



1. (1 punt)

Geef een vertaling van het bovenstaande E-R model naar een relationeel model, d.w.z. geef aan hoe deze informatie met behulp van relaties (tabellen) kan worden opgeslagen. Vermeld per relatie de attributen en sleutels.

2. (3 punten)

Geef een ER-model (ER-diagram) voor de data in het informatiesysteem. Geef aan welke voorwaarden/bepalingen niet in het model zichtbaar zijn en waarom niet.

De TU/e wil met behulp van een database registreren welke medewerkers wanneer op dienstreis zijn. Van elke medewerker registreert de faculteit naam, personeelsnummer en adres (straat, nummer, postcode, gemeente). We onderscheiden eendaagse en meerdaagse reizen. Elke reis heeft een aanvangsdatum en bestemming. Bij een eendaagse reis registreren we tijd van vertrek en terugkeer. Een meerdaagse reis heeft een terugkeerdatum, maar geen tijdstippen. De vergoeding van een eendaagse reis kan een ontbijt, lunch of dinervergoeding bevatten, al naargelang vertrek- en terugkeertijdstip. Een dagvergoeding en logiesvergoeding zijn er alleen voor meerdaagse reizen. Er zijn bij beide soorten dienstreizen ook reiskosten, die kunnen bestaan uit kosten voor auto, bus, trein, vliegtuig of boot. Er wordt geen vergoeding gegeven voor reiskosten als bestemming de eigen woonplaats is of Eindhoven.

Bestudeer de beschrijving van de bibliotheekdatabase die in de queries-deel van de toets gebruikt wordt.

De bibliotheekdatabase

Het onderstaande relationele schema stelt een (deel van een) sterk vereenvoudigde universiteitsbibliotheekdatabase voor. Het wordt gebruikt in alle onderstaande vragen.

In de query opgaven wordt een (deeltje van een) universiteitsbibliotheekdatabase gebruikt met de volgende tabellen en attributen:

boek : { ISBN, titel, uitgever, jaar }
auteur : { ISBN, voorletters, naam }
exemplaar : { barcode, ISBN, faculteit, exjaar, aanwezig }
reservering : { naam, faculteit, ISBN, datum, geannuleerd }
uitlening : { naam, faculteit, barcode, van, tot }

Korte beschrijving:

Elk boek heeft een uniek ISBN nummer. Het heeft een titel, uitgever, jaar van uitgifte en een aantal auteurs waarvan we de voorletters (samen als 1 string) en de achternaam bijhouden. Verschillende edities van eenzelfde boek hebben een verschillend ISBN nummer en zijn dus voor ons verschillende boeken. De database bevat alle auteurs van in de database aanwezige boeken, en geen auteurs die niet (met ISBN nummer) overeenkomen met een boek.

De universiteit heeft van elk boek 1 of meer exemplaren. Elk exemplaar is in een bepaald jaar (exjaar) aangekocht, heeft een sticker met een unieke barcode en is toegewezen aan een faculteit. (Boeken kunnen worden uitgeleend aan mensen van andere faculteiten.) Een exemplaar kan afwezig zijn omdat het nog niet geleverd is, wordt hersteld, of omdat het is uitgeleend.

Personen worden geïdentificeerd door hun achternaam en faculteit. Ze kunnen een boek reserveren op een bepaalde datum. (Het type van “datum” is zodanig dat data die niet NULL zijn met elkaar vergeleken kunnen worden, bijvoorbeeld met $<$ en \leq .) Personen kunnen een reservering op elk willekeurig moment weer annuleren. De datum blijft dan die van de reservering, dus de datum waarop de reservering is geannuleerd wordt niet bijgehouden. (Het “geannuleerd” attribuut heeft altijd een waarde “ja” of “nee”.)

Ze kunnen een exemplaar van een boek lenen op een bepaalde “van” datum. Ze zijn dan “lener”. Een eventuele reservering van dat boek (op hun naam) wordt dan automatisch geannuleerd (de record van de reservering blijft in de database staan maar “geannuleerd” wordt “ja”). De “tot” datum blijft NULL tot het boek is teruggebracht. De “tot” datum is minstens 1 dag later dan de “van” datum. Wanneer een boek is teruggebracht wordt de “uitlening” in de database bewaard en wanneer een reservering wordt geannuleerd (bijvoorbeeld omdat het boek wordt geleend maar het kan ook om andere redenen zijn) wordt die reservering ook bewaard.

De bibliotheek bestaat al een hele tijd. Bijgevolg heeft elke faculteit wel wat (exemplaren van) boeken, heeft er uit elke faculteit wel eens iemand een boek gereserveerd en iemand een exemplaar geleend, en is er van elke faculteit wel eens een boek gereserveerd en wel eens een (exemplaar van) een boek geleend.

Let zeker op de volgende “valstrikken”:

- De attributen “naam” en “faculteit” worden in deze database in twee verschillende betekenissen gebruikt. Zorg dat je die niet per ongeluk door elkaar gebruikt. Let ook op voor de verschillen tussen “boeken” en “exemplaren”.

- In een vraag schrijven we misschien wel eens dat een “boek” wordt geleend, maar we bedoelen dan dat een “exemplaar” (eender welk!) van dat “boek” wordt geleend. Twee uitleningen van eenzelfde boek kunnen uitleningen van verschillende exemplaren zijn. Zorg dat je altijd goed in de gaten hebt of er een “boek” als generiek object wordt bedoeld of een “exemplaar” als concreet item dat kan worden geleend.

3. (1 punt) Vertaal naar de relationele algebra, de tupel calculus en SQL:

“Geef de naam en voorletters van elke auteur wiens boeken bij ten minste twee verschillende faculteiten te vinden zijn.” (Het kan maar het hoeft niet dat dat de exemplaren van eenzelfde boek van die auteur zijn.)

4. (1 punt) Vertaal naar de relationele algebra of de tupel calculus of SQL (naar keuze, maar slechts 1 van de drie):

“Geef alle uitgevers wiens boeken bij elke faculteitsbibliotheek aanwezig zijn.”

5. (1 punt) Vertaal naar de relationele algebra of de tupel calculus of SQL (naar keuze, maar slechts 1 van de drie):

“Geef de naam en faculteit van elke lener die nog nooit een (exemplaar van een) boek bij zijn eigen faculteit heeft geleend.”

6. (1 punt) Vertaal naar het Nederlands:

```

SELECT    u.naam, u.faculteit
FROM      uitlening AS u
WHERE     NOT EXISTS
( SELECT  *
  FROM    uitlening AS uu
  WHERE   uu.faculteit NOT IN
  (SELECT  e.faculteit
   FROM    uitlening uuu, exemplaar e
   WHERE   uuu.barcode = e.barcode AND
           uuu.naam =u.naam AND uuu.faculteit = u.faculteit
  ))

```