**Exercise SQL**

Berikut dataset yang akan digunakan:

1. order\_detail

2. sku\_detail

3. customer\_detail

4. payment\_detail

Data yang digunakan adalah data yang berasal dari Tokopedia (***bukan data sesungguhnya***). Mengenai penjelasan dataset adalah sebagai berikut:

**order\_detail**:

1. id → angka unik dari order / *id\_order*

2. customer\_id → angka unik dari pelanggan

3. order\_date → tanggal saat dilakukan transaksi

4. sku\_id → angka unik dari produk (sku adalah *stock keeping unit*)

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

**sku\_detail:**

1. id → angka unik dari produk (dapat digunakan untuk *key* saat join)

2. sku\_name → nama dari produk

3. base\_price → harga barang yang tertera pada tagging harga / *price*

4. cogs → *cost of goods sold* / total biaya untuk menjual 1 produk

5. category → kategori produk

**customer\_detail:**

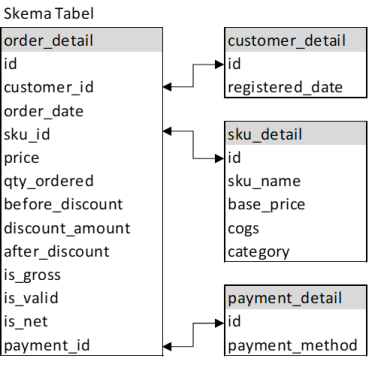
1. id → angka unik dari pelanggan

2. registered\_date → tanggal pelanggan mulai mendaftarkan diri sebagai anggota

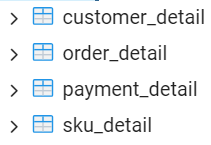
**Payment\_detail:**

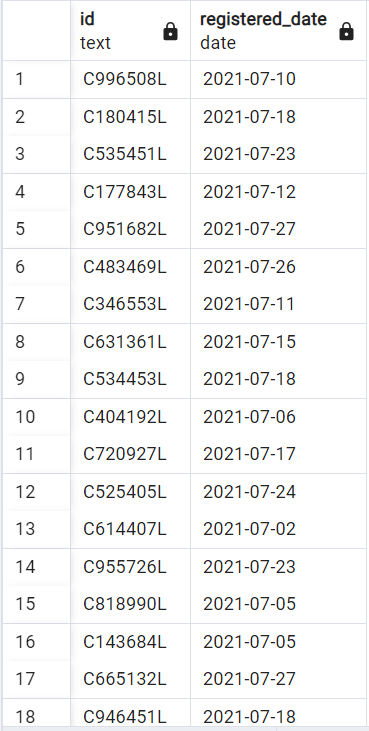
1. id → angka unik dari metode pembayaran

