Institut für Informatik
Prof. Dr. Oliver Vornberger
Dipl.-Math. Patrick Fox

Universität Osnabrück, 21.04.2009
http://www-lehre.inf.uos.de/~dbs
Abgabe bis 27.04.2009, 12:00 Uhr

Übungen zu Datenbanksysteme

Sommersemester 2009

Blatt 1

Übungsablauf

Es wird 10 Aufgabenblätter geben, die jeweils dienstags am Ende der Vorlesung ausgegeben werden. Da in diesem Jahr wegen der Vielzahl der Teilnehmer keine Testate stattfinden, sind die Aufgaben **schriftlich** in **Dreier-Teams** zu lösen. Die Lösungen sind bis zum Montag der Folgewoche vor 12 Uhr in die entsprechenden Kästen in der Mathematik (Gebäude 69, Erdgeschoss) einzuwerfen. Die Abgabe erfolgt getrennt nach Übungsleiter. Die korrigierten Lösungen werden in der selben Woche in den Übungen zurück gegeben und besprochen.

Zeichnungen sind grundsätzlich mit dem Rechner anzufertigen. Dazu und zum Lösen der Aufgaben steht Ihnen der CIP-Raum 31/145 zur Verfügung.

Aufgabe 1.1 (20 Punkte)

Erläutern Sie kurz die folgenden Begriffe und legen Sie anhand eines Beispiels dar, wie es dazu kommen und wie ein Datenbanksystem Abhilfe schaffen kann.

1. Datenredundanz
2. Dateninkonsistenz
3. Mehrbenutzerprobleme
4. Sicherheitsprobleme

Musterlösung vom 27.04.2009:

1. Datenredundanz bedeutet, daß dieselben Daten an zwei (oder mehr) verschiedenen Stellen unabhängig voneinander vorhanden sind. Beispiel: In einer Datei stehen alle Studenten mit ihren Daten drin, die jetzt Informatik B hören. In einer anderen Datei stehen alle Studenten mit ihren Daten drin, die jetzt DBS hören. Wenn ein Student beide LV besucht, dann sind seine Daten redundant.

2. Dateninkonsistenz bedeutet, daß aufgrund von Redundanz zwei (oder mehr) verschiedene 'Versionen' von Daten existieren, die eigentlich identisch sein sollten. Beispiel: In einer von zwei LV wurden die Daten des Studenten falsch erfaßt und in der anderen richtig.
   Inkonsistenz ist deutlich von Integrität zu trennen, denn darunter versteht man einerseits, daß die Attributwerte der Instanzen einer Entity bestimmten Regeln entsprechen und andererseits, daß Veränderungen an einer Instanz (insbesondere deren Löschung) aufgrund der Beziehungen im DBS Veränderungen an Instanzen anderer Entities (oder gar deren Löschung) hervorrufen.

3. Mehrbenutzerprobleme entstehen, wenn ein Benutzer beginnt, Daten zu editieren die schon von einem anderen Benutzer editiert werden, und zwar bevor der zeitlich erste Benuzter seine Änderungen gespeichert hat. Wenn beide Benutzer anschließend die Daten wieder in das DBS zurückschreiben, gehen die Änderungen desjenigen verloren, der als erster schreibt ('lost update'). Beispiel: Der Dozent einer LV erweitert das Vorlesungsskript um ein Kapitel, während der Übungsleiter einen Druckfehler beseitigt.

4. Sicherheitsprobleme enstehen dadurch daß mehrere Benutzer ein DBS benutzen und nicht jeder Benutzer Zugriff auf alle Daten haben soll. Dabei gibt es zwei Aspekte:

   - Die Art des Zugriffs (lesend/schreibend) soll für jeden Benutzer bzw. jede Benutzergruppe individuell regelbar sein.
   - Nicht alle Benutzer(gruppen) sollen lesenden Zugriff auf die Gesamtheit der gespeicherten Daten haben.

   Beispiel: Jeder Student darf bei einem Klausurergebnis die Matrikelnummern und die vergebenen Noten einsehen. Der Blick auf die zugehörigen Namen der Studenten bleibt aber verwehrt. Nur der Dozent der LV darf die Noten schreibend ändern.

Aufgabe 1.2 (10 Punkte)

Erklären Sie den Unterschied zwischen physischer und logischer Datenunabhängigkeit. In welchem Umfang werden beide in einem Datenbanksystem erreicht?

