

Datenbankpraktikum

Dozent: Hr. Schäfer

Kursfahrten

Erstellt von:

Thorsten Hartung
Sebastian Herker
Thomas Nordmann
Alisa Vogt

Inhaltsverzeichnis

<i>Beschreibung der Miniwelt Kursfahrten.....</i>	<i>2</i>
<i>Das ER-Modell.....</i>	<i>2</i>
<i>Im ER-Modell nicht dargestellte semantische Bezüge.....</i>	<i>3</i>
<i>Funktionale Abhängigkeiten.....</i>	<i>3</i>
<i>Das Relationenmodell mit allgemeinen Typen- und Schlüsselangaben.....</i>	<i>3</i>
<i>Die Transformation des ER-Modells in das Relationenmodell.....</i>	<i>3</i>
<i>Textvorlage: Aufgaben zur Modellierung.....</i>	<i>4</i>
<i>Daten der Tabellen.....</i>	<i>4</i>
<i>Anfragen an die Datenbank.....</i>	<i>6</i>
<i>SQL: Create, Insert, Drop.....</i>	<i>6</i>
<i>SQL: Anfragen.....</i>	<i>9</i>
<i>Textvorlage: Aufgaben zu Anfragen.....</i>	<i>11</i>
<i>DES: Kursfahrten-Schema.....</i>	<i>13</i>
<i>DES: Daten zu den Relationen.....</i>	<i>13</i>
<i>DES: Anfragen in Relationenalgebra.....</i>	<i>13</i>
<i>DES: Anfragen im Tupelkalkül.....</i>	<i>14</i>
<i>DES: Antworten zu den Anfragen im Tupelkalkül.....</i>	<i>15</i>
<i>DES: Anfragen im Domänenkalkül.....</i>	<i>16</i>
<i>Beschreibung und Motivation des View.....</i>	<i>17</i>
<i>Basismöglichkeiten und Erweiterung des Views.....</i>	<i>17</i>
<i>Textvorlage: Aufgaben bzgl. des Views.....</i>	<i>17</i>

Beschreibung der Miniwelt Kursfahrten

Im Zentrum dieser Miniwelt stehen die Fahrten (Klassen-, Kurs-, Jahrgangsfahrt, o. ä.), die mit einem Namen bezeichnet sind, ein Ziel und ein Start- und Enddatum haben.

Bei jeder Fahrt fahren theoretisch unbegrenzt viele Teilnehmer mit. Jeder Teilnehmer kann an mehreren Fahrten mitfahren, solange sich die Fahrten nicht zeitlich überschneiden. Neben Name und Adresse spielt bei den Teilnehmern z. B. für die Zimmeraufteilung das Geschlecht eine Rolle, ebenso sollten eine Mobilnummer und eine Notfallnummer hinterlegt sein. Auch sollten eventuelle Besonderheiten wie Medikamente, Allergien, Krankheiten o. ä. vermerkt werden.

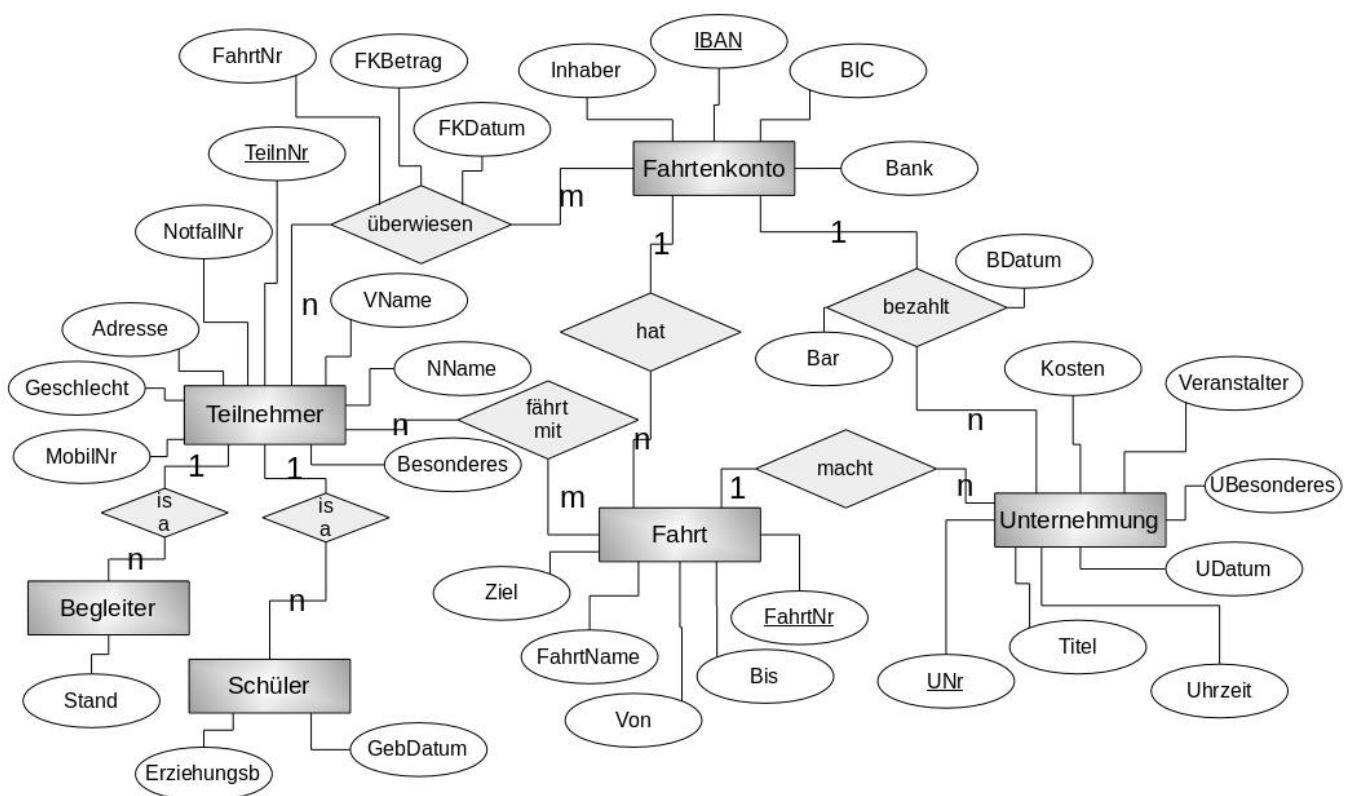
Die Gruppe der Teilnehmer teilt sich auf in Begleiter, welche Lehrer, Eltern, Erzieher o. ä. sein können, und Schüler, bei denen es eine Rolle spielt, ob sie volljährig sind und wer ggf. erziehungsbe-rechtigt ist. Die Volljährigkeit könnte auch auf der Fahrt erreicht werden, weswegen das Geburtsdatum angegeben wird. An jeder Fahrt muss mind. ein Lehrer teilnehmen.

