

# Learning Progress Review

**WEEK 3 - BY CHARLIE'S ANGELS**



# CRUD

<b>C</b>	<b>=</b>	<b>Create</b>	<b>=</b>	Adalah proses untuk pembuatan suatu database
<b>R</b>	<b>=</b>	<b>Read</b>	<b>=</b>	Adalah proses menampilkan database
<b>U</b>	<b>=</b>	<b>Update</b>	<b>=</b>	Adalah proses merubah database
<b>D</b>	<b>=</b>	<b>Delete</b>	<b>=</b>	Adalah proses menghapus database

## CRUD Adalah

Proses yang memiliki keterkaitan dengan Transaksi atau Pengambilan data dari dan atau ke Database

## Contohnya :

```
CREATE TABLE persons (  
  person_id int,  
  last_name varchar(255),  
  first_name varchar(255),  
  phone_number int,  
  created_at timestamp);
```

```
INSERT INTO persons  
(last_name, first_name,  
  phone_number, created_at)  
VALUES ('Cardinal', 'Stavanger',  
  123456, NOW());
```

```
ALTER TABLE persons  
ADD email varchar(255);
```

```
ALTER TABLE persons  
DROP COLUMN email;
```

```
UPDATE persons  
SET  
  last_name = 'Alfred Schmidt',  
  first_name = 'Frankfurt'  
WHERE person_id = 2;
```

```
SELECT first_name, last_name  
FROM persons;
```

```
SELECT * FROM persons;
```

```
TRUNCATE TABLE persons;
```

```
DROP TABLE persons;
```

# Primary Key

Dalam SQL database management system, primary key adalah tanda pengenal unik yang membedakan satu record dari yang lain. Oleh karena itu, setiap record dalam SQL database management system harus memiliki primary key.

Beberapa aturan yang harus diikuti ketika menentukan primary key untuk tabel:

- Primary key harus berisi nilai unik. Jika primary key terdiri dari beberapa kolom, kombinasi value pada kolom ini harus unik.
- Kolom primary key tidak boleh berisi nilai NULL. Ini berarti bahwa kalian harus mendeklarasikan kolom primary key dengan atribut NOT NULL. Jika tidak, MySQL akan memaksa kolom primary key sebagai NOT NULL secara implisit.
- Sebuah tabel hanya memiliki satu primary key

Primary key + Opsi			
IDPelanggan	NamaDepan	NamaBelakang	NomorTelepon
0001	Johnny	Cage	081-111-222-333
0002	Aric	Norma	081-444-555-666
0003	David	Luiz	081-777-888-999
0004	Johnny	Cage	081-999-888-777

# Foreign Key

foreign key adalah pengenal unik atau kombinasi pengenal unik yang menghubungkan dua tabel atau lebih dalam database. Dari empat SQL database management system yang ada, relational database management system adalah yang paling populer.

