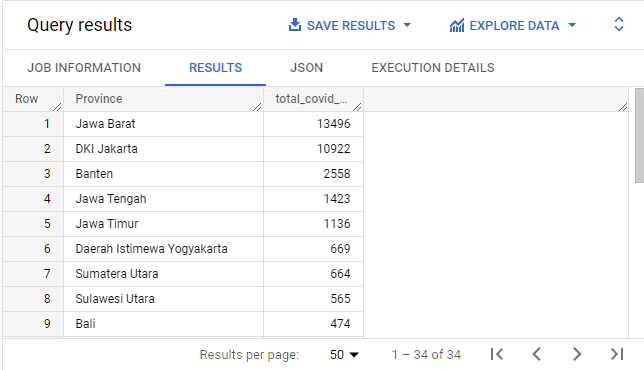
**STUDI KASUS SQL CHALLENGE 1 FGA-DS BINAR**

1. **Hitung jumlah total kasus covid aktif yang baru di setiap provinsi. Urutkan berdasarkan jumlah kasus yang paling besar**

**SQL:**

| -- 1. Hitung Jumlah total kasus covid aktif yang baru di setiap provinsi. -- Urutkan berdasarkan jumlah kasus yang paling besar  SELECT Province, SUM(New\_Active\_Cases) as total\_covid\_aktif\_baru\_provinsi FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` WHERE Province != 'null' GROUP BY Province ORDER BY total\_covid\_aktif\_baru\_provinsi DESC |
| --- |

**RESULT:**

****

**PENJELASAN:**

Pada pertanyaan no.1, **total kasus covid aktif** yang baru merupakan kolom **“New\_Active\_Cases”**, sedangkan **jumlah total kasus covid aktif yang baru** merupakan **SUM()** dari kolom tersebut (dapat dilihat pada SQL). Oleh karena itu, pada SQL akan diambil kolom **“Province”** dan **“total\_covid\_aktif\_baru\_provinsi”** yang merupakan jumlah total kasus covid aktif yang baru di **setiap provinsi.**

Selanjutnya, kolom tersebut diambil dari (FROM) tabel yang telah dibuat sebelumnya menggunakan dataset dari link: [https://www.kaggle.com/datasets/hendr](https://www.kaggle.com/datasets/hendratno/covid19-indonesia?resource=download)

[atno/covid19-indonesia?resource=download](https://www.kaggle.com/datasets/hendratno/covid19-indonesia?resource=download) pada projek challenge1-fga-binar-ds.

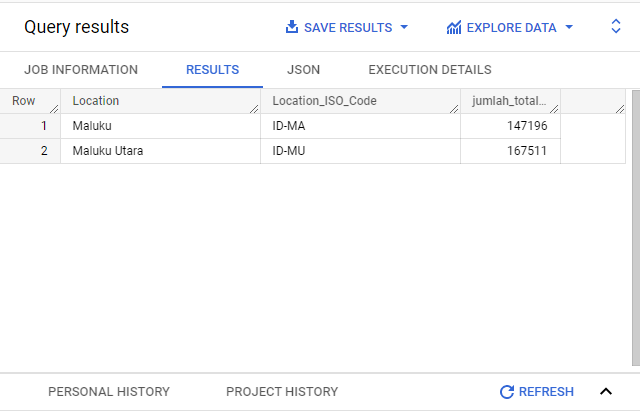
Oleh karena data yang akan ditampilkan **diurutkan** berdasarkan jumlah kasus (**dalam hal ini, jumlah kasus total covid aktif yang baru tiap provinsi**) **yang paling besar**, maka dikelompokkan **(GROUP BY)** berdasarkan **“Province”** dan tidak bernilai null pada kolom tersebut serta diurutkan **(ORDER BY)** berdasarkan **“total\_covid\_aktif\_baru\_provinsi”** secara **DESCENDING** (dari besar ke kecil).

1. **Ambil 2 location iso code yang memiliki jumlah total kematian karena covid paling sedikit.**

**SQL:**

| **-- 2. Ambil 2 location iso code yang memiliki jumlah total kematian karena covid paling sedikit  SELECT Location, Location\_ISO\_Code, SUM(Total\_Deaths) as jumlah\_total\_kematian FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` GROUP BY Location\_ISO\_Code, Location ORDER BY jumlah\_total\_kematian ASC LIMIT 2** |
| --- |

**RESULT:**



**PENJELASAN:**

Pada pertanyaan no.2 dapat diketahui bahwa **total kematian** adalah kolom **“Total\_Deaths”** dan **jumlah** total kematian merupakan **SUM()** dari kolom tersebut. Selain itu, dikarenakan **diambil 2 location iso code** diurutkan dari **paling sedikit** berarti **GROUP BY** kolom “Location\_Iso\_Code”, dilimit (**LIMIT**) 2 data saja dan **ORDER BY** “jumlah\_total\_kematian” **ASCENDING** (dari kecil ke besar).