Jede Fahrt macht eine Reihe von eigenen Unternehmungen, wobei auch Unterkunft, An- und Abreise, gemeinsames Essen usw. als Unternehmung geführt wird. Zu diesen Unternehmungen werden Datum und Uhrzeit, Kosten, Veranstalter und Besonderheiten (Badehose & Surfbrett, Museumsführer, Skiausrüstung, o. ä.) eingetragen.

Jede Fahrt hat ein Fahrtenkonto bei einer Bank mit IBAN, BIC und einem Inhaber. Während jede Fahrt nur ein Konto haben soll, aber ein Konto für mehrere Fahrten genutzt werden, z. B. das Konto des Klassenlehrers für die Klassenfahrten in mehreren Jahren.

Auf dieses Fahrtenkonto überweisen die Teilnehmer je einen Betrag zu einem Datum für eine bestimmte Fahrt und von diesem Konto aus werden die Unternehmungen zu einem Datum bezahlt, wobei vermerkt werden soll, ob es sich um eine Barzahlung (z. B. bei einem Essen) handelt.

Das ER-Modell



Im ER-Modell nicht dargestellte semantische Bezüge

Im ER-Modell konnte nicht dargestellt werden, dass bei jeder Fahrt mind. ein Lehrer teilnehmen soll. Ebenso konnte nicht dargestellt werden, dass Teilnehmer nicht an sich zeitlich überschneidenden Fahrten teilnehmen können.

In der Beziehung "überwiesen" wird als Fremdschlüssel das Attribut "FahrtNr" verwendet. Dieser muss über die Entität "Fahrt" geprüft werden. (z. B. mit CONSTRAINT in SQL)

Funktionale Abhängigkeiten

Im Modell existieren folgende nicht-triviale Funktionale Abhängigkeiten (s. Relationenmodell):

In der Relation "Fahrtenkonto" sind die Attribute "BIC" und "Bank" gegenseitig voneinander abhängig (da sie im Prinzip das Gleiche bezeichnen).

Es wäre auch denkbar, lediglich eines der Attribute einzufügen (z. B. BIC), jedoch ist die gegebene Form benutzerfreundlicher.

Ebenso sind in der Relation "Teilnehmer" die Attribute Vor- und Nachname (zusammen genommen) und Mobilnummer gegenseitig voneinander abhängig, zumindest, wenn wir davon ausgehen, dass jeder Teilnehmer nur eine Nummer hat.

Das Relationenmodell mit allgemeinen Typen- und Schlüsselangaben

Fahrtenkonto (IBAN text, BIC text, Inhaber text, Bank text);

Fahrt (FahrtNr zahl, Fahrtnamen text, Ziel text, Von datum, Bis datum, Fahrtenkonto.IBAN);

Teilnehmer (TeilnNr zahl, MobilNr text, Geschlecht zeichen, Adresse text, NotfallNr text, VName text, NName text, Besonderes text);

Begleiter (Teilnehmer.TeilnNr zahl, Stand text);

Schueler (Teilnehmer.TeilnNr zahl, Erziehungsberechtigte text, GebDatum datum);

FaehrtMit (Teilnehmer.TeilnNr zahl, Fahrt.FahrtNr zahl);

Unternehmung (UNr zahl, Titel text, Kosten kommazahl, Veranstalter text, UBesonderes text, UDatum datum, Uhrzeit time, Fahrt.FahrtNr zahl);

Bezahl (Unternehmung.UNr zahl, Fahrtenkonto.IBAN text, BDatum datum, Bar bool);

Ueberwiesen (Teilnehmer.TeilnNr zahl, Fahrt.FahrtNr zahl, FKBetrag kommazahl, FKDatum datum);

Die Transformation des ER-Modells in das Relationenmodell

Insgesamt wurden 9 Entitäten bzw. Beziehungen zu Relationen transformiert und 4 Beziehungen aufgelöst.

Die 6 Entitäten Fahrt, Teilnehmer, Begleiter, Schüler, Fahrtenkonto und Unternehmung wurden zunächst vollständig in Relationen überführt.

Aufgrund der n:m-Kardinalitäten wurden die Beziehungen "fährt mit" und "überwiesen" ebenfalls in Relationen überführt, jeweils mit Angabe der Schlüssel der verbundenen Entitäten als Fremdschlüssel. Diese Fremdschlüssel bilden zusammengesetzt den Primärschlüssel der Relationen.

