

Nonstandard Datenbanken

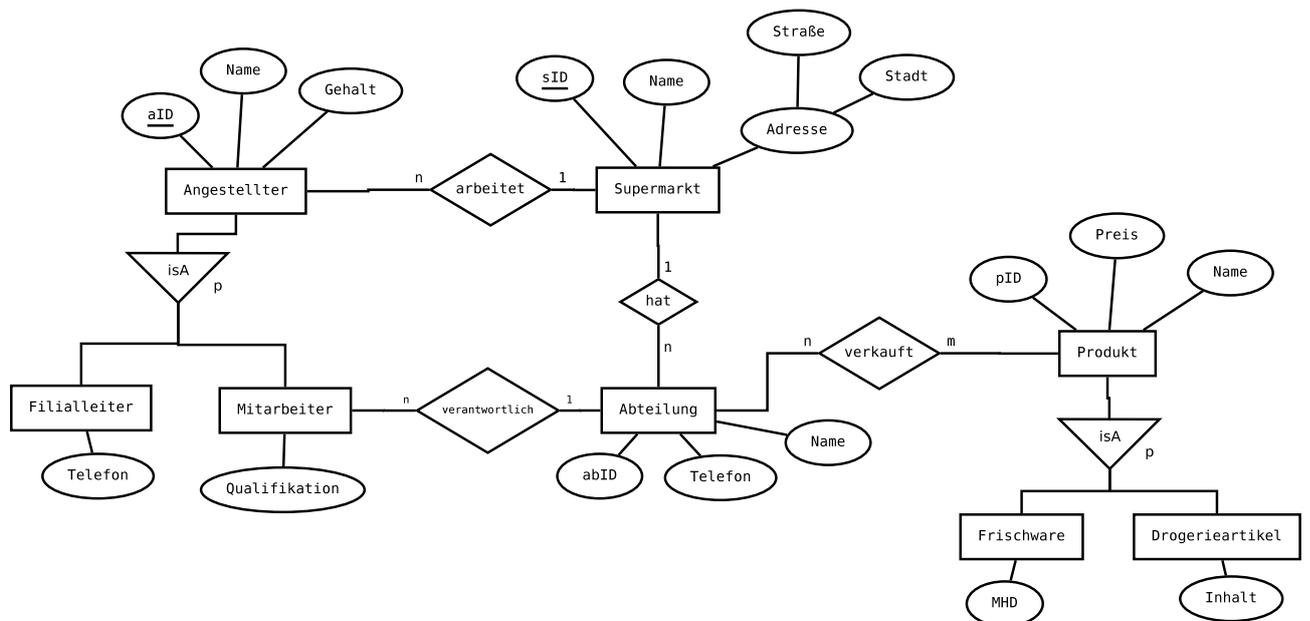
Wintersemester 2008/2009

13. Übungsblatt: Probe-Klausur

Hinweise:

- Bitte versuchen Sie, die Aufgaben zum Test in der in der Klausur vorgegebenen Zeit (90 Minuten) zu lösen.
- Die Lösungen dieses Aufgabenblattes werden gemeinschaftlich in der Übung am 13.02.2009 besprochen.

Aufgabe 1: Relationale und objektorientierte Modellierung

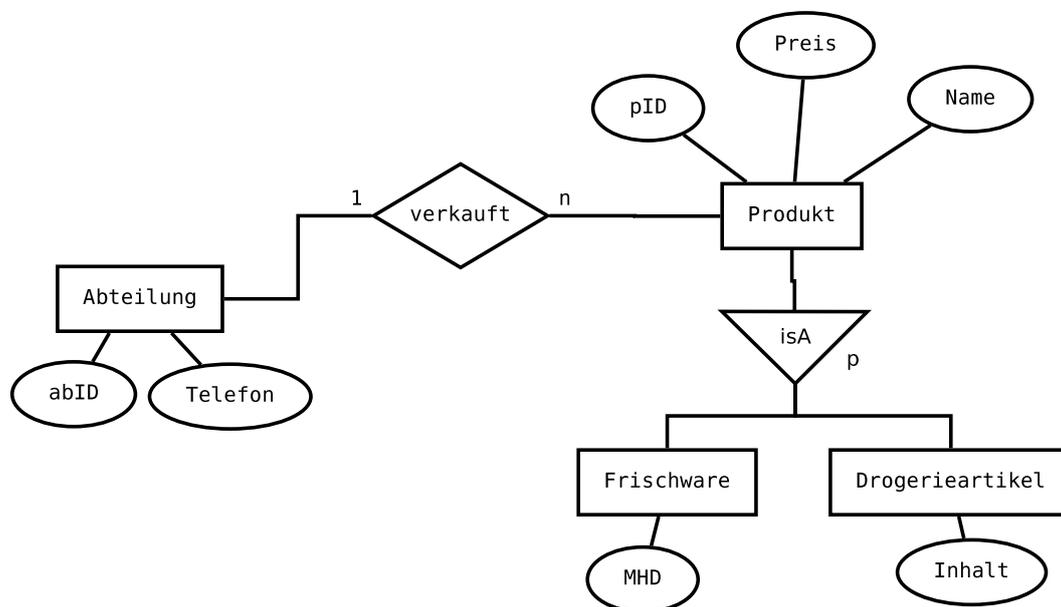


- a) Gegeben sei obiges ER Diagramm. Formulieren Sie folgende Anfragen in SQL und in OQL oder in der Syntax des *db4o*-DBMS.

