

Advance SQL (using Snowflake)





What will we learn today?

- Best Practice for Query
- Join Operation
- Sorting and grouping
- Function



Best Practice for query



Best Practice



- Dengan mengikuti kaidah format penulisan yang baik, akan memudahkan query di baca dan dimengerti.
- Konsistensi dalam penulisan Query
- SQL Syntax gunakan UPPER CASE
- Untuk major Syntax, buat menjadi baris baru
 - o SELECT
 - o WHERE
 - o FROM
 - o GROUP BY
 - o ORDER BY
 - o LIMIT



Best Practice



- Tambah indentasi untuk Syntax seperti berikut
 - o Sub-query
 - o ON
 - o AND/OR
 - o Beberapa kolom yang mau di ekstraksi
- Gunakan alias untuk query Panjang / sub query
- Selalu tutup query dengan titik-koma(;) sebagai penutup.



Best Practice



- Tambahkan komentar dalam query untuk memberikan informasi mengenai Sebagian atau keseluruhan dari query yang ditulis
- Komentar ditulis dengan menggunakan tanda berikut
 - o -- untuk satu baris
 - o /* untuk komentar banyak baris */



Best Practice



- Tambahkan komentar dalam query untuk memberikan informasi mengenai Sebagian atau keseluruhan dari query yang ditulis
- Komentar ditulis dengan menggunakan tanda berikut
 - o -- untuk satu baris
 - o /* untuk komentar banyak baris */



Example



```
1  SELECT
2      county, total_population, total_male_population,
3      total_female_population
4  FROM
5      covid19_epidemiological_data.public.demographics
6  WHERE
7      state='TX'
8  ORDER BY
9      total_male_population DESC;
```

