



Course

Database Implementation

Information System Engineering

Onderwerpen

- **Introductie**
- **Correlated subqueries**
- **EXISTS operator**
- **UNION / INTERSECT / EXCEPT operators**
- **Search condition in ON clause**
- **Logical query processing**

Information System

Betekenis

- geheel van mensen, apparatuur en procedures binnen een organisatie, dat gegevens verzamelt, verwerkt, opslaat, terugzoekt en kan oproepen ten dienste van gebruikers

Voorbeelden

- systeem voor polisadministratie en schade-afhandeling bij verzekeringsmaatschappijen
- boekingssysteem bij reisbureaus
- elektronisch patientendossier
- voorraadadministratiesysteem
- studentadministratie- en studievoortgangssysteem bij onderwijsinstellingen (bijvoorbeeld SAS, Alluris)

Course Database Implementation

Beroepstaak

- Realiseer een relationele database met bijbehorende multi-user applicatie waarbij optimaal gebruik wordt gemaakt van de mogelijkheden van moderne relationele database management systemen (RDBMS)

Beroepsproduct

- Relationele Database inclusief business logica
- Client applicatie

Course Database Implementation

Onderwerpen

- Advanced SQL
- Constraints
- Transact-SQL
- Stored procedures
- Triggers
- Security
- Transactions
- Indexing
- Concurrency
- [Codegeneratie, User Defined Functions]
- [Thema Client/Server Architecturen]

Course Database Implementation

Boeken

- Jorgensen, ***Microsoft SQL Server 2012 Bible***, Wiley
- Wiegerink, Bijpost, de Groot, ***Relationele Databases en SQL***, Academic Service.

