



# The Foundation Series

**DATABASES EN SQL**

Certified by  


**Voorbeeldexamen**

Editie 202310

Copyright © EXIN Holding B.V. 2023. All rights reserved.  
EXIN® is a registered trademark.

No part of this publication may be reproduced, stored, utilized or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written permission from EXIN.

# Inhoud

Inleiding	4
Voorbeeldexamen	5
Antwoordsleutel	21
Evaluatie	46

# Inleiding

Dit is het EXIN Databases en SQL Foundation (FS\_DBSQLF.NL) voorbeeldexamen. Op dit examen is het Reglement voor de Examens van EXIN van toepassing.

Dit examen bestaat uit 30 meerkeuzevragen. Elke vraag heeft een aantal antwoorden, waarvan er één correct is.

Het maximaal aantal te behalen punten is 30. Elke goed beantwoorde vraag levert u 1 punt op. U hebt minimaal 17 punten nodig om te slagen.

De beschikbare tijd is 60 minuten.

Veel succes!

# Voorbeeldexamen

**1 / 30**

Wat is het voordeel van een index voor een tabel in een database?

- A) Een extra sleutel, de zogenaamde alternate key
- B) Een extra zoekmogelijkheid voor de betreffende tabel
- C) Een snellere toegang tot de gegevens

**2 / 30**

Met welk onderdeel van een databasemanagementsysteem (DBMS) wordt de structuur van een database vastgelegd?

- A) Data Control Language (DCL)
- B) Data Definition Language (DDL)
- C) Data Manipulation Language (DML)

**3 / 30**

Een database is beschreven op basis van ANSI/SPARC.

Waar zorgt gegevensafhankelijkheid voor?

- A) Dat een programma dat de database raadpleegt, moet worden veranderd als er een andere indexeringstechniek wordt toegepast
- B) Dat een programma niet hoeft te worden gewijzigd wanneer er iets verandert in de structuur van de beschrijving op conceptueel niveau van de database
- C) Dat een programma niet sneller of langzamer verwerkt zal worden wanneer een wijziging in de interne beschrijving van de database is gedaan
- D) Dat een wijziging in de structuur van de database alleen gevolgen heeft als deze wijziging voorkomt in de interne beschrijving van de database van het programma

**4 / 30**

Een databasemanagementsysteem (DBMS) is gebaseerd op ANSI/SPARC.

Wat is het doel van het interne niveau?

- A) Het beschrijft tot welke gegevens de toegang van een gebruiker beperkt is zodat de gebruiker niet alle gegevens te zien kan krijgen.
- B) Het beschrijft welke recordtypen er zijn, hoe de lay-out daarvan is, welke identificaties er zijn, welke waardebereiken mogen voorkomen.
- C) Het zorgt ervoor dat duidelijk is met welke blok grootte en welke indexeringstechnieken gegevens worden opgeslagen.
- D) Het zorgt ervoor dat gebruikers ongeacht hun views dezelfde gegevens in dezelfde formats aangeboden krijgen.

5 / 30

Wat wordt in het relationeel model bedoeld met het begrip 'relatie'?

- A) De beschrijving van een tabel in combinatie met alle rijen met waarden uit die tabel
- B) Een gegeven in een tabel dat verwijst naar een rij in een andere tabel
- C) Een volledige rij met waarden uit een tabel

6 / 30

Eén van de gegevens die een bedrijf wil vastleggen is de naam van hun leverancier, Jansen.

Welk begrip is **niet** van toepassing op de naam Jansen?

- A) Attribuutnaam
- B) Attribuutwaarde
- C) Deel van een rij
- D) Deel van een tupel

7 / 30

Met welk soort regels wordt vastgelegd hoe de nieuwe waarde van een veld afhankelijk is van de huidige waarde?

- A) Afhankelijkheidsregels
- B) Afleidingsregels
- C) Domeinregels
- D) Transitierregels

8 / 30

Een museum beschikt over een groot aantal schilderijen. Voor de administratie van deze schilderijen wordt een relationele database voorgesteld met de volgende relaties:

```
SCHILDER (schildernaam, geboortedatum, datum overlijden, geboorteplaats,
geboorteland)
SCHILDERIJ (<schildernaam>, volgnummer, titel, afmetingen, materiaal
ondergrond, verfsoort)
```

Omdat de schilder van een schilderij niet altijd bekend is, wordt voorgesteld om in die gevallen de naam van de schilder leeg te laten en alleen een volgnummer toe te kennen.

Is dit voorstel mogelijk?

- A) Ja, mits voor deze schilderijen een aparte nummerreeks wordt gedefinieerd die alleen voor deze schilderijen gebruikt mag worden.
- B) Ja, omdat schildernaam een foreign key (vreemde sleutel) is in de relatie SCHILDERIJ, mag deze leeg worden gelaten.
- C) Nee, omdat de naam van de schilder niet bekend is, zal dit schilderij niet in de database kunnen worden opgenomen.
- D) Nee, omdat de primary keys (primaire sleutels) in een relationele database altijd moeten worden ingevuld.

**9 / 30**

Een handelsmaatschappij heeft voor haar verkopen een databasemodel met de volgende relaties:

KLANT (klantnummer, naam, adres, plaats, telefoonnummer)  
FACTUUR (<klantnummer>, factuurnummer, factuurdatum, <artikelnummer>, aantal)  
ARTIKEL (artikelnummer, artikelomschrijving, bedrag excl. BTW)

De handelsmaatschappij wil met een database die op dit model is gebaseerd een factuurbedrag inclusief (incl.) BTW kunnen berekenen.

Moet de handelsmaatschappij daarvoor iets veranderen aan dit databasemodel?

- A) Ja, er moet een regelnummer aan het factuurnummer worden toegevoegd zodat een factuur uit meerdere regels kan bestaan.
- B) Ja, het bedrag incl. BTW moet worden toegevoegd aan de relatie ARTIKEL.
- C) Nee, want het factuurbedrag incl. BTW kan met behulp van een afleidingsregel worden berekend.
- D) Nee, want het factuurbedrag incl. BTW kan met behulp van een transitieregel worden berekend.

**10 / 30**

Een dansschool houdt in een database bij:

- een overzicht van de leerlingen;
- met wie een leerling graag danst;
- wie met wie al een danspaar vormt.

De database is gebaseerd op het volgende relationele model:

LEERLING (leerlingnummer, naam, geboortedatum, geslacht, roepnaam)  
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>)  
DANSPAAR (<leerlingnummer3, leerlingnummer4>)

Ondanks het opnemen van een regel in de database dat de leerlingen die een danspaar vormen vijf jaar of minder in leeftijd mogen verschillen, kan het toch voorkomen dat de leeftijd van een danspaar meer dan vijf jaar verschilt.

Waarom kan dit in de database voorkomen?

- A) Omdat de regel maar op één tupel betrekking mag hebben en de leerlingen in verschillende tupels worden vastgelegd
- B) Omdat de regel wel op meer dan één tupel betrekking mag hebben, maar niet op tupels van dezelfde relatie
- C) Omdat het attribuut geboortedatum in de relatie LEERLING niet verplicht is
- D) Omdat in de relatie DANSPAAR de attributen geboortedatum3 van de eerste leerling en geboortedatum4 van de tweede leerling ontbreken

**11 / 30**

Een dansschool houdt van hun leerlingen bij:

- een overzicht van persoonsgegevens;
- met wie een leerling graag danst (de leerling aangeduid met leerlingnummer1 danst graag met de leerling aangeduid met leerlingnummer2);
- wie met wie een danspaar vormt (de leerling aangeduid met leerlingnummer3 vormt een danspaar met de leerling aangeduid met leerlingnummer4).

De database is gebaseerd op het volgende relationele model:

```
LEERLING (leerlingnummer, naam, geboortedatum, geslacht, roepnaam)
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>)
DANSPAAR (<leerlingnummer3, leerlingnummer4>)
```

Leerling 1011 en 1064 dansen graag met elkaar en zij vormen samen ook al een danspaar. Om dit aan te geven worden in de relatie DANSTGRAAGMET en in de relatie DANSPAAR tupels vastgelegd.

Welke tupels moeten hiervoor worden vastgelegd?

- A) Eén tupel in de relatie DANSPAAR en één tupel in de relatie DANSTGRAAGMET
- B) Eén tupel in de relatie DANSPAAR en twee tupels in de relatie DANSTGRAAGMET
- C) Twee tupels in de relatie DANSPAAR en één tupel in de relatie DANSTGRAAGMET
- D) Twee tupels in de relatie DANSPAAR en twee tupels in de relatie DANSTGRAAGMET

**12 / 30**

Een verhuurder hanteert een database waarin gegevens van panden en subsidies zijn vastgelegd. Per pand kan één keer per jaar subsidie worden aangevraagd. Het aangevraagde bedrag wordt ingevuld zodra de subsidie wordt aangevraagd, het toegekende bedrag pas op het moment dat de subsidie is toegekend.

```
PAND (pandnr, pandnaam, adres, plaats, huur)
SUBSIDIE (<pandnr>, jaar, aangevraagdbedrag, toegekendbedrag, subsidiernr, plaats)
VERHUURDER (naam, pandnr, jaar)
```

Waarom moet er voor iedere relatie SUBSIDIE een relatie PAND voorkomen?

- A) Omdat er voor ieder aangevraagd bedrag een pandnr in de relatie SUBSIDIE zit
- B) Omdat er voor ieder toegekend bedrag een pandnr in de relatie SUBSIDIE zit
- C) Omdat in de relaties PAND en SUBSIDIE het attribuut pandnr deel is van de primary key (primaire sleutel)
- D) Omdat in de relaties PAND en SUBSIDIE het attribuut pandnr voorkomt



### 13 / 30

Een verhuurder van appartementen heeft een database gebaseerd op het volgende relationele model:

```
VERHUUROBJECT (objectnummer, naam, omschrijving, soort, aantal_personen,  
keuken_apart j/n, zwembad j/n, eigen_terras j/n, verhuurdersnummer)  
OBJECTPERIODE (<objectnummer>, weeknummer)  
BOEKING (<objectnummer>, klantnummer, weeknummer, status, betaald j/n)
```

De verhuurder plant de weken dat het object verhuurd kan worden. Hiervoor worden tupels opgenomen in de relatie OBJECTPERIODE.

Wat moet worden gedaan om te voorkomen dat er verhuurobjecten zijn waarvan de verhuurder **niet** bekend is?