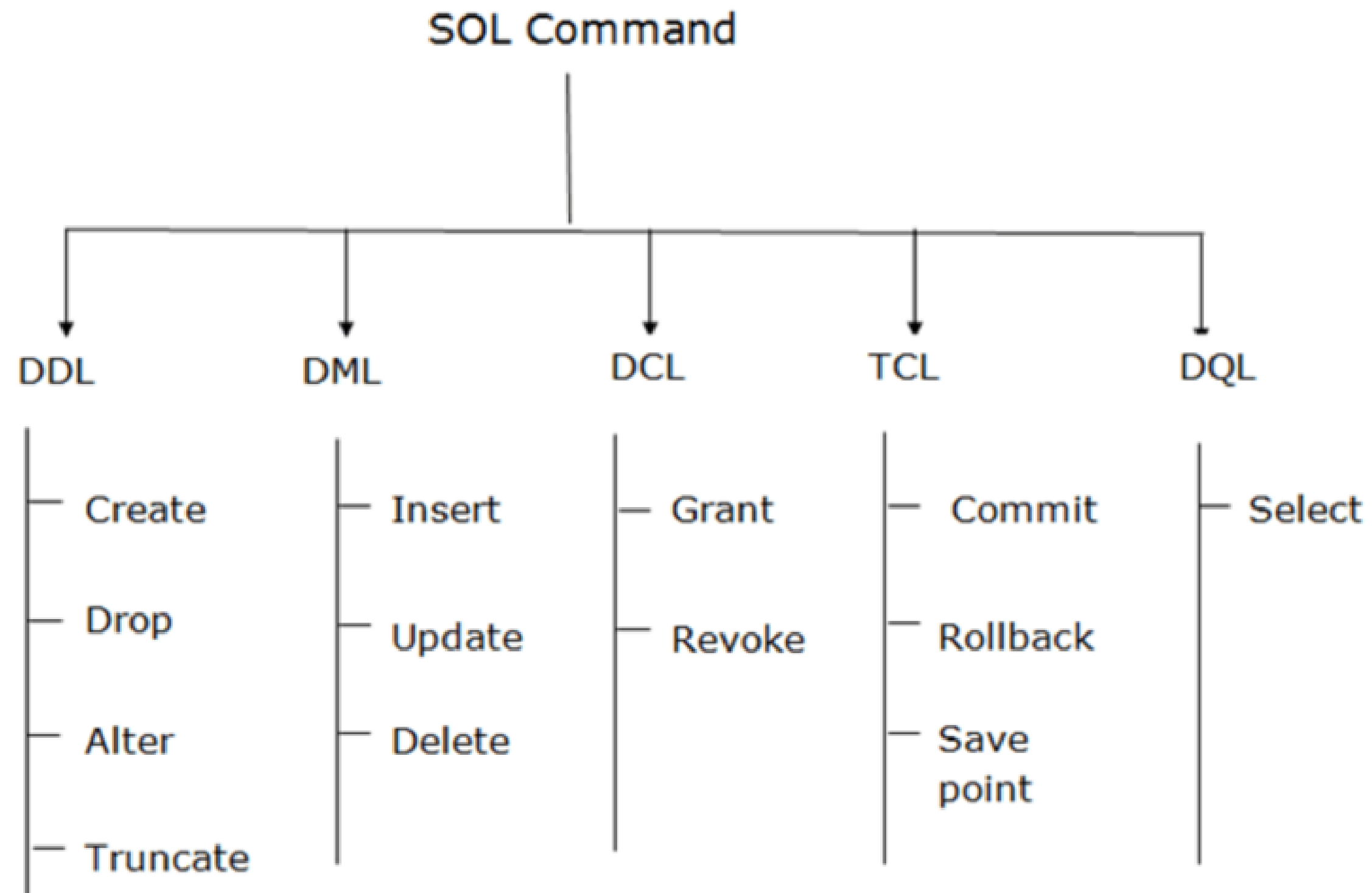
## Kesimpulan

Jadi apa itu primary key dan foreign key? Primary key adalah aturan-aturan yang berguna agar setiap baris data atau nilai bersifat unik, agar masing-masing baris berbeda satu sama lain. Sedangkan foreign key adalah satu attribute (atau satu set attribute) yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya.

Foreign Key			
+ Opsi	NamaDepan	NamaBelakang	NomorTelepon
IDPelanggan	Johnny	Cage	081-111-222-333
0001	Aric	Norma	081-444-555-666
0002	David	Luiz	081-777-888-999
0003	Johnny	Cage	081-999-888-777
0004			

# SQL Command

- **DDL**
  - **Data Definition Language**, untuk membuat, menghapus, dan mengubah table
- **DML**
  - **Data Manipulation Language**, untuk memodifikasi database. Command DML tidak bersifat auto commit dan dapat di rollback
- **DCL**
  - **Data Control Language**, untuk memberikan dan menghapus permit dari setiap database user
- **TCL**
  - **Transaction Control Language**, hanya dapat digunakan dengan command DML, bersifat auto commit
- **DQL**
  - **Data Query Language**, digunakan untuk mengambil data dari database



[www.javatpoint.com/dbms-sql-command](http://www.javatpoint.com/dbms-sql-command)

# WHERE CLAUSE

- Untuk filter hasil record sesuai kondisi yang ditentukan

## Syntax

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

- Dalam penggunaanya dapat digabung dengan operator berikut



Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. <b>Note:</b> In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Source: [https://www.w3schools.com/mysql/mysql\\_where.asp](https://www.w3schools.com/mysql/mysql_where.asp)



# Example Where Clause

- DB Table 'tes'

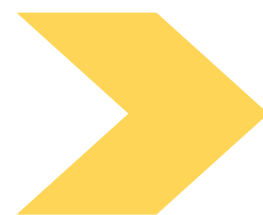
123 person_id 🔼🔼	ABC last_name 🔼🔼	ABC first_name 🔼🔼	123 phone_number 🔼🔼	🕒 created_at 🔼🔼	123 salary 🔼🔼	ABC gender 🔼🔼
1	Afta	Beta	123,456	2022-05-29 17:39:35.669	3,500,000	P
2	Charlie	Delta	126,456	2022-05-29 17:39:35.669	1,000,000	L
3	Elf	Fiko	123,459	2022-05-29 17:39:35.669	2,000,000	L
4	Jeje	Kiku	123,556	2022-05-29 17:39:35.669	3,500,000	P
5	Lea	Mia	223,456	2022-05-29 17:39:35.669	4,000,000	P
6	Nano	Olaf	553,456	2022-05-29 17:39:35.669	3,200,000	L

- Query

```
SELECT first_name , last_name FROM tes WHERE last_name like '%a%' and salary > 3500000;
```

- Result

ABC first_name 🔼🔼	ABC last_name 🔼🔼
Mia	Lea



Hasil hanya menampilkan record dengan last\_name yang memiliki huruf 'a' dan salary diatas 3500000 sesuai dengan kondisi where yang diberikan



# SORTING & GROUPING

- Untuk mengurutkan dan mengelompokkan hasil record

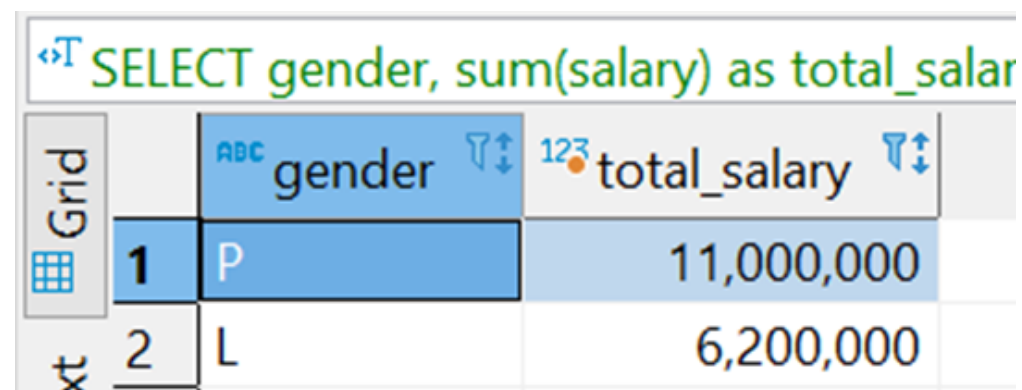
## Syntax

```
• SELECT column1, column2,.....  
  FROM table_name  
  WHERE [ conditions ]  
  GROUP BY column1, column2....  
  ORDER BY column1, column2....
```

- Example (using database table 'tes' in the previous slide)
  - Query

```
SELECT gender, sum(salary) as total_salary  
FROM tes  
group by gender  
order by length(sum(salary)) desc, sum(salary) desc
```