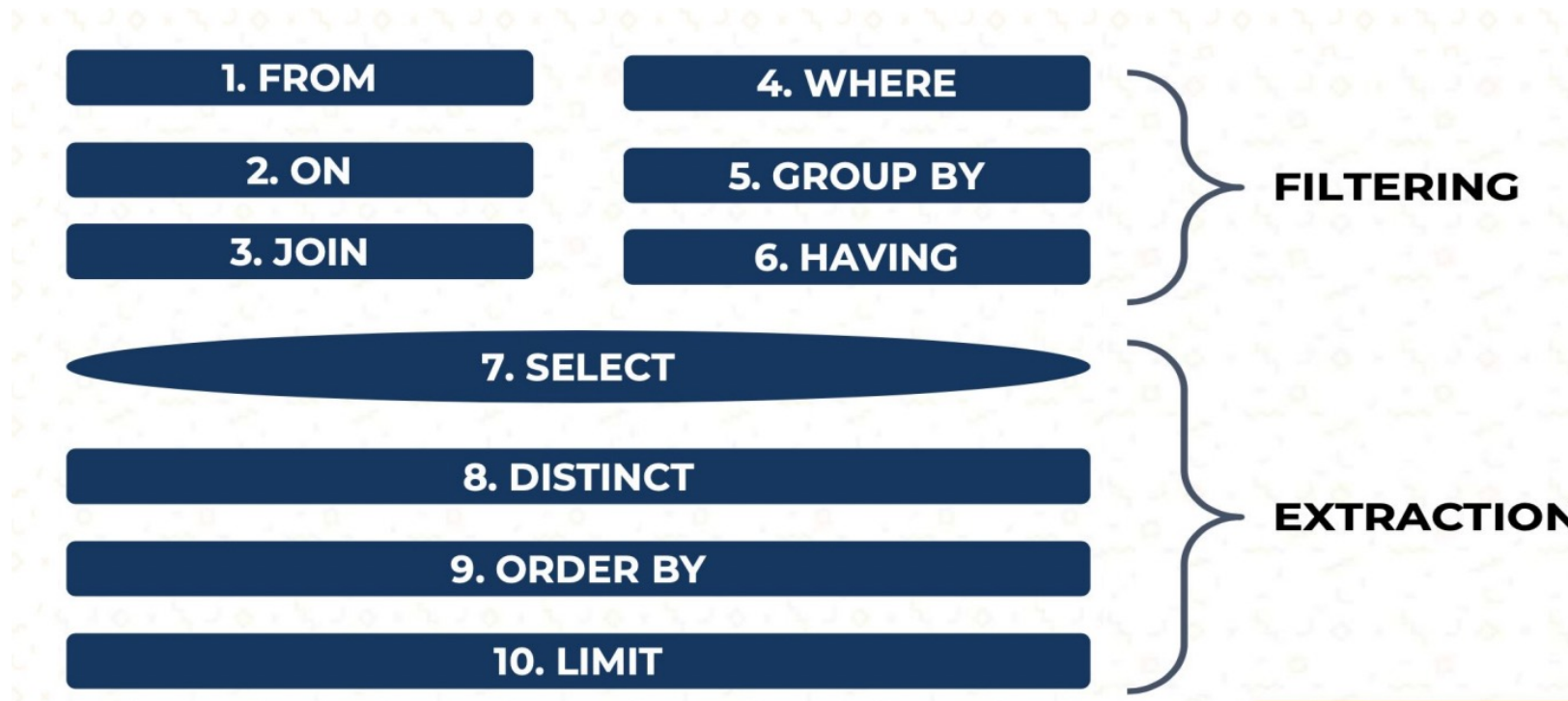



Urutan pemrosesan





Urutan pemrosesan





DB Clustering



proses menggabungkan lebih dari satu server atau instance yang menghubungkan satu database

- Failover/High Availability clusters

Cluster menyiapkan ketersediaan layanan dengan mereplikasi server dan dengan konfigurasi ulang perangkat lunak dan perangkat keras yang berlebihan. Jadi, setiap sistem mengendalikan yang lain dan bekerja berdasarkan permintaan jika salah satu node gagal. Jenis cluster ini menguntungkan bagi para pengguna yang bergantung pada sistem komputer mereka sepenuhnya. Misalnya e-commerce, website, dll.

