

DB1 Herkansingstentamen, 19 augustus 2008

Puntenverdeling

1	20
2	15
3a	10
3b	10
3c	10
3d	10
4	8
5	7
gratis	10
totaal	100

1. UML

Geef een UML class diagram aan de hand van de volgende beschrijving.

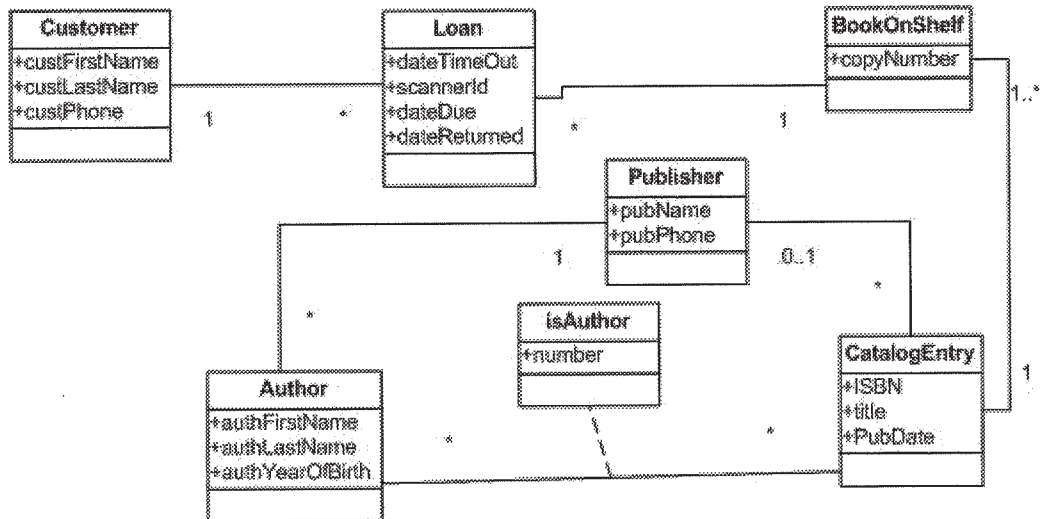
Een bedrijf bestaat uit een aantal afdelingen, elke afdeling heeft een naam en een locatie. Het bedrijf heeft een aantal werknemers, die elk onder een bepaalde afdeling vallen. Een werknemer heeft een naam (voornaam, achternaam), een uniek SOFI-nummer en een geboortedatum. Er zijn drie typen werknemers: staf-medewerkers, ondersteunend personeel en managers (elke werknemer behoort tot precies één van deze drie typen). Elke afdeling wordt geleid door een manager.

Binnen het bedrijf worden projecten uitgevoerd, elk project heeft een naam en een unieke projectcode en valt onder precies één afdeling. Een project wordt geleid door een staf-medewerker, verder is er een aantal staf-medewerkers aan het project verbonden. In sommige gevallen kan ondersteunend personeel ook aan een project verbonden zijn. Elke medewerker wordt geacht per week ongeveer het zelfde aantal uren aan een project te werken, dit aantal uren wordt vooraf vastgelegd. Vervolgens wordt tijdens het project elke week bepaald hoeveel uren de medewerker daadwerkelijk aan het project heeft besteed.

Voor elke stafmedewerker wordt een loopbaanplan opgesteld, om het verloop ervan te volgen wordt er per stafmedewerker een mentor (d.w.z. een andere medewerker) toegewezen die toeziet op de invulling ervan.

2. Het relationele model

Geef een relationeel model dat behoort bij het volgende UML class diagram:



3. SQL

Beschouw het volgende SQL datamodel.

```
create table customers (
  customer_id      INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  firstname VARCHAR (100),
  lastname  VARCHAR (100),
  phone     VARCHAR (20),
  email     VARCHAR (100),
  street    VARCHAR (100),
  housenr   VARCHAR (10),
  zipcode   VARCHAR (10),
  city      VARCHAR (100)
);
```

```
create table products (
  upc      BIGINT NOT NULL PRIMARY KEY,
  name     VARCHAR (100),
  price    VARCHAR (100),
  category VARCHAR (100),
  in_stock INT
);
```

```
create table orders (
  customer_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  order_date  DATE,
  sold_by     VARCHAR(100),
  PRIMARY KEY (customer_id, order_date),
);
```

```
FOREIGN KEY (customer_id) references customers (id)
);
```

```
create table order_lines (
  customer_id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  order_date DATE,
  upc BIGINT,
  quantity INT,
  PRIMARY KEY (customer_id, order_date, upc),
  FOREIGN KEY (customer_id, order_date)
    REFERENCES orders (customer_id, order_date),
  FOREIGN KEY (upc) REFERENCES products (upc)
);
```

- a.** Geef per customer (firstname, lastname) het aantal verkochte producten, zorg dat ook de klanten die nog niets hebben gekocht in de lijst voorkomen (met 0 verkochte producten uiteraard).
- b.** Geef de producten (upc en name) die de klant Arie Adelaar tussen 1 januari 2008 en 1 juli 2008 heeft gekocht, geef per product ook de bijbehorende verkoper (dwz. sold_by) en de datum van aanschaf.
- c.** Geef de producten die door klanten uit Rotterdam zijn aangeschaft.
- d.** Geef de producten die alleen maar door klanten uit Rotterdam zijn aangeschaft.

4. Concurrency

Geef de voor- en nadelen van resp. Optimistisch Concurrency Control en Pessimistic Concurrency Control.

5. JPA

Geef het voordeel aan van JPA (Java Persistence API) t.o.v. de "oude" JDBC aanpak.