- Result



Grid	gender	total_salary
1	P	11,000,000
2	L	6,200,000

Hasil menampilkan jumlah salary yang sudah dikelompokkan berdasarkan genre dan diurutkan berdasarkan total salary terbesar

# VIEW

- Untuk membuat table virtual yang menampilkan hasil query

## Syntax

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Membuat Virtual tabel

```
drop view total_salary_per_gender;
```

Menghapus virtual table yang telah dibuat

- Example (using database table 'tes' in the previous slide)

```
CREATE VIEW total_salary_per_gender AS  
SELECT gender, sum(salary) as total_salary  
FROM tes  
group by gender  
order by length(sum(salary)) desc, sum(salary)  
desc
```

```
select * from total_salary_per_gender
```

Membuat table virtual dengan nama table  
total\_salary\_per\_gender

Memanggil virtual table yang telah dibuat

# Indexes



Index adalah sebuah objek dalam sistem database yang dapat mempercepat proses pencarian (query) data.

## Clustered Index

- Index dibuat berdasarkan primary key
- 1 table hanya bisa memiliki 1 clustered index
- Record pada table diurutkan ulang sesuai index

## Non-clustered Index

- 1 table bisa memiliki beberapa non clustered index
- Record pada table diurutkan ulang secara logical

## Partial Index

- Meningkatkan performansi query sambil mengurangi size index
- Berguna saat query menggunakan where clause yang bernilai konstan

# Indexes

## Expression Index

- Dikenal juga sebagai functional based index
- Digunakan untuk membuat index berdasarkan expression pada sebuah kolom table
- Digunakan saat retrieval speed lebih penting daripada update dan insert speed

## Unique Index

- Membuat value column menjadi unique
- Column tidak bisa menyimpan beberapa row dengan nilai yang sama

## Multi Column Index

- Memungkinkan index untuk dibuat lebih dari 1 kolom pada sebuah table
- Secara default maksimal multicolumn index pada sebuah table adalah 32

# Partitioning



Partitioning adalah sebuah proses dimana data pada table yang sangat besar dikelompokkan menjadi beberapa bagian yang lebih kecil untuk mempercepat query

## List Partition

- Data dikelompokkan berdasarkan nilainya
- Cocok untuk data yang tidak memiliki banyak variasi data, ex : area

## Range Partition

- Data dikelompokkan berdasarkan rentang nilai yang ditentukan
- Cocok untuk kolom yang valuenya terdistribusi secara merata, ex : tanggal

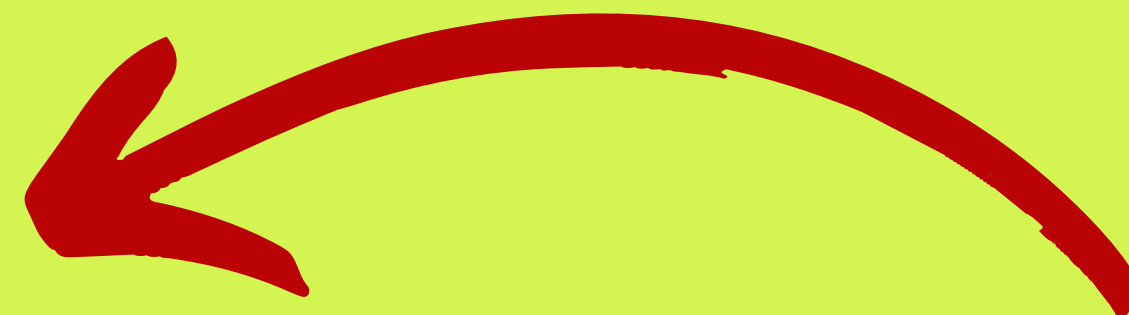
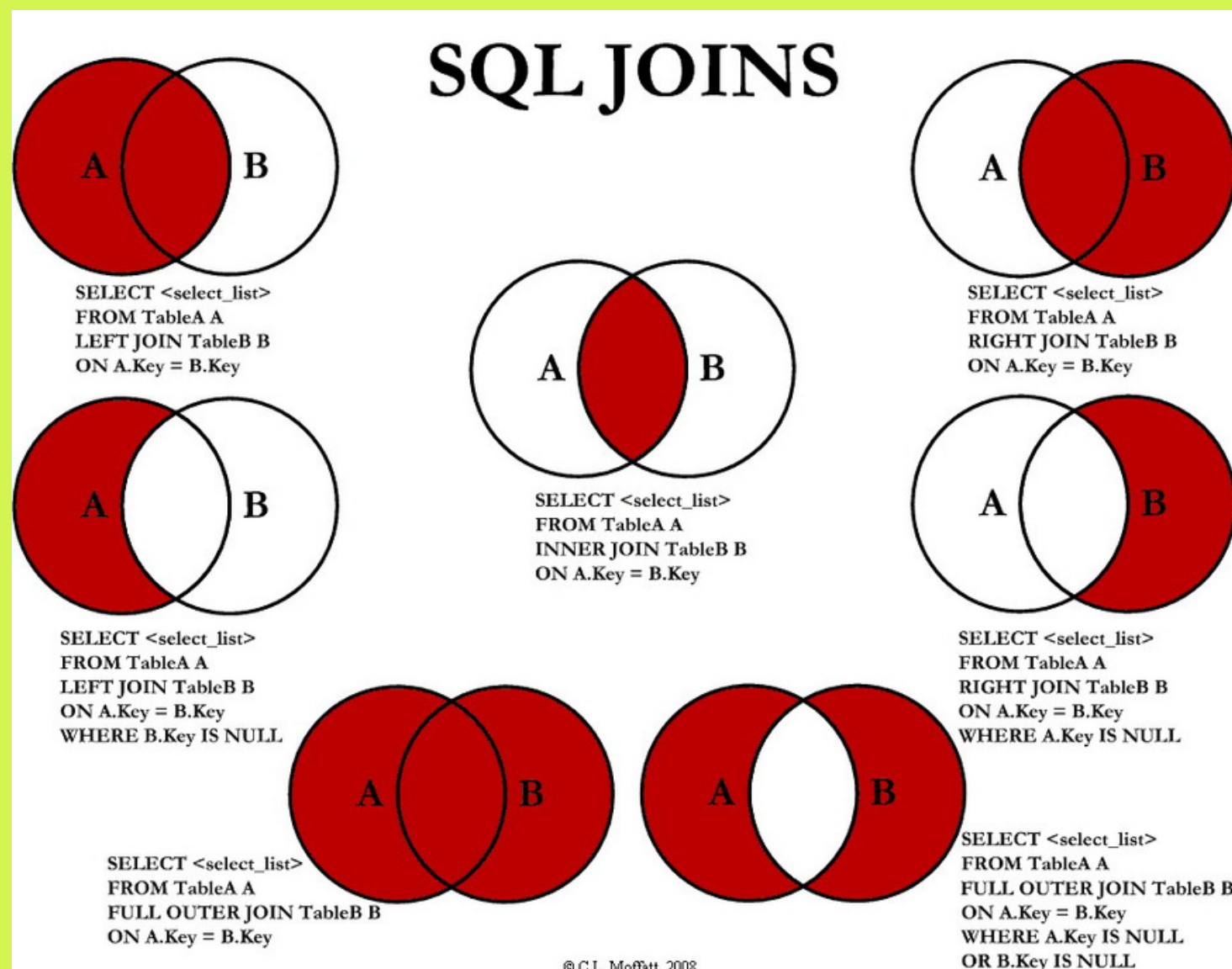
## Hash Partition

