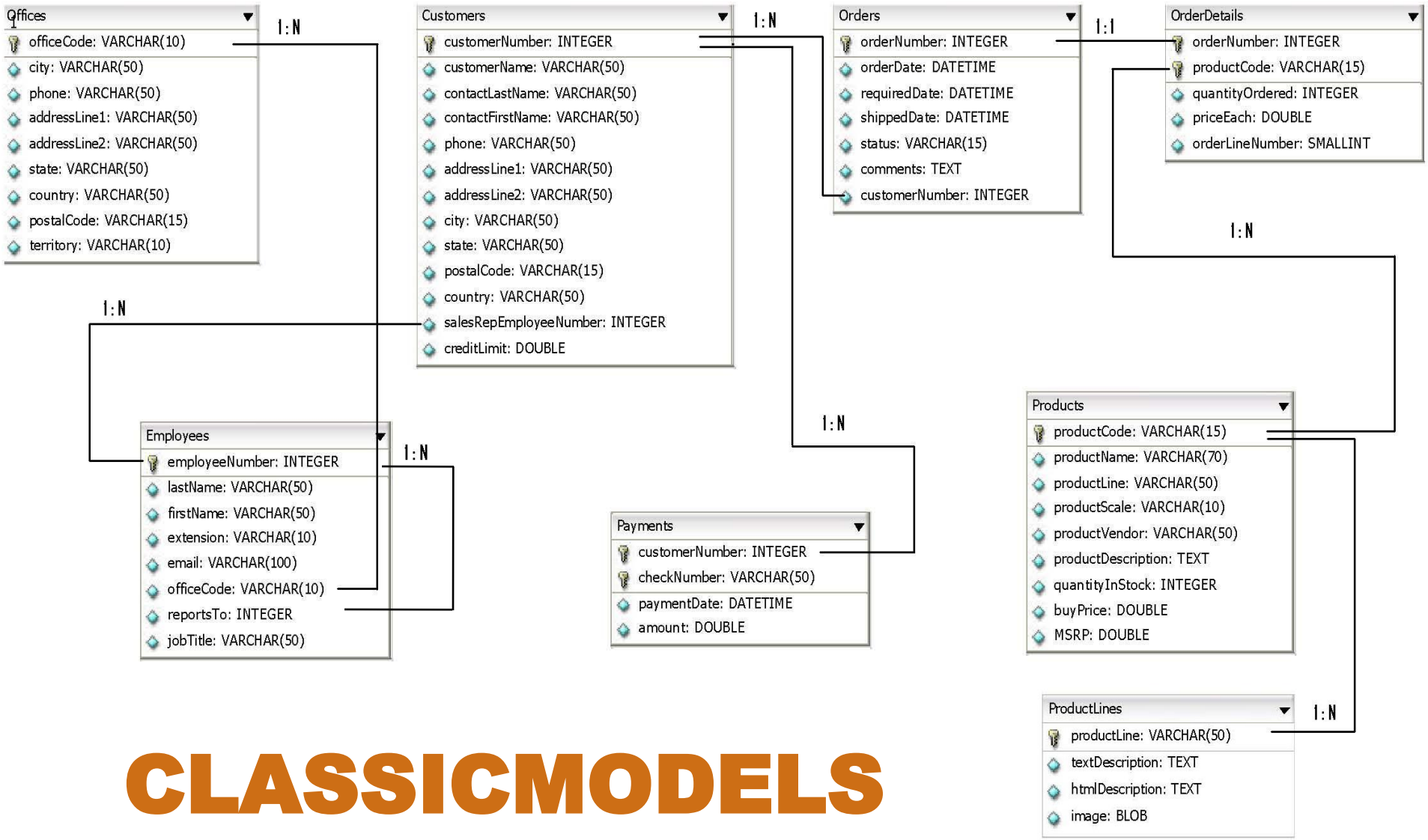
7

AGGREGATE FUNCTION

SKEMA RELASI YANG DIGUNAKAN

CLASSICMODELS



CLASSICMODELS

CONTOH TABEL T3

Tabel T3

a	b
1	4
2	7
3	9
4	4
5	NULL

Aggregate Functions

Nama	Deskripsi
AVG	Rata-rata nilai
BIT_AND	Operasi bit AND
BIT_OR	Operasi bit OR
BIT_XOR	Operasi bit XOR
COUNT	Banyak baris/record
GROUP_CONCAT	Menggabungkan menjadi string
MAX	Nilai terbesar
MIN	Nilai terkecil

