

/* T1 — Monthly Performance

- Tujuan bisnis: Amati momentum penjualan bulanan.
- Pertanyaan: Tampilkan jumlah order, total revenue, dan avg order value per bulan (di tahun 2025).
- Tabel: orders, order_details
- revenue = unit_price * quantity * (1 - discount). */

SELECT

```
TO_CHAR(o.order_date, 'MM-YYYY') AS month_label,  
COUNT(DISTINCT o.order_id) AS total_orders,  
ROUND(SUM(od.unit_price * od.quantity * (1 - od.discount))::numeric, 2) AS total_revenue,  
ROUND((SUM(od.unit_price * od.quantity * (1 - od.discount))  
/ COUNT(DISTINCT o.order_id))::numeric, 2) AS avg_order_value
```

FROM orders o

JOIN order_details od

```
ON o.order_id = od.order_id
```

WHERE EXTRACT(YEAR FROM o.order_date) = 2025

GROUP BY TO_CHAR(o.order_date, 'MM-YYYY')

ORDER BY month_label;

	month_label ▲	total_orders ▲	total_revenue ▲	avg_order_value ▲
1	01-2025	35	62272.49	1779.21
2	02-2025	33	47409.58	1436.65
3	03-2025	37	64985.56	1756.37
4	04-2025	35	41732.07	1192.34
5	05-2025	41	62938.35	1535.08
6	06-2025	52	85037.14	1635.33
7	07-2025	55	98377.31	1788.68
8	08-2025	74	101278.75	1368.63
9	09-2025	67	138957.34	2073.99
10	10-2025	44	50066.11	1137.87

/* T2 — Shipper SLA (YTD)

- Tujuan bisnis: Evaluasi kurir (SLA).
- Pertanyaan: Untuk order yang sudah dikirim (YTD), tampilkan per kurir: shipped_orders, late_orders (shipped_date > required_date), dan late_rate_%.
- Tabel: orders, shippers */

```
SELECT
  s.company_name,
  COUNT(*) AS shipped_orders,
  SUM(
    CASE
      WHEN o.shipped_date > o.required_date THEN 1
      ELSE 0
    END
  ) AS late_orders,
  ROUND(100.0 * SUM(
    CASE
      WHEN o.shipped_date > o.required_date THEN 1
      ELSE 0
    END) / COUNT(*), 2
  ) AS late_rate_pct
FROM orders o
JOIN shippers s
  ON o.ship_via = s.shipper_id
WHERE (o.shipped_date IS NOT NULL) AND o.shipped_date >= date_trunc('year', CURRENT_DATE)
GROUP BY s.company_name
ORDER BY late_rate_pct DESC;
```

	company_name ▲	shipped_orders ▲	late_orders ▲	late_rate_pct ▲
1	Speedy Express	146	9	6.16
2	United Package	181	10	5.52
3	Federal Shipping	134	3	2.24

/* T3 — Top 10 Products by Revenue (YTD)

- Tujuan bisnis: Fokus ke produk penyumbang pendapatan.
- Pertanyaan: Tampilkan 10 produk ber-revenue tertinggi (YTD) beserta nama kategori dan price_range (<10, 10–20, 20–50, >50).
- Tabel: orders, order_details, products, categories */

SELECT

```
p.product_name,  
c.category_name,  
ROUND(SUM(od.unit_price * od.quantity * (1 - od.discount))::numeric, 2) AS total_revenue,
```

CASE

```
WHEN p.unit_price < 10 THEN '<10'  
WHEN p.unit_price BETWEEN 10 AND 20 THEN '10–20'  
WHEN p.unit_price BETWEEN 20 AND 50 THEN '20–50'  
ELSE '>50'
```

```
END AS price_range
```

```
FROM order_details od
```

```
JOIN orders o
```

```
ON od.order_id = o.order_id
```

```
JOIN products p
```

```
ON od.product_id = p.product_id
```

```
JOIN categories c
```

```
ON p.category_id = c.category_id
```

```
WHERE
```

```
o.order_date >= DATE_TRUNC('year', CURRENT_DATE)
```

```
GROUP BY
```

```
p.product_name,  
c.category_name,  
price_range
```

```
ORDER BY
```

```
total_revenue DESC
```

```
LIMIT 10;
```

	product_name ▲	category_name ▲	total_revenue ▲	price_range ▲
1	Côte de Blaye	Beverages	78588.87	>50
2	Thüringer Rostbratwurst	Meat/Poultry	57042.43	>50
3	Raclette Courdavault	Dairy Products	43626.00	>50
4	Camembert Pierrot	Dairy Products	29934.62	20–50
5	Tarte au sucre	Confections	28256.29	20–50
6	Manjimup Dried Apples	Produce	26513.25	>50
7	Gnocchi di nonna Alice	Grains/Cereals	22877.90	20–50
8	Alice Mutton	Meat/Poultry	19055.40	20–50
9	Carnarvon Tigers	Seafood	18584.37	>50
10	Uncle Bob's Organic Dried Pears	Produce	18532.50	20–50

