



MySkill | *#RintisKarirImpian*

Portfolio - Intensive Bootcamp

# Exercise 1

## SQL

**Owner:** Devita Eka Ratriningrum

Build your skill and portfolio via [myskill.id/bootcamp](https://myskill.id/bootcamp)

# Dataset

## order\_detail:

- |                    |                                                                  |
|--------------------|------------------------------------------------------------------|
| 1. id              | → angka unik dari order / <i>id_order</i>                        |
| 2. customer_id     | → angka unik dari pelanggan                                      |
| 3. order_date      | → tanggal saat dilakukan transaksi                               |
| 4. sku_id          | → angka unik dari produk (sku adalah <i>stock keeping unit</i> ) |
| 5. price           | → harga yang tertera pada tagging harga                          |
| 6. qty_ordered     | → jumlah barang yang dibeli oleh pelanggan                       |
| 7. before_discount | → nilai harga total dari produk ( $price * qty\_ordered$ )       |
| 8. discount_amount | → nilai diskon product total                                     |
| 9. after_discount  | → nilai harga total produk ketika sudah dikurangi dengan diskon  |
| 10. is_gross       | → menunjukkan pelanggan belum membayar pesanan                   |
| 11. is_valid       | → menunjukkan pelanggan sudah melakukan pembayaran               |
| 12. is_net         | → menunjukkan transaksi sudah selesai                            |
| 13. payment_id     | → angka unik dari metode pembayaran                              |

# Dataset



## **sku\_detail:**

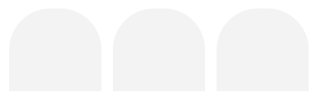
- |               |                                                                       |
|---------------|-----------------------------------------------------------------------|
| 1. id         | → angka unik dari produk (dapat digunakan untuk <i>key</i> saat join) |
| 2. sku_name   | → nama dari produk                                                    |
| 3. base_price | → harga barang yang tertera pada tagging harga / <i>price</i>         |
| 4. cogs       | → <i>cost of goods sold</i> / total biaya untuk menjual 1 produk      |
| 5. category   | → kategori produk                                                     |

## **customer\_detail:**

- |                    |                                                             |
|--------------------|-------------------------------------------------------------|
| 1. id              | → angka unik dari pelanggan                                 |
| 2. registered_date | → tanggal pelanggan mulai mendaftarkan diri sebagai anggota |

## **Payment\_detail:**

- |                   |                                     |
|-------------------|-------------------------------------|
| 1. id             | → angka unik dari metode pembayaran |
| 2. payment_method | → metode pembayaran yang digunakan  |



# Code Dataset

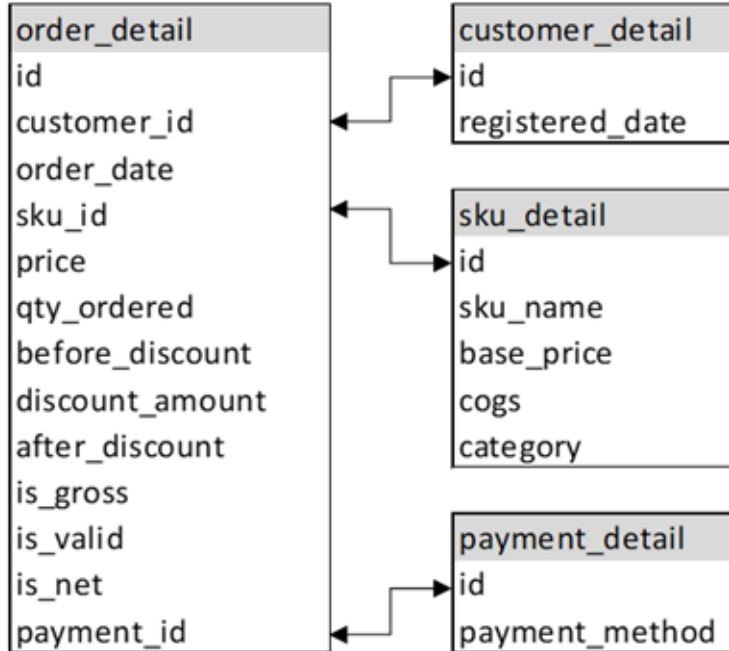
```
CREATE TABLE IF NOT EXISTS customer_detail (  
  id TEXT,  
  registered_date DATE  
)
```

```
CREATE TABLE IF NOT EXISTS payment_detail (  
  id INT,  
  payment_method TEXT  
)
```

```
CREATE TABLE IF NOT EXISTS sku_detail (  
  id TEXT,  
  sku_name TEXT,  
  base_price FLOAT,  
  cogs FLOAT,  
  category TEXT  
)
```

```
CREATE TABLE IF NOT EXISTS order_detail (  
  id TEXT,  
  customer_id TEXT,  
  order_date DATE,  
  sku_id TEXT,  
  price INT,  
  qty_ordered INT,  
  before_discount FLOAT,  
  discount_amount FLOAT,  
  after_discount FLOAT,  
  is_gross INT,  
  is_valid INT,  
  is_net INT,  
  payment_id INT  
)
```

