DBSQL WERKBOEK_

Opleiding HBO-ICT - Propedeuse

Mark Achten
Jorg Janssen

12 December 2024

HAN_UNIVERSITY
OF APPLIED SCIENCES

INHOUDSOPGAVE

1 INLEIDING

1.1 INTRO

Voor je ligt het werkboek voor het vak **DBSQL**, ofwel **DATSTQ03: Databases - Structured Query Language**. Het bevat oefeningen in de onderwerpen die je behandeld krijgt.

1.2 DOEL

SQL leer je niet van alleen theorie, je moet het vooral veel doen. Het doel van dit werkboek is om je zelfstandig (extra) te laten oefenen met SQL. Je vindt hierin diverse oefeningen van verschillend nivo. Je kunt vanaf de eerste les al oefeningen uit dit werkboek maken.

1.3 WERKWIJZE

Na iedere les heb je voldoende kennis om de opgaven te maken die in dit werkboek bij het betreffende thema staan. Oefeningen die als huiswerk zijn opgegeven zullen daarna zo goed als altijd ook in de klas worden behandeld. Ieder hoofdstuk heeft oefeningen over het betreffende thema en die hebben altijd eerst met de Muziekdatabase te maken. Je vindt die dan ook altijd als eerste paragraaf. Het advies is om per hoofdstuk en paragraaf één sql-bestand te maken met daarin jouw uitwerkingen. Bedenk dat er soms meerdere mogelijkheden zijn om tot hetzelfde eindresultaat te komen. Het kan dus voorkomen dat je een uitwerking gemaakt hebt die afwijkt van de antwoorden. Probeer dan zelf te bepalen of jouw uitwerking ook goed is. Let daarbij op dat als de resutaten van de query's overeenkomen het niet per se zo hoeft te zijn dat de query dan goed is. Bespreek dit bij twijfel altijd met je docent. Als je het niet vraagt, weet je ook niet of het goed is…

1.4 DATABASES

De oefeningen zijn onder andere gebaseerd op de muziekdatabase die je kent uit de lessen. Daarnaast gebruikt het werkboek nog een aantal andere databases. De schema's van deze databases en hoe je ze in je eigen SQL Server omgeving krijgt vind je op OnderwijsOnline.

2 EENVOUDIGE QUERY'S

In dit hoofdstuk ga je oefenen met eenvoudige query's op één tabel. Ook ga je zien hoe je kunt filteren met de `WHERE`-clause, van eenvoudig tot wat moeiljker. Tenslotte komt ook het sorteren aan bod.

2.1 MUZIEKDATABASE

1. Geef alle informatie van alle stukken.
2. Geef de naam en geboortedatum van alle componisten.
3. Geef stuknr, titel en genre van alle stukken.
4. Geef alle informatie van klassieke stukken.
5. Geef de namen en geboortedatums van die componisten die na 1920 geboren zijn. Tip: je kunt in veel programmeertalen gebruik van datum-functies, ook in SQL. Eén van die functies is de `YEAR()` functie die van een gegeven datum het jaar teruggeeft.
6. Geef alle informatie van speelstukken, ofwel stukken waarvan een niveau bekend is.
7. Geef alle informatie van klassieke stukken van niveaucode A met een speelduur van meer dan 5 minuten.
8. Geef stuknr en titel van klassieke stukken waar het niveau niet bekend is.
9. Geef de stukken die tussen 1950 en 1980 geschreven zijn. Schrijf de query op twee manieren.
10. Geef stuknummer en titel van alle jazz-stukken. Sorteer op titel.
11. Geef van alle speelstukken van na 1995 het stuknummer, het genre, de niveaucode en de speelduur. Sorteer het overzicht op speelduur (van groot naar klein) en bij gelijke speelduur op genre.
12. Geef alle stukken van niveau A of B maar sowieso van origineel 1. Schrijf ook deze query op twee manieren.
13. Geef alle info van componisten waarbij ergens in de naam 'mozart' staat.
14. Geef stuknummer, titel en genre van alle stukken die een bewerking zijn van stuk 5. Sorteer op titel.
15. Geef de componistId, de naam en de geboortedatum van alle componisten die als docent verbonden zijn aan een muziekschool. Sorteer op geboortedatum (van jong naar oud).
16. Bij het sorteren van het resultaat kun je ook getallen gebruiken. Bijvoorbeeld:

    ```
    SELECT  titel, jaartal
    FROM    Stuk
    ORDER   BY jaartal;
    ```

    Deze query levert hetzelfde op als:

    ```
    SELECT  titel, jaartal
    FROM    Stuk
    ORDER   BY 2;
    ```

    Hier wordt op de 2e kolom gesorteerd en dat is `jaartal`. Kun je voor- en nadelen bedenken van de ene manier en van de andere manier? En als je één manier zou moeten aanraden, welke zou dat dan zijn?

2.2 PREMIER DATABASE

1. Vermeld het onderdeelnummer en de -beschrijving van alle onderdelen.
2. Maak een lijst van alle rijen en kolommen voor de volledige REP-tabel.
3. Maak een lijst van de namen van klanten met een kredietlimiet van ten minste $10.000.
4. Vermeld het bestelnummer voor elke bestelling geplaatst door klantnummer 148 op 23-10-2003. (Tip: om een datum in een voorwaarde in te voeren, voer je de dag van de maand, een koppelteken, de afkorting van drie tekens voor de maand, nog een koppelteken en het jaar in. De datum moet tussen enkele aanhalingstekens worden geplaatst. Om november in te voeren 15, 2003, bijvoorbeeld, typ je dus '15-nov-2003' in de WHERE-component).
5. Maak een lijst van nummer en naam van elke klant die wordt vertegenwoordigd door verkoper 20 of verkoper 65.
6. Vermeld het onderdeelnummer en de -beschrijving van elk onderdeel dat niet in artikelklasse HW valt.
7. Maak een lijst van het onderdeelnummer, de beschrijving en het aantal beschikbare eenheden voor elk onderdeel dat tussen de 10 en 25 eenheden op voorraad heeft.
8. Maak een lijst van het onderdeelnummer, de onderdeelbeschrijving en de voorhanden waarde (voorhanden eenheden * eenheidsprijs) van elk onderdeel in artikelklasse AP. (Voorhanden waarde is in werkelijkheid beschikbare eenheden * kosten, maar er is geen COST-kolom in de PART-tabel). Wijs de naam ON_HAND_VALUE toe aan de berekening.
9. Maak een lijst van het onderdeelnummer, de onderdeelbeschrijving en de voorhanden waarde voor elk onderdeel waarvan de voorhanden waarde minimaal $ 7.500 is. Wijs de naam ON_HAND_VALUE toe aan de berekening.
10. Gebruik de IN-operator om het onderdeelnummer en de onderdeelbeschrijving van elk onderdeel in itemklasse HW of SG weer te geven.
11. Zoek het nummer en de naam van elke klant wiens naam begint met de letter "D".
12. Maak een lijst van alle details over alle onderdelen. Sorteer de uitvoer op onderdeelbeschrijving.
13. Maak een lijst van alle details van alle onderdelen. Sorteer de uitvoer op onderdeelnummer en artikelklasse.