- A) Alleen de primary key (primaire sleutel) in de relatie VERHUUROBJECT hoeft veranderd te worden, zodat deze primary key bestaat uit de attributen objectnummer en verhuurdersnummer.
- B) Een relatie VERHUURDER moet opgenomen worden en het attribuut verhuurdersnummer in de relatie VERHUUROBJECT moet een foreign key (vreemde sleutel) worden naar de relatie VERHUURDER.
- C) Het attribuut verhuurdersnummer in de relatie VERHUUROBJECT moet een verplicht attribuut worden.

### 14 / 30

Een dansschool houdt in een database bij:

- een overzicht van de leerlingen;
- met wie een leerling graag danst;
- wie met wie al een danspaar vormt.

De database is gebaseerd op het volgende relationele model:

```
LEERLING (leerlingnummer, naam, geboortedatum, geslacht, roepnaam)  
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>,  
gewenst_geslacht_leerling1, gewenst_geslacht_leerling2)  
DANSPAAR (<leerlingnummer3, leerlingnummer4>)
```

Ondanks het in een tuple van de relatie DANSTGRAAGMET opnemen van verschillende waarden voor de attributen gewenst\_geslacht\_leerling1 en gewenst\_geslacht\_leerling2 is het toch mogelijk dat de twee in de tuple aangeduide leerlingen van gelijk geslacht zijn en een danspaar vormen.

Waardoor komt dit?

- A) De waarden voor de attributen gewenst\_geslacht\_leerling1 en gewenst\_geslacht\_leerling2 hoeven niet over respectievelijk de leerling met het leerlingnummer1 en de leerling met het leerlingnummer2 te gaan.
- B) Ook al zijn de waarden voor de attributen gewenst\_geslacht\_leerling1 en gewenst\_geslacht\_leerling2 verschillend, de geslachten in de verwezen tupels van de relatie LEERLING mogen gelijk zijn.
- C) Voor het attribuut leerlingnummer3 in de relatie DANSPAAR hoeft niet te gelden dat de verwezen leerling als leerlingnummer1 in de relatie DANSTGRAAGMET voorkomt.
- D) Twee leerlingen aangeduid met de combinatie van de attributen leerlingnummer3 en leerlingnummer4 hoeven niet voor te komen als een combinatie van leerlingen aangeduid met leerlingnummer1 en leerlingnummer2.

**15 / 30**

Een verhuurder van appartementen heeft een database gebaseerd op het volgende relationele model:

```
VERHUUROBJECT (objectnummer, naam, omschrijving, soort, aantal_personen,  
keuken_apart j/n, zwembad j/n, eigen_terras j/n, verhuurdernummer)  
OBJECTPERIODE (<objectnummer>, weeknummer)  
BOEKING (<objectnummer>, klantnummer, weeknummer, status, betaald j/n)
```

De verhuurder plant de weken dat het object verhuurd kan worden. Hiervoor worden tupels opgenomen in de relatie OBJECTPERIODE.

Hoe kan bereikt worden dat van alle boekingen de klant bekend is?

- A) Een attribuut klantnummer toevoegen aan de relatie OBJECTPERIODE.
- B) Een relatie KLANT opnemen met een foreign key (vreemde sleutel) objectnummer + klantnummer die verwijst naar de relatie BOEKING.
- C) Er hoeft niets te veranderen aan het model, omdat het al wordt afgedwongen.
- D) Het attribuut klantnummer in de relatie BOEKING een verplicht attribuut maken.

**16 / 30**

Een bedrijf maakt reisgidsen en houdt alle gidsen bij in een relationele database gebaseerd op het volgende schema:

```
REISGIDS (gidsnr, titel, ingangsdatum, einddatum)  
REISPRODUCT (productnr, soortproduct, <gidsnr>, omschrijving, bestemming,  
aantal dagen, soortverblijf, soortvervoer, soortvermaak)
```

Het bedrijf wil een reisgids aan de tabel REISGIDS toevoegen. Het gidsnr is 3067 en nog nooit eerder gebruikt. De titel is 'Exotische Bestemmingen'. De ingangsdatum is 01-01-2023, de einddatum is 01-07-2033. Er zijn nog geen reisproducten voor deze reisgids aanwezig in de relationele database.

Vervolgens worden er tien reisproducten aan gids 3067 toegevoegd.

Hoe kan het bedrijf bij het toevoegen van deze gegevens aan de relationele database met zo min mogelijk bewerkingen voldoen aan dit relationele model?

- A) Bij het opnemen van de reisproducten voor de reisgids er voor zorgen dat de waarde 3067 voor gidsnr al voorkomt in de tabel REISGIDS door eerst de betreffende gids vast te leggen
- B) Eerst voor ieder van de reisproducten een rij in de tabel REISPRODUCT toevoegen en daarna pas voor de reisgids een rij in de tabel REISGIDS
- C) Voor het attribuut gidsnr in de tabel REISPRODUCT NULL opnemen en pas na toevoegen van de reisgids in iedere rij een waarde voor dit attribuut opnemen

**17 / 30**

Een vereniging van muziekkorpsen houdt de ledenadministratie bij in een relationele database, die als volgt is gedefinieerd:

VERENIGING (verenigingnaam, contributie, betaalperiode)

LID (lidnaam, adres, postcode, plaats, ingangsdatum, betaalperiode)

Van de relatie VERENIGING zal één tupel worden ingevoerd met als key (sleutel) 'St. Caecilia'. Betaalperiode in een tupel van relatie LID kan NULL zijn, want dan geldt voor dat lid de betaalperiode uit de relatie VERENIGING.

Waarom moet de vereniging zich **volgens het gegeven relationele model** houden bij het invullen van waarden voor de attributen betaalperiode?

- A) Als bij LID betaalperiode een waarde heeft, dan mag bij VERENIGING betaalperiode NULL zijn.
- B) Als bij LID betaalperiode NULL is, dan moet bij VERENIGING een waarde ingevuld zijn.
- C) Als bij VERENIGING betaalperiode NULL is, dan moet bij LID een waarde ingevuld zijn.
- D) Bij VERENIGING en LID maakt het niet uit of betaalperiode ingevuld is of niet.

**18 / 30**

Welke eigenschap is **niet** van toepassing op de werking van een relationeel databasemanagementsysteem (DBMS)?

- A) Bij het leggen van relaties tussen records zijn de verwijzingen zichtbaar voor de gebruiker.
- B) Bij het opslaan van gegevens kan hooguit één record tegelijk toegevoegd worden.
- C) Conceptuele beschrijvingen kunnen views bevatten.
- D) Externe beschrijvingen kunnen tabellen bevatten.

**19 / 30**

De organisatoren van een wandeltocht hebben een relationele database met een tabel gebaseerd op het onderstaande deel van een relationele schema:

WANDELTOCHT (datum, deelnemer, kilometers)

Welke SELECT-opdracht geeft antwoord op de vraag welke deelnemers in één wandeltocht meer dan 50 km gelopen hebben?

- A) 

```
SELECT deelnemer
FROM wandeltocht
GROUP BY deelnemer
HAVING SUM(kilometers) > 50
```
- B) 

```
SELECT deelnemer
FROM wandeltocht
WHERE kilometers > 50
```
- C) 

```
SELECT deelnemer
FROM wandeltocht
WHERE SUM(kilometers) > 50
```

20 / 30

Een zangvereniging houdt de deelname van zangers aan een voorstelling in een database bij. Hieronder staan twee gebruikte tabellen:

ZANGER

zangernr	naam	soort
1	Koos	bariton
2	Karel	bas
3	Jan	sopraan
4	Judith	sopraan
5	Kelly	bas
6	Toos	sopraan

VOORSTELLING

voorstellingsnr	zangernr
4	2
4	3
5	1
5	2
5	4
6	1
6	5
7	1
7	3
7	4

Welke SELECT-opdracht geeft als uitkomst de zanger die nergens aan heeft meegedaan?

- A) `SELECT *`  
`FROM zanger`  
`WHERE zangernr IN`  
`(SELECT zangernr`  
`FROM voorstelling)`
- B) `SELECT *`  
`FROM zanger`  
`WHERE zangernr IN`  
`(SELECT zangernr`  
`FROM voorstelling`  
`AND zanger.zangernr < > voorstelling.zangernr)`
- C) `SELECT *`  
`FROM zanger`  
`WHERE zangernr NOT IN`  
`(SELECT zangernr`  
`FROM voorstelling)`
- D) `SELECT *`  
`FROM zanger`  
`WHERE zangernr NOT IN`  
`(SELECT zangernr`  
`FROM voorstelling, zanger`  
`WHERE voorstelling.zangernr = zanger.zangernr)`

21 / 30

Een boerenbedrijf heeft een tabel METING voor het volgen van informatie over de koeien. De rijen 100 tot en met 149 gaan over koe 4.

METING

metingnr	soort	eenheid	waarde
99	urine	liter	15
100	temperatuur	°C	38
101	temperatuur	°C	39
102	melk	liter	24
103	melk	liter	22
104	melk	liter	26

enzovoort

Welke SELECT-opdracht geeft als uitkomst hoeveel melk koe 4 geproduceerd heeft?

- A) 

```
SELECT SUM(waarde)
FROM meting
WHERE eenheid = 'liter'
AND metingnr > 99
AND metingnr < 150
```
- B) 

```
SELECT SUM(waarde)
FROM meting
WHERE soort = 'melk'
AND metingnr > 99
AND metingnr < 150
```
- C) 

```
SELECT soort, SUM(waarde)
FROM meting
WHERE soort = 'melk'
AND metingnr > 99
AND metingnr > 150
```
- D) 

```
SELECT SUM(waarde)
FROM meting
WHERE soort = 'melk'
AND metingnr > 99
OR metingnr < 150
```

## 22 / 30

Een bedrijf houdt zijn salarisadministratie bij in een database gebaseerd op het volgende relationele schema:

WERKNEMER (werknemernr, naam, achternaam, geboortedatum, functie, salaris)  
 AFDELING (<wknemernr>, afdelingnaam, manager\_naam)  
 UITBETALING (<wknemernr>, maand, jaar, bedrag, status, betaald j/n)  
 PENSIOEN (<wknemernr>, jaar, verzekerd j/n, extra\_pensioen j/n)

In het kader van een CAO-aanpassing krijgen alle werknemers met een maandsalaris lager dan € 3000,- een verhoging van € 50,- per maand.

Hoe luidt de UPDATE-opdracht hiervoor?

- A) UPDATE Salaris  
FROM Werknemer  
SET Salaris = Salaris + 50  
WHERE Salaris < 3000
- B) UPDATE Werknemer  
SET Salaris = Salaris + 50  
WHERE Salaris < 3000
- C) UPDATE Werknemer  
SET Salaris TO Salaris + 50  
WHERE Salaris < 3000

## 23 / 30

Een vereniging voor muziekkorpsen houdt een database bij van de scores van de verschillende korpsen op hun concoursen. Ze houden ook bij of het korps een amateurkorps (A) of een professioneel korps (P) is.