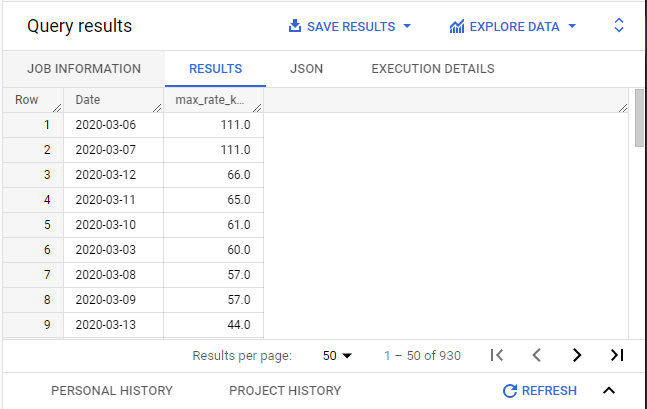
Pada SQL selain diambil kolom “Location\_Iso\_Code” juga kolom “Location” hanya untuk memastikan lokasinya benar sehingga didapat 2 location iso code yang memiliki jumlah total kematian karena covid paling sedikit, yakni **ID-MA** dan **ID-MU**

1. **Kapan (tanggal) Indonesia memiliki rate kasus recovered paling tinggi? Berapa Ratenya?**

**SQL:**

| -- 3. kapan (tanggal) Indonesia memiliki rate kasus recovered paling tinggi. -- berapa ratenya?  SELECT Date, MAX(Case\_Recovered\_Rate) as max\_rate\_kasus\_recovered FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` GROUP BY Date ORDER BY max\_rate\_kasus\_recovered DESC |
| --- |

**RESULT:**



**PENJELASAN:**

Pada pertanyaan no.3 dapat diketahui bahwa **rate kasus recovered** merupakan kolom **“Case\_Recovered\_Rate”**, sedangkan paling tinggi maka menggunakan agregasi MAX().

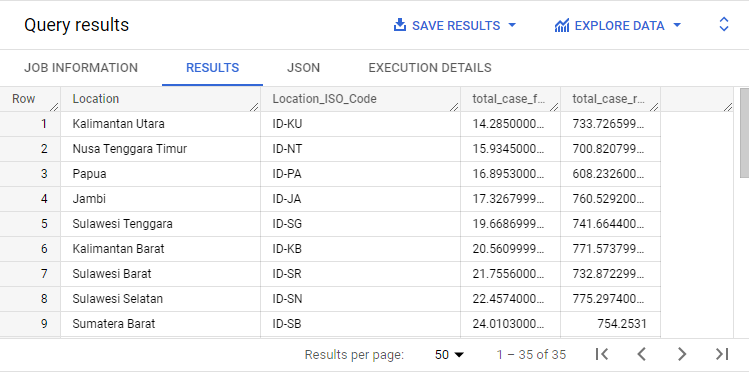
Pada SQL diambil kolom **“Date”** dan **“max\_rate\_kasus\_recovered”**. Dikarenakan terdapat agregasi **MAX()** maka butuh pengelompokkan (**GROUP BY**) yang mana pada hal ini dikelompokkan **berdasarkan tanggal** atau Date. Setelah itu, dikarenakan dilihat paling tinggi (rate kasus recovered) maka **ORDER BY** “max\_rate\_kasus\_recovered” **DESC** (dari besar ke kecil). Sehingga diketahui bahwa tanggal **2020-03-06** dan **2020-03-07** Indonesia memiliki rate kasus recovered paling tinggi.