Wegen der zugehörigen Attribute wurde für die Übersichtlichkeit auch die 1:n-Beziehung "bezahlt" in eine Relation überführt, jeweils mit Angabe der Schlüssel der verbundenen Entitäten als Fremdschlüssel. Diese Fremdschlüssel bilden zusammengesetzt den Primärschlüssel der Relation.

Die 1:n-Beziehungen ohne Attribute "is a" (2x), "hat" und "macht" wurden aufgelöst, wofür in die Relationen der n-Seite der Schlüssel der verbundenen Entität als Fremdschlüssel eingefügt wurde.

Textvorlage: Aufgaben zur Modellierung

Szenario:

Die Schule plant die Einführung einer Datenbank zur Verwaltung von Klassen- bzw. Kursfahrten. Zu jeder Fahrt gibt es Teilnehmer, dies können Begleiter oder Schüler_Innen sein. Zu jeder Fahrt gibt es Unternehmungen, diese werden bei Veranstaltern gebucht. Jede Fahrt hat ein eigenes Fahrtenkonto, in welches die Teilnehmer einzahlen und von dem aus die Unternehmungen bezahlt werden. Schüler_Innen können an mehreren Fahrten teilnehmen.

Aufgabe 1)

Setzt euch zu dritt zusammen und überlegt euch ein ER Modell und beschriftet die Kardinalitäten.

Aufgabe 2)

a) Überführt euer Modell in ein Relationenmodell.

b) Überlegt euch Zusammenhänge, die gesondert geprüft werden müssen und sich nicht aus euren Tabellen ergeben. Kann es zu Überschneidungen oder Widersprüchen kommen?

Aufgabe 3)

Teilt euch auf und trifft euch mit einem_r Schüler_In einer anderen Gruppe und stellt euer Modell vor und vergleicht es mit dem Modell der anderen Gruppen.

Aufgabe 4)

Verbessert euer Modell nach den Erkenntnissen aus Aufgabe 3).

Daten der Tabellen

(Die vollständige Tabelle ist zu finden in: ../modell/kursfahrten-Daten.txt)

```

--- Fahrt ---
fahrtnr | fahrtname | ziel | von | bis | iban
-----+-----+-----+-----+-----+-----
201701 | Englisch_Kose | Orlando | 2017-05-25 | 2017-05-30 | DE19200456783417345678
201702 | Geschichte_Aktmann | Dresden | 2017-06-05 | 2017-06-10 | DE19167944520905876021
201703 | Informatik_Schachmann | London | 2017-06-07 | 2017-06-12 | DE19183417341534123414
(3 Zeilen)

```

```

--- Fahrtenkonto ---
iban | bic | inhaber | bank
-----+-----+-----+-----
DE19183417341534123414 | DEUTDEFF | Conrad Weiland | Deutsche Bank
DE19167944520905876021 | COBADEFF | Daniela Altmann | Norisbank
DE19200456783417345678 | DEUTDEFF | Matthias Lang | Deutsche Bank
(3 Zeilen)

```

--- Teilnehmer --- [jeweils über 2 Zeilen]

teilnnr	moblnr	geschlecht	adresse	notfallnr
vname	nname	besonderes		
1	+49 159 0416361	w	Gruenauer Strasse 60 12353 Berlin	+49 30 271166
Claudia	Nadel			
2	+49 117 9933864	w	Kurfuerstenstraße 77 12353 Berlin	+49 30 278966
Lisa	Vogel	Veganerin		
3	+49 174 0281630	w	Fugger Strasse 23 12353 Berlin	+49 30 270356
Maria	Roth	Pollenallergie, Notfallspray beim Lehrer		
4	+49 174 5192538	w	Eichendorffstr. 69 12353 Berlin	+49 30 454566
Jennifer	Werner			
5	+49 159 0484162	w	An Der Urania 32 12353 Berlin	+49 30 343666
[...]				

(28 Zeilen)

--- Begleiter ---

teilnnr	stand
21	Elternteil
22	Lehrer
23	Elternteil
24	Lehrer
[...]	

(8 Zeilen)

--- Schueler ---

teilnnr	erziehungsberechtigte	gebdatum
1	Anette Nadel	1999-09-24
2	Mark Vogel	2003-07-12
3	Felix Roth	2002-02-01
4	Andreas Werner	2002-12-04
[...]		

(20 Zeilen)

--- FaehrtMit ---

teilnnr	fahrtNr
1	201701
1	201703
2	201701
3	201703
[...]	

(30 Zeilen)

--- Unternehmungen --- [jeweils über 2 Zeilen]

unr	titel	kosten	veranstalter	ube-
sonderes	udatum	uhrzeit	fahrtNr	
1	Anreise Bus	1400	Bussard-Travel	10 Std. Nachtfahrt
2017-06-07 18:00:00 201703				
2	Abreise Bus	0	Bussard-Travel	Abreisekosten incl.
2017-06-12 08:00:00 201703				
3	Mittagessen	0	Queen Hostel	
2017-06-08 12:00:00 201703				
4	Fuehrung durch London	75	Guides International	
2017-06-08 12:00:00 201703				
[...]				

(50 Zeilen)

--- Beahlt ---

unr	iban	bdatum	bar
1	DE19183417341534123414	2017-04-01	f
4	DE19183417341534123414	2017-06-08	t
5	DE19183417341534123414	2017-06-08	t
7	DE19183417341534123414	2016-12-12	f
[...]			

(23 Zeilen)

--- Ueberwiesen ---

teilnnr	fahrtNr	fkbetrag	fkdatum
1	201701	1600	2017-01-17
1	201703	750	2016-12-06
2	201701	1600	2017-01-22
3	201703	750	2017-02-23

[...]

