basic SQL



2. Beberapa Fungsi Basic SQL

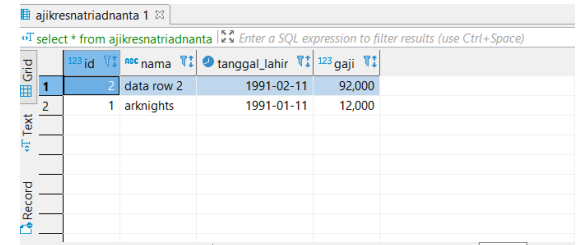
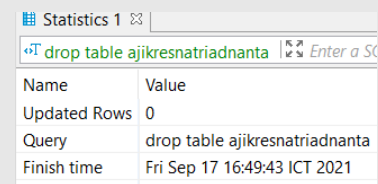
- Berikut adalah beberapa fungsi dan screenshot hasil basic SQL, mulai dari Create Table, Insert Into, Update, Alter Table, Select Data, Delete rows, cleanse table dan delete table.

Fungsi Basic SQL	Kegunaan	SQL Screenshot	Hasil
Create Table	Membuat Tabel baru - > setelah mengcreate table, table dapat diakses didalam schemas -> public -> Table	<pre>create table AjikresnaTriAdnanta(ID int, Nama varchar, tanggal_lahir date, gaji float)</pre>	
Insert Into	Memasukkan Data kedalam table	<pre>insert into ajikresnatriadnanta values (1,'data row 1','1991-01-11',12000), (2,'data row 2','1991-02-11',92000)</pre>	
Update	Mengupdate data didalam table	<pre>update ajikresnatriadnanta set nama = 'arknights' where id = 1</pre>	
Alter Table (add)	Menambahkan kolom	<pre>alter table ajikresnatriadnanta add alamat varchar</pre>	

#2 : Basic SQL



2. Beberapa fungsi Basic SQL

Fungsi Basic SQL	Kegunaan	SQL Screenshoot	Hasil																
Alter Table (Drop)	Menghapus Kolom	<pre>alter table ajikresnatriadnanta drop column alamat</pre>	<table><tr><th>123 id</th><th>abc nama</th><th>tanggal_lahir</th><th>123 gaji</th></tr><tr><td>1</td><td>data row 2</td><td>1991-02-11</td><td>92,000</td></tr><tr><td>2</td><td>1 arknight</td><td>1991-01-11</td><td>12,000</td></tr></table>	123 id	abc nama	tanggal_lahir	123 gaji	1	data row 2	1991-02-11	92,000	2	1 arknight	1991-01-11	12,000				
123 id	abc nama	tanggal_lahir	123 gaji																
1	data row 2	1991-02-11	92,000																
2	1 arknight	1991-01-11	12,000																
Select	Memilih data dari database(seluruh kolom), untuk hasil dari fungsi ini akan muncul otomatis di windows baru	<pre>select * from ajikresnatriadnanta</pre>																	
Delete Rows	Menghapus satu baris	<pre>delete from ajikresnatriadnanta where id = 2</pre>	<table><tr><th>123 id</th><th>abc nama</th><th>tanggal_lahir</th><th>123 gaji</th></tr><tr><td>1</td><td>arknight</td><td>1991-01-11</td><td>12,000</td></tr></table>	123 id	abc nama	tanggal_lahir	123 gaji	1	arknight	1991-01-11	12,000								
123 id	abc nama	tanggal_lahir	123 gaji																
1	arknight	1991-01-11	12,000																
Cleanse Table	Menghapus isi table tanpa menghapus table	<pre>truncate table ajikresnatriadnanta</pre>	<table><tr><th>123 id</th><th>abc nama</th><th>tanggal_lahir</th><th>123 gaji</th></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	123 id	abc nama	tanggal_lahir	123 gaji												
123 id	abc nama	tanggal_lahir	123 gaji																
Drop Table	Menghapus table	<pre>drop table ajikresnatriadnanta</pre>																	

#3 : Advance SQL

Beberapa fungsi Advance SQL

1. Select Distinct

- **Select Distinct** digunakan untuk memilih data/value yang berbeda-beda, tanpa dupikasi.
 - Berguna untuk menghindari duplikasi data/value.
 - Rumus Query **Distinct**:

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```
- Contoh penggunaan **Select Distinct**:
 - Untuk melihat apa saja profesi yang terdapat dalam suatu perusahaan.

```
select distinct job_id  
from datasource.job_employee;
```

Query

Hasil

	job_id
1	SH_CLERK
2	AD_VP
3	SA_MAN
4	PU_MAN
5	IT_PROG
6	ST_CLERK
7	FI_MGR
8	PU_CLERK



#3 : Advance SQL

2. Where Function

- **Where** berfungsi untuk memfilter data/value pada kumpulan data yang banyak.
 - Dapat digunakan jika ada **kondisi tertentu**.

