**DEEL A [25] E-CHOPPERS**  
In het straatbeeld heden ten dage valt het op dat er erg veel elektrische scootertjes rondrijden. Veel studenten rijden af en aan van het centrum naar de gebouwen van de Hanzehogeschool.   
Een clubje studenten met een commerciële inslag ziet een gat in de markt en heeft bedacht dat het ook wel handig zou zijn voor bedrijven als ze elektrische vervoersmiddelen ter beschikking zouden stellen voor hun personeelsleden wanneer deze op verplaatsing moeten.   
Voor docenten van de Hanzehogeschool zou het heel handig zijn als ze zich van De Deimten naar het U-gebouw moeten verplaatsen en het volgende lesuur weer in omgekeerde richting of naar Hanze Plaza.  
Het bedrijf E-CHOP was snel opgericht en is een groot succes. Er worden vooral leasecontracten opgesteld voor E-Choppers met name omdat die veel bredere banden hebben en daardoor veel beter op de weg liggen en voorkomen dat personeelsleden met een E-Chopper onderuit gaan.

Er is een speciaal verdienmodel bedacht. De leaseprijs is extreem laag gehouden om het aantrekkelijk te maken voor bedrijven om in te stappen. Het addertje onder het gras is dat alle reparatiekosten en onderhoudsbeurten voor rekening komen van de leasende partij. Hieronder staat een afrekening van de reparatiekosten voor een aantal E-choppers die geleaset zijn door de Hanzehogeschool.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Kostenoverzicht reparaties maand September 2024* | | | | | |
| **Klantnummer:** 12346  **Naam:** Hanzehogeschool **Adres:** Zernikeplein 7 **Postcode:** 9747 AS **Woonplaats:** Groningen **Contactpersoon:** Hanz Hanze | | | | | |
| **Kenteken** | **Merk** | **Datum** | **Reparatienr** | **Omschrijving** | **Kosten** |
| DPZ22P | Spyder E-Chopper 2.3 | 05-09-2024 | 01 | E-Chopper ophalen | 25,00 |
|  |  | 06-09-2024 | 24 | Band vervangen | 70,00 |
|  |  |  | 16 | Accu vervangen | 60,00 |
|  |  |  | 02 | E-Chopper brengen | 25,00 |
| FBR24S | Ecruiser Cafecruiser | 16-09-2024 | 01 | E-Chopper ophalen | 25,00 |
|  |  | 17-09-2024 | 38 | Remblokken vervangen | 55,00 |
|  |  |  | 02 | E-Chopper brengen | 25,00 |

Gevraagd: Teken een strokendiagram voor de beschreven situatie.   
De volgende gegevens moeten in het strokendiagram duidelijk te zien zijn:

* optionele, niet verplichte kolommen
* berekende attributen
* tabelnamen
* primaire sleutels
* relaties

Tip: Splits eerst de herhalende groep af en elimineer vervolgens de overige ongewenste redundantie.

**DEEL B [65]**Hier staat (een gedeelte) van de bekende OpenSchool database.  
Over deze database worden 12 vragen gesteld. De vragen hebben betrekking op de hoofdstukken 6, t/m 12.

Hieronder is het strokendiagram te zien wat hoort bij deze database.  
De afbeelding is integraal overgenomen uit het boek.

  
De inhoud van de tabellen kan op de volgende pagina’s bekeken worden.  
Een uitgebreid populatiediagram is te vinden achter in het boek.

D**e vragen:**

Binnen kaders staat de output vermeld, die door het uitvoeren van het statement zichtbaar moet zijn. Het SQL-statement moet in hetzelfde kader genoteerd worden.

1. Gevraagd:  
   Een overzicht van de gegevens van de tabel Cursus (naam en credits). Voorwaarde is dat de examinator niet ‘DAT’ mag zijn.  
   Sorteer het overzicht op naam. **[4]**

|  |
| --- |
| SELECT naam, credits  FROM cursus  WHERE examinator <> 'DAT'  ORDER BY naam; |

1. Gevraagd:

Een overzicht van alle tentamens die met een voldoende zijn afgesloten.   
In het overzicht moeten de volgende gegevens zichtbaar zijn, student, cursus, datum en poging.   
In de vierde kolom moet de tekst “Poging 1” of “Poging 2 te zien zijn. **[5]**

|  |
| --- |
| SELECT student, cursus, datum,  IIF(volgnr = 1, 'Poging 1', 'Poging 2') AS poging  FROM tentamen |

