

# Bericht Datenbank Praktikum

Julian Sobott (76511), David Sugar (76050), Lukas Mendel (76509)

9. Januar 2020

## Inhaltsverzeichnis

<b>1</b>	<b>Aufgabe 1</b>	<b>2</b>
1.1	a) . . . . .	2
1.1.1	Entities . . . . .	3
1.1.2	Relations . . . . .	3
1.2	b) . . . . .	4
1.2.1	Relationen . . . . .	4
1.2.2	Referenzen . . . . .	4
1.3	c) . . . . .	5
<b>2</b>	<b>Aufgabe 2</b>	<b>6</b>
<b>3</b>	<b>Aufgabe 3</b>	<b>7</b>
<b>4</b>	<b>Aufgabe 4</b>	<b>8</b>
4.1	a) . . . . .	8
4.2	c) . . . . .	8

# 1 Aufgabe 1

## 1.1 a)

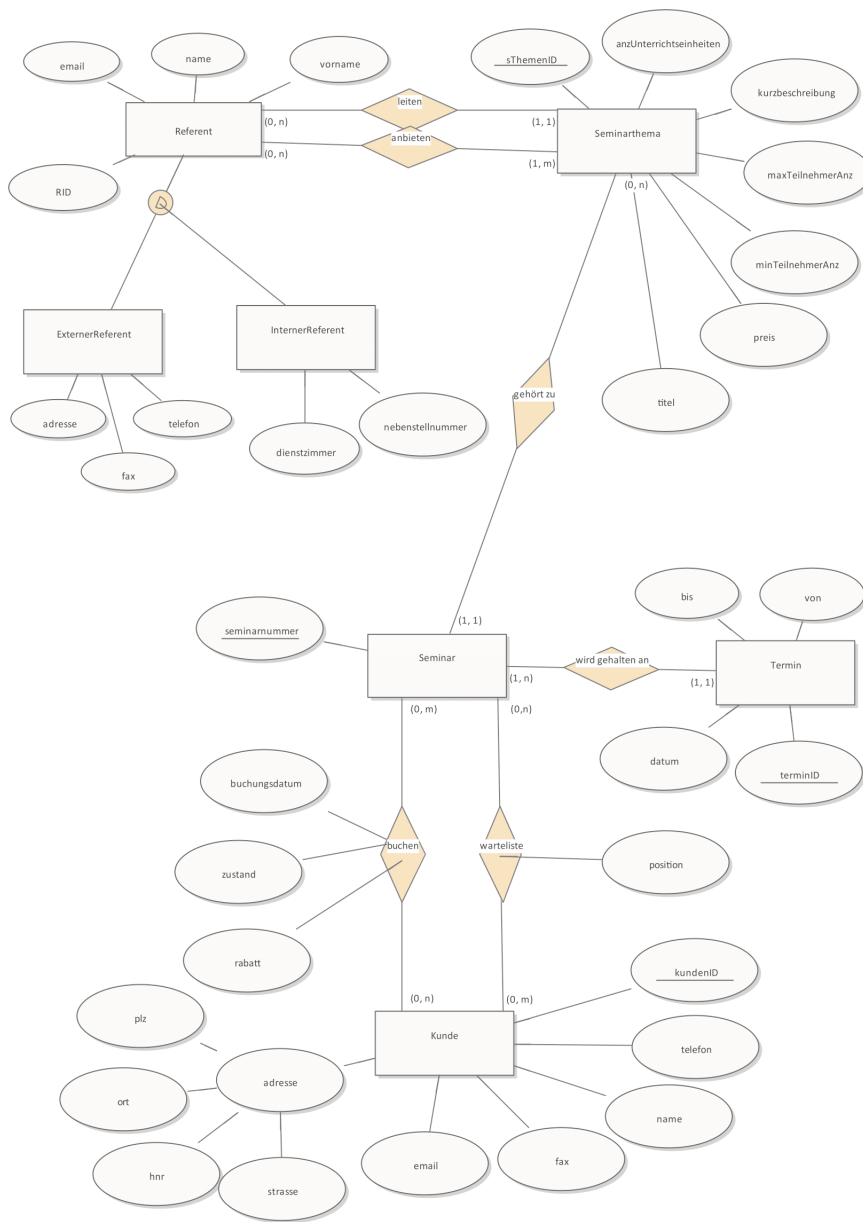


Abbildung 1: ER-Modell Seminarverwaltung

### 1.1.1 Entities

seminar = ({seminarnummer: Integer})