Musterlösung vom 27.04.2009:

Durch die Unterteilung des DBS in drei Ebenen (extern, logisch und intern) wird (mit Hilfe von klar definierten Schnittstellen) die Verbindung der einzelnen Teile gelockert.

1. Physische Datenunabhängigkeit: Die Art und Weise, auf die die Daten gespeichert werden (interne Ebene), kann verändert werden, ohne daß das Datenbankschema (logische Ebene) verändert werden muß. Da die logische Ebene (im Gegensatz zum konzeptuellen Schema) die Implementation berücksichtigt, kann die Einhaltung der Schnittstelle bei einer veränderten Speicherstruktur uneffizient sein. In den meisten heutigen DBS wird die physische Datenunabhägigkeit komplett erreicht.

2. Logische Datenunabhängigkeit: Veränderungen im Datenbankschema (logische Ebene) sollen möglich sein, ohne daß die darauf basierenden Anwendungen (externe Ebene) angepaßt werden müssen. Da jede Anwendung eine spezielle Sicht auf die Daten ermöglicht und da diese Sicht ein Ausschnitt aus der Gesamtsicht ist, ist es fast unmöglich, die Gesamtsicht (das DBSchema) ohne Konsequenzen für die Teilsichten zu ändern. Selbst kleine Änderungen der Entity- oder Relationship-Typen müssen unter Effizienzverlusten nach außen verborgen werden. Deshalb ist die logische Datenunabhängigkeit in heutigen Systemen kaum erreicht.

Aufgabe 1.3 (15 Punkte)

Erklären Sie die Begriffe

1. Entity und Entity-Typ,

2. Attribut und Attribut-Wert,

3. Relationship und Relationship-Typ

anhand von Beispielen.

Musterlösung vom 27.04.2009:

Ein **Entity-Typ** (z.B. Studenten) wird beschrieben durch einen Satz von Eigenschaften und Merkmalen, den sogenannten **Attributen** (z.B. Studienfach).
Jeder konkrete Gegenstand/jedes gedanklich existierende Konzept ist eine **Entity** (z.B. Erika Musterstudentin). Sie ist eine Instanz oder Ausprägung eines bestimmten Entity-Typen, wenn sie sich durch die Attribute des Typen charakterisieren läßt. Die Attribute einer Entity nehmen konkrete **Attribut-Werte** an (z.B. das Attribut 'Studienfach' den Wert 'Informatik'), deren Kombination für jede Entity einzigartig ist. Dadurch werden die einzelnen Entities wohlunterscheidbar. Die Entity-Typen heissen wie die Mehrzahl der Gegenstände, die sie zusammenfassen (z.B. Student Erika ist eine Instanz von Studenten).
Zwischen den einzelnen Entities verschiedener oder gleicher Entity-Typen bestehen **Relatonships** oder Beziehungen. Bisher kennen wir nur binäre Beziehungen. Diese (meist gerichteten) Relationships bestehen jeweils aus einer Entity von der die Beziehung ausgeht und einer Entity, bei der die Beziehung endet (z.B. Erkia Musterstudentin hört Datenbanken). Dazwischen steht mindestens ein Verb, das die Art der Beziehung ausdrückt.
Genauso, wie die Entity-Typen die Entities mit gleichen Merkmalen zu abstrakten Klassen zusammenfassen, fassen die **Relationship-Typen** alle Realtionships mit Anfangs-Entities des gleichen Typs und End-Entities des gleichen Typs, die von derselben Art sind (also mit demselben Verb dazwischen) zusammen. Die Art der Beziehung legt dann den Namen des Relationship-Typen fest - allerdings in der 3. Person Plural (z.B. Studenten hören Vorlesungen).

Aufgabe 1.4 (55 Punkte)

Erstellen Sie zu folgendem Anwendungsbereich ein möglichst komplettes relationales Datenmodell mit Charakterisierung der Beziehungen in (min, max)-Notation. Es sind nicht unbedingt alle Enity-Typen und Attribute aufgeführt. Sie sollten diese ergänzen, falls das für das Modell nötig und sinnvoll ist.

In einem Restaurant existieren 27 Tische, die je nach Besucherandrang so auf die 5 Tisch-Kellner verteilt werden, dass keiner von ihnen weniger als drei Tische, aber auch nicht mehr als sieben bedient. Außerdem gibt es drei Kellner, die nur für die Getränke zuständig sind. Jeder Mitarbeiter wird mit seiner Adresse (Straße, Hausnummer, PLZ, Ort) und einem Gehalt (bei Tischkellnern) bzw. Stundenlohn (bei Getränke-Kellnern) in einer Datenbank gespeichert.

