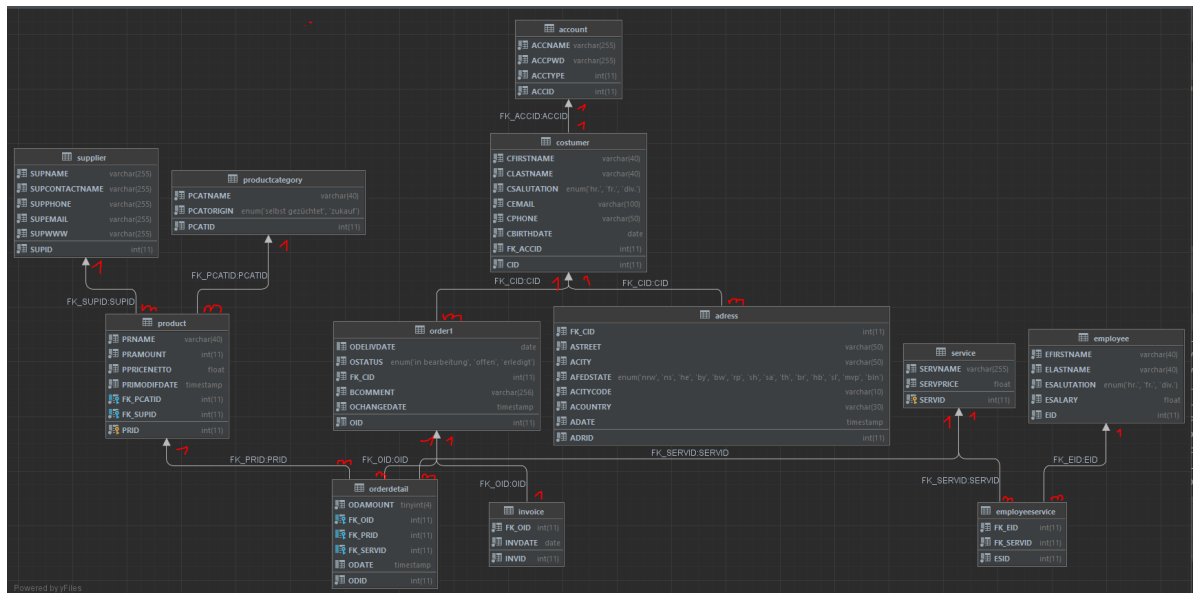


Zweites DBA Praktikum

Aufgabe 1

Erstellen Sie eine relationale Datenbank mit mindestens 12 Tabellen und füllen Sie Ihre Tabellen mit Datensätzen (mind. 10 Datensätze in Elterntabellen und mind. fünf Datensätze pro Kindtabelle)!



```
1 drop database floristik;
2 create database floristik;
3 use floristik;
4 # 1. account
5 create table account
6 (
7     ACCID int auto_increment primary key,
8     ACCNAME varchar(255) not null,
9     ACCPWD varchar(255) not null,
10    ACCTYPE enum ('Nobody', 'User', 'Admin') not null
11 );
12 # 2. customer
13 create table customer(
14     CID int auto_increment primary key,
15     CFIRSTNAME varchar(40) not null,
16     CLASTNAME varchar(40) not null,
17     CSALUTATION enum ('Hr.', 'Fr.', 'Div.') not null,
18     CEMAIL varchar(100) not null,
19     CPHONE varchar(50) not null,
20     CBIRTHDATE date not null,
21     FK_ACCID int not null,
22     foreign key (FK_ACCID) references account(ACCID)
23 );
24
25 # 3. address
26 create table address
27 (
```

```

28     ADRID      int                auto_increment primary key,
29     FK_CID     int                not null,
30     ASTREET    varchar(50)        not null,
31     ACITY      varchar(50)        not null,
32     AFEDSTATE  enum ('NRW', 'NS', 'HE', 'BY', 'BW', 'RP', 'SH', 'SA', 'TH',
    'BR', 'HB', 'SL', 'MVP', 'BLN') not null,
33     ACITYCODE  varchar(10)        not null,
34     AOUNTRY    varchar(30)        not null,
35     ADATE      timestamp default current_timestamp() not null,
36     foreign key (FK_CID) references customer(cid)
37 );
38 # 4. order
39 CREATE TABLE order1 (
40     OID int not null auto_increment primary key ,
41     ODELIVDATE date ,
42     OSTATUS enum('In Bearbeitung', 'Offen', 'Erledigt'),
43     FK_CID int,
44     BCOMMENT varchar(256),
45     OCHANGEDATE timestamp,
46     foreign key (FK_CID) references customer(CID)
47 );
48
49 # 5. productcategory
50 create table productcategory
51 (
52     PCATID      int      auto_increment primary key,
53     PCATNAME    varchar(40)        not null,
54     PCATORIGIN  enum('Selbst gezüchtet', 'Zukauf')
55
56 );
57 # 6. Lieferant
58
59 create table supplier
60 (
61     SUPID      int      auto_increment primary key,
62     SUPNAME    VARCHAR(255)        not null,
63     SUPCONTACTNAME  VARCHAR(255)        not null,
64     SUPPHONE   VARCHAR(255)        not null,
65     SUPEMAIL   VARCHAR(255)        not null,
66     SUPWWW     VARCHAR(255)        not null
67 );
68 # 7. product
69 create table product
70 (
71     PRID      int      auto_increment primary key,
72     PRNAME    varchar(40)        not null,
73     PRAMOUNT  int                not null,
74     PPRICENETTO  FLOAT          not null,
75     PRIMODIFDATE    timestamp default current_timestamp() not null,
76     FK_PCATID  int                not null,
77     FK_SUPID   int                not null,
78     foreign key (FK_PCATID) references productcategory(PCATID),
79     foreign key (FK_SUPID) references supplier(SUPID)
80
81 );