1. **LANJUTAN PERTANYAAN**
2. **Hitung total case fatality rate dan case recovered rate dari masing-masing location iso code. Urutkan dari yang paling rendah.**

**SQL:**

| **-- A. LANJUTAN PERTANYAAN -- 1. Hitung total case fatality rate dan case recovered rate dari masing-masing location iso code. -- urutkan dari yang paling rendah  SELECT Location, Location\_ISO\_Code ,   SUM(Case\_Fatality\_Rate) as total\_case\_fatal,   SUM(Case\_Recovered\_Rate) as total\_case\_recovered FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` GROUP BY Location\_ISO\_Code, Location ORDER BY total\_case\_fatal ASC** |
| --- |

**RESULT:**

****

**PENJELASAN:**

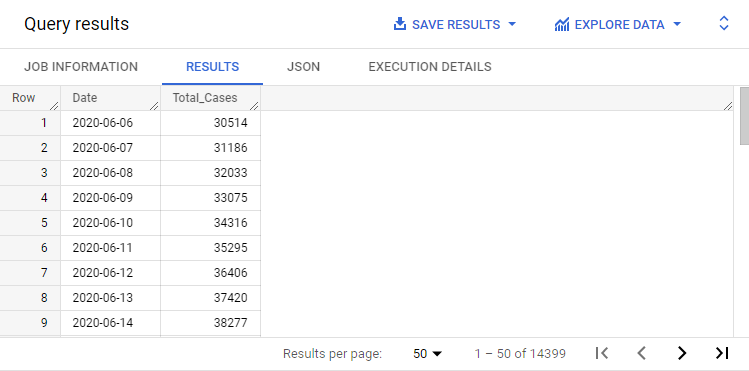
Pada pertanyaan no.1 lanjutan dapat diketahui bahwa total case fatality rate dan total case recovered rate merupakan “**SUM(Case\_Fatality\_Rate) as total\_case\_fatal, SUM(Case\_Recovered\_Rate) as total\_case\_recovered**”. Selain itu, dikarenakan “dari masing-masing location iso code” maka data yang diambil dikelompokkan (GROUP BY) kolom “Location\_Iso\_Code”. Namun pada pertanyaan **“diurutkan dari yang paling rendah” tidak disebutkan berdasarkan kolom apa** **sehingga diambil dari nilai paling rendah antara total\_case\_fatal dan total\_case\_recovered**. Selanjutnya, dari dua kolom tersebut diketahui nilai paling rendah terdapat pada kolom total\_case\_fatal sehingga **ORDER BY total\_case\_fatal ASC**

1. **Pada tanggal berapa total kasus covid mulai menyentuh angka 30.000-an? Hitung ada berapa data yang tercatat ketika kasus covid lebih dari atau sama dengan 30.000?**

**SQL 1:**

| **-- 2. Pada tanggal berapa total kasus covid mulai menyentuh angka 30.000-an? -- Hitung ada berapa data yang tercatat ketika kasus covid lebih dari atau sama dengan 30.000?  SELECT Date, Total\_Cases FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` WHERE Total\_Cases >= 30000 ORDER BY Date ASC** |
| --- |

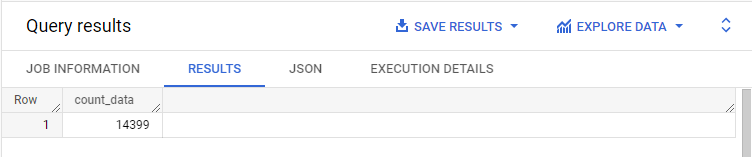
**RESULT 1:**

****

**SQL 2:**

| **SELECT COUNT(Date) as count\_data FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn` WHERE Total\_Cases IN(  SELECT Total\_Cases  FROM `challenge1-fga-binar-ds.covid19\_indonesia.covid19idn`  WHERE Total\_Cases >= 30000 )** |
| --- |

**RESULT 2:**

****

**PENJELASAN:**

Pada pertanyaan no.2 lanjutan terdapat **dua buah pertanyaan**, yakni: pada tanggal kasus covid menyentuh angka 30.000-an; dan berapa banyak data saat covid >= 30.000 sehingga terdapat **2 SQL dan 2 RESULT (bisa hanya 1 SQL dan melihat banyak data dari jajaran result per page bagian bawah)**.

Pada SQL 1 diambil **Date** dan **Total\_Cases** lalu dikondisikan (**WHERE**) total **kasus >= 30.000** untuk mendapatkan tanggal saat **total kasus mencapai 30,000 an**. Oleh karena itu, didapat mulai tanggal **2020-06-06** dengan total kasus **30514** dengan banyak data yang dapat dilihat dibawah RESULT sebanyak **14399 data**.

Pada **SQL 2 memastikan banyaknya data pada SQL 1 menggunakan metode subquery dan didapat jumlah data yang sama pada SQL 1**