

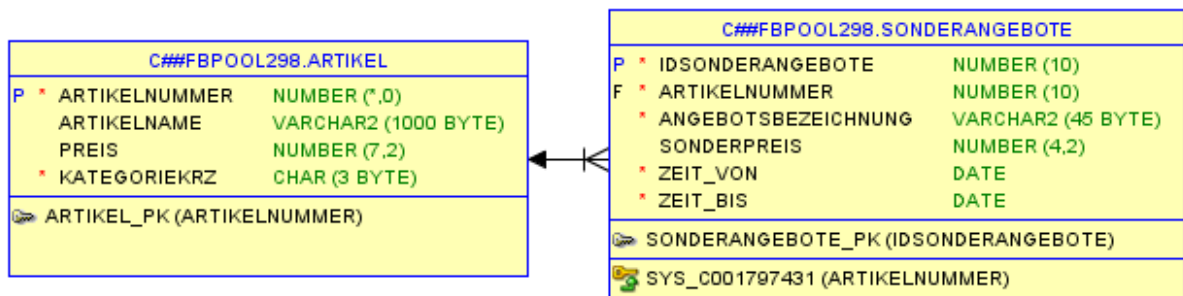
**Hinweis:**

Installieren Sie die Datenbank.

## Aufgabe 1 – Erweiterung der Datenbank

Die Datenbank soll um die Möglichkeit erweitert werden zeitlich begrenzte Angebote hinzuzufügen.

- a) Erweitern Sie das relationale Modell der Datenbank, um auch Sonderangebote erfassen zu können



- b) Ergänzen Sie das Schema der Datenbank, um auch Sonderangebote erfassen zu können.

DROP TABLE sonderangebote CASCADE CONSTRAINTS;

```
CREATE TABLE sonderangebote (
  idSonderangebote    NUMBER(10) PRIMARY KEY,
  artikelnummer       NUMBER(10) NOT NULL,
  angebotsbezeichnung VARCHAR(45) NOT NULL,
  sonderpreis         NUMERIC(4,2) CHECK (Sonderpreis >=0),
  zeit_von            DATE NOT NULL,
  zeit_bis            DATE NOT NULL,
  FOREIGN KEY (artikelnummer) REFERENCES Artikel(artikelnummer)
);
```

- c) Fügen Sie ein Sonderangebot für den Artikel „Bordeaux“ als Monatsangebot in der Datenbank zu einem 10% günstigeren Preis ein.

```
INSERT INTO Sonderangebote (idSonderangebote, artikelnummer,
  Angebotsbezeichnung, Sonderpreis, zeit_von, Zeit_bis)
VALUES (1, 4, 'Monatsangebot', 4.50, '01.10.2017', '31.10.2017');
```

- d) Frau Herta Müller ist von diesem Sonderangebot so begeistert, dass sie diesen Angebotsartikel gleich heute zehnmal bestellt und morgen geliefert bekommen möchte.

