

- Het examen is volledig open boek. Je mag al het materiaal gebruiken dat voorzien is tijdens de lessen (bv. het boek ‘Principes van databases’ van prof. dr. Guy De Tré, slides, opgaves, (opgeloste) oefeningen. . .) of dat je zelf hebt gebruikt ter verwerking van de leerstof (bv. samenvattingen, notities. . .).
- Alle communicatie met personen (behalve de examenbegeleiders) is strikt verboden en zal bestraft worden met uitsluiting van het examen.
- Het examen dient opgelost te worden op witte bladeren. Gebruik voor elk deel van het examen (sectie 1-4) een apart blad. Vermeld zeker op elk oplossingsblad je studentenummer, naam en richting.
- Indien er vragen of onduidelijkheden zijn, schrijf deze dan nauwgezet op je oplossingsblad, zodat wij hier tijdens het quoteren rekening mee kunnen houden.

1 Open vragen

Beantwoord onderstaande vragen.

1. (Databanksystemen)
 - a) Wat wordt bedoeld met het geïntegreerd zijn van de data in een databank? Waarom is dit praktisch belangrijk?
 - b) Wat wordt bedoeld met het gestructureerd zijn van de data in een databankschema? Waarom is dit praktisch belangrijk?
2. (Beveiliging) Beschouw de meerniveaurelatie die wordt weergegeven in Figuur 1. Wat zou er gebeuren indien iemand met toegangsniveau P een waarde 30.000.000 invult voor het schilderij ‘Mona Lisa’? Wat zou er gebeuren als iemand op niveau G de ‘Mona Lisa’ zou verwijderen?
3. (NoSQL)
 - a) Waarom wordt het variëteitsprobleem als een grotere uitdaging gezien dan het volume probleem?
 - b) Geef drie verschilpunten tussen het manipuleren en bevragen van data bij een column store en bij een relationele databank. Geef bij elk verschilpunt ook aan bij welke databank de operatie het efficiëntst werkt en leg kort uit waarom.

Toegangs niveaus **TG** en **G**: SELECT * FROM Kunstwerk

Tabel Kunstwerk primaire sleutel = {KID, B ₁ } vreemde sleutel = {eigenaar, B ₅ }										
KID: char(3)	B ₁ : T _B	naam: varchar	B ₂ : T _B	periode: integer	B ₃ : T _B	waarde: real	B ₄ : T _B	eigenaar: varchar	B ₅ : T _B	B: T _B
K0001	P	Vissershuis	P	1882	P	16.000.000	G	Boijmans	V	G
K0003	P	Mona Lisa	P	1499	P	30.000.000	V	Louvre	P	V
K0003	G	Mona Lisa	G	1499	G	75.000.000	G	Louvre	G	G

Toegangs niveau **V**: SELECT * FROM Kunstwerk

Tabel Kunstwerk primaire sleutel = {KID, B ₁ } vreemde sleutel = {eigenaar, B ₅ }										
KID: char(3)	B ₁ : T _B	naam: varchar	B ₂ : T _B	periode: integer	B ₃ : T _B	waarde: real	B ₄ : T _B	eigenaar: varchar	B ₅ : T _B	B: T _B
K0001	P	Vissershuis	P	1882	P	NULL	V	Boijmans	V	V
K0003	P	Mona Lisa	P	1499	P	30.000.000	V	Louvre	P	V

Toegangs niveau **P**: SELECT * FROM Kunstwerk

Tabel Kunstwerk primaire sleutel = {KID, B ₁ } vreemde sleutel = {eigenaar, B ₅ }										
KID: char(3)	B ₁ : T _B	naam: varchar	B ₂ : T _B	periode: integer	B ₃ : T _B	waarde: real	B ₄ : T _B	eigenaar: varchar	B ₅ : T _B	B: T _B
K0001	P	Vissershuis	P	1882	P	NULL	P	NULL	P	P
K0003	P	Mona Lisa	P	1499	P	NULL	P	Louvre	P	P

TG: topgeheim > G: geheim > V: vertrouwelijk > P: publiek

Figuur 1: Meerniveaurelatie

2 Conceptueel ontwerp

De organisatoren van een muziekfestival vragen om een nieuwe databank te ontwikkelen voor de opslag van hun gegevens. Stel een volledig en correct conceptueel ontwerp op, uitgaande van onderstaande vereisten.