- Data dikelompokkan berdasarkan algoritma hash
- Tidak cocok untuk mengatur historical data karena sifat partitioningnya yang random



# SQL JOIN

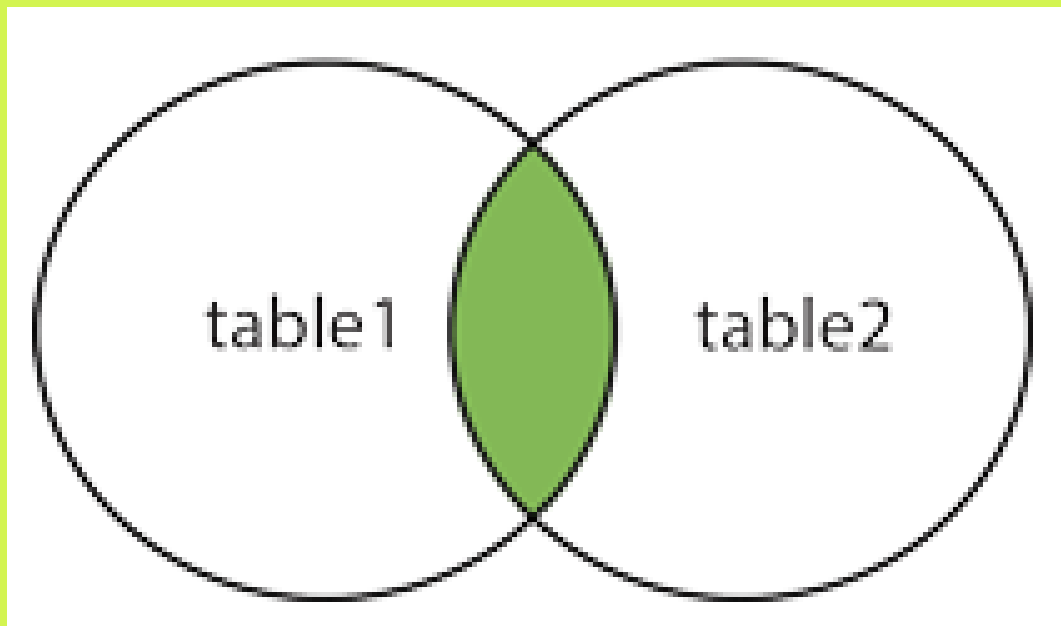
SQL JOIN adalah salah satu perintah di dalam SQL. Perintah JOIN ini digunakan untuk menggabungkan data dari dua atau lebih tabel sehingga menjadi satu. Untuk dapat melakukan perintah JOIN syaratnya adalah terdapat kolom yang mempunyai hubungan diantara tabel-tabel tersebut.



**SQL JOIN COMMAND**

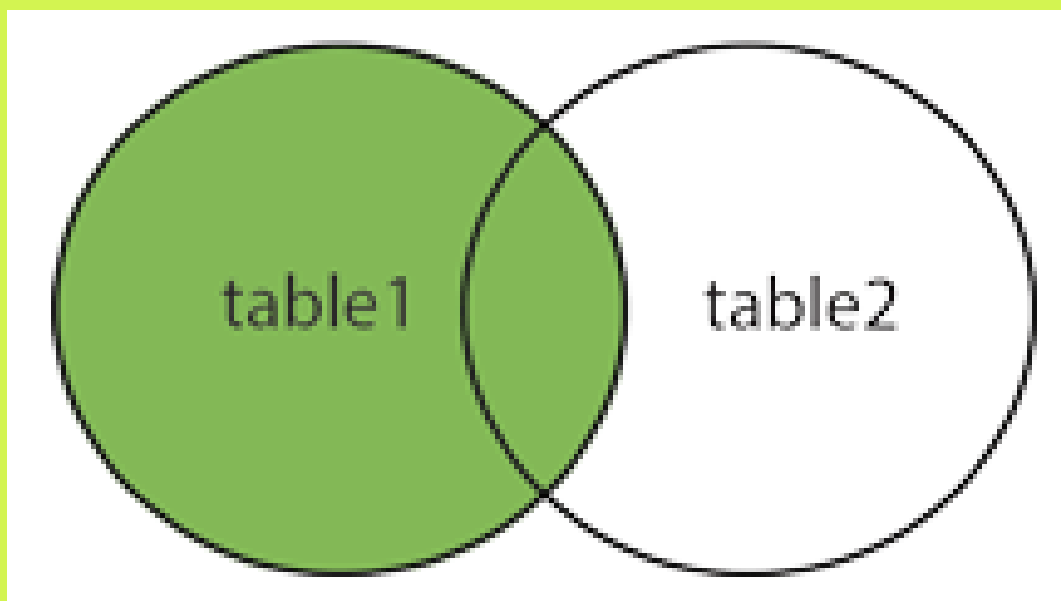
## Beberapa Jenis SQL JOIN yang sering digunakan

