



Databases

7. SQL introductie (SELECT)

Inhoud

- 7.0. SQL
- 7.1. SELECT
- 7.2. WHERE
- 7.3. Formatteren van resultaten

SQL standaarden en dialecten

- Definitie: Standard Query Language
 - Relationale gegevenstaal voor relationele databanksystemen
 - Niet-procedurele taal
- Standaard : ANSI/ISO-1992

SQL standaarden en dialecten

- Database systemen
 - Oracle : PL/SQL
 - SQL Server : TRANSACT-SQL
 - Db2 (IBM)
 - Informix
 - Sybase
 - MySQL

SQL overzicht

- **Data Definition Language (DDL)**
 - Creatie van een database, en het definiëren van database-objecten (tabellen, stored procedures, views, ...)
 - CREATE, ALTER, DROP
- **Data Manipulation Language (DML)**
 - Opvragen en manipuleren van de gegevens in een database
 - SELECT, INSERT, UPDATE, DELETE
- **Data Control Language (DCL)**
 - Gegevensbeveiliging en autorisatie
 - GRANT, REVOKE, DENY
- Operatoren, functies, control of flow (dialect gebonden)

DML: raadplegen van gegevens

- Eén tabel raadplegen
 - Basisvorm
 - SELECT clausule
 - WHERE clausule
 - Formatteren van rijen
 - Statistische functies
 - Groeperen
- Meerdere tabellen raadplegen

7.1. SELECT

Basisvorm SELECT statement

- SELECT voor raadplegen van één tabel

SELECT [ALL | **DISTINCT**] {*}|uitdrukking [,uitdrukking ...]}

FROM tabelnaam

[**WHERE** voorwaarde(n)]

[**GROUP BY** kolomnaam [,kolomnaam ...]]

[**HAVING** voorwaarde(n)]

[**ORDER BY** {kolomnaam|volgnr}{ASC|**DESC**}[,...]]

Basisvorm SELECT statement

- **SELECT** clausule: specificeert de kolommen die je wenst te zien
 - **DISTINCT** zorgt ervoor dat de getoonde rijen allen uniek zijn
- **FROM** clausule: geeft aan uit welke tabel de gegevens afkomstig zijn
- **WHERE** clausule: opgave van de voorwaarden waaraan de getoonde rijen moeten voldoen
- **ORDER BY** clausule: bepaalt de volgorde waarin de rijen getoond moeten worden
- **GROUP BY** en **HAVING** clausule: groeperen van de gegevens

