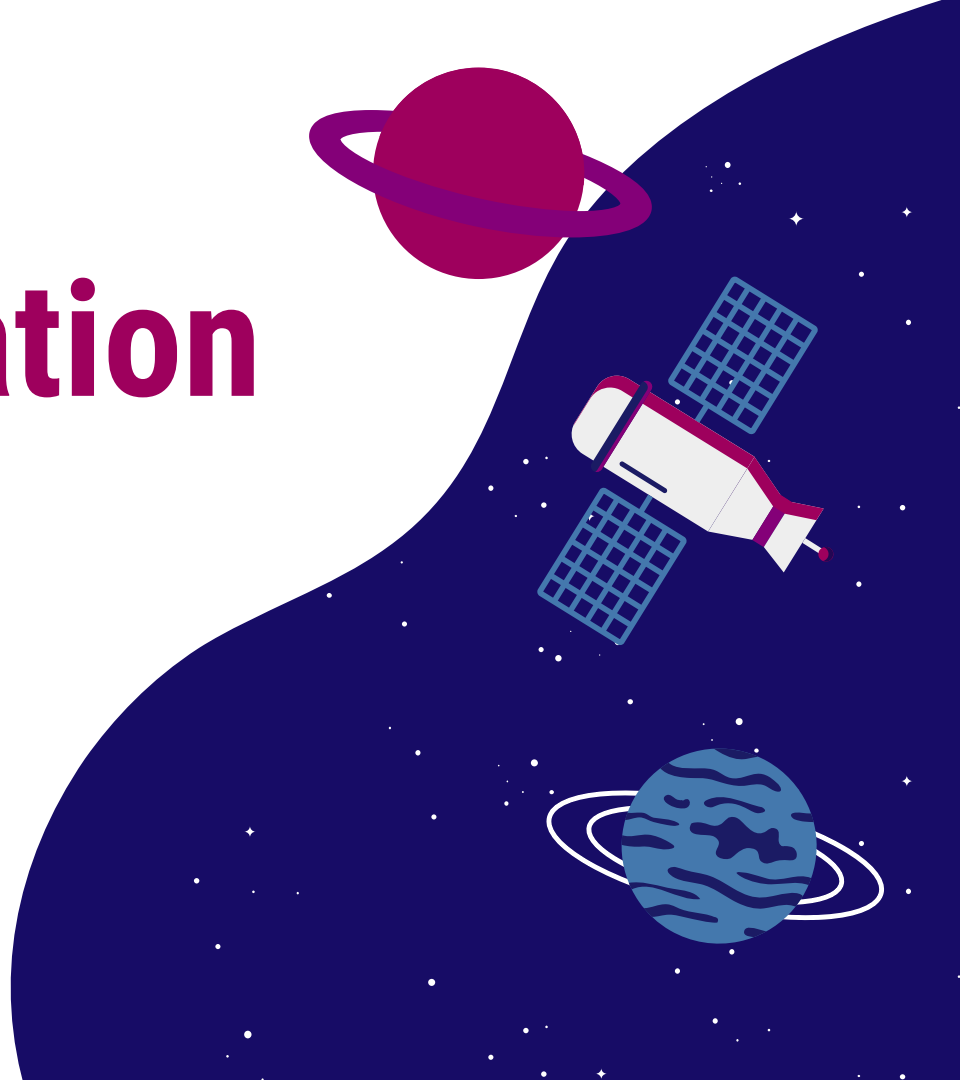


Data Manipulation Language 3

Praktikum Basis Data

Diyah Utami Kusumaning Putri, S.Kom., M.Sc., M.Cs.



