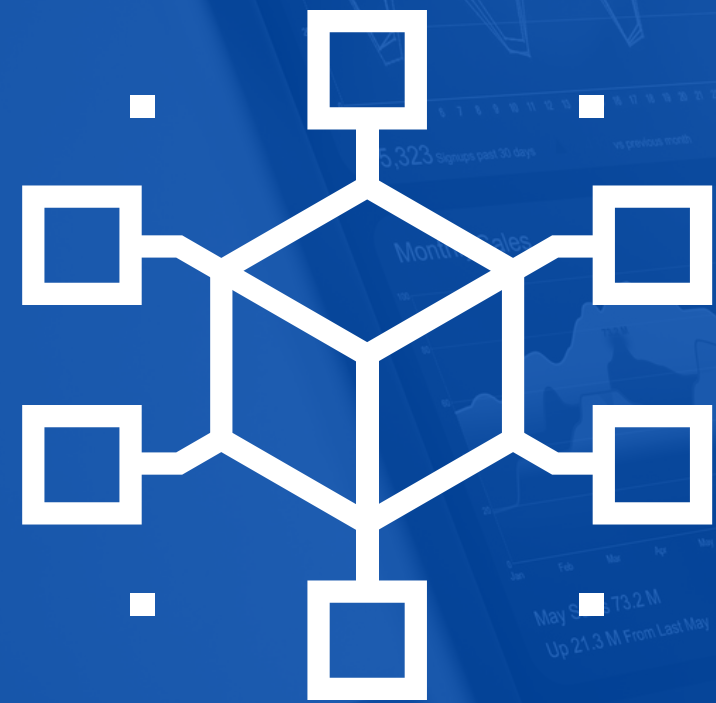


Project SQL



Sales Operation Analyst



BY: SAFFA RAHMATULLAH

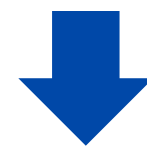
The Dataset Link :

[*https://bit.ly/dataset_dwh_phi*](https://bit.ly/dataset_dwh_phi)

The tools that are used :



