

Project SQL

XERATIC ANALYST

Final Test DQLab

Oleh: Saffa Rahmatullah



Link Dataset

https://bit.ly/dataset_xeratic_finaltest

Tools yang digunakan



Project ini menggunakan Query SQL
Basic to Advance

Link Github

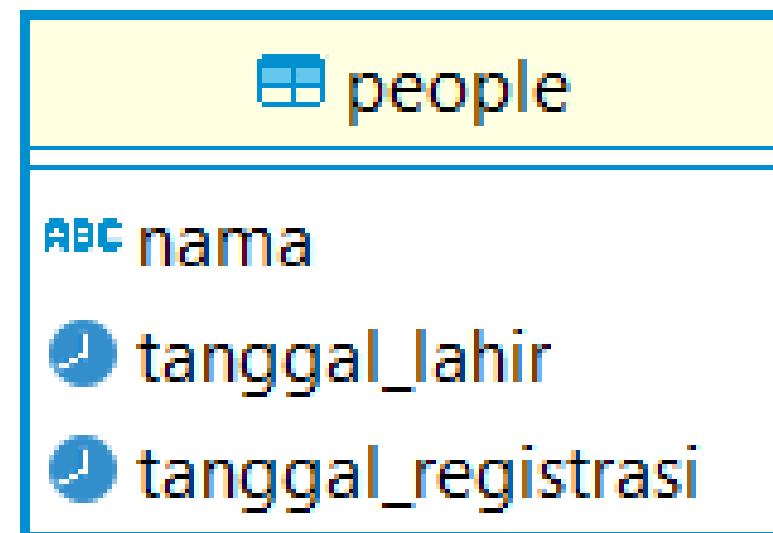
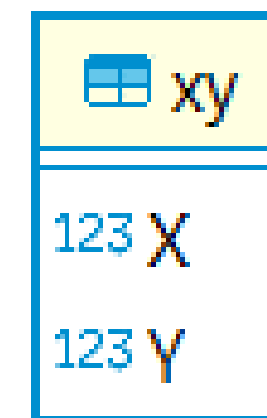
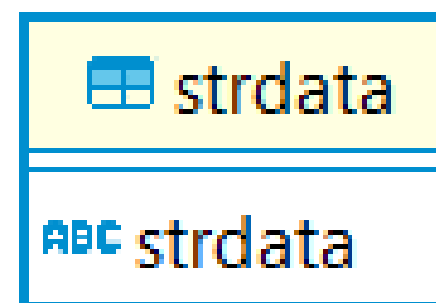
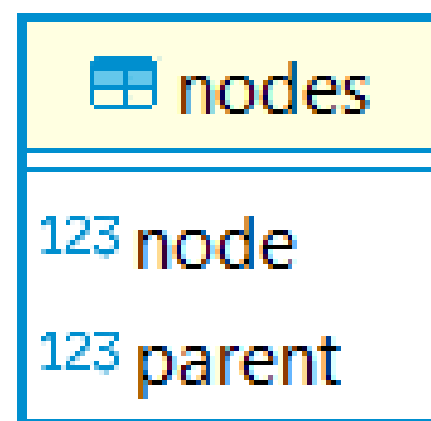
Untuk melihat secara lengkap analisis project ini, dapat dilihat pada link github berikut:

https://bit.ly/github_xeraticfinaltest_analyst

SELECT, ALIAS, LIMIT, Aggregation,
Group By, Having By, Order By, WHERE,
WHERE NOT IN, JOIN, CASE WHEN,
REGEX, SubQuery



ER DIAGRAM



★ Node and Position

node	parent
1	2
3	2
6	8
9	8
2	5
8	5
5	(NULL)

Perhatikan isi table **nodes** disamping:

- Jika node tidak memiliki parent, maka posisinya adalah Akar
- Jika node memiliki parent dan tidak memiliki anak, maka posisinya adalah Daun
- Jika node memiliki parent dan memiliki anak, maka posisinya adalah Batang

Buatlah query untuk menghasilkan output yang berisi column **node** dan **position** !!

Ketentuan : Urutkan baris berdasarkan kolom **node**

OUTPUT

QUERY

```
1 SELECT
2     n.node,
3     CASE
4         WHEN n.parent IS NULL THEN 'Akar'
5         WHEN (SELECT COUNT(*) FROM nodes WHERE parent = n.node) = 0 THEN 'Daun'
6         ELSE 'Batang'
7     END AS position
8 FROM
9     nodes n
10 ORDER BY
11     n.node;
```

	node	position
▶	1	Daun
	2	Batang
	3	Daun
	5	Akar
	6	Daun
	8	Batang
	9	Daun

- Column **position** diperoleh dari fungsi CASE WHEN yang digunakan untuk menentukan posisi suatu node dalam sebuah struktur pohon berdasarkan relasinya dengan node lain dalam struktur tersebut



Pasangan Simetris

X	Y
30	40
40	30
20	21
23	22
22	23
21	20
25	24
88	77
44	55
55	44

Perhatikan isi table **xy** disamping:

Buatlah query untuk mencari pasangan simetris. Pasangan (X1, Y1) dan (X2, Y2) disebut sebagai pasangan simetris jika $X1 = Y2$ dan $X2 = Y1$, sehingga menghasilkan output dengan column **X** dan **Y** !!