Fungsi Agregat

suatu fungsi yang melakukan
perhitungan matematika
menggunakan beberapa nilai dan
mengembalikan satu nilai

Fungsi Agregat

01

MAX

02

MIN

03

AVG

04

SUM

05

COUNT

MAX

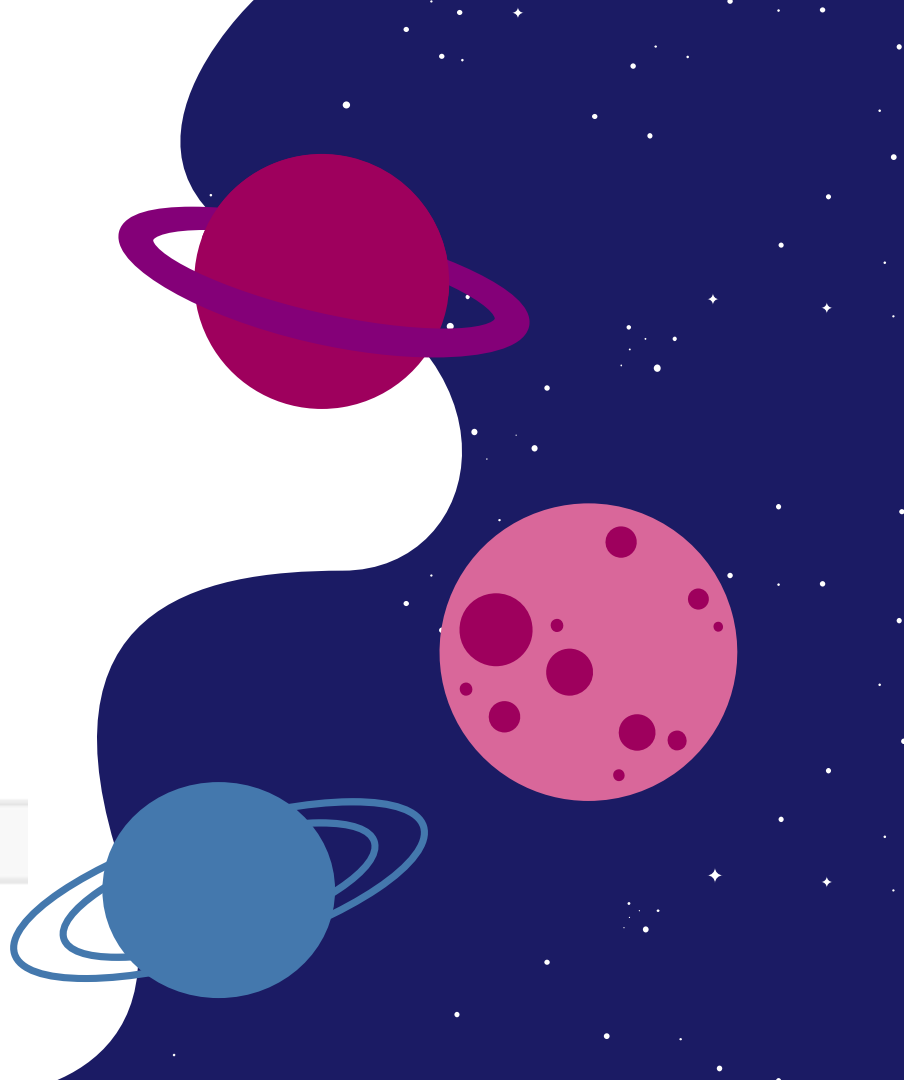
menghasilkan nilai maksimum dari sekumpulan nilai

Contoh

- ❑ menemukan angka terbesar
- ❑ produk yang paling mahal
- ❑ jumlah pembayaran terbesar dari pelanggan

Sintaks

```
1 | MAX(DISTINCT expression)
```



Cuplikan tabel PAYMENTS

customerNumber	checkNumber	paymentDate	amount ▾ 1
141	JE105477	2005-03-18	120166.58
141	ID10962	2004-12-31	116208.40
124	KI131716	2003-08-15	111654.40
148	KM172879	2003-12-26	105743.00
124	AE215433	2005-03-05	101244.59
321	DJ15149	2003-11-03	85559.12
124	BG255406	2004-08-28	85410.87
167	GN228846	2003-12-03	85024.46
124	ET64396	2005-04-16	83598.04
114	MA765515	2004-12-15	82261.22