### INNER JOIN



Bentuk join yang hanya akan mengambil data yang beririsan saja untuk masing-masing tabel. Sementara untuk data yang tidak sama untuk kedua tabel akan diabaikan.

### LEFT JOIN

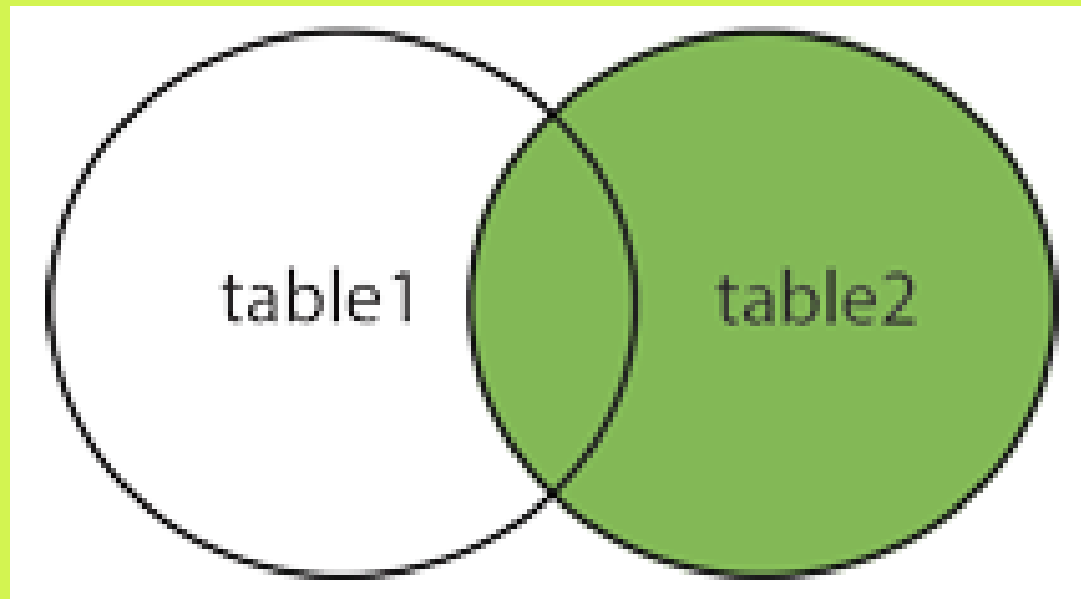


Left Join merupakan cara menghubungkan tabel dan menampilkan semua data (kiri) pada tabel yang tidak berhubungan, sedangkan data yang kosong akan bernilai NULL.



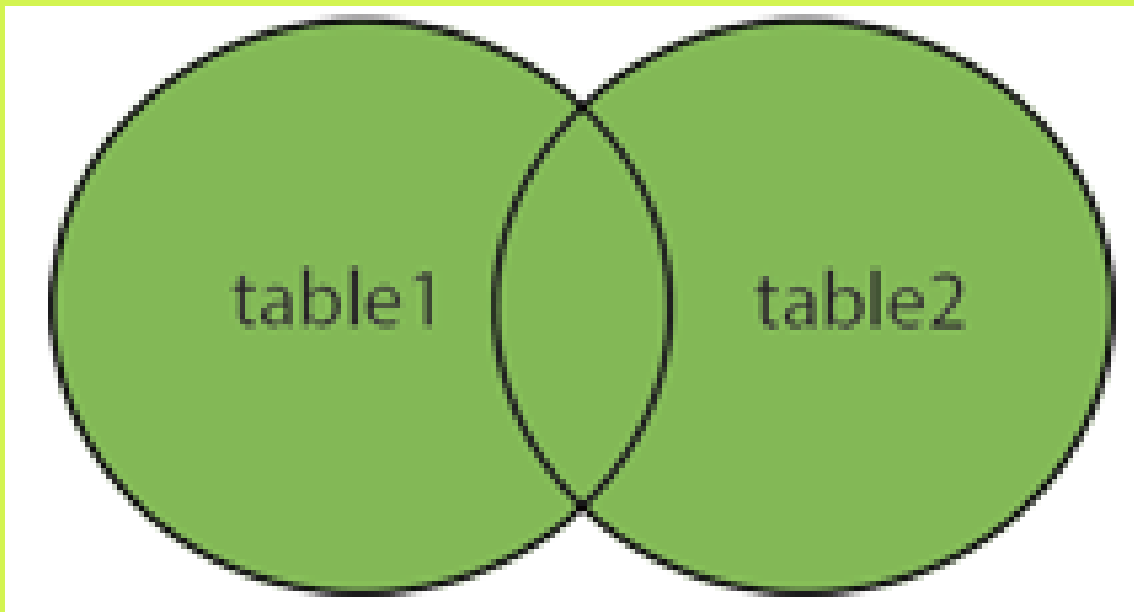
## Beberapa Jenis SQL JOIN yang sering digunakan

### RIGHT JOIN



Right Join akan menghubungkan tabel dan menampilkan semua data (kanan) pada tabel yang tidak berhubungan, dan data yang kosong akan bernilai NULL.