2. payment\_method → metode pembayaran yang digunakan

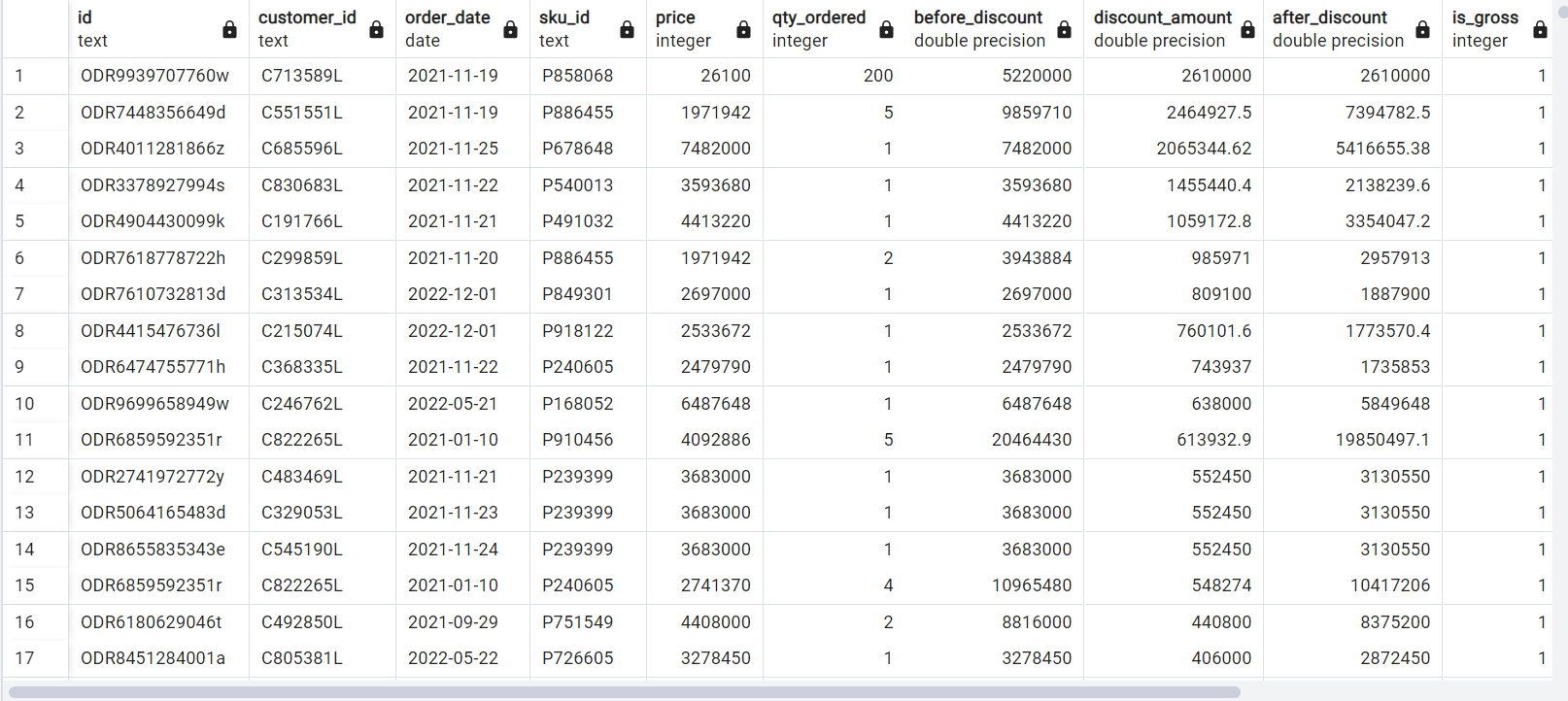
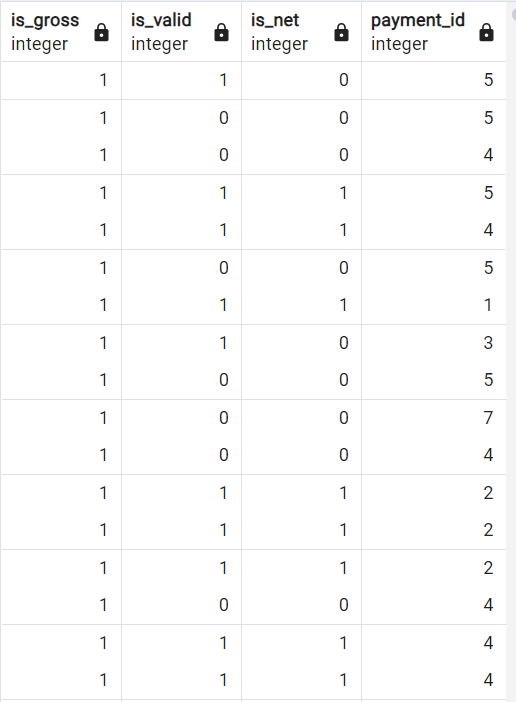


