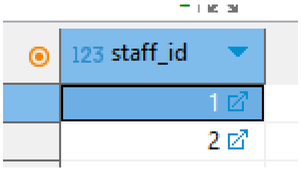
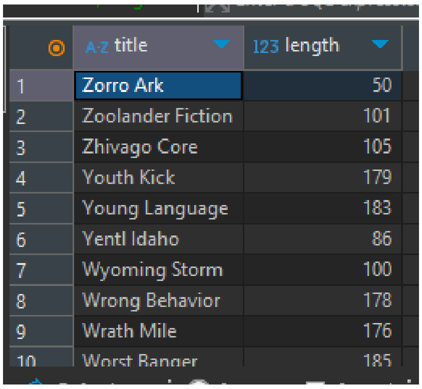
DVD RENTAL

```
--select all table payment

select*from payment p ;

--select distinct, as:mengambil nilai unik dari satu kolom/sekelompok tabel. nanti hasilnya disini akan muncul angka 1 dan 2, karna hanya 2 angka ini saja yg unik.

select distinct staff_id from payment p ;
```



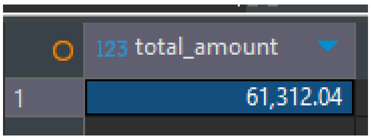
	staff_id
	1
	2

```
--order by

select title, length

from film

order by title desc;
```



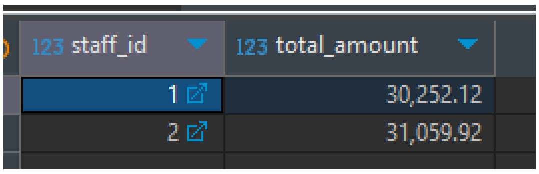
	title	length
1	Zorro Ark	50
2	Zoolander Fiction	101
3	Zhivago Core	105
4	Youth Kick	179
5	Young Language	183
6	Yentl Idaho	86
7	Wyoming Storm	100
8	Wrong Behavior	178
9	Wrath Mile	176
10	Worst Banger	185

```
--Aggregation:AS, MENGHITUNG MIN MAX SUM AVG COUNT. ERAT KAITANNYA DGN GRUP BY

select sum(amount) as total_amount

from payment;
```



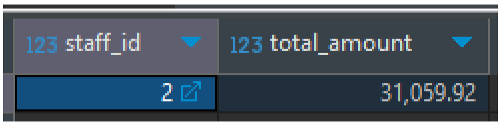
	total_amount
1	61,312.04

```
--Group By & Aggregation: mau melihat berapa total amount yang dihasilkan setiap staff id

select staff_id, sum(amount) as total_amount

from payment p

group by staff_id;
```



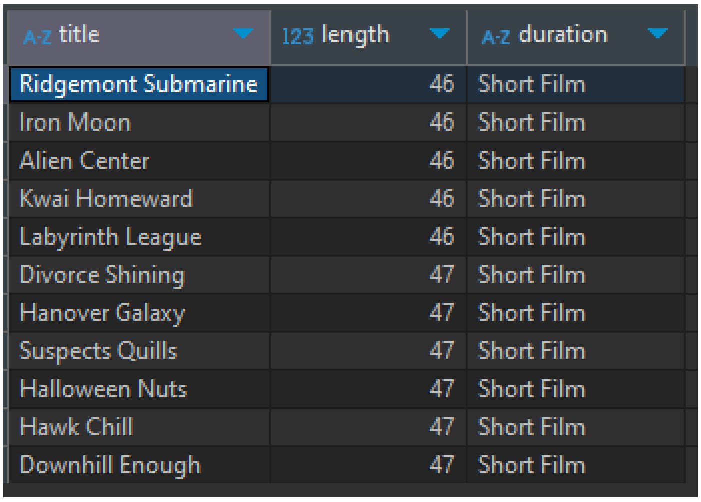
staff_id	total_amount
1	30,252.12
2	31,059.92

```
--Group By Having--as: kyk melakukan filtering. Siapakah staff ID yang total amountnya diatas 31k?

select staff_id, sum(amount) as total_amount

from payment p

group by staff_id

having sum(amount)>31000;
```