termin = ({terminid: Integer, datum: Date, von: DateTime, bis: DateTime})

kunde = ({kundenid: Integer, telefon: Varchar, name: Varchar, fax: Varchar, email: Varchar, adresse: (plz: Varchar, ort: Varchar, hnr: Varchar, str: Varchar)})

referent = ({rid: Integer, email: Varchar, name: Varchar, vorname: Varchar})

seminarthema = ({sthemaid: Integer, anzunterrichtseinheiten: Integer, kurzbeschreibung: Varchar, maxteilnehmeranz: Integer, minteilnehmeranz: Integer, preis: Float, titel: Varchar})

externerReferent = ({rid: Integer, adresse: (strasse: Varchar, hnr: Varchar, plz: Varchar, ort: Varchar)})

internerReferent = ({rid: Integer, dienstzimmer: Varchar, nebenstellnummer: Integer})

### 1.1.2 Relations

leiten = (referent x seminarthema)

anbieten = (referent x seminarthema)

gehört\_zu = (seminarthema x seminar)

buchen = (seminar x kunde, buchungsdatum: Date, zustand: Varchar, rabatt: Float)

warteliste = (kunde x seminar, datum: Date)

wird\_gehalten\_an = (seminar x termin)

## 1.2 b)

### 1.2.1 Relationen

referent = (rid:Integer, email:Varchar, name:Varchar, vorname:Varchar)

externerReferent = (rid:Integer, strasse:Varchar, hnr:Varchar, plz:Varchar, ort:Varchar)

internerReferent = (rid:Integer, dienstzimmer:Varchar, nebenstellnummer:Integer)

seminarthema = (sthemaid:Integer, anzunterrichtseinheiten:Integer, kurzbeschreibung:Varchar, maxteilnehmeranz:Integer, minteilnehmeranz:Integer, preis:Float, titel:Varchar, leiter:Integer)

anbieten = (referentid:Integer, seminarthemaid:Integer)

seminar = (seminarnummer:Integer, seminarthemaid:Integer)

termin = (terminid:Integer, von:Datetime, bis:Datetime, datum:Date, seminarid:Integer)

kunde = (kundenid:Integer, telefon:Varchar, name:Varchar, fax:Varchar, email:Varchar, plz:Varchar, ort:Varchar, hnr:Varchar, str:Varchar)

buchen = (kundenid:Integer, seminarnr:Integer, buchungsdatum:Date, zustand:Varchar, rabatt:Float)

warteliste = (kundenid:Integer, seminarnr:Integer, datum:Date)

### 1.2.2 Referenzen

$seminarthema|LEITER \subseteq referent|RID$   
 $anbieten|REFERENTID \subseteq referent|RID$   
 $anbieten|SEMINARTHEMAID \subseteq seminarthema|STHEMAID$   
 $seminar|SEMINARTHEMAID \subseteq seminarthema|STHEMAID$   
 $termin|SEMINARID \subseteq seminar|SEMINARNUMMER$   
 $buchen|KUNDENID \subseteq kunde|KUNDENID$   
 $buchen|SEMINARNR \subseteq seminar|SEMINARNR$   
 $warteliste|KUNDENID \subseteq kunde|KUNDENID$   
 $warteliste|SEMINARNR \subseteq seminar|SEMINARNR$

