

# 2018 voorbeeldvragen

## Datamodellering

Vragen ivm consistentie, (referentiële) integriteit, redundantie.

Nadelen redundantie:

- meer kans op inconsistentie
- meer gegevens bijhouden dus hogere capaciteit nodig.
- complexere aanpassing van gegevens met mogelijks fouten tot gevolg.

Geef (of bespreek, verbeter) het bijhorende conceptueel, logisch en/of fysisch model

## Groep 2

Bespreek volledig de uitvoering van een SELECT statement (volgorde en acties).

1. FROM clause
2. ON clause
3. OUTER clause
4. WHERE clause
5. GROUP BY clause

6. HAVING clause
7. SELECT clause
8. DISTINCT clause
9. ORDER BY clause
10. TOP clause

## Wat is een cartesisch product? Wat moet je meestal doen om dit te vermijden?

de verzameling van alle koppels of geordende paren waarvan het eerste element uit de eerste verzameling en het tweede uit de tweede verzameling komt.

## Groep 3

### Wat is een niet-gecorreleerde subquery?

Als de inner query afhankelijk is van de outer query, is er sprake van een gecorreleerde subquery.

### Geef een alternatieve oplossing voor deze query zonder subquery.

## Groep 4

### Wat is het verschil tussen een gecorreleerde subquery en een niet-gecorreleerde subquery?

Als de inner query afhankelijk is van de outer query, is er sprake van een gecorreleerde subquery.

Los op een met gecorreleerde subqueries:  
"Geef de spelers die voor elke team  
gespeeld heeft"

## Groep 5

Wat is het verschil tussen condities in de  
WHERE en de FROM?

Als conditie in FROM wordt alles zonder combinatie gecombineerd met NULL. Een conditie in de WHERE gaat rijen filteren (eruit gooien).

Geef een zinnig voorbeeld van een theta  
join en leg uit.

Vergelijking met een andere vergelijkingsoperator.

```
“ SELECT s.spelersnr, s.naam, count(sp.spelersnr)
    FROM spelers s INNER JOIN spelers sp
    ON(s.spelersnr sp.spelersnr)
   WHERE length(s.naam) == length(sp.naam)
  GROUP BY s.spelersnr, s.naam
```

# Welke set-operatoren ken je en wat doen ze (kort)?

- UNION combineert resultaten van twee SQL-query's tot een enkele tabel van alle overeenkomende rijen. De twee query's moeten resulteren in hetzelfde aantal kolommen en compatibele gegevenstypen om zich te verenigen. Dubbele records worden automatisch verwijderd tenzij UNION ALL wordt gebruikt.
- INTERSECT neemt de resultaten van twee query's en retourneert alleen rijen die in beide resultaatsets voorkomen.
- EXCEPT: neemt de afzonderlijke rijen van één query en retourneert de rijen die niet in een tweede resultaatset voorkomen.

# Waar dient LC\_COLLATE voor en waarom kan dit belangrijk zijn?

Sorteervolgorde om in de nieuwe database te gebruiken. Dit beïnvloedt de sorteervolgorde toegepast op tekenreeksen, b.v. in query's met ORDER BY, evenals de volgorde die wordt gebruikt in indexen in tekstkolommen.

## Groep 6

# Hoe kan je een constraint van de tabel verwijderen?

```
ALTER TABLE custom_table
```

```
DROP CONSTRAINT fk_states_list;
```

# Geef een alternatief voor de LIKE operator en geef enkele voorbeelden van wat dit

# alternatief meer kan.

SIMILAR TO → ondersteunt POSIX pattern matching\* | denotes alternation (either of two alternatives).

- \* denotes repetition of the previous item zero or more times.
- + denotes repetition of the previous item one or more times.
- ? denotes repetition of the previous item zero or one time.
- {m} denotes repetition of the previous item exactly m times.
- {m,} denotes repetition of the previous item m or more times.
- {m,n} denotes repetition of the previous item at least m and not more than n times.
- Parentheses ( ) can be used to group items into a single logical item.

bron

Leg uit: "connection pooling client side" met tekst en tekening.

Welke types connecties kan je maken naar database vanuit een programmeertaal?

Waarom kan het gebruik van een genereerde serial id gevaarlijk zijn?

## Groep 7

Een query met de volgende elementen:(S F W G H en 1 of meer subqueries).

## Groep 8

Wat is een segmenttype?

Wat typeert een netwerk databank?\*

bouwstenen: recordtypes, settypes, 1:n-  
verband owner-record member-record

- ER → netwerk: n-m → 1-n
  - conceptueel: schema (schema DDL)
  - intern: intern schema (DSDL)
  - extern: subschema (subschema DDL)
- 
- member kan zonder owner
  - meerdere settypes tussen zelfde record types
  - recordtype kan member zijn in meerdere settypes

recordtype: beschrijft gegevens in gegevensbank

settype: geven relaties tussen recordtypes aan

## Groep 9

Geef een voorbeeld van een CTE met recursie.

Geef Louis V en zijn 5 voorgangers.

```
“ WITH RECURSIVE lijst (naam, zoon, nr) AS (
```

```
    SELECT naam, zoon, 0
```

```
    FROM kings
```

```
    WHERE naam == 'Louis V'
```

```
    UNION ALL
```

```
    SELECT naam, zoon, nr + 1
```

```
    FROM lijst L INNER JOIN kings K
```

```
    ON (L.naam == K.zoon)
```

```
    WHERE nr < 5
```

OF

```
“ with recursive boom(naam, cirkeltrond, level) as (
```

```
    select objectnaam, satellietvan, 0
```

```
    from hemelobjecten
```

```
    where objectnaam like '%i%a%a'
```

```
    union all
```

```
    select h.objectnaam, h.satellietvan, level + 1
```

```
    from hemelobjecten h inner join boom b
```

```
    on h.objectnaam == b.cirkeltrond
```

```
    where level < 3
```

```
)
```

```
select naam || '-' || cirkeltrond as stam  
from boom
```

Geef een alternatief met gecorreleerde subqueries om een beperkt deel van de resultaten te tonen (cf het limiteren van resultaten).