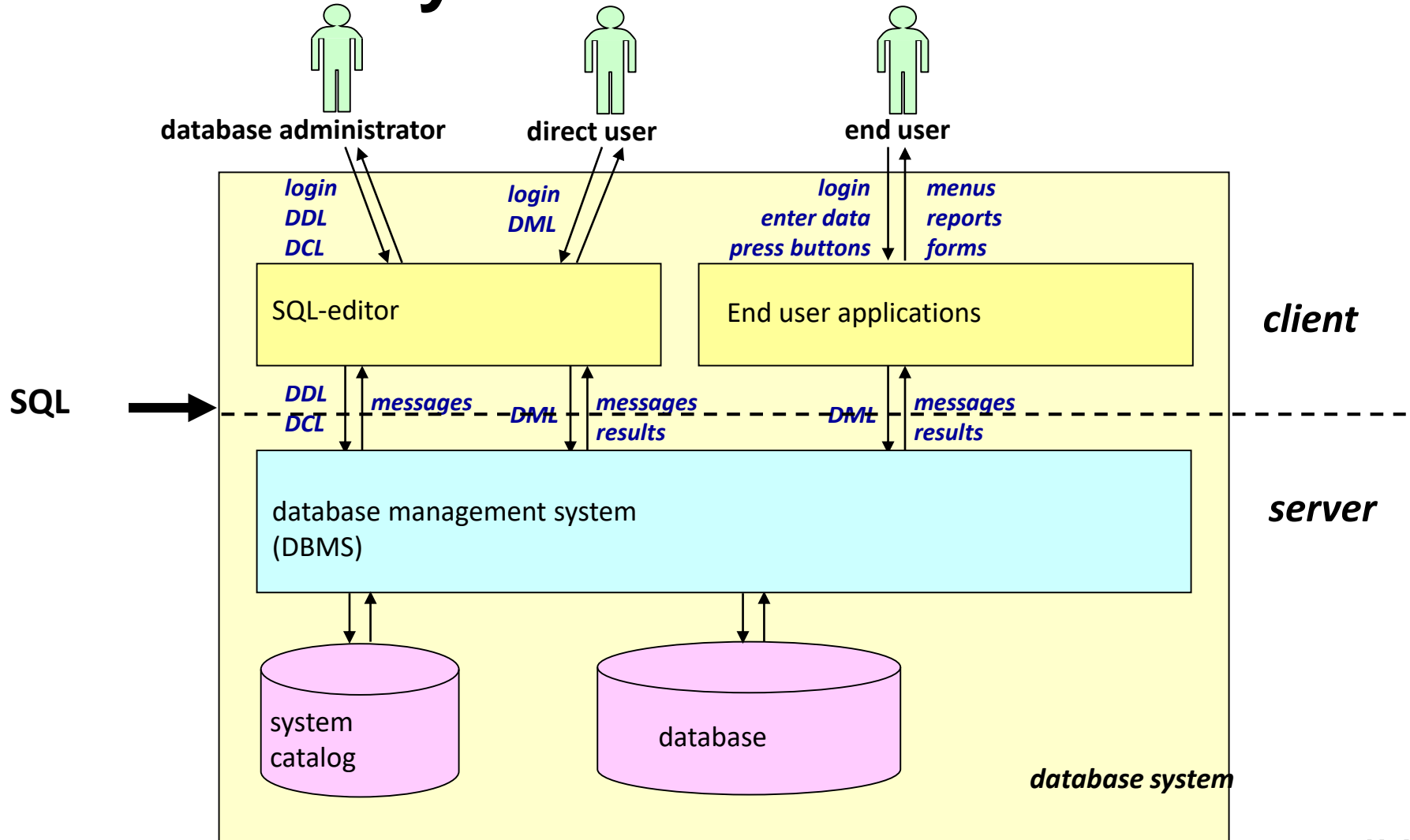
Overig materiaal

- Studiehandleiding
- Werkboek (kan bijgewerkt worden tijdens de course)
- Sheets
- Case Study

Tools

- MS SQL Server 2014 (of 2016) Developer Edition.
- SAP PowerDesigner 16 of hoger.
- AdventureWorks (2014) en overige database downloaden, zie werkboek

Database system



SQL and Relational Databases

Relational Model developed by Codd, early 70s

***Sequel* as query language for rdbms *System R*
(IBM)**

First commercial version in 1979 for the Oracle database

Declarative language, set-oriented

- A query specifies the tables, columns, row qualifiers
- SQL database server figures out HOW to manipulate the data

SQL: DDL, DCL and DML

Data Definition Language (DDL)

- To create and drop data structures

Data Control Language (DCL)

- To secure table data

Data Manipulation Language (DML)

- To retrieve, insert, delete and modify data

Database programming language

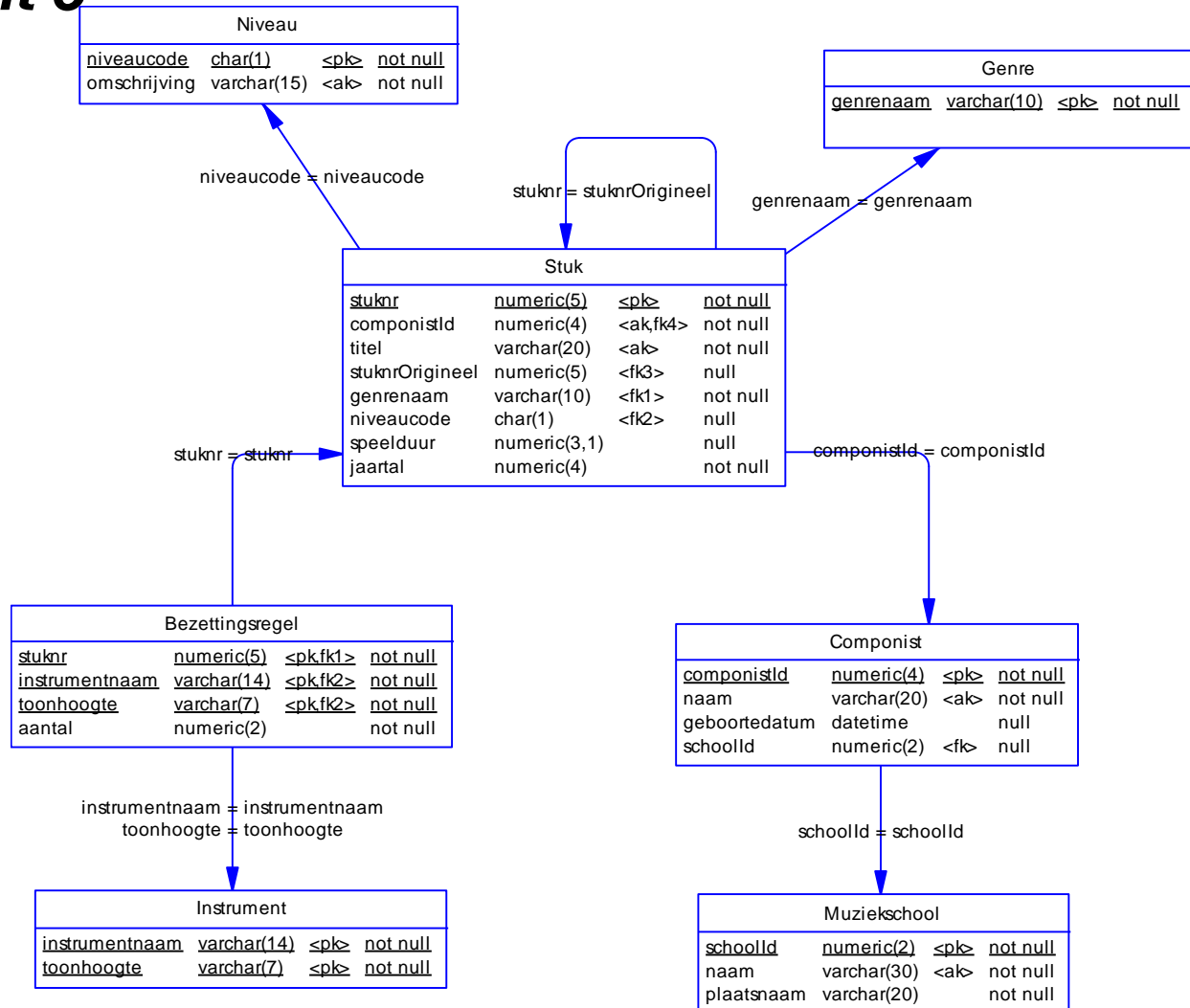
- Embedding in other languages
- Vendors offer SQL-specific programming languages
 - MS SQL Server: *Transact-SQL*