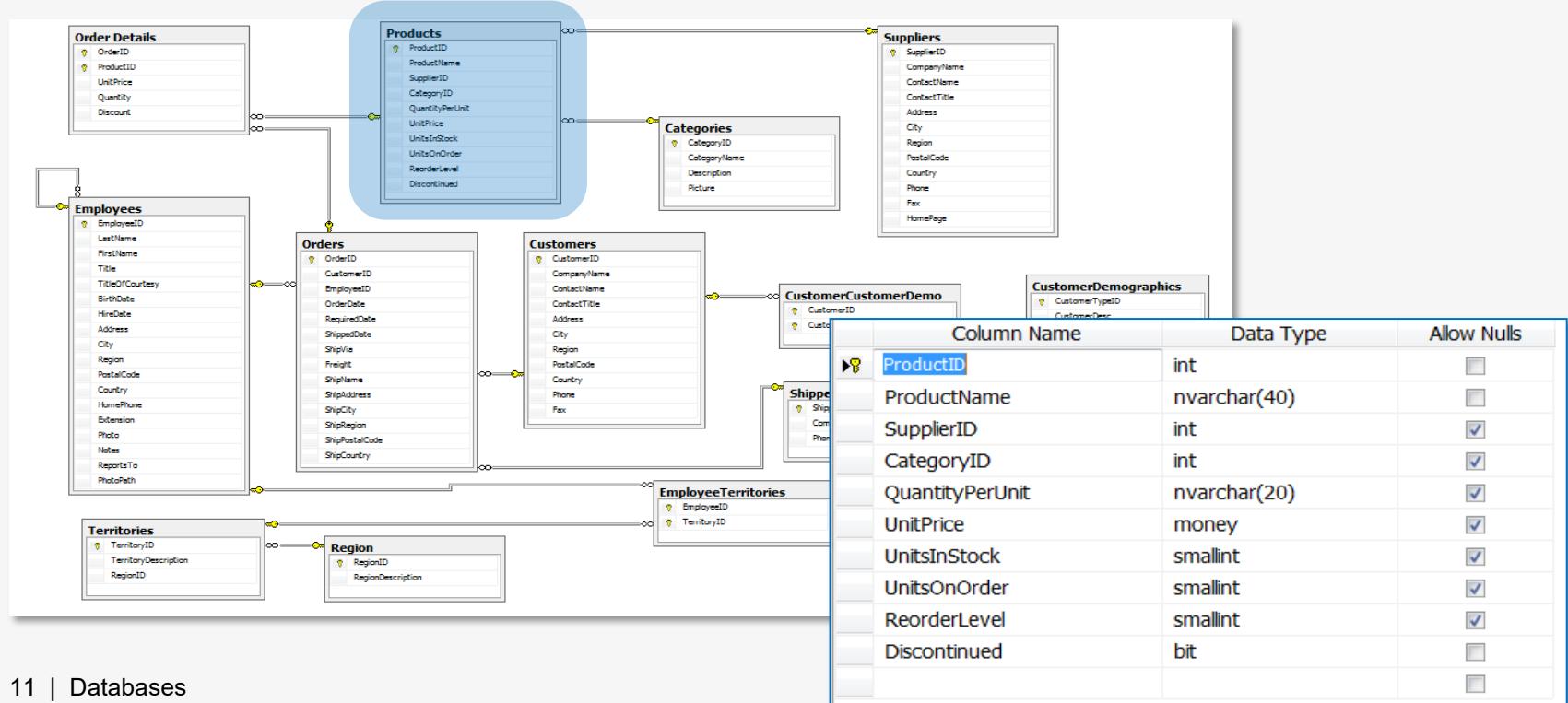
SELECT statement

SELECT clausule: **KOLOMMEN SELECTEREN**

- Ophalen van alle kolommen uit 1 tabel: gebruik *
→ SELECT *
- Ophalen van specifieke kolommen: gebruik kolomnamen of uitdrukking
→ SELECT kolom1, kolom2, ...

Voorbeeld 1

Toon alle gegevens van alle producten



Voorbeeld 1

Toon alle gegevens van alle producten

```
SELECT *  
FROM products
```

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays various database management options like Server Status, Client Connections, and Performance Reports. Below that, the Schemas pane shows the current database selected is 'northwind'. The central area contains a SQL editor window with the query: 'select * from products;'. To the right of the SQL editor is a Result Grid displaying the data from the products table. The table has 17 rows and the following columns: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, and Discontinued. The data includes various products like Chai, Chang, Aniseed Syrup, etc., with their respective details.

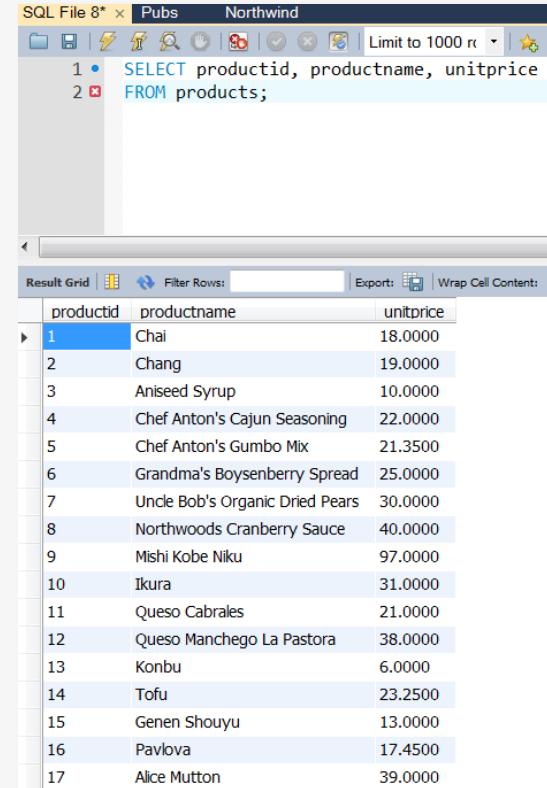
ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	Chai	8	1	10 boxes x 30 bags	18.0000	39	0	10	1
2	Chang	1	1	24 - 12 oz bottles	19.0000	17	40	25	1
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.0000	13	70	25	0
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.0000	53	0	0	0
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.3500	0	0	0	1
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.0000	120	0	25	0
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30.0000	15	0	10	0
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40.0000	6	0	0	0
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97.0000	29	0	0	1
10	Ikura	4	8	12 - 200 ml jars	31.0000	31	0	0	0
11	Queso Cabrales	5	4	1 kg pkg.	21.0000	22	30	30	0
12	Queso Manchego La Pastora	5	4	10 - 500 g pkgs.	38.0000	86	0	0	0
13	Konbu	6	8	2 kg box	6.0000	24	0	5	0
14	Tofu	6	7	40 - 100 g pkgs.	23.2500	35	0	0	0
15	Genen Shouyu	6	2	24 - 250 ml bottles	13.0000	39	0	5	0
16	Pavlova	7	3	32 - 500 g boxes	17.4500	29	0	10	0
17	Alice Mutton	7	6	20 - 1 kg tins	39.0000	0	0	0	1