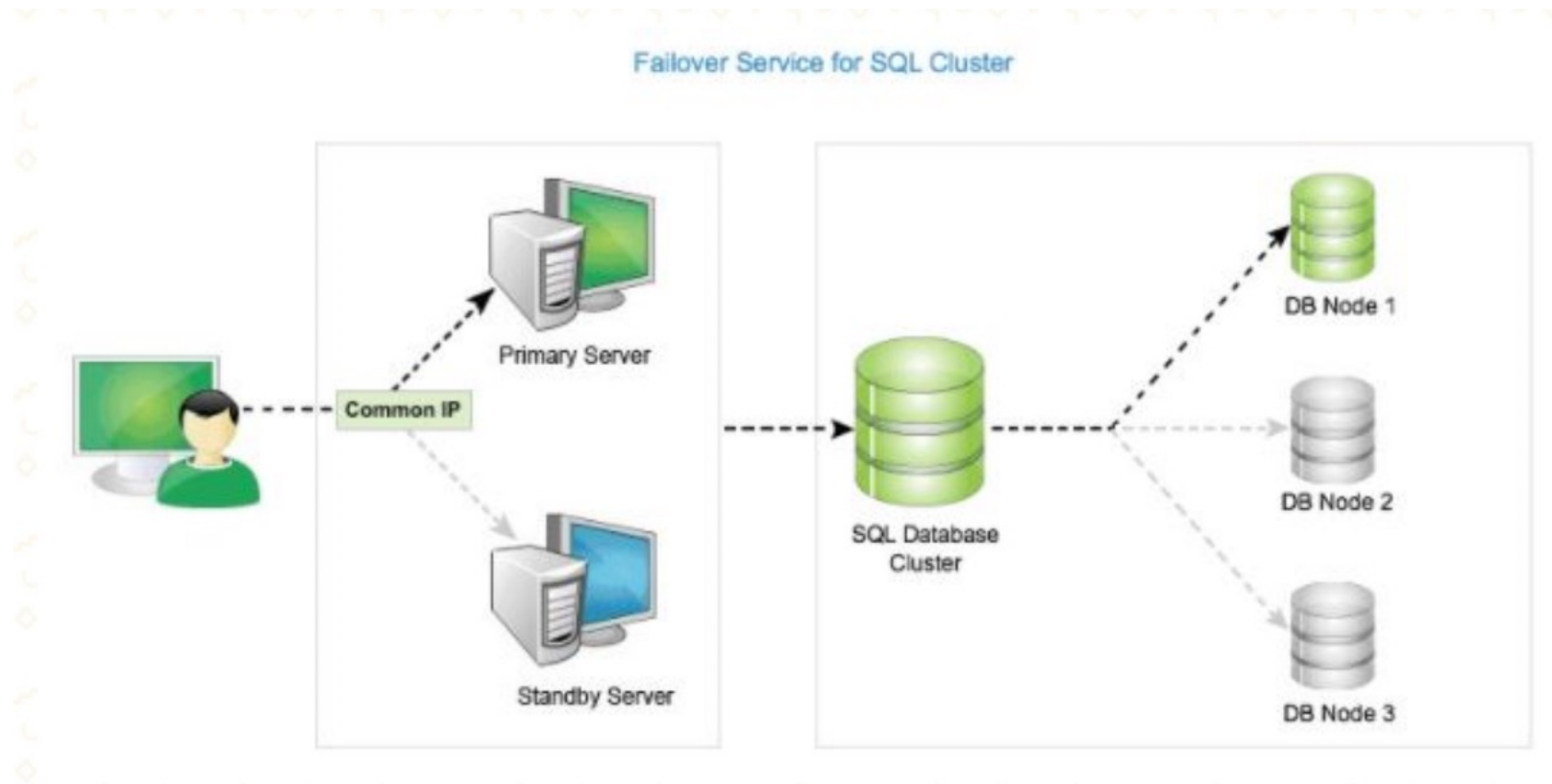
- High-Performance Clusters

Tujuan pengembangan cluster database berkinerja tinggi adalah untuk menghasilkan sistem komputer berkinerja tinggi. Mereka mengoperasikan co-extending program yang diperlukan untuk perhitungan yang menghabiskan waktu. Biasanya digunakan oleh perusahaan yang memerlukan perhitungan / workload tinggi seperti AI Industri dll.

- Load Balancing Clusters

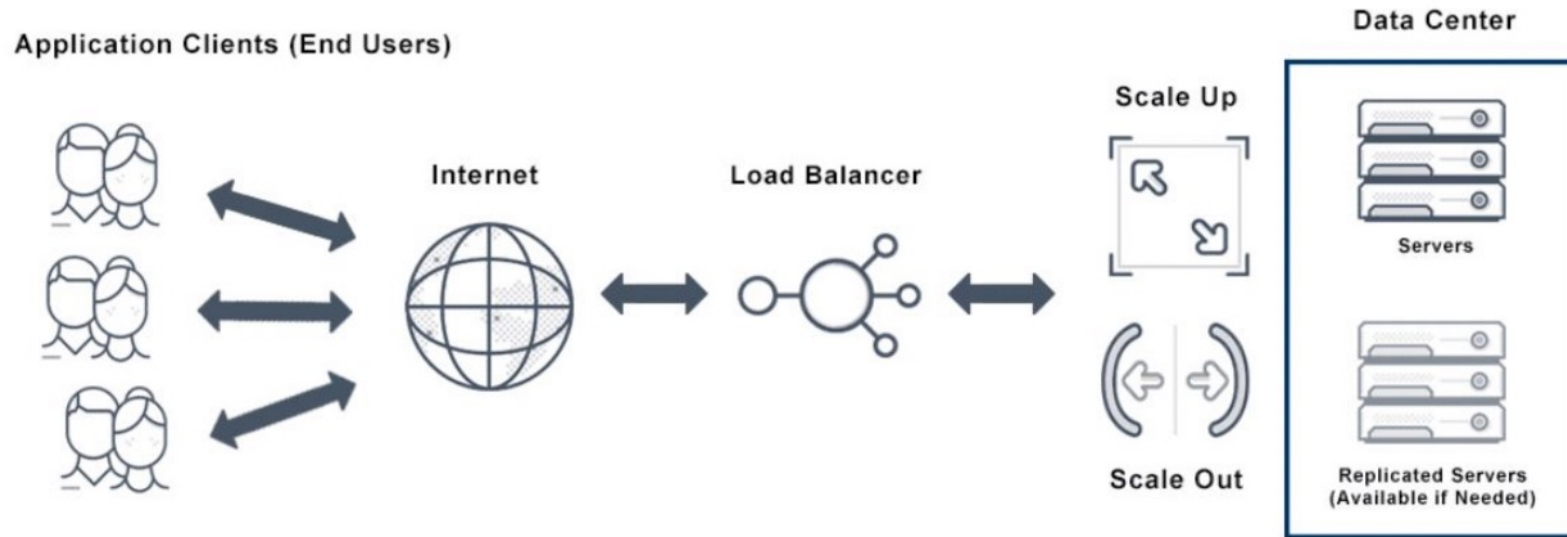
Cluster database ini berfungsi untuk mendistribusikan beban antara server yang berbeda. Sistem dalam jaringan ini mengintegrasikan node mereka, dengan bantuan permintaan pengguna yang dibagi rata di seluruh node yang berpartisipasi. Sistem tidak bekerja bersama-sama, melainkan mengarahkan permintaan secara individual saat terjadi