(30 Zeilen)

Anfragen an die Datenbank

1. Zeige alle Fahrten.
2. Zeige alle teilnehmenden Personen.
3. Zeige alle Ziele der Fahrten.
4. Zeige die Nachnamen der Teilnehmer mit den entsprechenden Notfallnummern.
5. Zeige den vollständigen Namen aller weiblichen Teilnehmer.
6. Zeige alle Teilnehmer mit dem Nachnamen Vogel.
7. Zeige alle Unternehmungen, die teurer als 100 € sind.
8. Zeige alle Unternehmungen der Fahrt Nr. 201701, die teurer als 100 € sind.
9. Gib die nach dem Nachnamen sortierte Liste aller Teilnehmer aus.
10. Gib die nach Kosten absteigend sortierte Liste der Unternehmungen aus.
11. Zeige nur die teuerste Unternehmung.
12. Zeige alle Teilnehmer, die nach Dresden fahren.
13. Zeige die Nachnamen aller begleitenden Lehrer.
14. Zeige alle Besonderheiten der Unternehmungen auf der Fahrt nach Orlando.
15. Zeige alle Besonderheiten der teilnehmenden Schüler mit Vor- und Nachnamen der Schüler.
16. Zeige nur Schüler (mit Vor- und Nachnamen) und ihre Besonderheiten, wenn sie welche haben.
17. Zeige eine sortierte Liste aller Unternehmungen (Titel), die nicht bei der Fahrt nach Orlando stattfinden.
18. Zeige alle Teilnehmernummern und vollständigen Namen der Teilnehmer, die nach Dresden fahren.
19. Zeige alle Schüler (mit vollständigem Namen), die nach London fahren.
20. Zeige, welche Aktivitäten bei der Kursfahrt in die USA (nach Datum sortiert) stattfinden.
21. Zeige die Teilnehmernummern und den Nachnamen der Schüler, die an mehreren Fahrten teilnehmen.
22. Zeige die Summe der Kosten der Unternehmungen der Londonfahrt (FahrtNr=201703).

SQL: Create, Insert, Drop

(Die vollständigen Daten sind zu finden in: ../sql/)

```
CREATE TABLE Fahrtenkonto (IBAN varchar(22) primary key, BIC varchar(11), Inhaber
varchar(40), Bank varchar(40));
```

```
CREATE TABLE Fahrt (FahrtNr integer primary key, Fahrtrname varchar(40), Ziel varchar(40), Von
date, Bis date, IBAN varchar(22) references Fahrtenkonto);
```

```
CREATE TABLE Teilnehmer (TeilnNr integer primary key, MobilNr varchar(20), Geschlecht char,
Adresse varchar(100), NotfallNr varchar(20), VName varchar(40), NName varchar(40),
Besonderes varchar(100));
CREATE TABLE Begleiter (TeilnNr integer references Teilnehmer primary key, Stand varchar(20));
CREATE TABLE Schueler (TeilnNr integer references Teilnehmer primary key,
Erziehungsberechtigte varchar(40), GebDatum date);
CREATE TABLE Faehrtmit (TeilnNr integer references Teilnehmer, FahrtNr integer references
Fahrt, CONSTRAINT beg PRIMARY KEY (TeilnNr, FahrtNr));
CREATE TABLE Unternehmung (UNr integer primary key, Titel varchar(40), Kosten float(20),
Veranstalter varchar(40), UBesonderes varchar(200), UDatum date, Uhrzeit time, FahrtNr integer
references Fahrt);
CREATE TABLE Bezahlt (UNr integer references Unternehmung, IBAN varchar(22) references
Fahrtenkonto , BDatum date, Bar bool, CONSTRAINT bez PRIMARY KEY (IBAN, UNr));
CREATE TABLE Ueberwiesen (TeilnNr integer references Teilnehmer, FahrtNr integer references
Fahrt, FKBetrag float(20), FKDatum date, CONSTRAINT ueb PRIMARY KEY (TeilnNr, FahrtNr));
```

```
INSERT INTO Fahrtenkonto VALUES ('DE19183417341534123414', 'DEUTDEFF', 'Conrad Wei-
land', 'Deutsche Bank');
INSERT INTO Fahrtenkonto VALUES ('DE19167944520905876021', 'COBADEFF', 'Daniela Alt-
mann', 'Norisbank');
INSERT INTO Fahrtenkonto VALUES ('DE19200456783417345678', 'DEUTDEFF', 'Matthias
Lang', 'Deutsche Bank');
```

```
INSERT INTO Fahrt VALUES (201701, 'Englisch_Kose', 'Orlando', '2017-05-25', '2017-05-30',
'DE19200456783417345678');
INSERT INTO Fahrt VALUES (201702, 'Geschichte_Aktmann', 'Dresden', '2017-06-05', '2017-06-
10', 'DE19167944520905876021');
INSERT INTO Fahrt VALUES (201703, 'Informatik_Schachmann', 'London', '2017-06-07', '2017-06-
12', 'DE19183417341534123414');
```

```
insert into Teilnehmer values(1,'+49 159 0416361','w','Gruenauer Strasse 60 12353 Berlin','+49 30
271166','Claudia','Nadel','');
insert into Teilnehmer values(2,'+49 117 9933864','w','Kurfuerstenstraße 77 12353 Berlin','+49 30
278966','Lisa','Vogel','Veganerin');
insert into Teilnehmer values(3,'+49 174 0281630','w','Fugger Strasse 23 12353 Berlin ','+49 30
270356','Maria','Roth','Pollenallergie, Notfallspray beim Lehrer');
insert into Teilnehmer values(4,'+49 174 5192538','w','Eichendorffstr. 69 12353 Berlin','+49 30
454566','Jennifer','Werner','');
[...]
```

```
insert into Begleiter values (21,'Elternteil');
insert into Begleiter values (22,'Lehrer');
insert into Begleiter values (23,'Elternteil');
insert into Begleiter values (24,'Lehrer');
[...]
```

```
insert into Schueler values (1,'Anette Nadel','1999-09-24');
insert into Schueler values (2,'Mark Vogel','2003-07-12');
insert into Schueler values (3,'Felix Roth','2002-02-01');
insert into Schueler values (4,'Andreas Werner','2002-12-04');
[...]
```

```
insert into Faehrtmit values (1,201701);
insert into Faehrtmit values (1,201703);
insert into Faehrtmit values (2,201701);
insert into Faehrtmit values (3,201703);
[...]
```

```
insert into Unternehmung values (1,'Anreise Bus',1400.00,'Bussard-Travel','10 Std. Nachtfahrt',
'2017-06-07','18:00:00',201703);
insert into Unternehmung values (2,'Abreise Bus',0.00,'Bussard-Travel','Abreisekosten incl.','2017-
06-12','8:00:00',201703);
insert into Unternehmung values (3,'Mittagessen',0.00,'Queen Hostel','',2017-06-
08,'12:00:00',201703);
insert into Unternehmung values (4,'Fuehrung durch London',75.00,'Guides International','',2017-
06-08,'12:00:00',201703);
[...]
```

```
insert into Beahlt values (1,'DE19183417341534123414','2017-04-01',false);
insert into Beahlt values (4,'DE19183417341534123414','2017-06-08',true);
insert into Beahlt values (5,'DE19183417341534123414','2017-06-08',true);
insert into Beahlt values (7,'DE19183417341534123414','2016-12-12',false);
[...]
```

```
insert into Ueberwiesen values (1,201701,1600.00,'2017-01-17');
insert into Ueberwiesen values (1,201703,750.00,'2016-12-06');
insert into Ueberwiesen values (2,201701,1600.00,'2017-01-22');
insert into Ueberwiesen values (3,201703,750.00,'2017-02-23');
[...]
```

```
DROP TABLE Ueberwiesen;
DROP TABLE Beahlt;
DROP TABLE Unternehmung;
DROP TABLE Faehrtmit;
DROP TABLE Schueler;
DROP TABLE Begleiter;
DROP TABLE Teilnehmer;
DROP TABLE Fahrt;
DROP TABLE Fahrtenkonto;
```