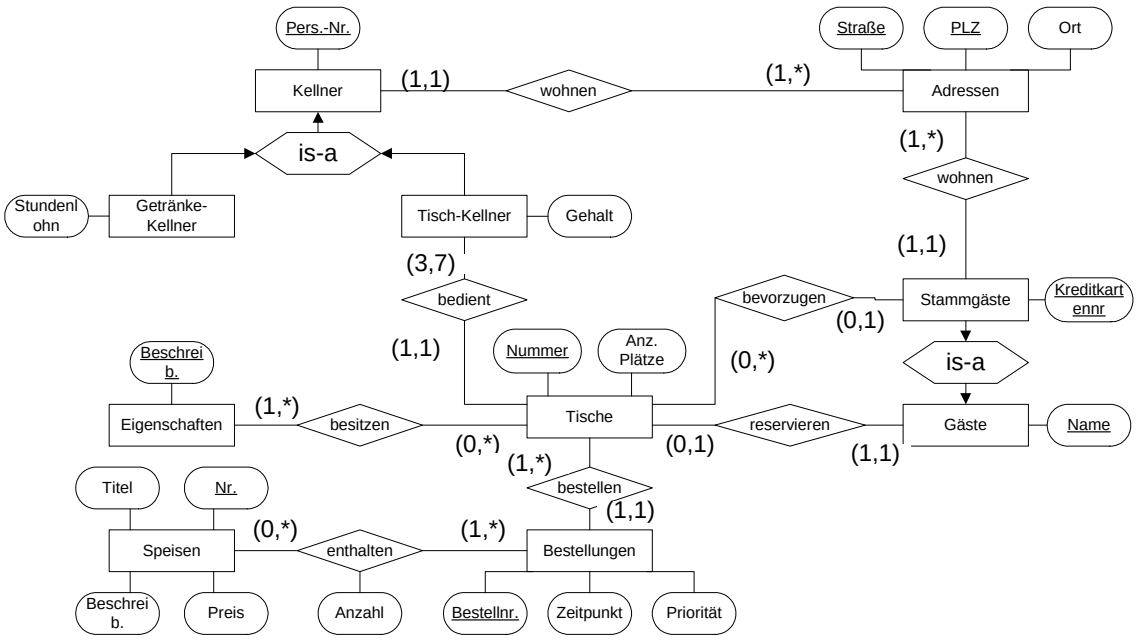
Für jeden Tisch wird eine Liste von Bestellungen erfasst. Jeder Tisch hat eine feste Anzahl Sitzplätze und eine Liste von Bemerkungen bzgl. seines Standortes (Fensterplatz, Nichtraucherbereich o.ä.). Jede Bestellung besteht aus der Nummer der Speise, dem Preis, dem Zeitpunkt der Bestellung und einer Dringlichkeitsstufe.

Alle Speisen und Getränke sind auf der Speisekarte mit einer Nummer versehen. Die Karte enthält außerdem je einen Titel und eine kurze Beschreibung zu jedem Eintrag.

Um die Reservierungen verwalten zu können, gibt es zusätzlich ein Liste von Stammgästen (Name, Adresse, Kreditkartennummer) mit je einem persönlichen Lieblingstisch.

Welche der hier aufgeführten verbal beschriebenen Tatsachen können Sie nicht in Ihr Modell aufnehmen? Begründen Sie Ihre Aussage!

Musterlösung vom 27.04.2009:



Folgende Tatsachen können in einem ER-Diagramm nicht modelliert werden:

- Man kann nicht modellieren, daß es 27 Tische gibt.
- Man kann nicht modellieren, daß es fünf Tischkellner gibt.
- Die Zuordnung zwischen Tischen und Kellnern läßt sich nur in der Form ”jeder Tisch wird - wenn er besetzt ist - von genau einem Kellner betreut” modellieren. Welcher Kellner genau welchen Tisch betreut (und insbesondere die Dynamik dieser Zuordnung im Laufe eines Abends) kann man im ER-Diagramm ebenfalls nicht modellieren.
- ”Laufkundschaft”, die ohne Reservierung an einem Tisch Platz nimmt und etwas bestellt, taucht nur über die Bestellung in der DB auf. Anders läßt sie sich nicht erfassen (vermutlich wäre ihr das auch nicht recht).

Alle vier Tatsachen sind Bestandteil einer Ausprägung und nicht Teil des Modells.