

Tentamen Databases

26 januari 2009

09:00 - 11:00, Educatorium-Alfa

- Scheur de antwoordvellen doormidden.
- Maak elke vraag op een ander vel.
- Vermeld op elk vel je naam en studentnummer. Indien één van deze zaken ontbreekt, wordt het vel niet nagekeken.
- Toon bij het inleveren je collegekaart.
- Schrijf en formuleer duidelijk.
- Bij elke vraag wordt verwacht dat je laat zien hoe je aan het antwoord komt (tenzij anders wordt vermeld).
- Je mag uitsluitend originele indexcards met aantekeningen raadplegen.
- Het tentamen duurt 2 uur en bestaat uit 5 opgaven.
- Gebruikte afko's:
 - 2PL = Two-phase locking
 - 2PC = Two-phase commit
 - BCNF = Boyce-Codd normaalvorm
 - COORD = Coordinator
 - CTP = Cooperative termination protocol
 - DP = Dependency preserving
 - FD = functional dependency
 - RA = Relatieve algebra
 - SQL = SQL (Structured Query Language)
- **Succes!**

Puntentelling:

- 1: 15
- 2: 20
- 3: 25
- 4: 20
- 5: 20

1 Queries

Een uitgeverij van een tijdschrift over console games houdt een database bij van de games die in het tijdschrift gerecenseerd worden. Een recensie betreft een spel (uniek identificeerbaar op basis van de naam) voor een bepaald platform (Xbox360, Playstation3, Wii) en is opgenomen in een uitgave van het tijdschrift (uniek identificeerbaar op basis van het jaar en de maand). Een spel krijgt bij een recensie een bepaalde beoordeling. Een spel kan meerdere keren gerecenseerd worden - al dan niet in dezelfde uitgave van het tijdschrift -, mits het telkens een ander platform betreft. Van elk spel wordt naast de naam het genre en de uitgever van het spel geregistreerd. Van elke uitgave van het tijdschrift wordt het jaar en de maand van uitgave bijgehouden, alsmede het aantal bladzijden en de oplage. Het databaseschema van de database is als volgt (primary keys zijn in cursief):

Uitgave (*jaar, maand*, bladzijden, oplage)
Recensie (*naam, platform*, jaar, maand, beoordeling)
Spel (*naam*, genre, uitgever)

Gegeven zijn de volgende queries:

- Q1: Geef het genre en de uitgever van de spellen die minstens eenmaal gerecenseerd zijn.
Q2: Geef de oplage van de uitgave in 2008 waarin de spellen de laagste gemiddelde beoordeling krijgen per uitgave in dat jaar.
Q3: Geef het genre en de uitgever van de spellen die nooit gerecenseerd zijn.

Hieronder volgen expressies in de RA of in SQL. Geef voor elke query aan welke expressie(s) met de query corresponderen. De relatie tussen queries en expressies is many-to-many en optioneel.

E1: $\pi_{genre, uitgever}((\pi_{naam}(Recensie) \cap \pi_{naam}(Spel)) \bowtie Spel)$

E2: $\pi_{genre, uitgever}((\pi_{naam}(Spel) - \pi_{naam}(Recensie)) \bowtie Spel)$

E3: $\pi_{genre, uitgever}((\pi_{naam}(Recensie) \cup \pi_{naam}(Spel)) \bowtie Spel)$

E4: $\pi_{genre, uitgever}((\pi_{naam}(Recensie) \div \pi_{naam}(Spel)) \bowtie Spel)$

E5: $\pi_{genre, uitgever}((\pi_{naam}(Recensie) - \pi_{naam}(Spel)) \bowtie Spel)$

E6: $\pi_{genre, uitgever}(Spel \bowtie Recensie)$

E7:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.jaar = 2008
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) >= ALL (
    SELECT AVG(Recensie.beoordeling)
    FROM Recensie, Uitgave
    WHERE Recensie.jaar = Uitgave.jaar
    AND Recensie.jaar = 2008
    GROUP BY Recensie.jaar)
```

E8:

```
SELECT Spel.genre, Spel.uitgever
FROM Spel
WHERE Spel.naam IN (
    SELECT Recensie.naam
    FROM Recensie)
```

E9:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.maand = Uitgave.maand
AND Recensie.jaar = 2008
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) <= ALL (
    SELECT AVG(Recensie.beoordeling)
    FROM Recensie, Uitgave
    WHERE Recensie.jaar = Uitgave.jaar
    AND Recensie.maand = Uitgave.maand
    AND Recensie.jaar = 2008
    GROUP BY Recensie.jaar, Recensie.maand)
```

E10:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.maand = Uitgave.maand
AND Recensie.jaar = 2008
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) < ALL (
    SELECT AVG(Recensie.beoordeling)
    FROM Recensie, Uitgave
    WHERE Recensie.jaar = Uitgave.jaar
    AND Recensie.maand = Uitgave.maand
    AND Recensie.jaar = 2008
    GROUP BY Recensie.jaar, Recensie.maand)
```

E11:

```
SELECT Spel.genre, Spel.uitgever
FROM Spel
WHERE Spel.naam NOT IN (
    SELECT Spel.naam
    FROM Spel, Recensie
    WHERE Spel.naam = Recensie.naam)
```

E12:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.maand = Uitgave.maand
AND Recensie.jaar = 2008
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) >= SOME (
    SELECT AVG(Recensie.beoordeling)
    FROM Recensie, Uitgave
    WHERE Recensie.jaar = Uitgave.jaar
    AND Recensie.maand = Uitgave.maand
    AND Recensie.jaar = 2008
    GROUP BY Recensie.jaar, Recensie.maand)
```

E13:

```
SELECT Spel.genre, Spel.uitgever
FROM Spel, Recensie
WHERE Spel.naam = Recensie.naam
```

E14:

```
SELECT Spel.genre, Spel.uitgever
FROM Spel
WHERE Spel.naam NOT IN (
    SELECT Recensie.naam
    FROM Recensie
    WHERE Recensie.naam = Spel.naam)
```

E15:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.maand = Uitgave.maand
AND Recensie.jaar = 2007
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) <= ALL
```

E16:

```
SELECT Uitgave.oplage
FROM Recensie, Uitgave
WHERE Recensie.jaar = Uitgave.jaar
AND Recensie.maand = Uitgave.maand
AND Recensie.jaar = 2008
GROUP BY Recensie.jaar, Recensie.maand, Uitgave.oplage
HAVING AVG(Recensie.beoordeling) <= ALL (
    SELECT AVG(Recensie.beoordeling)
    FROM Recensie, Uitgave
    WHERE Recensie.jaar = Uitgave.jaar
    AND Recensie.maand = Uitgave.maand
    GROUP BY Recensie.jaar, Recensie.maand)
```

E17:

```
SELECT Spel.genre, Spel.uitgever
FROM Spel
WHERE NOT EXISTS (
    SELECT Recensie.naam
    FROM Recensie
    WHERE Recensie.naam = Spel.naam)
```

2 Query optimization

We gaan uit van twee relaties R en S die één attribuut A delen. We definiëren een algebraïsche operator \blacktriangleright (anti-join) als volgt:

$$R \blacktriangleright S \equiv \{t \in R \mid \neg \exists u \in S : t.A = u.A\}$$

Met andere woorden: je houdt de tupels in R over die een natural join met S niet overleven.

- (i) Druk deze operator uit met behulp van andere algebraïsche operatoren.
- (ii) Beschrijf hoe de selectie distribueert over de anti-join. Dus herschrijf de expressie $\sigma_p(R \blacktriangleright S)$
- (iii) Geef een voorbeeld van een query in SQL waar het equivalent van een anti-join in voorkomt.

3 Normalisatie

- (i) We hebben een schema $R(ABCDEFGHK)$ met FD's $\{G \rightarrow H, C \rightarrow E, B \rightarrow CD, DE \rightarrow K, DH \rightarrow BF, K \rightarrow H\}$.

Geef twee candidate keys voor R die uit hoogstens drie attributen bestaan.

- (ii) We hebben een schema $R(ABCDEF)$ met FD's $\{A \rightarrow CD, A \rightarrow BE, B \rightarrow CD, AD \rightarrow F\}$.

Breng dit schema in BCNF via het BCNF-splitsalgoritme *met behoud van de DP eigenschap*. Geef voor elke stap aan via welke FD wordt gesplitst en wat na deze stap het nieuwe database-schema is. Licht toe waarom het uiteindelijke schema DP is. Indien de stappen niet duidelijk worden aangegeven of de toelichting op de DP eigenschap ontbreekt, worden voor deze opgave geen punten toegekend.

4 Concurrency

We beschouwen de volgende twee schedules.

- (i) Geef aan of deze schedules serialiseerbaar zijn of niet. Licht toe. Geef zo mogelijk de equivalente seriële schedules.
- (ii) Geef eveneens aan of de schedules getolereerd worden door een 2PL-scheduler. Geef hierbij een korte toelichting.

<i>S1</i>					<i>S2</i>				
T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
	w(y)				w(y)				
r(z)				w(x)				w(z)	
	w(y)						r(x)		r(x)
r(x)				w(y)					r(z)
		w(z)				r(z)			r(y)
			w(y)				w(y)		
				r(z)	r(z)				

5 Distributie

We veranderen het 2PC-protocol als volgt:

- De COORD stuurt een Commit eerst rond voordat hij deze in zijn log schrijft.
 - Het CTP (Cooperative Termination Protocol) blijft gehandhaafd.
- (i) Kan COORD na een crash nog autonoom een recovery uitvoeren, of zal hij (soms) de participants moeten raadplegen? Licht het antwoord toe.
 - (ii) Is nu, afgezien van bovengenoemde complicatie, het probleem van blocking bij het CTP opgelost? Licht het antwoord toe.

Einde