1. Gevraagd:

Een SQL-statement dat een overzicht oplevert van de namen van alle docenten die een vervanger hebben en de naam van de vakgroep waar ze deel van uitmaken. Ook de namen van hun vervangers moeten in het overzicht worden opgenomen. **[5]**

|  |
| --- |
| SELECT d.naam docent, vd.naam vakgroep\_docent, v.naam vervanger  FROM docent d  JOIN vakgroep vd ON d.vakgroep = vd.code  JOIN docent v ON v.vervanger = d.acr |

1. Gevraagd:  
   Ontwerp een SQL-statement dat een overzicht geeft van het aantal inschrijvingen per cursus. Zichtbaar moeten zijn code, naam van de cursus en het aantal inschrijvingen. [**5]**

|  |
| --- |
| SELECT c.code, c.naam, count(\*) as aantal\_inschrijvingen  FROM cursus c  JOIN inschrijving i on c.code = i.cursus  GROUP BY c.code, c.naam |

1. Gevraagd:

Een overzicht van de tentamens (cursus en gemiddeld cijfer) waarvan het gemiddelde cijfer minstens een 7 is. [**5]**

|  |
| --- |
| SELECT cursus, AVG(cijfer) AS gemiddeld\_cijfer  FROM tentamen  GROUP BY cursus  HAVING AVG(cijfer) >6 |

1. Gevraagd: een overzicht van de namen van de cursussen die DB of SW als voorkennis vereisen. Er moet een sortering plaatsvinden op de cursusnaam, in alfabetische volgorde. **[5]**

|  |
| --- |
| SELECT naam FROM cursus WHERE code IN  (SELECT cursus  FROM voorkenniseis  WHERE voorkennis IN ('DB', 'SW')) ORDER BY naam |

1. Gevraagd:

Een overzicht van de begeleiders (naam is voldoende) die geen begeleider zijn van de cursus “Informatiemodelleren”.   
Maak bij het opstellen van de query gebruik van (een) SUBSELECT (‘s). **[6]**

|  |
| --- |
| SELECT naam  FROM docent  WHERE acr IN  (SELECT docent  FROM begeleider  WHERE cursus NOT IN  (SELECT code  FROM cursus  WHERE naam ='Discrete Wiskunde'  OR naam = 'Informatiemodelleren')) |

1. Op de school worden een aantal wijzigingen doorgevoerd binnen de tabel vakgroep. De namen kloppen niet meer en moeten vervangen worden door actuelere namen. IS moet worden gewijzigd in BI met naam ‘Business Intelligence’ en ST door SE met naam ‘Software Engineering’.Als refererende actieregel wordt RESTRICTED UPDATED gehanteerd.

Gevraagd: ontwerp de query(‘s) die nodig is (zijn) om de wijzigingen door te voeren. Zijn er meer query’s nodig, schrijf ze op in de volgorde zoals ze na elkaar uitgevoerd moeten worden. **[6]**

|  |
| --- |
| INSERT INTO vakgroep  values ('BITM', 'Business IT en Management');  INSERT INTO vakgroep  values ('SE', 'Software Engineering');  UPDATE docent  set vakgroep = 'BITM'  WHERE vakgroep = 'IS';  UPDATE docent  set vakgroep = 'SE'  WHERE vakgroep = 'ST';  DELETE FROM vakgroep  WHERE code = 'IS';  DELETE FROM vakgroep  WHERE code = 'ST';  Na:  SELECT \*  FROM vakgroep |

1. Bij een cursus hoort studiemateriaal. Hier worden voornamelijk boeken en online-cursussen als studiemateriaal gebruikt in het onderwijs.   
   De tabel LESMATERIAAL moet nu worden aangemaakt.  
   De tabel kent de volgende verplichte velden: code (uniek nummer, het sleutelveld), titel (titel van het boek of de URL) en auteur (naam van de schrijver of van de organisatie die het online-materiaal beschikbaar heeft gesteld.  
   Gevraagd:

Het SQL-statement om de tabel lesmateriaal aan te maken. **[4]**

|  |
| --- |
| create table lesmateriaal  (code int not null,  titel varchar (40) not null,  auteur varchar (25) not null,  constraint pk\_lesmateriaal primary key(code) rood mag hoeft niet  ); |

