

# Cursus Databanken

SELECT-instructie

**WHERE**, vergelijkingsoperatoren  
en logische operatoren

Docenten:

Damien Decorte

Wim Goedertier

Luc Vervoort

Tom Vande Wiele

Eric Juchtmans



**HO  
GENT**

# Herinner je volgende tips

Bij het begin van een nieuwe sessie  
selecteer je de databank die je wilt gebruiken:

`USE db_foundations;`  
of `USE bib;`

Bewaar je SQL-instructies op het einde van de les,  
als studiemateriaal voor later (gebruik hiervoor **\history**)

# De **WHERE**-clausule

We willen niet ALLE rijen selecteren, maar slechts **BEPAAALDE rijen**

```
SELECT * FROM employees  
WHERE department_id = 60;
```

## OPDRACHT:

Maak een resultatenlijst met (departements)**naam** en (departements)**manager\_id**  
van alle departementen met **location\_id 1700** (4 rows returned)

## Merk op:

De location\_id (1700) hoeft hierbij **niet** per se  
zichtbaar te zijn in de resultatenlijst.

	department NAME	Manager id
►	Administration	200
	Executive	100
	Accounting	205
	Contracting	<small>HULL</small>

# De WHERE-clausule: **vergelijkings-operatoren**

De vergelijkings-operatoren zijn (in deze volgorde van voorrang):

=, <, <=, >, >=, <> of !=

IS (NOT) NULL, (NOT) LIKE, (NOT) IN (... , ...)

(NOT) BETWEEN...AND...

Ze worden één voor één behandeld in de volgende slides.

# De WHERE-clausule: operator =

Voorbeeld met **NUMERIEK** veld:

```
SELECT * FROM employees WHERE department_id = 60 ;
```

OPDRACHT met **TEKST**-veld: maak een resultatenlijst met alle gegevens van alle medewerkers waarvan de job 'IT\_PROG' is. (3 rows returned)

**Tekst moet *steeds* tussen enkelvoudige aanhalingstekens:** bvb. 'IT\_PROG'

OPDRACHT met **TIJD**-veld: maak een resultatenlijst met alle gegevens van alle medewerkers die aangeworven zijn op 16 november 1999. (Antw: Kevin Mourgous)

**Datums** moeten steeds in '**yyyy-mm-dd**' (ISO-8601 én single quotes). Bvb. '**2021-12-31**'

(Datum-formaat bij Oracle: '31-DEC-2021')

# De WHERE-clausule: operator $\neq$ (of $\neq$ )

OPDRACHT: toon alle gegevens van alle medewerkers  
(20 rows returned)

OPDRACHT: toon alle gegevens van alle medewerkers die in departement 60 zitten  
(3 rows returned)

OPDRACHT: toon alle gegevens van alle medewerkers die **NIET** in departement 60 zitten. Gebruik hiervoor  $\neq$  of  $\neq$   
(16 rows returned)

De rekening klopt niet. Waar is de twintigste rij?

# De WHERE-clausule: operator **IS NULL**

OPDRACHT: toon alle gegevens van alle medewerkers waarvan het departement niet ingevuld is (gebruik hiervoor **IS NULL**)  
(1 row returned)

De NULL-value betekent:

“Het is niet ingevuld” of “We weten het niet”  
unassigned, unavailable, unknown, inapplicable  
NULL is ***niet*** hetzelfde als het cijfer nul (0)  
NULL is ook ***niet*** hetzelfde als een lege string ("")

# De WHERE-clausule: operatoren < en >

OPDRACHT: toon alle gegevens van alle medewerkers die in departement 60 zitten  
(3 rows returned)

OPDRACHT: toon alle medewerkers met een department\_id groter dan 60  
(8 rows returned)

OPDRACHT: toon alle medewerkers met een department\_id kleiner dan 60  
(8 rows returned)

$3 + 8 + 8 = 19$  rijen. Waar is de 20ste rij?

NULL is niet gelijk, noch kleiner, noch groter dan 60 !!

Een voorwaarde waarin met NULL vergeleken wordt, zal ***nooit*** voldaan zijn.



# De WHERE-clausule: operatoren $\leq$ en $\geq$

## OPDRACHT:

Toon alle gegevens van alle medewerkers  
waarvan het salaris **minstens 9000** bedraagt (8 rows returned)

## OPDRACHT:

Toon alle gegevens van alle medewerkers  
waarvan het salaris **hoogstens 9000** bedraagt (13 rows returned)

De WHERE-clausule:

operatoren < , > , <= , >= in **tekst-** en **tijds**velden

OPDRACHT: toon alle gegevens van alle medewerkers waarvan de voornaam alfabetisch na "Peter" komt. (5 rows returned)

OPDRACHT: test wat je krijgt bij, en vergelijk met het vorige

```
SELECT * FROM employees  
WHERE first_name > 'p';
```

 (7 rows returned)

OPDRACHT: toon alle gegevens van alle medewerkers die aangeworven zijn sedert 1999-11-16 (2 rows returned)