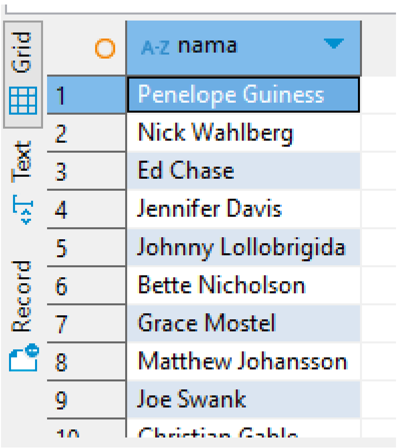
staff_id	total_amount
2	31,059.92

```
--case when: membuat kategori film berdasarkan panjang filmnya

select title, length,

case

	when length > 0 and length <=50 then 'Short Film'

	when length > 50 and length <=120 then 'Medium Film'

	when length > 120 then 'Long Film'

end duration

from film

order by length asc;
```



title	length	duration
Ridgemont Submarine	46	Short Film
Iron Moon	46	Short Film
Alien Center	46	Short Film
Kwai Homeward	46	Short Film
Labyrinth League	46	Short Film
Divorce Shining	47	Short Film
Hanover Galaxy	47	Short Film
Suspects Quills	47	Short Film
Halloween Nuts	47	Short Film
Hawk Chill	47	Short Film
Downhill Enough	47	Short Film

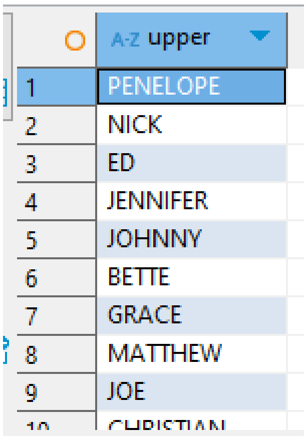
```
--concat
select concat(a.first_name,' ',a.last_name)as nama
from actor a ;
```



	nama
1	Penelope Guiness
2	Nick Wahlberg
3	Ed Chase
4	Jennifer Davis
5	Johnny Lollobrigida
6	Bette Nicholson
7	Grace Mostel
8	Matthew Johansson
9	Joe Swank
10	Christian Gable

as:menggabungkan 2 kolom menjadi 1

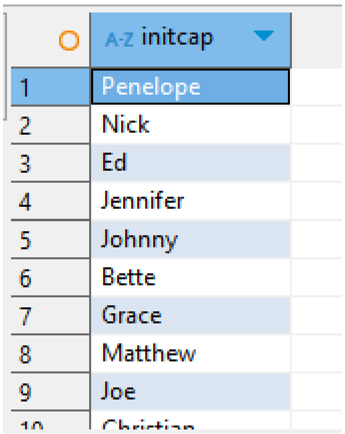
```
--upper
select upper(a.first_name)
from actor a ;
```



	upper
1	PENELOPE
2	NICK
3	ED
4	JENNIFER
5	JOHNNY
6	BETTE
7	GRACE
8	MATTHEW
9	JOE
10	CHRISTIAN

as: memperbesar huruf

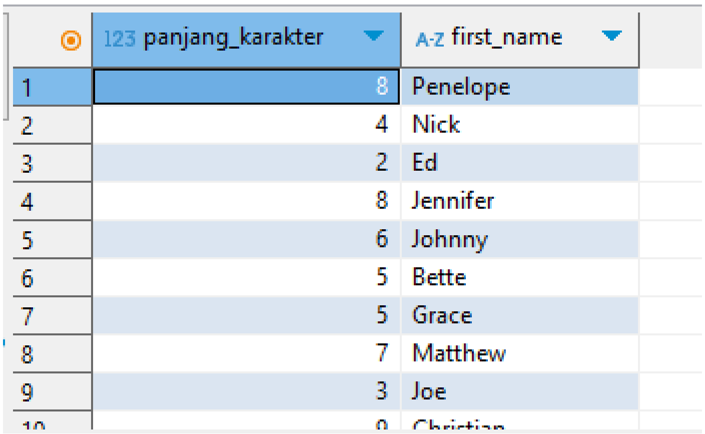
```
--initcap
select initcap('fahira nurul ichza');--Berubah jadi Fahira.
select initcap(a.first_name)
from actor a ;
```



	initcap
1	Penelope
2	Nick
3	Ed
4	Jennifer
5	Johnny
6	Bette
7	Grace
8	Matthew
9	Joe
10	Christian

as: membuat huruf kapital di awal

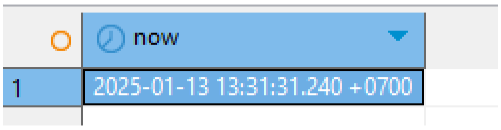
```
--length

select length(a.first_name) as panjang_karakter, first_name

from actor a ;
```



	panjang_karakter	first_name
1	8	Penelope
2	4	Nick
3	2	Ed
4	8	Jennifer
5	6	Johnny
6	5	Bette
7	5	Grace
8	7	Matthew
9	3	Joe
10	9	Christian

as: menghitung jumlah huruf dari kolom first name

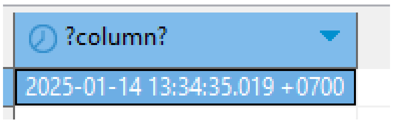
```
--timestamp,current time/now (update data, update data penjualan)

select now();

--atau

select current_timestamp;
```