KORPS

Naam	Klasse	Score
St. Jozef	A	332
St. David	A	328
Eendracht	P	330
Adevendo	P	326
KMK	A	334
JWFK	A	332
Cecilia	P	328

Met welke opdracht kunnen alle korpsen met een score van 330 verwijderd worden?

- A) DELETE FROM Korps  
HAVING Score = 330
- B) DELETE FROM Korps  
WHERE Score = 330
- C) DELETE Naam, Klasse, Score  
FROM Korps  
WHERE Score = 330

24 / 30

In een relationele database is de tabel TAALBEHEERSING opgenomen:

TAALBEHEERSING

Naam	Taal	Niveau
Anneke	Engels	Zeer goed
Anneke	Frans	Matig
Karel	Duits	Goed
Karel	Engels	Zeer goed
Willem	Duits	Goed
Willem	Engels	Matig

Het volgende SQL-statement wordt uitgevoerd:

```
SELECT Naam, Taal, Niveau
FROM taalbeheersing
WHERE Taal IN
(SELECT Taal
FROM taalbeheersing
WHERE Naam = 'Anneke')
```

Uit hoeveel rijen bestaat het resultaat van dit SQL-statement?

- A) 1
- B) 2
- C) 4
- D) 5

**25 / 30**

Een bedrijf houdt zijn wagenpark bij in een relationele database. Bijgehouden wordt welke auto ze hebben, hoeveel wielen er bij de auto horen (maximaal acht bij een set van zomerbanden en winterbanden), en wanneer de auto is aangekocht.

WAGENPARK (auto, wielen, aankoop)

Hierbij is auto een tekstveld van maximaal tien letters, wielen is een geheel getal, aankoop is een datum. Auto is de primary key (primaire sleutel) en alleen aankoop is optioneel.

Met welke SQL-opdracht wordt bovengenoemde tabel gemaakt?

- A) 

```
CREATE TABLE wagenpark
(auto CHAR(10) NOT NULL,
wielen INTEGER NOT NULL,
aankoop DATE,
PRIMARY KEY(auto))
```
- B) 

```
CREATE TABLE wagenpark
(auto CHAR(10) NOT NULL,
wielen NUMBER NOT NULL,
aankoop DATE,
PRIMARY KEY(auto))
```
- C) 

```
CREATE TABLE wagenpark
(auto CHAR(10) PRIMARY KEY NOT NULL,
wielen INTEGER NOT NULL,
aankoop DATE)
```
- D) 

```
CREATE TABLE wagenpark
(auto CHAR(10) PRIMARY KEY NOT NULL,
wielen NUMBER NOT NULL,
aankoop DATE)
```



26 / 30

Voor de jaarlijkse meterstanden heeft een energiemaatschappij een relationeel schema gemaakt:

```
VERBRUIKSADRES (postcode, huisnummer, straatnaam, plaats)
METING (<postcode, huisnummer>, jaar, kWh, gas, water, wie)
```

Hiervoor zijn de volgende twee CREATE TABLE-opdrachten gebruikt om tabellen te maken in de database:

```
CREATE TABLE VERBRUIKSADRES
(postcode CHAR(6),
huisnummer INTEGER NOT NULL,
straatnaam VARCHAR(100) NOT NULL,
plaats VARCHAR(60) NOT NULL,
PRIMARY KEY(postcode, huisnummer))
```

```
CREATE TABLE METING
(postcode CHAR(6),
huisnummer INTEGER,
jaar SMALLINT NOT NULL,
kWh SMALLINT NOT NULL,
gas SMALLINT NOT NULL,
water SMALLINT NOT NULL,
wie CHAR(1) NOT NULL,
PRIMARY KEY(postcode, huisnummer, jaar),
FOREIGN KEY(postcode, huisnummer) REFERENCES verbruiksadres
ON DELETE CASCADE
ON UPDATE CASCADE)
```

Welk gevolg heeft het uitvoeren van de opdracht `DROP TABLE verbruiksadres`?

- A) De opdracht leidt tot het verwijderen van alle rijen in de tabel VERBRUIKSADRES. De tabel zelf blijft bestaan.
- B) De opdracht leidt zonder meer tot het verwijderen van de tabel VERBRUIKSADRES en de daarin opgenomen rijen.
- C) De opdracht wordt nooit uitgevoerd omdat die in strijd is met de referentiële integriteit.
- D) De tabel VERBRUIKSADRES wordt alleen verwijderd als er geen rijen in de tabel METING voorkomen.

27 / 30

Een bedrijf voor feestartikelen heeft de volgende tabel in de database:

```
ORDERREGEL (<ordernr>, regelnr, artikelnaam, bestelaantal, leverdatum)
```

Het ordernr en bestelaantal kunnen grote getallen bevatten, maar het aantal regels per order is beperkt. Geen enkel artikel heeft een naam van meer dan 25 tekens. De leverdatum is niet altijd bekend. Het ordernummer moet uiteraard ook bekend zijn in de relatie 'order'.

Om voor deze relatie in het databasemanagementsysteem (DBMS) een tabel aan te maken wordt de volgende SQL-opdracht uitgevoerd:

```
CREATE TABLE orderregel  
(ordernr INTEGER NOT NULL,  
regelnr SMALLINT NOT NULL,  
artikelnaam CHAR(25) NOT NULL,  
bestelaantal INTEGER NOT NULL,  
leverdatum DATE,  
PRIMARY KEY(ordernr),  
PRIMARY KEY(regelnr),  
FOREIGN KEY(ordernr) REFERENCES order  
ON DELETE RESTRICT  
ON UPDATE RESTRICT)
```

Waarom is dit SQL-statement **onjuist**?

- A) Bij leverdatum ontbreekt NULL.
- B) Er mag slechts één keer PRIMARY KEY worden gebruikt.
- C) Het gegevenstype van artikelnaam moet VARCHAR(25) zijn.
- D) Na ON UPDATE mag RESTRICT niet worden gebruikt.

**28 / 30**

Voor de jaarlijkse meterstanden heeft een energiemaatschappij een relationeel schema:

```
VERBRUIKSADRES (postcode, huisnummer, straatnaam, plaats)
METING (<postcode, huisnummer>, jaar, kWh, gas, water, wie)
```

Hiervoor zijn de volgende twee CREATE TABLE-opdrachten gebruikt om tabellen te maken in de database:

```
CREATE TABLE VERBRUIKSADRES
(postcode CHAR(6),
huisnummer INTEGER NOT NULL,
straatnaam VARCHAR(100) NOT NULL,
plaats VARCHAR(60) NOT NULL,
PRIMARY KEY(postcode, huisnummer))
```

```
CREATE TABLE METING
(postcode CHAR(6),
huisnummer INTEGER,
jaar SMALLINT NOT NULL,
kWh SMALLINT NOT NULL,
gas SMALLINT NOT NULL,
water SMALLINT NOT NULL,
wie CHAR(1) NOT NULL,
PRIMARY KEY(postcode, huisnummer, jaar),
FOREIGN KEY(postcode, huisnummer) REFERENCES verbruiksadres
ON DELETE CASCADE
ON UPDATE CASCADE)
```

Wat is het gevolg van het opnemen van de UPDATE-regel in de foreign key (vreemde sleutel)?

- A) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in de tabel METING naar NULL-waarden zorgen dat de bijbehorende rij in de tabel VERBRUIKSADRES wordt verwijderd.
- B) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel METING worden ook doorgevoerd in de rij waarnaar verwezen wordt in de tabel VERBRUIKSADRES.
- C) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel VERBRUIKSADRES hebben hooguit gevolgen voor één rij in de tabel METING.
- D) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel VERBRUIKSADRES worden ook doorgevoerd in de rijen van de tabel METING die naar de rij in de tabel VERBRUIKSADRES verwijzen.

**29 / 30**

Gegeven is de volgende reeks autorisatie-opdrachten:

```
GRANT DELETE ON rapport TO Peters, Jansen, Kuyten, Soons
REVOKE DELETE ON rapport FROM Jansen, Kuyten
GRANT ALL ON rapport TO Kuyten
GRANT INSERT ON rapport TO Peters
GRANT ALTER ON rapport TO Soons
```

Wie is **niet** geautoriseerd om rapporten te verwijderen nadat bovenstaande opdrachten in de weergegeven volgorde zijn uitgevoerd?

- A) Jansen
- B) Kuyten
- C) Peters
- D) Soons

**30 / 30**

Gegeven is de volgende tabel in een relationele database:

```
LOGIN (loginnummer, gebruikersnaam, wachtwoord, toegangsniveau)
```

De eigenaar van de database voert de volgende opdracht uit:

```
REVOKE UPDATE ON LOGIN FROM user6
```

Wat verandert er voor gebruiker user6?

- A) User6 mag de structuur van de tabel LOGIN niet meer wijzigen.
- B) User6 mag geen nieuwe rechten meer vastleggen in de tabel LOGIN.
- C) User6 mag geen rechten uit de tabel LOGIN meer aanpassen.
- D) User6 mag haar rechten voor de tabel LOGIN niet meer doorgeven.

# Antwoordsleutel

1 / 30

Wat is het voordeel van een index voor een tabel in een database?

- A) Een extra sleutel, de zogenaamde alternate key
  - B) Een extra zoekmogelijkheid voor de betreffende tabel
  - C) Een snellere toegang tot de gegevens
- 
- A) Incorrect. Een index is geen alternate key, maar wel een snellere toegang tot de gegevens.
  - B) Incorrect. Een index levert geen extra zoekmogelijkheid op, maar wel een snellere toegang tot de gegevens.
  - C) Correct. Een index zorgt voor een snellere toegang tot de gegevens. (Literatuur: A, Hoofdstuk 2.2)

2 / 30

Met welk onderdeel van een databasemanagementsysteem (DBMS) wordt de structuur van een database vastgelegd?

- A) Data Control Language (DCL)
  - B) Data Definition Language (DDL)
  - C) Data Manipulation Language (DML)
- 
- A) Incorrect. Met DCL worden autorisatieregels vastgelegd.
  - B) Correct. DDL legt de structuur van de database vast. (Literatuur: A, Hoofdstuk 2.2)
  - C) Incorrect. Met DML kunnen gegevens opgehaald en gewijzigd worden.

3 / 30

Een database is beschreven op basis van ANSI/SPARC.

Waar zorgt gegevensonafhankelijkheid voor?

