

## Datenbanken Abgabe 2

### A) Artikel XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="TABLE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ARTIKEL" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ARTNR"
type="xs:int"></xs:element>
              <xs:element name="ARTBEZ"
type="xs:string"></xs:element>
              <xs:element name="MGE"
type="xs:string"></xs:element>
              <xs:element name="PREIS"
type="xs:double"></xs:element>
              <xs:element name="KUEHL"
type="xs:string"></xs:element>
              <xs:element name="EDAT"
type="xs:string"></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:attribute name="NAME" type="xs:string"></xs:attribute>
  </xs:element>
</xs:schema>
```

### B) Kurzbereich

#### Objekt Bestellung

Es wurde eine Klasse zur Repräsentation der Bestellungen mit Gettern und Settern + einer Validate Methode hinzugefügt

```
public class Bestellung {

    private int bstNR;
    private int kNR;
    private int status;
    private double rSUM = 0;
    private int mge;
    private List<Artikel> artList = new ArrayList<>();
    ...

    public boolean validate(){
```

```

        // alle not null Prüfen
        if(bstNR <= 0 || knr <= 0 || status == 0 || rSUM < 0 || mge == 0 ||
artList.size() < 1){
            return false;
        }

        // VALIDATE KUEHL = TK || KS || NK
        if (!(status > 0 && status <=5)){
            return false;
        }

        //VALIDATE PREIS decimal(7,2)
        double temp_d = rSUM*100;
        int temp_i;

        // Nachkommastellen Check (max 2)
        if(temp_d % 1 != 0){
            return false;
        }else{
            //checke stellen <= 7 vor ","
            if(!((int)temp_d < 100000000)){
                return false;
            }
        }

        return true;
    }
}

```

## Content Handler für Bestellungen

Einziger unterschied zu den anderen Handlern : die endElement Methode.

```

@Override
public void endElement(String uri, String localName, String qName) throws SAXException
{
    System.out.println("-E-> Ende des Elements: " + qName + " .");
    //System.out.println("aktwert: "+aktwert);
    try {

        switch (qName) {
            case "KNR":
                bestellung = new Bestellung();
                bestellung.setkNR(Integer.parseInt(aktwert));

```

```

        break;

    case "STATUS":
        bestellung.setStatus(Integer.parseInt(aktwert));
        break;

    // ab hier Artikel
    case "ARTNR":

        artikel = sql.select_Artikel_by_ARTNR(Integer.parseInt(aktwert));

        //WENN validierung an und Artikel OK hinzufügen
        if (val) {
            if (artikel.validateArtikel()) {
                System.out.println("-I-> Artikel Valide");
                bestellung.addArt(artikel);
            } else {
                throw new SAXException();
            }
        } else {
            bestellung.addArt(artikel);
        }
        break;

    case "BESTELLUNG":
        bestellungList.add(bestellung);
        break;

    }
} catch (Exception e){
    //Wenn oben irgendetwas nicht funktioniert (Parsen, checken etc)
    System.out.println("ERROR : " + e.getMessage());
    throw new SAXException();
}
}

```

### SQL Handler (insert Bestellung)

```

public boolean insert_bestellung(Bestellung b){
    boolean erg = false;

    int knr = b.getkNR();
    int status = b.getStatus();
    double rSUM = b.getrSUM();
    List<Artikel> ls = b.getArtList();

    //INSERT INTO BESTELLUNG2 (KNR,STATUS,RSUM,Lieferposition) VALUES (2,'3',4,
    Positionen(BestPos(1,'2','3',4,'5','17-05-2019',7)));
    //BestPos(artnr,'artbez','mge',preis,'kuehl','17-05-2019',anzahl)

    Artikel t;

```

```

int anz=0;
List<Artikel> ls_temp = new ArrayList<>();
List<String> ls_lief = new ArrayList<>();

Map<String, Integer> vergl = new HashMap<>();

for(Artikel a : ls){

    if(!vergl.containsKey(a.getArtbez())) {
        vergl.put(a.getArtbez(), 1);
        ls_temp.add(a);
    }else{
        int anzahl = vergl.get(a.getArtbez());
        anzahl ++;
        vergl.put(a.getArtbez(),anzahl);
    }

    //ls_lief.add("BestPos(" + a.getArtnr() + ", '" + a.getArtbez() + "', '" +
a.getMge()+"', "+a.getPreis()+", '" +a.getKuehl()+"', '" +a.getEdat()+"', "+ anz+""));
}

String datum;
String[] aendern;
for(Artikel a : ls_temp){
    aendern = a.getEdat().split(" ");
    aendern = aendern[0].split("-");
    datum = aendern[2] + "-" + aendern[1] + "-" + aendern[0];
    ls_lief.add("BestPos(" + a.getArtnr() + ", '" + a.getArtbez() + "', '" +
a.getMge()+"', "+a.getPreis()+", '" +a.getKuehl()+"', '" +datum+"', "+ vergl.get(a.getArtbez())
+""));
}

String BestPos = "";

boolean anfang = true;
for(String s:ls_lief){
    if(!anfang){
        BestPos += ",";
    }
    BestPos += s;
    anfang = false;
}

String positionen = "Positionen("+BestPos+")";

sql = "INSERT INTO BESTELLUNG2 (KNR,STATUS,RSUM,Lieferposition) VALUES " +
      "(" + knr + ", '" + status + "', " + rSUM + ", " + positionen + ")";

```

```

try {
    stmt = con.createStatement();
    stmt.executeUpdate(sql);
    erg = true;

} catch (SQLException e) {
    System.out.println(e.getMessage());
}
return erg;
}

```

## Ergänzung der Main

```

case 4:
    // test Bestellung insert XSD
    ContentHandlerVALBestellung bc_handler = new
ContentHandlerVALBestellung();
    MyErrorHandlerE be_handler = new MyErrorHandlerE();
    saxParserE_xsd.parseXmlFile("bestellung.xml",bc_handler,be_handler,true);
    break;

```

## C1) ADT zu 5a

```
BESTELLUNG := TUPLE OF (bstNR: int; kNR: int; status: int; artList: LIST OF (ak: ARTIKEL)
mit ARTIKEL := TUPLE OF (artnr: int; artbez: String; mge: String; preis: double; kuehl:
String; edat: date)
```

## C2) SQL Skript für Bestellungen

```
CREATE TYPE BestPos AS OBJECT (
    artnr Integer,
    artbez VARCHAR2(20),
    mge varchar2(20),
    preis decimal(10,2),
    kuehl VARCHAR2(20),
    edat Date ,
    anzahl Integer
)
```

```
CREATE TYPE Positionen AS TABLE OF BestPos
```

```
CREATE TABLE BESTELLUNG2(
    BSTNR Integer,
    KNR Integer,
    STATUS Varchar2(20),
    RSUM decimal (10,2),
    lieferposition Positionen
) NESTED TABLE lieferposition STORE AS pos_tab
```