	now
1	2025-01-13 13:31:31.240 +0700

as: menunjukkan waktu sekarang

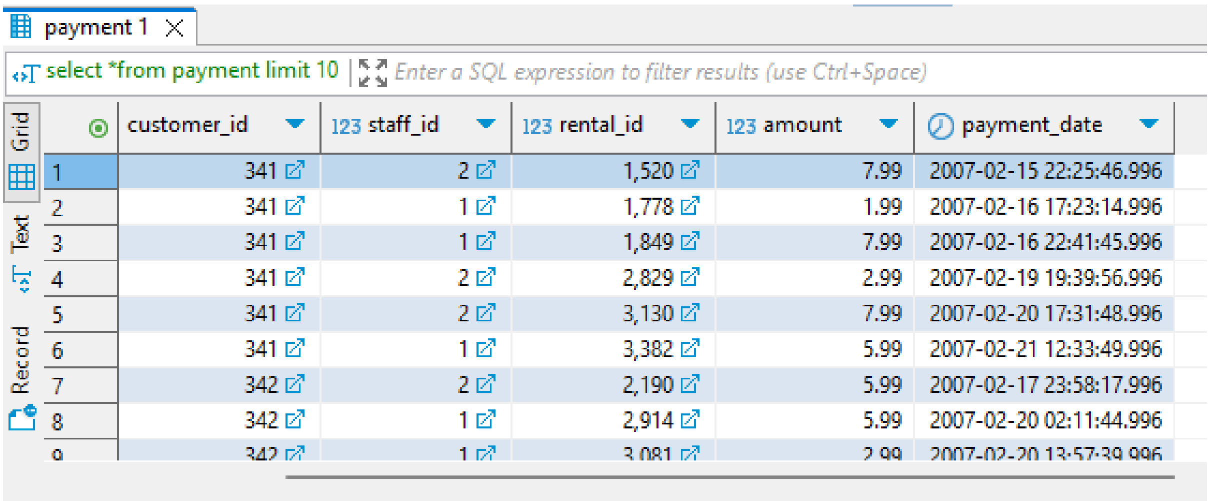
```
--add (menambah) and sub (mengurangi)

select current_timestamp + interval '1 day';
```



?column?
2025-01-14 13:34:35.019 +0700

as: mengganti tanggal nya jadi tambah 1 hari. sekarang 13 jan.

```
select *from payment limit 10;
```

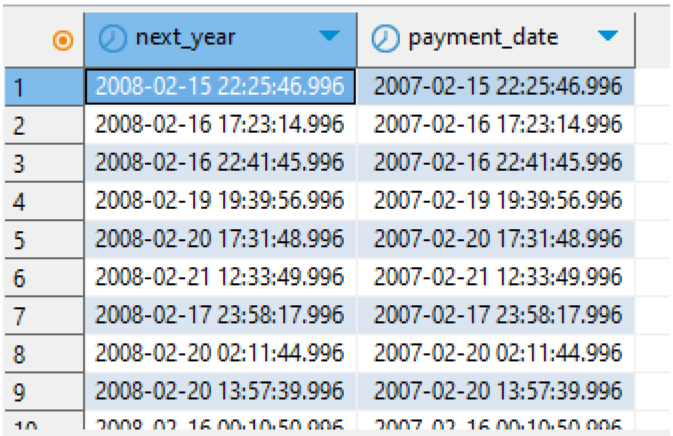


payment 1
select *from payment limit 10
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record

	customer_id	staff_id	rental_id	amount	payment_date
1	341	2	1,520	7.99	2007-02-15 22:25:46.996
2	341	1	1,778	1.99	2007-02-16 17:23:14.996
3	341	1	1,849	7.99	2007-02-16 22:41:45.996
4	341	2	2,829	2.99	2007-02-19 19:39:56.996
5	341	2	3,130	7.99	2007-02-20 17:31:48.996
6	341	1	3,382	5.99	2007-02-21 12:33:49.996
7	342	2	2,190	5.99	2007-02-17 23:58:17.996
8	342	1	2,914	5.99	2007-02-20 02:11:44.996
9	342	1	3,081	2.99	2007-02-20 13:57:39.996

as: menunjukkan kolom payment 10 baris teratas, ternyata ada kolom payment date.

```
select payment_date + interval '1 year' as next_year, payment_date

from payment p

limit 10;
```