3 AANMAKEN EN VULLEN TABELLEN

In dit hoofdstuk ga je oefenen met aanmaken en vullen van tabellen.

3.1 MUZIEKDATABASE

Zorg er bij deze opdrachten voor dat je dit uitvoert in de juiste database: de **MUZIEKDATABASE**.

1. Tabel **Orkest**.
   a. Schrijf het CREATE TABLE statement voor een tabel genaamd **Orkest** met de volgende kolommen:
      - **naam**, varchar(100), verplicht
      - **plaats**, varchar(50), verplicht
      - **url**, varchar(100), optioneel.

      Verder:
      - De primary key ligt op de kolom **naam**. Geef deze sleutel een betekenisvolle naam (bijvoorbeeld `PK_` gevolgd door de naam van de tabel).

   b. Voeg twee rijen toe aan deze tabel, één orkest met een waarde voor **url** en één zonder. Bedenk zelf de namen van de orkesten of gebruik internet ter inspratie.
   c. Kun je nu twee orkesten met dezelfde naam in dezelfde plaats toevoegen?
   d. En kun je meerdere orkesten hebben in dezelfde plaats?
2. Tabel **Muzikant**.
   a. Schrijf het CREATE TABLE statement voor een tabel genaamd **Muzikant** met de volgende kolommen:
      - **lidnr**, numeric(7), verplicht
      - **naam**, varchar(100), verplicht
      - **instrument**, varchar(100), verplicht

      Verder: De primaire sleutel ligt op de kolom **lidnr**. Geef deze sleutel een betekenisvolle naam.
   b. Kan dezelfde muzikant meerdere instrumenten bespelen volgens deze tabel?
   c. Voeg ook nu aantal zelf bedachte rijen toe aan deze tabel.
3. Schrijf nu twee SQL-statements om de tabellen **Orkest** en **Muzikant** te verwijderen.

4 REFERENTIËLE INTEGRITEIT

In dit hoofdstuk ga je oefenen met aanmaken en vullen van complexere tabellen: tabellen met foreign en alternate keys.

4.1 MUZIEKDATABASE

1. Welke constraints ken je nu allemaal?
2. Noem een aantal verschillen tussen een primaire sleutel en een alternatieve sleutel.
3. Wat betekenen de volgende termen als het gaat om een `FOREIGN KEY`?
   a. Parent tabel
   b. Child tabel
   c. Mag een kolom in de child tabel, die onderdeel is van een foreign key, optioneel zijn?
4. Tabel **Concertgebouw**:
   a. Maak een nieuwe tabel genaamd **Concertgebouw** met de onderstaande kolommen:
   – **naam**, varchar(100), verplicht
   – **plaats**, varchar(50), verplicht
   – **url** varchar(100), optioneel.

   De primaire sleutel van deze tabel ligt over de kolom **naam** en **plaats**. Geef deze PK een betekenisvolle naam. Verder mogen er geen concertgebouwen zijn die dezelfde url hebben. Geef ook deze constraint een betekenisvolle naam.
   b. Kunnen er volgens deze tabel meerdere concertgebouwen zijn in dezelfde plaats?
   c. Kunnen er volgens deze tabel concertgebouwen zijn met dezelfde naam maar in verschillende plaatsen?
   d. Kunnen er meerdere gebouwen in deze tabel staan zonder url?
   e. Vul de tabel met minimaal twee rijen.
5. Tabel **Zaal**:
   a. Maak een nieuwe tabel aan genaamd **Zaal** met de onderstaande kolommen:
   – **naam**, varchar(50), verplicht. (naam van de zaal)
   – **gebouw**, varchar(100), verplicht. (naam van het concertgebouw)
   – **plaats**, varchar(50), verplicht.
   – **capaciteit**, numeric(4), verplicht. (hoeveel bezoekers kunnen er in deze zaal) De primaire sleutel van deze tabel ligt over de kolommen **naam**, **gebouw** en **plaats**. De concertgebouwen moeten bestaande concertgebouwen zijn (ofwel: ze moeten voorkomen in de tabel **Concertgebouw**).
   b. Kunnen er zalen zijn met dezelfde naam maar in verschillende concertgebouwen?
   c. Vul ook deze tabel, nu met minimaal vijf rijen.
   d. Ben je bij de vorige vraag tegen een fout aangelopen? Zo ja, wat was die fout en verklaar waarom die optrad. Kreeg je geen fouten? Waar heb je dan bewust rekening mee gehouden?

6. Tabel **Uitvoering**:

   a. Maak een nieuwe tabel aan genaamd **Uitvoering** met de onderstaande kolommen:

   – **stuknr**, numeric(5), verplicht
   – **zaal**, verplicht (kies hier zelf het beste datatype voor)
   – **gebouw**, verplicht (kies hier zelf het beste datatype voor)
   – **plaats**, verplicht (kies hier zelf het beste datatype voor)
   – **datum**, verplicht (kies hier zelf het beste datatype voor)
   – **bezoekers**, optioneel (kies hier zelf het beste datatype voor) De primaire sleutel van deze tabel ligt over de kolommen **stuknr**, **zaal**, **gebouw**, **plaats** en **datum**. De concertzalen moeten bestaan in tabel **Zaal** en de stuknummers moeten bestaande stukken zijn.

   b. Misschien heb je al bedacht dat het aantal bezoekers niet altijd al ingevuld kan worden (daarom is die ook optioneel). Want als een uitvoering nog moet komen dan weet je nog niet hoeveel bezoekers er zullen zijn. Er is nog zo'n regel die te maken heeft met het aantal bezoekers. Weet jij welke? Voorlopig mag je dit negeren, we zien later in deze course hoe we dit beter kunnen bewaken.

   c. Voeg aan deze tabel minimaal drie rijen toe.

7. Schrijf nu een drietal SQL-statements om de tabellen **Concertgebouw**, **Zaal** en **Uitvoering** te verwijderen.

5 TOEVOEGEN, WIJZIGEN EN VERWIJDEREN DATA

In dit hoofdstuk ga je leren om data in tabellen bij te werken, te verwijderen of te kopieren naar andere tabellen.