HA Cluster



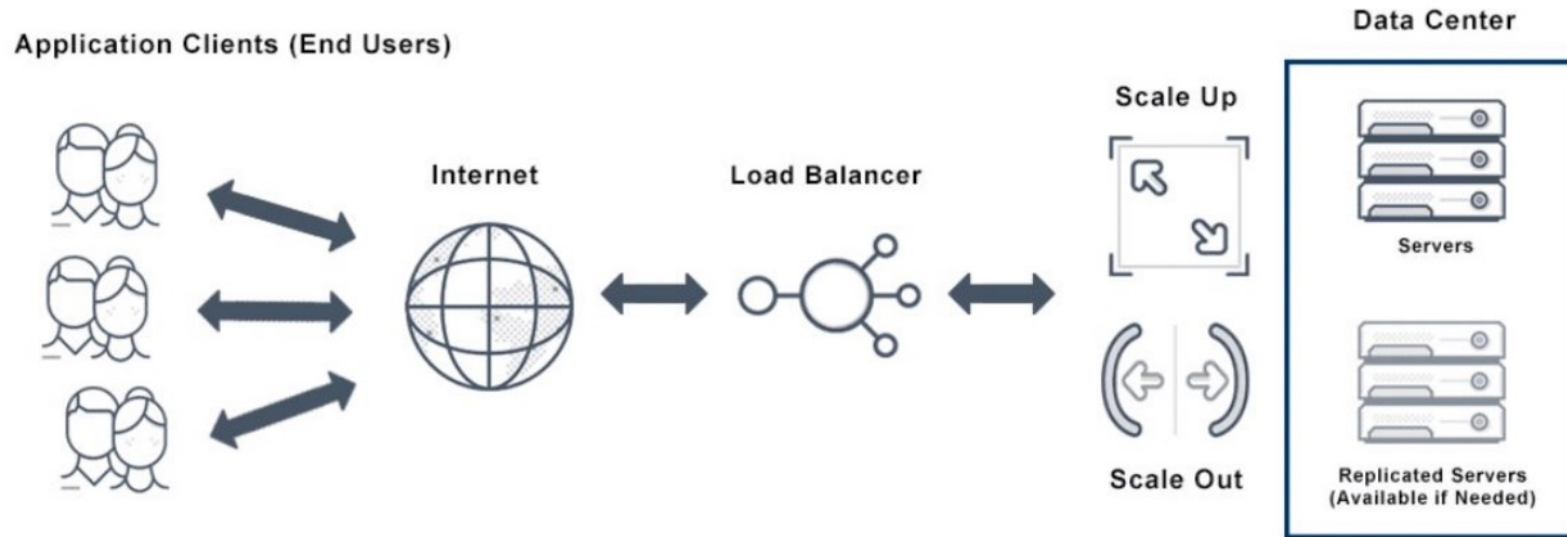


High Performance Cluster





High Performance Cluster