Query

```
1 SELECT
2     MAX(amount)
3 FROM
4     payments;
```

MIN

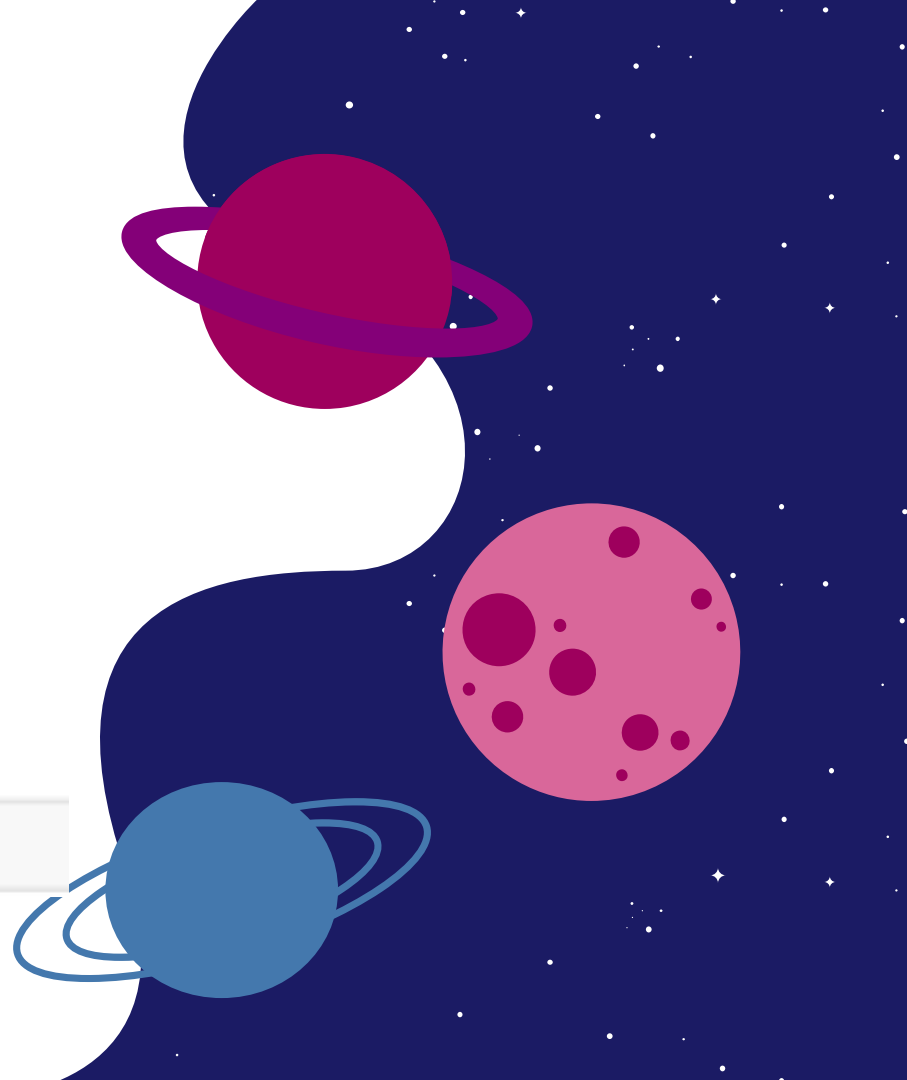
menghasilkan nilai minimum dari sekumpulan nilai

Contoh

- ❑ menemukan nilai terkecil
- ❑ memilih produk yang termurah
- ❑ memperoleh *credit limit* terendah

Sintaks

```
1 MIN(DISTINCT expression);
```



Cuplikan tabel PRODUCTS

productCode	productName	quantityInStock	buyPrice	MSRP
S24_2840	1958 Chevy Corvette Limited Edition	2542	15.91	35.36
S24_2972	1982 Lamborghini Diablo	7723	16.24	37.76
S24_2022	1938 Cadillac V-16 Presidential Limousine	2847	20.61	44.80
S24_3969	1936 Mercedes Benz 500k Roadster	2081	21.75	41.03
S24_1937	1939 Chevrolet Deluxe Coupe	7332	22.57	33.19
S18_4668	1939 Cadillac Limousine	6645	23.14	50.31
S32_2206	1982 Ducati 996 R	9241	24.14	40.23
S18_2625	1936 Harley Davidson El Knucklehead	4357	24.23	60.57
S18_1367	1936 Mercedes-Benz 500K Special Roadster	8635	24.26	53.91
S18_2432	1926 Ford Fire Engine	2018	24.92	60.77