Allereerst moeten de gegevens met betrekking tot de programmatie opgeslagen kunnen worden. Het festival bestaat uit een aantal festivaldagen waarop er concerten van artiesten op podia doorgaan. Een festivaldag wordt uniek bepaald door een datum en heeft bijhorende tijdstippen waarop de festivalwei opent en sluit. Concerten worden ingepland op podia, die elk uniek geïdentificeerd kunnen worden door een naam. Om de veiligheid te kunnen garanderen wordt er, voor elk podium, een maximale capaciteit vastgelegd door de organisatie. Een artiest heeft een unieke naam, een land van herkomst, en optioneel een beschrijving en een (link naar een) foto. Daarnaast worden er ook een of meerdere muziekgenres toegekend aan een artiest. Een concert heeft een begintijdstip en een eindtijdstip, die liggen tussen het vastgelegde openingstijdstip en sluitingstijdstip van de festivaldag waarop het

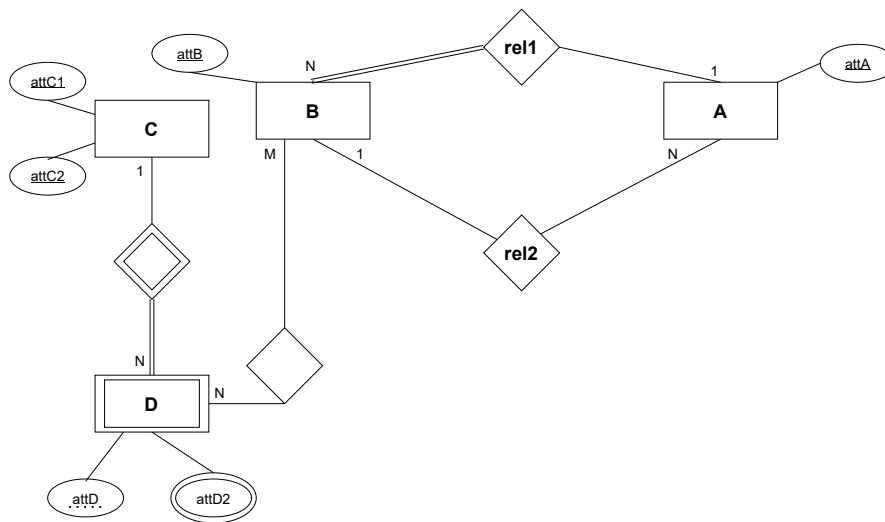
concert plaatsvindt. Belangrijk is dat concerten die doorgaan op eenzelfde podium niet mogen overlappen. Het is mogelijk dat een artiest meerdere concerten speelt op het festival, die vanzelfsprekend ook niet mogen overlappen.

Personen kunnen een of meerdere tickets voor het festival kopen. Er zijn verschillende soorten tickets: dagtickets, combitickets en campingtickets. Een dagticket laat toe om het festival gedurende 1 festivaldag te bezoeken. Er zijn, met andere woorden, evenveel soorten dagtickets als dat er festivaldagen zijn. Met een combiticket kan je alle dagen van het festival bezoeken. Een campingticket hoort daarnaast bij een bepaalde camping. Elke camping kan uniek geïdentificeerd worden door de combinatie van een naam en een nummer.

Om een ticket, met een unieke barcode, van een bepaald type te kunnen kopen dient een persoon zich eerst te registreren via de website van het festival. Daarbij wordt er om een uniek e-mailadres, een uniek telefoonnummer, een voornaam, een achternaam en (optioneel) een adres (straat, huisnummer, postcode, gemeente, land) gevraagd. Per aangekocht festivalticket (dagticket/combiticket) mag een persoon hoogstens 1 campingticket kopen.

3 Logisch ontwerp

Zet het (abstract) EER-diagram gegeven in Figuur 2 (dus niet je oplossing op de vorige oefening...) om naar een logisch databankschema. Lijst in dit logisch databankschema enkel de tabellen, attributen, primaire sleutels, vreemde sleutels, uniciteits- en NOT NULL-voorwaarden op. Gebruik de notatie en de methodiek die we hebben geïntroduceerd tijdens de oefeningenlessen.



Figuur 2: Abstract EER-diagram

4 Workshops

In Appendix 1 is de DDL-code gegeven voor de aanmaak van een databank waarin data in verband met de spelgeschiedenis op spelconsoles worden opgeslagen. Bekijk deze DDL-code nauwgezet en los daarna onderstaande vragen op.