Load Balancing

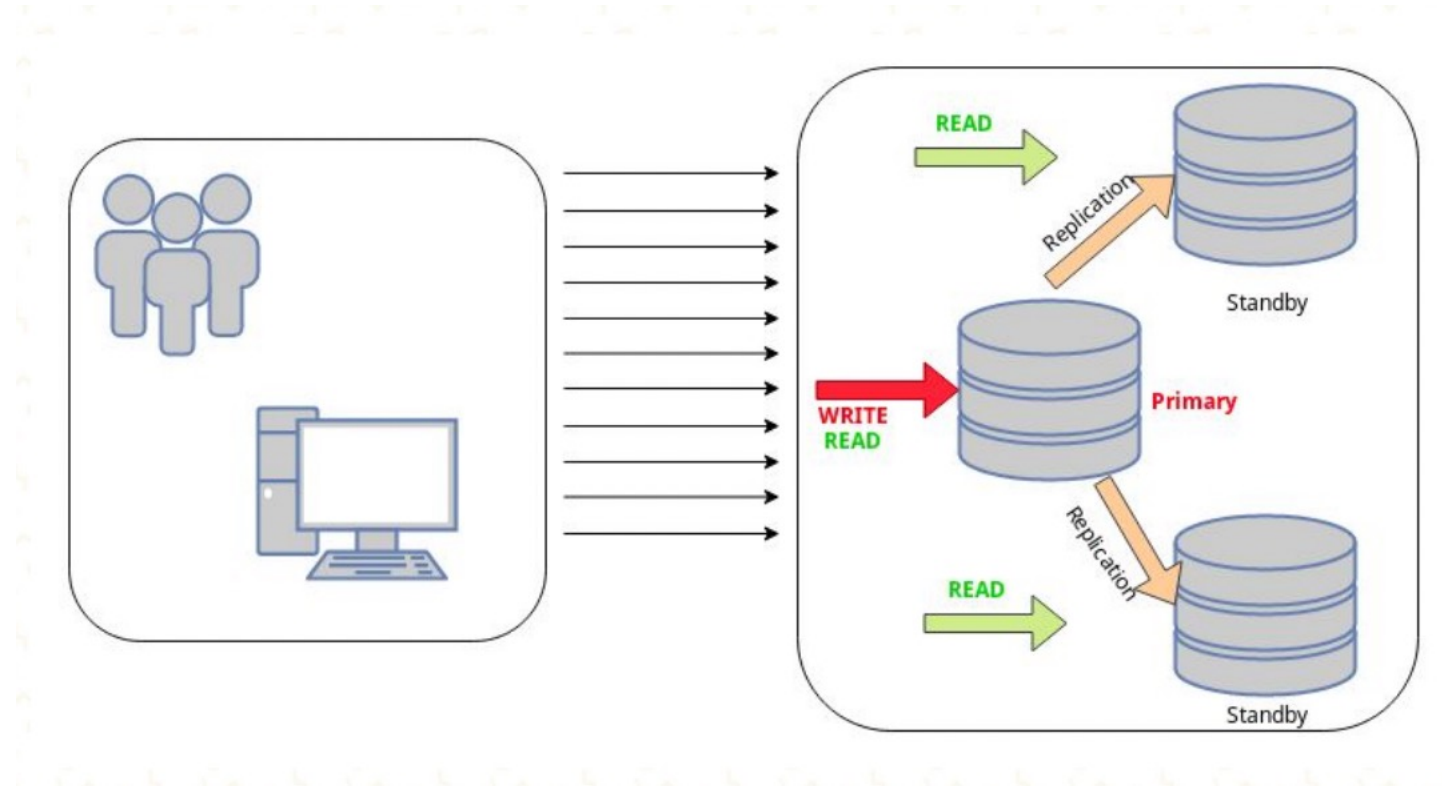
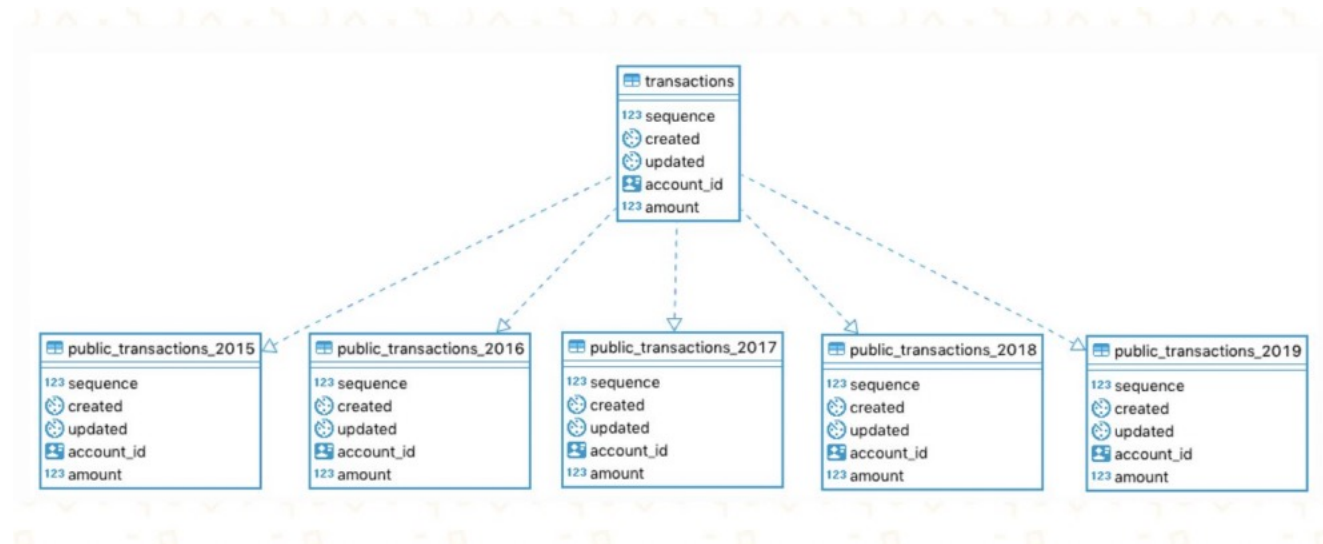




