

Mock-Examen: Information Modelling & Exchange

Deel 1: Conceptueel Modelleren (ERD / EER)

Vraag 1: In de tekst staat: *"Elk kunstobject heeft een unieke code en een titel."* Hoe modelleer je 'titel' in een ER-diagram?

- A) Als een Entity Type
- B) Als een Attribute**
- C) Als een Subtype
- D) Als een Relationship

Vraag 2: In het diagram zie je een driehoekje tussen ArtObject (boven) en de subtypes Painting en Sculpture (onder) met de tekst **{or, optional}**. Wat betekent dit specifiek voor de data?

- A) Elk kunstobject in de database moet verplicht óf een schilderij óf een beeldhouwwerk zijn.
- B) Een kunstobject kan een schilderij zijn, een beeldhouwwerk zijn, of een "algemeen" object blijven (zoals een foto of poster).**
- C) Een kunstobject kan in dit model zowel een schilderij als een beeldhouwwerk tegelijkertijd zijn.
- D) De tekst "optional" betekent dat het veld artObjectCode leeg mag blijven.

Vraag 3: Je hebt de entiteiten Artist en ArtObject. De relatie is "is gemaakt door". Eén kunstenaar maakt vele objecten, een object is door maximaal één kunstenaar gemaakt (maar soms is de kunstenaar onbekend). Wat is de cardinaliteit en optionaliteit aan de kant van de Artist?

- A) 1..1 (Mandatory One)
- B) 0..* (Optional Many)**
- C) 0..1 (Optional One)
- D) 1..* (Mandatory Many)

Vraag 4: Wat is een 'Weak Entity' volgens de slides van Chen (1976)?

- A) Een entiteit zonder attributen.
- B) Een entiteit die voor zijn bestaan afhankelijk is van een andere entiteit.**
- C) Een entiteit die alleen in GraphQL voorkomt.
- D) Een attribuut dat meerdere waarden kan bevatten.

Vraag 5: Hoe identificeer je een Subscription als dit een 'Weak Entity' is van Member?

- A) Alleen met een eigen subscriptionId.
 - B) Met de combinatie van memberNo en bijvoorbeeld startDate.**
 - C) Met een Foreign Key die niet in de Primary Key zit.
 - D) Zwakke entiteiten hebben geen identiteit.
-

Deel 2: OpenAPI (OAS 3.x)

Vraag 6: Je wilt in OpenAPI aangeven dat een Sculpture alle eigenschappen van ArtObject overneemt, plus het veld material. Welke syntax gebruik je?

- A) oneOf
- B) anyOf
- C) allOf** (remember: allOf: combinatie van basis + extra velden)
- D) not

Vraag 7: In je YAML-schema zie je:

type: object

properties:

dimensions:

\$ref: '#/components/schemas/Dimensions'

Wat is de functie van \$ref?

- A) Het definieert een nieuwe tabel in de database.
- B) Het hergebruikt een bestaande definitie om duplicatie te voorkomen.**
- C) Het maakt het veld verplicht.
- D) Het is een verwijzing naar een externe website.

Vraag 8: Hoe geef je in OpenAPI aan dat het veld creationDate wel door de server verstuurd wordt, maar niet door de client mag worden aangepast (zoals een ID)?

- A) writeOnly: true
- B) nullable: false
- C) readOnly: true**
- D) deprecated: true

Vraag 9: Je hebt een endpoint GET /artists/{id}. Waar bevindt de {id} parameter zich?

- A) in: query
- B) in: header
- C) in: path**
- D) in: body

Voorbeeld:

De URL & De Aanvraag

PUT https://api.museum.be/v1/paintings/AO-123?notify_curator=true

De Headers:

- Authorization: Bearer XYZ123
- Content-Type: application/json

De Body (JSON):

codeJSON

```
{  
  "title": "De Nachtwacht - Gerestaureerd",  
  "paintType": "Olieverf"  
}
```

De uitsplitsing naar OpenAPI syntax:

1. Path Parameter (in: path)

- **Wat:** AO-123
- **In de spec:** /paintings/{artCode}
- **Uitleg:** Het identificeert **welk** specifiek object je wilt hebben. Zonder dit weet de API niet welk schilderij hij moet aanpassen. Het is een vast onderdeel van het "adres".