Nama	Deskripsi
STD	Standar deviasi populasi
STDDEV_POP	Sama dengan STD
STDDEV_SAMP	Standar deviasi dari data sample
STDDEV	Sama dengan STD
SUM	Penjumlahan nilai
VAR_POP	Variance populasi
VAR_SAMP	Variance dari data sample
VARIANCE	Sama dengan VAR_POP

SUM([DISTINCT] *expr*)

- SUM digunakan untuk mencari hasil penjumlahan semua *expr* dari ekspresi yang menjadi argumen/parameternya.
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- Ekspresi boleh berupa nama kolom, function atau rumus.
- DISTINCT dapat digunakan jika ingin mencari total dari nilai yang berbeda. Sebuah nilai hanya akan dihitung 1 kali. Jika ditemukan nilai yang sama dalam baris lain, maka tidak akan dihitung.

COUNT([DISTINCT] *expr*) / COUNT(*)

- COUNT(expr) akan menghitung banyaknya baris yang nilai expr bernilai NOT NULL
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- DISTINCT dapat digunakan jika ingin banyaknya baris dari nilai yang berbeda. Sebuah nilai hanya akan dihitung 1 kali. Jika ditemukan nilai yang sama dalam baris lain, maka tidak akan dihitung.
- COUNT(*) digunakan untuk mencari banyaknya baris

CONTOH COUNT() & SUM()

```
SELECT SUM(buyprice),SUM(DISTINCT buyprice),
       COUNT(buyprice),COUNT(DISTINCT buyprice), COUNT(*)
FROM products;
```

SUM(buyprice)	SUM(DISTINCT buyprice)	COUNT(buyprice)	COUNT(DISTINCT buyprice)	COUNT(*)
5983.47	5872.270000000001	110	108	110

CONTOH COUNT() & SUM()

```
SELECT SUM(b),SUM(DISTINCT b),
       COUNT(b),COUNT(DISTINCT b), COUNT(*)
FROM t3;
```

SUM(b)	SUM(DISTINCT b)	COUNT(b)	COUNT(DISTINCT b)	COUNT(*)
24	20	4	3	5

AVG([DISTINCT] *expr*)

- AVG digunakan untuk mencari nilai rata-rata dari ekspresi yang menjadi argumen/parameternya.
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- Ekspresi boleh berupa nama kolom, function atau rumus.
- DISTINCT dapat digunakan jika ingin mencari nilai rata-rata dari nilai yang berbeda. Sebuah nilai hanya akan dihitung 1 kali. Jika ditemukan nilai yang sama dalam baris lain, maka tidak akan dihitung.

MAX([DISTINCT] *expr*)

- MAX digunakan untuk mencari nilai paling besar dari ekspresi yang menjadi argumen/parameternya.
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- Ekspresi boleh berupa nama kolom, function atau rumus.
- DISTINCT dapat digunakan, tetapi akan menghasilkan hasil yang sama dengan tanpa menggunakan DISTINCT.

MIN([DISTINCT] *expr*)

- MIN digunakan untuk mencari nilai paling kecil dari ekspresi yang menjadi argumen/parameternya.
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- Ekspresi boleh berupa nama kolom, function atau rumus.
- DISTINCT dapat digunakan, tetapi akan menghasilkan hasil yang sama dengan tanpa menggunakan DISTINCT.

CONTOH AVG(), MAX(), MIN()

- Mencari nilai terkecil, terbesar dan rata-rata harga beli (buyprice) dari tabel products.

```
SELECT MIN(buyprice),MAX(buyprice),
       AVG(buyprice)
FROM products;
```

MIN(buyprice)	MAX(buyprice)	AVG(buyprice)
15.91	103.42	54.395181818181818

CONTOH AVG(), MAX(), MIN()

- Mencari nilai terkecil, terbesar dan rata-rata dari field b dari tabel t3.

```
SELECT MIN(b),MAX(b),AVG(b)
FROM t3;
```

MIN(b)	MAX(b)	AVG(b)
4	9	6.0000

- Perhatikan, rata-rata hanya dihitung berdasarkan data yang tidak NULL (4+7+9+4)/4 = 24/4 = 6.00

GROUP_CONCAT(*expr*)