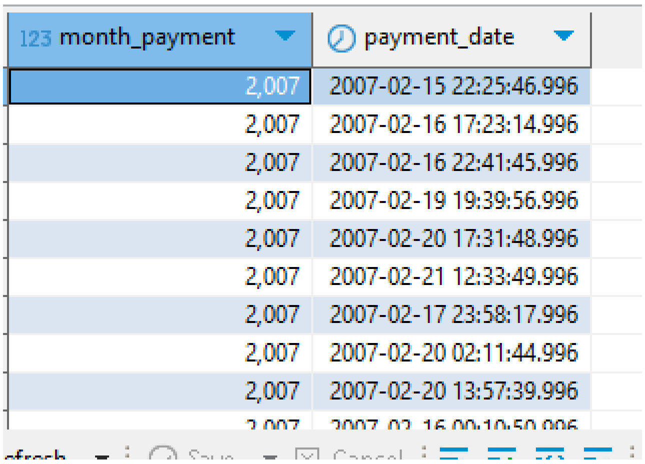


	next_year	payment_date
1	2008-02-15 22:25:46.996	2007-02-15 22:25:46.996
2	2008-02-16 17:23:14.996	2007-02-16 17:23:14.996
3	2008-02-16 22:41:45.996	2007-02-16 22:41:45.996
4	2008-02-19 19:39:56.996	2007-02-19 19:39:56.996
5	2008-02-20 17:31:48.996	2007-02-20 17:31:48.996
6	2008-02-21 12:33:49.996	2007-02-21 12:33:49.996
7	2008-02-17 23:58:17.996	2007-02-17 23:58:17.996
8	2008-02-20 02:11:44.996	2007-02-20 02:11:44.996
9	2008-02-20 13:57:39.996	2007-02-20 13:57:39.996
10	2008-02-16 00:10:50.996	2007-02-16 00:10:50.996

as: membuat kolom baru bernama next year yang isinya + 1 tahun dari payment date di table payment.

```
--extract timestamp
```

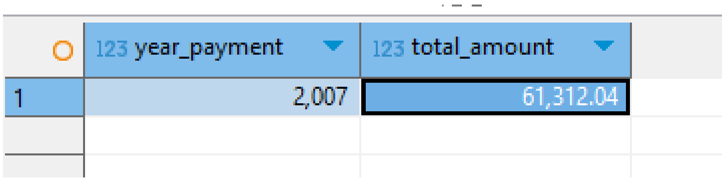
```sql
select extract('year' from payment_date ) as month_payment, payment_date
from payment p ;
```



month_payment	payment_date
2,007	2007-02-15 22:25:46.996
2,007	2007-02-16 17:23:14.996
2,007	2007-02-16 22:41:45.996
2,007	2007-02-19 19:39:56.996
2,007	2007-02-20 17:31:48.996
2,007	2007-02-21 12:33:49.996
2,007	2007-02-17 23:58:17.996
2,007	2007-02-20 02:11:44.996
2,007	2007-02-20 13:57:39.996
2,007	2007-02-16 09:10:50.996

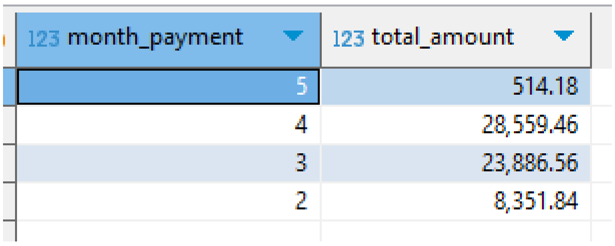
as:mengambil tahun dari kolom payment date. bisa juga diganti ganti misalnya month, hour, date, second

```sql
--extract time: berapa total amount dari setiap tahunnya?
select extract ('year' from payment_date) as year_payment, sum(amount) as total_amount
from payment p
group by year_payment;
```



	year_payment	total_amount
1	2,007	61,312.04

```sql
--atau
select extract ('month' from payment_date) as month_payment, sum(amount) as total_amount
from payment p
group by month_payment
order by month_payment desc;
```

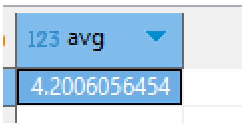


month_payment	total_amount
5	514.18
4	28,559.46
3	23,886.56
2	8,351.84

as: sama dengan diatas

--SUB QUERY

select avg(amount) from payment p ;



avg
4.2006056454

as: menampilkan rata-rata dari kolom amount. jadi ad 4,2 rata-rata totalnya.

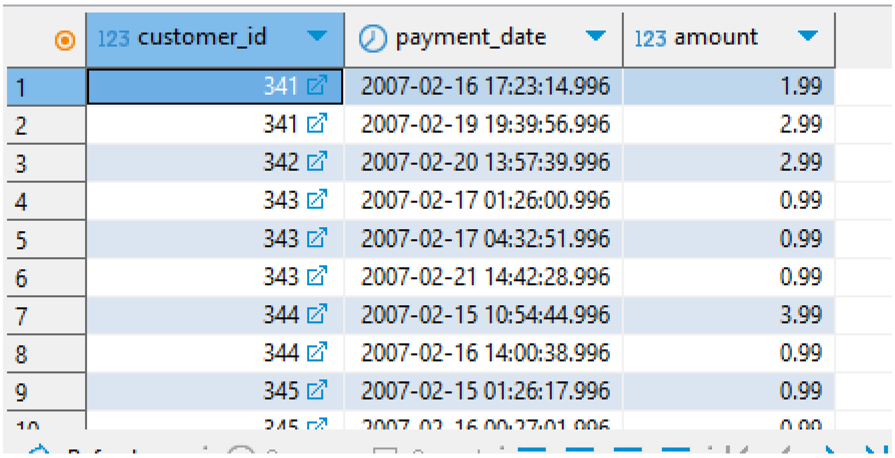
select customer_id

from payment p

where amount <(select avg (amount)