Herhaling SQL

Doe het zelf....

Maak de herhalingsopdrachten op de muziekdatabase

Nu: Opdracht 5



**Bij welke stukken bestaat de bezetting uit
ondermeer een saxofoon?**

Herhalingsopdracht 3

Bij welke stukken bestaat de bezetting uit ondermeer een saxofoon?

Herhalingsopdracht 3

```
SELECT stuknr, titel
FROM stuk
WHERE stuknr IN (
    SELECT stuknr
    FROM bezettingsregel
    WHERE instrumentnaam = 'saxofoon'
)
```

-- of

```
SELECT DISTINCT s.stuknr, s.titel
FROM stuk s
    INNER JOIN bezettingsregel b
    ON s.stuknr = b.stuknr
WHERE b.instrumentnaam = 'saxofoon'
```

Correlated Subquery

Een niet-gecorreleerde subquery

Geef de speelstukken met het meest voorkomende instrument
(instrumentnaam + toonhoogte)

Anders geformuleerd:

Geef het instrument
(instrumentnaam + toonhoogte)
dat het vaakst voorkomt in een stuk.
Toon stuknummer, instrumentnaam
en toonhoogte.

stuknr	instrumentnaam	toonhoogte	aantal
2	drums		1
2	piano		1
2	saxofoon	alt	2
2	saxofoon	tenor	1
3	fluit		1
5	fluit		3
9	fluit		1
9	fluit	alt	1
9	piano		1
12	fluit		2
12	piano		1
13	drums		1
13	fluit		2
13	saxofoon	alt	1
13	saxofoon	tenor	1
14	fluit		1
14	piano		1
15	fluit	alt	2
15	piano		1
15	saxofoon	alt	2

Een niet-gecorreleerde subquery

Geef de speelstukken met het meest voorkomende instrument (naam + toonhoogte)

```
SELECT stuknr, instrumentnaam, toonhoogte  
FROM bezettingsregel  
WHERE aantal = (SELECT MAX(aantal)  
                FROM bezettingsregel)
```