- Digunakan untuk menggabungkan isi-isi expr yang NOT NULL sebagai sebuah string.
- Ekspresi yang akan dihitung adalah ekspresi yang NOT NULL.
- Ekspresi boleh berupa nama kolom, function atau rumus.
- DISTINCT dapat digunakan untuk hanya menampilkan data yang berbeda saja.

```
GROUP_CONCAT([DISTINCT] expr [,expr ...]
             [ORDER BY {unsigned_integer | col_name | expr}
                    [ASC | DESC][,col_name ...]]
             [SEPARATOR str_val])
```

CONTOH GROUP_CONCAT()

- Digunakan untuk menggabungkan isi-isi expr sebagai sebuah string.

```
SELECT GROUP_CONCAT(b),
       GROUP_CONCAT(DISTINCT B ORDER BY B DESC SEPARATOR '+')
FROM t3;
```

GROUP_CONCAT(b)	GROUP_CONCAT(DISTINCT B ORDER BY B DESC SEPARATOR '+')
4,7,9,4	9+7+4

BIT_AND(*expr*), BIT_OR(*expr*) , BIT_XOR(*expr*)

- BIT_AND : melakukan operasi bit AND
- BIT_OR : melakukan operasi bit OR
- BIT_XOR : melakukan operasi bit XOR

BIT_AND(*expr*), BIT_OR(*expr*) , BIT_XOR(*expr*)

```
SELECT BIT_AND(b),BIT_OR(b),BIT_XOR(b)
FROM t3;
```

BIT_AND(b)	BIT_OR(b)	BIT_XOR(b)
0	15	14

b	Biner	AND	OR	XOR
4	0100	0100	0100	0100
7	0111	0111	0111	0111
9	1001	1001	1001	1001
4	0100	0100	0100	0100
		0000	1111	1110

VARIANCE(*expr*), VAR_POP(*expr*) , VAR_SAMP(*expr*)

- VARIANCE() digunakan untuk mencari nilai variance dari semua data (populasi) nilai expr.
- VAR_POP() = VARIANCE()
- VAR_SAMP() digunakan untuk mencari nilai variance dari sample data nilai expr

CONTOH VARIANCE(), VAR_POP() , VAR_SAMP()

```
SELECT VARIANCE(b),VAR_POP(b), VAR_SAMP(b)
FROM t3;
```

VARIANCE(b)	VAR_POP(b)	VAR_SAMP(b)
4.5000	4.5000	6.0000

STD(*expr*) , STDDEV(*expr*) , STDDEV_POP(*expr*) , STDDEV_SAMP(*expr*)

- STD() digunakan untuk mencari nilai standar deviasi dari semua data (populasi) nilai expr.
- STDDEV() = STDDEV_POP() = VARIANCE()
- STDDEV() digunakan untuk mencari nilai standar deviasi dari sample data nilai expr

STD(*expr*) , STDDEV(*expr*) , STDDEV_POP(*expr*) , STDDEV_SAMP(*expr*)

```
SELECT STD(b),STDDEV(b), STDDEV_POP(b), STDDEV_SAMP(b)
FROM t3;
```

STD(b)	STDDEV(b)	STDDEV_POP(b)	STDDEV_SAMP(b)
2.1213	2.1213	2.1213	2.4495

$$stddev = \sqrt{\text{var}\,iance}$$

NULL VALUE PROBLEM

- Function-function agregasi sebelumnya, hanya melakukan perhitungan terhadap data yang tidak NULL.
- Jika anda ingin melakukan perhitungan terhadap semua data (NULL atau tidak NULL) maka ada trik yang bisa anda lakukan yaitu dengan menambahkan function IFNULL

NULL VALUE PROBLEM

```
SELECT AVG(b), AVG(IFNULL(b,0)),
       COUNT(b),COUNT(IFNULL(b,0))
FROM t3;
```

AVG(b)	AVG(IFNULL(b,0))	COUNT(b)	COUNT(IFNULL(b,0))
6.0000	4.8000	4	5

Dengan menambahkan IFNULL(b,0), maka ketika b bernilai NULL, maka akan dianggap bernilai 0.

GROUP BY

- GROUP BY digunakan jika anda ingin melakukan proses perhitungan function aggregasi berdasarkan kelompok data.
- SELECT yang menggunakan function aggregasi tanpa menggunakan GROUP BY hanya akan menghasilkan 1 baris data
- SELECT yang menggunakan function aggregasi dan GROUP BY akan mungkin menghasilkan lebih dari 1 baris, tergantung dengan banyaknya group yang ada (DISTINCT dari kolom yang menjadi acuan GROUPING).