- A) Dat een programma dat de database raadpleegt, moet worden veranderd als er een andere indexeringstechniek wordt toegepast
  - B) Dat een programma niet hoeft te worden gewijzigd wanneer er iets verandert in de structuur van de beschrijving op conceptueel niveau van de database
  - C) Dat een programma niet sneller of langzamer verwerkt zal worden wanneer een wijziging in de interne beschrijving van de database is gedaan
  - D) Dat een wijziging in de structuur van de database alleen gevolgen heeft als deze wijziging voorkomt in de interne beschrijving van de database van het programma
- 
- A) Incorrect. Als er sprake is van fysieke gegevensonafhankelijkheid zullen wijzigingen in fysieke zaken, zoals die beschreven staan in de beschrijving van het interne niveau geen gevolgen hebben voor de code van een programma.
  - B) Correct. Gegevensonafhankelijkheid heeft als doel dat een programma niet hoeft te wijzigen bij wijzigingen van de beschrijving op conceptueel niveau van de database. (Literatuur: A, Hoofdstuk 2.2)
  - C) Incorrect. Gegevensonafhankelijkheid zorgt er alleen voor dat de code van een programma niet hoeft te worden aangepast. Gegevensonafhankelijkheid zegt niets over de snelheid van verwerking of over de techniek waarvan de database gebruikmaakt.
  - D) Incorrect. De wijziging zal alleen gevolgen hebben als deze onderdeel is van de beschrijving van het externe niveau waar een programma gebruik van maakt.

4 / 30

Een databasemanagementsysteem (DBMS) is gebaseerd op ANSI/SPARC.

Wat is het doel van het interne niveau?

- A) Het beschrijft tot welke gegevens de toegang van een gebruiker beperkt is zodat de gebruiker niet alle gegevens te zien kan krijgen.
  - B) Het beschrijft welke recordtypen er zijn, hoe de lay-out daarvan is, welke identificaties er zijn, welke waardebereiken mogen voorkomen.
  - C) Het zorgt ervoor dat duidelijk is met welke blok grootte en welke indexeringstechnieken gegevens worden opgeslagen.
  - D) Het zorgt ervoor dat gebruikers ongeacht hun views dezelfde gegevens in dezelfde formats aangeboden krijgen.
- 
- A) Incorrect. Het externe niveau is onder andere bedoeld om te beperken wat de gebruiker aan gegevens aangeboden krijgt, maar het DBMS mag een gebruiker ook alle gegevens in de database laten zien.
  - B) Incorrect. Bij ANSI/SPARC beschrijft het conceptuele niveau de structuur van de recordtypen en de regels zoals identificaties en waardebereiken.
  - C) Correct. Bij ANSI/SPARC gaat het interne niveau over de techniek van het opslaan van gegevens. (Literatuur: A, Hoofdstuk 2.2)
  - D) Incorrect. Het externe niveau kan meer dan één beschrijving van de database bevatten, met afwijkende wijzen waarop de gegevens aangeboden moeten worden.

5 / 30

Wat wordt in het relationeel model bedoeld met het begrip 'relatie'?

- A) De beschrijving van een tabel in combinatie met alle rijen met waarden uit die tabel
  - B) Een gegeven in een tabel dat verwijst naar een rij in een andere tabel
  - C) Een volledige rij met waarden uit een tabel
- 
- A) Correct. Een relatie bestaat uit de inhoud van de tabel in combinatie met de beschrijving, waaruit de betekenis van die inhoud blijkt. (Literatuur: A, Hoofdstuk 3.1)
  - B) Incorrect. Dit is een omschrijving van het begrip 'vreemde sleutel'. Een relatie bestaat uit de inhoud van de tabel in combinatie met de beschrijving, waaruit de betekenis van die inhoud blijkt.
  - C) Incorrect. Dit is een omschrijving van het begrip 'tupel'. Een relatie bestaat uit de inhoud van de tabel in combinatie met de beschrijving, waaruit de betekenis van die inhoud blijkt.

6 / 30

Eén van de gegevens die een bedrijf wil vastleggen is de naam van hun leverancier, Jansen.

Welk begrip is **niet** van toepassing op de naam Jansen?

- A) Attribuutnaam
  - B) Attribuutwaarde
  - C) Deel van een rij
  - D) Deel van een tupel
- 
- A) Correct. De attribuutnaam is de naam waaronder de betreffende gegevens (attribuutwaarden) bekend zullen zijn. 'Leveranciersnaam' zou in dit geval een mogelijke attribuutnaam zijn, maar Jansen is een attribuutwaarde en tegelijkertijd een deel van een rij én een deel van een tupel. (Literatuur: A, Hoofdstuk 3.1)
  - B) Incorrect. Jansen is een attribuutwaarde. De naam van het attribuut zou Leveranciersnaam kunnen zijn.
  - C) Incorrect. Jansen is een attribuutwaarde en daarmee ook onderdeel van een rij.
  - D) Incorrect. Jansen is een attribuutwaarde en daarmee ook een deel van een tupel.

7 / 30

Met welk soort regels wordt vastgelegd hoe de nieuwe waarde van een veld afhankelijk is van de huidige waarde?

- A) Afhankelijkheidsregels
  - B) Afleidingsregels
  - C) Domeinregels
  - D) Transitierregels
- A) Incorrect. Afhankelijkheidsregels geven aan dat de waarde van attributen en/of tupels elkaar onderling beïnvloeden. Dit is echter op een statische manier en niet hoe waarden elkaar mogen opvolgen. Een voorbeeld van een afhankelijkheidsregel is dat wanneer een attribuut aangeeft of iemand een kind, volwassene of bejaarde is, dit de waarde beperkt voor het attribuut geboortedatum.
- B) Incorrect. Afleidingsregels geven aan hoe de waarden van attributen uit de waarden van andere attributen kunnen worden afgeleid.
- C) Incorrect. Domeinregels geven aan dat een gebruikte waarde tot een vooraf gedefinieerd domein moet behoren.
- D) Correct. Transitierregels geven aan hoe de overgang van de ene waarde naar de andere waarde wordt bepaald. (Literatuur: A, Hoofdstuk 3.3)

8 / 30

Een museum beschikt over een groot aantal schilderijen. Voor de administratie van deze schilderijen wordt een relationele database voorgesteld met de volgende relaties:

```
SCHILDER (schildernaam, geboortedatum, datum overlijden, geboorteplaats,
geboorteland)
SCHILDERIJ (<schildernaam>, volgnummer, titel, afmetingen, materiaal
ondergrond, verfsoort)
```

Omdat de schilder van een schilderij niet altijd bekend is, wordt voorgesteld om in die gevallen de naam van de schilder leeg te laten en alleen een volgnummer toe te kennen.

Is dit voorstel mogelijk?

- A) Ja, mits voor deze schilderijen een aparte nummerreeks wordt gedefinieerd die alleen voor deze schilderijen gebruikt mag worden.
  - B) Ja, omdat schildernaam een foreign key (vreemde sleutel) is in de relatie SCHILDERIJ, mag deze leeg worden gelaten.
  - C) Nee, omdat de naam van de schilder niet bekend is, zal dit schilderij niet in de database kunnen worden opgenomen.
  - D) Nee, omdat de primary keys (primaire sleutels) in een relationele database altijd moeten worden ingevuld.
- A) Incorrect. In een relationele database mogen sleutelwaarden niet NULL zijn, dus deze oplossing is niet toegestaan.
- B) Incorrect. Schildernaam is de primary key in de relatie SCHILDER en kan daarom niet leeg worden gelaten.
- C) Incorrect. Het schilderij kan worden opgenomen met bijvoorbeeld als naam van de schilder: Onbekend. In de tabel met schilders moet dan ook een schilder met de naam Onbekend worden opgenomen.
- D) Correct. In een relationele database mogen primary keys niet NULL zijn. (Literatuur: A, Hoofdstuk 3.4)



9 / 30

Een handelsmaatschappij heeft voor haar verkopen een databasemodel met de volgende relaties:

KLANT (klantnummer, naam, adres, plaats, telefoonnummer)

FACTUUR (<klantnummer>, factuurnummer, factuurdatum, <artikelnummer>, aantal)

ARTIKEL (artikelnummer, artikelomschrijving, bedrag excl. BTW)

De handelsmaatschappij wil met een database die op dit model is gebaseerd een factuurbedrag inclusief (incl.) BTW kunnen berekenen.

Moet de handelsmaatschappij daarvoor iets veranderen aan dit databasemodel?

- A) Ja, er moet een regelnummer aan het factuurnummer worden toegevoegd zodat een factuur uit meerdere regels kan bestaan.
  - B) Ja, het bedrag incl. BTW moet worden toegevoegd aan de relatie ARTIKEL.
  - C) Nee, want het factuurbedrag incl. BTW kan met behulp van een afleidingsregel worden berekend.
  - D) Nee, want het factuurbedrag incl. BTW kan met behulp van een transitieregel worden berekend.
- 
- A) Incorrect. Met een extra regelnummer kunnen er meerdere artikelen op een factuur komen, maar dat was de vraag niet.
  - B) Incorrect. Het bedrag incl. BTW kan met een afleidingsregel waarin het BTW-percentage is opgenomen worden berekend.
  - C) Correct. Het model hoeft niet te worden aangepast, omdat het factuurbedrag incl. BTW kan worden berekend door middel van een afleidingsregel. Het factuurbedrag incl. BTW bestaat namelijk uit het bedrag excl. BTW x aantal x BTW %. (Literatuur: A, Hoofdstuk 3.2)
  - D) Incorrect. Het factuurbedrag incl. BTW kan inderdaad berekend worden, maar dit is door middel van een afleidingsregel, niet door een transitieregel.

10 / 30

Een dansschool houdt in een database bij:

- een overzicht van de leerlingen;
- met wie een leerling graag danst;
- wie met wie al een danspaar vormt.

De database is gebaseerd op het volgende relationele model:

```
LEERLING (<leerlingnummer, naam, geboortedatum, geslacht, roepnaam>
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>)
DANSPAAR (<leerlingnummer3, leerlingnummer4>)
```

Ondanks het opnemen van een regel in de database dat de leerlingen die een danspaar vormen vijf jaar of minder in leeftijd mogen verschillen, kan het toch voorkomen dat de leeftijd van een danspaar meer dan vijf jaar verschilt.

Waarom kan dit in de database voorkomen?