from payment)

and extract (year from payment_date)=2007;



	customer_id	payment_date	amount
1	341	2007-02-16 17:23:14.996	1.99
2	341	2007-02-19 19:39:56.996	2.99
3	342	2007-02-20 13:57:39.996	2.99
4	343	2007-02-17 01:26:00.996	0.99
5	343	2007-02-17 04:32:51.996	0.99
6	343	2007-02-21 14:42:28.996	0.99
7	344	2007-02-15 10:54:44.996	3.99
8	344	2007-02-16 14:00:38.996	0.99
9	345	2007-02-15 01:26:17.996	0.99
10	345	2007-02-16 00:27:01.996	0.99

as: menampilkan kolom **customer id** dari tabel payment dimana kondisi nilai amount (jumlah pembayaran) lebih kecil dari **rata-rata jumlah pembayaran** di seluruh tabel payment. dan dipilih hanya tahun yang payment datenya 2007.

-- Row Number Partition By : Memberikan angka transaksi untuk setiap customer

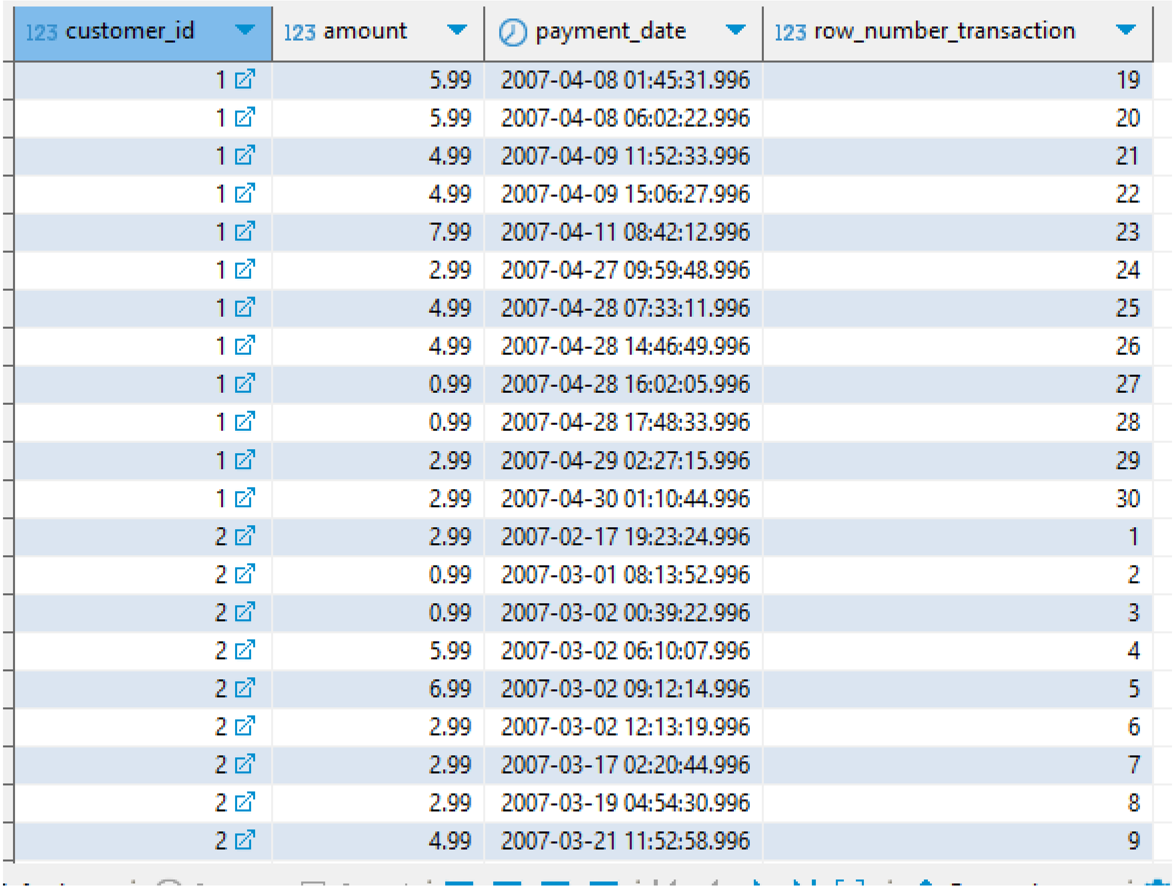
select customer_id,

amount,

payment_date,

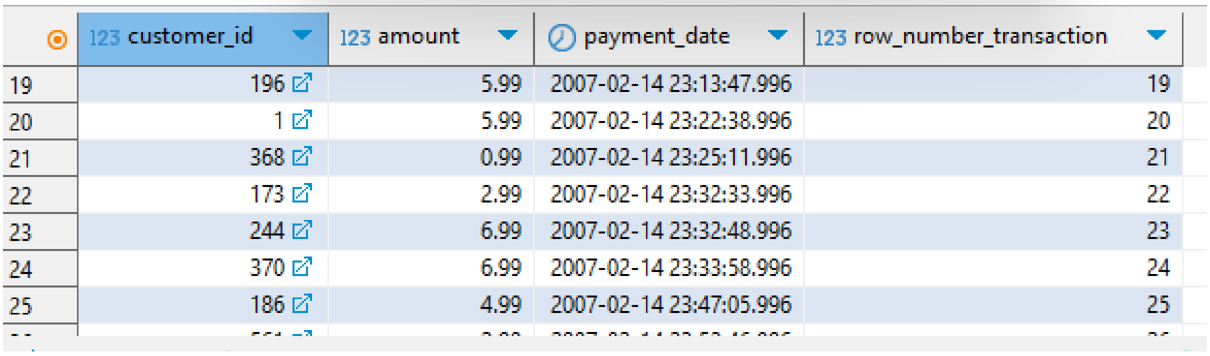
row_number() **over** (**partition by** customer_id **order by** payment_date) **as** row_number_transaction

from payment p;



customer_id	amount	payment_date	row_number_transaction
1	5.99	2007-04-08 01:45:31.996	19
1	5.99	2007-04-08 06:02:22.996	20
1	4.99	2007-04-09 11:52:33.996	21
1	4.99	2007-04-09 15:06:27.996	22
1	7.99	2007-04-11 08:42:12.996	23
1	2.99	2007-04-27 09:59:48.996	24