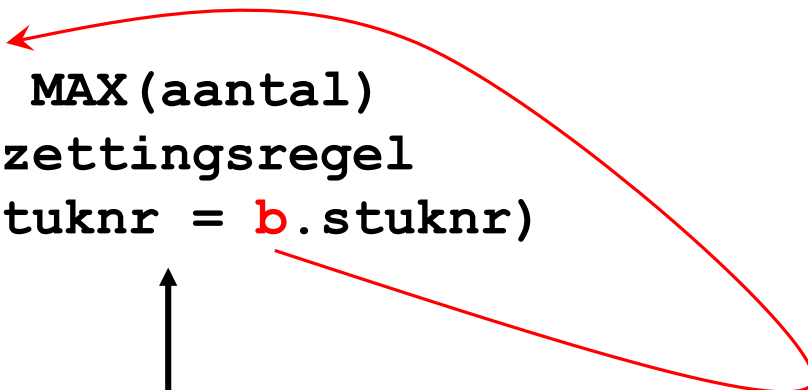


zelfstandig executeerbaar

Gecorreleerde subquery

Geef voor ieder stuk het instrument (naam + toonhoogte) dat het meest voorkomt **in dat stuk**

```
SELECT stuknr, instrumentnaam, toonhoogte, aantal
FROM bezettingsregel b
WHERE aantal = (SELECT MAX(aantal)
                FROM bezettingsregel
                WHERE stuknr = b.stuknr)
```



niet zelfstandig executeerbaar

Bij welke stukken bestaat de bezetting uit ondermeer een saxofoon?

Herhalingsopdracht 3

```
SELECT stuknr, titel
FROM stuk
WHERE stuknr IN (
    SELECT stuknr
    FROM bezettingsregel
    WHERE instrumentnaam = 'saxofoon'
)
```

-- of

```
SELECT distinct s.stuknr, s.titel
FROM stuk s
    INNER JOIN bezettingsregel b
    ON s.stuknr = b.stuknr
WHERE b.instrumentnaam = 'saxofoon'
```

Nu de informatiebehoefte anders gesteld

```
SELECT stuknr, titel  
FROM stuk s  
WHERE
```

er bestaat voor dat stuk een bezetting met een saxofoon

EXISTS met gecorreleerde subquery

```
SELECT stuknr, titel
FROM stuk s
WHERE EXISTS

    (SELECT 1

     FROM bezettingsregel

     WHERE stuknr = s.stuknr
        AND
        instrumentnaam = 'saxofoon')
```

EXISTS met gecorreleerde subquery

```
SELECT stuknr, titel  
FROM stuk s  
WHERE EXISTS      -- logische operator op verzameling
```



```
(SELECT 1  -- of: 'onzin'
```

```
FROM bezettingsregel
```

```
WHERE stuknr = s.stuknr
```

```
AND
```