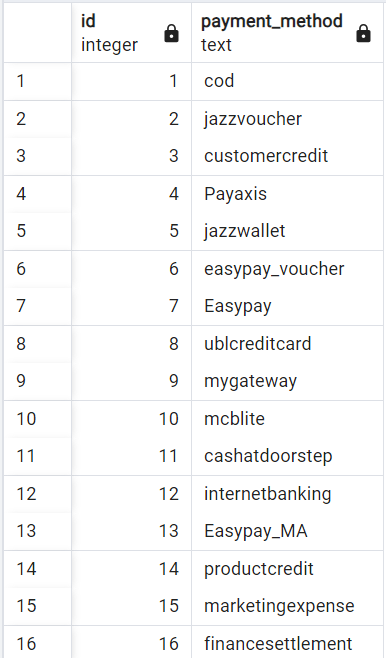
Dataset all



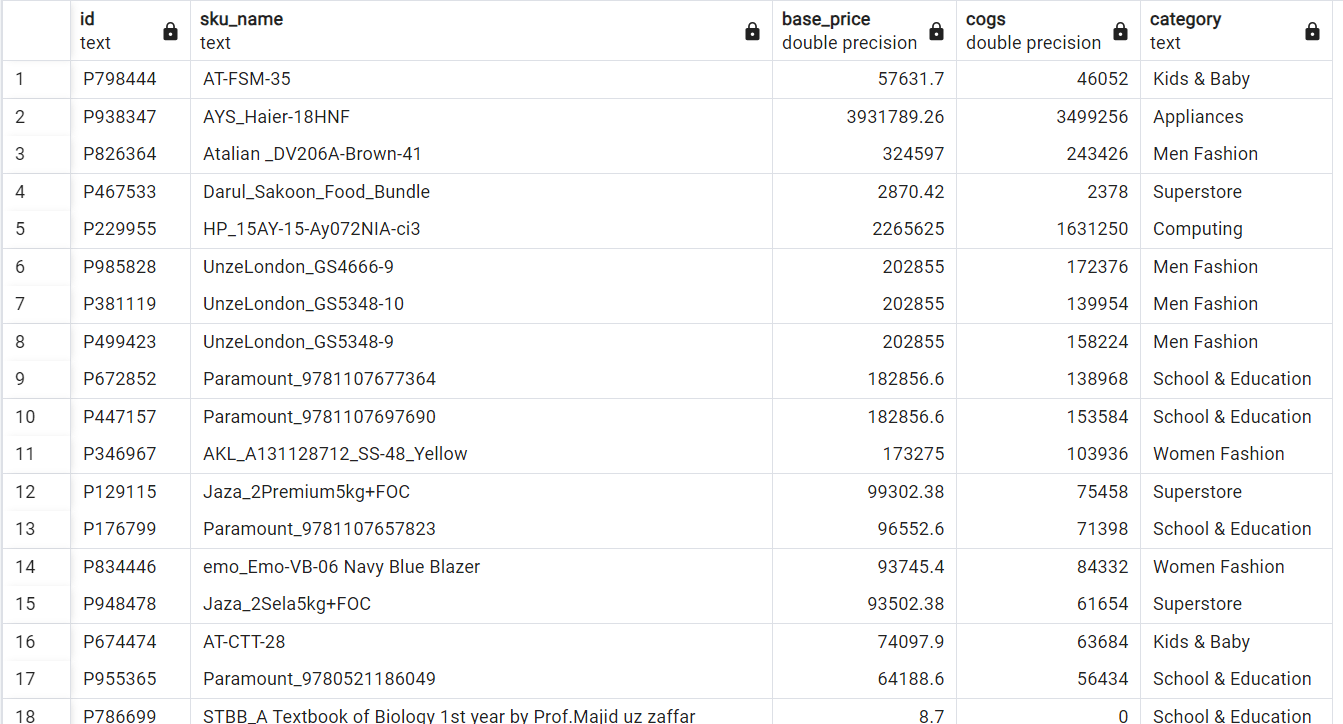
Sampel Data customer\_detail  


Sampel data order\_detail



Sampel data payment\_detail  


Sampel data sku\_detail



Pertanyaan

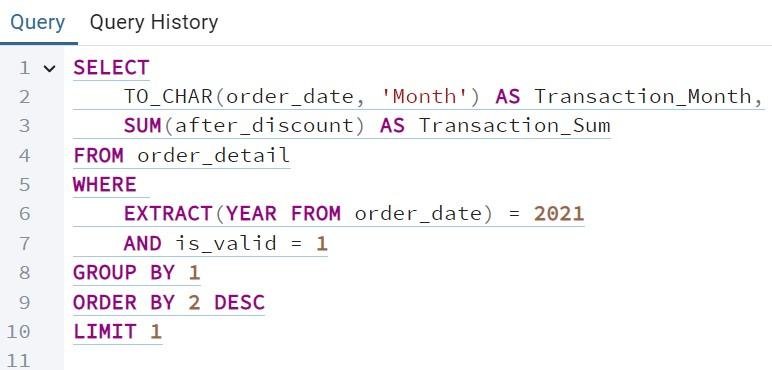
Nomor 1

Q: Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (*after\_discount*) paling besar? Gunakan **is\_valid = 1** untuk memfilter data transaksi. Source table: order\_detail

Q1 Logical Thinking:

Pertanyaan tersebut meminta untuk dibuatkan table data yang berisi nama bulan pada 2021 yang memiliki total nilai transaksi paling besar dengan harga setelah diskon (after\_discount) yang telah dibayarkan (is\_valid = 1). Dengan demikian, pada table tersebut harus ada 2 kolom: kolom 1 berisi nama bulan dan kolom 2 berisi data total nilai transaksi pada bulan tersebut. Table tersebut juga dibatasi hanya 1 baris data saja karena meminta 1 nilai paling besar. Dari pertanyaan tersebut juga dapat diketahui bahwa sumber data yg diperlukan ialah order\_detail

Q1 Query:



Q1 Query penjelasan:

Fungsi SELECT dan Agregat:

* **SELECT TO\_CHAR(order\_date, 'Month') AS Transaction\_Month** untuk memilih bulan dari kolom order\_date dan menampilkannya dalam bentuk data **character** serta mengaliaskan kolom tersebut menjadi Transaction\_Month
* **SUM(after\_discount) AS Transaction\_Sum** untuk memilih kolom after\_discount dan menjumlahkan nilai di dalamnya (data berbentuk **integer**) berdasarkan bulan di kolom Transaction\_Month lalu mengaliaskan nama kolom menjadi Transaction\_Sum

Sumber Data:

* **FROM order\_detail** untuk menjadikan table order\_detail sebagai sumber data

Kalusa WHERE:

* **WHERE EXTRACT(YEAR FROM order\_date) = 2021 AND is\_valid = 1** untuk membatasi hasil berdasarkan kondisi 1 (hanya pada transaksi di tahun 2021) dan kondisi 2 (is\_valid = 1 atau transaksi sudah dibayarkan)

Pernyataan GROUP BY

* **GROUP BY 1** mengelompokkan hasil query berdasarkan kolom 1 yaitu Transaction\_Month

Perintah ORDER BY

* **ORDER BY 2 DESC** untuk mengurutkan data berdasarkan kolom 2 yaitu Transaction\_Sum dan dari nilai terbesar ke nilai terkecil

Klausa LIMIT

* **LIMIT 1** untuk membatasi baris yang ditampilkan menjadi 1 baris

Q1 Query Result:



Q1 Insight:

Data menunjukkan bahwa bulan di tahun 2021 dengan total nilai transaksi yang telah dibayarkan dengan harga after\_discount terbesar ialah bulan **August** dengan nilai transaksi sebesar **227**.**862**.**744**

Nomor 2

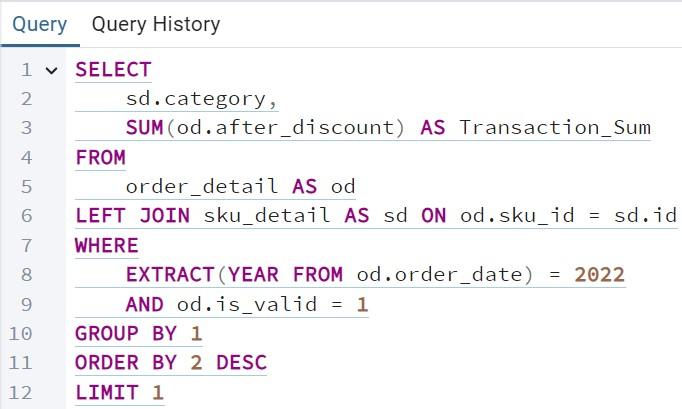
Q: Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar? Gunakan **is\_valid = 1** untuk memfilter data transaksi.

Source table: order\_detail, sku\_detail

Q2 Logical Thinking:

Pertanyaan tersebut meminta untuk dibuatkan table data yang berisi nama kategori pada 2022 yang memiliki total nilai transaksi paling besar dengan harga setelah diskon (after\_discount) yang telah dibayarkan (is\_valid = 1). Dengan demikian, pada table tersebut harus ada 2 kolom: kolom 1 berisi nama bulan dan kolom 2 berisi data total nilai transaksi pada bulan tersebut. Table tersebut juga dibatasi hanya 1 baris data saja karena meminta 1 nilai paling besar. Dari pertanyaan tersebut juga dapat diketahui bahwa sumber data yg diperlukan ialah gabungan table order\_detail dengan table sku\_detail karena hasil data tersebut memerlukan data di kolom kategori dari table sku\_detail serta data di kolom after\_discount dari table order\_detail.

Q2 Query:



Q2 Query Penjelasan:

Fungsi SELECT dan Agregat:

* **SELECT sd.category** untuk menampilkan nama kategori dari kolom category di table sku\_detail
* **SUM(od.after\_discount) AS Transaction\_Sum** untuk memilih kolom after\_discount dari table order\_detail dan menjumlahkan nilai di dalamnya (data berbentuk **integer**) berdasarkan kategoti di kolom category lalu mengaliaskan nama kolom menjadi Transaction\_Sum

Sumber Data:

* **FROM order\_detail AS od** untuk menjadikan table order\_detail sebagai sumber data dengan alias od
* **LEFT JOIN sku\_detail AS sd ON od.sku\_id = sd.id** menggabungkan table sku\_detail (alias sd) dengan order\_detail dengan primary key yaitu kolom id dari table sku\_detail dan sku\_id dari kolom order\_detail

Kalusa WHERE:

* **WHERE EXTRACT(YEAR FROM order\_date) = 2022 AND is\_valid = 1** untuk membatasi hasil berdasarkan kondisi 1 (hanya pada transaksi di tahun 2022) dan kondisi 2 (is\_valid = 1 atau transaksi sudah dibayarkan)

Pernyataan GROUP BY

* **GROUP BY 1** mengelompokkan hasil query berdasarkan kolom 1 yaitu category

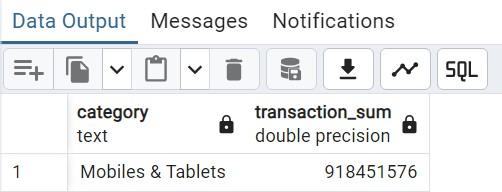
Perintah ORDER BY

* **ORDER BY 2 DESC** untuk mengurutkan data berdasarkan kolom 2 yaitu Transaction\_Sum dan dari nilai terbesar ke nilai terkecil

Klausa LIMIT

* **LIMIT 1** untuk membatasi baris yang ditampilkan menjadi 1 baris

Q2 Query result:



Q2 Insight:

Data menunjukan bahwa kategori dengan transaksi terbayar dengan harga after\_discount di tahun 2022 terbanyak ialah kategori **Mobiles & Tablets** dengan nilai transaksi sebesar **918**.**451**.**576**

Nomor 3

Q: 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. Gunakan **is\_valid = 1** untuk memfilter data transaksi.