2. Query Parameter (in: query)

- **Wat:** notify_curator=true (alles achter het ?)
- **In de spec:** ?notify_curator
- **Uitleg:** Dit is een **extra optie** of filter. Het schilderij wordt sowieso aangepast, maar deze parameter geeft de extra instructie: "stuur ook even een mailtje naar de beheerder". Het is meestal optioneel.

3. Header Parameter (in: header)

- **Wat:** Authorization: Bearer XYZ123
- **In de spec:** in: header, name: Authorization
- **Uitleg:** Dit is **metadata**. Het staat niet in de URL (dat zou onveilig zijn), maar het wordt "onder de motorkap" meegestuurd. Het geeft context over **wie** de aanvraag doet (beveiliging).

4. Request Body (requestBody)

- **Wat:** De JSON met de nieuwe titel en verfsoort.
- **In de spec:** requestBody -> content -> application/json -> schema

- **Uitleg:** Dit is de **inhoud** of de "lading" (payload). Hier staan de feitelijke gegevens die je wilt opslaan in de database. De body wordt bijna nooit gebruikt bij een GET, maar wel bij POST, PUT en PATCH.
-

Vraag 10: Welk keyword gebruik je in een OpenAPI schema om polymorfisme te ondersteunen, zodat een client direct ziet of een object een 'Painting' of 'Sculpture' is op basis van een veldnaam?

A) discriminator

B) enum

C) required

D) additionalProperties

Vraag 11: Je API moet een lijst van kunstwerken teruggeven. Hoe ziet de schema sectie van de 200-response eruit?

A) type: object met properties

B) type: array met items

C) type: string

D) oneOf: [Painting, Sculpture] zonder type-aanduiding.

Vraag 12: Welke HTTP statuscode is het meest passend bij een POST request die succesvol een nieuw object heeft aangemaakt?

A) 200 OK

B) 201 Created

C) 204 No Content

D) 409 Conflict

Deel 3: Normalisatie

Vraag 13: Een tabel heeft een samengestelde Primary Key {StudentID, VakCode}. Het attribuut VakNaam hangt alleen af van VakCode. In welke normaalvorm staat deze tabel **niet**?

A) 1NF

B) 2NF

C) 3NF

D) BCNF

Vraag 14: Wat is de definitie van een transitieve afhankelijkheid (overtreding van 3NF)?

A) Een attribuut hangt af van een deel van de PK.

B) Een niet-sleutel attribuut hangt af van een ander niet-sleutel attribuut.

C) Een tabel heeft dubbele rijen.

D) Een attribuut heeft meerdere waarden in één cel.

Vraag 15: In de case van het museum: In de koppeltabel ArtObjectExhibition staat {artId, exhId, insurancePremium}. De premie wordt bepaald door de tentoonstelling (elke tentoonstelling heeft één vaste premie voor alle objecten). Waar moet insurancePremium staan voor 2NF?

- A) In de tabel ArtObject
 - B) In de tabel Exhibition
 - C) In de koppeltabel laten staan
 - D) In een nieuwe tabel Insurance
-

Deel 4: GraphQL

Vraag 16: Wat is een groot voordeel van GraphQL ten opzichte van REST?

- A) Het gebruikt altijd HTTP GET.
- B) Het voorkomt 'over-fetching' (de client vraagt precies wat hij nodig heeft).
- C) Het heeft geen schema nodig.
- D) Het is sneller omdat het binair is.