SQL: Anfragen

--"Select"

--1. Zeige alle Fahrten:

```
SELECT *
```

```
FROM fahrt;
```

--2. Zeige alle teilnehmenden Personen:

```
SELECT *
```

```
FROM teilnehmer;
```

--3. Zeige alle Ziele der Fahrten:

```
SELECT ziel
```

```
FROM fahrt;
```

--4. Zeige die Nachnamen der Teilnehmer mit den entsprechenden Notfallnummern:

```
SELECT nname, notfallnr
```

```
FROM teilnehmer;
```

--Bedingung "WHERE"

--5. Zeige den vollständigen Namen aller weiblichen Teilnehmer:

```
SELECT nname, Vname
```

```
FROM teilnehmer
```

```
WHERE geschlecht = 'w';
```

--6. Zeige alle Teilnehmer mit dem Nachnamen Vogel:

```
SELECT *
```

```
FROM teilnehmer
```

```
WHERE nname = 'Vogel';
```

--7. Zeige alle Unternehmungen, die teurer als 100 € sind:

```
SELECT *
```

```
FROM unternehmung
```

```
WHERE kosten > 100;
```

--8. Zeige alle Unternehmungen der Fahrt Nr. 201701, die teurer als 100 € sind:

```
SELECT *
```

```
FROM unternehmung
```

```
WHERE kosten > 100 AND fahrtnr = 201701;
```

--Sortieren "ORDER BY"

--9. Gib die nach dem Nachnamen sortierte Liste aller Teilnehmer aus:

```
SELECT *
```

```
FROM teilnehmer
```

```
ORDER BY nname;
```

--10. Gib die nach Kosten absteigend sortierte Liste der Unternehmungen aus:

```
SELECT *  
FROM unternehmung  
ORDER BY kosten DESC;
```

--11. Zeige nur die teuerste Unternehmung:

```
SELECT *  
FROM unternehmung  
ORDER BY kosten DESC  
LIMIT 1;
```

--"NATURAL JOIN"

--12. Zeige alle Teilnehmer, die nach Dresden fahren:

```
SELECT *  
FROM faehrtmit NATURAL JOIN fahrt  
WHERE ziel = 'Dresden';
```

--13. Zeige die Nachnamen aller begleitenden Lehrer:

```
SELECT nname  
FROM begleiter NATURAL JOIN teilnehmer  
WHERE stand = 'Lehrer';
```

--14. Zeige alle Besonderheiten der Unternehmungen auf der Fahrt nach Orlando:

```
SELECT titel, ubesonderes  
FROM unternehmung NATURAL JOIN fahrt  
WHERE ziel = 'Orlando' AND ubesonderes <> '';
```

--15. Zeige alle Besonderheiten der teilnehmenden Schüler mit Vor- und Nachnamen der Schüler:

```
SELECT vname, nname, besonderes  
FROM teilnehmer NATURAL JOIN schueler;
```

--16. Zeige nur Schüler (mit Vor- und Nachnamen) mit ihren Besonderheiten, wenn sie welche haben:

```
SELECT vname, nname, besonderes  
FROM teilnehmer NATURAL JOIN schueler  
WHERE besonderes <> '';
```

