

Project SQL

XERATIC ANALYST

Final Test DQLab

Oleh: Saffa Rahmatullah



The Dataset Link

https://bit.ly/dataset_xeratic_finaltest

The tools that are used



This project employs SQL queries from basic to advanced levels

The Github Link

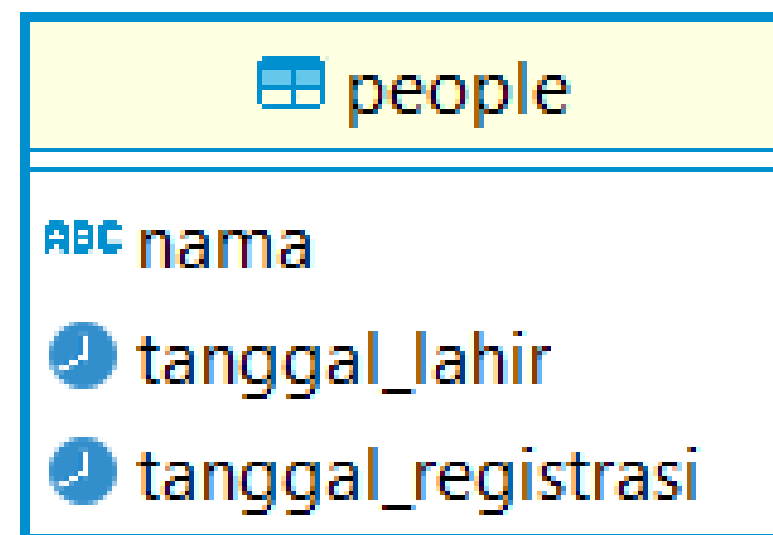
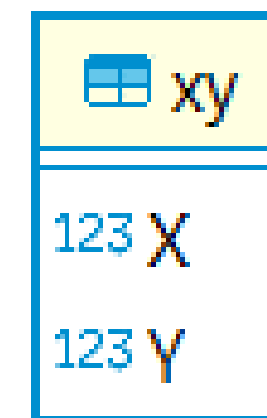
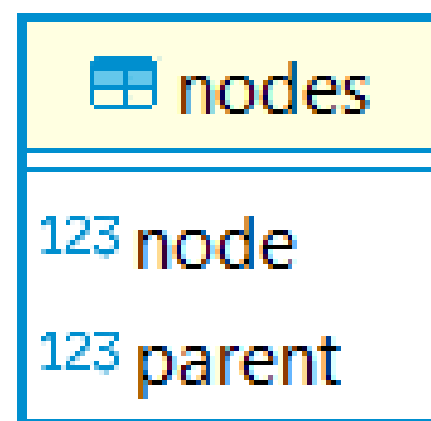
For a comprehensive view of this project's analysis, it can be accessed at the following GitHub link:

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)

Consider the content of the table "**nodes**" below:

- If a node has no parent, then its position is Akar.
- If a node has a parent but no children, then its position is Daun.
- If a node has a parent and children, then its position is Batang.

Create a query to generate an output containing the '**node**' and '**position**' columns!!

Criteria : Sort the rows based on the 'node' column.

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**' is obtained from the CASE WHEN function used to determine the position of a node within a tree structure based on its relationship with other nodes in the structure.

★ Symmetric Pairs

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

Consider the content of the table "**xy**" below:

Create a query to find symmetric pairs. Pairs (X1, Y1) and (X2, Y2) are called symmetric pairs if $X1 = Y2$ and $X2 = Y1$, resulting in an output with columns **X** and **Y**!!

Criteria : Sort the rows based on the **X** and **Y** columns in ascending order.

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

- The query selects the smallest value (LEAST) from X and Y, and the largest value (GREATEST) from X and Y from the xy table. Then, the result is filtered so that only values that are not in the subquery that swaps the positions of X and Y will be selected.



String Extraction

Consider the content of the table

"**strdata**" below:

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

Create a query to generate output with columns **tanggal**, **qty**, **harga_satuan**, dan **total** !!

Criteria: The data type of the date column is **DATE**, the data type of the **qty**, **harga_satuan**, dan **total** columns is INTEGER. Sort the rows based on the tanggal column.

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

- The query retrieves data from the 'strdata' table, processes it to obtain information about the sales date, quantity sold, unit price, and total sales for each transaction. Processing is done using the CASE clause and string manipulation functions in a subquery, then the data is selected and formatted in the main SELECT statement.



Age Difference

The **people** table is a list of individuals with their date of birth and registration date at an online school. The people table is a list of individuals with their date of birth and registration date at an online school. Create a query to calculate the age of each individual at the time of registration and find the two names with the smallest age difference at registration!! The output should be a table with columns **nama1**, **nama2**, and **selisih**. Criteria: The order of **nama1** and **nama2** is based on the older age.

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	

- The query compares two entities in the 'people' table, p1 and p2, based on the absolute difference between the registration date and the date of birth. First, the query evaluates this difference using a CASE clause, then sorts it based on this difference.

Thank You

Let's Connect



saffarahmatullah@gmail.com



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



<https://github.com/RSaff>