### OUTER JOIN



Pada Outer Join, semua data baik yang ada di tabel yang ada di sebelah kiri maupun tabel yang ada di sebelah kanan akan dimasukkan semuanya. Kemudian untuk data yang kosong di masing-masing tabel akan diberi nilai Null.

# ●●● SQL Subquery

Subquery adalah query nested atau sebuah query yang ada di dalam query. Subquery merupakan pernyataan SELECT yang merupakan bagian dari pernyataan INSERT, SELECT. Subquery digunakan untuk menangani masalah dalam query yang kompleks bahkan kita tidak tahu nilai berapa yang akan di select atau di insert.

## Query dasar dalam subquery :

```
SELECT namakolom  
FROM namatabel  
WHERE namakolom operator (subquery);
```

## Aturan dalam penggunaan Subqueries

Subqueries harus tertutup dalam tanda kurung.

Sebuah subquery hanya dapat memiliki satu kolom pada klausa SELECT, kecuali beberapa kolom yang di query utama untuk subquery untuk membandingkan kolom yang dipilih.

ORDER BY tidak dapat digunakan dalam subquery, meskipun permintaan utama dapat menggunakan ORDER BY. GROUP BY dapat digunakan untuk melakukan fungsi yang sama seperti ORDER BY dalam subquery.

Subqueries yang kembali lebih dari satu baris hanya dapat digunakan dengan beberapa value operator, seperti operator IN.

Sebuah subquery tidak dapat segera tertutup dalam fungsi set.

Operator BETWEEN tidak dapat digunakan dengan subquery; Namun, BETWEEN dapat digunakan dalam subquery.

# Partial Results

## LIMIT

Limit adalah salah satu perintah di SQL yang digunakan untuk membatasi jumlah record yang akan ditampilkan.

```
SELECT column1, column2, columnN  
FROM table_name  
LIMIT [no of rows]
```

## OFFSET

offset digunakan untuk melewati baris sebelum mengembalikan hasil query.

```
SELECT column_name1, ..., column_nameN  
FROM table_name  
OFFSET N (Number of rows that we have  
skipping in query result)
```

## FETCH

Fetch digunakan dengan klausa order by mengambil bagian dari baris yang dikembalikan oleh pernyataan dari cursor hingga cursor mencapai kondisi terminasi disebutkan dalam perintah.

```
SELECT column_name(s)  
FROM table_name  
ORDER BY column_name  
FETCH NEXT terminating_point ROWS  
ONLY;
```

# ●●● Visualisasi Data

Data Visualization adalah representasi grafis dari informasi dan data. Dengan menggunakan elemen visual seperti bagan, grafik, dan peta, alat visualisasi data menyediakan cara yang dapat diakses untuk melihat dan memahami tren, outlier, dan pola dalam data.



**Contoh visualisasi data**

## ●● Tujuan

Tujuan utama dari visualisasi data adalah untuk memudahkan dalam mengidentifikasi pola, tren, dan outlier dalam kumpulan data yang besar

Visualisasi data merupakan salah satu langkah dari proses data science, yang menyatakan bahwa setelah data dikumpulkan, diproses dan dimodelkan, harus divisualisasikan untuk dapat diambil kesimpulan.

## ●● Tujuan

Tujuan utama dari visualisasi data adalah untuk memudahkan dalam mengidentifikasi pola, tren, dan outlier dalam kumpulan data yang besar

Visualisasi data merupakan salah satu langkah dari proses data science, yang menyatakan bahwa setelah data dikumpulkan, diproses dan dimodelkan, harus divisualisasikan untuk dapat diambil kesimpulan.



# ••• **Mengapa Visualisasi data itu penting?**

Visualisasi data saat ini diperlukan di hampir setiap pekerjaan. misalnya di sekolah, visualisasi data diperlukan oleh guru untuk menampilkan ranking dari siswa. atau oleh eksekutif yang ingin berbagi informasi dengan stakeholder

Visualisasi data juga memiliki peran penting dalam big data. karena bisnis sedang mengumpulkan data yang sangat besar selama tahun tahun awal trend big data, mereka membutuhkan cara untuk mendapatkan overview/summary dari data mereka. itulah mengapa visualisasi data diperlukan

# Data visualization timeline



**1644**  
Flemish astronomer Michael Florent van Langren provides the first representation of statistical data.

**1600s**

**1700s**  
Thematic mapping emerged and abstract graphs of functions, measurement errors and the collection of empirical data were introduced.

**1700s**



**1800s**  
William Playfair, among others, introduced some of today's most popular graphs and various statistical chart types were invented.

**1854**  
Physician John Snow maps the outbreaks of cholera that occurred across London during the 1854 epidemic.



PIE CHART FROM WILLIAM PLAYFAIR'S "STATISTICAL BREVIARY"

**1800s**

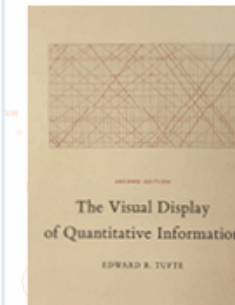
**1900s**

**Early 1900s**  
Statisticians are less concerned with data visualization and more focused on exact numbers. Simultaneously, data visualization gains public popularity, and charts and graphs start appearing in textbooks and business applications.