5.1 MUZIEKDATABASE

1. Het blijkt dat stuknr 10 een bewerking is van stuk 2. Schrijf een UPDATE-query die dat in de tabel bijwerkt. Schrijf daarna een SELECT-query om te kijken of je update gelukt is.
2. Voeg een nieuw genre toe aan de genre tabel: Rock. Controleer met een SELECT-query of de INSERT is gelukt.
3. Verander het genre van 'Swinging Lina' (stuknr 13) naar rock en het niveau naar vergevorderden (= C). Ook nu graag testen om de UPDATE te controleren.
4. De speelduur van Blue bird (stuknr 1) klopt niet. Haal deze speelduur maar weg. Ook hier weer testen.
5. Verander het genre van 'Swinging Lina' (nr 13) terug naar jazz.
6. Verwijder het eerder toegevoegde genre 'Rock'.
7. Selecteer alle klassieke stukken en kopieer die naar een nieuwe tabel **KlassiekeStukken**.
    a. Doe dit met één statement.
    b. Heeft deze tabel exact dezelfde structuur als de tabel **Stuk**?
    c. Verwijder de tabel **KlassiekeStukken** weer.
    d. Maak nu weer die tabel **KlassiekeStukken**, maar nu door in de `WHERE` ervoor te zorgen dat er geen enkele rij geretourneerd wordt. Je maakt dus een 'kopie' van de tabel zonder data.
    e. Vul nu die tabel met een SELECT-statement op **Stuk** met alleen de klassieke stukken.
    f. Verwijder de tabel **KlassiekeStukken** weer.
8. Een moeilijke: Het Muziekpakhuis (id 3) is verhuisd naar 's Hertogenbosch en heet nu: Het Bosche Huus. Pas dit aan. HINT: Je moet nu proberen om die quote van **'**s Hertogenbosch mee te nemen in de waarde van het veld **plaatsnaam**. SQL Server moet dus die quote niet zien als begin of eind van een stuk tekst, maar als onderdeel van een stuk tekst. Dat heet 'escaping'. Zoek op het internet op hoe je dat in SQL Server doet.

5.2 OPENSCHOOL

1. Geef het SQL statement dat de examinator verwijdert voor cursus DB. Let op, niet de docent verwijderen!!
2. Men wil graag het aantal uren dat voor een cursus nodig is verhogen met 10%. Hoe luidt het SQL statement waarmee dat in één keer mogelijk wordt gemaakt?
3. Beschrijf wat er moet gebeuren als men van de course DB een tweede examinator (namelijk DAT) wil vastleggen.

6 WIJZIGEN EN VERWIJDEREN VAN TABELLEN

In dit hoofdstuk ga je leren om de structuur van tabellen bij te werken. Je gaat oefenen met het aanmaken van kolommen in bestaande tabellen, het wijzigen van kolommen en je oefent met het verwijderen van kolommen en tabellen.

6.1 MUZIEKDATABASE

1. We willen graag de genre tabel uitbreiden met een kolom waarin we meer informatie over dat genre kunnen opnemen. Gebruik als kolomnaam **info**, type `varchar(255)`.
   a. Schrijf het SQL-statement om de kolom aan de tabel toe te voegen.
   b. Maakt het hier uit of je NULL of NOT NULL gebruikt?
   c. Voeg voor alle genres in de nieuwe kolom een link toe naar een (relevante) corresponderende wikipedia-pagina. Schrijf hiervoor dus een aantal UPDATE-statements om dat voor alle genres te doen. Zoek de genres op Wikipedia (https://wikipedia.org) op en gebruik die url in je query's.
   d. Nu alle rijen een waarde hebben voor die nieuwe kolom mag de kolom verplicht worden gemaakt. Schrijf het SQL-statement daarvoor.
   e. Om te zorgen dat de info betrouwbaar is willen we er zeker van zijn dat er niet meerdere genres dezelfde info hebben. Schrijf een SQL-statement om dat te realiseren.
   f. Verwijder de constraint en de kolom.
   g. Maakt de volgorde van de DROP-statements hierboven nog uit?
2. De geboortedatum van een componist bevat nu ook de tijd. Dat is niet nodig. Verander het datatype van die kolom naar datatype `date`.

6.2 OPENSCHOOL

1. Geef het SQL statement dat de examinator verwijdert voor cursus DB. Let op, niet de docent verwijderen!!
2. Men wil graag het aantal uren dat voor een cursus nodig is verhogen met 10%. Hoe luidt het SQL statement waarmee dat in één keer mogelijk wordt gemaakt?
3. Eerder heb je beschrreven wat er moet gebeuren als men van de course DB een tweede examinator (namelijk DAT) wil vastleggen. Schrijf nu de SQL-statements om dat voor elkaar te krijgen.

7 REFERENTIËLE INTEGRITEITSREGELS

In dit hoofdstuk ga je oefenen met wat de effecten kunnen zijn op andere tabellen als data bijgewerkt of verwijderd wordt.

7.1 MUZIEKDATABASE

1. Beschrijf in eigen woorden wat het eigenlijk inhoudt: een relatie tussen twee tabellen.
2. Een relatie tussen tabellen implementeer je bij een `FOREIGN KEY` constraint.
   a. Op welk van de twee tabellen implementeer je de constraint? De parent-tabel of de child-tabel?
   b. Waarom daar?
   c. Als er een foreign key constraint wordt gemaakt, moet dan iedere waarde in de parent-kolom voorkomen in de child-kolom?
   d. Als er een foreign key constraint wordt gemaakt, hoe vaak kan een waarde die in de parent-kolom staat dan voorkomen in de child-tabel?
   e. Welke acties op de parent-tabel zouden problemen op kunnen leveren als het gaat om referentiële integriteit? Geef bij elk van die acties een voorbeeld.
   f. Welke acties op de child-tabel zouden problemen op kunnen leveren als het gaat om referentiële integriteit? Geef bij elk van die acties een voorbeeld.
   g. Vul nu, op basis van je antwoorden hierboven de laatste kolom in de volgende tabel verder in. De kolom **Maatregel** geeft aan welke maatregel toegepast dient te worden. Gebruik alleen de opties:
      – nvt (niet van toepassing)
      – RI regel
      – FK constraint

Actie	Op tabel	Maatregel
INSERT	parent	
UPDATE	parent	
DELETE	parent	
INSERT	child	
UPDATE	child	
DELETE	child	

3. Beschrijf van de onderstaande SQL-statements welke rijen gewijzigd danwel verwijderd worden en motiveer je bevindingen. TIP: gebruik het CREATE-script voor de Muziekdatabase om te zien hoe de `FOREIGN KEY` constraints zijn aangemaakt.
   a. UPDATE Niveau SET niveaucode = ‘E’ WHERE niveaucode = ‘A’;
   b. DELETE FROM Niveau WHERE niveaucode = ‘C’;
   c. DELETE FROM Niveau WHERE niveaucode = ‘B’;
   d. UPDATE Stuk SET niveaucode = ‘F’ WHERE niveaucode = ‘B’;
   e. UPDATE Stuk SET stuknr = 99 WHERE stuknr = 1;
   f. UPDATE Stuk SET stuknr = 99 WHERE stuknr = 2;

8 DEFINITIE VAN CONSTRAINTS

In dit hoofdstuk ga je oefenen met het bedenken en implementeren van CHECK-constraints.