- Rumus Query **Where**:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

- Beberapa contoh kondisi

1) Condition **“LIKE”**, **“ILIKE”**, dan **“=”**

- **“LIKE”**, **“ILIKE”**, dan **“=”** digunakan untuk menggambarkan kondisi tertentu yang cukup spesifik
- **“LIKE”**, **“ILIKE”** umumnya digunakan untuk data tipe String.
- **“=”** umumnya digunakan untuk data tipe Integer, Date.

- Rumus Query **WHERE ... LIKE**:

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```

Condition



#3 : Advance SQL

2. Where Function

Beberapa contoh kondisi

1) Condition **"LIKE"**, **"ILIKE"**, dan **"="**

Contoh penggunaan **WHERE ... LIKE** :

- Untuk mencari siapa saja **karyawan yang Bernama David**
- Untuk mencari siapa saja yang namanya **berawalan huruf S.**

Condition

Menggunakan fungsi wildcard (%)

Query

```
select first_name, job_id  
from datasource.job_employee  
where first_name like 'David';
```

Hasil

	first_name	job_id
1	David	IT_PROG
2	David	SA_REP
3	David	SA_REP

Query

```
select first_name  
from datasource.job_employee je  
where first_name like 'S%';
```

Hasil

	first_name
1	Steven
2	Shelli
3	Sigal
4	Shanta

Contoh penggunaan **WHERE ... =** :

- Untuk mencari Karyawan yang memiliki no ID **"110"**.

Condition

Query

```
select first_name, employee_id  
from datasource.job_employee je  
where employee_id = 110;
```

Hasil

	first_name	employee_id
1	John	110

#3 : Advance SQL



2. Where Function

▪ Beberapa contoh kondisi

2) Condition “<” dan “>”

- “<” (lebih kecil dari) dan “>” (lebih besar dari) digunakan untuk menggambarkan kondisi dalam range tertentu (untuk tipe data integer atau date).

- Rumus Query **WHERE** ... < dan/atau >:

```
SELECT column1,...  
FROM table_name  
WHERE columnN > ...
```

```
SELECT column1,...  
FROM table_name  
WHERE columnN < ...
```

- Contoh penggunaan **WHERE** ... < dan > :

- Untuk mencari Karyawan yang memiliki gaji lebih dari **7800**.
- Untuk mencari Karyawan yang memiliki gaji kurang dari **7800**.

Query

```
select employee_id, first_name, salary  
from datasource.job_employee je  
where salary > 7800;
```

Hasil

	employee_id	first_name	salary
1	100	Steven	24,000
2	101	Neena	17,000
3	102	Lex	17,000
4	103	Alexander	9,000
5	108	Nancy	12,000
6	109	Daniel	9,000

Query

```
select employee_id, first_name, salary  
from datasource.job_employee je  
where salary < 7800;
```

Hasil

	employee_id	first_name	salary
1	104	Bruce	6,000
2	105	David	4,800
3	106	Valli	4,800
4	107	Diana	4,200
5	111	Ismael	7,700

#3 : Advance SQL



3. String Functions

- String Functions adalah fungsi yang digunakan untuk memanipulasi **data string**.
- Beberapa contoh String Functions:

String Functions	Kegunaan	Rumus Query SQL
LENGTH	Untuk menjumlah banyaknya karakter dalam suatu string.	<pre>SELECT LENGTH(column1),... FROM table_name;</pre>
UPPER	Untuk mengubah string menjadi huruf besar.	<pre>SELECT UPPER(column1),... FROM table_name;</pre>
LOWER	Untuk mengubah string menjadi huruf kecil.	<pre>SELECT LOWER(column1),... FROM table_name;</pre>
SUBSTRING	Untuk mengekstrak data string menjadi string yang lebih pendek.	<pre>SELECT SUBSTRING(column1, <start>, <length>),... FROM table_name;</pre>
CONCAT	Untuk menggabungkan beberapa string.	<pre>SELECT CONCAT (string1, string2, stringN, ...) FROM table_name;</pre>

#3 : Advance SQL

3. String Functions

- Contoh penggunaan functions “**LENGTH**”, “**UPPER**”, “**LOWER**”, dan “**SUBSTRING**”

```
select first_name, length(first_name),  
upper(first_name), lower(first_name),  
substring(first_name, 3, 3)  
from datasource.job_employee je
```

Query

Hasil

	first_name	length	upper	lower	substring
1	Steven	6	STEVEN	steven	eve
2	Neena	5	NEENA	neena	ena
3	Lex	3	LEX	lex	x
4	Alexander	9	ALEXANDER	alexander	exa
5	Bruce	5	BRUCE	bruce	uce

Mengambil 3
karakter kebelakang

Di mulai dari huruf ke 3

- Contoh penggunaan functions “**CONCAT**”
- Digunakan untuk menggabungkan string ‘first_name’ dan ‘last_name’

Ditambahkan data string sapsi menggunakan ‘ ’

```
select first_name, last_name ,  
concat(first_name, ' ', last_name)  
from datasource.job_employee je
```

Query

Hasil

	first_name	last_name	concat
1	Steven	King	Steven King
2	Neena	Kochhar	Neena Kochhar
3	Lex	De Haan	Lex De Haan
4	Alexander	Hunold	Alexander Hunold
5	Bruce	Ernst	Bruce Ernst