1	4.99	2007-04-28 07:33:11.996	25
1	4.99	2007-04-28 14:46:49.996	26
1	0.99	2007-04-28 16:02:05.996	27
1	0.99	2007-04-28 17:48:33.996	28
1	2.99	2007-04-29 02:27:15.996	29
1	2.99	2007-04-30 01:10:44.996	30
2	2.99	2007-02-17 19:23:24.996	1
2	0.99	2007-03-01 08:13:52.996	2
2	0.99	2007-03-02 00:39:22.996	3
2	5.99	2007-03-02 06:10:07.996	4
2	6.99	2007-03-02 09:12:14.996	5
2	2.99	2007-03-02 12:13:19.996	6
2	2.99	2007-03-17 02:20:44.996	7
2	2.99	2007-03-19 04:54:30.996	8
2	4.99	2007-03-21 11:52:58.996	9

```
-- Row Number : Hanya memberikan angka urutan berdasarkan payment_date terlama -> terbaru
select customer_id,
                amount,
                payment_date,
                row_number() over (order by payment_date) as row_number_transaction
from payment p;
```

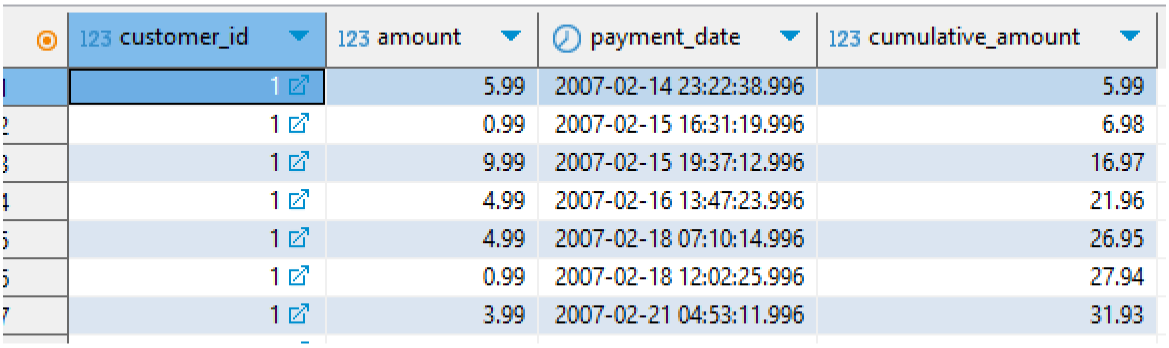


	customer_id	amount	payment_date	row_number_transaction
19	196	5.99	2007-02-14 23:13:47.996	19
20	1	5.99	2007-02-14 23:22:38.996	20
21	368	0.99	2007-02-14 23:25:11.996	21
22	173	2.99	2007-02-14 23:32:33.996	22
23	244	6.99	2007-02-14 23:32:48.996	23
24	370	6.99	2007-02-14 23:33:58.996	24
25	186	4.99	2007-02-14 23:47:05.996	25

as: bedanya dengan diatas, ini tidak ada partision by customer id nya. jadi hanya mengurutkan by payment date aja. customer id nya acak aj.

