BIOINFORMATICA

INFORMATIESYSTEMEN MET PYTHON_

OPERATOREN EN FUNCTIES

HAN_UNIVERSITY
OF APPLIED SCIENCES

STUDIEWIJZER

Les	Onderwerp	Theorie
1	Introductie tot Informatie Systemen	H1 Introduction to Database Development H2 Entity Relationships H3 Complex Relationships H4 Logical Database Design
2	Database ontwerp Constraints en normaliseren	H5 Normalisation H6 Introduction to Oracle SQL H7 Foreign Keys
3	SQL, operatoren en functies	H8 Selecting data from a Table
4	Rijen ophalen uit meerdere tabellen	H9 Selecting data from multiple tables
5	Subquery's en groepfuncties	H10 Subqueries and Group Functions
6	SQL Embedden in Python	
7	Python en SQL: mogelijkheden en risico's	

DE 5 TAALGROEPEN

Groep	Doel	Statements
DRL	Data Retrieval	**`select`**
DML	Data Manipulation	**`update`** **`delete`** **`insert`**
DDL	Data Definition	**`create`** **`alter`** **`drop`**
TCL	Transaction Control	**`commit`** **`rollback`**
ACL	Access Control	**`grant`** **`revoke`**

AGENDA

- Operatoren
  - Logische operatoren
  - Lijsten
  - Like operator
  - Bereiken
  - Is null operatoren
- Functies

HET GENERIEKE SELECT STATEMENT

```
select <kolomnamen|*>
from   <tabel>
where  <conditie>
order by <kolomnaam>
```

Een conditie is een voorwaarde waaraan een rij moet voldoen om getoond te worden.

Bijvoorbeeld: naam = ‘Piet’

VERGELIJKINGS OPERATOREN

Operator	Betekenis
=	Gelijk aan
>	Groter dan
<	Kleiner dan
>=	Groter of gelijk aan
<=	Kleiner of gelijk aan
!= en <>	Niet gelijk aan

VOORBEELDEN VAN CONDITIES

```
where geb_datum > ’01-jan-1999’

where voornaam = ‘Piet’

where student_nr > 1000

where student_nr = 1000
```

LOGISCHE OPERATOREN

Logische operatoren maken het mogelijk om meerdere condities te combineren

Bijvoorbeeld toon alle studenten die in Nijmegen wonen en na 1999 zijn geboren

```
where woonplaats = ‘Nijmegen’
   and geb_datum > ‘01-jan-1999’
```

VOORBEELD

Als je wilt weten welke studenten uit Nijmegen, Arnhem of Oss komen dan gebruik je de volgende query

```
select *
from   student
where woonplaats = 'Nijmegen’
   or woonplaats = ‘Arnhem'
   or woonplaats = ‘Oss'
```

OVERZICHT LOGISCHE OPERATOREN

Operator	Betekenis
and	Beide expressies moeten waar zijn
or	Een van beide expressies moet waar zijn
not	De expressie moet niet waar zijn

AGENDA

- Operatoren
  - Logische operatoren
  - Lijsten
  - Like operator
  - Bereiken
  - Is null operatoren
- Functies

IN OPERATOR

Met de IN operator kun je een conditie opgeven waarbij je wilt dat een waarde voorkomt in een groep van waardes

Bijvoorbeeld toon alle studenten die in Nijmegen, Arnhem of Oss wonen

VOORBEELD

Als je wilt weten welke studenten uit Nijmegen, Arnhem of Oss komen dan gebruik je de volgende query

Dit kan korter

```
select *
from   student
where woonplaats = 'Nijmegen'
   or woonplaats = ‘Arnhem'
   or woonplaats = ‘Oss'
```

VOORBEELD

Als je wilt weten welke studenten uit Nijmegen, Arnhem of Oss komen dan gebruik je de volgende query

Dit kan korter

```
select *
from   student
where  woonplaats in ('Nijmegen’,’Arnhem’, ‘Oss')
```

AGENDA

- Operatoren
  - Logische operatoren
  - Lijsten
  - Like operator
  - Bereiken
  - Is null operatoren
- Functies

