

Datenbankprogrammierung (Oracle)

Aufgabe 1: Sequenzen

Erstellen Sie auf der Oracle-Datenbank eine Relation,

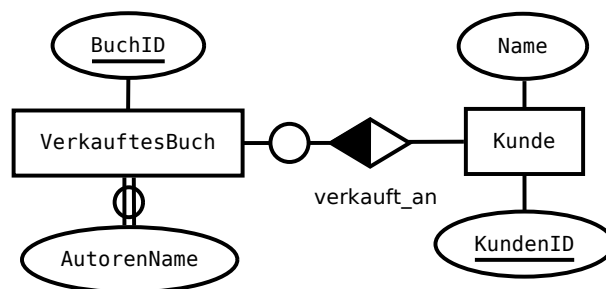
- die 5 Attribute unterschiedlichen Datentyps besitzt. Unter anderem soll der Datentyp `DATE` mit Default *'aktueller Tag'* verwendet werden.
- deren Primärschlüsselwert durch eine Sequenz erzeugt wird. Die Sequenz soll nur ungerade Zahlen generieren. Startwert 999, max. Wert 1003, zyklisch.
- Fügen Sie - wenn möglich - 4 Datensätze ein.
- Löschen Sie Sequenz und Relation.

Aufgabe 2: Identitätsspalten

Setzen Sie Aufgabe 3 mit Hilfe von Identitätsspalten um.

Aufgabe 3: Nested Tables

Für einen Online-Shop soll die Beziehung zwischen verkauften Büchern und Kunden modelliert werden. Folgendes ER-Modell ist gegeben:



Übersetzen Sie das ER-Modell in **Oracle konforme SQL-DDL**. Beachten Sie dabei die in der Vorlesung vorgestellten Konsistenzvorgaben und Integritätsbedingungen.

Hinweis: Das Attribut `AutorenName` ist ein *optionales mehrwertiges* Attribut (Name des 1. Autors, Name des 2. Autors, ...) und soll mit Hilfe von *Nested Tables* in der Datenbank realisiert werden.

Aufgabe 4: Objekte in Oracle (Freiwillige Zusatzaufgabe)

In der Vorlesung haben Sie *Nested Tables* für die Darstellung von mehrwertigen Attributen kennengelernt. Nested Tables sind beschränkt auf eine Spalte. Eine Erweiterung dieses Konzepts sind Objekte.

Unter anderem unterstützt Oracle das Ablegen von Objekten innerhalb von Relationen (objekt-relationale Daten). Um ein Objekt in einer Relation abzuspeichern, muss - analog zu Nested Tables - zunächst ein entsprechender Datentyp erzeugt werden:

```
CREATE TYPE meineKlasse AS OBJECT (  
    attribut01 INTEGER,  
    attribut02 VARCHAR2(20),  
    ... );
```

Analog zu Nested Tables gibt es für Objekte einen Konstruktor bestehend aus Typ-Name als Schlüsselwort und den Attributen als Parameter: `meineKlasse(12, 'attribut02-Wert', ...)`

Der so erstellte Typ kann in jeder Tabelle als Datentyp verwendet werden. Auch kann damit eine Objekt-Tabelle angelegt werden:

```
CREATE TABLE tabelle OF meineKlasse (PRIMARY KEY(attribut01));
```

Der *Table Constraint* gibt an, dass ein Attribut des Objekts als Primärschlüssel verwendet wird. Wird das Objekt nur als Spalte in einer Relation verwendet, so muss der Spaltenname ebenfalls angegeben werden:

```
PRIMARY KEY (<SPALTE>.<ATTRIBUT>).
```

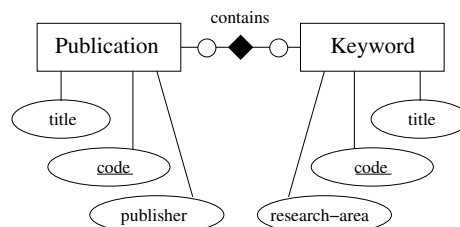
Um in einer anderen Tabelle einen Fremdschlüssel auf ein Element einer Objekt-Tabelle anzulegen, muss eine Spaltendefinition mit dem Schlüsselwort `REF` (eine Referenz) angegeben werden:

```
CREATE TABLE meineTabelle (  
    elem REF meineKlasse REFERENCES tabelle,  
    ... );
```

In diese Tabelle können Sie dann Referenzen auf Objekte aus `tabelle` einfügen. Eine Referenz erhalten Sie mit der Funktion `REF`. Die Funktion `DEREF` dient der Dereferenzierung. Ausgehend von einer Referenz erhalten Sie dadurch das entsprechende Objekt zurück.

```
SELECT REF(t), DEREF(REF(t)) FROM tabelle t WHERE ...
```

Gegeben sei nun das folgende ER-Diagramm:



- Erstellen Sie für beide Entitäten Objekt-Typen und Tabellen, die die Objekte aufnehmen können. Die Relationstabelle soll Referenzen auf Publikations- bzw. Keyword-Einträge enthalten.
- Fügen Sie einige Einträge in die erstellten Relationen ein.
- Geben Sie die Einträge aus `Publikation` mit den passenden `Keywords` aus. Attribute von Objekten werden in Oracle-SQL mit Punkt-Syntax ausgewählt.