## Wat is het verschil tussen een string literal en een identifieer?

String literals are enclosed in single quotes. Double quotes are used to force a string containing otherwise special characters to be interpreted as identifiers.

Ex. assuming table foo with column bar : `SELECT bar as "Bar", 'zz' as OTHER from foo;`

In this select, the first column will have name Bar. Without the ", the name would have been bar since SQL does not normally distinguish case. The second column will be the constant zz and the column name will be other.

The " are also useful to have accented chars like è for non english languages.

bron

## Geef een alternatieve oplossing voor deze query (GROUP BY GROUPING SETS..)



# Groep 10

Waarom maken we niet op elke kolom per definitie een INDEX?

Stelling: "Het gebruik van JOINS is steeds beter dan het gebruik van subqueries".  
Kies een standpunt en verdedig helder.

Waarvoor kan je VIEWS gebruiken?

- Vereenvoudigen van routinematige instructies
- Beveiligen van gegevens, gebruikers kunnen bv. aan de view maar niet aan de onderliggende tabellen
- Hulpmiddel bij het reorganiseren van tabellen
- Stapsgewijs opstellen van queries

Wat is SQL injectie (injection) en hoe kan je dit vermijden?

SQL statements in bv. form input fields zetten. Vermijden door gebruik te maken van prepared statements.

# Groep 11

Wat zijn transacties?

Verzameling van sql-instructies die door 1 gebruiker ingevoerd wordt en waarvan de mutaties blijvend moeten zijn of ongedaan moeten gemaakt worden.

## Geef een voorbeeld van embedded sql.

Bv SQL binnen PHP

## Schrijf een methode die .. (generiek).

## Waarvoor kunnen we TRIGGERS gebruiken?

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events \* A database manipulation (DML) statement (DELETE, INSERT, or UPDATE)

- A database definition (DDL) statement (CREATE, ALTER, or DROP).
- A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be written for the following purposes\* Generating some derived column values automatically

- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

bron

## Groep 12

# Geef de typische kenmerken van een ODMS.

- geneste objecten
- overerving (class ... extends ...)
- collections (set, list, ...)

## Wat zijn RULES? Geef een voorbeeld.

The PostgreSQL rule system allows one to define an alternative action to be performed on insertions, updates, or deletions in database tables. Roughly speaking, a rule causes additional commands to be executed when a given command on a given table is executed. [bron](#)

```
// CREATE RULE "NeKeerletsAnders" AS  
  
ON SELECT TO wedstrijden  
  
WHERE  
  
spelersnr == 7  
  
DO INSTEAD  
  
SELECT 'eerst drie toerkes rond tafel lopen en dan nog eens proberen';
```

===Stelling: "We hebben geen databanken nodig, we kunnen alles met XML." Kies een standpunt en verdedig helder.

XML is not a database. It was never meant to be a database. It is never going to be a database. Relational databases are proven technology with more than 20 years of implementation experience. They are solid, stable, useful products. They are not going away. XML is a very useful technology for moving data between different databases or between databases and other programs. However, it is not itself a database. Don't use it like one." [bron](#)

## Geef een voorbeeld van een NoSQL product en filosofie.

MongoDB → binaire JSON. Vooral document store.

## Leg uit : ACID

- Atomic: mate van garantie dat hele transactie uitgevoerd wordt
- Consistent: na transactie moeten alle integriteitsregels nog gelden
- Isolated: transacties geïsoleerd van elkaar uitgevoerd
- Durable: acties kunnen niet ongedaan gemaakt worden

BASE is tegenovergestelde van ACID

Stelling: "PostgreSQL is de relationele database in alle omstandigheden, we kunnen ze voor alles gebruiken" Kies een standpunt en verdedig helder.

PostgreSQL heeft een ruim aanbod aan datatypes, heeft naar mijn weten veel hogere size limits dan bv. MySQL/MariaDB.

*PostgreSQL decidedly strives to conform to the ANSI-SQL:2008 standard, is fully ACID (Atomicity, Consistency, Isolation and Durability) compliant, and is well-known for its rock-solid referential and transactional integrity. Primary keys, restricting and cascading foreign keys, unique constraints, not null constraints, check constraints and other data integrity features ensure only validated data is stored. [bron](#)*

- *This database management engine is scalable and can handle terabytes of data. [bron](#)*
- *PostgreSQL supports JSON and other NoSQL features like native XML support and key-value pairs with HSTORE. It also supports indexing JSON data for faster access. [bron](#)*
- *There are a variety of predefined functions. [bron](#)*
- *A number of interfaces are available. [bron](#)*

## 13 Groep 13

# Wat is het verschil tussen een venster en een groepering?

When a group by clause is used all the columns in the select list should either be in group by or should be in an aggregate function. While in over() clause, we don't have any such restriction. Aggregates or other functions like row number, lead, can be calculated using over function along with other columns in the select list.

bron

Geef een voorbeeld van een cumulatieve frequentietabel met een theta join enerzijds en een venster anderzijds.

Welke is het meest performant, hoe kan je dit controleren?

---

Revision #1

Created 31 October 2021 21:40:46 by Jasper G.

Updated 3 December 2021 22:13:18 by Jasper G.