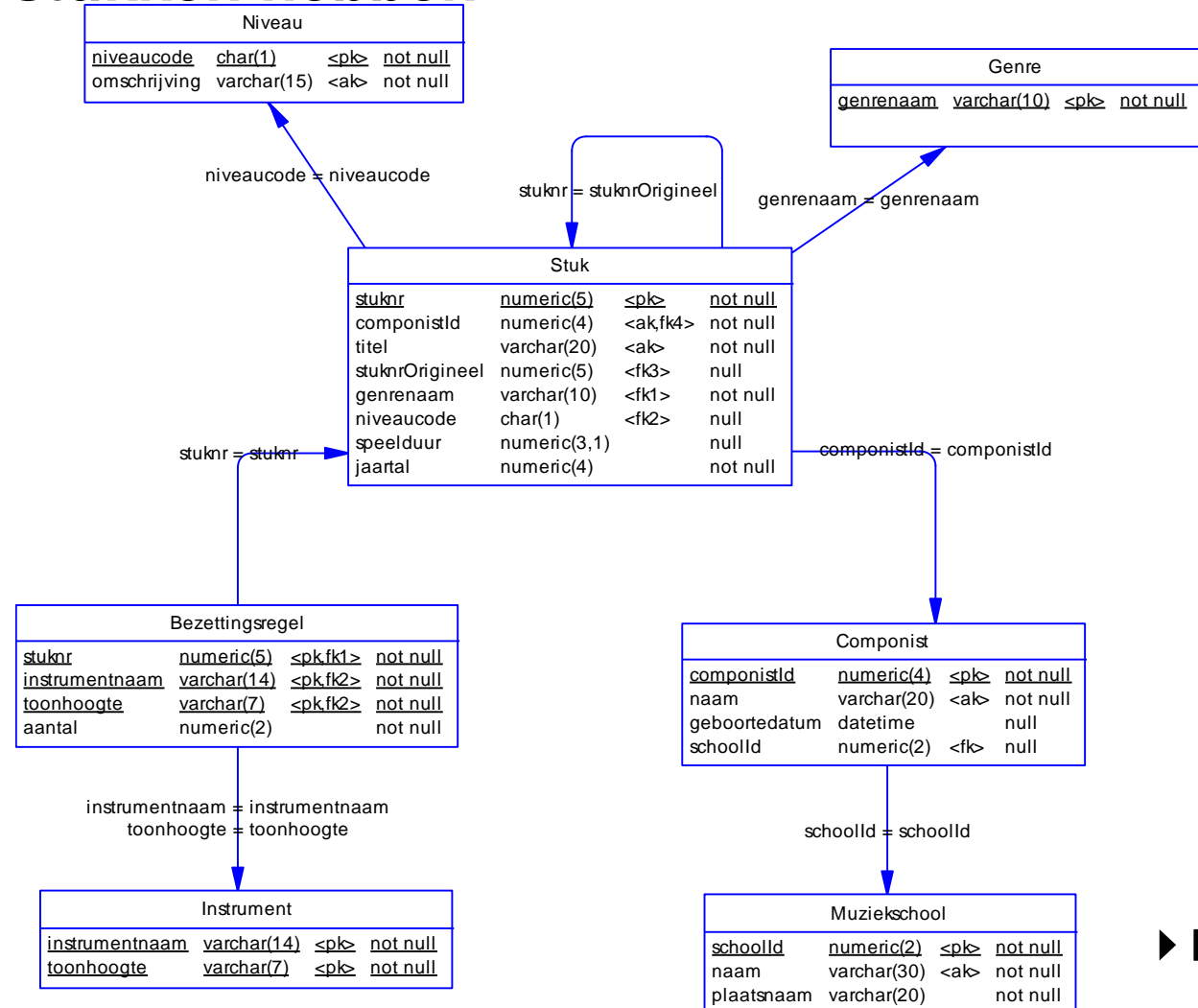
```
instrumentnaam = 'saxofoon')
```

geeft een verzameling voor elk stuknr uit de outer query



Geef de componisten die zowel voor fluit als voor saxofoon speelstukken hebben

Eerst even
bestuderen....



Oplossing

SELECT componistid, naam

FROM componist c

WHERE

er bestaat voor die componist een spelstuk met fluit

AND

er bestaat voor die componist een spelstuk met saxofoon

Oplossing

```
SELECT componistid, naam  
FROM componist c  
WHERE
```

er bestaat voor die componist een speelstuk met fluit

AND

er bestaat voor die componist een speelstuk met saxofoon

```
SELECT componistid, naam  
FROM componist c  
WHERE EXISTS (SELECT 1  
               FROM stuk s  
               INNER JOIN bezettingsregel b  
               ON s.stuknr = b.stuknr  
               WHERE  
                 b.instrumentnaam = 'fluit'  
                 AND componistid = c.componistid)  
AND  
EXISTS (...)
```