Source table: order\_detail, sku\_detail

Q3 Logical thinking:

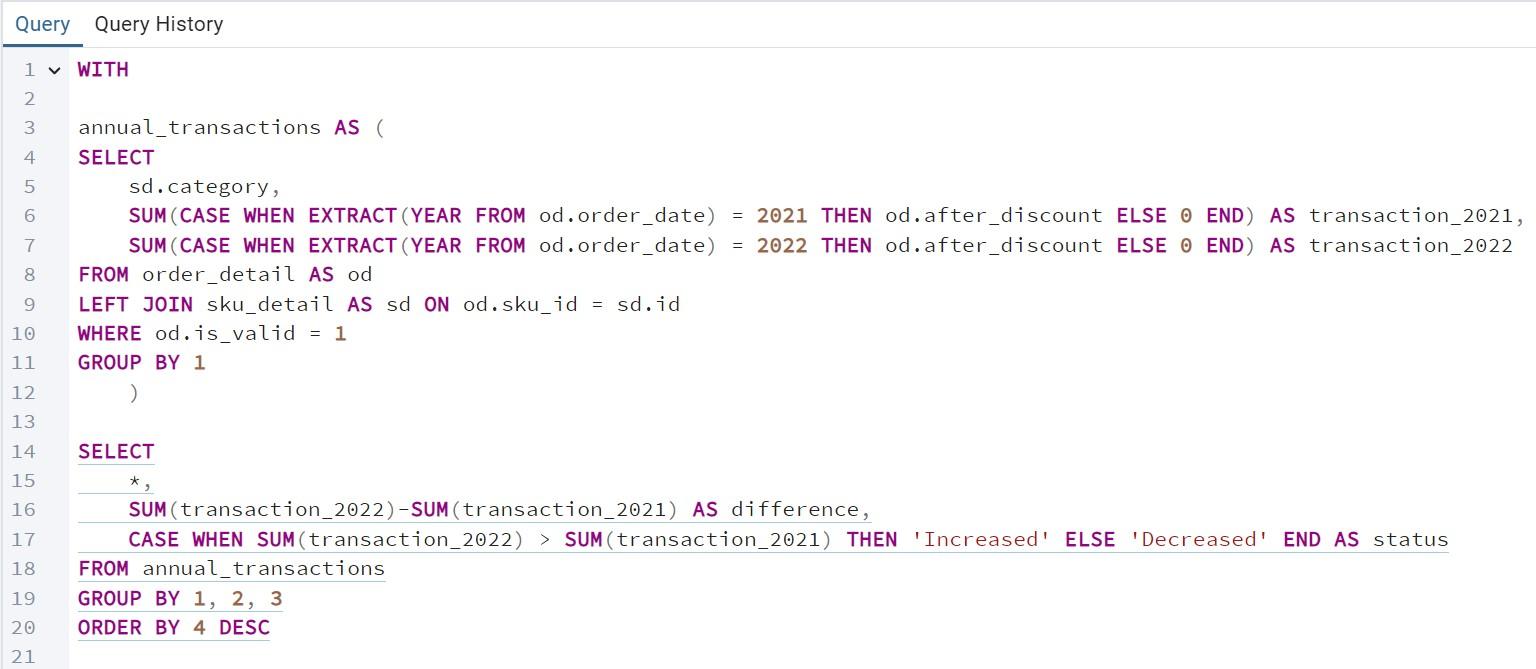
Pertanyaan tersebut meminta untuk dibuatkan table data yang berisi nama kategori yang memiliki transaksi dengan harga setelah diskon (after\_discount) yang telah dibayarkan (is\_valid = 1) pada 2021 dan 2022, serta terdapat data yang menunjukan selisih atau progres pada transaksi 2021 ke 2022 beserta statusnya.

Dengan demikian, pada table tersebut harus ada 5 kolom: kolom 1 berisi nama bulan, kolom 2 berisi data total nilai transaksi masing-masing kategori pada 2021, kolom 3 berisi data total nilai transaksi masing-masing kategori pada 2021, kolom 4 berisi data selisih transaksi 2021 dengan 2022, serta kolom 5 berisi status dari selisih tersebut.

Table tersebut tidak memiliki batas jumlah baris karena harus menampilkan semua nama kategori. Dari pertanyaan tersebut juga dapat diketahui bahwa sumber data yg diperlukan ialah gabungan table order\_detail dengan table sku\_detail karena hasil data tersebut memerlukan data di kolom kategori dari table sku\_detail serta data di kolom after\_discount dari table order\_detail.

Untuk menampilkan data tersebut juga dapat menggunakan metode CTE untuk mengambil data kategori dan data nilai transaksi (after\_discount) dari masing-masing kategori pada 2021 dan 2022. Setelah itu dengan data tersebut data selisih nilai transaksi serta data status selisih dapat diolah dan ditampilkan.

