

# Toets Parel 010 — Databases (201300070) Parels der Informatica

17 september 2015, 08:45-09:45

- Je mag 1 zelfgemaakt A4'tje met aantekeningen bij deze toets gebruiken. Rekenmachines, laptops, mobiele telefoons e.d. zijn niet toegestaan. **Stop deze nu in je tas!**
- Het aantal te behalen punten per opgave staat aangegeven.

1 *SQL (20 punten)* We gebruiken voor deze vraag de tabellen van een 'music' database. Een beschrijving van de tabelstructuur kun je vinden in Figuur 2. De database registreert liedjes die zijn gecomponeerd door artiesten en die uitgevoerd zijn door (mogelijkerwijs andere) artiesten op albums.

Om ook bijvoorbeeld covers, live-optredens en verschillende uitvoeringen van een klassiek stuk te kunnen registreren, is voor deze tabelstructuur gekozen met een aparte tabel "performed\_by". In Figuur 3 vind je voorbeelddata voor deze tabellen die dit principe illustreert. Kijk bijvoorbeeld naar artiest "Led Zeppelin" (artist\_id=1). Deze artiest heeft het liedje "Stairway to Heaven" gecomponeerd (song\_id=10) en die ook uitgevoerd op album "Led Zeppeling [Disc 2]" (album\_id=20). Maar het liedje staat ook op album "Top 2000 editie 2010" (album\_id=21). In de voorbeelddata kun je ook zien dat het liedje "Sing Our Own Song" van artiest "UB 40" gecoverd is door een onbekende artiest (artist\_id=NULL) op album "A Tribute to UB 40".

- Geef een SQL-query die voor artiest 'Led Zeppelin' de namen van alle liedjes geeft die zij gecomponeerd hebben.
- Geef een SQL-query die voor artiest 'Led Zeppelin' de namen van alle liedjes geeft die zij uitgevoerd hebben samen met de titel en het jaar van het album waarop de uitvoering staat.
- Geef een SQL-query die voor elk liedje de id en naam van dat liedje geeft samen met het aantal uitvoeringen van dat liedje.
- Geef een SQL-query die voor artiest 'UB 40' een overzicht geeft van alle albums (zowel id als title) waar uitvoeringen op staan van liedjes die zij gecomponeerd hebben, met ook het aantal uitvoeringen (van liedjes die zij gecomponeerd hebben).
- Men is bezorgd om datakwaliteit en vraagt zich af of er albums zijn waarop een liedje staat met een track\_nr dat hoger is dan de nr\_of\_tracks van het album. Geef een SQL-query die dergelijke schendingen oplevert met naam en id van het liedje, naam en id van het album en het track\_nr en de nr\_of\_tracks.
- De artiest die "Sing Our Own Song" van UB 40 heeft uitgevoerd op album "A Tribute to UB 40" is nog onbekend in onze database (artist\_id staat nog op NULL). Geef een SQL-statement of de SQL-statements die in de database registreren dat artiest 'The Pop Hit Crew' dit nummer uitgevoerd heeft. Verzin zelf unieke nummers voor eventuele id's die je nodig hebt. Je mag in de statements direct de juiste id's gebruiken (je hoeft geen subquery's te gebruiken).

□

☞ 2 Databases (10 punten)

- Stel je bent een applicatie aan het programmeren die SQL-query's stuurt naar een database server. Eén van die query's is zo langzaam dat de applicatie niet nuttig voor de beoogde toepassing gebruikt kan worden. Wie moet dit oplossen, maw welke *rol* is de aangewezen persoon om die query te versnellen? Geef uitleg bij je antwoord.
- Geef één belangrijke reden waarom je een *extern schema* zou definiëren. Geef uitleg bij je antwoord.
- Leg uit wat een *foreign key* is.
- Eén van de *ACID*-eigenschappen is *Consistency*. Wat doet een DBMS als het een transactie voorgeschoteld krijgt waarvan het resultaat de integriteitsregels schendt, dwz die de consistency van de data in gevaar brengt? Geef uitleg bij je antwoord.

□

☞ 3 Databaseontwerp (10 punten)

Figuur 1 bevat een klein stukje van een gegevensmodel van een registratie van externe stage- en afstudeeropdrachten. Extern betekent hier "bij een externe organisatie" zoals een bedrijf. De externe opdrachten hebben een titel, een beschrijving, een hoofdbegeleider, en ze zijn 'van' een bedrijf waarbij bedrijven worden geregistreerd met een naam en een adres. Van bedrijven registreren we ook een contactpersoon bij het bedrijf. Er zijn twee soorten externe opdrachten: stages (van verschillende duur) en afstudeeropdrachten. Die laatste hebben naast een hoofdbegeleider ook extra begeleiders. Van begeleiders en contactpersonen houden we een naam, e-mailadres en een telefoonnummer bij.