# De WHERE-clausule operatoren **IS NULL** en **IS NOT NULL**

OPDRACHT: Toon alle gegevens van alle medewerkers die  
**geen manager** hebben

(of waarvan we **niet weten** wie de manager is)

(1 row returned)

OPDRACHT: Toon alle gegevens van alle medewerkers  
die **WEL** een manager hebben

(19 rows returned)

# De WHERE-clausule operatoren **LIKE** en **NOT LIKE**

Er zijn 2 WILDCARD characters:

UNDERSCORE "\_" : precies één willekeurig karakter

PERCENT "%" : een willekeurig aantal willekeurige karakters  
("willekeurig aantal" kan ook nul of één zijn)

OPDRACHT: toon alle gegevens van alle medewerkers

waarvan voornaam **begint met "k"** (2 rows returned)  
*hoofdlettergevoelig?*

OPDRACHT: toon alle gegevens van alle medewerkers

waarvan voornaam **NIET begint met "k"** (18 rows returned)

# De WHERE-clausule operatoren **LIKE** en **NOT LIKE**

OPDRACHT: toon alle gegevens van alle medewerkers  
waarvan de voornaam **eindigt op "r"**

(3 rows returned)

OPDRACHT: toon alle gegevens van alle medewerkers  
waarvan de voornaam **een "r" bevat**

(8 rows returned)

OPDRACHT: toon alle gegevens van alle medewerkers  
waarvan de voornaam **een "o" heeft op de tweede positie**

(1 row returned)

OPDRACHT: toon alle gegevens van alle medewerkers  
waarvan de voornaam **een "e" heeft op de voorlaatste positie**

(8 rows returned)

# De WHERE-clausule operatoren **LIKE** en **NOT LIKE**

**LIKE** werkt (enigszins verrassend) ook op een **TIJDSVELD**

OPDRACHT: toon alle gegevens van alle medewerkers  
die aangeworven zijn in 1997 (2 rows returned)  
(datumnotatie: 'yyyy-mm-dd')

**LIKE** werkt (enigszins verrassend) ook op een **NUMERIEK VELD**

OPDRACHT: toon alle gegevens van alle medewerkers  
waarvan manager\_id een "2" bevat (7 rows returned)

# De WHERE-clausule operatoren **LIKE** en **NOT LIKE**

Om UNDERSCORE of PERCENT op te nemen als karakter (niet als wildcard) bij (NOT) LIKE:

→ laten voorafgaan door het *escape karakter* BACKSLASH: “\\_” en “\%”

Om BACKSLASH op te nemen als karakter (niet als escape karakter) bij (NOT) LIKE:

→ laten voorafgaan door het *escape karakter* BACKSLASH: “\\”

OPDRACHT: toon alle medewerkers waarvan de job\_id "A\_" bevat

(4 rows return)

OPDRACHT: toon alle medewerkers waarvan last\_name een BACKSLASH bevat

(0 rows returned)

# De WHERE-clausule operatoren **IN** en **NOT IN**

Voorbeeld: toon alle medewerkers in de departementen 110, 10 en 80.

```
select * from employees  
where department_id IN (110,10,80);    (6 rows returned)
```

OPDRACHT: toon alle gegevens van alle medewerkers  
die IT\_PROG of AD\_VP zijn. (5 rows returned)

OPDRACHT: toon gegevens van alle medewerkers  
die *géén* IT\_PROG en *géén* AD\_VP zijn. (15 rows returned)



# De WHERE-clausule operatoren (NOT) BETWEEN ... AND ...

- OPDRACHT: toon alle gegevens van alle medewerkers  
-- die aangeworven zijn tussen 1998-03-15 en 1998-04-01 . -- 2 rows returned
- BELANGRIJKE VASTSTELLING:  
-- '1998-03-15' voldoet ook aan **BETWEEN** '1998-03-15' **AND** '1998-04-01'  
-- INCLUSIEF de grenzen, dus !!  
-- Operator BETWEEN...AND mag je dus niet interpreteren als "TUSSEN" (exclusief de grenzen) !!
- OPDRACHT: toon alle gegevens van alle medewerkers met  
-- een department\_id tussen 20 en 90, met de grenzen **NIET** inbegrepen -- 11 rows returned
- OPDRACHT: toon alle gegevens van alle medewerkers met  
-- een department\_id groter dan 90 of kleiner dan 20 -- 3 rows returned

# Volgorde van clauses

SELECT ... FROM ... WHERE ... ORDER BY ... ;

-- De WHERE-clausule komt steeds vóór de ORDER BY-clausule

-- OPDRACHT: Maak een resultatenlijst met alle medewerkers

-- die meer dan 10000 salaris hebben,

-- gerangschikt op anciënniteit.