Q3 Query:



Q3 Query penjelasan:

Common Table Expression (CTE):

* **WITH annual\_transactions AS (….)** digunakan untuk menyederhanakan query yang kompleks dan mempermudah saat ingin menggunakan kembali query yang terdapat dalam CTE

Fungsi SELECT dan Agregat:

* **SELECT sd.category** untuk menampilkan nama kategori dari kolom category di table sku\_detail
* **SUM(CASE WHEN EXTRACT(YEAR FROM od.order\_date) = 2021 THEN od.after\_discount ELSE 0 END) AS transaction\_2021** untuk menampilkan nilai transaksi di kolom after\_discount dari table order\_detail yang terjadi di tahun 2021 dan menjumlahkan nilai di dalamnya (data berbentuk **integer**) berdasarkan kategori di kolom category lalu mengaliaskan nama kolom menjadi transaction\_2021
* **SUM(CASE WHEN EXTRACT(YEAR FROM od.order\_date) = 2021 THEN od.after\_discount ELSE 0 END) AS transaction\_2021** untuk menampilkan nilai transaksi di kolom after\_discount dari table order\_detail yang terjadi di tahun 2022 dan menjumlahkan nilai di dalamnya (data berbentuk **integer**) berdasarkan kategori di kolom category lalu mengaliaskan nama kolom menjadi transaction\_2022

Sumber Data:

* **FROM order\_detail AS od** untuk menjadikan table order\_detail sebagai sumber data dengan alias od
* **LEFT JOIN sku\_detail AS sd ON od.sku\_id = sd.id** menggabungkan table sku\_detail (alias sd) dengan order\_detail dengan primary key yaitu kolom id dari table sku\_detail dan sku\_id dari kolom order\_detail

Kalusa WHERE:

* **WHERE is\_valid = 1** untuk membatasi hasil berdasarkan kondisi 1 (is\_valid = 1 atau transaksi sudah dibayarkan)

Pernyataan GROUP BY

* **GROUP BY 1** mengelompokkan hasil query berdasarkan kolom 1 yaitu category

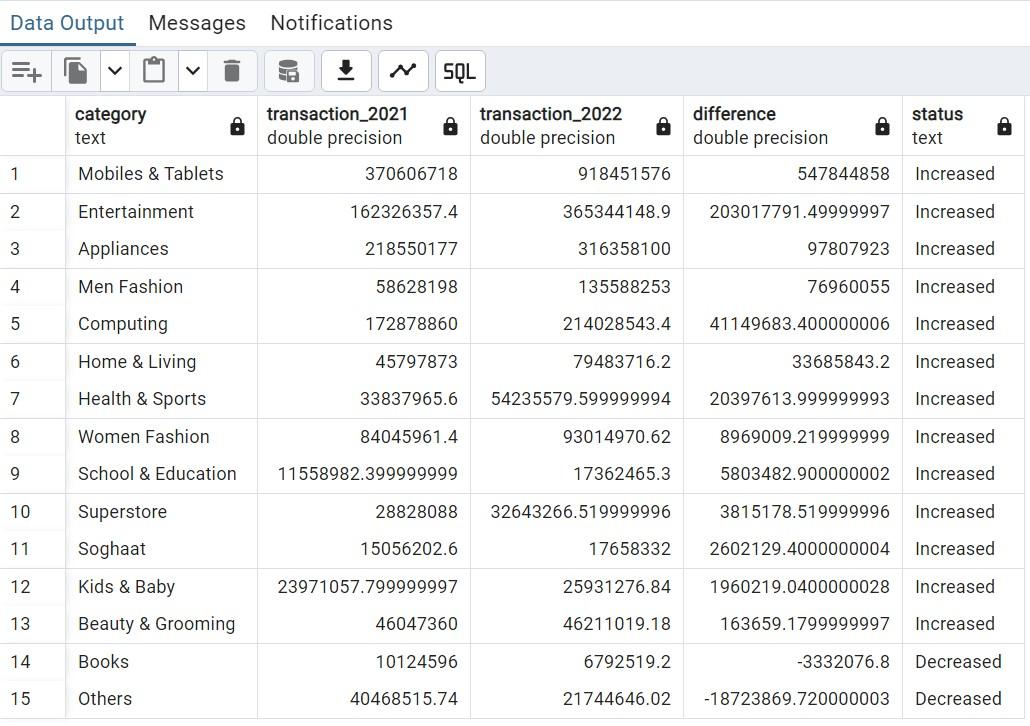
Perintah ORDER BY

* **ORDER BY 2 DESC** untuk mengurutkan data berdasarkan kolom 2 yaitu Transaction\_Sum dan dari nilai terbesar ke nilai terkecil

Query 2:

* **SELECT \*** untuk menampilkan kembali hasil dari CTE annual\_transactions
* **SUM(transaction\_2022)-SUM(transaction\_2021) AS difference** untuk menampilkan hasil penghitungan dari transaction\_2022 dikurangi transaction\_2021 dan memberi alias difference untuk kolom tersebut
* **CASE WHEN SUM(transaction\_2022) > SUM(transaction\_2021) THEN 'Increased' ELSE 'Decreased' END AS status** menampilkan kolom dengan alias status untuk menunjukan status dari hasil penghitungan di difference apakah ‘Increased’ atau ‘Decreased’
* **FROM annual\_transactions GROUP BY 1, 2, 3 ORDER BY 4 DESC** menjadikan CTE annual\_transactions sebagai sumber data, mengelompokkan table berdasarkan kolom category, transaction\_2021, dan transaction\_2022, serta mengurutkan datanya berdasarkan kolom difference dari data terbesar ke data terkecil

Q3 Query result:



Q3 Insight:

Dari data yang ditampilkan, ada 13 kategori yang mengalami peningkatan penjualan di tahun 2022 dibanding tahun 2021 yaitu *Mobiles & Tablets, Entertainment, Appliances, Men Fashion, Computing, Home & Living, Health & Sports, Women Fashion, School & Education, Superstore, Soghaat, Kids & Baby,* serta *Beauty & Grooming*. Sementara itu pada tahun 2022 hanya ada 2 kategori yang mengalami penurunan penjualan di tahun 2022 dibanding tahun 2021 yaitu *Books* dan *Others*.