- A) Omdat de regel maar op één tupel betrekking mag hebben en de leerlingen in verschillende tupels worden vastgelegd
  - B) Omdat de regel wel op meer dan één tupel betrekking mag hebben, maar niet op tupels van dezelfde relatie
  - C) Omdat het attribuut geboortedatum in de relatie LEERLING niet verplicht is
  - D) Omdat in de relatie DANSPAAR de attributen geboortedatum3 van de eerste leerling en geboortedatum4 van de tweede leerling ontbreken
- 
- A) Incorrect. Een regel mag betrekking hebben op meer tupels van één relatie of van meerdere relaties.
  - B) Incorrect. Een regel mag op meer tupels van dezelfde relatie betrekking hebben.
  - C) Correct. Het is alleen mogelijk regels af te dwingen met waarden die bekend zijn. Voor geboortedatum kan een waarde ontbreken. In dat geval kan de regel niet worden afgedwongen. (Literatuur: A, Hoofdstuk 3.1 t/m 3.4 en 5.3)
  - D) Incorrect. Het opnemen van deze attributen helpt niet om het probleem op te lossen. Het afdwingen van de regel is alleen mogelijk met waarden die bekend zijn. Waardes kunnen ontbreken voor zowel geboortedatum3 als geboortedatum4. Ook hoeven de geboortedata niet de geboortedata van de leerlingen te zijn. Dat dwingt het schema niet af.

11 / 30

Een dansschool houdt van hun leerlingen bij:

- een overzicht van persoonsgegevens;
- met wie een leerling graag danst (de leerling aangeduid met leerlingnummer1 danst graag met de leerling aangeduid met leerlingnummer2);
- wie met wie een danspaar vormt (de leerling aangeduid met leerlingnummer3 vormt een danspaar met de leerling aangeduid met leerlingnummer4).

De database is gebaseerd op het volgende relationele model:

```
LEERLING (leerlingnummer, naam, geboortedatum, geslacht, roepnaam)
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>)
DANSPAAR (<leerlingnummer3, leerlingnummer4>)
```

Leerling 1011 en 1064 dansen graag met elkaar en zij vormen samen ook al een danspaar. Om dit aan te geven worden in de relatie DANSTGRAAGMET en in de relatie DANSPAAR tupels vastgelegd.

Welke tupels moeten hiervoor worden vastgelegd?

- A) Eén tupel in de relatie DANSPAAR en één tupel in de relatie DANSTGRAAGMET
  - B) Eén tupel in de relatie DANSPAAR en twee tupels in de relatie DANSTGRAAGMET
  - C) Twee tupels in de relatie DANSPAAR en één tupel in de relatie DANSTGRAAGMET
  - D) Twee tupels in de relatie DANSPAAR en twee tupels in de relatie DANSTGRAAGMET
- A) Incorrect. Er is één tupel in de relatie DANSTGRAAGMET nodig voor ieder vastleggen van wie met wie anders graag danst. Voor twee dansers heb je daarvoor twee tupels nodig. Dit komt omdat als de een met de ander wil dansen, de ander nog niet graag met de eerste wil dansen. Een danspaar vormen is wel over en weer en kan daarom door één tupel worden weergegeven. In het antwoord is er dus één tupel in de relatie DANSTGRAAGMET te weinig.
- B) Correct. Er is één tupel in de relatie DANSTGRAAGMET nodig voor ieder vastleggen van wie met wie anders graag danst. Voor twee dansers heb je daarvoor twee tupels nodig. Dit komt omdat als de een met de ander wil dansen, de ander nog niet graag met de eerste wil dansen. Een danspaar vormen is wel over en weer en kan daarom door één tupel worden weergegeven. (Literatuur: A, Hoofdstuk 3.1 en 3.2)
- C) Incorrect. Er is één tupel in de relatie DANSTGRAAGMET nodig voor ieder vastleggen van wie met wie anders graag danst. Voor twee dansers heb je daarvoor twee tupels nodig. Dit komt omdat als de een met de ander wil dansen, de ander nog niet graag met de eerste wil dansen. Een danspaar vormen is wel over en weer en kan daarom door één tupel worden weergegeven. In het antwoord is er dus één tupel in de relatie DANSTGRAAGMET te weinig en één tupel in de relatie DANSPAAR te veel.
- D) Incorrect. Er is één tupel in de relatie DANSTGRAAGMET nodig voor ieder vastleggen van wie met wie anders graag danst. Voor twee dansers heb je daarvoor twee tupels nodig. Dit komt omdat als de een met de ander wil dansen, de ander nog niet graag met de eerste wil dansen. Een danspaar vormen is wel over en weer en kan daarom door één tupel worden weergegeven. In het antwoord is er dus één tupel in de relatie DANSPAAR te veel.

**12 / 30**

Een verhuurder hanteert een database waarin gegevens van panden en subsidies zijn vastgelegd. Per pand kan één keer per jaar subsidie worden aangevraagd. Het aangevraagde bedrag wordt ingevuld zodra de subsidie wordt aangevraagd, het toegekende bedrag pas op het moment dat de subsidie is toegekend.

PAND (pandnr, pandnaam, adres, plaats, huur)

SUBSIDIE (<pandnr>, jaar, aangevraagdbedrag, toegekendbedrag, subsidiernr, plaats)

VERHUURDER (naam, pandnr, jaar)

Waarom moet er voor iedere relatie SUBSIDIE een relatie PAND voorkomen?

- A) Omdat er voor ieder aangevraagd bedrag een pandnr in de relatie SUBSIDIE zit
  - B) Omdat er voor ieder toegekend bedrag een pandnr in de relatie SUBSIDIE zit
  - C) Omdat in de relaties PAND en SUBSIDIE het attribuut pandnr deel is van de primary key (primaire sleutel)
  - D) Omdat in de relaties PAND en SUBSIDIE het attribuut pandnr voorkomt
- 
- A) Incorrect. Er zit weliswaar een attribuut pandnr in de relatie SUBSIDIE, maar daarmee wordt een relatie PAND nog niet voorkomen.
  - B) Incorrect. Er zit weliswaar een attribuut pandnr in de relatie SUBSIDIE, maar daarmee wordt een relatie PAND nog niet voorkomen.
  - C) Correct. Er ligt een verplichte relatie tussen PAND en SUBSIDIE, omdat in beide relaties het pandnr onderdeel uitmaakt van de primary key. (Literatuur: A, Hoofdstuk 3.2, 3.3 en 5.3)
  - D) Incorrect. Het aanwezig zijn van eenzelfde attribuut in twee relaties is onvoldoende voor de samenhang. Deze samenhang kan alleen bereikt worden door middel van een foreign key (vreemde sleutel).

### 13 / 30

Een verhuurder van appartementen heeft een database gebaseerd op het volgende relationele model:

```
VERHUUROBJECT (objectnummer, naam, omschrijving, soort, aantal_personen,  
keuken_apart j/n, zwembad j/n, eigen_terras j/n, verhuurdernummer)  
OBJECTPERIODE (<objectnummer>, weeknummer)  
BOEKING (<objectnummer>, klantnummer, weeknummer, status, betaald j/n)
```

De verhuurder plant de weken dat het object verhuurd kan worden. Hiervoor worden tupels opgenomen in de relatie OBJECTPERIODE.

Wat moet worden gedaan om te voorkomen dat er verhuurobjecten zijn waarvan de verhuurder **niet** bekend is?

- A) Alleen de primary key (primaire sleutel) in de relatie VERHUUROBJECT hoeft veranderd te worden, zodat deze primary key bestaat uit de attributen objectnummer en verhuurdernummer.
  - B) Een relatie VERHUURDER moet opgenomen worden en het attribuut verhuurdernummer in de relatie VERHUUROBJECT moet een foreign key (vreemde sleutel) worden naar de relatie VERHUURDER.
  - C) Het attribuut verhuurdernummer in de relatie VERHUUROBJECT moet een verplicht attribuut worden.
- A) Incorrect. De primary key zou zo aangepast kunnen worden en dan is een verhuurdernummer verplicht, omdat het deel uitmaakt van de primary key. Echter, de relatie VERHUUROBJECT strookt dan niet meer met de rest van het model. Vanuit de relaties OBJECTPERIODE en BOEKING wordt dan verkeerd verwezen naar de relatie VERHUUROBJECT.
- B) Incorrect. Het voorgestelde is een mooie toevoeging, maar het blijft mogelijk dat er voor de foreign key verhuurdernummer in de relatie VERHUUROBJECT niets is opgenomen. Een foreign key hoeft geen waarde te bevatten. Dan kunnen er toch nog verhuurobjecten zijn waarvan de verhuurder niet bekend is.
- C) Correct. Het verplicht maken van het attribuut verhuurdernummer zorgt ervoor dat er altijd een verhuurdernummer moet zijn en daarmee een verhuurder bekend is. (Literatuur: A, Hoofdstuk 3.1 t/m 3.3 en 5.3)

14 / 30

Een dansschool houdt in een database bij:

- een overzicht van de leerlingen;
- met wie een leerling graag danst;
- wie met wie al een danspaar vormt.

De database is gebaseerd op het volgende relationele model:

```
LEERLING (leerlingnummer, naam, geboortedatum, geslacht, roepnaam)
DANSTGRAAGMET (<leerlingnummer1>, <leerlingnummer2>,
gewenst_geslacht_leerling1, gewenst_geslacht_leerling2)
DANSPAAR (<leerlingnummer3, leerlingnummer4>)
```

Ondanks het in een tuple van de relatie DANSTGRAAGMET opnemen van verschillende waarden voor de attributen `gewenst_geslacht_leerling1` en `gewenst_geslacht_leerling2` is het toch mogelijk dat de twee in de tuple aangeduide leerlingen van gelijk geslacht zijn en een danspaar vormen.

Waardoor komt dit?

- A)** De waarden voor de attributen `gewenst_geslacht_leerling1` en `gewenst_geslacht_leerling2` hoeven niet over respectievelijk de leerling met het `leerlingnummer1` en de leerling met het `leerlingnummer2` te gaan.
  - B)** Ook al zijn de waarden voor de attributen `gewenst_geslacht_leerling1` en `gewenst_geslacht_leerling2` verschillend, de geslachten in de verwezen tupels van de relatie LEERLING mogen gelijk zijn.
  - C)** Voor het attribuut `leerlingnummer3` in de relatie DANSPAAR hoeft niet te gelden dat de verwezen leerling als `leerlingnummer1` in de relatie DANSTGRAAGMET voorkomt.
  - D)** Twee leerlingen aangeduid met de combinatie van de attributen `leerlingnummer3` en `leerlingnummer4` hoeven niet voor te komen als een combinatie van leerlingen aangeduid met `leerlingnummer1` en `leerlingnummer2`.
- 
- A)** Incorrect. Wat er gezegd wordt is wel waar, maar niet de reden waarom het danspaar van gelijk geslacht kan zijn. Dit komt doordat het relationele schema niet afdwingt dat de geslachten in DANSTGRAAGMET overeenkomen met de geslachten in de relatie LEERLING.
  - B)** Correct. Dit komt doordat het relationele schema niet afdwingt dat de geslachten in DANSTGRAAGMET overeenkomen met de geslachten in de relatie LEERLING. (Literatuur: A, Hoofdstuk 3.1 t/m 3.4 en 5.3)
  - C)** Incorrect. Dat is juist wel het geval. Dat komt omdat `leerlingnummer3` en `leerlingnummer4` samen een foreign key (vreemde sleutel) vormen die verwijst naar de primary key (primaire sleutel) van de relatie DANSTGRAAGMET. Bij foreign keys geldt dat de waardencombinaties ook moeten voorkomen bij de primary key waarnaar verwezen wordt.
  - D)** Incorrect. Dat is juist wel het geval. Dat komt omdat `leerlingnummer3` en `leerlingnummer4` samen een foreign key (vreemde sleutel) vormen die verwijst naar de primary key (primary key) van de relatie DANSTGRAAGMET. Bij een foreign key geldt dat de waardencombinaties ook moeten voorkomen bij de primary key waarnaar verwezen wordt.