Ketentuan : Urutkan baris berdasarkan kolom **X** dan **Y** dari kecil ke besar

QUERY

```
1 • SELECT
2     LEAST(X, Y) AS X,
3     GREATEST(X, Y) AS Y
4 FROM
5     xy
6 WHERE
7     (X, Y) NOT IN (
8         SELECT
9             DISTINCT GREATEST(Y, X) AS X,
10            LEAST(Y, X) AS Y
11        FROM
12            xy
13    )
14 ORDER BY
15     X, Y;
```

Result Grid		
	X	Y
▶	20	21
	22	23
	30	40
	44	55

OUTPUT

- Query tersebut memilih nilai terkecil (LEAST) dari X dan Y, dan nilai terbesar (GREATEST) dari X dan Y dari tabel xy. Kemudian, hasilnya difilter sehingga hanya nilai yang tidak ada di subquery yang mengubah posisi X dan Y akan dipilih.



String Extraction

Perhatikan isi table **strdata** disamping:

strdata
2022-01-01 telah terjual 1 lusin seharga Rp 12000
Tanggal 2022-01-02 terjual 1 buah seharga Rp 1000
2 hari kemudian yaitu tanggal 2022-01-04 kembali terjual 6 buah seharga Rp 6000
Pada hari berikutnya, yaitu pada tanggal 2022-01-05 terjual 10 buah seharga Rp 10000

Buatlah query untuk menghasilkan output dengan column **tanggal**, **qty**, **harga_satuan**, dan **total** !!

Ketentuan : Tipe data kolom tanggal adalah **DATE**, tipe data kolom **qty**, **harga_satuan**, dan **total** adalah **INTEGER**, Urutkan baris berdasarkan kolom **tanggal**

QUERY

```
1 • SELECT
2     DATE_FORMAT(data.tanggal, '%Y-%m-%d') AS tanggal,
3     data.qty,
4     1000 AS harga_satuan,
5     data.qty * 1000 AS total
6 FROM (
7     SELECT
8         CASE
9             WHEN strdata LIKE 'Tanggal%' THEN
10                 CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(strdata, 'Tanggal ', -1), ' ', 1) AS DATE)
11             ELSE CAST(REGEXP_SUBSTR(strdata, '[0-9]{4}-[0-9]{2}-[0-9]{2}') AS DATE)
12         END AS tanggal,
13         CASE
14             WHEN strdata LIKE '%1 lusin%' THEN 12
15             WHEN strdata LIKE '%1 buah%' THEN 1
16             ELSE CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(strdata, ' terjual ', -1), ' ', 1) AS UNSIGNED)
17         END AS qty
18     FROM
19         strdata
20 ) AS data;
```

OUTPUT

Result Grid Filter Rows: <input type="text"/>				
	tanggal	qty	harga_satuan	total
▶	2022-01-01	12	1000	12000
	2022-01-02	1	1000	1000
	2022-01-04	6	1000	6000
	2022-01-05	10	1000	10000

- Query tersebut mengambil data dari tabel `strdata`, memprosesnya untuk mendapatkan informasi tentang tanggal penjualan, jumlah barang yang terjual, harga satuan, dan total penjualan untuk setiap transaksi. Pemrosesan dilakukan dengan menggunakan klausa `CASE` dan fungsi manipulasi string dalam subquery, kemudian data dipilih dan diubah formatnya dalam SELECT statement utama



Selisih Usia

Table **people** adalah daftar orang dengan tanggal lahir dan tanggal registrasi pada sebuah sekolah online.

Buatlah query untuk melakukan perhitungan umur setiap orang pada saat registrasi dan cari dua nama yang selisih umur pada saat registrasi paling kecil !! Outputnya berupa table dengan kolom

nama1, nama2, dan selisih

Ketentuan : Urutan **nama1** dan **nama2** berdasarkan umur yang lebih tua

QUERY

```
1 • SELECT
2   CASE WHEN ABS(DATEDIFF(p1.tanggal_registrasi, p1.tanggal_lahir)) > ABS(DATEDIFF(p2.tanggal_registrasi, p2.tanggal_lahir))
3     THEN p1.nama
4     ELSE p2.nama
5   END AS nama1,
6   CASE WHEN ABS(DATEDIFF(p1.tanggal_registrasi, p1.tanggal_lahir)) > ABS(DATEDIFF(p2.tanggal_registrasi, p2.tanggal_lahir))
7     THEN p2.nama
8     ELSE p1.nama
9   END AS nama2,
10  ABS(DATEDIFF(p1.tanggal_registrasi, p1.tanggal_lahir) - DATEDIFF(p2.tanggal_registrasi, p2.tanggal_lahir)) AS selisih
11 FROM
12   people p1
13 JOIN
14   people p2 ON p1.nama > p2.nama
15 ORDER BY
16   selisih
17 LIMIT 1;
```

OUTPUT

Result Grid				Filter Rows
	nama1	nama2	selisih	
▶	Chandra	Fadhil	182	

- Query tersebut membandingkan dua entitas dalam tabel 'people', p1 dan p2, berdasarkan perbedaan absolut antara tanggal registrasi dan tanggal lahir. Pertama, query mengevaluasi perbedaan tersebut menggunakan klausa CASE, kemudian mengurutkannya berdasarkan selisih tersebut

Terima Kasih

Let's Connect



saffarahmatullah@gmail.com



<https://linkedin.com/in/saffarahmatullah>



<https://github.com/RSaff>