




Query **DATE**, **SUBQUERY**, **AND** **CASE WHEN**

23 October 2023

Sesi 4 – Bootcamp Data Analyst with SQL and
Python using Google Platform



Fundamental SQL Menggunakan Google Big Query

1. DDL : Create, Drop 
2. Query Select and combination 
3. DML : Insert, Update, Delete 
4. Query Date, Subquery and Case When
5. Query Join

1

QUERY DATE

23 October 2023

**Sesi 4 – Bootcamp Data Analyst with SQL and
Python using Google Platform**



QUERY DATE

- CURRENT_DATE
- DATE
- DATE_ADD
- DATE_DIFF
- DATE_SUB
- EXTRACT
- FORMAT_DATE
- PARSE_DATE
- DATETIME

Date Function – Current Date()

Mengembalikan tanggal saat ini sebagai objek DATE. Tanda kurung bersifat opsional saat dipanggil tanpa argumen.

```
SELECT CURRENT_DATE() AS the_date;
```

```
/*-----*  
| the_date |  
+-----+  
| 2016-12-25 |  
*-----*/
```



```
4 --- current_date  
5 select current_date() as date_now ;  
6
```

Processing location: US

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

Row	date_now
1	2023-10-23

Date Function – DATE() structure

Membuat atau mengekstrak tanggal.

Fungsi ini mendukung argumen berikut:

- year: Nilai INT64 untuk tahun.
- month: Nilai INT64 untuk bulan.
- day: Nilai INT64 untuk hari.
- timestamp_expression: Ekspresi TIMESTAMP yang berisi tanggal.
- time_zone_expression: Ekspresi STRING yang mewakili zona waktu. Jika tidak ada zona waktu yang ditentukan dengan timestamp_expression, zona waktu default, UTC, akan digunakan.
- datetime_expression: Ekspresi DATETIME yang berisi tanggal.

```
SELECT
  DATE(2016, 12, 25) AS date_ymd,
  DATE(DATETIME '2016-12-25 23:59:59') AS date_dt,
  DATE(TIMESTAMP '2016-12-25 05:30:00+07', 'America/Los_Angeles') AS date_tstz;
```

```
/*-----+-----+-----*
| date_ymd | date_dt | date_tstz |
+-----+-----+-----+
| 2016-12-25 | 2016-12-25 | 2016-12-24 |
*-----+-----+-----*/
```

Date Function – DATE_ADD

DATE_ADD(date_expression, INTERVAL int64_expression date_part)

Menambahkan interval waktu tertentu ke DATE. DATE_ADD mendukung nilai date_part berikut:

- DAY
- WEEK. Equivalent to 7 DAYS.
- MONTH
- QUARTER
- YEAR

```
7 --- DATE_ADD(date_expression, INTERVAL int64_expression date_part)
8 SELECT
9   DATE_ADD(DATE '2023-10-22', INTERVAL 5 DAY) AS five_days_later,
10  DATE_ADD(DATE '2023-10-22', INTERVAL 10 DAY) AS ten_days_later,
11  DATE_ADD(DATE '2023-10-22', INTERVAL -10 DAY) AS ten_days_back
12 ;
13
```

Processing location: US

Query results

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXE
Row	five_days_later	ten_days_later	ten_days_back			
1	2023-10-27	2023-11-01	2023-10-12			

```
14 SELECT
15     DATE_ADD(DATE '2023-10-22', INTERVAL -1 MONTH) AS a_month_back,
16     DATE_ADD(DATE '2023-10-22', INTERVAL -2 WEEK) AS two_weeks_back,
17     DATE_ADD(DATE '2023-10-22', INTERVAL 1 YEAR) AS a_year_later
18 ;
19
```

Processing location: US

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXE

Row	a_month_back	two_weeks_back	a_year_later	
1	2023-09-22	2023-10-08	2024-10-22	

Date Function – DATE_DIFF

DATE_DIFF(date_expression_a, date_expression_b, date_part)

DATE_DIFF memberikan hasil dari seluruh jumlah interval date_part yang ditentukan antara dua objek DATE (date_expression_a - date_expression_b). Jika DATE pertama lebih awal dari yang kedua, hasilnya negatif.