8.1 MUZIEKDATABASE

1. In de Muziekdatabase is al een groot aantal constraints aanwezig. Maar ook nog niet. Voor deze vraag mag je uitgaan van de oorspronkelijke database zoals je die in les 1 hebt aangemaakt.
    a. Probeer zoveel mogelijk, nog niet geïmplementeerde constraints, te ontdekken. Begin met ze in woorden te beschrijven en denk out-of-the-box. Geef iedere constraint een nummer.
    b. Geef van deze constraints aan óf je ze kunt implementeren en als je denkt dat dat kan, geef dan de SQL-instructies daarvoor. Gebruik zinvolle namen.
    c. Schrijf nu voor iedere bedachte constraint twee SQL-testquery's (update of insert, dat maakt niet uit): één die niet voldoet aan de constraint en één die wel voldoet. Ofwel: een negatieve testquery en een positieve testquery.
        – TIP: Je hoeft bij een INSERT niet voor álle kolommen een waarde op te geven. Doe dat alleen voor de verplichte kolommen, dat scheelt wat typen.
2. Geef aan hoe je de volgende beperkingsregel afdwingt in het ALTER TABLE statement van de tabel Stuk: een stuk is nooit een bewerking van zichzelf.
3. Gegeven is de volgende beperkingsregel: een bewerking is altijd gemaakt van een origineel, en dus nooit van een andere bewerking. Beschrijf zo precies mogelijk wat dit betekent voor de tabel Stuk. Je hoeft het SQL-statement NIET te geven!!!

9 JOIN QUERY'S

9.1 INNER JOINS

In dit hoofdstuk ga je oefenen met het schrijven van query's met een `INNER JOIN`.

9.1.1 MUZIEKDATABASE

1. Geef alle informatie van componisten en hun gerelateerde stukken. Schrijf de query op twee manieren (oude en nieuwe syntax).
2. Geef van alle stukken met een niveaucode (de speelstukken) stuknr, titel en de omschrijving van het niveau. Schrijf de query op twee manieren.
3. Geef een lijst met namen van componisten en de naam van de school waaraan ze verbonden zijn/waren. Schrijf deze query op twee manieren.
4. Geef van alle jazz-stukken ouder dan 1950 het stuknr, de titel, jaar en de naam van de componist. Sorteer aflopend op jaartal.
5. Geef per klassiek stuk het stuknr, de titel, de naam van de componist en de naam van de muziekschool waar de componist aan verbonden is/was. Als een componist niet verbonden is aan een muziekschool dan hoeft die niet opgenomen te worden in het resultaat. LET OP: je hebt hier drie tabellen voor nodig.
6. Geef een overzicht van alle bewerkingen die zijn gecomponeerd door componisten die aan een muziekschool zijn/waren verbonden. Geef stuknr en titel van deze bewerkingen.
7. Geef van alle stukken het stuknummer en de titel met ' – ' en de componistnaam in één kolom genaamd **Stukinfo**. Sorteer op stuknr.
8. Geef van alle stukken het stuknr, titel, jaartal, de naam van de componist, het geboortejaar van de componist en het verschil tussen het jaar van schrijven en het jaar van geboorte.
9. Geef een lijst van stukken (stuknr en titel) waar een piano in wordt gebruikt.
10. Nogmaals een lijst van stukken (stuknr en titel) waar een saxofoon in wordt gebruikt.
    a. Wat valt op aan het resultaat?
    b. Hoe komt dat?
    c. Welk sleutelwoord moet je toevoegen aan de query om alle resutaten maar eenmaal te zien?
11. Geef een overzicht van alle titels van bewerkingen van stukken met hun originele titel. Toon daarbij ook de stuknr's van bewerking en origineel.
    – HINT: je moet hier een self-join gebruiken.
    – HINT: Bij het gebruik van een self-join (een join van een tabel met zichzelf) MOET je tabelaliassen gebruiken.

9.1.2 OPENSCHOOL DATABASE

1. Geef een lijst van mentoren en de studenten waarvan ze mentor zijn. Sorteer op naam mentor en daarbinnen op naam van de student.
2. Geef de studenten die ooit een onvoldoende voor een tentamen hebben gekregen. Geef naam van de student, naam van het vak, het cijfer en datum.
3. Geef een lijst van cursussen waar de examinator ook een begeleidend docent is (voor een willekeurig vak). In de output graag geen dubbele cursussen.

9.1.3 PREMIER DATABASE

1. Geef bestelnummer, datum en klantnaam voor elke bestelling. Sorteer op naam en datum van de klant.
2. Maak een lijst van elke klant met hun vertegenwoordiger (hun namen: achternaam, voornaam). Noem die kolom **REP**. LET OP: er zitten spaties in de namen. Zoek op internet hoe je de spaties kunt verwijderen. HINT: zoek op `TRIM`.
3. Vermeld voor elke bestelling het bestelnummer, de datum, het onderdeelnummer, de onderdeelbeschrijving en het aantal. Sorteer op een logische manier.

9.2 OUTER JOINS

In dit hoofdstuk ga je oefenen met het schrijven van query's met een `OUTER JOIN`.

9.2.1 MUZIEKDATABASE

1. Geef een lijst van alle stukken (stuknr en titel) met een omschrijving van het niveau. Geef ook die stukken waarvoor geen omschrijving gevonden kan worden.
2. Geef nu een lijst met alle niveau omschrijvingen en de stuknr's.
3. Pas de query uit de vorige aan zodat je die niveaus ziet waarvoor geen stukken geschreven zijn.
4. Geef een lijst van alle componisten met hun eventuele stukken. Geef naam van de componist met het nummer van het stuk en de titel. Houd ook rekening met componisten die geen stukken hebben geschreven.
5. Zijn er componisten die geen stukken hebben geschreven? Pas de query uit de vorige vraag aan zodat je daar antwoord op krijgt.
6. Welke klassieke stukken hebben een onbekende bezetting? Geef stuknr en titel.
7. Zijn er genres waarvoor geen stukken bestaan?
8. Geef een lijst van alle genres met hun eventuele stukken (alleen stuknr en titel).
    a. Test deze query door een genre aan de **Genre** tabel toe te voegen en te kijken of deze dan in het resultaat verschijnt. Schrijf dus een INSERT statement om een genre toe te voegen en voer de eerste query nogmaals uit.
    b. Als de query goed is, kun je het toegevoegde genre weer verwijderen. Schrijf daar een SQL-statement voor.
9. Geef een unieke lijst met namen van alle componisten en genres waar ze stukken voor geschreven hebben. Sorteer op genre, dan op naam van de componist.
10. Geef een lijst van stukken (stuknr en titel) waar geen bewerking voor is geschreven.