1. De tabel LESMATERIAAL heeft een relatie met de tabel CURSUS. Om de vraag niet al te complex te maken wordt er bij deze vraag vanuit gedaan dat elk boek of elke URL slechts aan een enkele cursus verbonden is.  
   Gevraagd:

Het SQL-statement om de tabellen LESMATERIAAL en CURSUS met elkaar te verbinden. **[5]**

|  |
| --- |
| Stap 1: **[2]**  ALTER TABLE cursus  ADD les\_code INTEGER  Stap2: **[3]**  ALTER TABLE cursus  ADD CONSTRAINT fk\_cursus\_lesmateriaal rood mag, hoeft niet  FOREIGN KEY(les\_code) REFERENCES lesmateriaal(code); |

1. De enige voorwaarde die nu is meegegeven aan een tentamencijfer is dat een cijfer maximaal mag bestaan uit twee (2) tekens. Dat betekent dat het natuurlijk ook mogelijk is een 12 voor een tentamen te scoren of een 63. Dat kan uiteraard niet de bedoeling zijn. En lager dan een 1 gaat ook niet  
   Gevraagd:

Welk statement is nodig om te controleren dat het tentamencijfer binnen de gestelde grenzen ligt? **[4]**

|  |
| --- |
| ALTER TABLE tentamen ADD CONSTRAINT ch\_cijfer check(cijfer >= 1 AND cijfer <= 10); |

1. Gevraagd:  
   Op de administratie van de school komen regelmatig vragen binnen over vrijstellingen. De medewerkster Julia heeft gevraagd een view aan te maken alleen alle vrijstellingen te zien zijn die het huidige schooljaar zijn toegekend. Julia krijgt wel de gelegenheid om nieuwe vrijstellingen in te voeren.  
   Daarbij dient de voorwaarde WHERE-clausule te fungeren als invoercontrole.   
   Stap 1: Creëer deze view! De velden student, cursus, datum en inschrijving zijn hiervoor nodig. **[5]**  
   Stap 2: Welk SQL-statement moet gebruikt worden om deze gegevens te kunnen bekijken? **[1]**

|  |
| --- |
| Stap 1: CREATE VIEW vVrij AS SELECT student, cursus, datum, vrijstelling FROM inschrijving  WHERE datum>31-08-2020 AND vrijstelling = ‘J’ with check option  hiermee wordt gecontroleerd of de rijen aan de voorwaarde voldoen  Stap 2: SELECT \* FROM vVrij |

1. De view is aangemaakt en klaar voor gebruik, maar Julia kan er nog niets mee. Ze heeft nog geen rechten gekregen. Om de view te kunnen gebruiken moet ze rechten krijgen om records toe te kunnen voegen aan de tabel INSCHRIJVING en rechten om de view te kunnen bekijken en om er records aan toe te voegen. Ook moet ze dezelfde rechten aan het collega’s op de afdeling administratie kunnen toekennen. [**5**]

|  |
| --- |
| Stap 1: **[2]** CREATE USER Julia PASSWORD 'vrij'  Stap 2: **[3]**  GRANT INSERT ON inschrijving TO Julia WITH GRANT OPTION; GRANT SELECT, INSERT ON vVrij TO Julia WITH GRANT OPTION |

**EINDE TOETS**

**ANTWOORDMODEL**

**DEEL A**

KLANT **(k-nr**, k-naam, k-adres, k-pc, k-plaats, k-cont) 3

sleutel 1  
velden 1

naam 1

Lease (**k-nr, kent**) 3

Dagen (**k-nr, kent, datum**) 4

Samen voor beide tabellen

sleutel 3

velden 1

koppeling 2

naam 1

Worden deze tabellen weggelaten worden de punten verdeeld over de andere tabellen

Chopper(**kent**, merk) 3

Sleutel 1

Velden 1

Naam 1

Problemen (**k-nr, kent, datum**, **r-nr**) 9

Sleutel 4

Velden 2

Naam 1

Koppeling 2  
  
Reparatie **(r-nr,** r-oms, r-kost) 3

Sleutel 1

Velden 2

Naam 1

**DEEL B SQL**

1. [**4]**

|  |
| --- |
| SELECT naam, credits  FROM cursus  WHERE examinator <> 'DAT'  ORDER BY naam;  SELECT 1  FROM 1  WHERE 1  ORDER BY 1 |