NOT EXISTS

Geef alle instrumenten (naam en toonhoogte) die NIET gebruikt worden in stukken

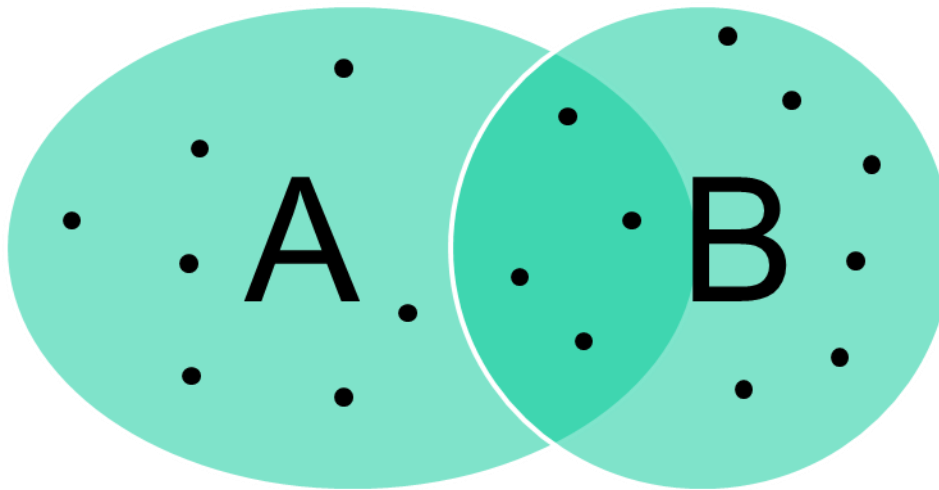
NOT EXISTS

Geef alle instrumenten (naam en toonhoogte) die NIET gebruikt worden in stukken

```
SELECT instrumentnaam, toonhoogte
FROM instrument i
WHERE NOT EXISTS
    (SELECT 1
     FROM bezettingsregel
     WHERE
         i.instrumentnaam = instrumentnaam
        AND
         i.toonhoogte = toonhoogte)
```

SET OPERATORS

A table is a set of records: table A, table B



- A UNION B ?
- A EXCEPT B ?
- A INTERSECT B ?
- A DIFFERENCE B ?

UNION operator

Geef voor *alle* componisten het aantal stukken dat hij/zij heeft gecomponeerd

UNION operator

Geef voor *ALLE* componisten het aantal stukken dat hij/zij heeft gecomponeerd

Foute oplossing:

```
SELECT C.ComponistId, Naam, COUNT(*)  
FROM Componist C  
      INNER JOIN Stuk S  
            ON C.ComponistId = S.ComponistId  
GROUP BY C.ComponistId, Naam
```

Waarom fout?

UNION operator

Geef voor *alle* componisten het aantal stukken dat hij/zij heeft gecomponeerd

```
SELECT C.ComponistId, Naam, COUNT(*)  
FROM Componist C  
      INNER JOIN Stuk S  
            ON C.ComponistId = S.ComponistId  
GROUP BY C.ComponistId, Naam
```

UNION

```
SELECT ComponistId, Naam, 0  
FROM Componist  
WHERE ComponistId NOT IN (SELECT ComponistId  
                          FROM Stuk)
```

EXCEPT / INTERSECT operators

Leveren een nieuwe tabel met **distinct** rows

- Er is geen EXCEPT ALL of INTERSECT ALL, zoals bij UNION

Voorbeeld:

- Geef de verschillende componisten die geen stukken geschreven hebben (EXCEPT)

Handig, maar niet echt nodig

Er is geen DIFFERENCE operator

- Hoe kun je een DIFFERENCE realiseren met EXCEPT en UNION?

ON <search condition>

Herhalingsopdracht 7

Geef voor elk niveau de niveaucode, de omschrijving en het aantal klassieke speelstukken, óók als er voor een niveau geen klassieke speelstukken zijn.

Goede oplossing:

```
SELECT n.niveaucode, n.omschrijving, COUNT(stuknr)
FROM niveau n
      LEFT OUTER JOIN stuk s
        ON n.niveaucode = s.niveaucode
        AND
           s.genrenaaam = 'klassiek' -- vóór de join uitgevoerd
GROUP BY n.niveaucode, n.omschrijving
```

Logical query processing phases

MS SQL Server

```
(7) SELECT (8) DISTINCT (10) <TOP_spec> <select_list>  
(1) FROM <left_table>  
(3)    <join_type> JOIN <right_table>  
(2)      ON <join_condition>  
(4) WHERE <where_condition>  
(5) GROUP BY <group_by_list>  
(6) HAVING <having_condition>  
(9) ORDER BY <order_by_list>
```

Logical processing (not Physical):

Each step generates a virtual table that is used as input to the next step

Hoe wordt de goede oplossing logisch uitgevoerd?

```
(5) SELECT n.niveaudecode, n.omschrijving, COUNT(stuknr)
(1) FROM niveau n
(3)     LEFT OUTER JOIN stuk s
(2)         ON n.niveaudecode = s.niveaudecode
           AND
           s.genrenaaam = 'klassiek'
(4) GROUP BY n.niveaudecode, n.omschrijving
```

