

Klausur Objektorientierte Datenbanken WS 2004/2005

Iwanowski 16.02.2005

Hinweise:

Bearbeitungszeit: 60 Minuten

Erlaubte Hilfsmittel: im Anhang, sonst keine

Diese Klausur besteht aus 9 Aufgaben (Seiten 2 bis 10) und einem Anhang (Seiten 11 und 12). (DI: zusätzlich noch eine Seite zu Datenbanken 1)

Bitte notieren Sie Ihre Antworten ausschließlich auf dem Aufgabenblatt! Bei Bedarf benutzen Sie die Rückseite! Für Skizzen und Entwürfe steht ebenfalls die Rückseite zur Verfügung. Entwürfe, die nicht gewertet werden sollen, sind durchzustreichen.

Hinter jeder Aufgabe steht die Anzahl der Bewertungseinheiten (BE), die für diese Aufgabe vergeben werden. Es gibt insgesamt 34 BE. Zum Bestehen benötigen Sie also 17 BE. Die Zeitangabe entspricht der empfohlenen Zeit, die Sie sich mit der Aufgabe beschäftigen sollten, um in der Bearbeitungszeit von 60 Minuten fertig zu werden. Sie ist nur als Richtwert zu sehen.

Viel Erfolg!

Aufgabe 1: Thema: Relationale und objektorientierte Realisierung eines ER-Modells (7 BE, 10 min)

- a) Formulieren Sie zum ER-Modell in Anhang I ein relationales Modell, indem Sie alle Tabellen und die zugehörigen Spalten benennen! Beachten Sie die üblichen Normalformen! Im Primärschlüssel soll auf jeden Fall, wenn möglich, Nr enthalten sein. Unterstreichen Sie für jede Tabelle den oder die Primärschlüssel!
(5 BE)
- b) Wenn Sie zusätzlich zum relationalen Datenmodell auch ein objektorientiertes erstellen (wird hier nicht verlangt), dann ist in einem der beiden Modelle Information verloren gegangen: Geben Sie an, in welchem Modell welche Information nicht explizit abgelegt ist!
(2 BE)

Lösung:

- a) Tabelle Kunde: Spalten Nr, Name, Alter
Tabelle Produkt: Spalten Nr, Name
Tabelle Verkäufer: Spalten Nr, Name, Alter
Tabelle Kauf: Spalten Kundennr, Produktnr, Verkäufern, Preis, Datum
- b) Im relationalen Modell ist die Information verloren gegangen, dass Kundennr in Kauf zu Nr in Kunde gehört, analog bei Produkt und Verkäufer.

Aufgabe 2: Thema: Persistenzkonzepte

(2 BE, 4 min)

Erklären Sie zwei verschiedene Varianten, wie Sie in ODMG für Java ein bisher im Hauptspeicher bereits vorhandenes transientes Objekt persistent machen können!

Lösung:

Erste Variante: Das transiente Objekt wird durch Methode `makePersistent` persistent gemacht.

Zweite Variante: Das Objekt wird von einem anderen Objekt, das persistent gemacht wurde, durch eine relationship oder ein Attribut erreicht.

Aufgabe 3: Thema: Transaktionen

(3 BE, 5 min)

Erklären Sie den Unterschied zwischen pessimistischen und optimistischen Transaktionen! Welchen Typ gibt es nur in JDO (und nicht in ODMG)?

Lösung:

Pessimistisch: Die Transaktion darf nur ausgeführt werden, wenn der Ausführende die alleinigen Rechte auf alle Ressourcen hat, die benutzt werden. Für alle anderen sind also alle Ressourcen, die von der Transaktion benutzt werden, gesperrt.

Optimistisch: Die Transaktion belegt keine Ressource mit einer Sperre. Kurz vor dem Commit wird nachgesehen, ob sich ein Wert durch eine andere Transaktion geändert hat. Nur, wenn das nicht der Fall ist, wird das Commit durchgeführt.

Optimistische Transaktionen gibt es nur in JDO.

Aufgabe 4: Thema: XML-Metadaten

(6 BE, 10 min)

- a) Warum ist die Verwendung einer XML-Metadatei in JDO notwendig? Geben Sie mindestens 2 verschiedene Gründe an!
(2 BE)
- b) Nennen Sie zwei verschiedene (orthogonale) Typen von Eigenschaften, die in den XML-Metadaten von JDO für die Attribute einer persistenzfähigen Klasse festgelegt werden können!
(4 BE)

Lösung:

- a) In der XML-Metadatei werden die persistenzfähigen Klassen sowie die Eigenschaften ihrer Attribute festgelegt.
- b) Persistenz der Attribute: persistent, transactional oder none.
Abhängigkeit der Attribute: first class (eigene Objekt-Id) oder second class (keine eigene Objekt-Id).