LIKE OPERATOR

Net als op Linux en Windows is het mogelijk wildcards te gebruiken

Dit kan alleen in combinatie met de operator like

De wildcards _ en % hebben dezelfde betekenis in SQL als ? en * op Linux

WILDCARDS IN SQL

SQL Wildcard	Betekenis	Op Linux
_	Exact 1 willekeurig teken	?
%	0, 1 of meer willekeurige tekens	*

OPDRACHT

Toon alle studenten met een ‘c’ in hun voornaam

LIKE OPERATOR

Als je wilt weten welke studenten een c in de naam hebben kun je de like operator gebruiken

```
select * from student
where voornaam like '%c%'
```

AGENDA

- Operatoren
  - Logische operatoren
  - Lijsten
  - Like operator
  - Bereiken
  - Is null operatoren
- Functies

BETWEEN

Welke studenten zijn geboren tussen 1 januari 2000 en 31 januari 2001

```
select *
from student
where geb_datum between '2000-01-01' and ‘2001-01-31'
```

AGENDA

- Operatoren
  - Logische operatoren
  - Lijsten
  - Like operator
  - Bereiken
  - Is null operatoren
- Functies

NULL

Om te toetsen of iets null is gebruik je niet `= null` maar `is null`

WAT KRIJG JE TERUG BIJ DEZE QUERY?

```
select *
from   student
where  null = null
```

NULL

Met de where clausule toon je rijen waar de conditie (voorwaarde) klopt

=> Kun je er ‘ja’ (of TRUE) op antwoorden?

Met ‘=’ vergelijk je 2 expressies (uitdrukkingen)

Je kunt geen ja antwoorden op de vraag:

- Bij welke rijen is niets gelijk aan niets?

WAT KRIJG JE TERUG BIJ DEZE QUERY?

```
select *
from   student
where  null is null
```

NULL

Met de where clausule toon je rijen waar de conditie klopt

Kun je er ‘ja’ (of TRUE) op antwoorden?

Met ‘IS NULL’ check je of een attribuut geen waarde heeft

Je kunt ja antwoorden op de vraag:

- Heeft geen attribuut geen waarde?

WAT KRIJG JE TERUG BIJ DEZE QUERY?

```
select *
from student
where voornaam like '%a%'
and voornaam like '%e%'
or 1 = 1
```

EXPRESSIES

Een expressie (uitdrukking) kun je opnemen in de select clausule

Bijvoorbeeld

```
select student_nr/2
,      voornaam
,      'hello world'
from   student
```

EXPRESSIES

Kolommen (tekst) kun je aan elkaar plakken met de functie concat

```
select concat(voornaam,' ',achternaam)
from student
```

EXPRESSIES

Maar wat als we er per ongeluk een null aan vast plakken?

```
select concat(voornaam,tussenvoegsels,achternaam)
from student
```

EXPRESSIES

Expressies hoeven niet altijd over de inhoud van kolommen te gaan

```
select 1+1
from   studenten
```

SELECT ZONDER FROM

In MySQL kun je de from clausule weglaten tenzij je kolommen/attributen gebruikt die refereren aan tabellen

```
select 1+1;

select curdate();

select now();
```

SAMENVATTING

Vergelijkingsoperatoren: =, <, >, !=

Logische operatoren: and, or, not

Vergelijkingsoperatoren: in, between, like, is null

Expressies ook in select clausule

AGENDA

- Operatoren
- Functies
  - Tekst functies
  - Numerieke functies
  - Datum functies
  - Algemene functies

FUNCTIE

Functies in een SQL zijn vergelijkbaar met functies in Python: je stopt waardes erin en er komt iets uit

VOORBEELD FUNCTIE

upper (‘bio-informatica’)

De functie upper zal alle tekst retourneren in uppercase (hoofdletters)

```
select upper(‘bio-informatica’);
```

In MySQL heb je niet perse een from clausule nodig. In Oracle bijvoorbeeld wel, daar heb je een dummy tabel nodig.

QUERY

Hoe toon ik alle gegevens van de oudste student?