## Datenbanken 1

## Praktikum 3

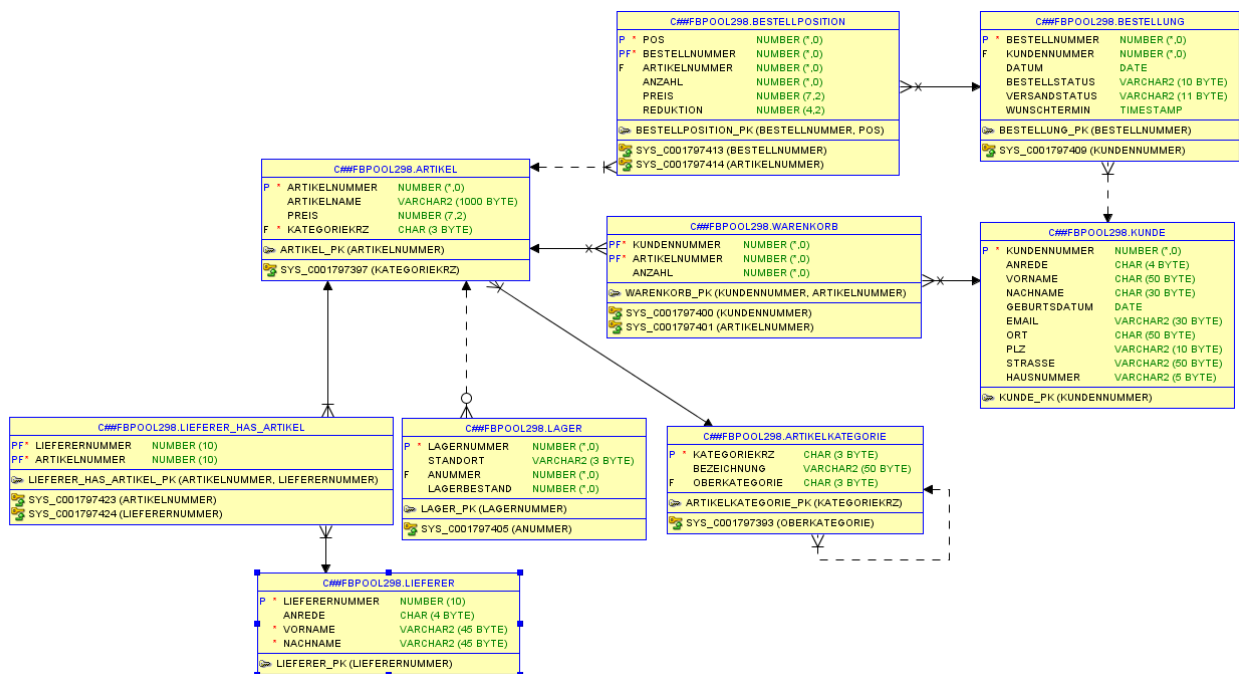
```
INSERT INTO Bestellung (bestellnummer, kundennummer, bestelldatum, bestellstatus,
versandstatus)
VALUES (62, 1, '17.10.2017', 'bezahlt', 'vorbereitet');
INSERT INTO Bestellposition (Position, bestellnummer, artikelnummer, anzahl, preis,
reduktion)
VALUES (119, 62, 4, 10, 5.0, 0.5);
```

Ergänzen Sie das Installationsskript entsprechend und überprüfen Sie die korrekte Ausführung.

### Aufgabe 2 – Erweiterung der Datenbank

Kunden sollen gekaufte Artikel und Lieferer zu einer getätigten Bestellung bewerten können.

- a) Erweitern Sie bitte das relationale Modell der Datenbank.



- b) Ergänzen Sie das Schema der Datenbank, um Bewertungen erfassen zu können.

```
DROP TABLE artikel_bewertung CASCADE CONSTRAINTS;
CREATE TABLE artikel_bewertung (
    artikel_Bewertung NUMBER(10) PRIMARY KEY,
    kundennummer NUMBER(10) NOT NULL,
    artikelnummer NUMBER(10) NOT NULL,
    bewertung INT CHECK (bewertung BETWEEN 0 AND 5) NOT NULL,
    bewertungText VARCHAR(45) NOT NULL,
    FOREIGN KEY (artikelnummer) REFERENCES Artikel (artikelnummer),
    FOREIGN KEY (kundennummer) REFERENCES Kunde (kundennummer)
);
```

```
DROP TABLE lieferer_bewertung CASCADE CONSTRAINTS;  
CREATE TABLE lieferer_bewertung (  
  lieferer_Bewertung NUMBER(10),  
  lieferernummer      NUMBER(10) NOT NULL,  
  bestellnummer       NUMBER(10) NOT NULL,  
  bewertung           INT CHECK (bewertung BETWEEN 0 AND 5) NOT NULL,  
  bewertungText       VARCHAR(45) NOT NULL,  
  PRIMARY KEY (lieferer_Bewertung),  
  FOREIGN KEY (bestellnummer) REFERENCES Bestellung(bestellnummer),  
  FOREIGN KEY (lieferernummer) REFERENCES Lieferer(lieferernummer)  
);
```

- c) Frau Herta Müller ist sehr angetan von dem Artikel und bewertet diesen entsprechend. Der Lieferer Eric hatte die Lieferung zwar direkt bestätigt. Gekommen ist die Lieferung aber erst einen Tag später. Frau Müller ist daher ziemlich sauer und gibt eine entsprechende Bewertung zu dem Lieferer ab.

```
INSERT INTO artikel_bewertung  
  VALUES (1, 1, 4, 5, 'Super!');  
INSERT INTO lieferer_bewertung  
  VALUES (1, 3, 62, 1, 'Unzuverlässig! Kam erst ein Tag später');
```

Ergänzen Sie das Installationsskript entsprechend und überprüfen Sie die korrekte Ausführung.