#3 : Advance SQL



4. Aggregate Functions

- Agregat Functions berguna untuk mengkalkulasi sekumpulan angka (**data integer**) dan mengubahnya menjadi satu nilai/*value*.
- Beberapa contoh String Functions:

String Functions	Kegunaan	Rumus Query SQL
SUM	Untuk menjumlah total nilai dari suatu kolom.	<pre>SELECT SUM(column_name) FROM table_name WHERE condition;</pre>
AVG	Untuk mencari nilai rata-rata dari suatu kolom.	<pre>SELECT AVG(column_name) FROM table_name WHERE condition;</pre>
MIN	Untuk mencari nilai tekecil dari suatu kolom.	<pre>SELECT MIN(column_name) FROM table_name WHERE condition;</pre>
MAX	Untuk mencari nilai terbesar dari suatu kolom.	<pre>SELECT MAX(column_name) FROM table_name WHERE condition;</pre>
COUNT	Untuk menjumlah banyak data dari suatu tabel.	<pre>SELECT COUNT(column_name) FROM table_name WHERE condition;</pre>

#3 : Advance SQL

4. Aggregate Functions

- Contoh penggunaan functions:
 - “**SUM**” untuk mencari **jumlah gaji** yang harus dibayarkan oleh suatu perusahaan.
 - “**AVG**” untuk mencari **rata-rata** gaji karyawan di suatu perusahaan.
 - “**MIN**” untuk melihat **gaji terkecil** yang diterima oleh karyawan.
 - “**MAX**” untuk melihat **gaji terbesar** yang diterima oleh karyawan.
 - “**COUNT**” untuk mencari **berapa jumlah karyawan** di perusahaan tersebut.

Query

```
select sum(salary), avg(salary), min(salary), max(salary), count(salary)  
from datasource.job_employee je ;
```



Hasil

	sum	avg	min	max	count
1	691,400	6,461.6822429907	2,100	24,000	107



#3 : Advance SQL

5. Group By dan Order By Functions

- Fungsi **Group By** berfungsi untuk mengelompokkan baris dengan value yang sama kedalam suatu baris ringkasan.
- Fungsi **Order By** berfungsi untuk mengurutkan baris dari nilai terkecil ke terbesar atau sebaliknya.

- Rumus Query **GROUP BY**:

```
SELECT column1,...  
FROM table_name  
WHERE condition  
GROUP BY column1,...
```

- Rumus Query **ORDER BY**:

```
SELECT column1,...  
FROM table_name  
WHERE condition  
ORDER BY column1,... ASC
```

ASC = Ascending

- Untuk mengurutkan hasil Query dari nilai tekecil hingga nilai terbesar

```
SELECT column1,...  
FROM table_name  
WHERE condition  
ORDER BY column1,... DESC
```

DESC = Descending

- Untuk mengurutkan hasil Query dari nilai terbesar hingga nilai terkecil



#3 : Advance SQL



5. Group By dan Order By Functions

- Contoh penggunaan Fungsi **Group By** dan **Order By**
 - "GROUP BY"** digunakan untuk mengelompokkan nilai gaji rata-rata dari setiap department.
 - "ORDER BY...DESC"** digunakan untuk mengetahui departemen yang memiliki gaji rata-rata tertinggi hingga terendah

```
select job_title, job_id, avg(salary)
from datasource.job_employee je
group by job_title, job_id
order by avg(salary) DESC;
```

Query

Hasil



	ABC job_title	ABC job_id	123 avg
1	President	AD_PREP	24,000
2	Administration Vice President	AD_VP	17,000
3	Marketing Manager	MK_MAN	13,000
4	Sales Manager	SA_MAN	12,200
5	Accounting Manager	AC_MGR	12,000
17	Shipping Clerk	SH_CLERK	3,215
18	Stock Clerk	ST_CLERK	2,785
19	Purchasing Clerk	PU_CLERK	2,780

#3 : Advance SQL



6. Limit Functions

- Fungsi **LIMIT** digunakan untuk membatasi secara khusus hasil dari Query.

- Rumus Query **LIMIT**:

```
SELECT column1,...  
FROM table_name  
WHERE condition  
ORDER BY column1,...  
LIMIT number
```

- Contoh penggunaan Fungsi **LIMIT**

- “**ORDER BY...DESC**” digunakan untuk mengetahui departemen yang memiliki gaji rata-rata tertinggi hingga terendah
- “**LIMIT**” digunakan untuk melihat **3 department** yang memiliki gaji rata-rata tertinggi.

```
select job_title, job_id, avg(salary)  
from datasource.job_employee je  
group by job_title, job_id  
order by avg(salary) desc  
limit 3;
```

Query

Hasil

	ABC job_title	ABC job_id	123 avg
1	President	AD_PRES	24,000
2	Administration Vice President	AD_VP	17,000
3	Marketing Manager	MK_MAN	13,000



Thank you !

Date: Monday, 20-9-2021