9.2.2 OPENSCHOOL DATABASE

1. Voor welke vakken geldt dat ze niet als voorkennis worden beschouwd voor andere vakken? Geef code en naam van de cursus, gesorteerd op naam.
2. Geef een lijst van cursussen waar de examinator juist geen begeleidend docent is (voor een willekeurig vak).

9.2.3 PREMIER DATABASE

1. Maak een lijst van de onderdelen die nooit zijn besteld.
2. Maak een lijst van verkopers zonder enige klant.
3. Zijn er klanten die nooit iets besteld hebben?

10 STATISTISCHE QUERY'S

10.1 DEEL 1

In dit hoofdstuk ga je onder andere oefenen met het schrijven van query's met een `GROUP BY`.

10.1.1 MUZIEKDATABASE

1. Wat is de langste speelduur van een stuk?
2. Wat is de gemiddelde speelduur van klassieke stukken?
3. Geef per genre waar een stuk voor geschreven is het aantal stukken. Sorteer op aantal aflopend.
4. Geef per stuk het stuknr en het totaal aantal instrumenten.
5. Hoeveel instrumenten spelen er in totaal mee bij stuk 2?
6. Geef per niveau waar een stuk voor geschreven is het aantal stukken. Sorteer op aantal aflopend.
7. Hoeveel componisten zijn niet verbonden aan een muziekschool?
8. Geef per muziekschool waar een componist aan verbonden is, het aantal componisten.
9. Geef een lijst van namen van componisten en het aantal stukken dat hij/zij geschreven heeft.
10. Geef de totale speelduur van alle stukken.
11. Geef per jaar waarin stukken geschreven zijn, het aantal stukken. Sorteer oplopend op jaar.
12. Geef een overzicht met per muziekschool het aantal componisten. Geef naam van de school en het aantal componisten.
13. Geef per niveau de omschrijving en het aantal stukken dat ervoor geschreven is.
14. Geef per genre het aantal componisten die er stukken voor geschreven hebben.
15. Geef een lijst van instrumenten met toonhoogten en hoevaak ze in totaal in stukken voorkomen.
16. Zoals de vorige, maar nu zonder de toonhoogte erbij.

10.2 DEEL 2

In dit hoofdstuk ga je oefenen met het schrijven van query's met een `GROUP BY` en `HAVING`. Ook worden de query's moeilijker door toepassing van outer joins.

10.2.1 MUZIEKDATABASE

1. Welk genre heeft van alle stukken een gemiddelde speelduur van meer dan 5 minuten?
2. Welk niveau heeft van alle stukken een gemiddelde speelduur van meer dan 5 minuten? Geef de niveaucode.
3. Voor welk niveau zijn meer dan drie stukken geschreven? Geef omschrijving en het aantal.
4. Welke componist heeft twee of meer stukken geschreven? Geef naam van de componist en het aantal stukken.
5. Geef een lijst van alle genres en het aantal stukken dat er voor geschreven is. Geef dus ook die genres waarvoor geen stukken geschreven zijn.
6. Geef een lijst van genres waar geen enkel stuk voor geschreven is. Schrijf deze query op zoveel mogelijk manieren.
7. Zijn er instrumenten die niet in een bezetting van een stuk voorkomen? Geef instrumentnaam en toonhoogte.
8. Geef voor alle componisten de naam en het aantal klassieke stukken stukken dat hij/zij geschreven heeft met daarbij de gemiddelde speelduur.
9. Geef voor alle genres het totaal aantal instrumenten dat bij alle stukken van dat genre gebruikt zijn.
10. Geef een lijst van de genres en niveaucodes waarvoor stukken geschreven zijn, samen met het aantal.

11 SUBQUERY'S

In het hoofdstuk ga je subquery's leren gebruiken. Je zult zien dat een aantal opgaven ook met een outer joini opgelost kan worden. Gebruik echter de subquery tenzij anders vermeld staat.

11.1 MUZIEKDATABASE

1. Geef de componistId en naam van alle componisten die klassieke stukken hebben geschreven.
2. Geef de componistId en naam van alle componisten die geen klassieke stukken hebben geschreven.
3. Voor welke genres is nog geen stuk geschreven?
4. Geef het stuknummer en de titel van het meest recent gecomponeerde muziekstuk.
5. Geef de naam van de oudste componist. Schrijf de query op twee manieren.
6. Geef het stuknummer en de titel van het minst recente klassieke speelstuk. LET OP: een speelstuk is dus een stuk met een niveaucode.
7. Geef die componisten die wel klassieke stukken maar geen jazz stukken hebben geschreven.
8. Welke componisten hebben alleen maar klassieke stukken geschreven? Geef id en naam van de componist.
9. Geef de namen van de componisten die een klassiek stuk hebben geschreven voor 'beginners'.
10. Geef die componist(en) die van alle componisten de laagste gemiddelde speelduur van hun stukken hebben.
11. Welk stuk heeft de meeste instrumenten in de bezetting? Geef stuknr, titel en aantal instrumenten.
12. Geef een lijst van stuknr en titel van stukken die geschreven zijn door componisten die verbonden zijn/waren aan een muziekschool uit Amsterdam.
13. Geef een lijst van componisten (naam) die minimaal één stuk hebben geschreven waar een piano in voorkomt.
14. In welk jaar of jaren zijn de meeste stukken geschreven? Geef het jaar of jaren met het aantal.
15. Van welk genre zijn de meeste stukken geschreven? Geef het genre (of genres) met het aantal.
16. Kun je de vorige uitwerking zo aanpassen dat je nu alleen die genres te zien krijgt waarvoor minder stukken geschreven zijn dan de meest voorkomende genres?

11.2 PREMIER DATABASE

1. Welke onderdelen zijn nooit besteld? Gebruik een subquery.
2. Vermeld de meest recente bestelling.
3. Welke klanten hebben twee of meer bestellingen geplaatst?
4. Maak een lijst van de klanten die meer dan eens op dezelfde dag hebben besteld. Vermeld de naam van de klant, de datum en het aantal geplaatste bestellingen. Gebruik de techniek die volgens jou het beste is.
5. Welke klant(en) hebben het vaakst besteld van alle klanten? Geef de naam van de klant op en het aantal keren dat de klant iets heeft besteld. Sorteer daarop.

6. Welk onderdeel is het meest besteld?
7. In welke stad hebben klanten het hoogste saldo? Geef staat, stad en dat saldo als TOTAL_BALANCE_COUNT.