Table Partition



Table partition adalah proses membagi tabel data besar menjadi bagian-bagian kecil yang dapat dikelola, sehingga setiap bagian memiliki nama dan karakteristiknya sendiri





Example



```
1 CREATE TABLE TESTING.PUBLIC.DEMOGRAPHICS PARTITION BY (state) AS
2 SELECT
3     county, total_population, total_male_population,
4     total_female_population
5 FROM
6     covid19_epidemiological_data.public.demographics
7 WHERE
8     state='TX'
9 ORDER BY
10    total_male_population DESC;
```



Join Operation





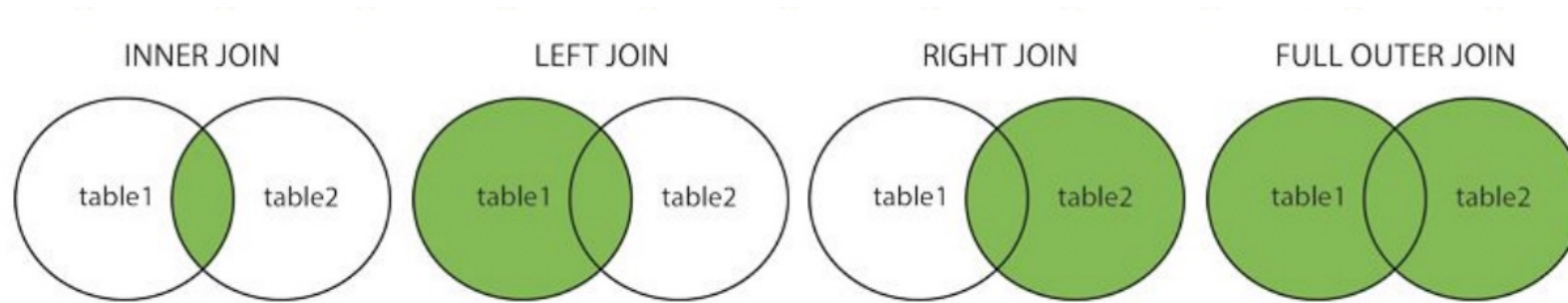
Join Operation



Digunakan untuk mengkombinasikan rows dari dua atau lebih table berdasarkan relasi table

- (INNER) Join : hanya mengembalikan baris-baris dari dua tabel atau lebih yang memenuhi syarat.
- Left Join : akan mengembalikan seluruh baris dari tabel disebelah kiri yang dikenai kondisi ON dan hanya baris dari tabel disebelah kanan yang memenuhi kondisi join.
- Right Join : Akan menampilkan semua data yang ada di table sebelah kanan dan mencari kecocokan key pada table sebelah kiri.
- Full Outer Join : akan mengembalikan seluruh baris dari kedua tabel yang dikenai ON termasuk data-data yang bernilai NULL.