- (i) Welche Mitarbeiter sind für die Fleischertheke verantwortlich? (4 Punkte)
- (ii) In welchen Abteilungen werden Frischwaren verkauft, deren MHD heute abläuft bzw. abgelaufen ist? (4 Punkte)
- b) Welche Anfragesprachen unterstützt das *db4o*-DBMS und welche sind typischer implementiert? (2 Punkte)
- c) Formulieren Sie folgende Anfrage in einer beliebigen Anfragesprache von *db4o*:
- Welches ist der teuerste Artikel, den der Supermarkt "Aldi Ratzeburger Allee" verkauft? (6 Punkte)

Aufgabe 2: Oracle

Im Folgenden soll nur ein Teilbereich des obigen ER Diagramms betrachtet werden.



- a) Modellieren Sie den dargestellten Sachverhalt objektrelational in der Oracle Syntax mit Hilfe von geeigneten Typen. Beachten Sie mögliche auftretende Referenzen und setzen Sie diese mittels eines Collection Types um. Erstellen Sie eine Tabelle für den Typ Abteilung und sichern Sie die Integritätsbedingung, dass abID eindeutig ist. Der Typ des MHD soll Varchar(8) sein, um eine Datumseingabe der Form 'TTMMJJJJ' zu ermöglichen. (8 Punkte)
- b) Erklären Sie kurz den Unterschied zwischen Functions, Procedures und Member Functions. (2 Punkte)
- c) Erstellen Sie eine Anfrage, die Ihnen alle Produkte (pID) liefert, die von der Abteilung abID=1 verkauft werden. (4 Punkte)
- d) Definieren Sie eine Funktion, die alle Frischwaren auflistet deren Haltbarkeitsdatum an einem per Parameter festgelegten Tag abläuft. (4 Punkte)

Aufgabe 4: XML Anfragesprachen: XPath

Gegeben sei die nachfolgende XML-Schema Definition. Zu diesem Schema liegt eine Datenbank in Form eines validierten XML Dokuments vor.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="address" type="xs:string"/>
  <xs:element name="auction">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bid" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="item"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="auctions">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="auction"/>
        <xs:element ref="closed_auction"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="rating" type="xs:string"/>
  <xs:element name="bid">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="bidder" type="xs:IDREF" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="closed_auction">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bid" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="item"/>
        <xs:element ref="price"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="item" type="xs:string"/>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="address"/>
        <xs:element ref="rating" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="pid" type="xs:ID" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    </xs:complexType>
</xs:element>
<xs:element name="persons">
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="person"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="price" type="xs:string"/>
<xs:element name="store">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="name"/>
      <xs:element ref="persons"/>
      <xs:element ref="auctions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Erstellen Sie unter Berücksichtigung der gegebenen Schema Datei XPath-Ausdrücke für die folgenden Anfragen:

- a) Ermitteln Sie die Namen der einzelnen Geschäfte. (3 Punkte)
- b) Lassen Sie alle Item-Elemente des Dokuments ausgeben, die 'Auto' enthalten. (3 Punkte)
- c) Welche Personen (Name) haben ein negatives Rating? (3 Punkte)
- d) Welche Personen (Name) wohnen in 'Lübeck'? (3 Punkte)
- e) In wie vielen Auktionen wurde oder wird 'Marzipan' versteigert? (3 Punkte)
- f) Ermitteln Sie die fünfte offene Auktion, in der 'Marzipan' versteigert wird. (3 Punkte)

Aufgabe 5: Abbildung von XML-Daten auf eine Relation

- a) Geben Sie zum XML Schema aus Aufgabe 4 ein geeignetes relationales Schema an. (6 Punkte)
- b) Wie geht man vor, wenn kein XML Schema bekannt ist?
 - (i) Wie sieht der generische Ansatz aus? Wo ist der Vorteil und wo ist der Nachteil des Verfahrens? (3 Punkte)
 - (ii) Nennen Sie ein allgemeines Beispiel für eine Anfrage, die zwar auf ein XML Dokument ausführbar ist, jedoch im erstellten relationalen Schema mittels SQL nicht mehr direkt darstellbar ist. Wo liegt das Problem? (3 Punkte)