# De WHERE-clausule de logische operatoren

/\*

de logische operatoren zijn (in deze volgorde van voorrang):

NOT

AND

OR

\*/

# De WHERE-clausule de logische operator **NOT**

-- VOORBEELD:

```
SELECT      first_name, salary
FROM        employees
WHERE       department_id = 60 ;
```

```
SELECT      first_name, salary
FROM        employees
WHERE NOT ( department_id = 60 ) ;
```

```
SELECT      first_name, salary
FROM        employees
WHERE       department_id <> 60 ;
```

-- geeft hetzelfde resultaat, maar op 2 verschillende manieren

# De WHERE-clausule

## de logische operator **NOT**

-- VOORBEELD:

```
SELECT      *
FROM        employees
WHERE       job_id IN ('IT_PROG','AD_VP');      -- 5 rows returned
```

```
SELECT      *
FROM        employees
WHERE       NOT ( job_id IN ('IT_PROG','AD_VP') );      -- 15 rows returned
                                                    -- dit is de operator "NOT", gecombineerd met de operator "IN"
```

```
SELECT      *
FROM        employees
WHERE       job_id NOT IN ('IT_PROG','AD_VP');      -- 15 rows returned
                                                    -- dit is operator "NOT IN"; dit is niet operator "NOT" en operator "IN"
                                                    -- maar leidt wel tot hetzelfde resultaat
```

# De WHERE-clausule de logische operator **OR**

-- VOORBEELD:

SELECT \*

FROM employees

WHERE salary >= 10000 ;

-- 7 rows returned

NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	BONUS
4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	90	NULL
4568	1989-09-21	AD_VP	17000.00	NULL	100	90	NULL
4569	1993-01-13	AD_VP	17000.00	NULL	100	90	NULL
344.429018	2000-01-29	SA_MAN	10500.00	0.20	100	80	1500
544.429267	1996-05-11	SA_REP	11000.00	0.30	149	80	1700
544.429265	1998-03-24	SA_REP	8600.00	0.20	149	80	1250
5555	1996-02-17	MK_MAN	13000.00	NULL	100	20	NULL
3080	1994-06-07	AC_MGR	12000.00	NULL	101	110	NULL
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT \*

FROM employees

WHERE bonus IS NOT NULL ;

-- 3 rows returned

SELECT \*

FROM employees

WHERE salary >= 10000

**OR** bonus IS NOT NULL ;

-- 8 rows returned

# De WHERE-clausule de logische operator **OR**

-- OPDRACHT:

- maak een resultatenlijst met
- alle gegevens van alle medewerkers
- waarvan job\_id 'IT\_PROG', 'ST\_CLERK' of 'ST\_MAN' is.
- 1. Doe dit a.d.h.v. de operator IN.
- 2. Doe dit **zonder** de operator IN. Gebruik de operator “OR”.

# De WHERE-clausule de logische operator **AND**

-- VOORBEELD:

```
SELECT *  
FROM employees  
WHERE phone_number LIKE '650%';  
-- 5 rows returned
```

```
SELECT *  
FROM employees  
WHERE hire_date < '1996-01-01';  
-- 9 rows returned
```

```
SELECT *  
FROM employees  
WHERE phone_number LIKE '650%'  
AND hire_date < '1996-01-01';  
-- 1 row returned
```



# De WHERE-clausule

## de logische operator **AND**

-- OPDRACHT:

-- maak een resultatenlijst met

-- alle gegevens van alle medewerkers met

-- een departement\_id tussen 20 en 90 . -- 11 rows returned

-- (Opgelet: met “tussen” bedoelen we normaal gezien: “exclusief de genzen”)

-- 1. Doe dit a.d.h.v. de operator BETWEEN...AND...

-- 2. Doe dit **zonder** de operator BETWEEN...AND...

# Interpretatie van OR vs. AND

- Hoe omschrijf je de resultatenlijsten
- die je bekomt met volgende instructies?
- De lijst van alle medewerkers die ...

```
SELECT employee_id, last_name, job_id, salary  
FROM employees  
WHERE salary >= 10000 AND job_id LIKE '%MAN%' ;
```

```
SELECT employee_id, last_name, job_id, salary  
FROM employees  
WHERE salary >= 10000 OR job_id LIKE '%MAN%' ;
```

# Vorrang van uitvoering: AND heeft vorrang op OR

Bij bewerkingen heeft maal (\*) vorrang op plus (+): “**2+3\*4**” idem als “**2+(3\*4)**”

Bij logica heeft AND vorrang op OR: “**a OR b AND c**” idem als “**a OR (b AND c)**”

## OPDRACHT:

maak, zonder gebruik te maken van de IN-operator, een lijst met alle medewerkers met een salaris dat hoger is dan 8000 en die als job\_id “SA\_REP” of “IT\_PROG” hebben.

(3 rows returned)

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	BONUS
▶	103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9000.00	NULL	102	60	NULL
	174	Ellen	Abel	EABEL	011.44.1644.429267	1996-05-11	SA_REP	11000.00	0.30	149	80	1700
	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	1998-03-24	SA_REP	8600.00	0.20	149	80	1250
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

# Vorrang van uitvoering: AND heeft vorrang op OR

Hoe omschrijf je de resultatenlijsten die je krijgt met volgende instructies?

```
SELECT last_name, job_id, salary  
FROM employees  
WHERE job_id = 'SA_REP' OR job_id = 'AD_PRES' AND salary > 15000;
```

```
SELECT last_name, job_id, salary  
FROM employees  
WHERE ( job_id = 'SA_REP' OR job_id = 'AD_PRES' ) AND salary > 15000;
```