AGENDA

- Operatoren
- Functies
  - Tekst functies
  - Numerieke functies
  - Datum functies
  - Algemene functies

TEKSTFUNCTIES

Functie	Uitwerking
instr	Zoek naar het voorkomen van een substring
length	Lengte bepalen van een string
lower	Zet alles om naar lowercase (kleine letters)
upper	Zet alles om naar uppercase (hoofdletters)
ltrim	Haal tekens (standaard spatie) weg links
rtrim	Haal tekens (standaard spatie) weg rechts
substr	Haal een substring uit een string
concat	Plakken van meerdere strings aanelkaar

VOORBEELD TEKSTFUNCTIE

```
select upper  (voornaam)
,      length (voornaam)
,      ltrim  (voornaam)
,      substr (voornaam,2,5)
from student
```

VRAAG

Hoe toon ik alle studenten die een a of A in de voornaam hebben. Zonder de operator like te gebruiken en zonder een and of or te gebruiken?

TEKSTFUNCTIES

Functie	Uitwerking
like	Eenvoudige tekstpatronen met wildcards % en _
sounds_like	Zoekopdrachten op uitspraak
regexp	Zoekopdrachten op basis van regular expressions

REGEXP

```
select *
from student
where voornaam regexp '.y.*’
```

AGENDA

- Operatoren
- Functies
  - Tekst functies
  - Numerieke functies
  - Datum functies
  - Algemene functies

NUMERIEKE FUNCTIES

Functie	Werking
mod	De modulus functie
log	Berekening van het logaritme
ln	Natuurlijk logaritme
round	Afronden van een getal
ceil	Afroning naar boven
floor	Afronding naar beneden
sqrt	Wortel

VRAAG

Hoe toon ik de wortel van het studentnummer delen door 9 naar beneden toe afgerond?

VRAAG

Wat is het gen met de meeste nucleotiden?

AGENDA

- Operatoren
- Functies
  - Tekst functies
  - Numerieke functies
  - Datum functies
  - Algemene functies

REKENEN MET DATUMS

De database is altijd in staat om tijdseenheden op te tellen bij een datum

Bijvoorbeeld:

```
SELECT DATE_ADD(curdate(), INTERVAL 10 DAY);
```

DATUM FUNCTIES

Functie	Werking
curdate	Retourneert de actuele datum
curtime	Retourneert de actuele tijd
now	Retourneert datum en tijd
period_diff	Verschil in maanden tussen twee data
datediff	Datum min andere datum (dagen)
dayname	Naam van de weekdag
day	Dag van de maand
week	Weeknummer

VRAAG

Schrijf de query om iedereen te tonen die op een zondag geboren is

AGENDA

- Operatoren
- Functies
  - Tekst functies
  - Numerieke functies
  - Datum functies
  - Algemene functies

ALGEMENE FUNCTIES

Algemene functies zijn functies die op alle datatypes werken

Voorbeelden van algemene functies zijn:

- case
- Ifnull
- nullif
- if

ALGEMENE FUNCTIES

Functie	Werking
case	Zet een waarde om in een andere waarde
ifnull	Vervangt een null door een andere waarde
nullif	Vervangt een waarde door null als…
if	If/else constructie

CASE

```
select voornaam
,      case klas_id when 1 then 'BI1a'
                    when 2 then 'BI1b'
                    when 3 then 'BI1c'
                    when 4 then 'BI1d'
       end
from student
```

IFNULL

```
select voornaam
     , ifnull(tussenvoegsels,'nvt')
     , achternaam
from student
```

IF

Controleer of het email-adres een @ en . bevat zoniet geef een error

```
select voornaam,
       email,
       if (email regexp '.{2,9}@.{2,9}[.].{2,5}' ,email,'error')
from student;
```

SAMENVATTING

- Naast de **groepsfuncties**: count(), min(), max()
  - Werken op groep rijen
  - Vooral in WHERE clausule
    - Geeft in select maar 1 rij terug!

- kennen we ook **rijfuncties**
  - werken op rij niveau
  - ook (vooral) in SELECT clausule

SAMENVATTING

- De functies die op rij niveau werken zijn in te delen naar de input die ze mee krijgen:
  - Tekst => upper(), substr()
  - Getallen => sqrt(), round()
  - Datum => datediff(), curdate()
  - Algmeen => ifnull(), if()

VERANTWOORDING

In deze uitgave is géén auteursrechtelijk beschermd werk opgenomen

Alle teksten © Martijn van der Bruggen/HAN tenzij expliciet externe bronnen zijn aangegeven

Screenshots op basis van eigen werk auteur en/of vernoemde sites

Eventuele images zijn opgenomen met vermelding van bron