Aufgabe 5: Thema: Anfragesprachen

(2 BE, 3 min)

Jede objektorientierte Datenbank bietet die Möglichkeit, alle persistenten Daten über ihren Klassenextent in den Hauptspeicher zu holen. Welchen Vorteil hat die Benutzung einer Objektenfragesprache wie OQL oder JDOQL gegenüber der Extent-Lösung?
(2 BE, 3 min)

Lösung:

In der Extent-Lösung werden alle Objekte einer Klasse geladen. Das können sehr viele sein, während vielleicht nur eines von Interesse ist. Bei einer Objektenfragesprache werden nur die Objekte in den Hauptspeicher geladen, die Eigenschaften erfüllen, nach denen explizit gefragt wurde.

Aufgabe 6: Thema: Datenidentitätskonzepte

(2 BE, 3 min)

Angenommen, Sie haben eine relationale Modellierung von Daten (z.B. das Modell aus Aufgabe 1) bereits vorliegen. Sie wollen nun eine Modellierung für JDO vornehmen und haben sich für ein Datenidentitätskonzept zu entscheiden. Welches würden Sie bevorzugen, Datastore-Identity oder Application-Identity? Begründen Sie Ihre Antwort! (2 BE)

Lösung:

Für Datastore-Identity spricht, dass der Benutzer weniger Arbeit hat, da alles voreingestellt ist.

Für Application-Identity spricht, dass Primärschlüssel bereits vorliegen und so mit bedeutungsvolleren Begriffen Anfragen stellen kann.

Aufgabe 7: Thema: Lebenszykluszustände in JDO

(4 BE, 8 min)

- a) Unter welcher Bedingung kann man auf die Attribute von Objekten, die bisher im Zustand `hollow` waren, **nicht** im Java-Programm zugreifen? In welchen Zustand wechseln Objekte, auf deren Attribute man zugreifen kann, wenn ein solcher Zugriff erfolgt? (siehe Anhang II) (2 BE)
- b) Geben Sie einen obligatorischen und einen optionalen Zustand an, in denen sich ein Objekt befinden darf, wenn man außerhalb einer Transaktion darauf zugreifen will! (2 BE)

Lösung:

- a) Außerhalb von Transaktionen kann man nicht auf `hollow` Objekte zugreifen. Sie wechseln dort nämlich nicht automatisch in einen persistenten Zustand. Innerhalb von Transaktionen geht solch ein Zugriff, da ein `hollow` Objekt automatisch in den Zustand `persistent-clean` wechselt.
- b) `transient`, `persistent-nontransactional`

Aufgabe 8: Thema: Datenbanken in betriebswirtschaftlichen Anwendungen (4 BE, 7 min)

- a) Nennen Sie die unterschiedlichen Zielsetzungen von Online Transaction Processes (OLTPs) und Datawarehouselösungen! (2 BE)
- b) Welcher Prozess steht zwischen dem eigentlichen Datawarehouse und dem Benutzer? Welches durch ein graphisches Logo leicht zu veranschaulichende Konzept für diesen Prozess ist in jeder Datawarehouselösung enthalten? (2 BE)

Lösung:

- a) OLTPs sind für Anwendungen mit häufigen Aktualisierungen und seltenen Anfragen, während Datawarehouselösungen für Anwendungen mit seltenen Aktualisierungen und häufigen Anfragen geeignet sind.
- b) Zwischen dem eigentlichen Datawarehouse und dem Benutzer steht die Datenanalyse (OLAP). Dieser Prozess hat einen virtuellen Würfel, mit dem er bequem verschiedene Sichten auf die Daten herstellen kann.