DATE_DIFF mendukung nilai date_part berikut:

- DAY
- WEEK Bagian tanggal ini dimulai pada hari Minggu.
- WEEK(<WEEKDAY>): Bagian tanggal ini dimulai pada WEEKDAY. Nilai yang valid untuk WEEKDAY adalah SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, dan SATURDAY
- MONTH, kecuali jika dua argumen pertama adalah objek TIMESTAMP.
- QUARTER
- YEAR

```
23
24 SELECT
25   DATE_DIFF(DATE '2023-10-25', DATE '2023-10-23', DAY) AS days_diff,
26   DATE_DIFF(DATE '2023-10-25', DATE '2023-10-23', WEEK) AS weeks_diff,
27   DATE_DIFF(DATE '2023-10-25', DATE '2023-09-20', MONTH) AS month_diff,
28   DATE_DIFF(DATE '2023-10-25', DATE '2020-10-20', YEAR) AS year_diff,
29   ;
30
```

Processing location: US


Query results

JOB INFORMATION		RESULTS		CHART	PREVIEW	JSON	EXECUTION
Row	days_diff	weeks_diff	month_diff	year_diff			
1	2	0	1	3			

`DATE_DIFF` with the `date` part `WEEK` returns `1` because `DATE_DIFF` counts the number of date part boundaries in this range of dates.

Each `WEEK` begins on Sunday, so there is one date part boundary between Wednesday, 2023-10-25 and Saturday, 2023-10-20.

```
34 SELECT
35   DATE_DIFF('2023-10-25', '2023-10-23', WEEK) AS week_diff,
36   DATE_DIFF('2023-10-25', '2023-10-23', WEEK(WEDNESDAY)) AS week_weekday_diff
37 ;
38
```

Processing location: US 

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUTION DETAILS

Row	week_diff	week_weekday_diff
1	0	1

Date Function – DATE_SUB

DATE_SUB(date_expression, INTERVAL int64_expression date_part)

Mengurangi interval waktu tertentu dari DATE.

DATE_SUB mendukung nilai date_part berikut:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

```
SELECT DATE_SUB(DATE '2008-12-25', INTERVAL 5 DAY) AS five_days_ago;

/*-----*
| five_days_ago |
+-----+
| 2008-12-20    |
*-----*/
```

```
45 --- DATE_SUB(date_expression, INTERVAL int64_expression date_part)
46 SELECT
47     DATE_ADD(DATE '2023-10-22', INTERVAL -5 DAY) AS five_days_ago,
48     DATE_SUB(DATE '2023-10-22', INTERVAL 5 DAY) AS five_days_ago
49 ;
50
```

Processing location: US 

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXEC

Row	five_days_ago	five_days_ago_1	
1	2023-10-17	2023-10-17	

Date Function – EXTRACT

EXTRACT(part FROM date_expression)

Mengembalikan nilai yang sesuai dengan bagian tanggal yang ditentukan. Bagian harus salah satu dari:

- DAYOFWEEK: Mengembalikan nilai dalam rentang [1,7] dengan hari Minggu sebagai hari pertama dalam seminggu.
- DAY
- DAYOFYEAR
- WEEK: Mengembalikan nomor minggu dari tanggal dalam rentang [0, 53]. Minggu dimulai dengan hari Minggu, dan tanggal sebelum hari Minggu pertama tahun ini berada di minggu 0.
- WEEK(<WEEKDAY>): Mengembalikan nomor minggu dari tanggal dalam rentang [0, 53]. Minggu dimulai pada WORK DAY. Tanggal sebelum WORKDAY pertama tahun ini berada di minggu 0. Nilai yang valid untuk WORKDAY adalah MINGGU, SENIN, SELASA, RABU, KAMIS, JUMAT, dan SABTU.
- MONTH
- QUARTER: Mengembalikan nilai dalam rentang [1,4].
- YEAR

```
57 SELECT
58   EXTRACT(DAY FROM DATE '2023-10-23') AS the_day,
59   EXTRACT(DAY FROM current_date()) AS the_day
60 ;
61
62
```

Processing location: US ✕

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

Row	the_day	the_day_1
1	23	23

```
62 SELECT
63   EXTRACT(DAY FROM DATE '2023-10-23') AS the_day,
64   EXTRACT(WEEK FROM DATE '2023-10-23') AS the_week,
65   EXTRACT(MONTH FROM DATE '2023-10-23') AS the_month,
66   EXTRACT(YEAR FROM DATE '2023-10-23') AS the_year
67 ;
```

Processing location: US ✕

Query results

JOB INFORMATION

RESULTS

CHART

PREVIEW

JSON

EXECUT

Row	the_day	the_week	the_month	the_year
1	23	43	10	2023

FORMAT_DATE

Memformat date_expr sesuai dengan format_string yang ditentukan.