GROUP BY

```
SELECT productLine, MAX(buyprice), MIN(buyprice),
       AVG(buyprice), SUM(quantityInStock)
FROM products
GROUP BY productLine;
```

productLine	MAX(buyprice)	MIN(buyprice)	AVG(buyprice)	SUM(quantityInStock)
Classic Cars	103.42	15.91	64.4463157894737	219183
Motorcycles	91.02	24.14	50.6853846153846	69401
Planes	77.27	29.34	49.6291666666667	62287
Ships	82.34	33.3	47.0077777777778	26833
Trains	67.56	26.72	43.9233333333333	16696
Trucks and Buses	84.76	24.92	56.3290909090909	35851
Vintage Cars	86.7	20.61	46.06625	124880

GROUP BY BERDASARKAN 2 KOLOM ATAU LEBIH

```
SELECT productLine,productScale, MAX(buyprice), MIN(buyprice),
       AVG(buyprice), SUM(quantityInStock)
FROM products
GROUP BY productLine,productScale;
```

productLine	productScale	MAX(buyprice)	MIN(buyprice)	AVG(buyprice)	SUM(quantityInStock)
Classic Cars	1:10	103.42	85.68	95.8933333333333	17348
Classic Cars	1:12	95.59	31.92	73.2983333333333	26845
Classic Cars	1:18	101.51	32.33	68.75	118929
Classic Cars	1:24	72.82	15.91	43.9990909090909	56061
Motorcycles	1:10	91.02	48.81	69.6066666666667	20140
Motorcycles	1:12	66.27	66.27	66.27	9997
...					
Trucks and Buses	1:50	68.29	68.29	68.29	1016
Vintage Cars	1:18	86.7	23.14	53.418	82016
Vintage Cars	1:24	57.46	20.61	34.8914285714286	35666
Vintage Cars	1:32	33.02	33.02	33.02	136
Vintage Cars	1:50	27.06	27.06	27.06	7062

GROUP BY BERDASARKAN FUNCTION

```
SELECT YEAR(orderDate) Tahun,COUNT(*) BanyakPesanan
FROM orders
GROUP BY YEAR(orderDate);
```

```
SELECT YEAR(orderDate) Tahun,COUNT(*) BanyakPesanan
FROM orders
GROUP BY Tahun;
```

Group By menggunakan Alias Kolom

Tahun	BanyakPesanan
2003	111
2004	151
2005	64

GROUP BY BERDASARKAN FUNCTION

```
SELECT YEAR(orderDate) Tahun, MONTH(orderDate) Bulan,
       COUNT(*) BanyakPesanan
FROM orders
GROUP BY Tahun, Bulan;
```

Group By menggunakan Alias Kolom

Tahun	Bulan	BanyakPesanan
2003	1	5
2003	2	3
2003	3	6
2003	4	7
2003	5	6
...		
2004	12	13
2005	1	12
2005	2	12
2005	3	13
2005	4	12
2005	5	15

PENGURUTAN DATA PADA GROUP BY

```
SELECT YEAR(orderDate) Tahun, MONTH(orderDate) Bulan,
       COUNT(*) BanyakPesanan
FROM orders GROUP BY Tahun, Bulan
ORDER BY BanyakPesanan DESC,Tahun;
```

Tahun	Bulan	BanyakPesanan
2004	11	33
2003	11	30
2003	10	18
2005	5	15
2004	10	13
2004	12	13
2005	3	13
...		
2003	8	5
2003	1	5
2003	2	3

PENGURUTAN DATA PADA GROUP BY

```
SELECT YEAR(orderDate) Tahun, MONTH(orderDate) Bulan,
       COUNT(*) BanyakPesanan
FROM orders GROUP BY Tahun, Bulan
ORDER BY BanyakPesanan DESC LIMIT 5;
```

Tahun	Bulan	BanyakPesanan
2004	11	33
2003	11	30
2003	10	18
2005	5	15
2004	10	13

WHERE + GROUP BY

- WHERE dapat digabungkan dengan GROUP BY. WHERE ditulis sebelum perintah GROUP BY
- WHERE digunakan untuk memfilter baris-baris data SEBELUM data tersebut digroupkan
- Tidak boleh menggunakan WHERE setelah GROUP BY
- Tidak boleh menggunakan function aggregasi pada bagian WHERE.

