

Aufgabenzettel 1

Abgabe: 8.10. bzw. 15.10.

[Hinweise für die Abgabe](#)

Aufgaben 1.1

Shop für T-Shirts

Dieser Shop soll nun neu entwickelt werden. Sie sind für den Datenbankteil verantwortlich. Welche Daten brauchen Sie, damit das System gebaut werden kann?

Stellen Sie eine vollständige Liste auf, was alles gespeichert werden muss. Erstellen Sie außerdem eine Liste, was gespeichert werden *kann*; also Daten, die nicht unbedingt zum Betrieb des Shops wichtig sind.

Erklären Sie kurz, warum Sie diese Daten für optional halten.

*Eine eindeutige Product-ID(**product_id**) ist immer als Primärschlüssel von Vorteil. Die Product-ID ist also eine eindeutige Kennung für jede „Variante“ unserer T-Shirts.*

Die verschiedenen Varianten eines T-Shirts haben:

- *Einen Preis (**price**)*
- *Unterschiedliche Größen (**size**)*
- *Unterschiedliche Farben (ggfs. auch Muster, Schnitt, Aufdruck) (**color**)*
- *Evtl. unterschiedliche Ausführungen, je nach Geschlecht*
- *Informationen über den Lagerbestand des Produkts*

*Hierbei ist nicht festgelegt ob alle Varianten unter einem Namen (**title**) von T-Shirts untergebracht sind oder ob das Produkt „Rotes T-Shirt“ ein Titel für sich, also ein eigenes Produkt ist. Auf jeden Fall unterscheidet sich die Product-ID.*

Optionale Daten sind:

- *Eine Beschreibung des Produkts*
- *Das Gewicht des Produkts*
- *Verkaufszahlen (z. B. um nach Topseller-T-Shirts zu suchen)*
- *Kundenbewertungen des Produkts*

Kunden

- *Kunden-ID*
- *Adresse (Straße/Hausnr. und PLZ/Ort, optional zweite Adresse)*
- *Kontakt (Telefonnr. / E-Mail)*
- *Bestellungen*

Bestellungen

- *Bestell-ID*
- *Kunden-ID (von Kundentabelle)*
- *Produkte (Array von Produkt-ID's)*
- *Status (Bestellt, Versand, Beahlt)*

Aufgabe 1.2

Suchen Sie im Internet nach DBMS und stellen Sie drei verschiedene DBMS in kurz vor. Was sind die wichtigsten Punkte, die aufgeführt werden?

Welche Typen an Datenbanksystemen gibt es generell? Finden Sie mindestens drei und arbeiten Sie kurz die Unterschiede heraus.

IMS/DB

Ist ein hierarchisches Datenbanksystem, welches mit dem IMS/TM (Transaktionsmonitor) das IMS (Information Management System) bildet.

- *IMS wurde von IBM entwickelt und kann auf IBM z Systemservern laufen.*
- *IMS wird bei hohen Transaktionsraten und hohen Systemverfügbarkeiten gerne verwendet.*
- *Es wird vorwiegend in Banken und Versicherungen eingesetzt.*

Hierarchisches Datenbanksystem

Dies ist das älteste Datenbanksystem.

Es bildet die reale Welt in einer hierarchischen Baumstruktur ab.

Jeder Satz (Record) hat genau einen Vorgängersatz, Ausnahme: (Ursatz:) Wurzel.

Die Daten werden in einer Reihe von Datensätzen gespeichert, mit denen verschiedene Felder verknüpft sind.

*Datenabbild ist eine Instanz eines bestimmten Datensatzes. (Ähnlich: Tabellen in relationalen Datenbanken)
Parent-Child Relationships : Verknüpfung zwischen Datenabbildern.*

Nachteil:

- *nur in Baumstruktur händelbar,*
- *Verknüpfungen zwischen Bäumen oder über mehreren Ebenen (Generationen) innerhalb eines Baumes nicht möglich.*

Ein Record-Typ ist die Urwurzel (Urvorfahre)

Jeder andere Record-Typ tritt genau einmal als "Child" auf.

Ein Record-Typ, der keine Nachkommen hat, wird Blatt genannt.

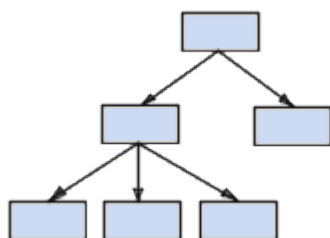
Es lassen sich also nur 1:n-Beziehungen darstellen.

Dateisysteme der Betriebssysteme sind annähernd hierarchisch. Die Verzeichnisse/Ordner sind die PCR und die Dateien/(auch .thumb) sind Blätter.

/ (root) ist der Urvorfahre.

Verwendung: Kleiner zweiter Frühling durch XML, Banken und Versicherungen.

Bild:



db4o (DataBase4Objects)

db4o ist eine Objektdatenbank für Java- und .NET-Plattform und gehört zu den NoSQL-Datenbanken(not only SQL: Datenbank ist nicht-relationell aufgebaut, kein festes Tabellenschemata). db4o wurde von Versant Corporation entwickelt und es existieren Unix, Linux und Windows-Programmversionen.

db4o hat einen sehr geringen Speicherbedarf (600kB) (ebenso geringer Arbeitsspeicherbedarf <1000kB) und eignet sich daher für die Einbettung in andere Programme oder auf Geräten mit beschränkten Speicherressourcen (bspw. PDAs).