/* T4 — Sales Rep Scorecard (YTD)

- Tujuan bisnis: Ringkas performa per sales rep.
- Pertanyaan: Per karyawan tampilkan jumlah order, avg order value, dan % order besar (order_total > 1000).
- Tabel: orders, order_details, employees */

```
WITH order_totals AS (  
  SELECT  
    o.order_id,  
    o.employee_id,  
    SUM(od.unit_price * od.quantity * (1 - od.discount)) AS order_total  
  FROM orders o  
  JOIN order_details od  
    ON o.order_id = od.order_id  
  WHERE o.order_date >= DATE_TRUNC('year', CURRENT_DATE) -- YTD  
  GROUP BY o.order_id, o.employee_id  
)  
  
SELECT  
  e.employee_id,  
  e.first_name || ' ' || e.last_name AS employee_name,  
  COUNT(ot.order_id) AS total_orders,  
  ROUND(AVG(ot.order_total)::numeric, 2) AS avg_order_value,  
  ROUND(100.0 * SUM(CASE WHEN ot.order_total > 1000 THEN 1 ELSE 0 END)  
    / COUNT(ot.order_id)::numeric, 2) AS pct_large_orders  
FROM order_totals ot  
JOIN employees e  
  ON ot.employee_id = e.employee_id  
GROUP BY  
  e.employee_id, employee_name  
ORDER BY  
  total_orders DESC;
```

	employee_id ▲	employee_name ▲	total_orders ▲	avg_order_value ▲	pct_large_orders ▲
1	4	Margaret Peacock	87	1331.06	44.83
2	1	Nancy Davolio	73	1683.25	50.68
3	3	Janet Leverling	72	1677.94	55.56
4	2	Andrew Fuller	59	1780.89	44.07
5	8	Laura Callahan	53	1428.06	52.83
6	7	Robert King	39	1862.82	38.46
7	6	Michael Suyama	37	1056.00	37.84
8	9	Anne Dodsworth	29	2094.16	51.72
9	5	Steven Buchanan	24	1681.26	58.33

/* T5 — New vs Loyal Customers (YTD)

- Tujuan bisnis: Pahami porsi akuisisi vs retensi.
- Pertanyaan: Segmentasi order YTD ke NEW (first order customer di tahun ini) vs LOYAL (first order sebelum tahun ini). Tampilkan orders dan revenue.
- Tabel: orders, order_details
- Output: segment, orders, revenue */

```
WITH customer_first_order AS (
```

```
  SELECT
```

```
    customer_id,
```

```
    MIN(order_date) AS first_order_date
```

```
  FROM orders
```

```
  GROUP BY customer_id
```

```
),
```

```
order_totals AS (
```

```
  SELECT
```

```
    o.order_id,
```

```
    o.customer_id,
```

```
    SUM(od.unit_price * od.quantity * (1 - od.discount)) AS revenue,
```

```
    o.order_date
```

```
  FROM orders o
```

```
  JOIN order_details od
```

```
    ON o.order_id = od.order_id
```

```
  WHERE o.order_date >= DATE_TRUNC('year', CURRENT_DATE)
```

```
  GROUP BY o.order_id, o.customer_id, o.order_date
```

```
),
```

```
segmented_orders AS (
```

```
  SELECT
```

```
    ot.*,
```

```
    CASE
```

```
      WHEN cfo.first_order_date >= DATE_TRUNC('year', CURRENT_DATE)
```

```
      THEN 'NEW'
```

```
      ELSE 'LOYAL'
```

```
    END AS segment
```

```
  FROM order_totals ot
```

```
  JOIN customer_first_order cfo
```

```
    ON ot.customer_id = cfo.customer_id
```

```
)
```

```
SELECT
```

```
  segment,
```

```
  COUNT(order_id) AS orders,
```

```
  ROUND(SUM(revenue)::numeric, 2) AS total_revenue
```

```
FROM segmented_orders
```

```
GROUP BY segment;
```

	segment ▲	orders ▲	total_revenue ▲
1	NEW	27	16969.15
2	LOYAL	446	736085.54