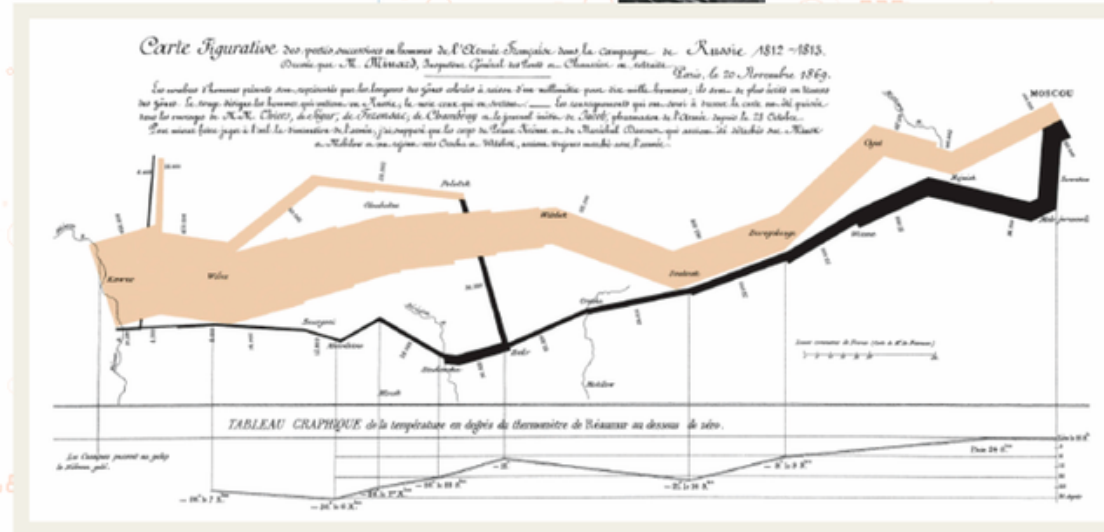
**Late 1900s**  
The emergence of computer processing allows statisticians to collect, store and efficiently visualize larger volumes of data.

**1960s-1970s**  
Researchers John W. Tukey and Jacques Bertin develop the science of data visualization in statistics and cartography, respectively.

**Early 1980s**  
Edward Tufte publishes *The Visual Display of Quantitative Information*, which is currently used in university courses.



**1869**  
Charles Joseph Minard charts the number of men in Napoleon's 1812 Russian army.



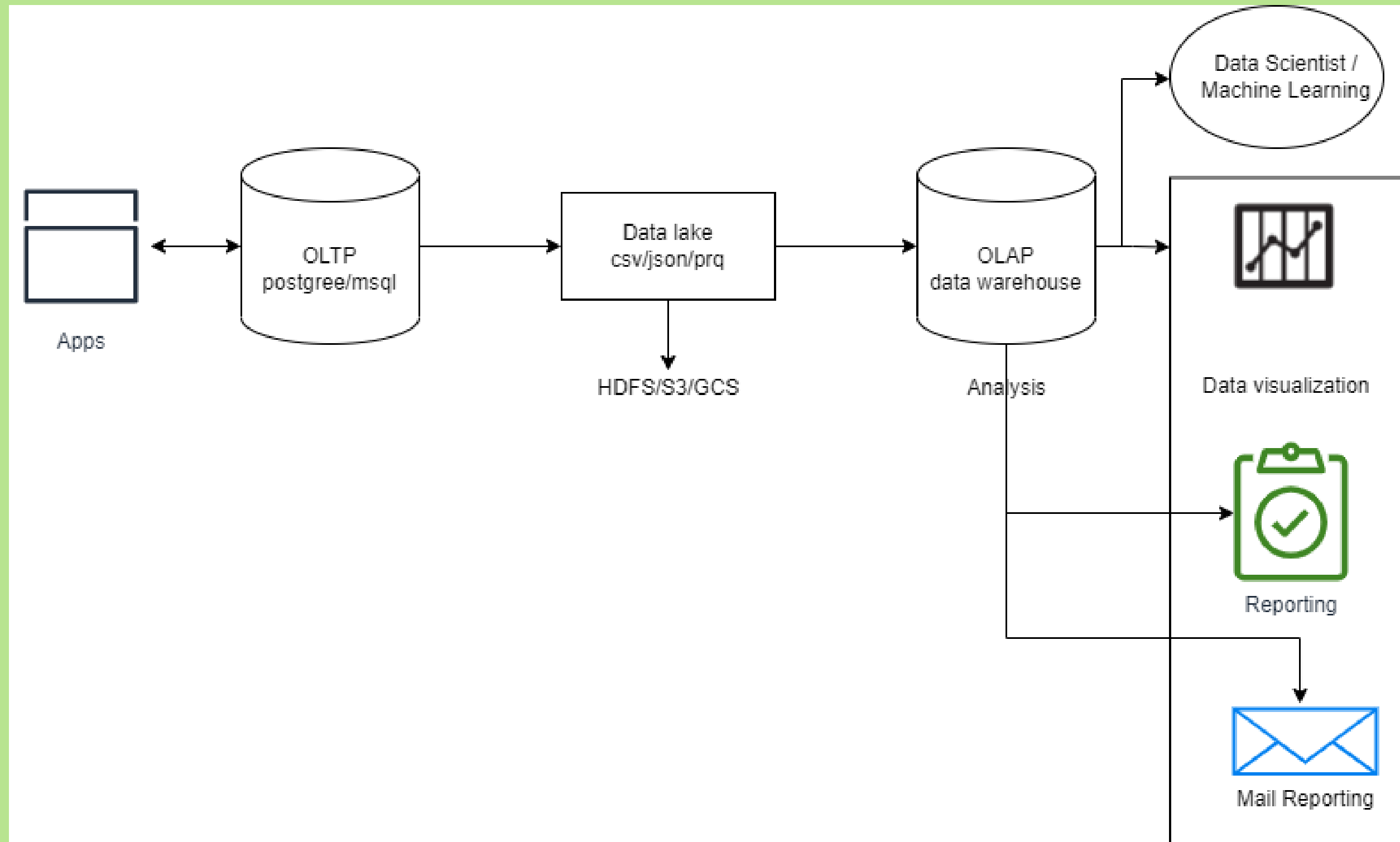
peningkatan kemampuan untuk mempertahankan minat audiens dengan informasi yang dapat mereka pahami

distribusi informasi yang mudah yang meningkatkan kesempatan untuk berbagi wawasan dengan stakeholder

menghilangkan kebutuhan data scientist karena data lebih mudah diakses dan dimengerti

peningkatan kemampuan untuk bertindak berdasarkan temuan dengan cepat dan, oleh karena itu, mencapai kesuksesan dengan kecepatan yang lebih besar dan lebih sedikit kesalahan