Aufgabe 9: Thema: JDBC

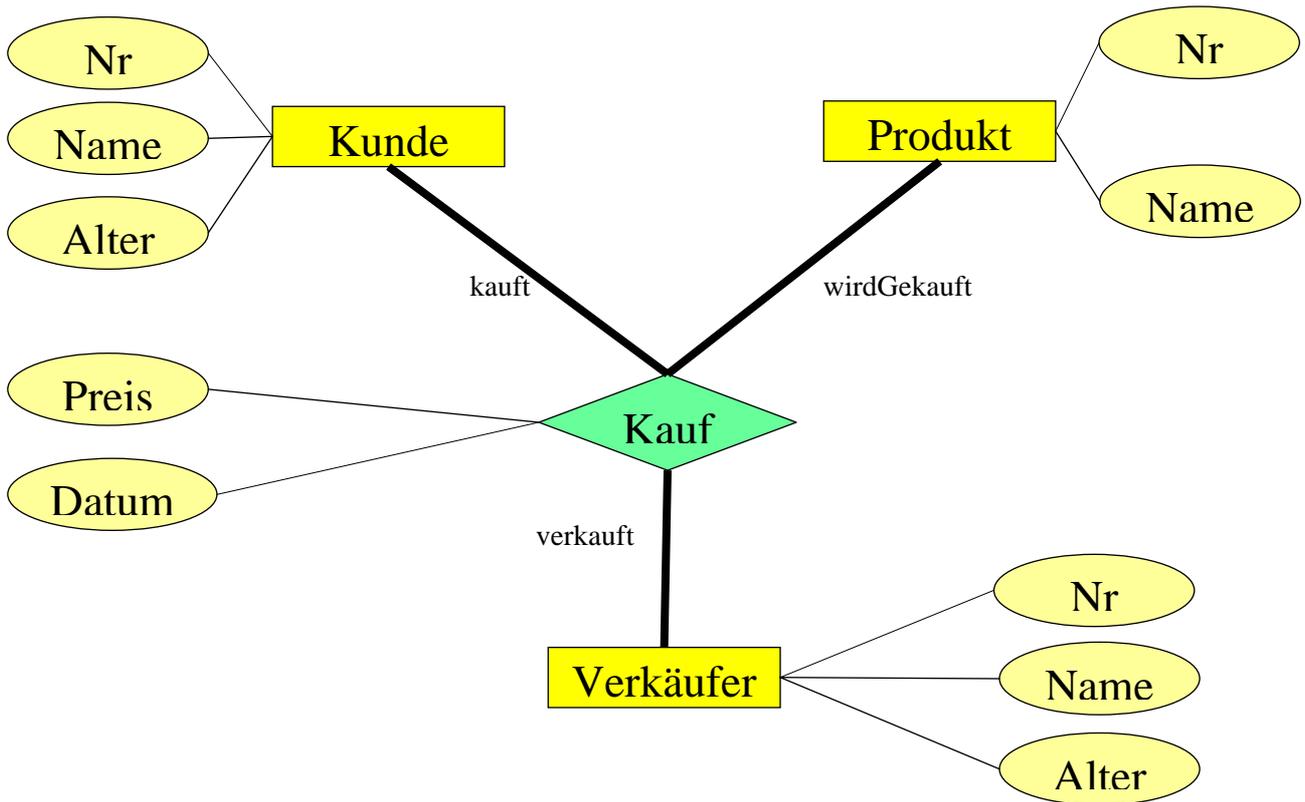
(4 BE, 10 min)

- a) Ihnen steht eine Java Virtual Machine (Standardversion) und eine beliebige Datenbank zur Verfügung. Geben Sie an, was Sie noch benötigen, um mit einem Javaprogramm die Datenbank ansprechen zu können! (Sie brauchen von den verschiedenen Möglichkeiten nur eine zu erwähnen) Geben Sie an, welche Ihrer Anforderungen plattformabhängig sind! (2 BE)
- b) Warum ist von einem Java-Programm aus eine objektorientierte Datenbank leichter zu bedienen als eine JDBC-Schnittstelle? (2 BE)

Lösung:

- a) Die häufigste Realisierung ist eine ODBC-JDBC-Bridge. Jede Datenbank hat einen plattformabhängigen ODBC-Treiber, jede JVM hat einen plattformunabhängigen JDBC-Treiber. Die ODBC-JDBC-Bridge selbst ist plattformabhängig.
- b) Die JDBC-Schnittstelle muss die Fragen in SQL formulieren, welche das dem objektorientierten Konzept sehr fern stehende relationale Modell kennen muss. Für die Bedienung einer objektorientierten Datenbank ist solch eine Konversion nicht nötig.

Anhang I



Anhang II

Obligatorische Lebenszykluszustände:

- **transient**
- **persistent-new**
- **hollow**
- **persistent-clean**
- **persistent-dirty**
- **persistent-deleted**
- **persistent-new-deleted**

Optionale Lebenszykluszustände:

- **transient-clean**
- **transient-dirty**
- **persistent-nontransactional**