--17. Zeige eine sortierte Liste aller Unternehmungen (Titel), die nicht bei der Fahrt nach Orlando stattfinden:

```
SELECT titel  
FROM unternehmung NATURAL JOIN fahrt  
WHERE ziel != 'Orlando'  
ORDER BY titel;
```

--18. Zeige alle Teilnehmernummern und vollständigen Namen der Teilnehmer, die nach Dresden fahren:

```
SELECT teilnnr, vname, nname  
FROM (teilnehmer NATURAL JOIN faehrtmit) NATURAL JOIN fahrt  
WHERE ziel = 'Dresden';
```

--19. Zeige alle Schüler (mit vollständigem Namen), die nach London fahren:

```
SELECT vname, nname  
FROM (teilnehmer NATURAL JOIN faehrtmit NATURAL JOIN fahrt NATURAL JOIN schueler)  
WHERE ziel = 'London';
```

--20. Zeige, welche Aktivitäten bei der Kursfahrt in die USA (nach Datum sortiert) stattfinden:

```
SELECT DISTINCT titel, udatum  
FROM unternehmung  
WHERE fahrtnr=201701  
ORDER BY udatum;
```

--21. Zeige die Teilnehmernummern und den Nachnamen der Schüler, die an mehreren Fahrten teilnehmen:

```
SELECT teilnnr, nname  
FROM faehrtmit NATURAL JOIN teilnehmer  
GROUP BY teilnnr, nname  
HAVING COUNT (*) > 1;
```

--"SUM"

--22. Zeige die Summe der Kosten der Unternehmungen der Londonfahrt (Fahrtnr=201703):

```
SELECT SUM(kosten) AS "Kosten insgesamt:"  
FROM unternehmung  
WHERE fahrtnr = 201703;
```

Textvorlage: Aufgaben zu Anfragen

Aufgabe 1): Partnerarbeit

Hinweise:

- Nutzt die einlaminieren Tabellen!
- Führt eure Anfragen schriftlich durch!
- Vergleicht eure Ergebnisse mit der Musterlösung (am Lehrerpult)

Notiere 4 Anfragen im Tupelkalkül und 4 Anfragen im Domänenkalkül. Suche dir 8 der folgenden Anfragen aus:

1. Zeige alle Fahrten.
2. Zeige alle teilnehmenden Personen.
3. Zeige alle Ziele der Fahrten.
4. Zeige die Nachnamen der Teilnehmer mit den entsprechenden Notfallnummern.

5. Zeige den vollständigen Namen aller weiblichen Teilnehmer.
6. Zeige alle Teilnehmer mit dem Nachnamen Vogel.
7. Zeige alle Unternehmungen, die teurer als 100 Euro sind.
8. Zeige alle Unternehmungen der Fahrt Nr. 201701, die teurer als 100 Euro sind.
9. Gib die nach dem Nachnamen sortierte Liste aller Teilnehmer aus.
10. Gib die nach Kosten absteigend sortierte Liste der Unternehmungen aus.
11. Zeige nur die teuerste Unternehmung.
12. Zeige alle Teilnehmer, die nach Dresden fahren.
13. Zeige die Nachnamen aller begleitenden Lehrer.
14. Zeige alle Besonderheiten der Unternehmungen auf der Fahrt nach Orlando.
15. Zeige alle Besonderheiten der teilnehmenden Schüler mit Vor- und Nachnamen der Schüler.
16. Zeige nur Schüler (mit Vor- und Nachnamen) und ihre Besonderheiten, wenn sie welche haben.
17. Zeige eine sortierte Liste aller Unternehmungen (Titel), die nicht bei der Fahrt nach Orlando stattfinden.
18. Zeige alle Teilnehmernummern und vollständigen Namen der Teilnehmer, die nach Dresden fahren.
19. Zeige alle Schüler (mit vollständigem Namen), die nach London fahren.
20. Zeige, welche Aktivitäten bei der Kursfahrt in die USA (nach Datum sortiert) stattfinden.
21. Zeige die Teilnehmernummern und den Nachnamen der Schüler, die an mehreren Fahrten teilnehmen.
22. Zeige die Summe der Kosten der Unternehmungen der Londonfahrt (Fahrtnr=201703).

Aufgabe 2): Partnerarbeit

Melde dich auf dem Raspberry Pi mit Putty an (IP-Adresse und Passwort an der Tafel). Gib dann psql ein.

- a) Durch \dt werden alle Tabellen angezeigt, die sich in der Datenbank befinden. Notiert euch zu jeder Tabelle die dazugehörigen Attribute, die Schlüssel sowie die Zusammenhänge zwischen den Tabellen.
- b) Wähle 10 Anfragen aus Aufg.1 aus und notiere sie in SQL-Syntax. Vergleiche eure Ergebnisse mit den Musterlösungen, die sich am Lehrerpult befinden.
- c) Überlegt euch noch eigene Anfragen und notiert die Ergebnisse.
- d) Trefft euch mit einem_r anderen Schüler_in und lasst diese_n jeweils zwei eurer Anfragen aus c) durchführen.

DES: Kursfahrten-Schema

fahrtenkonto (IBAN string, BIC string, inhaber string, bank string);
fahrt (fahrtnr integer, fahrtnr string, ziel string, von string, bis string, IBAN string);
teilnehmer (teilnnr integer, mobilnr string, geschlecht string, adresse string, notfallnr string, vname string, nname string, besonderes string);
begleiter (teilnnr integer, stand string);
schueler (teilnnr integer, erziehungsberechtigte string, gebdatum string);
faehrtnr (teilnnr integer, fahrtnr integer);
unternehmung (unr integer, titel string, kosten float, veranstalter string, ubesonderes string, udatum string, uhrzeit string, fahrtnr integer);
bezahlt (unr integer, IBAN string, bdatum string, bar string);
ueberwiesen (teilnnr integer, fahrtnr integer, fkbetrag float, fkdatum string);

DES: Daten zu den Relationen

(Die vollständigen Daten sind zu finden in: ../ralg/kursfahrten-Relationen/)

DES: Anfragen in Relationenalgebra

%Anfrage 1

%Gesucht sind alle bereits erfolgten Überweisungen für die Fahrt nach Dresden(Fahrtnr=201702).