-- Sum() di window function : Menghitung Cumulative dari setiap customer

```
select customer_id,
            amount,
            payment_date,
            sum(amount) over (partition by customer_id order by payment_date) as
cumulative_amount
from payment p ;
```

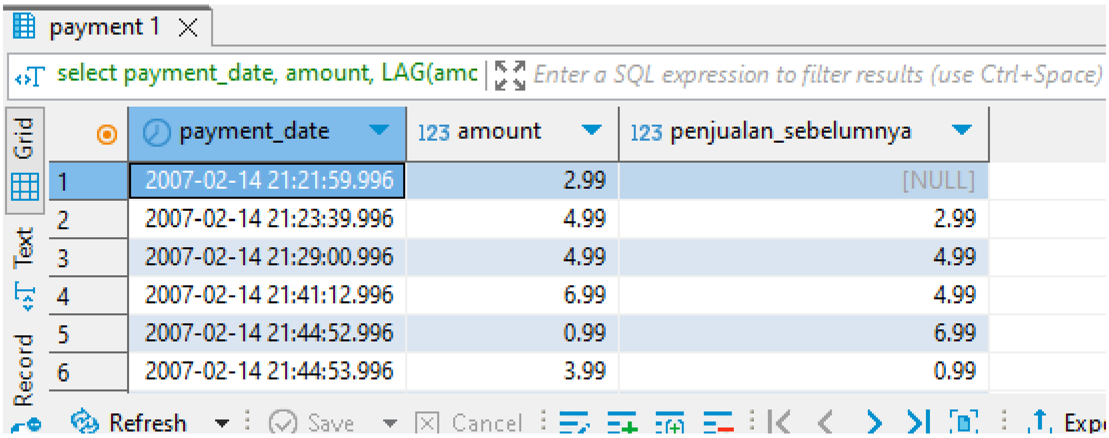


	customer_id	amount	payment_date	cumulative_amount
1	1	5.99	2007-02-14 23:22:38.996	5.99
2	1	0.99	2007-02-15 16:31:19.996	6.98
3	1	9.99	2007-02-15 19:37:12.996	16.97
4	1	4.99	2007-02-16 13:47:23.996	21.96
5	1	4.99	2007-02-18 07:10:14.996	26.95
6	1	0.99	2007-02-18 12:02:25.996	27.94
7	1	3.99	2007-02-21 04:53:11.996	31.93

as: ini membuat list per customer id dengan urutan terlama sampai terbaru serta menampilkan masing-masing jumlah kumulatif dari baris sebelumnya ke baris saat ini.

-- LAG() di Window Function : Mengambil data pada transaksi sebelumnya, untuk nantinya dilihat perbedaan(selisih) transaksi yang dilakukan

```
select payment_date,
            amount,
            lag(amount) over (order by payment_date) as penjualan_sebelumnya
from payment p;
```



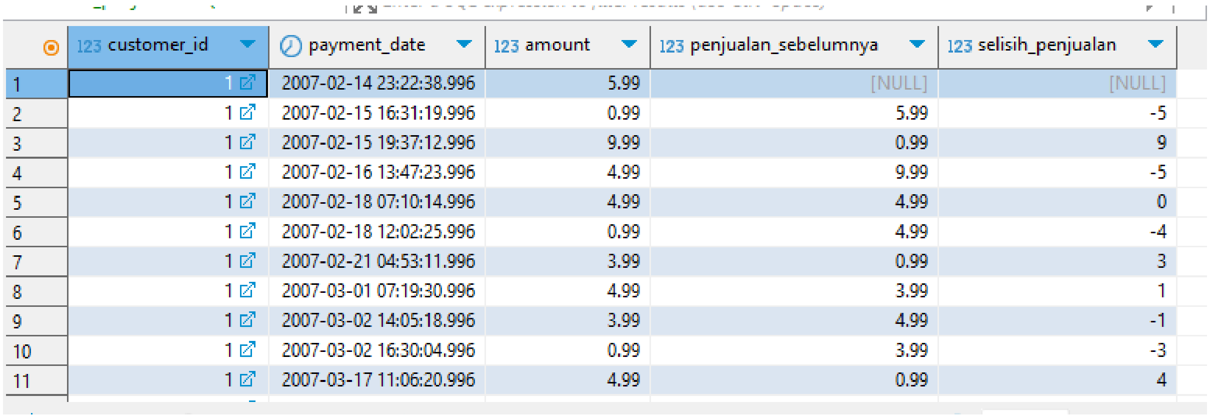
select payment_date, amount, LAG(amc

Enter a SQL expression to filter results (use Ctrl+Space)