### 1.3 c)

```
CREATE TABLE g8_referent (  
    rid serial PRIMARY KEY,  
    email VARCHAR(50),  
    name VARCHAR(50),  
    vorname VARCHAR(50)  
);  
  
CREATE TABLE g8_seminarthema (  
    sthemaid serial PRIMARY KEY,  
    anz_unterrichtseinheiten INTEGER,  
    kurzbeschreibung VARCHAR,  
    max_teilnehmeranzahl INTEGER,  
    min_teilnehmeranzahl INTEGER,  
    preis FLOAT,  
    titel VARCHAR(200),  
    leiter INTEGER REFERENCES g8_referent(rid)  
);  
  
CREATE TABLE g8_anbieten (  
    referenten_id INTEGER REFERENCES g8_referent(rid),  
    sthemaid INTEGER REFERENCES g8_seminarthema(sthemaid),  
    PRIMARY KEY (referenten_id, sthemaid)  
);  
  
CREATE TABLE g8_seminar (  
    seminarnummer serial PRIMARY KEY,  
    sthemaid INTEGER REFERENCES g8_seminarthema(sthemaid)  
);  
  
CREATE TABLE g8_termin (  
    terminid serial PRIMARY KEY,  
    von TIME,  
    bis TIME,  
    datum DATE,  
    seminarnummer INTEGER REFERENCES g8_seminar(seminarnummer)  
);  
  
CREATE TABLE g8_kunde (  
    kundenid serial PRIMARY KEY,  
    telefon VARCHAR(20),  
    name VARCHAR(50),  
    fax VARCHAR(20),  
    email VARCHAR(50),  
    plz VARCHAR(10),  
    ort VARCHAR(50),  
    hnr VARCHAR(10),  
    str VARCHAR(50)  
);  
  
CREATE TYPE g8_zustand as ENUM ('offen', 'gebucht', 'berechnet', 'gezahlt', 'storniert');  
  
CREATE TABLE g8_buchen (  
    kundenid INTEGER REFERENCES g8_kunde(kundenid),  
    seminarnummer INTEGER REFERENCES g8_seminar(seminarnummer),  
    datum DATE,  
    zustand g8_zustand,  
    rabatt FLOAT,
```

```

PRIMARY KEY (kundenid, seminarnummer)
);

CREATE TABLE g8_warteliste (
    kundenid INTEGER REFERENCES g8_kunde(kundenid),
    seminarnummer INTEGER REFERENCES g8_seminar(seminarnummer),
    datum Date,
    PRIMARY KEY (kundenid, seminarnummer)
);

CREATE TABLE g8_ExternerReferent(

RID int PRIMARY KEY,
fax VARCHAR(50),
telefon VARCHAR(20),
PLZ VARCHAR(15),
Strasse VARCHAR(50),
Hnr VARCHAR (20),
Ort VARCHAR(50),

FOREIGN KEY (RID) REFERENCES g8_referent(RID)
);

CREATE TABLE g8_InternerReferent(

RID int PRIMARY KEY,
Dienstnummer VARCHAR(30),
nebenstellenummer INTEGER,

FOREIGN KEY (RID) REFERENCES g8_referent(RID)
);

```

## 2 Aufgabe 2

```

INSERT INTO g8_referent(email, vorname, name) values
('julian.sobott@wtf.de', 'Julian', 'Sobott'),
('david.sugar@wtf.de', 'David', 'Sugar'),
('lukas.mendel@wtf.de', 'Lukas', 'Mendel'),
('gregor.grambow@wtf.de', 'Gregor', 'Grambow');

INSERT INTO g8_internerreferent(rid, plz, ort, strasse, hnr) values
((select rid from g8_referent where name = 'Grambow' and vorname = 'Gregor')
, '73434', 'Aalen', 'Uni-Str', '111');

INSERT INTO g8_externerreferent(rid, plz, ort, strasse, hnr) values
((select rid from g8_referent where name = 'Sugar' and vorname = 'David'), '
73434', 'Aalen', 'Uni-Str', '111'),
((select rid from g8_referent where name = 'Sobott' and vorname = 'Julian'),
'73434', 'Aalen', 'Uni-Str', '111'),
((select rid from g8_referent where name = 'Mendel' and vorname = 'Lukas'),
'73434', 'Aalen', 'Uni-Str', '111');

INSERT INTO g8_seminarthema(anz_unterrichtseinheiten, kurzbeschreibung,
max_teilnehmeranzahl, min_teilnehmeranzahl, preis, titel, leiter) values
(10, 'Datenbanken□Grundlagen□erlernen.', 30, 5, 152.50, 'Datenbanken', (
select rid from g8_referent where name = 'Grambow' and vorname = 'Gregor'
)),
(2, 'We□love□RISC', 10, 1, 0.0, 'The□ARM□Architecture', (select rid from
g8_referent where name = 'Sugar' and vorname = 'David')),