**This project employs SQL queries
from Basic to Advanced levels.**



**SELECT, ALIAS, LIMIT, WHERE, Aggregation, Group By,
Having By, Order By, Data Types, CASE WHEN, JOIN**

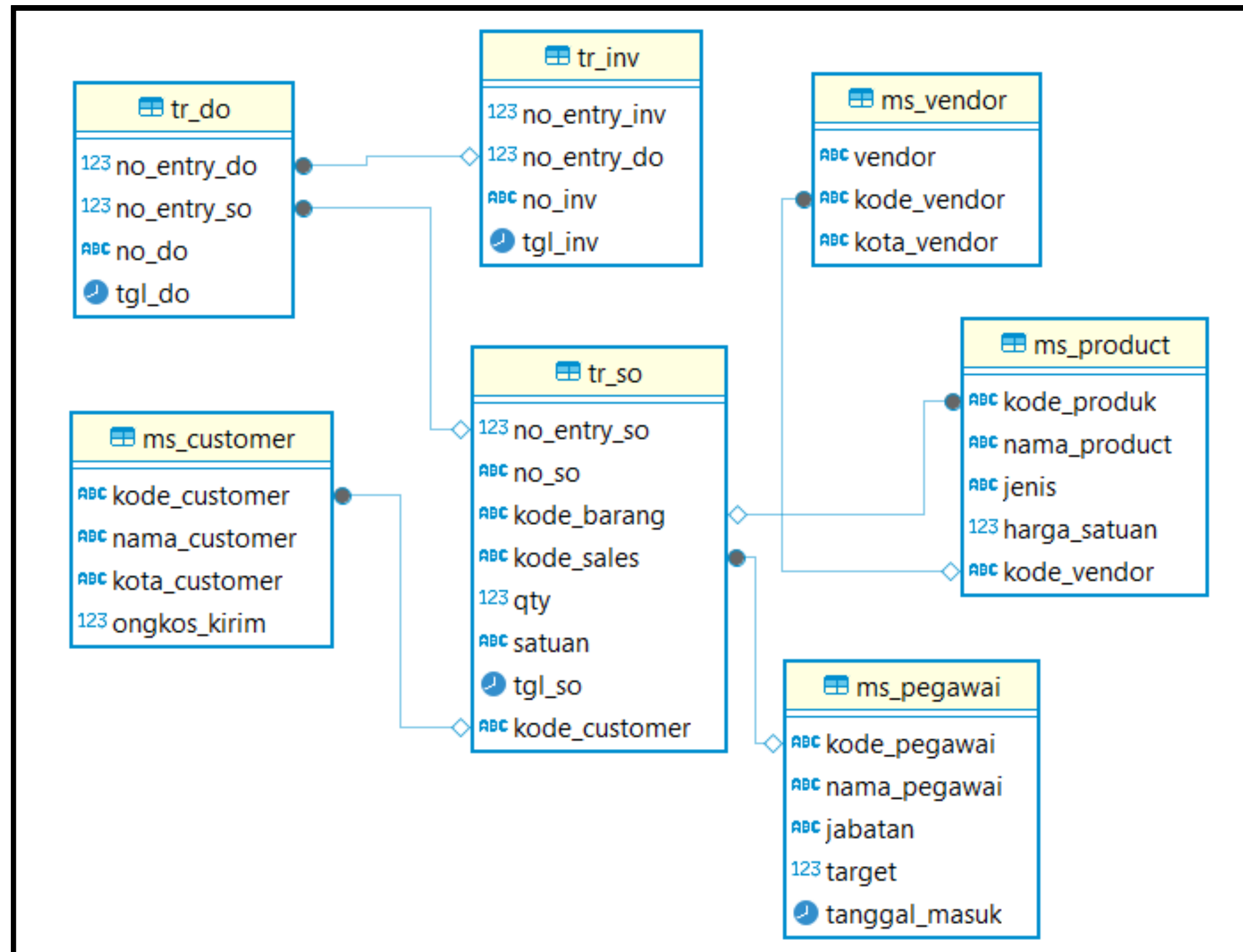
The Github Link :

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

[*https://bit.ly/github_salesoperation_and_lyst*](https://bit.ly/github_salesoperation_and_lyst)

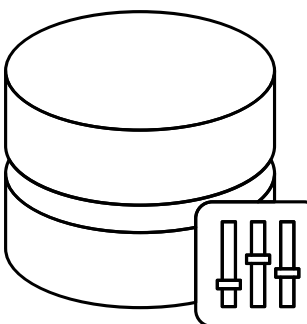


ER Diagram



Data Description

Column Name	Description
no_entry_do	nomor entry delivery order
no_entry_so	nomor entry sales order
no_do	nomor delivery order
tgl_do	tanggal delivery order
no_entry_inv	nomorentry invoice
no_inv	nomor invoice
tgl_inv	tanggal invoice
vendor	vendor
kode_vendor	kode vendor
kota_vendor	kota vendor
kode_customer	kode customer
nama_customer	nama customer
kota_customer	kota customer
ongkos_kirim	ongkos kirim
no_so	nomos sales order
kode_barang	kode barang
kode_sales	kode sales
qty	quantity
satuan	satuan
tgl_so	tanggal sales order
kode_produk	kode produk
nama_product	nama product
jenis	jenis product
harga_satuan	harga satuan
kode_pegawai	kode pegawai
jabatan	jabatan pegawai
target	target pegawai
tanggal_masuk	tanggal masuk pegawai



1

Productivity Sales

Question: Calculate the sales productivity based on the number of Sales Orders (SO) acquired in January, where one SO may contain multiple items.