1. In het huidige databankontwerp kan er niet worden opgeslagen of een gebruiker een bepaald spel gekocht heeft. Het bedrijf dat deze consoles en spelletjes maakt en verkoopt wil voortaan bijhouden welke spelletjes een gebruiker zelf heeft gekocht en zal andere gebruikers niet toelaten dit spel te spelen. Er kan dus, met andere woorden, geen spelgeschiedenis meer opgeslagen worden voor een gebruiker wanneer hij/zij een bepaald spel zelf niet gekocht heeft. Beschrijf hoe je het huidige databankontwerp zou aanpassen om deze nieuwe richtlijnen te reflecteren. Je hoeft geen code neer te schrijven, maar wees volledig en zorg dat je aangepaste databank nog steeds voldoet aan alle regels omtrent goed databankontwerp.
2. Een veelvoorkomende query die wordt uitgevoerd op deze databank zoekt naar personen op basis van de initialen van hun voor- en familienaam. Dit gebeurt telkens op een hoofdletterongevoelige manier. Neem de initialen *m.d.* als voorbeeld, waarmee ze alle gebruikers opvragen met een voornaam die start met de letter *m* en een familienaam die start met de letter *d*. Is er een manier om deze bevraging efficiënter te laten verlopen? Beschrijf concreet op welke manier je deze query kan optimaliseren of leg uit waarom deze query niet efficiënter gemaakt kan worden.
3. Beschouw de view met naam *view1*, gegeven in Appendix 1.
 - a) Leg, in jouw eigen woorden, uit wat het resultaat van deze view zal zijn. Het is niet de bedoeling om een gedetailleerde technische beschrijving te geven van deze view en de achterliggende code, maar wel om een semantische beschrijving te geven van welke data de resulterende tabel zal bevatten.
 - b) Vertrekkende van een initieel lege databank, teken de tabel die je verkrijgt na het uitvoeren van de volgende sequentie SQL-commando's. Je mag veronderstellen dat *CURRENT_DATE* overeenkomt met de datum van dit examen (9 september 2023).
 1. `INSERT INTO gebruiker VALUES ('madforch', 'maxime.deforche@ugent.be', 'Maxime', 'Deforche');`
 2. `INSERT INTO gebruiker VALUES ('tboeckli', 'toon.boeckling@ugent.be', 'Toon', 'Boeckling');`
 3. `INSERT INTO console VALUES (1, 'ps5');`
 4. `INSERT INTO console VALUES (2, 'ps5');`
 5. `INSERT INTO spel VALUES ('GTA V', 'Action');`

```

6. INSERT INTO spel VALUES ('Spider-Man', 'Adventure');
7. INSERT INTO spelgeschiedenis values ('madforch', 1, 'GTA V',
    '2023-09-07 20:00', '2023-09-07 22:00');
8. INSERT INTO spelgeschiedenis values ('tboeckli', 2, 'GTA V',
    '2023-09-07 23:00', '2023-09-08 01:00');
9. INSERT INTO spelgeschiedenis values ('madforch', 1, 'Spider-Man',
    '2023-09-08 12:00', '2023-09-08 14:00');
10. INSERT INTO spelgeschiedenis values ('madforch', 1, 'GTA V',
    '2023-09-08 14:00', '2023-09-08 17:00');
11. INSERT INTO spelgeschiedenis values ('madforch', 2, 'Spider-Man',
    '2023-09-08 17:00', '2023-09-08 19:00');
12. INSERT INTO spelgeschiedenis values ('tboeckli', 2, 'GTA V',
    '2023-09-08 19:00', '2023-09-08 21:00');
13. INSERT INTO spelgeschiedenis values ('tboeckli', 2, 'Spider-Man',
    '2023-09-08 23:00', '2023-09-09 02:00');
14. INSERT INTO spelgeschiedenis values ('madforch', 1, 'Spider-Man',
    '2023-09-09 08:00', '2023-09-09 11:00');
15. SELECT * FROM view1;

```

Appendix 1: DDL-code spelconsole databank

```
CREATE TABLE gebruiker (  
    gebruikersnaam VARCHAR PRIMARY KEY,  
    email VARCHAR UNIQUE NOT NULL,  
    voornaam VARCHAR NOT NULL,  
    familienaam VARCHAR NOT NULL,  
    telefoon VARCHAR  
);  
  
CREATE TABLE console (  
    serienummer INTEGER PRIMARY KEY,  
    versie VARCHAR NOT NULL  
);  
  
CREATE TABLE spel (  
    naam VARCHAR PRIMARY KEY,  
    genre VARCHAR NOT NULL  
);  
  
CREATE TABLE spelgeschiedenis (  
    gebruikersnaam VARCHAR NOT NULL REFERENCES gebruiker(gebruikersnaam),  
    serienummer INTEGER NOT NULL REFERENCES console(serienummer),  
    spelnaam VARCHAR NOT NULL REFERENCES spel(naam),  
    begin TIMESTAMP NOT NULL,  
    einde TIMESTAMP NOT NULL,  
    CHECK (einde > begin),  
    PRIMARY KEY (serienummer, begin)  
);  
  
CREATE VIEW view1 AS SELECT s1.gebruikersnaam, spelnaam FROM (  
    SELECT gebruikersnaam, MAX(a) as b FROM (  
        SELECT gebruikersnaam, SUM(einde - begin) a FROM spelgeschiedenis  
        WHERE begin::DATE = (CURRENT_DATE - INTERVAL '1 day')::DATE  
        GROUP BY gebruikersnaam, spelnaam, serienummer  
    ) s GROUP BY gebruikersnaam  
    ) s1 INNER JOIN (  
    SELECT gebruikersnaam, spelnaam, SUM(einde - begin) a  
    FROM spelgeschiedenis  
    WHERE begin::DATE = (CURRENT_DATE - INTERVAL '1 day')::DATE  
    GROUP BY gebruikersnaam, spelnaam, serienummer  
    ) s2 ON a = b AND s1.gebruikersnaam = s2.gebruikersnaam;
```