UNIVERSITÄT DES SAARLANDES
Fachbereich 14 – Informatik
Lehrstuhl Prof. Weikum

Datenbanksysteme WS 1999/2000

Übung 4 — Abgabe: 22.11.1999

Aufgabe 1: Integritätsbedingungen

Gegeben sei das aus früheren Übungen bekannte Schema einer Universitätsdatenbank:

Professor	(P_Name, Fachbereichs_Nr, Gebäude, Raum, Tel)
Fachbereich	(Fachbereichs_Nr, Fachbereich_Name, Prodekan)
Gebäude	(Gebäude, Hausmeister)
Student	(Matrikel_Nr, S_Name, Semester, Fachbereichs_Nr)
Prüfung	(Matrikel_Nr, Fach, Prüfer, Note)

a) Für die Relation Prüfung seien die folgenden Fremdschlüsselbedingungen spezifiziert:

```
CREATE TABLE Prüfung (
...
FOREIGN KEY Matrikel_Nr REFERENCES Student (Matrikel_Nr)
        ON DELETE CASCADE
FOREIGN KEY Prüfer REFERENCES Professor (P_Name)
        ON DELETE SET NULL )
```

Simulieren Sie mit Hilfe geeigneter Trigger-Spezifikationen den Effekt dieser Fremdschlüsselbedingungen.

b) Spezifizieren Sie die folgenden (nicht notwendigerweise mit der realen Welt übereinstimmenden) Integritätsbedigungen mit Hilfe von Assertion- oder Trigger-Deklarationen:

i) Studenten dürfen die Prüfung im Fach Datenbanksysteme erst ab dem 5. Semester ablegen.

ii) Studenten, die bereits im 15. Semester sind, müssen mindestens eine Prüfung abgelegt und bestanden haben.

iii) Professoren sollen sich ihr Büro nicht mit Kollegen teilen müssen.

iv) Die Matrikel-Nr eines Studenten darf nie verändert werden.

v) Die Note eines Studenten darf nur verbessert, nicht aber verschlechtert werden.

Aufgabe 2: Integritätsbedingungen und Trigger

Eine Electronic-Commerce-Vertriebsfirma (z.B. für Bücher oder CDs) betreibe eine Datenbank mit den folgenden Relationenschemata:

Bestellungen	(BestellNr, ProduktNr, KundenNr, Anzahl, Bestelldatum);
Kunden	(KundenNr, Name, Adresse, E-Mail, Bestellimit);
Rechnungen	(BestellNr, KundenNr, Rechnungsdatum, Zahldatum);
Lager	(ProduktNr, Bez, Preis, Vorrat);
Nachbestellung	(ProduktNr, Anzahl, Bestelldatum);
Lieferung	(BestellNr, ProduktNr, KundenNr, Anzahl, Liefdatum);

Die Kunden der Firma haben die Möglichkeit, Produkte über das Internet zu bestellen. Sie bekommen dazu eine BestellNr (also einen elektronischen Warenkorb) zugewiesen, unter deren Angabe sie nacheinander verschiedene Produkte bestellen können. Bei jeder einzelnen Bestellung wird ein Tupel mit der zugewiesenen Bestellnummer, der Kundennummer, der Produktnummer, sowie der Anzahl der bestellten Produkte in die Relation Bestellungen eingefügt.

Innerhalb der Relation Bestellungen sind immer nur die Bestellungen eingetragen, die noch nicht ausgeliefert wurden oder vom Kunden noch nicht bezahlt wurden.

Innerhalb der Firma sind folgende Abläufe vorgeschrieben:

- Neue Bestellungen von Kunden werden zurückgewiesen, falls der Gesamtbetrag aller noch nicht bezahlten Bestellungen ein in der Relation Kunden eingetragenes Limit überschreitet.
- Bei jeder Bestellung durch den Kunden wird automatisch geprüft, ob das bestellte Produkt in ausreichender Anzahl im Lager enthalten ist. Ist das Produkt in ausreichender Anzahl im Lager vorhanden, so wird die bestellte Anzahl vom Lagerbestand abgezogen, und es wird eine Lieferung für die Bestellung in der Relation Lieferungen vorgesehen. Ist das Produkt nicht in ausreichender Form vorhanden, so wird eine Nachbestellung für das Produkt in der Relation Nachbestellung eingetragen. Existiert schon eine Nachbestellung (aufgrund anderer Bestellungen) für das Produkt, so wird die Anzahl einfach inkrementiert. Ansonsten wird die Nachbestellung (erstmals) eingefügt.
- Wenn eine Bestellung (ggf. ohne die nachbestellten Produkte) per Lkw den Hof des Lagers verläßt, soll der entsprechende Eintrag in der Relation Lieferungen gelöscht werden.
- Nachbestellungen für Produkte werden immer komplett an den jeweiligen Kunden geliefert. Sowie eine Nachbestellung bei der Firma eintrifft, wird die Nachbestellung aus der Datenbank ausgetragen und der aktuelle Lagerbestand muß entsprechend erhöht werden. Nach Lieferung einer Nachbestellung sollen für alle Bestellungen, die auf das gelieferte Produkt warten und jetzt vollständig ausgeliefert werden können, Lieferungen generiert werden.

- Sowie alle bestellten Produkte einer Bestellung an den Kunden ausgeliefert sind, soll automatisch eine Rechnung über den Gesamtbetrag der Bestellung an den Kunden geschickt werden und in der Relation Rechnungen registriert werden. Das Zahldatum wird initial auf NULL gesetzt.
- Sowie der Kunde bezahlt hat, trägt die Rechnungsabteilung das Zahldatum in Rechnungen ein. Dabei sollen alle bezahlten Bestellungen aus der Relation Bestellungen gelöscht werden.
- Manchmal macht die Rechnungsabteilung Fehler und markiert fälschlich Rechnungen als bezahlt. Deshalb soll das Zahldatum nur setzbar sein, wenn der Kunde tatsächlich alle Produkte der entsprechenden Bestellung empfangen hat.

Versuchen Sie, die beschriebenen Abläufe und Vorschriften möglichst vollständig durch Integritätsbedingungen und Trigger-Mechanismen sowie ggf. ECA-Regeln allgemeinerer Art zu realisieren.

Aufgabe 3: Views

Betrachten Sie das Beispielschema der Vorlesung mit den Relationen:

Kunden (KNr, Name, Stadt, Salde, Rabatt)
Produkte (PNr, Bez, Gewicht, Preis, Lagerort, Vorrat)
Bestellungen (BestNr, Monat, Tag, KNr, PNr, Menge, Summe, Status)

Nehmen Sie an, die Organisation des Unternehmens wird derart geändert, dass ein Produkt an verschiedenen Orten vorrätig gehalten werden kann. Bestellungen sollen grundsätzlich von dem Lager geliefert werden, das den größten Vorrat des entsprechenden Produkts hat.

Die Relation *Produkte* wird dazu in die zwei folgenden Relationen aufgespalten:

Prod (PNr, Bez, Gewicht, Preis)
Lager (PNr, Lagerort, Vorrat)

Wie können Sie erreichen, dass trotz dieser einschneidenden Änderung des Datenbankschemas und der Datenbank die für den Vertrieb essentielle Abfrage der Produktverfügbarkeit:

```
SELECT Vorrat FROM Produkte WHERE PNr = ...;
```

und das Programm zur Erfassung von Lieferungen (siehe S.69/70 Vorlesungsskript) unverändert weiterlaufen können.