Query

```

2 • SELECT
3     p.nama_pegawai,
4     COUNT(DISTINCT s.no_so) AS jumlah_so,
5     p.target,
6     CASE
7         WHEN COUNT(DISTINCT s.no_so) < p.target THEN 'ya'
8         ELSE 'tidak'
9     END AS kurang_dari_target
10  FROM
11      tr_so s
12  LEFT JOIN
13      ms_pegawai p on s.kode_sales = p.kode_pegawai
14      AND month(s.tgl_so) = 01
15  GROUP BY
16      p.kode_pegawai
17  ORDER BY
18      p.nama_pegawai;
19

```

Output

Result Grid  Filter Rows: <input type="text"/> Export: 				
	nama_pegawai	jumlah_so	target	kurang_dari_target
▶	Andi	2	5	ya
	Ayu	2	2	tidak
	Budi	3	2	tidak
	Joko	1	2	ya
	Susi	3	2	tidak

- Column '**jumlah_so**' obtains its value from the count of distinct values in the 'no_so' column.
-
- Column '**kurang_dari_target**' consists of two statements, "ya" or "tidak", with the condition that $\text{target} > \text{jumlah_so}$, using a conditional query with CASE WHEN.

Delivery Order Report

Question: The tables required to form the Delivery Order report are the 'tr_do' table. The displayed value is the sum of the price, PPn, and delivery cost. Note: Ppn = 10%.

The price quantity on the Sales Order (SO) must be converted first to obtain its unit value, then multiplied by the unit price of the product, with the following conditions.

Satuan	Qty
Krat	24
Dus	30
Botol	1

Output

Query

```

1  SELECT
2      d.no_do,
3      s.kode_customer,
4      d.tgl_do,
5      CASE
6          WHEN s.satuan = 'Krat' THEN (s.qty * 24)
7          WHEN s.satuan = 'Dus' THEN (s.qty * 30)
8          ELSE s.qty
9      END AS qty,
10     ROUND(SUM(
11         CASE
12             WHEN s.satuan = 'Krat' THEN ((s.qty * 24) * p.harga_satuan)
13             WHEN s.satuan = 'Dus' THEN ((s.qty * 30) * p.harga_satuan)
14             ELSE (s.qty * p.harga_satuan)
15         END * 1.1
16     ) + c.ongkos_kirim, 0) AS amount
17 FROM
18     tr_do d
19 LEFT JOIN
20     tr_so s ON d.no_entry_so = s.no_entry_so

```

```

21 INNER JOIN
22     ms_product p ON s.kode_barang = p.kode_produk
23 INNER JOIN
24     ms_customer c ON s.kode_customer = c.kode_customer
25 GROUP BY
26     d.no_do, s.kode_customer, d.tgl_do
27 ORDER BY
28     d.no_do;

```

no_do	kode_customer	tgl_do	qty	amount
DO001	C001	2018-01-03	36	500200
DO002	C002	2018-01-08	750	3530000
DO003	C001	2018-01-09	18	108160
DO004	C003	2018-01-10	930	3615500
DO005	C003	2018-01-15	1560	20627000
DO006	C002	2018-01-15	630	4916000
DO007	C001	2018-01-08	21	152050
DO008	C001	2018-01-08	240	1345000
DO009	C002	2018-01-12	48	329000
DO010	C004	2018-01-20	1224	6083800
DO011	C003	2018-01-27	76	453000
DO012	C004	2018-01-30	82	521100
DO013	C002	2018-01-20	192	1332200

- Column '**qty**' is obtained from the conversion of its unit value using a conditional query with CASE WHEN
- Column '**amount**' is obtained from the total calculation ((konversi qty * harga_satuan) * 1.1) + ongkos_kirim using a conditional query with CASE WHEN

Customer Accounts Payable Aging Report

Question: Customer accounts payable are the Delivery Orders (DO) that have not been paid yet in the invoice table. Determine the aging of accounts payable as of February 1, 2018.

Criteria: Sort the rows based on the **aging** column in descending order, and the **no_do** column in ascending order.