Virtual Table 1

(1) **FROM:** Cartesian product between *niveau* and *stuk*, results in VT1

A	beginners	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
A	beginners	2	2	Blue bird	1	jazz	B	4.0	1988
A	beginners	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
A	beginners	5	5	Lina	NULL	klassiek	B	5.0	1979
A	beginners	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
A	beginners	9	2	Cradle song	8	klassiek	B	3.5	1990
A	beginners	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
A	beginners	12	9	I'll never go	10	pop	A	6.0	1996
A	beginners	13	10	Swinging Lina	5	jazz	B	8.0	1997
A	beginners	14	5	Little Lina	5	klassiek	A	4.3	1998
A	beginners	15	10	Blue sky 1	jazz	A	4.0	1998	
B	gevorderden	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
B	gevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
B	gevorderden	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
B	gevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
B	gevorderden	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
B	gevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990
B	gevorderden	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
B	gevorderden	12	9	I'll never go	10	pop	A	6.0	1996
B	gevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
B	gevorderden	14	5	Little Lina	5	klassiek	A	4.3	1998
B	gevorderden	15	10	Blue sky 1	jazz	A	4.0	1998	
C	vergevorderden	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
C	vergevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
C	vergevorderden	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
C	vergevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
C	vergevorderden	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
C	vergevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990
C	vergevorderden	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
C	vergevorderden	12	9	I'll never go	10	pop	A	6.0	1996
C	vergevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
C	vergevorderden	14	5	Little Lina	5	klassiek	A	4.3	1998
C	vergevorderden	15	10	Blue sky 1	jazz	A	4.0	1998	

VT2

(2) **ON:** Only rows for which the join condition is TRUE are inserted in VT2

A	beginners	14	5	Little Lina	5	klassiek A	4.3	1998
B	gevorderden	3	4	Air pur char..	NULL	klassiek B	4.5	1953
B	gevorderden	5	5	Lina NULL	klassiek B	5.0	1979	
B	gevorderden	9	2	Cradle song	8	klassiek B	3.5	1990

VT3

(3) **LEFT OUTER**: Rows for which a match was not found are added to VT2 as outer rows, results in VT3

A	beginners	14	5	Little Lina	5	klassiek A	4.3	1998
B	gevorderden	3	4	Air pur char..	NULL	klassiek B	4.5	1953
B	gevorderden	5	5	Lina NULL	klassiek B	5.0	1979	
B	gevorderden	9	2	Cradle song	8	klassiek B	3.5	1990
C	vergevorde..	NULL	NULL	NULL NULL	NULL	NULL	NULL	NULL

VT4

(4) GROUP BY: Rows from VT3 are arranged in groups based on the column list in the GROUP BY clause, results in VT4

<i>Group</i>	<i>rows of VT3</i>									
A beginners	A beginners	14	5	Little Lina	5	klassiek A	4.3	1998		
B gevorderden	B gevorder..	3	4	Air pur charmer	NULL	klassiek B	4.5	1953		
	B gevorder..	5	5	Lina	NULL	klassiek B	5.0	1979		
	B gevorder..	9	2	Cradle song	8	klassiek B	3.5	1990		
C vergevorderden	C vergev..	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

VT5: result to caller

(5) **SELECT** n.niveaucode, n.omschrijving, COUNT(stuknr)

Only base columns of VT4. If you refer to columns of VT3, then they must be aggregated

A	beginners	1
B	gevorderden	3
C	vergeevorderden	0

Herhalingsopdracht 7, revisited

Foute oplossing:

```
SELECT n.niveaude, n.omschrijving, COUNT(stuknr)
FROM niveau n
      LEFT OUTER JOIN stuk s
        ON n.niveaude = s.niveaude
WHERE
      s.genreanaam = 'klassiek' -- NA de join uitgevoerd!
GROUP BY n.niveaude, n.omschrijving
```

Hoe wordt de foute oplossing logisch uitgevoerd?

```
(6) SELECT n.niveaudecode, n.omschrijving, COUNT(stuknr)
(1) FROM niveau n
(3)     LEFT OUTER JOIN stuk s
(2)         ON n.niveaudecode = s.niveaudecode
(4) WHERE s.genrenaaam = 'klassiek'
(5) GROUP BY n.niveaudecode, n.omschrijving
```