```

```

82
83
84
85 # 8. service
86 create table SERVICE
87 (
88     SERVID int auto_increment primary key,
89     SERVNAME VARCHAR(255) not null,
90     SERVPRICE FLOAT not null
91 );
92 );
93
94 # 9. orderdetails
95 create table orderdetail
96 (
97     ODID int auto_increment primary key,
98     ODAMOUNT tinyint not null,
99     ORDTYPE enum ('PRODUCT', 'SERVICE') not null,
100     FK_OID int not null,
101     FK_PRID int ,
102     FK_SERVID int ,
103     ODATE timestamp default current_timestamp() not null,
104     foreign key (FK_OID) references order1(OID),
105     foreign key (FK_PRID) references product(prid),
106     foreign key (FK_SERVID) references service(SERVID)
107 );
108
109
110 # 10. employee
111 create table employee(
112     EID int auto_increment primary key,
113     EFIRSTNAME varchar(40) not null,
114     ELASTNAME varchar(40) not null,
115     ESALUTATION enum ('Hr.', 'Fr.', 'Div.') not null,
116     ESALARY FLOAT not null
117 );
118
119 # 11. Mitarbeiterservice
120 create table employeeservice
121 (
122     ESID int auto_increment primary key,
123     FK_EID int ,
124     FK_SERVID int ,
125     foreign key (FK_EID) references employee(eid),
126     foreign key (FK_SERVID) references service(SERVID)
127 );
128
129 #12. Rechnung
130 create table invoice
131 (
132     INVID int auto_increment primary key,
133     FK_OID int not null,
134     INVDATE date ,
135     foreign key (FK_OID) references order1(oid)
136 );