12 USER DEFINED FUNCTIONS

12.1 ZONDER CHECK-CONSTRAINTS

12.1.1 MUZIEKDATABASE

1. Schrijf een user-defined function `UF_geef_geboortejaar_van_componist` die van een gegeven componistId het geboortejaar teruggeeft. De functie krijgt dus de componistId als input en retourneert een getal (het geboortejaar). Schrijf daarnaast ook een query om de functie te testen.
2. Schrijf een user-defined function `UF_geef_school_componist` die voor een opgegeven componistId de naam van de muziekschool teruggeeft. Als de componist niet aan een muziekschool gekoppeld is, mag er een NULL terug gegeven worden. Schrijf daarnaast ook een query om de functie te testen.
3. Schrijf een user-defined function `UF_geef_aantal_instrumenten_voor_stuk` die voor een opgegeven stuk het aantal instrumenten teruggeeft. Als voor een gegeven stuk geen bezetting bekend is, geef dan 0 terug. Schrijf de body van de functie met één statement. Schrijf daarnaast ook een query om de functie te testen.
   - HINT: Zoek op internet naar de functie `ISNULL()`. Gebruik die om 0 terug te geven bij een onbekende bezetting.
4. Schrijf een user-defined function genaamd `UF_geef_info_van_bewerking` die voor een gegeven stuknr kijkt of het een bewerking is. Als het een bewerking is, dan retourneert de functie het stuknr en de titel als één geheel (bv 1 - 'Blue Bird'). Als het geen bewerking is, dan retourneert de functie de tekst 'Dit is een origineel'. Schrijf daarnaast ook een query om de functie te testen.
5. **UITSMIJTER**: Een bewerking kan zelf ook weer bewerkt worden. Schrijf een UDF, die van een opgegeven stuknr het aller-origineelste stuknr teruggeeft. Dus stel, uitgaande van de oorspronkelijke vulling van de database, dat stuk 15 geen bewerking is van stuk 1 maar van stuk 2. Stuk 2 blijft een bewerking van stuk 1.
   a. Schrijf een UPDATE-statement die stuk 15 daarvoor bijwerkt.
   b. Schrijf nu de UDF zelf. Je krijgt een paar hints:
      - Je kunt gebruik maken van de `IIF()` functie. Deze functie roep je aan in de SELECT en krijgt drie argumenten:
        1. Conditie (is dit stuk een bewerking?)
        2. Actie als conditie TRUE is (selecteer origineel van stuk)
        3. Actie als conditie FALSE is (selecteer dan stuknr)
      - En in een UDF kun je diezelfde UDF weer gebruiken.

12.2 MET CHECK-CONSTRAINTS

We gaan nu user-defined functions in combinatie met CHECK-constraints gebruiken om complexere business rules te implementeren.

12.2.1 MUZIEKDATABASE

1. Als eerste de regel dat het jaartal van een stuk later in tijd moet zijn dan het geboortejaar van de componist.
    a. Controleer met een SQL statement of er nu stukken zijn die niet aan deze regel voldoen. Maak nog geen gebruik van de eerder geschreven functie (die uit het vorige hoofdstuk).
    b. Herschrijf deze query en gebruik nu de eerder geschreven functie.
    c. Schrijf nu een CHECK-constraint die dat voor je gaat doen.
    d. Schrijf nu twee SQL-statements waarbij je één stuk toevoegt aan de tabel `Stuk` dat niet voldoet aan de regel en één stuk toevoegt die wel voldoet.
2. We willen afdwingen dat alle bewerkingen later geschreven zijn dan het origineel.
    a. Controleer met een SQL statement of er nu stukken zijn die niet aan deze regel voldoen. HINT: hier heb je een self-join voor nodig.
    b. Schrijf een functie, die van een gegeven stuknr het jaartal teruggeeft.
    c. Schrijf nu een SQL statement die controleer of er stukken zijn die niet voldoen aan de regel. Gebruik de zojuist geschreven functie.
    d. Schrijf nu een CHECK-constraint die dit voor je gaat controleren.
    e. Schrijf nu wederom twee query's om deze CHECK-constraint te testen. Eén met een foutief geval en één met een correct geval.
3. We gaan nu eens kijken wat dit (de regel uit de vorige vraag) voor originele stukken betekent. Stuk 14 is een bewerking van stuk 5. Stuk 14 is later in tijd geschreven dan stuk 5.
    a. Schrijf een UPDATE-statement voor het origineel stuk 5 en verander het jaartal naar één jaar later dan stuk 14. Ofwel: het origineel (stuknr 5) is nu geschreven in 1999.
    b. Is de UPDATE gelukt? Hoe komt dat?
    c. Hoe zou je dat oplossen?

12.2.2 OPENSCHOOL DATABASE

Schrijf nu zelf CHECK-constraints die gebruik maken van zelf gemaakte user-defined functions om de business rules te implementeren.

1. Een tentamen moet altijd later in tijd zijn dan de inschrijving.
    a. Schrijf een query om te controleren of er gevallen zijn die niet aan de regel voldoen.
    b. Schrijf nu een functie die de datum van de inschrijving van een student op een cursus teruggeeft.
    c. Schrijf wederom een query om te controleren of er gevallen zijn die niet aan de regel voldoen maar gebruik nu de zojuist gemaakte functie.
    d. Schrijf nu een CHECK-constraint om dit te laten controleren.
    e. Schrijf twee SQL-statements om een tentamen toe te voegen waarbij de datum van het tentamen voor de datum van inschrijving ligt en bij de tweede de datum van het tentamen erna ligt.
    f. Wat zou er nu gebeuren als de datum van de inschrijving van een student op een cursus zou worden aangepast naar een datum die na het eerste tentamen van die cursist en die cursus zou liggen?
2. Er mogen alleen cijfers voor tentamens worden geregistreerd als die student voor dat vak geen vrijstelling heeft.
    a. Schrijf een query die controleert of er rijen zijn waarvoor de regel niet geldt.

b. Schrijf een UDF die, op basis van de student en de cursus, de code van de vrijstelling teruggeeft.

c. Herschrijf de query uit a) en maak nu gebruik van de udf.

d. Wijzig de tabel Tentamen en voeg daar een CHECK-constraint toe die gebruik maakt van die UDF.

e. Schrijf een positieve en negatieve testquery die de werking van de constraint test.

13 NORMALISEREN

13.1 OEFENINGEN

1. Gegeven onderstaande tabel. De primaire sleutel ligt over **StudentNr** en **CursusCode**. Verder zijn er de volgende aannames:
   - Een student doet dezelfde cursus maar één keer.
   - De naam van de student wordt gezien als één geheel.

