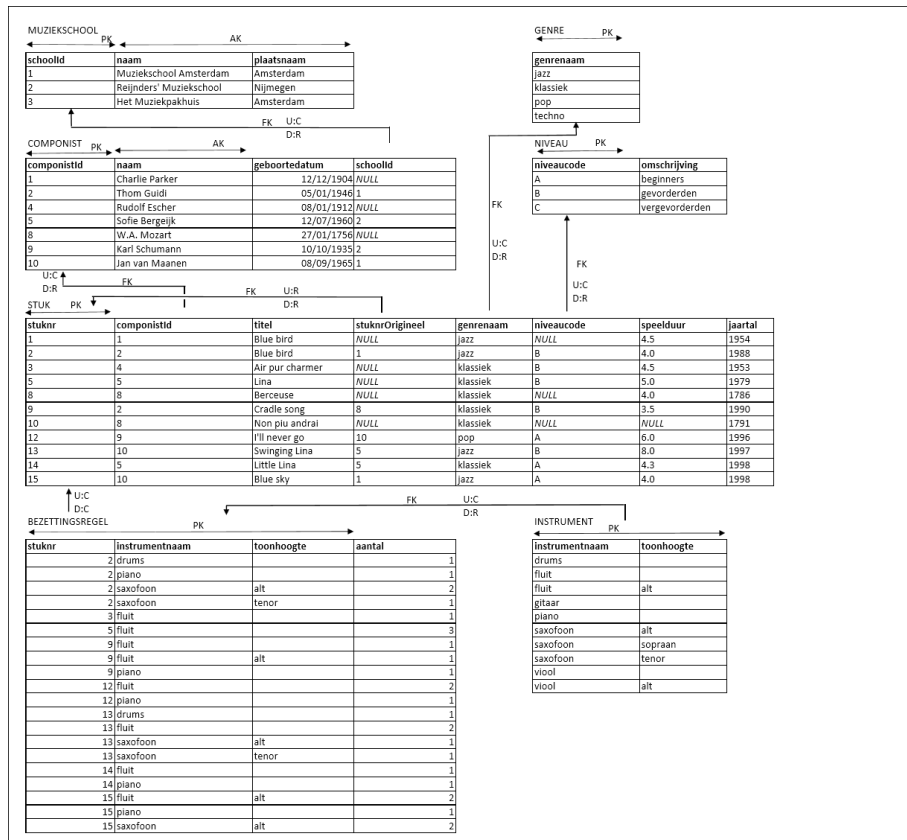


## DB – Oefentoets SQL Toets (2)

De muziekdatabase uit de course is de basis voor deze toets.



### Opgave 1 (8 punten)

Geef het resultaat van de volgende query, uitgaande van de inhoud van de tabellen die op blad 1 is gegeven.

```
select c.componistId,  
       c.naam,  
       c.geboortedatum  
from   Componist c  
left outer join Muziekschool m  
on      c.schoolId = m.schoolId  
where   m.schoolId <> 2  
order by c.geboortedatum desc
```

#### Uitwerking

componistId	naam	geboortedatum
10	Jan van Maanen	1965-09-08
2	Thom Guidi	1946-01-05

#### Nakijkmodel:

- 2 bij fout in sortering asc ipv desc
- 1 voor elke kolom die ontbreekt, max -2
- 4 wanneer componisten met meegenomen zijn die niet verbonden zijn aan een school (null)
- 2 wanneer left join als right join is gelezen

### Opgave 2 (14 punten)

A. (8 punten) Geef het SQL statement voor de volgende informatiebehoefte:

*Welke speelstukken (dus stukken met een niveau-aanduiding) zijn gemaakt door een componist werkzaam voor een school waarvan de plaats eindigt op 'dam'.*

*Geef in de uitvoer van het SQL statement de volgende informatie weer:*

- de componistid,
- de naam van de componist,
- de naam van de muziekschool,
- de totale speelduur en
- het totaal aantal stukken dat de componist componeerde.

Het resultaat dient er als volgt uit zien:

Componistid	Naam_componist	Naam_Muziekschool	Totale_Speelduur	Aantal_stukken
2	Thom Guidi	Muziekschool Amsterdam	7.5	2
10	Jan van Maanen	Muziekschool Amsterdam	12.0	2

#### Uitwerking

```

SELECT componist.componistId,
       componist.naam AS Naam_componist,
       Muziekschool.naam AS Naam_Muziekschool,
       SUM(speelduur) AS totale_speelduur,
       COUNT(stuknr) AS aantal_stukken
FROM   STUK
       INNER JOIN
       Componist
       ON   stuk.componistId = Componist.componistId
       INNER JOIN
       Muziekschool
       ON   componist.schoolid = muziekschool.schoolid
WHERE  stuk.niveaucode IS NOT NULL
AND    Muziekschool.plaatsnaam LIKE '%dam'
GROUP BY componist.componistid,componist.naam, muziekschool.naam