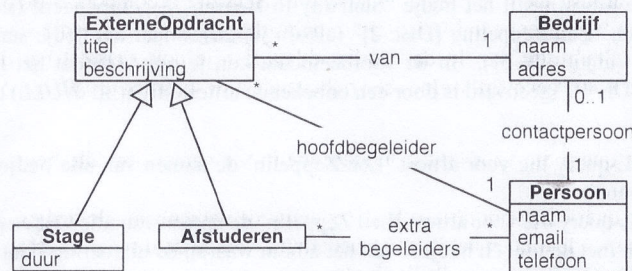


Figure 1: ER-model van een klein stukje van een gegevensmodel van een registratie van externe stage- en afstudeeropdrachten.

- Bij de relatie 'contactpersoon' staat "0..1". Dit betekent "0 of 1". Waarom is dat belangrijk? Waarom staat hier niet gewoon "1"? Leg je antwoord uit.
- Geef een tabelstructuur voor het ER-model van Figuur 1. Geef het antwoord in termen van een lijst tabellen met attributnamen. De types van de attributen mag je weglaten.

□

| Tabel        | Attributen / Beschrijving  |
|--------------|--|
| artist       | <b>artist_id</b> , name<br>Voor elke artiest een uniek nummer (artist_id; ook primary key) en de naam van de artiest.  |
| song         | <b>song_id</b> , name, composed_by<br>Voor elk liedje een uniek nummer (song_id; ook primary key), de naam van het liedje en wie het liedje gecomponeerd heeft (composed_by bevat een artist_id).  |
| album        | <b>album_id</b> , title, year, nr_of_tracks<br>Voor elk album een uniek nummer (album_id; ook primary key), de titel van het album, jaar van uitgave en het aantal tracks op het album.  |
| performed_by | <b>artist_id</b> , <b>song_id</b> , <b>album_id</b> , track_nr, duration.<br>Een liedje (song_id) wordt uitgevoerd door een artiest (artist_id), niet noodzakelijkerwijs dezelfde artiest als wie het gecomponeerd heeft. De uitvoering staat vervolgens op een album (album_id). Op deze manier kunnen er voor hetzelfde liedje meerdere uitvoeringen bestaan zelfs door verschillende artiesten (bijvoorbeeld covers van liedjes of uitvoeringen van verschillende orkesten van hetzelfde klassieke stuk). Naast de primary key (artist_id, song_id, album_id) wordt ook het nummer van de track (track_nr) en de duur van de uitvoering opgeslagen. |

Figure 2: Beschrijving van tabelstructuur van de "music"-database (Primary keys in **bold**).

| artist    |              | album    |                       |      |              |
|-----------|--------------|----------|-----------------------|------|--------------|
| artist_id | name         | album_id | title                 | year | nr_of_tracks |
| 1         | Led Zeppelin | 20       | Led Zeppelin [Disc 2] | 1990 | 15           |
| 2         | UB 40        | 21       | Top 2000 editie 2010  | 2010 | 2000         |
|           |              | 22       | A Tribute to UB 40    | 1993 | 14           |
|           |              | 23       | Rat in the Kitchen    | 1986 | 9            |

| song    |                    |             | performed_by |         |          |          |          |
|---------|--------------------|-------------|--------------|---------|----------|----------|----------|
| song_id | name               | composed_by | artist_id    | song_id | album_id | track_nr | duration |
| 10      | Stairway to Heaven | 1           | 1            | 10      | 20       | 15       | 8:00     |
| 11      | Sing Our Own Song  | 2           | 1            | 10      | 21       | 5        | 8:01     |
|         |                    |             | 2            | 11      | 23       | 9        | 7:21     |
|         |                    |             | 2            | 11      | 21       | 1957     | 4:04     |
|         |                    |             | NULL         | 11      | 22       | 8        | 2:54     |

Figure 3: Voorbeelddata voor de "music"-database

In the informal syntax, we use the following notations

- $A|B$  to indicate a choice between A and B
- $[A]$  to indicate that A is optional
- $A^*$  to indicate that A appears 0 or more times
- $A^+$  to indicate that A appears 1 or more times
- 'A' to indicate that the symbol A is literally that symbol

We are not precise in punctuation in the syntax, but this is irrelevant in this exam anyway.

**SQL**

```

createtable: CREATE TABLE tablename '(' columndef+ constraint* ')'
columndef: colname type [NOT NULL] [UNIQUE] [PRIMARY KEY] [REFERENCES tablename (colname+)]
constraint: PRIMARY KEY (colname, ... ) | CHECK ( condition )
           | FOREIGN KEY(colname, ... ) REFERENCES tablename(colname, ... )
query: SELECT ( column [ AS colname ] )+
       FROM ( tablename [ AS colname ] )+
       WHERE condition [ GROUP BY column + ] [ ORDER BY column + ]
column: [ tablename ':' ] colname | '*'
Examples of condition: column = value [ (OR | AND) [NOT] column <> value ]
                    | column IS [NOT] NULL
                    | column [NOT] IN (value, ... ) ...

```

Figure 4: Informal syntax of SQL