Studentnr	Naam	Adres	Woonplaats	**Cursuscode**	Cursusnaam	Startdatum	Gebouw	Locatie
1234	Jan de Smet	Keistraat 30	Edam	63info	Informatica	1-9-2022	BBME	Rotterdam
5678	Willem Wever	Dorpstraat 1	Schiedam	62taal	Talen	1-9-2022	Atheneum	Den Haag
5678	Willem Wever	Dorpstraat 1	Schiedam	63info	Informatica	1-10-2022	Atheneum	Den Haag
9511	Willem Wever	Steenderens 12	Westervoort	63info	Informatica	1-11-2022	BBME	Rotterdam

   a. Wat zijn de afhankelijkheden?

   b. In welke normaalvorm staat deze tabel? Geef hierbij je motivatie en ook daarbij waarom een tabel niet voldoet aan de eerstvolgende normaalvorm. Dus als je vindt dat de tabel in 2NV staat, waarom vind je dat en waarom niet in 3NV. Stop in je motivatie ook een voorbeeld die dit onderschrijft.

2. Gegeven onderstaande tabel met kandidaatsleutel op **Titel**, **Jaar** en **Acteur**. Er zijn films met dezelfde titel die in een ander jaar worden gemaakt. Deze films kunnen dezelfde regisseur of acteurs hebben.

Titel	**Jaar**	Regisseur	**Acteur**
Grease	1978	Randal Kleiser	Olivia Newton-John
Grease	1978	Randal Kleiser	John Travolta
Metro	1997	Thomas Carter	Eddie Murphy
Metro	1997	Thomas Carter	Kim Kiyori
Metro	1997	Thomas Carter	Michael Rapaport

   a. Welke afhankelijkheden zijn er?

   b. Waarom staat deze tabel niet in 2NV?

   c. Zet de tabel om in twee tabellen in 2e normaalvorm. Geef de primary keys en eventuele foreign keys van die nieuwe tabellen.

3. Een theater registreert vóór elk seizoen welke voorstellingen er op welke dagen en tijdstippen gegeven gaan worden, samen met de vastgestelde toegangsprijzen. Deze gegevens staan in een tabel PROGRAMMA. Zie hier beneden voor een voorbeeld. De primary key ligt op attributen **Datum** en **Tijd**. Een voorstelling wordt uniek geïdentificeerd door het voorstellingnummer (Vnr). De kolommen Normaal en Abonnement geven de prijzen weer.

   PROGRAMMA:

Datum	**Tijd**	Vnr	Titel	Normaal	Abonnement
5-03-2008	20:15	24	Ballet Don Quichote	40	35
6-03-2008	14:00	25	Cowboy Billie Boem	15	12
6-03-2008	20:15	26	Lebbis en Jansen	25	20
7-03-2008	20:30	27	Aïda	50	45
8-03-2008	20:30	27	Aïda	50	45

a. Wat zijn de functionele afhankelijkheden?

b. Waarom staat deze tabel in de 2e normaalvorm?

c. Waarom staat deze tabel niet in de 3e normaalvorm?

d. Zet de tabel om in twee tabellen in 3e normaalvorm. Geef de primary keys en eventuele foreign keys van de nieuwe relaties.

4. Hieronder zie je de tabellen **Student** en **Studieresultaat** die onderdeel uitmaken van het studievolgsysteem van een ICT-opleiding. LET OP: Bij deze ICT-opleiding wordt elk vak afgesloten met precies één toets. Als een student een onvoldoende voor een toets, dan kan hij/zij die toets bij een volgende gelegenheid herkansen. De primary key van de tabel **Student** wordt gevormd door de kolom **studnr**.

Student

studnr	roepnaam	achternaam	startdatum	einddatum	vooropleiding
567422	Jip	Straatman	1-9-2016	31-8-2017	HAVO
567803	Britt	Dalen	1-9-2016	31-8-2017	VWO
570419	Rik	Janssen	1-9-2016		VMBO, MBO
586399	Mila	Verbeek	1-9-2017		HAVO
588197	Tom	Janssen	1-9-2017		HAVO, VWO
589477	Sophie	Ridder	1-9-2017		MBO

Studieresultaat

studnr	**vakcode**	vaknaam	**datum**	cijfer	dcode	dnaam
567422	Dtbs	Relationele Databases en SQL	5-12-2016	7,5	PTR	Potters
567803	Dtbs	Relationele Databases en SQL	5-12-2016	6,8	PTR	Potters
570419	Dtbs	Relationele Databases en SQL	5-12-2016	4,3	PTR	Potters
570419	Dtbs	Relationele Databases en SQL	12-3-2017	5,3	VRM	Vermeulen
586399	Dtbs	Relationele Databases en SQL	5-12-2016	5,1	PTR	Potters
586399	Dtbs	Relationele Databases en SQL	12-3-2017	6,4	VRM	Vermeulen
567422	Prgr1	Inleiding Programmeren	3-11-2017	6,9	KGM	Kaagman
567803	Prgr1	Inleiding Programmeren	3-11-2017	7,8	KGM	Kaagman
570419	Prgr1	Inleiding Programmeren	3-11-2017	5,6	KGM	Kaagman
586399	Prgr1	Inleiding Programmeren	3-11-2017	7,3	KGM	Kaagman
588197	Prgr1	Inleiding Programmeren	3-11-2017	2,9	VRM	Vermeulen
589477	Prgr1	Inleiding Programmeren	3-11-2017	8,4	VRM	Vermeulen

a. Staat de tabel **Student** in de 1e normaalvorm (1NV)? Motiveer je antwoord.
b. Zet de tabel **Student** indien nodig om naar tabellen in de 3e normaalvorm (3NV).
c. Breng voor de tabel Studieresultaat de functionele afhankelijkheden in kaart.
d. In welke normaalvorm staat de tabel Studieresultaat? Motiveer je antwoord.
e. Zet de tabel Studieresultaat indien nodig om naar tabellen in 3NV.

14 DIVERSE OEFENINGEN

14.1 MUZIEKDATABASE

1. Aan welke scholen zijn meer dan 2 componisten verbonden? Geef alle informatie over die scholen.
2. Welke componisten hebben de meeste klassieke stukken geschreven? Geef id, naam en geboortedatum.
3. Welke klassieke stukken hebben een onbekende bezetting? Geef stuknr en titel. Schrijf de query op twee manieren. Eén keer me een subquery, de andere met een join.
4. Wie is de oudste componist?
5. Wat is het jongste stuk?
6. Je oma van 91 wil graag piano leren spelen. Gezien haar leeftijd wil ze graag een stuk voor beginners met een zo kort mogelijke speelduur leren spelen. Ze vraagt aan jou welk stuk het meest geschikt is. Ze heeft een brede muzieksmaak, dus genre maakt niet uit. Geef stuknr en titel.