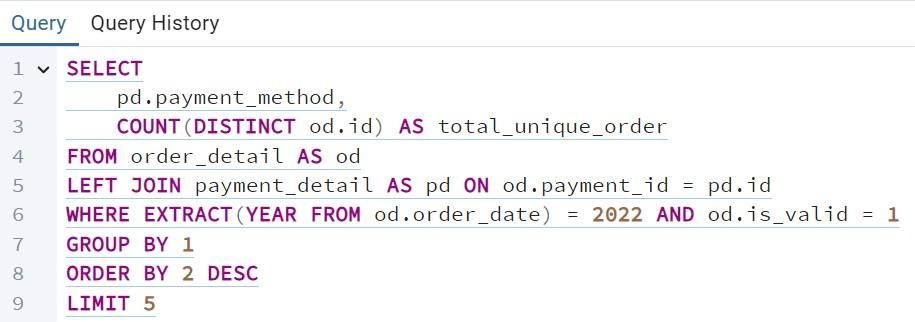
Nomor 4

Q: Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order). Gunakan **is\_valid = 1** untuk memfilter data transaksi. Source table: order\_detail, payment\_method

Q4 Logical Thinking:

Pertanyaan tersebut meminta untuk dibuatkan table data yang berisi 5 nama metode pembayaran (payment\_method) pada 2022 yang memiliki total jumlah unique\_order (order\_id) paling banyak dengan nilai transaksi yang telah dibayarkan (is\_valid = 1). Dengan demikian, pada table tersebut harus ada 2 kolom: kolom 1 berisi nama metode pembayaran dan kolom 2 berisi data total jumlah unique\_order yang telah membayar transaksi tersebut. Table tersebut juga dibatasi hanya 5 baris data saja karena meminta 5 jumlah paling besar. Dari pertanyaan tersebut juga dapat diketahui bahwa sumber data yg diperlukan ialah gabungan table order\_detail dengan table payment\_detail karena hasil data tersebut memerlukan data di kolom payment\_method dari table sku\_detail serta data di kolom order\_id dari table order\_detail.

Q4 Query:



Q4 Query penjelasan:

Fungsi SELECT dan Agregat:

* **SELECT pd.payment\_method** untuk menampilkan nama metode pembayaran dari kolom payment\_method table payment\_detail
* **COUNT(DISTINCT od.id) AS total\_unique\_order** untuk memilih kolom id dari table order\_detail dan menampilkan jumlah id masing-masing transaksi (wajib bersifat unik sehingga tidak ada duplikasi) berdasarkan nama metode pembayaran di kolom payment\_method lalu mengaliaskan nama kolom menjadi total\_unique\_order

Sumber Data:

* **FROM order\_detail AS od** untuk menjadikan table order\_detail sebagai sumber data dengan alias od
* **LEFT JOIN payment\_detail AS pd ON od.payment\_id = pd.id** menggabungkan table payment\_detail (alias pd) dengan order\_detail dengan primary key yaitu kolom id dari table payment\_detail dan payment\_id dari kolom order\_detail

Kalusa WHERE:

* **WHERE EXTRACT(YEAR FROM order\_date) = 2022 AND is\_valid = 1** untuk membatasi hasil berdasarkan kondisi 1 (hanya pada transaksi di tahun 2022) dan kondisi 2 (is\_valid = 1 atau transaksi sudah dibayarkan)

Pernyataan GROUP BY

* **GROUP BY 1** mengelompokkan hasil query berdasarkan kolom 1 yaitu payment\_method

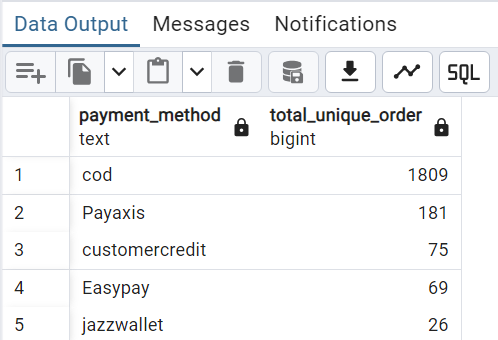
Perintah ORDER BY

* **ORDER BY 2 DESC** untuk mengurutkan data berdasarkan kolom 2 yaitu total\_unique\_order dan dari nilai terbesar ke nilai terkecil

Klausa LIMIT

* **LIMIT 5** untuk membatasi baris yang ditampilkan menjadi 5 baris

Q4 Query result:



Q4 Insight:

Dari data yang ditampilkan dapat diketahui bahwa *cod* menjadi metode pembayaran yang paling popular di tahun 2022 disusul oleh *Payaxis, customercredit, Easypay* dan *jazzwallet*. Jumlah total\_unique\_order *cod*, yaitu 1809, juga jauh lebih besar dibanding gabungan 4 metode pembayaran berikutnya.