# Relasi Antar Tabel



## Description:

- **customer\_id** pada tabel **order\_detail** merupakan foreign key yang diambil dari primary key **id** pada tabel **customer\_detail**
- **sku\_id** pada tabel **order\_detail** merupakan foreign key yang diambil dari primary key **id** pada tabel **sku\_detail**
- **payment\_id** pada tabel **order\_detail** merupakan foreign key yang diambil dari primary key **id** pada tabel **payment\_detail**

# Pertanyaan 1:

Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after\_discount) paling besar?

## Answer:

```
SELECT
    to_char(order_date, 'Month') month_2021,
    SUM(after_discount) AS total_after_disc
FROM order_detail
WHERE
    EXTRACT(YEAR FROM order_date) = 2021
    AND is_valid = 1
GROUP BY 1
ORDER BY 2 DESC
```

Untuk mencari tahu nilai transaksi paling besar pada tahun 2021:

- Mengubah format kolom **order\_date** pada tabel **order\_detail** dari tipe **datetime** menjadi **STRING (to\_char)** agar dapat menampilkan nama bulan dan bukan angka
- Menjumlahkan semua nilai dari kolom **after\_discount** dan diberi nama sebagai **total\_after\_disc** dari tabel **order\_detail**
- Melakukan filter dimana tahun yang akan kita pilih adalah 2021 dengan menggunakan **EXTRACT YEAR** pada kolom **order\_date** dikarenakan tipe **order\_date** merupakan **datetime** dan **is\_valid = 1**

# Pertanyaan 1:

Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after\_discount) paling besar?

## Result:

month_2021	total_after_disc
August	227862744
December	217309963
October	207603259.875
November	180396009.703125
July	148007735.00156212
September	145943335.24414062
June	43154552
January	36822126.5
February	35611797
May	34163856
March	23643062
April	22208472.59375

Dari hasil dari coding sebelumnya, maka dapat disimpulkan bahwa bulan **Agustus** merupakan bulan dimana nilai total transaksi tertinggi pada tahun 2021

# Pertanyaan 2:

Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?

## Answer:

```
SELECT
    sk.category,
    SUM(after_discount) AS total_after_disc
FROM order_detail AS od
    LEFT JOIN sku_detail AS sk
    ON od.sku_id = sk.id
WHERE
    EXTRACT(Year FROM od.order_date) = 2022
    AND od.is_valid = 1
GROUP BY 1
ORDER BY 2 DESC
```

Untuk mencari tahu nilai transaksi pada kategori mana paling besar pada tahun 2022:

- Menentukan kolom apa saja yang ingin ditampilkan, dikarenakan kita harus mencari tau nama kategori yang memiliki transaksi terbesar maka kita akan menampilkan kolom **category** pada tabel **sku\_detail** dan penjumlahan dari kolom **after\_discount** pada tabel **order\_detail** yang kemudian hasil penjumlahannya akan disimpan sebagai kolom baru bernama **total\_after\_disc**
- Menggunakan **LEFT JOIN** dikarenakan kita akan menggunakan 2 tabel yaitu **order\_detail** dan **sku\_detail** dengan menghubungkan kolom **id** pada tabel **sku\_detail** dengan **sku\_id** pada tabel **order\_detail**
- Melakukan filter tahun, dimana tahun yang digunakan adalah tahun 2022 dan **is\_valid = 1**
- Menggunakan **GROUP BY** dikarenakan untuk memisahkan antara fungsi perhitungan aritmatika dengan kolom biasa.
- Diurutkan berdasarkan penjumlahan secara **descending**



# Pertanyaan 2:

Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?

## Result:

category	total_after_disc
Mobiles & Tablets	918451576
Entertainment	365344151
Appliances	316358100
Computing	214028543.375
Men Fashion	135588253
Women Fashion	93014970.625
Home & Living	79483716.1953125
Health & Sports	54235579.5859375
Beauty & Grooming	46211019.1875

Dari hasil dari coding sebelumnya, maka dapat disimpulkan bahwa kategori **Mobiles & Tablets** merupakan kategori yang memiliki transaksi paling besar pada tahun 2022.

# Pertanyaan 3:

Bandungkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022.