%Relationenalgebra: $\sigma_{\text{fahrtnr}=201702}(\text{ueberwiesen})$

%Relationenalgebra in DES:

select fahrtnr=201702(ueberwiesen);

%Anfrage 2

%Gesucht sind alle weiblichen Teilnehmerinnen (Vor- und Nachname).

%Relationenalgebra: $\pi_{\text{vname}, \text{nname}}(\sigma_{\text{geschlecht}='w'}(\text{teilnehmer}))$

%Relationenalgebra in DES:

project vname,nname(select geschlecht='w'(teilnehmer));

%Anfrage 3

%Gesucht sind alle Unternehmungen (Titel, Kosten, Datum und Uhrzeit) auf der Fahrt nach London (Fahrtnummer: 201703).

%Relationenalgebra: $\pi_{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit}}(\sigma_{\text{fahrtnr}=201703}(\text{unternehmung}))$

%Relationenalgebra in DES:

project titel,kosten,udatum,uhrzeit (select fahrtnr=201703(unternehmung));

%Anfrage 4

%Welche Unternehmungen sind teurer als 100 €?

%Relationenalgebra: $\sigma_{\text{kosten}>100.00}(\text{unternehmung})$

%Relationenalgebra in DES:

select kosten>100.00(unternehmung);

%Anfrage 5

%Gesucht sind alle Teilnehmer (Vor-, Nachname) der Fahrt nach Orlando.

%Relationenalgebra: $\pi_{\text{vname}, \text{nname}}(\sigma_{\text{ziel}='Orlando'}(\text{fahrt}) \bowtie \text{faehrtmit} \bowtie \text{teilnehmer})$

%Relationenalgebra in DES:

project vname,nname(select ziel='Orlando' (fahrt) njoin faehrtmit njoin teilnehmer);

DES: Anfragen im Tupelkalkül

%Anfrage t1

%Gesucht sind alle bereits erfolgten Überweisungen für die Fahrt nach Dresden(FahrtNr=201702).

%Relationenalgebra: $\sigma_{\text{fahrtnr}=201702}(\text{ueberwiesen})$

%TRC: $\{x | (\text{ueberwiesen}(x) \wedge x[\text{fahrtnr}]=201702)\}$

%TRC in DES:

t1 := $\{X | (\text{ueberwiesen}(X) \text{ and } X.\text{fahrtnr}=201702)\}$;

%Anfrage t2

%Gesucht sind alle Teilnehmer der Fahrt nach Orlando (fahrtnr=201701)

%TRC: $\{x | \text{faehrtmit}(x) \wedge x[\text{fahrtnr}]=201701\}$

%TRC in DES:

t2 := $\{X | \text{faehrtmit}(X) \text{ and } X.\text{fahrtnr}=201701\}$;

%Anfrage t3

%Gesucht sind alle Unternehmungen (Titel, Kosten, Datum, Uhrzeit und Fahrtnummer) auf der Fahrt nach London (Fahrtnummer: 201703).

%Relationenalgebra: $\pi_{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit}, \text{fahrtnr}}(\sigma_{\text{fahrtnr}=201703}(\text{Unternehmung}))$

%TRC: $\{x | (\text{unternehmung}(x) \wedge x[\text{fahrtnr}]=201703)\}$

%TRC in DES:

t3 := $\{X.\text{titel}, X.\text{kosten}, X.\text{udatum}, X.\text{uhrzeit}, X.\text{fahrtnr} | (\text{unternehmung}(X) \text{ and } X.\text{fahrtnr}=201703)\}$;

%Anfrage t4

%Welche Unternehmungen der Fahrt nach Dresden (Fahrtnr=201702) sind teurer als 150€?

%Relationenalgebra: $\sigma_{\text{fahrtnr}=201702 \wedge \text{kosten} > 150.00}(\text{Unternehmung})$

%TRC: $\{x | (\text{unternehmung}(x) \wedge x[\text{fahrtnr}]=201702 \wedge x[\text{kosten}] > 150.00)\}$

%TRC in DES:

t4 := $\{X | (\text{unternehmung}(X) \text{ and } X.\text{fahrtnr}=201702 \text{ and } X.\text{kosten} > 150.00)\}$;

%Anfrage t5

%Gesucht sind alle bar bezahlten Unternehmungen.

%Relationenalgebra: $\sigma_{\text{bar}='True'}(\text{Bezahl})$

%TRC: $\{x | (\text{bezahlt}(x) \wedge x[\text{bar}]='True')\}$

%TRC in DES:

t5 := $\{X | \text{bezahlt}(X) \text{ and } X.\text{bar}='True'\}$;

DES: Antworten zu den Anfragen im Tupelkalkül

(Die vollständige Liste ist zu finden in: ../ralg/kursfahrten-query.ans)

%Anfrage 1:

```
answer(6,201702,600.0,'2017-01-17'),  
answer(8,201702,600.0,'2017-01-17'),  
answer(9,201702,600.0,'2017-01-17'),  
answer(10,201702,600.0,'2017-01-17'),  
[...]
```

Info: 11 tuples computed

%Anfrage 2:

```
answer('Angelika','Fuhrmann'),  
answer('Christina','Eichelberger'),  
answer('Christine','Vogel'),  
answer('Claudia','Nadel'),  
[...]
```

Info: 15 tuples computed.