Query

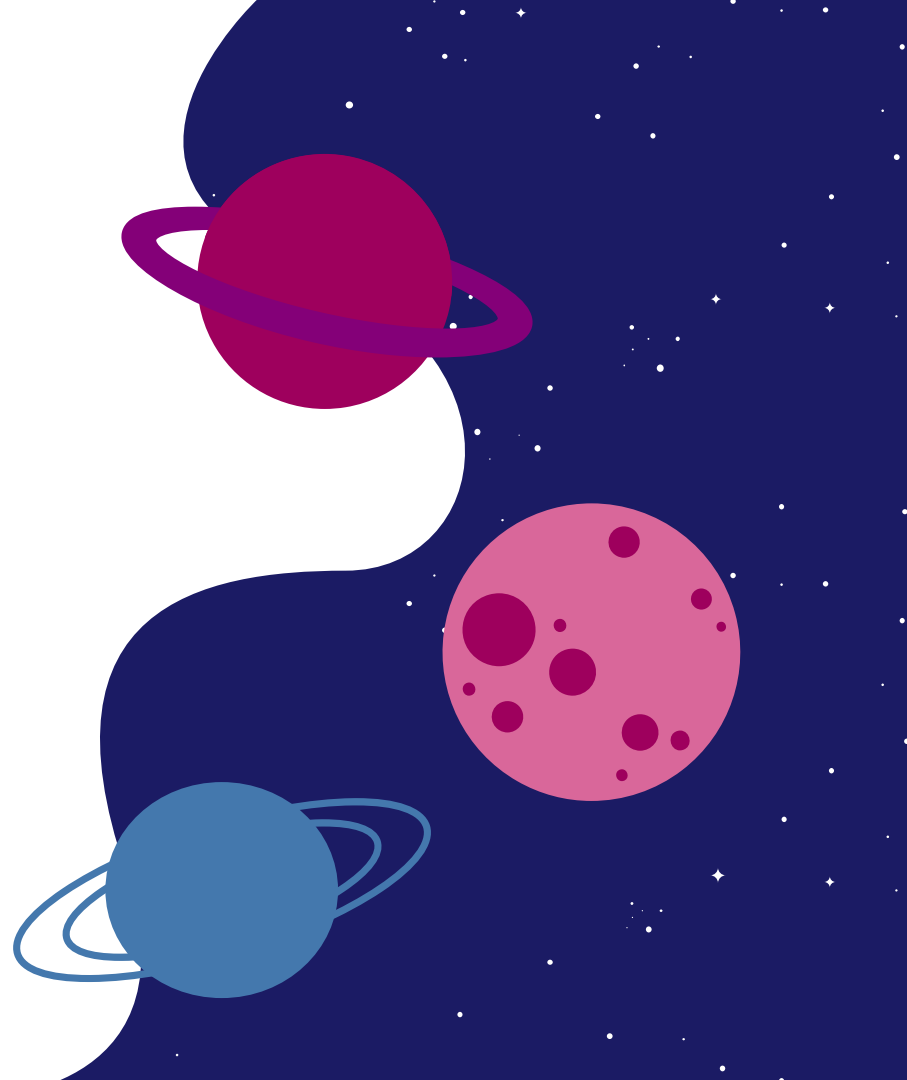
```
1 SELECT
2     MIN(buyPrice)
3 FROM
4     products;
```