## Answer:

```
WITH a AS (  
  SELECT  
    sd.category,  
    SUM(CASE WHEN EXTRACT(YEAR FROM od.order_date) = 2021 THEN od.after_discount END) AS total_2021,  
    SUM(CASE WHEN EXTRACT(YEAR FROM od.order_date) = 2022 THEN od.after_discount END) AS total_2022  
  FROM order_detail AS od LEFT JOIN sku_detail AS sd ON sd.id = od.sku_id  
  WHERE is_valid = 1  
  GROUP BY 1  
  ORDER BY 2 DESC  
)  
SELECT  
  a.*,  
  total_2022 - total_2021 AS pengurangan_2022_2021  
FROM a  
ORDER BY 4 DESC
```

# Pertanyaan 3:

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022.

## Answer:

Untuk mencari tahu kategori mana saja yang mengalami peningkatan dan penurunan dari tahun 2021 ke 2022:

- Menggunakan fungsi **WITH clause** untuk memisahkan 2 query utama
- Dalam **WITH clause** kita akan menampilkan kolom **category** yang diambil dari tabel **sku\_detail**
- Menampilkan hasil penjumlahan pada kolom **after\_discount** pada tabel **order\_detail** dari masing-masing tahun menggunakan fungsi **CASE WHEN** untuk memisahkan antara hasil penjumlahan tahun 2021 dan 2022
- Menggunakan **LEFT JOIN** dikarenakan kita menggunakan 2 tabel yaitu **sku\_detail** dan **order\_detail** dengan parameter kolom **id** pada **sku\_detail** dengan kolom **sku\_id** pada tabel **order\_detail**
- Menggunakan filter **is\_valid = 1**
- Mengurutkan dari hasil penjumlahan tahun 2021 secara **descending**
- Setelah isi query dari clausa **WITH** selesai, kita dapat memanggil kembali pada query selanjutnya untuk ditampilkan
- Melakukan pengurangan pada hasil penjumlahan yang ada pada clausa **WITH** yang selanjutnya disimpan sebagai kolom **pengurangan\_2022\_2021**

# Pertanyaan 3:

Bandinkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022.

## Result: Peningkatan

category	total_2021	total_2022	pengurangan_2022_2021
Mobiles & Tablets	370606718	918451576	547844858
Entertainment	162326357.375	365344151	203017793.625
Appliances	218550177	316358100	97807923
Men Fashion	58628198	135588253	76960055
Computing	172878860	214028543.375	41149683.375
Home & Living	45797873	79483716.1953125	33685843.1953125
Health & Sports	33837965.6015625	54235579.5859375	20397613.984375
Women Fashion	84045961.3984375	93014970.625	8969009.2265625
School & Education	11558982.392187119	17362465.301635742	5803482.909448624
Superstore	28828088	32643266.52734375	3815178.52734375

Dari hasil dari coding sebelumnya, maka dapat disimpulkan bahwa kategori **Mobiles & Tablets** merupakan kategori yang mengalami peningkatan transaksi dari tahun 2021 ke 2022

# Pertanyaan 3:

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022.

## Result: Penurunan

category	total_2021	total_2022	pengurangan_2022_2021
Others	40468515.75	21744646.021484375	-18723869.728515625
Books	10124596	6792519.203125	-3332076.796875
Beauty & Grooming	46047360	46211019.1875	163659.1875
Kids & Baby	23971057.796875	25931276.83984375	1960219.04296875
Soghaat	15056202.603515625	17658332	2602129.396484375
Superstore	28828088	32643266.52734375	3815178.52734375
School & Education	11558982.392187119	17362465.301635742	5803482.909448624
Women Fashion	84045961.3984375	93014970.625	8969009.2265625
Health & Sports	33837965.6015625	54235579.5859375	20397613.984375

SELECT

a.\*,

total\_2022 - total\_2021

AS pengurangan\_2022\_2021

FROM a

ORDER BY 4 ASC

Masih dengan coding yang sama dengan sebelumnya dan mengubah sedikit coding sebelumnya dari **DESC** menjadi **ASC** pada query kedua, maka dapat disimpulkan bahwa kategori **Others** merupakan kategori yang mengalami penurunan transaksi dari tahun 2021 ke 2022

# Pertanyaan 4:

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022

## Answer:

```
SELECT
    pd.payment_method,
    COUNT(DISTINCT od.id) AS total_payment
FROM order_detail AS od LEFT JOIN payment_detail AS pd
    ON od.payment_id = pd.id
WHERE
    EXTRACT(Year FROM od.order_date) = 2022
    AND od.is_valid = 1
GROUP BY 1
ORDER BY 2 DESC LIMIT 5
```

Untuk mencari tahu 5 metode pembayaran populer pada tahun 2022:

- Menampilkan **payment\_method** yang diambil dari tabel **payment\_detail** dan menggunakan fungsi **COUNT DISTINCT** pada kolom **id** yang berada pada tabel **order\_detail** agar **id** yang sama tidak akan terhitung kembali.
- Menggunakan **LEFT JOIN** dikarenakan kita menggunakan 2 tabel yaitu **payment\_detail** dan **order\_detail** dengan parameter kolom **id** pada **payment\_detail** dengan kolom **payment\_id** pada tabel **order\_detail**
- Menggunakan filter tahun 2022 menggunakan **EXTRACT YEAR** dan **is\_valid = 1**
- Menggunakan **LIMIT 5** dikarenakan kita hanya akan menampilkan 5 pembayaran populer

# Pertanyaan 4:

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022

## Result:

payment_method	total_payment
cod	1809
Payaxis	181
customercredit	75
Easypay	69
jazzwallet	26

Dari hasil dari coding sebelumnya, maka dapat disimpulkan bahwa pembayaran **COD** merupakan pembayaran paling populer dan diikuti oleh **Payaxis**, **customercredit**, **Easypay**, dan **jazzwallet**

# Pertanyaan 5:

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.

1. Samsung 2. Apple 3. Sony 4. Huawei 5. Lenovo

## Answer:

```
SELECT
  CASE
    WHEN lower(sd.sku_name) LIKE '%samsung%' THEN 'Samsung'
    WHEN lower(sd.sku_name) LIKE '%apple%' OR lower(sd.sku_name) LIKE '%iphone%' THEN 'Apple'
    WHEN lower(sd.sku_name) LIKE '%sony%' THEN 'Sony'
    WHEN lower(sd.sku_name) LIKE '%huawei%' THEN 'Huawei'
    WHEN lower(sd.sku_name) LIKE '%lenovo%' THEN 'Lenovo'
  END AS merk,
  SUM(od.after_discount) AS total
FROM order_detail AS od
LEFT JOIN sku_detail AS sd
  ON od.sku_id = sd.id
```



# Pertanyaan 5:

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.

1. Samsung 2. Apple 3. Sony 4. Huawei 5. Lenovo

**Answer:**

```
WHERE
    to_char(order_date, 'yyyy-mm-dd') BETWEEN '2022-01-01' AND '2022-12-31' AND
    (
        lower(sd.sku_name) LIKE '%samsung%' OR
        lower(sd.sku_name) LIKE '%apple%' OR
        lower(sd.sku_name) LIKE '%iphone%' OR
        lower(sd.sku_name) LIKE '%sony%' OR
        lower(sd.sku_name) LIKE '%huawei%' OR
        lower(sd.sku_name) LIKE '%lenovo%'
    )
    AND od.is_valid = 1
GROUP BY 1
ORDER BY total DESC
```

# Pertanyaan 5:

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.

1. Samsung 2. Apple 3. Sony 4. Huawei 5. Lenovo

## Answer:

Untuk menampilkan nilai transaksi sesuai urutannya:

- Membuat kondisi melalui klausa **CASE WHEN** untuk mencari nama yang mengandung kata seperti samsung menggunakan fungsi **LOWER** dikarenakan kita tidak tahu apakah data menggunakan huruf besar semua ataupun kecil semua, maka dari itu fungsi **LOWER** digunakan untuk membuat nilai menjadi kecil dan penggunaan fungsi **LIKE** untuk mencari kata yang mirip ataupun mengandung kata samsung dan menggunakan symbol '%' didepan dan dibelakang kalimat samsung untuk memastikan dimanapun kata samsung berada maka data tersebut akan dikategorikan sebagai Samsung, dan dari hasil klausa **CASE WHEN** tersebut akan disimpan sebagai kolom baru bernama merk.
- Menampilkan penjumlahan kolom **after\_discount** dari tabel **order\_detail**
- Menggunakan **LEFT JOIN** dikarenakan kita menggunakan 2 tabel yaitu **sku\_detail** dan **order\_detail** dengan parameter kolom **id** pada **sku\_detail** dengan kolom **sku\_id** pada tabel **order\_detail**
- Menggunakan filter tahun 2022 dan mencari kembali kalimat yang mengandung kata yang ingin dicari serta **is\_valid = 1**

# Pertanyaan 5:

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.

1. Samsung 2. Apple 3. Sony 4. Huawei 5. Lenovo

## Result:

merk	total
Samsung	412357846
Apple	153144592
Sony	32343178
Huawei	30942826
Lenovo	23621590.375

Dari hasil dari coding sebelumnya, maka dapat ditampilkan nilai transaksi sesuai urutan produk yang ingin diketahui.



MySkill | *#RintisKarirImpian*

**Portfolio - Intensive Bootcamp**

**Thank  
You!**