```

```
(3, 'Its not a snake', 15, 3, 43.90, 'Python', (select rid from g8_referent
where name = 'Julian' and vorname = 'Sobott')));
```

```
INSERT INTO g8_seminar (sthemaId)
values (1), (2), (3), (1), (3);
```

```
INSERT INTO g8_termin (von, bis, datum, seminarnummer)
values
('09:30', '13:00', '18/1/1999', 1),
('09:30', '13:00', '19/1/1999', 2),
('09:30', '13:00', '20/1/1999', 3),
('09:30', '13:00', '21/1/1999', 4),
('09:30', '13:00', '22/1/1999', 5),
('10:30', '18:45', '18/1/2050', 1),
('10:30', '18:45', '19/1/2050', 5);
```

```
INSERT INTO g8_kunde (telefon, name, fax, email, plz, ort, hnr, str)
values
('0176111', 'Pete', '0176-54', 'pete@bs.de', '12345', 'Buxdehude', '3', '
kennIchNichtWeg'),
('0176112', 'Steve', '0176-55', 'steve@bs.de', '12345', 'Buxdehude', '3', '
kennIchNichtWeg'),
('0176113', 'Eve', '0176-56', 'eve@bs.de', '12345', 'Buxdehude', '3', '
kennIchNichtWeg'),
('0176114', 'Paula', '0176-57', 'paula@bs.de', '12345', 'Buxdehude', '3', '
kennIchNichtWeg'),
('0176115', 'Klaus', '0176-58', 'klaus@bs.de', '12345', 'Buxdehude', '3', '
kennIchNichtWeg');
```

```
INSERT INTO g8_buchen (kundenid, seminarnummer, datum, zustand, rabatt)
values
(1, 1, '13/1/1999', 'gezahlt', 0.0),
(1, 2, '13/1/1999', 'gezahlt', 0.0),
(2, 1, '13/1/1999', 'berechnet', 0.3),
(2, 3, '13/1/1999', 'gezahlt', 0.0),
(3, 3, '14/1/1999', 'gebucht', 0.0),
(3, 4, '14/1/1999', 'gezahlt', 0.0),
(4, 3, '14/1/1999', 'gebucht', 0.0),
(5, 3, '14/1/1999', 'gebucht', 0.0),
(3, 2, '15/1/1999', 'offen', 0.0),
(1, 3, '14/1/1999', 'berechnet', 0.7);
```

```
INSERT INTO g8_warteliste (kundenid, seminarnummer, position)
values
(1,1,1),
(2,1,2),
(3,2,1);
```

### 3 Aufgabe 3

```
SELECT(
(SELECT COUNT (rid) as AnzahlInterne
FROM g8_InternerReferent),
(SELECT COUNT (rid) as AnzahlExterne
FROM g8_ExternerReferent),
(SELECT COUNT (rid) as AnzahlGesamt
FROM g8_referent)
);

SELECT seminarnummer, COUNT(seminarnummer)
```

```
FROM g8_seminar s JOIN g8_termin t on s.seminarnummer = t.seminarnummer
GROUP BY seminarnummer;
```

```
SELECT Seminarummer, COUNT(Seminarummer) as Teilnehmeranzahl
FROM g8_buchen b JOIN g8_seminar s on b.seminarnummer = s.seminarnummer
GROUP BY(Seminarummer)
```

```
SELET Seminarummer, , MAX()
```