```

```

137
138 insert into account(ACCNAME,ACCPWD,ACCTYPE) values
139     ('kunde1', 'kunde1pwd', 2),
140     ('kunde2', 'kunde2pwd', 2),
141     ('kunde3', 'kunde3pwd', 2),
142     ('kunde4', 'kunde4pwd', 2),
143     ('kunde5', 'kunde5pwd', 2),
144     ('kunde6', 'kunde6pwd', 2),
145     ('kunde7', 'kunde7pwd', 2),
146     ('kunde8', 'kunde8pwd', 2),
147     ('kunde9', 'kunde9pwd', 2),
148     ('kunde10', 'kunde10pwd', 2),
149     ('admin1', 'admin1pwd', 3)
150 ;
151 insert into customer(CFIRSTNAME,CLASTNAME,CSALUTATION
,CEMAIL,CPHONE,CBIRTHDATE,FK_ACCID) values
152     ('Mila',      'Müller',   'Fr.', 'milamuel@icloud.com',
0521654654, '1971-12-08',1),
153     ('Emilia',    'Schmidt',  'Fr.', 'emiliaschmidt@verizon.net',
0521545245, '1972-08-13',2),
154     ('Leni',      'Schneider','Fr.', 'lenischneider@verizon.net',
052156464,  '1975-08-25',3),
155     ('Mira',      'Fischer',   'Fr.', 'mirafischer@aol.com',
0521685745, '1975-09-01',4),
156     ('Sophia',    'Weber',     'Fr.', 'sophiaweber@icloud.com',
0521654821, '1982-08-06',5),
157     ('Adam',      'Meyer',     'Hr.', 'adammeyer@msn.com',
0521654826, '1986-08-17',6),
158     ('Noah',      'Wagner',    'Hr.', 'noahwagner@icloud.com',
0521729287, '1987-07-28',7),
159     ('Felix',     'Becker',    'Hr.', 'felixbecker@outlook.com',
0521569654, '1993-09-21',8),
160     ('Leon',     'Schulz',    'Hr.', 'leonschulz@gmail.com',
0521875412, '1996-09-14',9),
161     ('Emil',      'Hoffmann',  'Hr.',
'emilhoffmann@sbcglobal.net',0521632548, '2006-10-04',10)
162 ;
163
164 insert into adress(FK_CID,ASTREET,ACITY,AFEDSTATE,ACITYCODE,ACOUNTRY)values
165     (1, 'Hauptstraße 1', 'Hamburg', 'NRW', '05996', 'Deutschland' ),
166     (2, 'Schulstraße 2', 'Dresden', 'NS', '60137', 'Deutschland'),
167     (3, 'Gartenstr. 3', 'München', 'HE', '74405', 'Deutschland'),
168     (4, 'Bahnhofstr. 4', 'Lübeck', 'BY', '09610', 'Deutschland'),
169     (5, 'Dorfstraße 5', 'Freiburg', 'RP', '88672', 'Deutschland'),
170     (6, 'Bergstraße 6', 'Berlin', 'SH', '36301', 'Deutschland'),
171     (7, 'Birkenweg 7', 'Potsdam', 'SA', '70427', 'Deutschland'),
172     (8, 'Lindenstr. 8', 'Trier', 'TH', '74057', 'Deutschland'),
173     (9, 'Kirchstraße 9', 'Lage', 'BR', '55899', 'Deutschland'),
174     (1, 'Waldstr. 1', 'Herford', 'HB', '61893', 'Deutschland');
175
176 insert into order1(odelivdate, ostatus, fk_cid, bcomment ) value
177     ('2022-06-30','Offen',1,'Bitte morgens liefern!'),
178     ('2022-07-04','offen',1,''),
179     ('2022-09-14','Erledigt',3,''),
180     ('2022-10-11','Erledigt',4,''),

```

```

181      ('2022-11-29','In Bearbeitung',5,''),
182      ('2022-11-30','Erledigt',6,''),
183      ('2022-11-30','In Bearbeitung',6,'');
184
185
186
187 insert into productcategory(pcatname, pcatorigin) values
188
189      ('Schnittblume', 'Zukauf'),
190      ('Topfpflanzen', 'Selbst gezüchtet'),
191      ('Beetpflanzen', 'Selbst gezüchtet'),
192      ('Balkonpflanzen', 'Zukauf'),
193      ('Büsche', 'Zukauf')
194      ;
195
196 insert into supplier(supname, supcontactname, supphone, supemail, supwww)
197 values
198      ('Blumen Posch',                                'Elisa Posch', 065464138,
199      'bestellung@blumen-posch.de',                    'www.blumen-posch.de'),
200      ('Flower Gmbh Blumenhandel',                    'Ayla Bauer', 068565456,
201      'info@flower-blumenhandel.de',                    'www.flower-blumenhandel.de'),
202      ('DB Blumeneinzelhandel GmbH & Co. KG',          'Lea Koch', 058646545,
203      'order@db-blumeneinzelhandels.de',                'www.db-blumeneinzelhandels.de'),
204      ('Sperling Blumen Import',                        'Erik Schmitt', 065879634,
205      'bestellung@sperling.de',                          'www.sperling-blumen.de'),
206      ('Pflanzen-Discount',                              'Alex Braun', 078465345,
207      'info@pflanzen-discount.de',                      'www.pflanzen-discount.de')
208      ;
209
210
211
212
213
214
215
216
217
218
219
220 insert into product(prname, pramount, ppricenetto, fk_pcatid, fk_supid)
221 values
222      ('Gänseblümchen',                                99, 0.99,1,1),
223      ('Rosen',                                          10, 5.00,1,1),
224      ('Zamioculcas',                                   20, 2.00,2,2),
225      ('Bogenhanf',                                     5, 3.50,2,2),
226      ('Mexikanische Sonnenblume',                     10, 9.20,3,3),
227      ('Löwenohr',                                      10, 5.99,3,3),
228      ('Geranie ',                                     5, 6.60,4,4),
229      ('Petunie ',                                     10, 1.99,4,4),
230      ('Schmetterlingsflieder',                        1, 2.40,5,5),
231      ('Lavendelheide',                                2, 6.80,5,5)
232      ;
233
234
235
236
237
238
239
240 insert into service(servname, servprice) value
241      ('Trauerkranz',                                   100.00),
242      ('gesteckte Arrangement',                        200.00),
243      ('bepflanzte Trauerschale',                      300.00),
244      ('Brautstrauß',                                  400.00),
245      ('Königinnenstrauß',                             100.00),
246      ('Kutschengesteck',                             200.00),
247      ('Autogesteck',                                  300.00),
248      ('Neuanlage Grabstätte',                        400.00),