Voorbeeld 2

Toon van alle producten enkel:

- productID
- naam
- eenheidsprijs

```
SELECT productid, productname, unitprice  
FROM products
```



The screenshot shows a SQL query window titled "SQL File 8*". The query is:

```
1 • SELECT productid, productname, unitprice  
2 ✘ FROM products;
```

The results are displayed in a "Result Grid" table:

productid	productname	unitprice
1	Chai	18.0000
2	Chang	19.0000
3	Aniseed Syrup	10.0000
4	Chef Anton's Cajun Seasoning	22.0000
5	Chef Anton's Gumbo Mix	21.3500
6	Grandma's Boysenberry Spread	25.0000
7	Uncle Bob's Organic Dried Pears	30.0000
8	Northwoods Cranberry Sauce	40.0000
9	Mishi Kobe Niku	97.0000
10	Ikura	31.0000
11	Queso Cabrales	21.0000
12	Queso Manchego La Pastora	38.0000
13	Konbu	6.0000
14	Tofu	23.2500
15	Genen Shouyu	13.0000
16	Pavlova	17.4500
17	Alice Mutton	39.0000

7.2. WHERE

WHERE clause

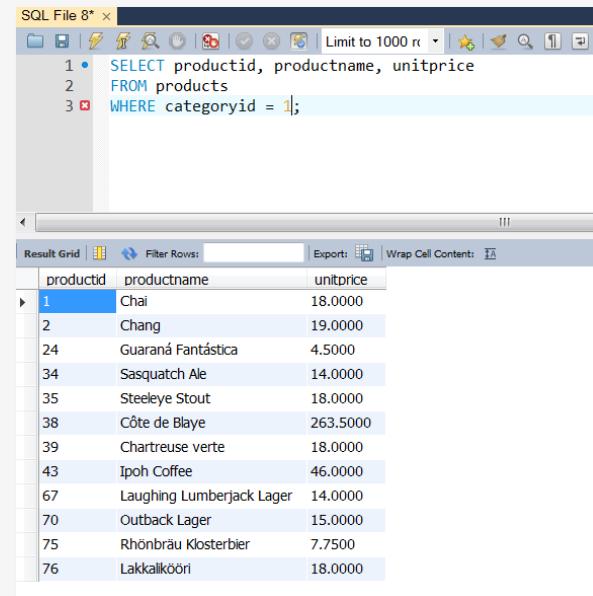
WHERE = specificatie van de voorwaarden waaraan de getoonde rijen moeten voldoen (**RIJEN SELECTEREN**)

Voorbeeld: Selecteer:

- productID
- naam
- eenheidsprijs

van producten **die behoren tot categorie 1**

```
SELECT productid, productname, unitprice  
FROM products  
WHERE categoryid = 1
```



The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the following query is displayed:

```
1 • SELECT productid, productname, unitprice  
2   FROM products  
3 □ WHERE categoryid = 1;
```

In the Result Grid tab, the results of the query are shown:

productid	productname	unitprice
1	Chai	18.0000
2	Chang	19.0000
24	Guaraná Fantástica	4.5000
34	Sasquatch Ale	14.0000
35	Steeleye Stout	18.0000
38	Côte de Blaye	263.5000
39	Chartreuse verte	18.0000
43	Ipoh Coffee	46.0000
67	Laughing Lumberjack Lager	14.0000
70	Outback Lager	15.0000
75	Rhönbräu Klosterbier	7.7500
76	Lakkaliköori	18.0000