WHERE + GROUP BY

```
SELECT productLine, MAX(buyprice), MIN(buyprice),
       AVG(buyprice), SUM(quantityInStock)
FROM products
WHERE buyprice>=50 AND buyprice<75
GROUP BY productLine;
```

productLine	MAX(buyprice)	MIN(buyprice)	AVG(buyprice)	SUM(quantityInStock)
Classic Cars	73.49	50.51	62.7633333333333	82629
Motorcycles	68.99	56.13	63.834	27144
Planes	68.8	51.15	60.084	26778
Ships	53.63	51.09	52.11	11303
Trains	67.56	67.56	67.56	6450
Trucks and Buses	74.86	53.93	62.075	25438
Vintage Cars	72.56	52.66	61.372	55980

SELEKSI BERDASARKAN HASIL GROUP BY

- Jika ingin melakukan seleksi berdasarkan hasil group by dapat menggunakan HAVING bukan WHERE.
- HAVING ditulis setelah perintah GROUP BY, karena HAVING akan melakukan seleksi terhadap hasil GROUP BY
- Kriteria HAVING dapat menggunakan nama function aggregasi atau alias kolomnya.

SELEKSI BERDASARKAN HASIL GROUP BY

```
SELECT productLine, MAX(buyprice), MIN(buyprice),
       AVG(buyprice), SUM(quantityInStock)
FROM products
GROUP BY productLine
HAVING SUM(quantityInStock)>50000;
```

productLine	MAX(buyprice)	MIN(buyprice)	AVG(buyprice)	SUM(quantityInStock)
Classic Cars	103.42	15.91	64.4463157894737	219183
Motorcycles	91.02	24.14	50.6853846153846	69401
Planes	77.27	29.34	49.6291666666667	62287
Vintage Cars	86.7	20.61	46.06625	124880

SELEKSI BERDASARKAN HASIL GROUP BY

```
SELECT productLine, MAX(buyprice), MIN(buyprice),
       AVG(buyprice), SUM(quantityInStock) Total
FROM products
GROUP BY productLine
HAVING Total>50000;
```

productLine	MAX(buyprice)	MIN(buyprice)	AVG(buyprice)	Total
Classic Cars	103.42	15.91	64.4463157894737	219183
Motorcycles	91.02	24.14	50.6853846153846	69401
Planes	77.27	29.34	49.6291666666667	62287
Vintage Cars	86.7	20.61	46.06625	124880

ROLLUP

- Dengan mengambahkan WITH ROLLUP, maka akan ada baris tambahan yang berisi penjumlahan dari data-data di atasnya.
- ROLLUP bisa digunakan untuk melakukan aggregasi multilevel dalam sebuah perintah SQL.

ROLLUP

```
SELECT YEAR(orderDate) Tahun, MONTH(orderDate) Bulan,
       COUNT(*) BanyakPesanan
FROM orders
WHERE YEAR(orderDate) in (2004,2005)
GROUP BY Tahun, Bulan WITH ROLLUP;
```

Tahun	Bulan	BanyakPesanan
2004	1	8
2004	2	11
2004	3	8
2004	4	10
2004	5	8
2004	6	12
2004	7	11
2004	8	12
2004	9	12
2004	10	13
2004	11	33
2004	12	13
2004	NULL	151
2005	1	12
2005	2	12
2005	3	13
2005	4	12
2005	5	15
2005	NULL	64
NULL	NULL	215

LATIHAN-LATIHAN

Tampilkan nama negara pelanggan yang pemesanannya paling banyak.

country	BanyakPesanan
USA	112

LATIHAN-LATIHAN

Berapa total pembayaran per customer yang pernah dilakukan oleh customer yang bernegara di France.

CustomerNumber	CustomerName	TotalPembayaran
103	Atelier graphique	22314.36
119	La Rochelle Gifts	116949.68
146	Saveley & Henriot, Co.	130305.35
171	Daedalus Designs Imports	61781.7
172	La Corne D'abondance, Co.	86553.52
209	Mini Caravy	75859.32
242	Alpha Cognac	60483.36
250	Lyon Souveniers	67659.19
256	Auto Associés & Cie.	58876.41
350	Marseille Mini Autos	71547.53
353	Reims Collectables	126983.19
406	Auto Canal+ Petit	86436.97

12 Rows