AVG

untuk menghitung nilai rata-rata dari sekumpulan nilai atau ekspresi

Contoh

- ❑ mencari harga rata-rata
- ❑ mendapatkan nilai rata-rata dari *credit limit*



Sintaks

```
1 AVG(DISTINCT expression)
```

tidak menggunakan DISTINCT

1	1	2	2	3
---	---	---	---	---

menggunakan DISTINCT

1	2	3
---	---	---

Cuplikan tabel PRODUCTS

productCode	productName	quantityInStock	buyPrice	MSRP
S24_2840	1958 Chevy Corvette Limited Edition	2542	15.91	35.36
S24_2972	1982 Lamborghini Diablo	7723	16.24	37.76
S24_2022	1938 Cadillac V-16 Presidential Limousine	2847	20.61	44.80
S24_3969	1936 Mercedes Benz 500k Roadster	2081	21.75	41.03
S24_1937	1939 Chevrolet Deluxe Coupe	7332	22.57	33.19
S18_4668	1939 Cadillac Limousine	6645	23.14	50.31
S32_2206	1982 Ducati 996 R	9241	24.14	40.23
S18_2625	1936 Harley Davidson El Knucklehead	4357	24.23	60.57
S18_1367	1936 Mercedes-Benz 500K Special Roadster	8635	24.26	53.91
S18_2432	1926 Ford Fire Engine	2018	24.92	60.77

Query

```
1 SELECT
2     AVG(buyprice) 'Average Price'
3 FROM
4     products;
```

SUM

untuk menghitung jumlah dari sekumpulan nilai atau ekspresi

Sintaks

```
1 SUM(DISTINCT expression)
```

Bagaimana fungsi SUM dijalankan

- ❑ Apabila kita menggunakan fungsi SUM() dalam *statement* SELECT yang tidak mengembalikan hasil yang sesuai, fungsi SUM() menghasilkan NULL, bukan nol.
- ❑ Operator DISTINCT dapat digunakan untuk menghitung nilai-nilai berbeda dalam sekumpulan nilai.
- ❑ Fungsi SUM() mengabaikan nilai NULL dalam perhitungan.

Cuplikan tabel ORDERDETAILS

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10100	S18_1749	30	136.00	3
10100	S18_2248	50	55.09	2
10100	S18_4409	22	75.46	4
10100	S24_3969	49	35.29	1
10101	S18_2325	25	108.06	4
10101	S18_2795	26	167.06	1
10101	S24_1937	45	32.53	3
10101	S24_2022	46	44.35	2

Query

```
1 SELECT
2     FORMAT(SUM(quantityOrdered * priceEach),
3           2) total
4 FROM
5     orderdetails
6 WHERE
7     orderNumber = 10100;
```

COUNT

untuk menghitung semua baris dalam tabel atau hanya baris yang cocok dengan kondisi tertentu

Sintaks

```
1 COUNT(expression)
```

Bentuk Fungsi COUNT

- ❑ **COUNT(*)** → menghasilkan nilai berupa jumlah baris dari *result set* yang didapatkan dari *statement* SELECT yang **berisi nilai non-NULL dan NULL**
- ❑ **COUNT(expression)** → menghasilkan jumlah baris yang **tidak berisi nilai NULL**
- ❑ **COUNT (DISTINCT expression)** → menghasilkan jumlah baris **unik** yang **tidak berisi nilai NULL**

Tabel DEMOS

id	val
1	1
2	1
3	2
4	2
5	NULL
6	3
7	4
8	NULL
9	5

Query

```
1 SELECT COUNT(*)  
2 FROM demos;
```

Hasil

	COUNT(*)
▶	9

Query

```
1 SELECT COUNT(*)  
2 FROM demos  
3 WHERE val = 2;
```

Hasil

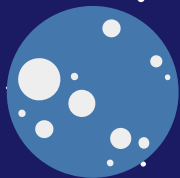
	COUNT(*)
▶	2

Query

```
1 SELECT COUNT(DISTINCT val)  
2 FROM demos;
```

Hasil

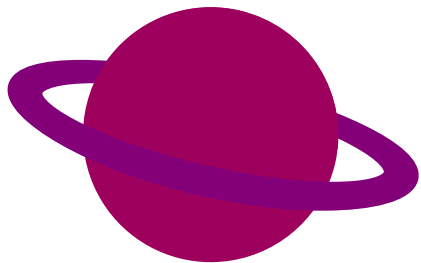
	COUNT(DISTINCT val)
▶	5



Grouping Data

Klausula **GROUP BY** digunakan untuk mengelompokkan sekumpulan baris menjadi sekumpulan ringkasan data menurut nilai kolom atau ekspresi

Klausula **GROUP BY** menghasilkan satu baris untuk setiap grup



Kita sering menggunakan klausa **GROUP BY** dengan **fungsi agregat** seperti **SUM, AVG, MAX, MIN, dan COUNT**. Fungsi agregat yang digunakan pada klausa SELECT memberikan informasi tentang setiap grup.