WHERE clause

- Gebruik van constanten:
 - Numerische waarden: ... **WHERE categoryID = 1**
 - Alfanumerische waarden: ... **WHERE productName = 'Chai'**
 - Datums: ... **WHERE orderDate = '1996-07-04 00:00:00'**

WHERE clause

- Voorwaarden aan rijen opleggen
 - Vergelijkingsoperatoren
 - Wildcards
 - Logische operatoren
 - Een interval van specifieke waarden
 - Een lijst van waarden
 - Onbekende waarden
 - Je kan haakjes gebruiken om de prioriteitsregels te doorbreken of het geheel leesbaarder te maken

Vergelijkingsoperatoren

Vergelijkingsoperatoren

=, >, >=, <, <=, <>

Voorbeeld 1:

- Toon productID, naam, aantal in stock van de producten waarvan er minder dan 5 in stock zijn

```
SELECT productid, productname, unitsinstock  
FROM products  
WHERE unitsinstock < 5
```

The screenshot shows the MySQL Workbench interface. At the top, there is a SQL editor window with two numbered lines of code:
1 • select productid, productname, unitsinstock
2 from products

Below the editor is a "Result Grid" table. The table has three columns: productid, productname, and unitsinstock. The data is as follows:

	productid	productname	unitsinstock
1	5	Chef Anton's Gumbo Mix	0
2	17	Alice Mutton	0
	21	Sir Rodney's Scones	3
	29	Thüringer Rostbratwurst	0
	31	Gorgonzola Telino	0
	53	Perth Pasties	0
	66	Louisiana Hot Spiced Okra	4
	74	Longlife Tofu	4

Vergelijkinsoperatoren

Vergelijkinsoperatoren

=, >, >=, <, <=, <>

Voorbeeld 2:

- Toon productID, naam, aantal in stock van de producten waarvan de naam begint met een A

```
SELECT productid, productname  
FROM products  
WHERE productname >= 'A' AND productname < 'B'
```

The screenshot shows the MySQL Workbench interface. On the left, the SQL editor contains the following code:

```
1 • select productid, productname  
2   from products  
3   where productname >= 'A' and  
4       productname < 'B'
```

On the right, the Result Grid displays the query results:

productid	productname
17	Alice Mutton
3	Aniseed Syrup

Wildcards

Wildcards (zoeken naar patronen)

- De operator **LIKE, NOT LIKE**
- In combinatie met wildcards:
 - % willekeurige tekenrij met 0 of meerdere tekens
 - _ 1 teken

Wildcards

Wildcards (zoeken naar patronen)

- De operator **LIKE, NOT LIKE**

Voorbeeld:

Toon productID, naam van de producten waarbij de tekenreeks 'anton' voorkomt in de naam

```
SELECT productid, productname  
FROM products  
WHERE productname LIKE '%anton%'
```

The screenshot shows a database interface with a code editor and a result grid. The code editor contains the following SQL query:

```
1 select productid, productname  
2 FROM products  
3 WHERE productname LIKE '%anton%'  
4 |
```

The result grid below shows the output:

productid	productname
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix

Logische operatoren

Logische operatoren

- **OR, AND, NOT** (volgens stijgende prioriteit)

Voorbeelden:

```
SELECT productid, productname, supplierid, unitprice
FROM products
WHERE (productname LIKE 'T%' OR productid = 46) AND unitprice > 16.00
      WHERE (productname LIKE 'T%') OR (productid = 46 AND unitprice > 16.00)
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a code editor window containing the following SQL query:

```
1 • SELECT productid, productname, supplierid, unitprice
2   FROM products
3   WHERE (productname LIKE 'T%' OR productid = 46) AND unitprice > 16.00
4
5
```

Below the code editor is a "Result Grid" table with four columns: productid, productname, supplierid, and unitprice. The data rows are:

productid	productname	supplierid	unitprice
14	Tofu	6	23.2500
29	Thüringer Rostbratwurst	12	123.7900
62	Tarte au sucre	29	49.3000

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a code editor window containing the same SQL query as the previous screenshot:

```
1 • SELECT productid, productname, unitprice
2   FROM products
3   WHERE (productname LIKE 'T%') OR (productid = 46 AND unitprice > 16.00 )
4
5
```

Below the code editor is a "Result Grid" table with three columns: productid, productname, and unitprice. The data rows are:

productid	productname	unitprice
14	Tofu	23.2500
19	Teatime Chocolate Biscuits	9.2000
23	Tunnbröd	9.0000
29	Thüringer Rostbratwurst	123.7900
54	Tourtière	7.4500
62	Tarte au sucre	49.3000

Waarden in een interval

Waarden in een interval

- **BETWEEN, NOT BETWEEN**

Voorbeeld:

Selecteer de producten (naam en eenheidsprijs) waarvan de eenheidsprijs ligt tussen 10 en 15 euro (grenzen inbegrepen)

```
SELECT productid, unitprice  
FROM products  
WHERE unitprice BETWEEN 10 AND 15
```

	productid	unitprice
▶	3	10.0000
	15	13.0000
	21	10.0000
	25	14.0000
	31	12.5000
	34	14.0000
	42	14.0000
	46	12.0000
	48	12.7500
	58	13.2500
	67	14.0000
	68	12.5000
	70	15.0000
	73	15.0000
	74	10.0000
	77	13.0000

Lijst van waarden

Lijst van waarden

- **IN, NOT IN**

Voorbeeld:

Geef productID, naam en supplierID van de producten die geleverd worden door de suppliers met ID 1, 3 of 5

```
SELECT productid, productname, supplierid  
FROM products  
WHERE supplierid IN (1,3,5)
```

	productid	productname	supplierid
▶	2	Chang	1
	3	Aniseed Syrup	1
	6	Grandma's Boysenberry Spread	3
	7	Uncle Bob's Organic Dried Pears	3
	8	Northwoods Cranberry Sauce	3
	11	Queso Cabrales	5
	12	Queso Manchego La Pastora	5

Testen op onbekende waarden

Testen op onbekende waarden (niet-ingevulde waarden)

– IS NULL, IS NOT NULL

- NULL = niet-ingevulde kolommen (zonder defaultwaarde)
- NULL is **niet** gelijk aan 0 of een blanco string!
- NULL velden worden onderling als gelijk beschouwd (bv. in DISTINCT)
- Rekenkundige uitdrukkingen met NULL resulteren in NULL

Voorbeeld:

Selecteer de leveranciers van wie de regio niet gekend is

```
SELECT companyname, region  
FROM suppliers  
WHERE region IS NULL
```

companyname	region
Exotic Liquids	NULL
Tokyo Traders	NULL
Mayumi's	NULL
Specialty Biscuits, Ltd.	NULL
PB Knäckebröd AB	NULL
Refrescos Americanas LTDA	NULL
Heli Süßwaren GmbH & Co. KG	NULL
Plutzer Lebensmittelgroßmärkte AG	NULL
Nord-Ost-Fisch Handelsgesellschaft m...	NULL
Formaggi Fortini s.r.l.	NULL

Let op met NULL!

```
SELECT companyname, region  
FROM suppliers  
WHERE region <> 'OR'
```

The screenshot shows a MySQL Workbench interface. The SQL pane contains the following code:

```
1 • 1. SELECT companyname, region  
2   FROM suppliers  
3  WHERE region <> 'OR'  
4  
5
```

The Result Grid pane displays the following data:

companyname	region
New Orleans Cajun Delights	LA
Grandma Kelly's Homestead	MI
Cooperativa de Quesos 'Las Cabras'	Asturias
Pavlova, Ltd.	Victoria
New England Seafood Cannery	MA
G'day, Mate	NSW
Ma Maison	Québec
Forêts d'étables	Québec

```
SELECT companyname, region  
FROM suppliers  
WHERE region <> 'OR' OR region IS NULL
```

The screenshot shows a MySQL Workbench interface. The SQL pane contains the following code:

```
1 • 1. SELECT companyname, region  
2   FROM suppliers  
3  WHERE region <> 'OR' OR region IS NULL  
4  
5
```

The Result Grid pane displays the following data:

companyname	region
Exotic Liquids	NULL
New Orleans Cajun Delights	LA
Grandma Kelly's Homestead	MI
Tokyo Traders	NULL
Cooperativa de Quesos 'Las Cabras'	Asturias
Mayumi's	NULL
Pavlova, Ltd.	Victoria
Specialty Biscuits, Ltd.	NULL
PB Knäckebröd AB	NULL
Refrescos Americanas LTDA	NULL
Heli Süßwaren GmbH & Co. KG	NULL

Oefeningen

NR	VNAAM	INIT	FNAAM	AFD	IN DIENST	CODE	NIV	GESL	GEBDAT	SALARIS
10	Christine	I	Haas	A00	650101	66	18	V	330814	52750
20	Michel	L	Theunis	B01	731001	61	18	M	480202	41250
30	Sally	A	Kramer	C01	750405	60	20	V	410511	38250
50	Johan	B	Geysen	E01	490817	58	16	M	250915	40175
60	Irving	F	Steur	D11	730914	55	16	M	450707	32250
70	Eva	D	Pulanski	D21	800930	56	16	V	530526	36170
90	Evelien	W	Hendriks	E11	700815	55	16	V	410515	29750
100	Theo	Q	Spencer	E21	800619	54	14	M	561218	26150
110	Vincent	G	Leman	A00	631205	58	19	M	291105	46500
120	Sean		Connors	A00	580516	58	14	M	421018	29250
130	Danielle	M	Scheire	C01	710728	55	16	V	250915	23800
140	Hilde	A	Nagels	C01	761215	56	18	V	460119	28420
150	Bruno		Adams	D11	720212	55	16	M	470517	25280
160	Els	R	Placke	D11	771011	54	17	V	550412	22250
170	Mats	J	Sierens	D11	780915	54	16	M	510105	24680
180	Marleen	S	Schouters	D11	730707	53	17	V	490221	21340
190	Jan	E	Wauters	D11	740726	53	16	M	520625	20450
200	David		De Bruyn	D11	660303	55	16	M	410529	27740
210	Willem	T	Jansens	D11	790411	25	17	M	530223	18270
220	Jennifer	K	Luyckx	D11	680829	55	18	V	480319	29840

tabel Werknemer

AFDNR	AFDNAAM	MANNR
A00	Computer	10
B01	Planning	20
C01	Informatie	30
D01	Ontwikkelingasc	50
E01	Support	60
D11	Administratie	70
D21	Software	80
E21	Tools	90

tabel Afdeling

**HO
GENT**

Oefeningen

- Geef voornaam en familienaam van werknemers met code 54, die in een willekeurige afdeling werken met uitsluiting van afdeling D11
- Geef nummer, naam en afdelingsnummer van alle werknemers met salaris tussen 15000 en 24000 en niveau tussen 17 en 20
- Geef nummer, naam en opleidingsniveau van alle werknemers met niveau 16, 18 of 20
- Geef nummer, naam van vrouwelijke werknemers waarvan familienaam start met een 'S' of 'T'
- Geef nummer, naam van alle werknemers met onbekende jobcode
- Geef nummer, naam en afdelingsnummer van alle werknemers, waarvan de familienaam start met een P en die in een afdeling werken beginnend met D en als 3^e karakter een 1 hebben

Oplossingen

- Geef voornaam en familienaam van werknemers met code 54, die in een willekeurige afdeling werken met uitsluiting van afdeling D11
 - `Select fnaam, vnaam from werknemer where code=54 and afd <> 'D11'`
- Geef nummer, naam en afdelingsnummer van alle werknemers met salaris tussen 15000 en 24000 en niveau tussen 17 en 20
 - `Select nr, fnaam, afd from werknemer where salaris between 15000 and 24000 and niv between 17 and 20`
- Geef nummer, naam en opleidingsniveau van alle werknemers met niveau 16, 18 of 20
 - `Select nr, fnaam, afd from werknemer where niv in (16,18,20)`

Oplossingen

- Geef nummer, naam van vrouwelijke werknemers waarvan familienaam start met een 'S' of 'T'
 - `Select nr, fnaam from werknemer where gesl='V' and (fnaam like 'S%' or fnaam like 'T%')`
- Geef nummer, naam van alle werknemers met onbekende jobcode
 - `Select nr, fnaam from werknemer where code IS NULL`
- Geef nummer, naam en afdelingsnummer van alle werknemers, waarvan de familienaam start met een P en die in een afdeling werken beginnend met D en als 3^e karakter een 1 hebben
 - `Select nr, fnaam, afd from werknemer where fnaam like 'P%' and afd like 'D_1'`

7.3. Formatteren van resultaten

Formatteren van resultaten

- Formatteren van de resultaten
 - Sorteren data
 - Eliminatie van duplicaten
 - Wijzigen van kolomnaam
 - Berekende resultaatkolommen
- Commentaar
 - /* commentaar */
 - -- commentaar (beperkt zich tot 1 lijn)

ORDER BY clause

ORDER BY = sorteren van de data

- Kan meerdere sorteervelden bevatten
- Een sorteerveld is een kolomnaam of een volgnummer dat overeenkomt met de volgorde van het gegeven achter de SELECT
- Standaard gebeurt het sorteren in stijgende volgorde (ASC)
Een dalende volgorde moet expliciet vermeld worden (DESC)

Voorbeeld:

- Toon een alfabetische lijst van de productnamen