```

```

229      ('Grabumrandung mit Heckenpflanzen', 100.00),
230      ('Bodendeckende Bepflanzung',      200.00)
231  ;
232
233  insert into orderdetail(odamount, fk_oid, ordtype,fk_prid, fk_servid)
234  value
235      (1,2, 'PRODUCT', 3,NULL),
236      (5,1, 'PRODUCT', 2,NULL),
237      (4,5, 'PRODUCT', 1,NULL),
238      (3,4, 'PRODUCT', 5,NULL),
239      (2,3, 'PRODUCT', 4,NULL),
240      (4,6, 'PRODUCT', 3,NULL),
241      (4,7, 'PRODUCT', 6,NULL),
242
243      (1,2, 'SERVICE', Null, 3),
244      (5,1, 'SERVICE', Null, 2),
245      (4,5, 'SERVICE', Null, 1),
246      (3,4, 'SERVICE', Null, 5),
247      (2,3, 'SERVICE', Null, 4),
248      (4,6, 'SERVICE', Null, 3),
249      (4,7, 'SERVICE', Null, 6)
250  ;
251
252  insert into employee(efirstname, elastname, esalutation, esalary) value
253      ('Lio', 'Krause', 'Hr.', 2000.00),
254      ('Emilio', 'Zimmermann', 'Hr.', 2300.00),
255      ('David', 'Hartmann', 'Hr.', 1900.00),
256      ('Ahmad', 'Schmitz', 'Hr.', 3000.00),
257      ('Tom', 'Hofmann', 'Div.', 3500.00);
258
259  insert into employeeservice(fk_eid, fk_servid) value
260      (1,2),
261      (1,3),
262      (2,2),
263      (3,3),
264      (4,4),
265      (5,5)
266  ;
267
268  insert into invoice(FK_OID, INVDATE) value
269      (1, '2022-06-30'),
270      (2, '2022-07-04'),
271      (3, '2022-09-14'),
272      (4, '2022-10-11'),
273      (5, '2022-11-29');

```

Aufgabe 2

1. Führen Sie über das SQL-Kommando SELECT die folgenden SQL-Abfragen in der Tabelle product durch:

- Sortieren Sie die Datensätze mit einer SELECT-Anweisung nach Namen und Typ (ORDER BY - Klausel)!

```

1 | select * from product order by PRNAME,FK_PCATID ; # Wenn zuerst nach
   | Name und dann nach Typ (macht weniger Sinn)
2 | select * from product order by FK_PCATID,PRNAME ; # Zuerst Typ und
   | dann nach Namen sortiert

```

2. Selektieren Sie nacheinander (!) die Namen der Produkte (Alias: Produktname) und Einzelpreis mit einem Einzelpreis zwischen Preis 1 bis Preis 2 (WHERE - Klausel mit logischer AND-Verknüpfung) und den Produktnummern zwischen unterer und oberer Grenze!

```

1 | select PRNAME as Produktname, PPRICENETTO from product where
   | PPRICENETTO between 3.5 and 6.5;
2 | select PRNAME as Produktname, PPRICENETTO from product where
   | (PPRICENETTO between 3.5 and 6.5) and (PID between 2 an 5);

