

# Herhaling SQL

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0  
Unported Licentie

# SQL

SQL=Structured Query Language

ISO, no dialects

# Lezen van gegevens

SELECT

# Muteren van gegevens

INSERT INTO

UPDATE

DELETE FROM

# Bepalen structuur voor gegevens

CREATE TABLE

GRANT

# Volgorde?

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

# Voorbeeld

```
SELECT      klantnr, MAX(reisnr) AS  
                                         laatste_reis  
FROM        deelnames  
WHERE       reisnr > 10  
GROUP BY   klantnr  
HAVING      COUNT(reisnr) > 1  
ORDER BY   klantnr DESC;
```

# Funcities

EXTRACT ( )

CAST ( )

TO\_CHAR ( )

ROUND ( )

...

productspecifieke SQL?!

<http://www.postgresql.org/docs/current/interactive/functions.html>



# Alias

```
SELECT  voornaam AS naam
```

```
SELECT  *  
FROM    spelers AS s
```

```
SELECT  *  
FROM    spelers s
```

# CASE

# CASE

```
WHEN satellietvan IS NULL THEN 'Zon'
```

```
WHEN satellietvan = 'Zon' THEN
                                'Planeet'
```

```
ELSE 'Satelliet'
```

END

# CASE

```
SELECT
  CASE
    WHEN satellietvan IS NULL THEN 'Zon'
    WHEN satellietvan = 'Zon' THEN 'Planeet'
    ELSE 'Satelliet'
  END AS naamplaneet
FROM hemellichaam
WHERE
  CASE
    WHEN satellietvan IS NULL THEN 'Zon'
    WHEN satellietvan = 'Zon' THEN 'Planeet'
    ELSE 'Satelliet'
  END
LIKE '%et'
```

# Joins

```
SELECT  *  
FROM    reizen INNER JOIN deelnames  
        ON reizen.reisnr = deelnames.reisnr
```

# LEFT OUTER JOIN

```
SELECT  hemelobjecten.objectnaam,  
        bezoeken.reisnr  
  
FROM    hemelobjecten LEFT OUTER JOIN  
        bezoeken  
  
        USING (objectnaam)
```

# LEFT OUTER JOIN

```
SELECT    s.spelersnr, w.wedstrijdnr
FROM      spelers s LEFT OUTER JOIN
          wedstrijden w
          ON s.spelersnr = w.spelersnr
          AND w.gewonnen > w.verloren

/* condition in de join */
```

# LEFT OUTER JOIN

```
/* versus: */
```

```
SELECT  s.spelersnr, w.wedstrijdnr  
FROM    spelers s LEFT OUTER JOIN wedstrijden w  
        ON s.spelersnr = w.spelersnr  
WHERE   w.gewonnen > w.verloren
```

```
/* condition in the where */
```

# LEFT OUTER JOIN

```
SELECT  s.spelersnr
FROM    spelers s LEFT OUTER JOIN
        wedstrijden w
        ON s.spelersnr = w.spelersnr
        AND w.gewonnen > w.verloren
GROUP BY s.spelersnr
```



# Selecties op groeperingen

GROUP BY

HAVING

# Aggregatiefuncties

COUNT ( )

MAX ( )

MIN ( )

SUM ( )

AVG ( )

STDDEV ( )

<http://www.postgresql.org/docs/current/interactive/functions-aggregate.html>

# Oefening 1

Toon alle spelers en het aantal wedstrijden dat ze hebben gespeeld.

# Manier 1?

```
SELECT      s.spelersnr, COUNT(*) AS  
           aantal  
  
FROM        spelers s INNER JOIN  
           wedstrijden w  
  
           ON s.spelersnr = w.spelersnr  
  
GROUP BY s.spelersnr;
```

# Manier 2?

```
SELECT      s.spelersnr, COUNT(*) AS  
            aantal  
  
FROM        spelers s LEFT OUTER JOIN  
            wedstrijden w  
            ON s.spelersnr = w.spelersnr  
  
GROUP BY   s.spelersnr;
```

# Manier 3?

```
SELECT    s.spelersnr, COUNT (w.wedstrijdnr)
          AS aantal
FROM      spelers s LEFT OUTER JOIN
          wedstrijden w
          ON s.spelersnr = w.spelersnr
GROUP BY  s.spelersnr;
```

# Toevoegen

```
INSERT INTO bezoeken (  
    reisnr,  
    volgnr,  
    objectnaam,  
    verblijfsduur  
) VALUES (  
    34,  
    4,  
    'maan',  
    2  
)
```

# Oefening 2

Hoeveel spelers hebben een wedstrijd gespeeld, geef het totaal.



# Manier 1

```
SELECT  COUNT(DISTINCT s.spelersnr)
        AS aantal
FROM    spelers s INNER JOIN
        wedstrijden w
        ON s.spelersnr = w.spelersnr;
```

# Bijwerken

```
UPDATE  reizen
SET     vertrekdatum ='2030-12-31',
        reisduur = 30,
        prijs = 1.23
WHERE   reisnr = 33;
```

# Verwijderen

```
DELETE  
FROM hemelobjecten  
WHERE objectnaam = 'Pluto';  
  
/* zonder where clause?;) */
```

# Aanmaken

```
CREATE TABLE bezoeken (  
    reisinr          numeric(4,0)          NOT NULL,  
    objectnaam       character varying(10) NOT NULL,  
    volgnummer       numeric(2,0)          NOT NULL,  
    verblijfsduur    numeric(4,0)          NOT NULL,  
    CONSTRAINT bezoeken_pkey PRIMARY KEY (reisinr, volgnummer),  
    CONSTRAINT bezoeken_objectnaam_fkey  
        FOREIGN KEY (objectnaam) REFERENCES hemelobjecten (objectnaam),  
    CONSTRAINT bezoeken_reisinr_fkey  
        FOREIGN KEY (reisinr) REFERENCES reizen (reisinr)  
);  
  
-- datatypes zijn productspecifieke SQL?  
-- http://www.postgresql.org/docs/current/interactive/datatype.html
```

# Rechten geven

```
GRANT    SELECT  
ON       TABLE bezoeken  
TO       student;
```

-- Voldoende?

# Rechten geven

```
GRANT  USAGE
ON     SCHEMA ruimtereizen
TO     public;
```

## Referenties:

- \* Slides herhaling sql 2012-13, K. Beheydt
- \* SQL Leerboek, R. Van der lans