```
SELECT productname  
FROM products  
ORDER BY productname -- of ORDER BY 1
```

Result Grid	Filter Rows:
productname	Alice Mutton
	Aniseed Syrup
	Boston Crab Meat
	Camembert Pierrot
	Carnarvon Tigers
	Chai
	Chang
	Chartreuse verte
	Chef Anton's Cajun Seasoning
	Chef Anton's Gumbo Mix
	Chocolate
	Côte de Blaye
	Escargots de Bourgogne
	Filo Mix
	Flotemysost

ORDER BY clause

Voorbeeld:

- Toon productid, naam, categoryid en eenheidsprijs van de producten gesorteerd op categoryid. Indien binnen één categorie meerdere producten bestaan, dient het product met de hoogste prijs bovenaan te staan

```
SELECT productid, productname, categoryid, unitprice  
FROM products  
ORDER BY categoryid, unitprice DESC
```

productid	productname	categoryid	unitprice
38	Côte de Blaye	1	263.5000
43	Ipoh Coffee	1	46.0000
2	Chang	1	19.0000
1	Chai	1	18.0000
39	Chartreuse verte	1	18.0000
76	Lakkalkööri	1	18.0000
35	Steeleye Stout	1	18.0000
70	Outback Lager	1	15.0000
34	Sasquatch Ale	1	14.0000
67	Laughing Lumberjack Lager	1	14.0000
75	Rhönbräu Klosterbier	1	7.7500
24	Gourmet Fortification	1	4.5000

ALL/DISTINCT

ALL (default) toont alle rijen, ook dubbels

DISTINCT produceert een lijst waarin alle rijen uniek zijn, identieke rijen worden uit het resultaat verwijderd

Voorbeeld:

- Toon alle leveranciers die producten leveren

```
SELECT supplierid  
FROM products  
ORDER BY supplierid
```



supplierid
1
1
2
2
2
2
3
3
3
3
4
4
4
5
5
5
6

```
SELECT DISTINCT supplierid  
FROM products  
ORDER BY supplierid
```



supplierid
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Aliassen

Alias = benoemen van kolommen

- Standaard : kolomtitel = naam van kolom in resultatentabel; berekende kolommen krijgen geen kolomnaam
- Alias = een kolom een andere titel geven via (**AS** keyword en) een string
- De alias kan je ook gebruiken in ORDER BY (niet in WHERE, HAVING, GROUP BY)

Voorbeeld:

- Selecteer ProductID en ProductNaam van alle producten en geef als kolomtitels 'ProductNummer' en 'Naam Product'

```
SELECT productid AS ProductNummer,  
       productname AS 'Naam Product'  
FROM products
```

OF

```
SELECT productid ProductNummer,  
       productname 'Naam Product'  
FROM products
```

Result Grid		Filter Rows:	Export:
	ProductNummer	Naam Product	
▶	17	Alice Mutton	
	3	Aniseed Syrup	
	40	Boston Crab Meat	
	60	Camembert Pierrot	
	18	Carnarvon Tigers	
	1	Chai	
	2	Chang	

Berekeningen

- Berekende resultaatkolommen
 - Via wiskundige operatoren: +, -, /, *
- Voorbeeld:
 - Geef van alle producten de naam en de inventariswaarde