Return Data Type (STRING)

```
SELECT FORMAT_DATE('%x', DATE '2008-12-25') AS US_format;
```

```
/*-----*  
| US_format |  
+-----+  
| 12/25/08  |  
*-----*/
```

```
SELECT FORMAT_DATE('%b-%d-%Y', DATE '2008-12-25') AS formatted;
```

```
/*-----*  
| formatted  |  
+-----+  
| Dec-25-2008 |  
*-----*/
```



```
74 SELECT
75   FORMAT_DATE('%b-%d-%Y', current_date()) AS formatted,
76   FORMAT_DATE('%d-%b-%Y', current_date()) AS formatted,
77   FORMAT_DATE('%Y-%b-%d', current_date()) AS formatted,
78   FORMAT_DATE('%d-%m-%Y', current_date()) AS formatted,
79 ;
80
```

Processing location: US

Query results

[SAVE RESULTS](#)

JOB INFORMATION		RESULTS	CHART	PREVIEW	JSON	EXECUTION DETAILS	EXECUTION G
Row	formatted	formatted_1	formatted_2	formatted_3			
1	Oct-23-2023	23-Oct-2023	2023-Oct-23	23-10-2023			

PARSE_DATE

Mengkonversi representasi string tanggal menjadi objek DATE.

format_string berisi elemen format yang menentukan bagaimana date_string diformat. Setiap elemen di date_string harus memiliki elemen yang sesuai di format_string. Lokasi setiap elemen dalam format_string harus sesuai dengan lokasi setiap elemen dalam date_string.

```
-- This works because elements on both sides match.  
SELECT PARSE_DATE('%A %b %e %Y', 'Thursday Dec 25 2008')  
  
-- This produces an error because the year element is in different locations.  
SELECT PARSE_DATE('%Y %A %b %e', 'Thursday Dec 25 2008')  
  
-- This produces an error because one of the year elements is missing.  
SELECT PARSE_DATE('%A %b %e', 'Thursday Dec 25 2008')  
  
-- This works because %F can find all matching elements in date_string.  
SELECT PARSE_DATE('%F', '2000-12-30')
```

DATETIME

Membuat objek DATETIME menggunakan nilai INT64 yang mewakili tahun, bulan, hari, jam, menit, dan detik.

Membuat objek DATETIME menggunakan objek DATE dan objek TIME opsional.

Membuat objek DATETIME menggunakan objek TIMESTAMP. Ini mendukung parameter opsional untuk menentukan zona waktu. Jika tidak ada zona waktu yang ditentukan, zona waktu default, UTC, akan digunakan.

Return Data Type (DATETIME)

```
SELECT
  DATETIME(2008, 12, 25, 05, 30, 00) as datetime_ymdhms,
  DATETIME(TIMESTAMP "2008-12-25 05:30:00+00", "America/Los_Angeles") as datetime_tstz;

/*-----+-----*
| datetime_ymdhms | datetime_tstz |
+-----+-----+
| 2008-12-25T05:30:00 | 2008-12-24T21:30:00 |
*-----+-----*/
```

DATETIME_ADD

Menambahkan unit int64_expression part ke objek DATETIME.

DATETIME_ADD mendukung nilai berikut untuk sebagian:

- MICROSECOND
- MILLISECOND
- SECOND
- MINUTE
- HOUR
- DAY
- WEEK. Equivalent to 7 DAYS.
- MONTH
- QUARTER
- YEAR

```
SELECT
    DATETIME "2008-12-25 15:30:00" as original_date,
    DATETIME_ADD(DATETIME "2008-12-25 15:30:00", INTERVAL 10 MINUTE) as later;

/*-----+-----*/
| original_date          | later                |
+-----+-----+
| 2008-12-25T15:30:00    | 2008-12-25T15:40:00 |
*-----+-----*/
```

Return Data Type (DATETIME)

DATETIME_DIFF

Mengembalikan seluruh jumlah interval bagian tertentu antara dua objek DATETIME (datetime_expression_a - datetime_expression_b). Jika DATETIME pertama lebih awal dari yang kedua, hasilnya negatif. Melempar kesalahan jika komputasi melebihi tipe hasil, seperti jika perbedaan dalam mikrodetik antara dua objek DATETIME akan melebihi nilai INT64.

DATETIME_DIFF mendukung nilai berikut untuk sebagian:

- MICROSECOND
- MILLISECOND
- SECOND
- MINUTE
- HOUR
- DAY
- WEEK: This date part begins on Sunday.
- MONTH
- QUARTER
- YEAR

SELECT

```
DATETIME "2010-07-07 10:20:00" as first_datetime,
DATETIME "2008-12-25 15:30:00" as second_datetime,
DATETIME_DIFF(DATETIME "2010-07-07 10:20:00",
  DATETIME "2008-12-25 15:30:00", DAY) as difference;
```

```
/*-----+-----+-----*
| first_datetime | second_datetime | difference |
+-----+-----+-----+
| 2010-07-07T10:20:00 | 2008-12-25T15:30:00 | 559 |
*-----+-----+-----*/
```

SELECT

```
DATETIME_DIFF(DATETIME '2017-10-15 00:00:00',
  DATETIME '2017-10-14 00:00:00', DAY) as days_diff,
DATETIME_DIFF(DATETIME '2017-10-15 00:00:00',
  DATETIME '2017-10-14 00:00:00', WEEK) as weeks_diff;
```

```
/*-----+-----*
| days_diff | weeks_diff |
+-----+-----+
| 1 | 1 |
*-----+-----*/
```

2

SUB- QUERY

23 October 2023

Sesi 4 - Bootcamp Data Analyst with SQL and Python using Google Platform



SubQuery

SubQuery adalah melakukan kombinasi lebih dari satu query untuk mendapatkan output tertentu.