```

Nakijkmodel:

- FOUT in SELECT per kolom: -1 MAX -2
- FOUT in STATISTISCHE functie, SUM / COUNT: -2 per fout MAX -2
- FOUT in FROM per table -1, INNER JOIN onjuist -1 MAX -3
- FOUT in WHERE -2 per onjuist criterium MAX -4
- FOUT in GROUP BY: - 2 (onjuiste groupering) MAX -2
- Fout in HAVING: -2
- Syntax fouten - 1 per fout

B. (2punten) Welke regel moet er toegevoegd worden aan het eind van het SQL-statement uit onderdeel A voor de volgende informatiebehoefte:

*Welke van de componisten uit opgave A schreef stukken met een totale speelduur groter dan 10?*

Uitwerking:

HAVING sum(speelduur) > 10

Nakijkmodel:

- 1 als totale\_speelduur is gebruikt ipv sum(speelduur)
- 2 bij gebruik where ipv having

C. (2 punten) Dezelfde vraag moet beantwoord worden met behulp van een view. Maak een view (vwComponistTotalen) aan voor de informatie behoefte uit onderdeel A. Je hoeft in het SQL statement dat de view definieert de query uit onderdeel A niet nog een keer op te nemen. Zet hiervoor in de plaats XX. De rest moet wel syntactisch correct zijn.

Uitwerking:

CREATE VIEW vwComponistTotalen AS

XX

Nakijkmodel:

-1 als AS vergeten

0 haakjes vergeten >> HOEFT NIET

-1 naam vergeten

D. (2 punten) Gebruik de view uit bovenstaande vraag voor het beantwoorden van de volgende informatiebehoefte:

*Welke van de componisten uit opgave A schreef stukken met een totale speelduur groter dan 10?*

*Geef in de uitvoer de componistID van de componisten.*

Commented [MN1]: Geef het SQL Statement....

Uitwerking:

```
select componistId
from vwComponistTotalen
where totale_speelduur > 10
```

Nakijkmodel:

-2 having ipv where

-1 foute kolom

-1 \* ipv kolom

### Opgave 3 (8 punten)

Geef het SQL-statement voor de volgende informatiebehoefte:

*Welke stukken zijn klassiek of een bewerking van een klassiek stuk?  
Geef in de uitvoer van het SQL statement de volgende informatie weer:  
- het stuknummer en de titel.*

Uitwerking:

```
SELECT stuknr, titel
FROM Stuk S
WHERE genrenaam = 'klassiek'
OR stuknrOrigineel IN (SELECT stuknr
FROM stuk
WHERE genrenaam = 'klassiek')
```

Nakijkmodel:

-1 bij fout in kolommen

-4 deze helft van WHERE vergeten

-4 als subselect ontbreekt

-2 bij AND ipv OR

```
-- Ook goed
SELECT stuknr, titel
FROM Stuk
WHERE stuknr IN (SELECT stuknr
                  FROM Stuk
                  WHERE genrenaam = 'klassiek')
OR stuknrOrigineel I
   IN (SELECT stuknr
        FROM stuk
        WHERE genrenaam = 'klassiek')
```

#### Nakijkmodel:

-1 bij fout in kolommen  
-4 als subselect ontbreekt  
-4 als subselect ontbreekt  
-2 bij AND ipv OR

```
-- Ook goed
SELECT S.stuknr, S.titel
FROM Stuk S
LEFT OUTER JOIN
    Stuk Sorg
ON S.stuknrOrigineel = Sorg.stuknr
WHERE S.genrenaam = 'klassiek'
OR Sorg.genrenaam = 'klassiek'
```

#### Nakijkmodel:

-1 bij fout in kolommen  
-4 bij INNER JOIN of foute ON conditie  
-4 bij ontbreken van één van beiden  
-2 bij AND ipv OR

### Opgave 4 (10 punten)

Geef het SQL-statement voor de volgende informatie behoefte:

*Welke componist heeft alleen klassieke stukken geschreven, met piano als onderdeel van de bezetting, maar zonder alt fluit in de bezetting?*

*Geef in de uitvoer van het SQL statement de volgende informatie weer:*

- de componistId,
- de naam van de componist.