Klausa **GROUP BY** merupakan klausa opsional dari *statement* SELECT.

Sintaks

```
1 SELECT
2     c1, c2,..., cn, aggregate_function(ci)
3 FROM
4     table
5 WHERE
6     where_conditions
7 GROUP BY c1 , c2,...,cn;
```

Klausa **GROUP BY** harus muncul setelah klausa FROM dan WHERE
Keyword **GROUP BY** diikuti dengan daftar kolom atau ekspresi yang dipisahkan dengan koma (digunakan sebagai kriteria untuk mengelompokkan data)

Cuplikan tabel ORDERS

orderNumber	orderDate	requiredDate	shippedDate	status	comments	customerNumber
10410	2005-04-29	2005-05-10	2005-04-30	Shipped	NULL	357
10411	2005-05-01	2005-05-08	2005-05-06	Shipped	NULL	233
10412	2005-05-03	2005-05-13	2005-05-05	Shipped	NULL	141
10413	2005-05-05	2005-05-14	2005-05-09	Shipped	Customer requested that DHL is used for this shipp...	175
10414	2005-05-06	2005-05-13	NULL	On Hold	Customer credit limit exceeded. Will ship when a p...	362
10415	2005-05-09	2005-05-20	2005-05-12	Disputed	Customer claims the scales of the models don't mat...	471
10416	2005-05-10	2005-05-16	2005-05-14	Shipped	NULL	386
10417	2005-05-13	2005-05-19	2005-05-19	Disputed	Customer doesn't like the colors and precision of ...	141
10418	2005-05-16	2005-05-24	2005-05-20	Shipped	NULL	412
10419	2005-05-17	2005-05-28	2005-05-19	Shipped	NULL	382
10420	2005-05-29	2005-06-07	NULL	In Process	NULL	282

Query

```
1 SELECT
2     status
3 FROM
4     orders
5 GROUP BY status;
```

Hasil

	status
►	Cancelled
	Disputed
	In Process
	On Hold
	Resolved
	Shipped

Query

```
1 SELECT
2     status, COUNT(*)
3 FROM
4     orders
5 GROUP BY status;
```