**15 / 30**

Een verhuurder van appartementen heeft een database gebaseerd op het volgende relationele model:

VERHUUROBJECT (objectnummer, naam, omschrijving, soort, aantal\_personen, keuken\_apart j/n, zwembad j/n, eigen\_terras j/n, verhuurdersnummer)  
OBJECTPERIODE (<objectnummer>, weeknummer)  
BOEKING (<objectnummer>, klantnummer, weeknummer, status, betaald j/n)

De verhuurder plant de weken dat het object verhuurd kan worden. Hiervoor worden tupels opgenomen in de relatie OBJECTPERIODE.

Hoe kan bereikt worden dat van alle boekingen de klant bekend is?

- A) Een attribuut klantnummer toevoegen aan de relatie OBJECTPERIODE.
  - B) Een relatie KLANT opnemen met een foreign key (vreemde sleutel) objectnummer + klantnummer die verwijst naar de relatie BOEKING.
  - C) Er hoeft niets te veranderen aan het model, omdat het al wordt afgedwongen.
  - D) Het attribuut klantnummer in de relatie BOEKING een verplicht attribuut maken.
- 
- A) Incorrect. Het toevoegen van dit attribuut leidt niet alleen tot dubbel vastleggen van een klant, maar zorgt er ook niet voor dat altijd de klant bekend is. Het is immers niet verplicht om er waarden voor op te nemen. De klant is wel bekend, omdat dit al wordt afgedwongen door het voorkomen van het attribuut klantnummer in de primaire sleutel van de relatie BOEKING.
  - B) Incorrect. Dit dwingt het niet af. Het wordt al afgedwongen doordat het attribuut klantnummer in de relatie BOEKING onderdeel uitmaakt van de primary key (primaire sleutel). Voor de primary key moet voor een boeking altijd een waarde voorkomen. Het heeft wel een ongewenst effect: een klant kan hierdoor maar één boeking doen, terwijl het doel van de verhuurder zal zijn dat de klant meer boekingen kan doen. Een boeking is slechts voor één week, terwijl veel klanten voor meer dan één week boeken.
  - C) Correct. Dit wordt inderdaad al afgedwongen, doordat het attribuut klantnummer onderdeel is van de primary key (primaire sleutel). Er moet dus altijd een waarde voor worden opgenomen bij het vastleggen van een boeking. (Literatuur: A, Hoofdstuk 3.1 t/m 3.4 en 5.3)
  - D) Incorrect. Dit lijkt handig, maar het dwingt niets extra's af. Het verplicht opnemen van een waarde wordt al afgedwongen doordat het attribuut klantnummer deel is van de primary key (primaire sleutel).

**16 / 30**

Een bedrijf maakt reisgidsen en houdt alle gidsen bij in een relationele database gebaseerd op het volgende schema:

```
REISGIDS (gidsnr, titel, ingangsdatum, einddatum)
REISPRODUCT (productnr, soortproduct, <gidsnr>, omschrijving, bestemming,
aantal dagen, soortverblijf, soortvervoer, soortvermaak)
```

Het bedrijf wil een reisgids aan de tabel REISGIDS toevoegen. Het gidsnr is 3067 en nog nooit eerder gebruikt. De titel is 'Exotische Bestemmingen'. De ingangsdatum is 01-01-2023, de einddatum is 01-07-2033. Er zijn nog geen reisproducten voor deze reisgids aanwezig in de relationele database.

Vervolgens worden er tien reisproducten aan gids 3067 toegevoegd.

Hoe kan het bedrijf bij het toevoegen van deze gegevens aan de relationele database met zo min mogelijk bewerkingen voldoen aan dit relationele model?

- A) Bij het opnemen van de reisproducten voor de reisgids er voor zorgen dat de waarde 3067 voor gidsnr al voorkomt in de tabel REISGIDS door eerst de betreffende gids vast te leggen
  - B) Eerst voor ieder van de reisproducten een rij in de tabel REISPRODUCT toevoegen en daarna pas voor de reisgids een rij in de tabel REISGIDS
  - C) Voor het attribuut gidsnr in de tabel REISPRODUCT NULL opnemen en pas na toevoegen van de reisgids in iedere rij een waarde voor dit attribuut opnemen
- 
- A) Correct. Op deze manier wordt de referentiële integriteit niet overtreden en is het aantal bewerkingen minimaal. (Literatuur: A, Hoofdstuk 3.2 en 3.3)
  - B) Incorrect. Op deze manier wordt de referentiële integriteit overtreden.
  - C) Incorrect. Weliswaar wordt zo de referentiële integriteit niet overtreden, maar er worden veel meer bewerkingen gedaan dan bij antwoord A.



17 / 30

Een vereniging van muziekkorpsen houdt de ledenadministratie bij in een relationele database, die als volgt is gedefinieerd:

VERENIGING (verenigingnaam, contributie, betaalperiode)

LID (lidnaam, adres, postcode, plaats, ingangsdatum, betaalperiode)

Van de relatie VERENIGING zal één tupel worden ingevoerd met als key (sleutel) 'St. Caecilia'. Betaalperiode in een tupel van relatie LID kan NULL zijn, want dan geldt voor dat lid de betaalperiode uit de relatie VERENIGING.

Waarom moet de vereniging zich **volgens het gegeven relationele model** houden bij het invullen van waarden voor de attributen betaalperiode?

- A) Als bij LID betaalperiode een waarde heeft, dan mag bij VERENIGING betaalperiode NULL zijn.
  - B) Als bij LID betaalperiode NULL is, dan moet bij VERENIGING een waarde ingevuld zijn.
  - C) Als bij VERENIGING betaalperiode NULL is, dan moet bij LID een waarde ingevuld zijn.
  - D) Bij VERENIGING en LID maakt het niet uit of betaalperiode ingevuld is of niet.
- 
- A) Incorrect. Of bij VERENIGING een NULL mag voorkomen, heeft niets te maken met wat bij LID voorkomt.
  - B) Incorrect. Het relationele model dwingt niets af over het opnemen of weglaten van waarden.
  - C) Incorrect. Of bij LID iets moet ingevuld zijn, heeft niets te maken met wat bij VERENIGING is ingevuld. Bovendien hoeft er sowieso niets ingevuld te zijn.
  - D) Correct. Het relationele model dwingt niets af over het opnemen of weglaten van waarden. (Literatuur: A, Hoofdstuk 3.3, 3.4 en 5.3)

18 / 30

Welke eigenschap is **niet** van toepassing op de werking van een relationeel databasemanagementsysteem (DBMS)?

- A) Bij het leggen van relaties tussen records zijn de verwijzingen zichtbaar voor de gebruiker.
  - B) Bij het opslaan van gegevens kan hooguit één record tegelijk toegevoegd worden.
  - C) Conceptuele beschrijvingen kunnen views bevatten.
  - D) Externe beschrijvingen kunnen tabellen bevatten.
- 
- A) Incorrect. De relaties worden gelegd met behulp van gegevens die zichtbaar zijn voor de gebruiker.
  - B) Correct. Er kunnen hele verzamelingen tegelijk toegevoegd worden. (Literatuur: A, Hoofdstuk 2.2)
  - C) Incorrect. Dit is wel van toepassing.
  - D) Incorrect. Dit is wel van toepassing.

19 / 30

De organisatoren van een wandeltocht hebben een relationele database met een tabel gebaseerd op het onderstaande deel van een relationele schema:

```
WANDELTOCHT (datum, deelnemer, kilometers)
```

Welke SELECT-opdracht geeft antwoord op de vraag welke deelnemers in één wandeltocht meer dan 50 km gelopen hebben?

A) 

```
SELECT deelnemer
FROM wandeltocht
GROUP BY deelnemer
HAVING SUM(kilometers) > 50
```

B) 

```
SELECT deelnemer
FROM wandeltocht
WHERE kilometers > 50
```

C) 

```
SELECT deelnemer
FROM wandeltocht
WHERE SUM(kilometers) > 50
```

- A) Incorrect. Hier wordt bepaald welke deelnemers over alle door hen gelopen wandeltochten samen meer dan 50 km gelopen hebben. Dat was de vraag niet.
- B) Correct. Een rij gaat over een wandeltocht van één deelnemer. Als de deelnemer in een wandeltocht meer dan 50 km gelopen heeft, is het voldoende dat voor kilometers meer dan 50 is ingevuld. De voorwaarde gaat over één wandeltocht. (Literatuur: A, Hoofdstuk 7.2)
- C) Incorrect. Een SUM-functie mag niet zo in een voorwaarde gebruikt worden. De SUM-functie mag alleen in de SELECT, de uitkomst van een subquery en HAVING gebruikt worden.

20 / 30

Een zangvereniging houdt de deelname van zangers aan een voorstelling in een database bij. Hieronder staan twee gebruikte tabellen:

ZANGER

zangernr	naam	soort
1	Koos	bariton
2	Karel	bas
3	Jan	sopraan
4	Judith	sopraan
5	Kelly	bas
6	Toos	sopraan

VOORSTELLING

voorstellingsnr	zangernr
4	2
4	3
5	1
5	2
5	4
6	1
6	5
7	1
7	3
7	4

Welke SELECT-opdracht geeft als uitkomst de zanger die nergens aan heeft meegedaan?