Join Operation





Join Operation



<https://docs.snowflake.com/en/sql-reference/constructs/join>



Example Join Operation

```
SELECT
    OV.DATE, OV.ISO3166_1, OV.COUNTRY_REGION ,OV.DAILY_VACCINATIONS, OV.PEOPLE_VACCINATED, IBA.INPATIENT_BEDS_OCCUPIED,
    IBA.TOTAL_INPATIENT_BEDS, IBA.INPATIENT_BEDS_IN_USE_PCT
FROM
    COVID19_EPIDEMIOLOGICAL_DATA.PUBLIC.OWID_VACCINATIONS AS OV
LEFT JOIN
    COVID19_EPIDEMIOLOGICAL_DATA.PUBLIC.CDC_INPATIENT_BEDS_ALL AS IBA
ON OV.ISO3166_1 = IBA.ISO3166_1 AND OV.DATE = IBA.DATE
```



Sorting and Grouping



Sorting



- ORDER BY keyword digunakan untuk mengurutkan data pada query SQL, By Default akan mengurutkan secara ASC, untuk melakukan pengurutan secara descending tambahkan keyword DESC
- Proses ORDER merupakan salah satu Expensive Process pada SQL, hanya digunakan jika dibutuhkan



Grouping



- Digunakan bersama dengan pernyataan SELECT untuk mengatur data yang identik ke dalam grup. GROUP BY ini mengikuti klausa WHERE dalam pernyataan SELECT dan mendahului klausa ORDER BY.

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



Sub Query



Sub Query



A subquery is a SQL query nested inside a larger query.

- A subquery may occur in :
 - - A SELECT clause
 - - A FROM clause
 - - A WHERE clause
- The subquery can be nested inside a SELECT, INSERT, UPDATE, or DELETE statement or inside another subquery.
- A subquery is usually added within the WHERE Clause of another SQL SELECT statement.
- You can use the comparison operators, such as >, <, or =. The comparison operator can also be a multiple-row operator, such as IN, ANY, or ALL.
- A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.
- The inner query executes first before its parent query so that the results of an inner query can be passed to the outer query.

You can use a subquery in a SELECT, INSERT, DELETE, or UPDATE statement to perform the following tasks:

- Compare an expression to the result of the query.
- Determine if an expression is included in the results of the query.
- Check whether the query selects any rows.



Sub Query Example



StudentID	Name
V001	Abe
V002	Abhay
V003	Acelin
V004	Adelphos

studentid	name	total_marks
V001	Abe	95
V004	Adelphos	81

KKN > 80

StudentID	Total_marks
V001	95
V002	80
V003	74
V004	81

```
1 SELECT a.studentid, a.name, b.total_marks
2 FROM student a, marks b
3 WHERE a.studentid = b.studentid AND b.total_marks >
4 (SELECT total_marks
5 FROM marks
6 WHERE studentid = 'V002');
```



Function Operation



User Defined Function

User Defined Function (UDF) adalah fungsi yang kita buat sendiri sehingga kita dapat memanggilnya dari SQL.

Seperti fungsi bawaan yang bisa kita panggil dari SQL, logika UDF biasanya memperluas atau menyempurnakan SQL dengan fungsionalitas yang tidak dimiliki atau tidak dilakukan dengan baik oleh SQL. UDF juga memberi kita cara untuk mengenkapsulasi fungsionalitas sehingga kita dapat memanggilnya berulang kali dari berbagai tempat dalam kode.



UDF Example with SQL



```
CREATE FUNCTION multiply1 (a number, b number)
  RETURNS number
  COMMENT='multiply two numbers'
  AS 'a * b';
```



UDF Example with Python

```
create or replace function py_udf(a int)
  returns variant
  language python
  runtime_version = '3.8'
  handler = 'udf'
as $$
def udf(a):
    b = a * 3.14
    return b
$$;
```




Thankyou