Hasil

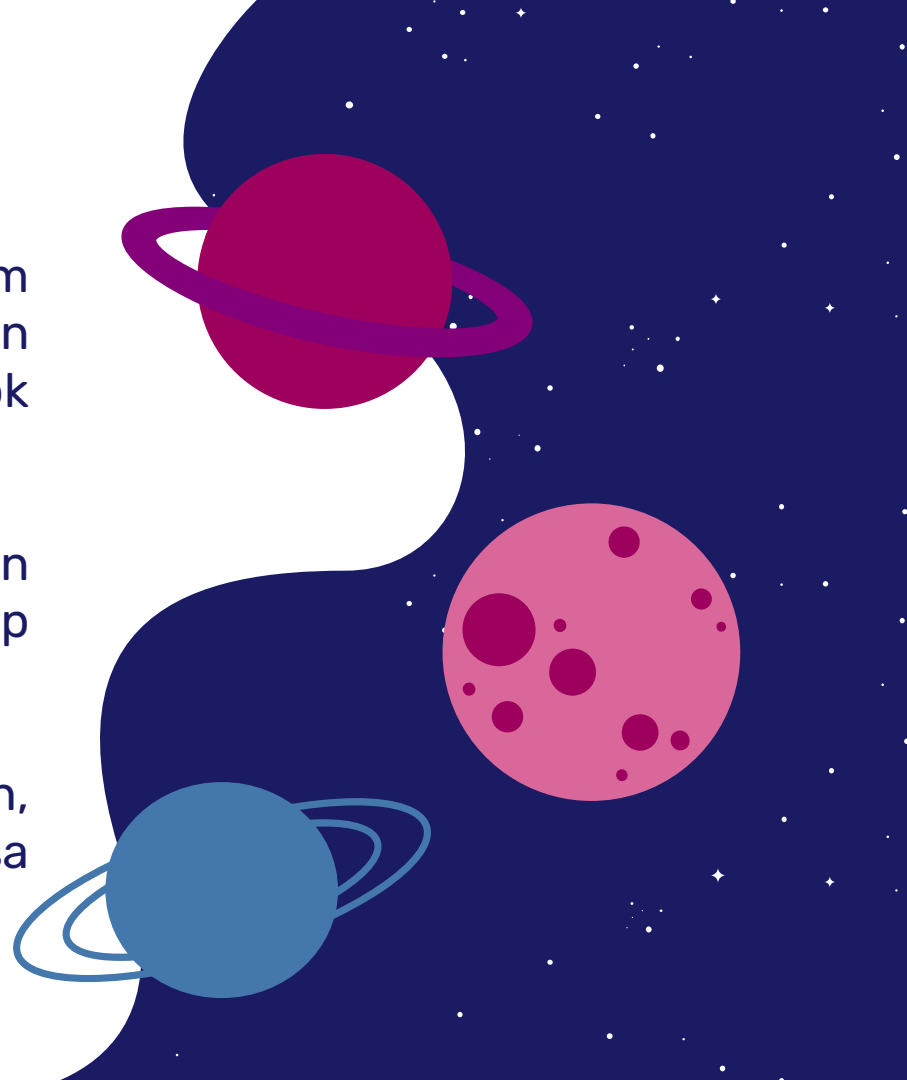
	status	COUNT(*)
►	Cancelled	6
	Disputed	3
	In Process	6
	On Hold	4
	Resolved	4
	Shipped	303

HAVING

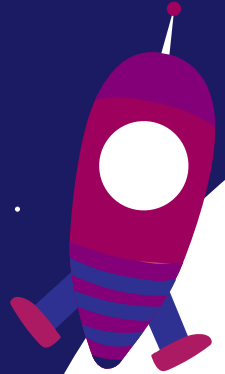
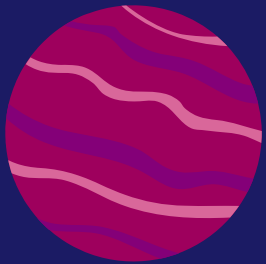
Klausa **HAVING** digunakan dalam *statement* SELECT untuk menetapkan kondisi yang menjadi *filter* sekelompok baris atau agregat

Klausa **HAVING** sering digunakan dengan klausa GROUP BY untuk mem-*filter* grup berdasarkan kondisi tertentu.

Apabila klausa **GROUP BY** dihilangkan, klausa HAVING dijalankan seperti klausa WHERE.



Perhatikan bahwa klausa **HAVING** menerapkan kondisi yang menjadi *filter* ke setiap **grup baris**, sedangkan klausa WHERE menerapkan kondisi *filter* ke **setiap baris**.



Cuplikan tabel ORDERDETAILS

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10100	S18_1749	30	136.00	3
10100	S18_2248	50	55.09	2
10100	S18_4409	22	75.46	4
10100	S24_3969	49	35.29	1
10101	S18_2325	25	108.06	4
10101	S18_2795	26	167.06	1
10101	S24_1937	45	32.53	3
10101	S24_2022	46	44.35	2

Query

```
1 SELECT
2     ordernumber,
3     SUM(quantityOrdered) AS itemCount,
4     SUM(priceeach*quantityOrdered) AS total
5 FROM
6     orderdetails
7 GROUP BY ordernumber
8 HAVING total > 1000;
```

Hasil

	ordernumber	itemsCount	total
▶	10100	151	10223.83
	10101	142	10549.01
	10102	80	5494.78
	10103	541	50218.95
	10104	443	40206.20
	10105	545	53959.21
	10106	675	52151.81
	10107	229	22292.62

**TERIMA
KASIH**