- A) `SELECT *`  
`FROM zanger`  
`WHERE zangernr IN`  
`(SELECT zangernr`  
`FROM voorstelling)`
- B) `SELECT *`  
`FROM zanger`  
`WHERE zangernr IN`  
`(SELECT zangernr`  
`FROM voorstelling`  
`AND zanger.zangernr < > voorstelling.zangernr)`
- C) `SELECT *`  
`FROM zanger`  
`WHERE zangernr NOT IN`  
`(SELECT zangernr`  
`FROM voorstelling)`
- D) `SELECT *`  
`FROM zanger`  
`WHERE zangernr NOT IN`  
`(SELECT zangernr`  
`FROM voorstelling, zanger`  
`WHERE voorstelling.zangernr = zanger.zangernr)`

De feedback staat op de volgende pagina.

- A) Incorrect. De voorwaarde met IN zorgt ervoor dat de zangers worden geselecteerd die hebben meegedaan aan een show.
- B) Incorrect. De voorwaarde `zanger.zangernr < > show.zangernr` heeft als effect dat de output van de subquery niet de gezochte zanger bevat. De voorwaarde IN eist dat juist wel. Het gevolg is dat er nooit output kan komen, ook geen output die overeenkomt met wat in de vraag gesteld wordt. Tevens komt de kolom `zangernr` zowel in de tabel `show` als de tabel `zanger` voor. Bij SELECT in de subquery had daarom `show.zangernr` of `zanger.zangernr` moeten staan.
- C) Correct. In de uitkomst van de subquery staan de zangernummers van alle zangers die aan een show hebben meegedaan. De voorwaarde met NOT IN levert een TRUE op als de onderzochte zanger er niet in voorkomt. Voor de zanger die nergens aan heeft meegedaan, zal dat het geval zijn. (Literatuur: A, Hoofdstuk 7.2 en 9)
- D) Incorrect. Er zit een syntaxfout in de subquery. De kolom `zangernr` komt in zowel de tabel `show` als de tabel `zanger` voor. Bij SELECT in de subquery had daarom `show.zangernr` of `zanger.zangernr` moeten staan. Daarnaast komt in de subquery een overbodige `join` voor.

21 / 30

Een boerenbedrijf heeft een tabel METING voor het volgen van informatie over de koeien. De rijen 100 tot en met 149 gaan over koe 4.

METING

metingnr	soort	eenheid	waarde
99	urine	liter	15
100	temperatuur	°C	38
101	temperatuur	°C	39
102	melk	liter	24
103	melk	liter	22
104	melk	liter	26

enzovoort

Welke SELECT-opdracht geeft als uitkomst hoeveel melk koe 4 geproduceerd heeft?

- A)** `SELECT SUM(waarde)`  
`FROM meting`  
`WHERE eenheid = 'liter'`  
`AND metingnr > 99`  
`AND metingnr < 150`
- B)** `SELECT SUM(waarde)`  
`FROM meting`  
`WHERE soort = 'melk'`  
`AND metingnr > 99`  
`AND metingnr < 150`
- C)** `SELECT soort, SUM(waarde)`  
`FROM meting`  
`WHERE soort = 'melk'`  
`AND metingnr > 99`  
`AND metingnr > 150`
- D)** `SELECT SUM(waarde)`  
`FROM meting`  
`WHERE soort = 'melk'`  
`AND metingnr > 99`  
`OR metingnr < 150`
- A)** Incorrect. De metingen zijn beperkt tot die in liters gemeten worden. Dit hoeft niet alleen over melk te gaan. Urine wordt ook in liters gemeten.
- B)** Correct. Niet alle metingen gaan over melkproductie. De metingen zijn hier beperkt door de voorwaarde over soort. Koe 4 heeft alleen metingen vanaf rij 100 en tot en met 149. De geproduceerde hoeveelheid is een totaal, vandaar de SUM-functie. (Literatuur: A, Hoofdstuk 7.2 en 8.1)
- C)** Incorrect. Door de voorwaarde > 150 kunnen alleen metingen boven de 150 worden meegenomen. Die gaan helemaal niet over koe 4.
- D)** Incorrect. Door de OR kunnen ook metingen buiten de range 99 - 150 meegenomen worden.

**22 / 30**

Een bedrijf houdt zijn salarisadministratie bij in een database gebaseerd op het volgende relationele schema:

WERKNEMER (werknemernr, naam, achternaam, geboortedatum, functie, salaris)  
AFDELING (<wknemernr>, afdelingnaam, manager\_naam)  
UITBETALING (<wknemernr>, maand, jaar, bedrag, status, betaald j/n)  
PENSIOEN (<wknemernr>, jaar, verzekerd j/n, extra\_pensioen j/n)

In het kader van een CAO-aanpassing krijgen alle werknemers met een maandsalaris lager dan € 3000,- een verhoging van € 50,- per maand.

Hoe luidt de UPDATE-opdracht hiervoor?

- A) UPDATE Salaris  
FROM Werknemer  
SET Salaris = Salaris + 50  
WHERE Salaris < 3000
- B) UPDATE Werknemer  
SET Salaris = Salaris + 50  
WHERE Salaris < 3000
- C) UPDATE Werknemer  
SET Salaris TO Salaris + 50  
WHERE Salaris < 3000

- A) Incorrect. De clause FROM komt niet voor bij UPDATE.
- B) Correct. De opdracht UPDATE brengt wijzigingen aan in de rijen waar voor de WHERE-clausule een TRUE geldt. De WHERE-clausule is TRUE voor rijen waar het salaris lager is dan 3000. (Literatuur: A, Hoofdstuk 11.4)
- C) Incorrect. De clause SET ..... TO..... bestaat niet.

23 / 30

Een vereniging voor muziekkorpsen houdt een database bij van de scores van de verschillende korpsen op hun concoursen. Ze houden ook bij of het korps een amateurkorps (A) of een professioneel korps (P) is.

KORPS

Naam	Klasse	Score
St. Jozef	A	332
St. David	A	328
Eendracht	P	330
Adevendo	P	326
KMK	A	334
JWFK	A	332
Cecilia	P	328

Met welke opdracht kunnen alle korpsen met een score van 330 verwijderd worden?

- A) `DELETE FROM Korps  
HAVING Score = 330`
  - B) `DELETE FROM Korps  
WHERE Score = 330`
  - C) `DELETE Naam, Klasse, Score  
FROM Korps  
WHERE Score = 330`
- A) Incorrect. HAVING hoort niet in deze opdracht.  
B) Correct. DELETE zorgt voor het verwijderen van rijen en de WHERE-clausule beperkt de verwijderde rijen tot alleen rijen waarin de waarde 330 staat. (Literatuur: A, Hoofdstuk 11.3)  
C) Incorrect. De afzonderlijke velden worden niet genoemd. Daarnaast moet de opdracht beginnen met DELETE FROM.

24 / 30

In een relationele database is de tabel TAALBEHEERSING opgenomen:

TAALBEHEERSING

Naam	Taal	Niveau
Anneke	Engels	Zeer goed
Anneke	Frans	Matig
Karel	Duits	Goed
Karel	Engels	Zeer goed
Willem	Duits	Goed
Willem	Engels	Matig

Het volgende SQL-statement wordt uitgevoerd:

```
SELECT Naam, Taal, Niveau
FROM taalbeheersing
WHERE Taal IN
(SELECT Taal
FROM taalbeheersing
WHERE Naam = 'Anneke')
```

Uit hoeveel rijen bestaat het resultaat van dit SQL-statement?

- A) 1
- B) 2
- C) 4
- D) 5

- A) Incorrect. De subquery heeft als resultaat alle talen die Anneke spreekt (Engels en Frans). De buitenste query levert daardoor alle rijen op waar één van die talen in voorkomt. Dat zijn alle rijen behalve het Duits van Karel en van Willem.
- B) Incorrect. De subquery heeft als resultaat alle talen die Anneke spreekt (Engels en Frans). De buitenste query levert daardoor alle rijen op waar één van die talen in voorkomt. Dat zijn alle rijen behalve het Duits van Karel en van Willem.
- C) Correct. De subquery heeft als resultaat alle talen die Anneke spreekt (Engels en Frans). De buitenste query levert daardoor alle rijen op waar één van die talen in voorkomt. Dat zijn alle rijen behalve het Duits van Karel en van Willem. (Literatuur: A, Hoofdstuk 9.2)
- D) Incorrect. De subquery heeft als resultaat alle talen die Anneke spreekt (Engels en Frans). De buitenste query levert daardoor alle rijen op waar één van die talen in voorkomt. Dat zijn alle rijen behalve het Duits van Karel en van Willem.



**25 / 30**

Een bedrijf houdt zijn wagenpark bij in een relationele database. Bijgehouden wordt welke auto ze hebben, hoeveel wielen er bij de auto horen (maximaal acht bij een set van zomerbanden en winterbanden), en wanneer de auto is aangekocht.

WAGENPARK (auto, wielen, aankoop)

Hierbij is auto een tekstveld van maximaal tien letters, wielen is een geheel getal, aankoop is een datum. Auto is de primary key (primaire sleutel) en alleen aankoop is optioneel.

Met welke SQL-opdracht wordt bovengenoemde tabel gemaakt?

- A)**

```
CREATE TABLE wagenpark
(auto CHAR(10) NOT NULL,
wielen INTEGER NOT NULL,
aankoop DATE,
PRIMARY KEY(auto))
```
  - B)**

```
CREATE TABLE wagenpark
(auto CHAR(10) NOT NULL,
wielen NUMBER NOT NULL,
aankoop DATE,
PRIMARY KEY(auto))
```
  - C)**

```
CREATE TABLE wagenpark
(auto CHAR(10) PRIMARY KEY NOT NULL,
wielen INTEGER NOT NULL,
aankoop DATE)
```
  - D)**

```
CREATE TABLE wagenpark
(auto CHAR(10) PRIMARY KEY NOT NULL,
wielen NUMBER NOT NULL,
aankoop DATE)
```
- A)** Correct. De opdracht heeft de juiste syntaxis en houdt zich aan wat het scenario beschrijft. (Literatuur: A, Hoofdstuk 6.1)
- B)** Incorrect. Number is geen datatype. Dit moet INTEGER zijn.
- C)** Incorrect. De definitie van PRIMARY KEY moet achter de opsomming van de kolommen staan.
- D)** Incorrect. De definitie van PRIMARY KEY moet achter de opsomming van de kolommen staan. Bovendien is number geen geldig datatype.