```

2. Führen Sie die folgenden SQL-Abfragen als Verbundabfrage über die Tabellen order und customer durch:

1. Ermitteln Sie Anzahl der Bestellungen eines Kunden (Alias: Anzahl Bestellungen für einen Kunden) (COUNT - Funktion) und geben Sie diese und den Namen des Kunden (Alias: Kunde) aus!

```

1 | select count(OID) as Anzahl1 , concat(CLASTNAME,', ',CFIRSTNAME) as
   | Kunde
2 |         from order1
3 |         inner join customer on order1.FK_CID = customer.CID
4 |         where CID=3 and FK_CID=CID;
5 |
6 |
7 | select count(OID) as Anzahl1 , concat(CLASTNAME,', ',CFIRSTNAME) as
   | Kunde from order1,customer where fk_cid=3 and FK_CID=CID
8 |
9 |

```

2. Ermitteln Sie die Namen aller Kunden (Alias: Kundenname) und die Anzahl der Bestellungen für jeden Kunden (Alias: Anzahl) und Gruppieren Sie die Ergebnisse nach der Anzahl (COUNT und GROUP BY - Klausel)!

```

1  select concat(CLASTNAME,', ',CFIRSTNAME) as Kunde,
2         count(OID) as Anzahl
3         from order1
4         inner join customer on order1.FK_CID = customer.CID
5         group by FK_CID;
6
7
8
9
10        select count(OID) as Anzahl , concat(CLASTNAME,',
11        ',CFIRSTNAME) as Kunde
12        from order1, customer
13        where FK_CID=CID
14        group by FK_CID;

```

3. Ermitteln Sie den Kundennamen (Alias: Kunde) und die Anzahl der Bestellungen für ein konkretes Produkt (via FK_PRID) für diesen Kunden (Alias: Anzahl) (COUNT und GROUP BY, INNER JOIN - Klausel)!

```

1  select concat(CLASTNAME,', ',CFIRSTNAME) as Kunde,
2         count(FK_CID) as Anzahl
3         from order1
4         inner join customer on order1.FK_CID = customer.CID
5         inner join orderdetail o on order1.OID = o.FK_OID
6         where FK_PRID=3
7         group by FK_CID;

```

3. Führen Sie die folgenden SQL-Abfragen ggf. als Verbundabfrage über die Tabellen order, customer, orderdetail und product durch:

1. Ermitteln Sie die Produktnamen (Alias: Produkt) für alle Produkttypen (Alias: Produkttyp) aller Einträge in der Tabelle product.

```

1  select PRNAME as Produkt,
2         PCATNAME as Produkttyp
3         from product
4         inner join productcategory
5         on product.FK_PCATID = productcategory.PCATID;

```

2. Ermitteln Sie die Summe aller Produkte in der Tabelle product.

```

1  select count(prid) from product;

```

3. Ermitteln Sie den Kundennamen (Alias: Kunde), die bestellten Produkte (Alias: Produktname), das Lieferdatum und

den jeweiligen Bestellpreis für einen bestimmten Kunden über alle Bestellungseinträge.

```
1 select CLASTNAME as Kunde,  
2        PRNAME as Produkt,  
3        ODELIVDATE as Lieferdatum,  
4        (PPRICENETTO*ODAMOUNT) as Bestellpreis  
5 from order1  
6 inner join customer c on order1.FK_CID = c.CID  
7 inner join orderdetail o on order1.OID = o.FK_OID  
8 inner join product p on o.FK_PRID = p.PRID  
9 where cid=6
```

4. Welche Produkte wurden noch nicht bestellt?

```
1 select PRNAME as Produkt  
2 from order1  
3 inner join orderdetail o on order1.OID = o.FK_OID  
4 right join product p on o.FK_PRID = p.PRID  
5 where OID is null;  
6  
7 select PRNAME as Produkt  
8 from orderdetail  
9 right join product p on FK_PRID = p.PRID  
10 where OdID is null;  
11  
12 select PRNAME as Produkt, o.ODID  
13 from product  
14 left join orderdetail o on product.PRID = o.FK_PRID  
15 where o.OdID is null;
```

5. Ermitteln Sie die Anzahl an Bestellungen für jedes Produkt!

```
1 #Anzahl der bestellten Blumen(Gesamtmenge)  
2 select PRNAME as Produkt,  
3        sum(ODAMOUNT) as Anzahl  
4 from product  
5 left join orderdetail o on product.PRID = o.FK_PRID  
6 group by Prid;  
7  
8  
9 #Anzahl der Bestellungen wo die Blumen vorkommen  
10 select PRNAME as Produkt,  
11        count(FK_PRID) as Anzahl  
12 from product  
13 left join orderdetail o on product.PRID = o.FK_PRID  
14 group by Prid;
```