# ●●● Whiteboard Business Intelligent



# ●●● **Membuat Dashboard (Data studio)**

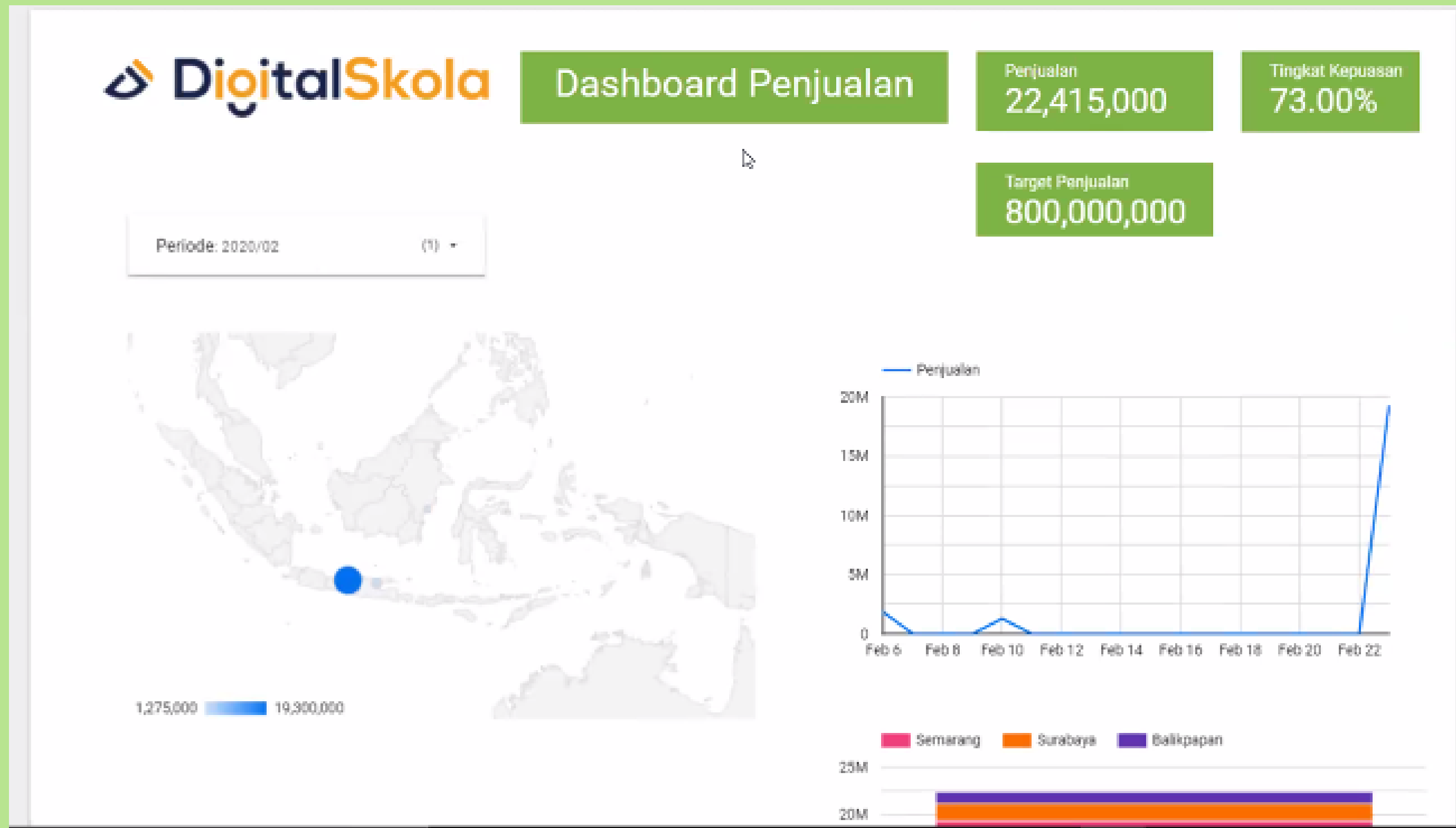
<https://datastudio.google.com/>

1. simpan file csv di googledrive
2. file -> save as google sheet (file yg diupload di googledrive)
3. masuk data studio -> create blank report
4. pilih google sheets
5. all items -> gdcs\_spaces -> sheet1 -> add
6. add to report

# ●●● **Membuat Dashboard (Data studio)**

- 1.add control -> drop down list
- 2.Add chart -> geo chart
- 3.Resource -> Manage Added Data Sources
- 4.Kota -> ganti dengan Geo -> city
- 5.done -> close
- 6.Tarik kota ke Geo Dimension
- 7.Zoom Area -> pilih Indonesia
- 8.Add Chart -> Score Card
- 9.tingkat kepuasan -> percent

# ●●● Membuat Dashboard (Data studio)





# Terima Kasih

Sampai jumpa di  
Learning Progress Review  
kami berikutnya!

Hilda Meiranita Prastika Dewi

Nur Indrasari

Rezha Sulvian

Thasha Dinya Ainsha