Het resultaat dient er als volgt uit zien:

ComponistId	Naam
5	Sofie Bergeijk

#### Uitwerking:

```
Select DISTINCT c.componistId, naam
From Componist C
Inner join Stuk S
```

```

On C.componistId = s.ComponistId
Where S.genrenaam = 'klassiek'
and S.stuknr in (select stuknr
                 from Bezettingsregel
                 where instrumentnaam = 'piano')

and S.stuknr not in(
    select stuknr
    from Bezettingsregel
    where instrumentnaam = 'fluit'
    and toonhoogte = 'alt')

```

ook goed:

```

Select DISTINCT c.componistId,naam
From Componist C
Inner join Stuk S On C.componistId = s.ComponistId
inner join Bezettingsregel B On S.stuknr = B.stuknr
Where S.genrenaam = 'klassiek'
and instrumentnaam = 'piano'
and S.stuknr not in(
    select stuknr
    from Bezettingsregel
    where instrumentnaam = 'fluit'
    and toonhoogte = 'alt')

```

#### Nakijkmodel:

- 1 voor per foutief geselecteerde kolom
- 3 fout in subselect
- 2 or ipv and
- 3 fout in on clause

Bovenstaande query geeft echter ook de componist als die andere stukken (anders dan klassiek met piano zonder alt fluit) heeft gecomponeerd. Verwarrend is het woord 'alleen' in de vraag. Het was niet bedoeling om slechts alleen die componisten op te vragen die alleen maar klassieke stukken met een piano en zonder fluit hadden gecomponeerd en dus geen andere stukken.

Anders zou je nog onderstaande conditie moeten toevoegen:

```

and componistId NOT IN
(
    select s.ComponistId
    from Bezettingsregel B right join Stuk S
        on B.stuknr = S.stuknr
    where S.genrenaam <> 'klassiek'
    or instrumentnaam <> 'fluit'
    or toonhoogte <> 'alt')

```

Ook goed is (zonder distinct nu):

```

Select componistId,naam
From Componist
WHERE componistId IN
(
    select s.ComponistId
    from Bezettingsregel B inner join Stuk S
        on B.stuknr = S.stuknr
    where S.genrenaam = 'klassiek'
    and instrumentnaam = 'piano')
and componistId NOT IN
(
    select s.ComponistId
    from Bezettingsregel B inner join Stuk S

```

```
on B.stuknr = S.stuknr
where S.genrenaam = 'klassiek'
and instrumentnaam = 'fluit'
and toonhoogte = 'alt')
```

NAKIJKMODEL

### Opgave 5 (10 punten)

Voor de muziekdatabase geldt de volgende beperkingsregel:

*"Er mogen maximaal 2 componisten verbonden zijn aan een muziekschool."*

Dit moet worden opgelost met een User Defined Function (UDF) en een check constraint op de tabel Componist.

Hieronder staat de SQL voor het maken van de UDF en het maken van de check constraint. De UDF moet het aantal componisten dat werkzaam is voor een betreffende school als resultaat teruggeven.

```
CREATE FUNCTION dbo.fn_CK_ACPSGeefAantalComponistenPerSchool (@school NUMERIC(4,0))
RETURNS INT
AS
BEGIN
    RETURN ( XX )
END
```

De UDF is nog niet klaar. Op de plek waar XX staat moet een SELECT statement komen te staan.

A. (4 punten) Geef het SELECT statement zodat de functie het juiste aantal componisten teruggeeft voor de betreffende school.



ANTW: `SELECT COUNT(schoolid) -- COUNT(*) mag ook  
FROM Componist  
WHERE schoolid = @school`

Nakijkmodel:

- 1 bij fout in count
- 2 @ ontbreekt
- 2 verkeerde tabel

B. (3 punten) Geef het SQL-statement die de check constraint aanmaakt en gebruik maakt van de UDF.

Uitwerking:

```
ALTER TABLE Componist  
ADD CONSTRAINT ck_aantalComponistenPerSchool CHECK(  
dbo.fn_CK_ACPSGeefAantalComponistenPerSchool(schoolid) < 3)
```

Nakijkmodel:

- 2 Alter ontbreekt
- 2 add ontbreekt
- 2 constraint zonder naam of andersom
- 2 Check ontbreekt
- 2 als parameter schoolid ontbreekt
- 1 geen haakjes

Ga ervanuit dat de UDF en de CHECK constraint succesvol zijn aangemaakt. Nu willen we testen of de constraint goed werkt als de beperkingsregel wordt overtreden.

C. (3 punten) Geef twee INSERT of UPDATE statements waarmee we de constraint kunnen testen, 1 statement dat negatief test en 1 dat positief test.

Uitwerking:

Negatief

```
INSERT INTO Componist VALUES (99, 'Testnaam', '1-1-1900', 1)  
of  
UPDATE Componist SET schoolid = 2 where componistid = 10
```

Positief

```
INSERT INTO Componist VALUES (99, 'Testnaam', '1-1-1900', 3)  
of  
UPDATE Componist SET schoolid = 3 where componistid = 10
```

Nakijkmodel:

- 2 postieve danwel negatieve test ontbreekt
- 1 fout in 1 van de statements (quotes, komma, haakjes etc)