	payment_date	amount	penjualan_sebelumnya
1	2007-02-14 21:21:59.996	2.99	[NULL]
2	2007-02-14 21:23:39.996	4.99	2.99
3	2007-02-14 21:29:00.996	4.99	4.99
4	2007-02-14 21:41:12.996	6.99	4.99
5	2007-02-14 21:44:52.996	0.99	6.99
6	2007-02-14 21:44:53.996	3.99	0.99

as: ini untuk membuat kolom penjualan sebelumnya yang isinya adalah ngka dri kolom amount dibarus sebelumnya. ini bsa berfungsi utk menilai selish amount saat ini dikurang jumlah sebelumnya. akan dijelaskan di query brktnya.

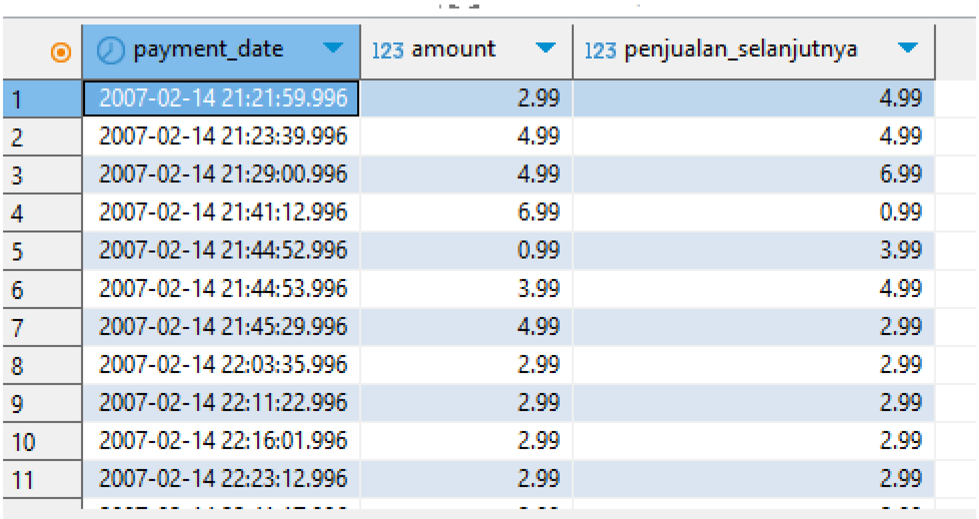
```
-- Menambahkan kolom untuk menghitung selisihnya

with table_penjualan as (select customer_id ,
                    payment_date,
                    amount,
                    lag(amount) over (partition by customer_id order by payment_date) as penjualan_sebelumnya
                    from payment p) -- CTE (a.k.a Table Bayangan)
select customer_id, payment_date, amount, penjualan_sebelumnya, (amount-penjualan_sebelumnya) as selisih_penjualan
from table_penjualan;
```



	customer_id	payment_date	amount	penjualan_sebelumnya	selisih_penjualan
1	1	2007-02-14 23:22:38.996	5.99	[NULL]	[NULL]
2	1	2007-02-15 16:31:19.996	0.99	5.99	-5
3	1	2007-02-15 19:37:12.996	9.99	0.99	9
4	1	2007-02-16 13:47:23.996	4.99	9.99	-5
5	1	2007-02-18 07:10:14.996	4.99	4.99	0
6	1	2007-02-18 12:02:25.996	0.99	4.99	-4
7	1	2007-02-21 04:53:11.996	3.99	0.99	3
8	1	2007-03-01 07:19:30.996	4.99	3.99	1
9	1	2007-03-02 14:05:18.996	3.99	4.99	-1
10	1	2007-03-02 16:30:04.996	0.99	3.99	-3
11	1	2007-03-17 11:06:20.996	4.99	0.99	4

```
-- LEAD() di Window Function : Mengambil data setelahnya
```

```
select payment_date, amount,
                lead(amount) over (order by payment_date) as penjualan_selanjutnya
from payment p;
```



	payment_date	amount	penjualan_selanjutnya
1	2007-02-14 21:21:59.996	2.99	4.99
2	2007-02-14 21:23:39.996	4.99	4.99
3	2007-02-14 21:29:00.996	4.99	6.99
4	2007-02-14 21:41:12.996	6.99	0.99
5	2007-02-14 21:44:52.996	0.99	3.99
6	2007-02-14 21:44:53.996	3.99	4.99
7	2007-02-14 21:45:29.996	4.99	2.99
8	2007-02-14 22:03:35.996	2.99	2.99
9	2007-02-14 22:11:22.996	2.99	2.99
10	2007-02-14 22:16:01.996	2.99	2.99
11	2007-02-14 22:23:12.996	2.99	2.99