Query

```

1 • SELECT
2     d.no_do,
3     nama_customer,
4     d.tgl_do,
5     '2018-02-01' AS date_measurement,
6     DATEDIFF('2018-02-01', d.tgl_do) AS aging
7 FROM
8     tr_inv i
9 RIGHT JOIN
10    tr_do d ON i.no_entry_do = d.no_entry_do
11 INNER JOIN
12    tr_so s ON d.no_entry_so = s.no_entry_so
13 INNER JOIN
14    ms_customer c ON s.kode_customer = c.kode_customer
15 WHERE
16     i.no_inv IS NULL
17 ORDER BY
18     aging DESC, d.no_do;

```

Output

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell					
	no_do	nama_customer	tgl_do	date_measurement	aging
▶	DO001	Warung bu Indah	2018-01-03	2018-02-01	29
	DO008	Warung bu Indah	2018-01-08	2018-02-01	24
	DO009	Toko Budi	2018-01-12	2018-02-01	20
	DO010	Toko Bu Endah	2018-01-20	2018-02-01	12
	DO013	Toko Budi	2018-01-20	2018-02-01	12
	DO011	Toko Pak Imin	2018-01-27	2018-02-01	5

- Column '**aging**' is obtained from the difference in days between the 'date_measurement' and 'tgl_do' using the DATEDIFF function

4

Top Three Products Based on Quantity

Question: Using a query and the 'tr_so' table, create an output with columns 'nama_product' and 'qty'.

Criteria: Limit the records to only the top 3 products, sort the rows based on the 'qty' column in descending order, and the 'nama_product' column in ascending order.



Query

```

1  SELECT
2      p.nama_product,
3      SUM(CASE
4          WHEN s.satuan = 'Krat' THEN s.qty * 24
5          WHEN s.satuan = 'Dus' THEN s.qty * 30
6          ELSE s.qty
7      END) AS qty
8  FROM
9      tr_so s
10 INNER JOIN
11     ms_product p ON s.kode_barang = p.kode_produk
12 GROUP BY
13     p.nama_product
14 ORDER BY
15     qty DESC,
16     nama_product ASC
17 LIMIT 3;

```

Output

Result Grid   Filter Rows		
	nama_product	qty
▶	Soda Segar	4663
	Soda gembira ria	3846
	Teh daun pucuk	1224

- Column 'qty' is obtained from the total conversion of the quantity values based on their respective units for each 'nama_product'

5

The Three Vendors with the Highest Sales Volume

Question: Using a query and the 'tr_inv' table, create an output with columns '**vendor**' and '**amount**'.

Criteria: Limit the records to only the top 3 vendors, sort the rows based on the '**amount**' column in descending order, and the '**vendor**' column in ascending order.



Query

```

1 • SELECT
2     v.vendor,
3     SUM(
4         CASE
5             WHEN s.satuan = 'Krat' THEN s.qty * 24 * p.harga_satuan
6             WHEN s.satuan = 'Dus' THEN s.qty * 30 * p.harga_satuan
7             ELSE s.qty * p.harga_satuan
8         END
9     ) AS amount
10 FROM
11     tr_inv i
12 INNER JOIN
13     tr_do d ON i.no_entry_do = d.no_entry_do
14 INNER JOIN
15     tr_so s ON d.no_entry_so = s.no_entry_so
16 INNER JOIN
17     ms_product p ON s.kode_barang = p.kode_produk
18 INNER JOIN
19     ms_vendor v ON p.kode_vendor = v.kode_vendor
20 GROUP BY
21     v.vendor
22 ORDER BY
23     amount DESC,
24     v.vendor ASC
25 LIMIT 3;

```

Output

Result Grid   Filter Rows: <input type="text"/>		
	vendor	amount
▶	PT Alam Maju	19286500
	PT Berjaya	4410000
	PT Maju Berkarya	3255000

- Column '**amount**' is obtained from the total conversion of the values qty * harga_satuan using the SUM function and CASE WHEN.

Thanks

Let's Connect



saffarahmatullah@gmail.com



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



<https://github.com/RSaff>