Misal, terdapat tabel karyawan dan departemen berikut dari database sampel:

Berikut query untuk menemukan semua karyawan yang termasuk dalam lokasi id 1700 (sub query) dengan menggunakan daftar id departemen dari query sebelumnya (main Query):

```
SELECT
  *
FROM
  departments
WHERE
  location_id = 1700;
```

Main Query

```
SELECT
  employee_id, first_name, last_name
FROM
  employees
WHERE
  department_id IN (1 , 3, 8, 10, 11)
ORDER BY first_name , last_name;
```

Sub Query



```
SELECT
  employee_id, first_name, last_name
FROM
  employees
WHERE
  department_id IN (SELECT
    department_id
  FROM
    departments
  WHERE
    location_id = 1700)
ORDER BY first_name , last_name;
```

Combined

3

SELECT *CASE WHEN*

Sesi 4 – Bootcamp Data Analyst with SQL and Python using Google Platform

23 October 2023



CASE FUNCTION

Untuk membuat conditional statement atau rules dari sebuah kolom

CASE

WHEN *condition1* THEN *result1*

WHEN *condition2* THEN *result2*

WHEN *conditionN* THEN *resultN*

ELSE *result*

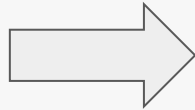
END;

Contoh data : Order Details

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

```
SELECT OrderID, Quantity,  
CASE  
  WHEN Quantity > 30 THEN 'The quantity is greater than 30'  
  WHEN Quantity = 30 THEN 'The quantity is 30'  
  ELSE 'The quantity is under 30'  
END AS QuantityText  
FROM OrderDetails;
```

OrderID	Quantity
10248	12
10248	10
10248	5
10249	9
10249	40
10250	10
10250	35
10250	15



OrderID	Quantity	QuantityText
10248	12	The quantity is under 30
10248	10	The quantity is under 30
10248	5	The quantity is under 30
10249	9	The quantity is under 30
10249	40	The quantity is greater than 30
10250	10	The quantity is under 30
10250	35	The quantity is greater than 30
10250	15	The quantity is under 30

4

Additional

FYI

23 October 2023

Sesi 4 – Bootcamp Data Analyst with SQL and Python using Google Platform



Data Type

Bilangan Bulat (INTEGER)

Tipe bilangan bulat (*Integer*) adalah tipe data numerik yang biasa digunakan apabila bertemu dengan bilangan bulat, seperti 1, 27, 100, dll

Bilangan Pecahan (FLOAT)

Tipe bilangan pecahan atau *floating point* adalah bilangan yang menangani bilangan desimal atau perhitungan secara detail.

Karakter (STRING)

Tipe data karakter tunggal yang biasa didefinisikan dengan tanda petik (') di awal dan di akhir karakternya.

Boolean

Tipe data *boolean* merupakan tipe yang memiliki dua nilai yaitu benar (*true*) atau salah (*false*). Nilai yang digunakan pada tipe ini sangat penting dalam mengambil keputusan suatu kejadian tertentu.

Format Angka (CAST- Float)

GoogleSQL mendukung transmisi ke tipe floating point. Parameter ekspresi dapat mewakili ekspresi untuk tipe data ini:

- INT64
- FLOAT64
- NUMERIC
- BIGNUMERIC
- STRING

```
CAST(expression AS FLOAT64)
```

Format Angka (CAST- INT64)

GoogleSQL mendukung transmisi ke tipe integer. Parameter ekspresi dapat mewakili ekspresi untuk tipe data ini:

- INT64
- FLOAT64
- NUMERIC
- BIGNUMERIC
- BOOL
- STRING

```
CAST(expression AS INT64)
```

Format Angka (CAST- NUMERIC)

GoogleSQL mendukung transmisi ke NUMERIC. Parameter ekspresi dapat mewakili ekspresi untuk tipe data ini:

- INT64
- FLOAT64
- NUMERIC
- BIGNUMERIC
- STRING

```
CAST(expression AS NUMERIC)
```

Thanks!