Vraag 17: Waarom is onderstaande GraphQL query ongeldig?

codeGraphql

```
query {  
  allArtists {  
    name  
    artObjects  
  }  
}
```

- A) allArtists moet een ID hebben.
- B) artObjects is een complex type/lijst en moet een 'selection set' (accolades met velden) hebben.
- C) query mag geen hoofdletter zijn.
- D) GraphQL ondersteunt geen lijsten.

Vraag 18: Welke term gebruikt GraphQL voor operaties die data wijzigen op de server?

- A) Query
- B) Mutation
- C) Subscription
- D) Fragment

Vraag 19: Wat is een 'Fragment' in GraphQL?

A) Een deel van de database.

B) Een herbruikbare set velden die je in meerdere queries kunt gebruiken.

C) Een foutmelding van de server.

D) Een manier om data te filteren.

Vraag 20: Waar of Niet waar: In GraphQL heeft de server voor elk type object een apart endpoint (URL).

A) Waar

B) Niet waar (er is slechts één endpoint)

Deel 5: Mapping naar C# (OO)

Vraag 21: Je hebt een verplichte specialisatie {mandatory} van ArtObject. Hoe dwing je in C# af dat er nooit een 'kaal' ArtObject kan worden aangemaakt?

A) Maak de klasse static.

B) Maak de klasse abstract.

C) Gebruik een private constructor.

D) Dit kan niet in C#.

Vraag 22: Hoe implementeer je een 1-op-N relatie (bijv. één Artist heeft meerdere ArtObjects) in de C# klasse Artist?

A) Met een private int _artObjectId;

B) Met een collectie, bijv. public IList<ArtObject> ArtObjects { get; }

C) Door overerving: class Artist : ArtObject

D) Met een enum.

Vraag 23: Je wilt in C# de identiteit van een Member vastleggen zodat Equals() werkt op basis van het unieke memberNo. Wat moet je doen?

A) Niets, C# doet dit automatisch voor klassen.

B) De methoden Equals() en GetHashCode() overriden.

C) Het veld memberNo de naam Id geven.

D) De klasse readonly maken.

Vraag 24: In de case staat: *"Een kunstenaar heeft een sterfdatum indien bekend"*. Welk C# type gebruik je voor DateOfDeath?

A) DateTime

B) string

C) DateOnly? (Nullable DateOnly)

D) int

Vraag 25: Gegeven deze constructor in C#:

```
public Painting(int id, string title, string paintType) : base(id, title)
```

Wat doet : base(id, title)?

A) Het slaat de data op in een basis-database.

B) Het roept de constructor van de parent-klasse (ArtObject) aan.

C) Het maakt een kopie van het object.

D) Het is een syntaxfout.

Deel 6: REST & HTTP Basics

Vraag 26: Welk HTTP verb gebruik je voor een **gedeeltelijke** update van een object?

A) PUT

B) POST

C) PATCH

D) UPDATE

Vraag 27: Volgens de 'Semantic URL' regels in de slides, wat is de beste URL voor een lijst met boeken?

A) /get_all_books

B) /book

C) /books

D) /services/booklist

Vraag 28: Je ziet de URL /games/{id}/events. Wat stelt dit voor?

A) Alle events van alle games.

B) Een specifiek event met een ID.

C) Een collectie events die horen bij één specifieke game.

D) Een foutieve URL.

Vraag 29: Wat is het verschil tussen synchrone en asynchrone communicatie bij HTTP/API's?

A) Synchroon is altijd sneller.

B) Bij synchroon wacht de zender op een direct antwoord; bij asynchroon is er geen real-time gesprek nodig.

C) Synchroon gebruikt alleen JSON.

D) Er is geen verschil.

Vraag 30: Welke statuscode groep (1xx t/m 5xx) geeft aan dat er een fout is gemaakt door de **client** (bijv. een typefout in de URL)?

A) 2xx

B) 3xx

C) 4xx

D) 5xx