2. [**5**]

|  |
| --- |
| SELECT student, cursus, datum, iif(volgnr = 1, 'Poging 1', 'Poging 2') poging  FROM tentamen  SELECT 1  IIF 3  FROM 1 |

3. [**5**]

|  |
| --- |
| SELECT d.naam docent, vd.naam vakgroep\_docent, v.naam vervanger  FROM docent d  JOIN vakgroep vd ON d.vakgroep = vd.code  JOIN docent v ON v.vervanger = d.acr  SELECT 1  JOINS 2 x 2 |

4. [**5**]

|  |
| --- |
| SELECT c.code, c.naam, count(\*) as aantal\_inschrijvingen  FROM cursus c  JOIN inschrijving i on c.code = i.cursus  GROUP BY c.code, c.naam  COUNT 2  JOIN 2  GROUP BY 1 |

5. [**5**]

|  |
| --- |
| SELECT cursus, AVG(cijfer) AS gemiddeld\_cijfer  FROM tentamen  GROUP BY cursus  HAVING AVG(cijfer) >6  AVG 1  GROUP BY 2  HAVING 2 |

6. [**5**]

|  |
| --- |
| SELECT naam FROM cursus WHERE code IN  (SELECT cursus  FROM voorkenniseis  WHERE voorkennis IN ('DB', 'SW')) ORDER BY naam  WHERE 1  SUBSELECT 3  ORDER BY 1 |

7. [**6**]

|  |
| --- |
| SELECT naam  FROM docent  WHERE acr IN  (SELECT docent  FROM begeleider  WHERE cursus NOT IN  (SELECT code  FROM cursus  WHERE naam ='Discrete Wiskunde'  OR naam = 'Informatiemodelleren'))  SUBSELECT 2 x 3 |

8. [**6**]

|  |
| --- |
| INSERT INTO vakgroep  values ('BITM', 'Business IT en Management');  INSERT INTO vakgroep  values ('SE', 'Software Engineering');  UPDATE docent  set vakgroep = 'BITM'  WHERE vakgroep = 'IS';  UPDATE docent  set vakgroep = 'SE'  WHERE vakgroep = 'ST';  DELETE FROM vakgroep  WHERE code = 'IS';  DELETE FROM vakgroep  WHERE code = 'ST';  Per juiste optie 1 punt (syntax + juiste plek in rijtje) |

9. [**4**]

|  |
| --- |
| create table lesmateriaal  (code int not null,  titel varchar (40) not null,  auteur varchar (25) not null,  constraint pk\_lesmateriaal primary key(code)  );  CREATE 1  JUISTE Velden + typen 1  NOT NULL 1  CONSTRAINT 1 |

10. [**5**]

|  |
| --- |
| Stap 1: **[2]**  ALTER TABLE cursus  ADD les\_code INTEGER  ALTER 1 ADD 1  Stap2: **[3]**  ALTER TABLE cursus  ADD CONSTRAINT fk\_cursus\_lesmateriaal  FOREIGN KEY(les\_code) REFERENCES lesmateriaal(code);  CONSTRAINT 1  JUISTE FOREIGN KEY 1  REFERENCES 1  Is ook juist als code bij boek staat. |

11. [**4**]

|  |
| --- |
| ALTER TABLE tentamen ADD CONSTRAINT ch\_cijfer check(cijfer >= 1 AND cijfer <= 10);  CONSTRAINT 1  ch 1  VOORWAARDE 2 |

12. [**6**]

|  |
| --- |
| Stap 1: [5] CREATE VIEW vVrij AS SELECT student, cursus, datum, vrijstelling FROM inschrijving  WHERE datum>’31-08-2020’ AND vrijstelling = ‘J’ with check option  CREATE met v 2 VOORWAARDE 2 CHECK OPTION 1  Stap 2: [1] SELECT \* FROM vVrij  Goed of fout |

13. [**5**]

|  |
| --- |
| Stap 1**:** [2] CREATE USER Julia PASSWORD 'vrij'  CREATE USER 1 PASSWORD 1  Stap 2: [3]  GRANT INSERT ON inschrijving TO Julia WITH GRANT OPTION  GRANT SELECT, INSERT ON vVrij TO Julia WITH GRANT OPTION  GRANT ACTIES 1  JUISTE BESTANDEN en VOLGORDE  GRANT OPTION 1 |