Nomor 5

Q: Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.

1. Samsung

2. Apple

3. Sony

4. Huawei

5. Lenovo

Gunakan **is\_valid = 1** untuk memfilter data transaksi.

Source table: order\_detail, sku\_detail

Q3 Logical thinking:

Pertanyaan tersebut meminta untuk dibuatkan table data yang berisi 5 nama produk (sku\_name) yaitu Samsung, Apple, Sony, Huawei, dan Lenovo yang memiliki transaksi dengan harga setelah diskon (after\_discount) yang telah dibayarkan (is\_valid = 1), serta mengurutkan 5 nama produk tersebut berdasarkan data nilai transaksi terbesar ke terkecil.

Dengan demikian, pada table tersebut harus ada 2 kolom: kolom 1 berisi nama produk, kolom 2 berisi data total nilai transaksi masing-masing nama produk tersebut.

Table tersebut tidak memiliki batas jumlah baris karena harus menampilkan semua nama produk yang telah diminta. Dari pertanyaan tersebut juga dapat diketahui bahwa sumber data yg diperlukan ialah gabungan table order\_detail dengan table sku\_detail karena hasil data tersebut memerlukan data di kolom sku\_name dari table sku\_detail serta data di kolom after\_discount dari table order\_detail.

Untuk menampilkan data tersebut juga dapat menggunakan metode CTE untuk mengambil data sku\_name dan data nilai transaksi (after\_discount) dari masing-masing sku\_name. Setelah itu data tersebut data ditampilkan kembali tanpa data yang bersifat null.

Q5 Query:



Q5 Query penjelasan:

Common Table Expression (CTE):

* **WITH top\_brands AS (….)** digunakan untuk menyederhanakan query yang kompleks dan mempermudah saat ingin menggunakan kembali query yang terdapat dalam CTE

Fungsi SELECT dan Agregat:

* **SELECT CASE WHEN LOWER(sd.sku\_name) LIKE '%samsung%' THEN 'Samsung' . . . . OR LOWER(sd.sku\_name) LIKE '%zuk%' THEN 'Lenovo' END AS product\_names** untuk menyeleksi data yang mengandung nama produk dari brand ‘Samsung’, ‘Lenovo’, ‘Sony’, ‘Apple’, dan ‘Huawei’ dari kolom sku\_name di table sku\_detail dan memberi alias product\_names pada kolom tersebut
* **SUM(od.after\_discount) AS transaction\_sum** untuk memilih kolom after\_discount dari table order\_detail dan menjumlahkan nilai di dalamnya (data berbentuk **integer**) berdasarkan kategoti di kolom category lalu mengaliaskan nama kolom menjadi transaction\_sum

Sumber Data:

* **FROM order\_detail AS od** untuk menjadikan table order\_detail sebagai sumber data dengan alias od
* **LEFT JOIN sku\_detail AS sd ON od.sku\_id = sd.id** menggabungkan table sku\_detail (alias sd) dengan order\_detail dengan primary key yaitu kolom id dari table sku\_detail dan sku\_id dari kolom order\_detail

Kalusa WHERE:

* **WHERE is\_valid = 1** untuk membatasi hasil berdasarkan kondisi 1 (is\_valid = 1 atau transaksi sudah dibayarkan)

Pernyataan GROUP BY

* **GROUP BY 1** mengelompokkan hasil query berdasarkan kolom 1 yaitu product\_names

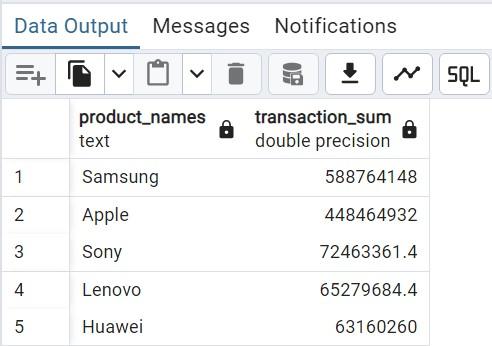
Perintah ORDER BY

* **ORDER BY 2 DESC** untuk mengurutkan data berdasarkan kolom 2 yaitu transaction\_sum dan dari nilai terbesar ke nilai terkecil

Query 2:

* **SELECT \*** untuk menampilkan kembali hasil dari CTE annual\_transactions
* **FROM top\_brands** menjadikan CTE top\_brands sebagai sumber data
* **WHERE product\_names IS NOT NULL** untuk menampilkan hany data yang tidak null saja

Q5 Query result:



Q5 Insight:

Dari data yang ditampilkan, **Samsung** menjadi produk dengan nilai transaksi terbesar yaitu **588**.**764**.**148** disusul **Apple** dengan **448**.**464**.**932**, kemudian **Sony**  dengan **72.463.361,4**, lalu **Lenovo** dengan **65.279.684,4**, dan terakhir **Huawei** dengan **63.160.260**