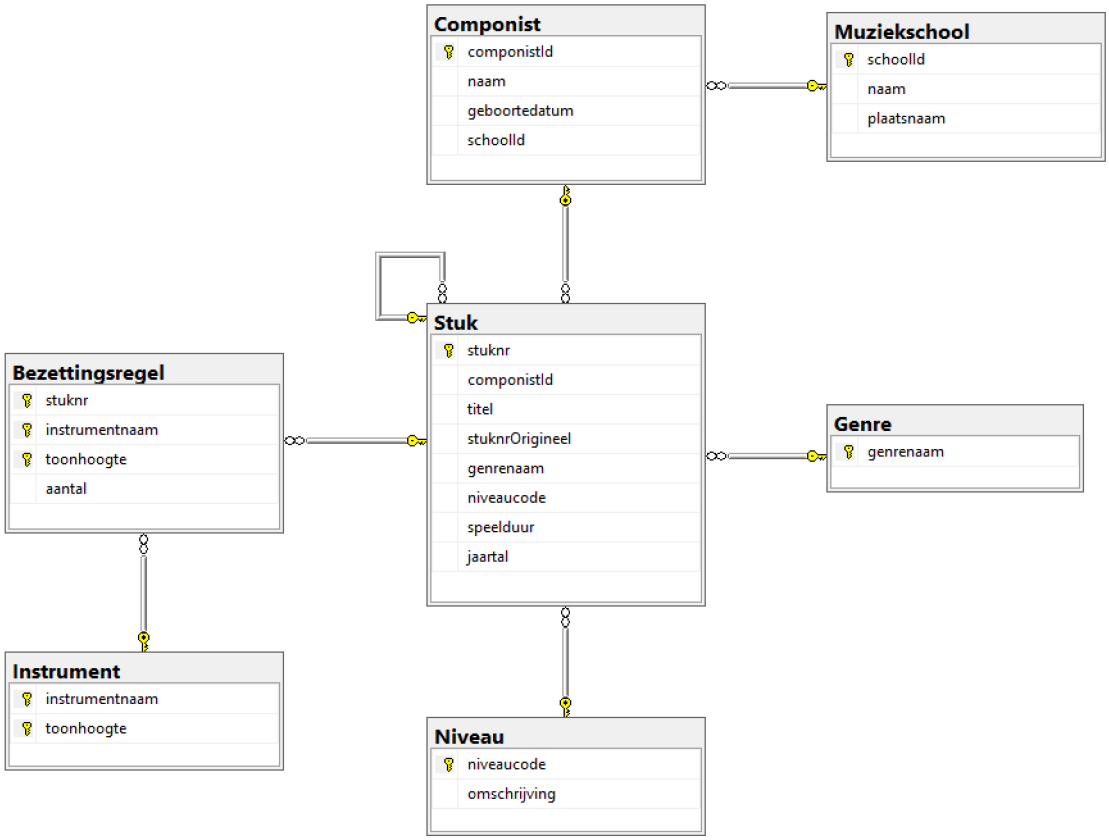
14.2 OPENSCHOOL DATABASE

1. Welke docent begeleidt geen enkele cursus? Geef twee oplossingen: een met een subquery en een met een outer join.
2. Welke docenten begeleiden 2 of meer cursussen? Geef twee oplossingen: een met een subquery en een met een join (maar zonder subquery).
3. Welke student heeft nog geen enkele cursus succesvol afgerond? Opmerking: Een cursus is succesvol afgerond als de student een voldoende heeft gehaald of een vrijstelling heeft gekregen.
4. Geef de code en de omschrijving van cursussen waarvoor 2 of meer vrijstellingen zijn verleend. Geef twee SELECT-statements, een met een join en een met een subquery.
5. Geef voor elke student het aantal behaalde credits.

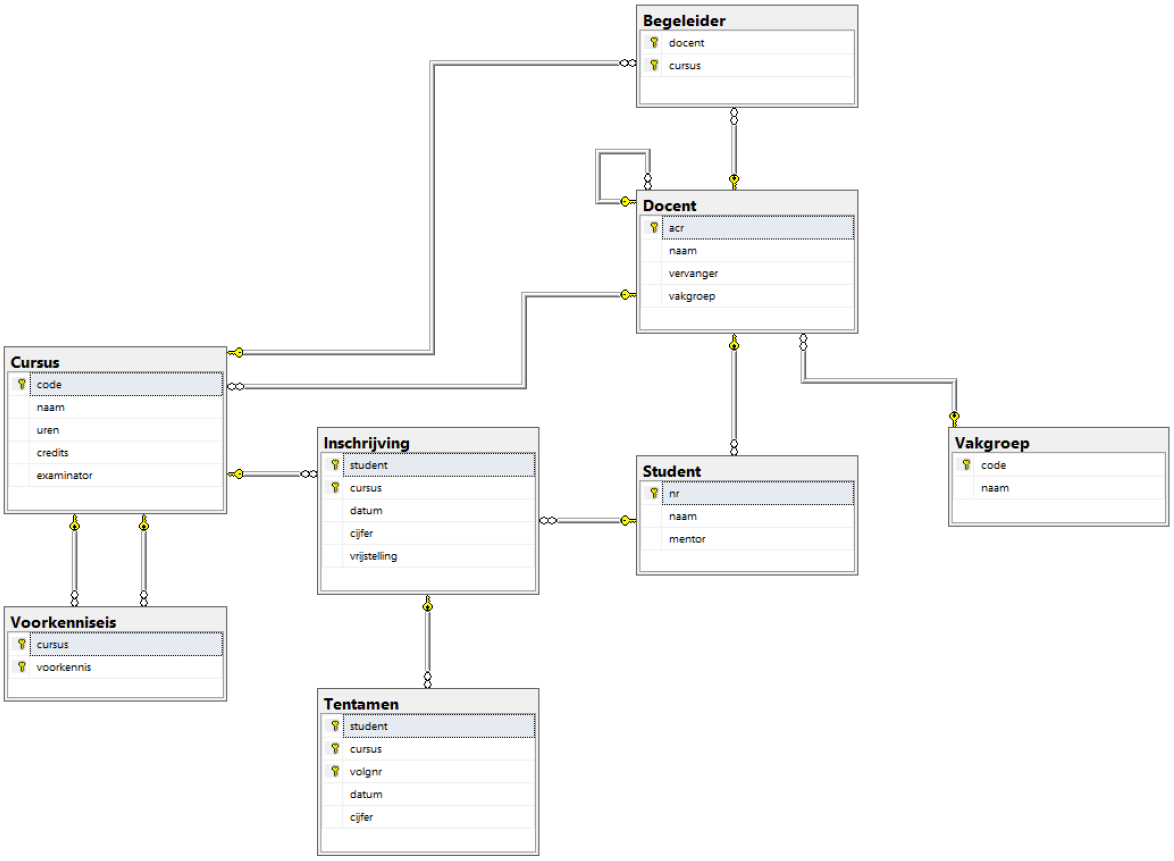
15 DATA MODELLEN

Hier vind je een uitleg van de databases die in het werkboek gebruikt worden. Per database vind je het schema en een korte beschrijving van de tabellen.

15.1 MUZIEKDATABASE



15.2 OPENSCHOOL



15.3 PREMIER DATABASE