- Keine Administration von usern. Einstellungen werden am Programmcode gemacht werden.
- Objekte können ohne weiteres gespeichert werden (keine Ableitung von Oberklassen oder Implementierung von Schnittstellen notwendig).
- Objekte haben beliebige Typen und beliebiger Tiefe.
- Persistenz mit wenig Aufwand
- API ist ähnlich zu Hibernate / JPA/EJB

ObjectContainer als Objektmanager: verweist auf eine Datei oder von einem Server bereitgestellte Datenbank.

Nachteil: Die Daten sind also nicht komplett unabhängig von der Klasse der Programmiersprache. Das Format ist proprietär und kann gegebenenfalls von nachfolgenden Programmiersprachen nicht mehr so einfach gelesen werden.

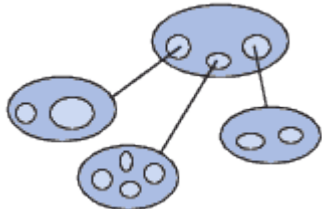
Vorteil: Diese kann sehr leicht zwischen Firmen ausgetauscht werden. Der Datenzugriff ist sehr schnell.

Der Objekttyp

ODBMS, Objekt-Datenbankmanagementsystem: die Daten werden in Objektform gespeichert, also in Strukturen, die Klassen genannt werden, innerhalb derer die Daten dargestellt werden. Felder sind Instanzen für diese Klassen. Zu Ende der 90er Jahre sind relationale Datenbanken die am meisten verbreiteten (ca. drei Viertel aller Datenbanken).

Ein objektorientiertes DBMS erlaubt es, zusammengehörende Einheiten auch zusammen abzuspeichern. Dabei lehnt es sich an die objektorientierte Programmierung an und vermeidet damit das wiederholte Zusammensuchen zusammengehörender Daten.

Bild:



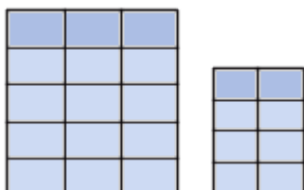
Apache Derby

Apache Derby ist eine Java-basiertes, relationales DBMS. Es ist in seiner Grundausstattung nur 2MB groß und ist sehr einfach zu installieren. Angewendet wird Derby oft in Java-Projekten. Seit Java 6 ist dieses Derby Bestandteil der JDK. Es ist plattformunabhängig und wird mit Java und SQL gesteuert. Zudem sind die erstellten Datenbanken mit allen Betriebssystemen kompatibel. Es benötigt keinen Administrator und lässt sich einfach installieren und integrieren.

Relationale Datenbank

Die Grundlage dieses Typs ist die mathematische Relation. Damit können flache Tabellen mit diesem Schlüssel-Schloss-Schema erstellt werden. Dabei ist jeder Zeile ein Datensatz (record) und die Spalten sind die beschreiben die Attribute. Als Schlüssel dient eine ID. Es kann mit Fremdschlüssel-Beziehungen beliebig Verknüpft werden. Als Werkzeug der Manipulation wird meist SQL verwendet. Es ist das heute am meist gebräuchliche Modell.

Bild:



Aufgabe 1.3

Nehmen Sie nun das Ergebnis von 1.1 und versuchen Sie die Daten zu strukturieren.
Überlegen Sie sich, wie Sie die Daten am besten darstellen bzw. bündeln.

Überlegen Sie sich außerdem, welche Abfragen man benötigt, um den T-Shirt-Shop zu betreiben.

Zwingend notwendig ist Product-ID (Primärschlüssel)

FIELD	TYPE	KEY	Extra
<i>Product_id</i>	<i>Int</i>	<i>PK</i>	<i>Auto_increment</i>
<i>Title</i>	<i>Text</i>		
<i>Price</i>	<i>float</i>		
<i>Color</i>	<i>Text (mit Auswahl-Optionen)</i>		
<i>Size</i>	<i>Text (mit Auswahl-Optionen)</i>		
<i>Sex</i>	<i>(Bool)</i>		
<i>Description</i>	<i>text</i>		
-----OPTIONAL-----			
<i>Weight</i>	<i>Float</i>		
<i>Stock</i>	<i>Int</i>		
<i>Items_sold</i>	<i>Int</i>		
<i>Picture</i>	<i>Object</i>		
<i>Reviews</i>	<i>Text</i>		
<i>Shipping_costs</i>	<i>float</i>		
<i>Import_date</i>	<i>Date</i>		
<i>....</i>			

BEWERTUNGS_TABELLE			
FIELD	TYPE	KEY	Extra
<i>Bewertungs-ID</i>	<i>Int</i>	<i>PK</i>	<i>Auto_increment</i>
<i>Kunden-ID</i>	<i>int</i>		
<i>Produkt-ID</i>	<i>int</i>		
<i>Sterne</i>	<i>int</i>		
<i>Text</i>	<i>text</i>		