VT1

(1) FROM: Cartesian product between *niveau* and *stuk*, results in VT1

A	beginners	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
A	beginners	2	2	Blue bird	1	jazz	B	4.0	1988
A	beginners	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
A	beginners	5	5	Lina	NULL	klassiek	B	5.0	1979
A	beginners	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
A	beginners	9	2	Cradle song	8	klassiek	B	3.5	1990
A	beginners	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
A	beginners	12	9	I'll never go	10	pop	A	6.0	1996
A	beginners	13	10	Swinging Lina	5	jazz	B	8.0	1997
A	beginners	14	5	Little Lina	5	klassiek	A	4.3	1998
A	beginners	15	10	Blue sky 1	jazz	A	4.0	1998	
B	gevorderden	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
B	gevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
B	gevorderden	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
B	gevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
B	gevorderden	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
B	gevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990
B	gevorderden	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
B	gevorderden	12	9	I'll never go	10	pop	A	6.0	1996
B	gevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
B	gevorderden	14	5	Little Lina	5	klassiek	A	4.3	1998
B	gevorderden	15	10	Blue sky 1	jazz	A	4.0	1998	
C	vergevorderden	1	1	Blue bird	NULL	jazz	NULL	4.5	1954
C	vergevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
C	vergevorderden	3	4	Air pur charmer	NULL	klassiek	B	4.5	1953
C	vergevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
C	vergevorderden	8	8	Berceuse	NULL	klassiek	NULL	4.0	1786
C	vergevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990
C	vergevorderden	10	8	Non piu andrai	NULL	klassiek	NULL	NULL	1791
C	vergevorderden	12	9	I'll never go	10	pop	A	6.0	1996
C	vergevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
C	vergevorderden	14	5	Little Lina	5	klassiek	A	4.3	1998
C	vergevorderden	15	10	Blue sky 1	jazz	A	4.0	1998	

VT2

(2) **ON**: Only rows for which the join condition is TRUE are inserted in VT2

A	beginners	12	9	I'll never go	10	pop	A	6.0	1996
A	beginners	14	5	Little Lina	5	klassiek	A	4.3	1998
A	beginners	15	10	Blue sky 1	jazz	A	4.0	1998	
B	gevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
B	gevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
B	gevorderden	3	4	Air pur char..	NULL	klassiek	B	4.5	1953
B	gevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
B	gevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990

VT3

(3) LEFT OUTER: Rows for which a match was not found are added to VT2 as outer rows, results in VT3

A	beginners	12	9	I'll never go	10	pop	A	6.0	1996
A	beginners	14	5	Little Lina	5	klassiek	A	4.3	1998
A	beginners	15	10	Blue sky 1	jazz	A	4.0	1998	
B	gevorderden	2	2	Blue bird	1	jazz	B	4.0	1988
B	gevorderden	3	4	Air pur char..	NULL	klassiek	B	4.5	1953
B	gevorderden	5	5	Lina	NULL	klassiek	B	5.0	1979
B	gevorderden	9	2	Cradle song	8	klassiek	B	3.5	1990
B	gevorderden	13	10	Swinging Lina	5	jazz	B	8.0	1997
C	vergevorde..	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

VT4

(4) **WHERE**: where-filter applied tot VT3. Only rows for which the where-condition is TRUE are inserted into VT4

A	beginners	14	5	Little Lina	5	klassiek A	4.3	1998
B	gevorderden	3	4	Air pur char..	NULL	klassiek B	4.5	1953
B	gevorderden	5	5	Lina NULL	klassiek B	5.0	1979	
B	gevorderden	9	2	Cradle song	8	klassiek B	3.5	1990

VT5

(5) GROUP BY: Rows from VT4 are arranged in groups based on the column list in the GROUP BY clause, results in VT5

<i>Group</i>	<i>rows of VT4</i>							
A beginners	A beginners	14	5	Little Lina	5	klassiek A	4.3	1998
B gevorderden	B gevorderden	3	4	Air pur charmer	NULL	klassiek B	4.5	1953
	B gevorderden	5	5	Lina	NULL	klassiek B	5.0	1979
	B gevorderden	9	2	Cradle song	8	klassiek B	3.5	1990

VT6: result to caller

(6) **SELECT** n.niveaucode, n.omschrijving, COUNT(stuknr)

Only base columns of VT5. If you refer to columns of VT4, then they must be aggregated

A	beginners	1
B	gevorderden	3

Een andere goede oplossing

```
SELECT n.niveaudecode, n.omschrijving, count(stuknr)
FROM niveau n
      LEFT OUTER JOIN stuk s
        ON n.niveaudecode = s.niveaudecode
WHERE
      s.genrenaaam = 'klassiek'
      OR
      s.genrenaaam IS NULL
GROUP BY n.niveaudecode, n.omschrijving
```