26 / 30

Voor de jaarlijkse meterstanden heeft een energiemaatschappij een relationeel schema gemaakt:

```
VERBRUIKSADRES (postcode, huisnummer, straatnaam, plaats)
METING (<postcode, huisnummer>, jaar, kWh, gas, water, wie)
```

Hiervoor zijn de volgende twee CREATE TABLE-opdrachten gebruikt om tabellen te maken in de database:

```
CREATE TABLE VERBRUIKSADRES
(postcode CHAR(6),
huisnummer INTEGER NOT NULL,
straatnaam VARCHAR(100) NOT NULL,
plaats VARCHAR(60) NOT NULL,
PRIMARY KEY(postcode, huisnummer))
```

```
CREATE TABLE METING
(postcode CHAR(6),
huisnummer INTEGER,
jaar SMALLINT NOT NULL,
kWh SMALLINT NOT NULL,
gas SMALLINT NOT NULL,
water SMALLINT NOT NULL,
wie CHAR(1) NOT NULL,
PRIMARY KEY(postcode, huisnummer, jaar),
FOREIGN KEY(postcode, huisnummer) REFERENCES verbruiksadres
ON DELETE CASCADE
ON UPDATE CASCADE)
```

Welk gevolg heeft het uitvoeren van de opdracht `DROP TABLE verbruiksadres`?

- A) De opdracht leidt tot het verwijderen van alle rijen in de tabel VERBRUIKSADRES. De tabel zelf blijft bestaan.
  - B) De opdracht leidt zonder meer tot het verwijderen van de tabel VERBRUIKSADRES en de daarin opgenomen rijen.
  - C) De opdracht wordt nooit uitgevoerd omdat die in strijd is met de referentiële integriteit.
  - D) De tabel VERBRUIKSADRES wordt alleen verwijderd als er geen rijen in de tabel METING voorkomen.
- 
- A) Incorrect. Het succesvol uitvoeren van de opdracht DROP heeft altijd tot gevolg dat de gehele tabel verdwijnt.
  - B) Incorrect. Daar zal het meestal toe leiden, maar niet als de referentiële integriteit wordt overtreden.
  - C) Incorrect. Het uitvoeren is altijd in strijd met de referentiële integriteit zodra er rijen in de tabel meting voorkomen. De referentiële integriteit wordt door de database afgedwongen.
  - D) Correct. Als er geen rijen in de tabel meting voorkomen, leidt het uitvoeren van de opdracht niet tot overtreding van de referentiële integriteit. Komen er wel rijen voor, dan zou verwijderen wel in strijd zijn met de referentiële integriteit. De referentiële integriteit wordt door de database afgedwongen. (Literatuur: A, Hoofdstuk 6.1)

**27 / 30**

Een bedrijf voor feestartikelen heeft de volgende tabel in de database:

```
ORDERREGEL (<ordernr>, regelnr, artikelnaam, bestelaantal, leverdatum)
```

Het ordernr en bestelaantal kunnen grote getallen bevatten, maar het aantal regels per order is beperkt. Geen enkel artikel heeft een naam van meer dan 25 tekens. De leverdatum is niet altijd bekend. Het ordernummer moet uiteraard ook bekend zijn in de relatie 'order'.

Om voor deze relatie in het databasemanagementsysteem (DBMS) een tabel aan te maken wordt de volgende SQL-opdracht uitgevoerd:

```
CREATE TABLE orderregel
(ordernr INTEGER NOT NULL,
regelnr SMALLINT NOT NULL,
artikelnaam CHAR(25) NOT NULL,
bestelaantal INTEGER NOT NULL,
leverdatum DATE,
PRIMARY KEY(ordernr),
PRIMARY KEY(regelnr),
FOREIGN KEY(ordernnr) REFERENCES order
ON DELETE RESTRICT
ON UPDATE RESTRICT)
```

Waarom is dit SQL-statement **onjuist**?

- A) Bij leverdatum ontbreekt NULL.
  - B) Er mag slechts één keer PRIMARY KEY worden gebruikt.
  - C) Het gegevenstype van artikelnaam moet VARCHAR(25) zijn.
  - D) Na ON UPDATE mag RESTRICT niet worden gebruikt.
- 
- A) Incorrect. Als een kolom NULL-waarden kan bevatten, wordt dat aangegeven door het ontbreken van NOT NULL.
  - B) Correct. Zo zouden twee primary keys (primaire sleutels) van elk één kolom worden gedefinieerd, wat in een relationele database niet is toegestaan. Hier had moeten staan: PRIMARY KEY(Ordernr, Regelnr). (Literatuur: A, Hoofdstuk 6.1)
  - C) Incorrect. Afhankelijk van de gemiddelde lengte van artikelnamen is VARCHAR(25) wellicht wel aan te bevelen boven CHAR(25), maar daarmee is het nog niet fout.
  - D) Incorrect. Zowel voor ON UPDATE als voor ON DELETE mag vrijelijk worden gekozen tussen RESTRICTED, NULLIFIES en CASCADES.

**28 / 30**

Voor de jaarlijkse meterstanden heeft een energiemaatschappij een relationeel schema:

```
VERBRUIKSADRES (postcode, huisnummer, straatnaam, plaats)
METING (<postcode, huisnummer>, jaar, kWh, gas, water, wie)
```

Hiervoor zijn de volgende twee CREATE TABLE-opdrachten gebruikt om tabellen te maken in de database:

```
CREATE TABLE VERBRUIKSADRES
(postcode CHAR(6),
huisnummer INTEGER NOT NULL,
straatnaam VARCHAR(100) NOT NULL,
plaats VARCHAR(60) NOT NULL,
PRIMARY KEY(postcode, huisnummer))
```

```
CREATE TABLE METING
(postcode CHAR(6),
huisnummer INTEGER,
jaar SMALLINT NOT NULL,
kWh SMALLINT NOT NULL,
gas SMALLINT NOT NULL,
water SMALLINT NOT NULL,
wie CHAR(1) NOT NULL,
PRIMARY KEY(postcode, huisnummer, jaar),
FOREIGN KEY(postcode, huisnummer) REFERENCES verbruiksadres
ON DELETE CASCADE
ON UPDATE CASCADE)
```

Wat is het gevolg van het opnemen van de UPDATE-regel in de foreign key (vreemde sleutel)?

- A) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in de tabel METING naar NULL-waarden zorgen dat de bijbehorende rij in de tabel VERBRUIKSADRES wordt verwijderd.
  - B) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel METING worden ook doorgevoerd in de rij waarnaar verwezen wordt in de tabel VERBRUIKSADRES.
  - C) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel VERBRUIKSADRES hebben hooguit gevolgen voor één rij in de tabel METING.
  - D) Wijzigingen in de waarden voor de kolommen postcode en/of huisnummer in een rij van de tabel VERBRUIKSADRES worden ook doorgevoerd in de rijen van de tabel METING die naar de rij in de tabel VERBRUIKSADRES verwijzen.
- 
- A) Incorrect. Dit is niet mogelijk. De kolommen postcode en huisnummer maken deel uit van de primary key. Daar mogen geen NULL-waarden in voorkomen.
  - B) Incorrect. Dat is niet zo. Het is juist andersom. Wijzigingen van de rij in de tabel VERBRUIKSADRES worden doorgevoerd in de bijbehorende rijen van de tabel METING.
  - C) Incorrect. Meer dan één rij van de tabel METING kan verwijzen naar dezelfde rij in de tabel VERBRUIKSADRES.
  - D) Correct. Voor rijen van de child-tabel die naar een rij van de parent-tabel verwijzen, worden wijzigingen in de waarden van de kolommen van de primary key van de parent-tabel doorgevoerd in de waarden in de kolommen van de foreign key van de child-tabel. (Literatuur: A, Hoofdstuk 3.2 en 6.1)

29 / 30

Gegeven is de volgende reeks autorisatie-opdrachten:

```
GRANT DELETE ON rapport TO Peters, Jansen, Kuyten, Soons
REVOKE DELETE ON rapport FROM Jansen, Kuyten
GRANT ALL ON rapport TO Kuyten
GRANT INSERT ON rapport TO Peters
GRANT ALTER ON rapport TO Soons
```

Wie is **niet** geautoriseerd om rapporten te verwijderen nadat bovenstaande opdrachten in de weergegeven volgorde zijn uitgevoerd?

- A) Jansen
  - B) Kuyten
  - C) Peters
  - D) Soons
- 
- A) Correct. Na opdracht 1 en 2 mogen Peters en Soons rapporten verwijderen. Na opdracht 3 mag Kuyten dat ook. De opdrachten 4 en 5 wijzigen niets op het gebied van verwijderen. Jansen is dus niet geautoriseerd om rapporten te verwijderen. (Literatuur: A, Hoofdstuk 6.3)
  - B) Incorrect. Na opdracht 3 mag Kuyten ook rapporten verwijderen.
  - C) Incorrect. Na opdracht 1 en 2 mogen Peters en Soons rapporten verwijderen.
  - D) Incorrect. Na opdracht 1 en 2 mogen Peters en Soons rapporten verwijderen.

30 / 30

Gegeven is de volgende tabel in een relationele database:

```
LOGIN (loginnummer, gebruikersnaam, wachtwoord, toegangsniveau)
```

De eigenaar van de database voert de volgende opdracht uit:

```
REVOKE UPDATE ON LOGIN FROM user6
```

Wat verandert er voor gebruiker user6?

- A) User6 mag de structuur van de tabel LOGIN niet meer wijzigen.
  - B) User6 mag geen nieuwe rechten meer vastleggen in de tabel LOGIN.
  - C) User6 mag geen rechten uit de tabel LOGIN meer aanpassen.
  - D) User6 mag haar rechten voor de tabel LOGIN niet meer doorgeven.
- 
- A) Incorrect. UPDATE gaat niet over het aanpassen van de structuur, maar over het aanpassen van de inhoud van rijen.
  - B) Incorrect. Hiervoor zou in de opdracht INSERT moeten zijn opgenomen.
  - C) Correct. Door de opdracht kan user6 niet langer de inhoud van rijen in de tabel LOGIN aanpassen. Daarmee kan user6 dus ook niet langer de rechten die met de rijen worden vastgelegd aanpassen. (Literatuur: A, Hoofdstuk 6.3)
  - D) Incorrect. Het kan goed zijn dat user6 hierna nog wel andere rechten heeft dan het aanpassen van de inhoud van rijen. User6 kan die rechten nog steeds doorgeven als daar een clause WITH GRANT OPTION voor bestaat.

# Evaluatie

De juiste antwoorden op de vragen in dit voorbeeldexamen staan in onderstaande tabel.

Vraag	Antwoord	Vraag	Antwoord
1	C	16	A
2	B	17	D
3	B	18	B
4	C	19	B
5	A	20	C
6	A	21	B
7	D	22	B
8	D	23	B
9	C	24	C
10	C	25	A
11	B	26	D
12	C	27	B
13	C	28	D
14	B	29	A
15	C	30	C





Driving Professional Growth

**Contact EXIN**

[www.exin.com](http://www.exin.com)