## 4 Aufgabe 4

### 4.1 a)

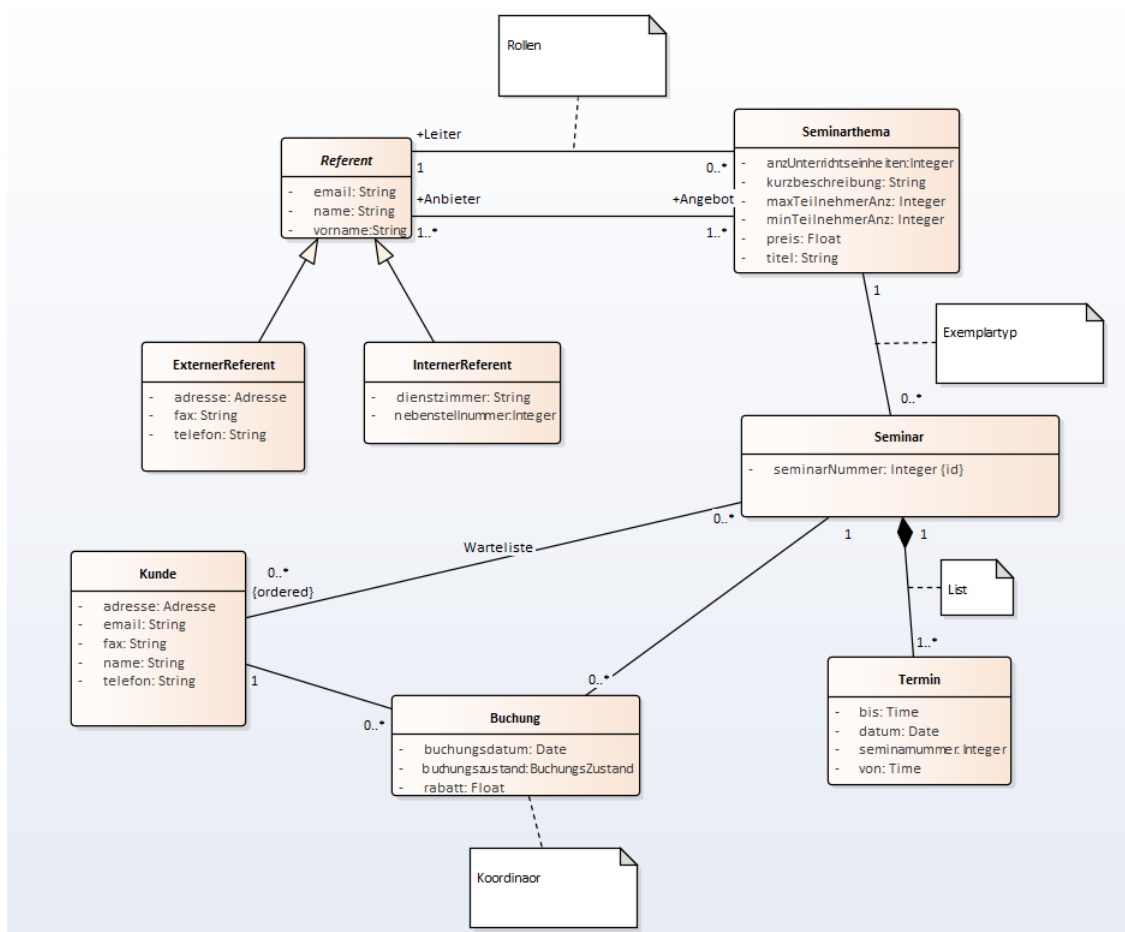


Abbildung 2: OOM Modell

Da Vererbung so nicht in einem ER-Modell möglich ist, haben wir eine Anpassung vorgenommen. Ein Interner/Externer Referent hat jeweils noch einen Verweis auf den Eintrag von Referent. AuSSerdem haben wir statt einer Klasse Buchung eine Beziehung buchen mit den Attributen aus der Klasse.

### 4.2 c)

Ein **Referent** kann 0 Seminarthemen leiten, da er auch nur Seminare anbieten kann oder beliebig viele da hier keine Beschränkung vorgegeben war. Laut Aufgabe kann ein Seminar von genau einem Referent geleitet werden aber mehrere Anbieter haben. Ein Referent kann gleichzeitig Anbieter und Leiter sein.

Ein Seminarthema kann 0 Seminare haben, um sicherzustellen, dass zur Anmeldung nicht schon ein Seminar eingetragen werden muss. Es kann aber im Laufe mehrere Seminare haben. In einem Seminar kann nur genau ein Seminarthema behandelt werden.

Ein Seminar kann an mehreren Terminen statt finden. Muss aber an mindestens eins. Ein Termin kann nur zu genau einem Seminar gehören.



Kunden können angelegt werden ohne an einem Seminar teilzunehmen oder auf einer Warteliste zu stehen. Deshalb die 0 Kardinalitäten. Im Laufe können sie aber an beliebig vielen Seminaren teilnehmen oder auf Wartelisten stehen. In die andere Richtung gilt genau das gleiche.