```
SELECT ProductName,  
       Unitprice * UnitsInStock AS InventoryValue  
FROM Products
```

ProductName	InventoryValue
Chai	702.0000
Chang	323.0000
Aniseed Syrup	130.0000
Chef Anton's Cajun Seasoning	1166.0000
Chef Anton's Gumbo Mix	0.0000
Grandma's Boysenberry Spread	3000.0000
Uncle Bob's Organic Dried Pears	450.0000
Northwoods Cranberry Sauce	240.0000
Mishi Kobe Niku	2813.0000
Ikura	961.0000
Queso Cabrales	462.0000
Queso Manchego La Pastora	3268.0000

Functies

Functies

- **String** functies: concat, left, right, length, substring, replace, ...
- **DateTime** functies: dateAdd, dateDiff, day, month, year, ...
 - now(): retourneert huidige datum en tijd in datetime formaat: JJJJ-MM-DD UU:MM:SS
 - curdate() → huidige datum in JJJJ-MM-DD
- **Rekenkundige** functies: round, floor, ceil, cos, sin, ...
- **Aggregate** functies: avg, sum, min, max ...
- **IFNULL(kolom, x)** : vervangt gevonden NULL-waardes door x

Functies

IFNULL() voorbeeld:

- Toon orders hun OrderID, CustomerID, ShipPostalCode (met aanduiding “Niet ingevuld” indien leeg)

```
SELECT OrderID, CustomerID,  
       IFNULL(ShipPostalCode,'Niet ingevuld')  
             AS PostCode  
FROM orders
```

OrderID	CustomerID	PostCode
10302	SUPRD	B-6000
10303	GODOS	41101
10304	TORTU	05033
10305	OLDWO	99508
10306	ROMEY	28001
10307	LONEP	97219
10308	ANATR	05021
10309	HUNGO	Niet ingevuld
10310	THEBI	97201
10311	DUMON	44000

Dataconversie

- Implicit voor sommige omzettingen
 - Voorbeeld: UnitPrice * 0,5 → UnitPrice (money) wordt automatisch naar decimal geconverteerd
- Explicit met:
 - **CAST** (<value expression> AS <data type>)
 - PRINT CAST(-25.25 AS INTEGER) → -25
 - SELECT CAST(38,8 AS CHAR) → '38,8'
 - **CONVERT** (<expression>, type)
 - CONVERT(CURDATE(), CHAR) → '2016-11-24'

Strings

- Stringoperator: concatenatie

```
SELECT CONCAT(productid, ' ', productname)
      AS Product
  FROM Products
```

Product
17 , Alice Mutton
3 , Aniseed Syrup
40 , Boston Crab Meat
60 , Camembert Pierrot
18 , Carnarvon Tigers
1 , Chai
2 , Chang
39 , Chartreuse verte
4 , Chef Anton's Cajun Seasoning
5 , Chef Anton's Gumbo Mix
48 , Chocolade
38 , Côte de Blaye
58 , Escargots de Bourgogne
52 , Filo Mix
71 , Flotemysost

- Gebruik maken van tekst (literals)

```
SELECT ProductName, '$', Unitprice
  FROM Products
```

ProductName	\$	Unitprice
Chai	\$	18.0000
Chang	\$	19.0000
Aniseed Syrup	\$	10.0000
Chef Anton's Cajun Seasoning	\$	22.0000
Chef Anton's Gumbo Mix	\$	21.3500
Grandma's Boysenberry Spread	\$	25.0000
Uncle Bob's Organic Dried Pears	\$	30.0000
Northwoods Cranberry Sauce	\$	40.0000
Mishi Kobe Niku	\$	97.0000
Ikura	\$	31.0000
Queso Cabrales	\$	21.0000
Queso Manchego La Pastora	\$	38.0000
Konbu	\$	6.0000

CASE functie

- Voorbeeld voor verschepingskost:

```
SELECT OrderID, Freight,  
CASE  
    WHEN Freight IS NULL THEN 'Kost onbekend'  
    WHEN Freight = 0 THEN 'Gratis verscheping'  
    WHEN Freight < 20 THEN 'Lage verschepingskost'  
    WHEN Freight < 100 THEN 'Gemiddelde verschepingskost'  
    ELSE 'Hoge verschepingskost'  
END AS 'Verschepingskost'  
FROM orders
```

OrderID	Freight	Verschepingskost
10248	32.3800	Gemiddelde verschepingskost
10249	11.6100	Lage verschepingskost
10250	65.8300	Gemiddelde verschepingskost
10251	41.3400	Gemiddelde verschepingskost
10252	51.3000	Gemiddelde verschepingskost
10253	58.1700	Gemiddelde verschepingskost
10254	22.9800	Gemiddelde verschepingskost
10255	148.3300	Hoge verschepingskost
10256	13.9700	Lage verschepingskost