%Anfrage 3:

```
answer('4 uebernachtungen',4200.0,'2017-06-08','8:00:00'),  
answer('Abendessen',0.0,'2017-06-10','18:00:00'),  
answer('Abendessen',0.0,'2017-06-11','18:00:00'),  
answer('Abendessen',70.0,'2017-06-09','18:00:00'),  
[...]
```

Info: 20 tuples computed.

%Anfrage 4:

```
answer(1,'Anreise Bus',1400.0,'Bussard-Travel','10 Std. Nachtfahrt','2017-06-  
07','18:00:00',201703),  
answer(5,'Abendessen',300.0,'Hangmans Soup Corner','', '2017-06-08','18:00:00',201703),  
answer(8,'Shakespeare Workshop',150.0,'Globe Theater','', '2017-06-09','16:00:00',201703),  
answer(17,'St. Paul Cathedral',170.0,'Monastery of the cross','', '2017-06-11','14:00:00',201703),  
[...]
```

Info: 16 tuples computed.

%Anfrage 5:

```
answer('Christina','Eichelberger'),  
answer('Christine','Vogel'),  
answer('Claudia','Nadel'),  
answer('Diana','Wannemaker'),  
[...]
```

Info: 9 tuples computed.

DES: Anfragen im Domänenkalkül

%Anfrage d1

%Gesucht sind alle bereits erfolgten Überweisungen für die Fahrt nach Dresden(FahrtNr=201702).

%Relationenalgebra: $\sigma_{\text{fahrtnr}=201702}(\text{ueberwiesen})$

%DRC: $\{\text{teilnnr}, \text{fkbetrag}, \text{fkdatum} \mid (\exists \text{ fahrtnr})(\text{ueberwiesen}(\text{teilnnr}, \text{fahrtnr}, \text{fkbetrag}, \text{fkdatum}) \wedge \text{fahrtnr}=201702)\}$

%DRC in DES:

d1 := $\{\text{teilnnr}, \text{fkbetrag}, \text{fkdatum} \mid (\text{exists fahrtnr})(\text{ueberwiesen}(\text{teilnnr}, \text{fahrtnr}, \text{fkbetrag}, \text{fkdatum}) \text{ and fahrtnr}=201702)\}$;

%Anfrage d2

%Gesucht sind alle weiblichen Teilnehmerinnen (Vor- und Nachname).

%Relationenalgebra: $\pi_{\text{vname}, \text{nname}}(\sigma_{\text{geschlecht}='w'}(\text{teilnehmer}))$

%DRC: $\{\text{vname}, \text{nname} \mid (\exists \text{ teilnnr})(\exists \text{ mobilnr})(\exists \text{ geschlecht})(\exists \text{ adresse})(\exists \text{ notfallnr})(\exists \text{ besonderes})(\text{teilnehmer}(\text{teilnnr}, \text{mobilnr}, \text{geschlecht}, \text{adresse}, \text{notfallnr}, \text{vname}, \text{nname}, \text{besonderes}) \wedge \text{geschlecht}='w')\}$

%DRC in DES:

d2 := $\{\text{vname}, \text{nname} \mid (\text{exists teilnnr}, \text{mobilnr}, \text{geschlecht}, \text{adresse}, \text{notfallnr}, \text{besonderes})(\text{teilnehmer}(\text{teilnnr}, \text{mobilnr}, \text{geschlecht}, \text{adresse}, \text{notfallnr}, \text{vname}, \text{nname}, \text{besonderes}) \text{ and geschlecht}='w')\}$;

%Anfrage d3

%Gesucht sind alle Unternehmungen (Titel, Kosten, Datum und Uhrzeit) auf der Fahrt nach London (Fahrtnummer: 201703).

%Relationenalgebra: $\pi_{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit}}(\sigma_{\text{fahrtnr}=201703}(\text{unternehmung}))$

%DRC: $\{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit} \mid (\exists \text{ unr})(\exists \text{ veranstalter})(\exists \text{ ubesonderes})(\exists \text{ fahrtnr})(\text{unternehmung}(\text{unr}, \text{titel}, \text{kosten}, \text{veranstalter}, \text{ubesonderes}, \text{udatum}, \text{uhrzeit}, \text{fahrtnr}) \wedge \text{fahrtnr}=201703)\}$

%DRC in DES:

d3 := $\{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit} \mid (\text{exists unr}, \text{veranstalter}, \text{ubesonderes}, \text{fahrtnr})(\text{unternehmung}(\text{unr}, \text{titel}, \text{kosten}, \text{veranstalter}, \text{ubesonderes}, \text{udatum}, \text{uhrzeit}, \text{fahrtnr}) \text{ and fahrtnr}=201703)\}$;

%Anfrage d4

%Welche dieser Unternehmungen (Ergebnis der Anfrage d3) sind teurer als 100 €?

%Relationenalgebra: $\sigma_{\text{kosten}>100.00}(\text{d3})$

%DRC: $\{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit} \mid (\text{d3}(\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit}) \wedge \text{kosten}>100.00)\}$

%DRC in DES:

d3 := $\{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit} \mid (\text{exists unr}, \text{veranstalter}, \text{ubesonderes}, \text{fahrtnr})(\text{unternehmung}(\text{unr}, \text{titel}, \text{kosten}, \text{veranstalter}, \text{ubesonderes}, \text{udatum}, \text{uhrzeit}, \text{fahrtnr}) \text{ and fahrtnr}=201703)\}$;

d4 := $\{\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit} \mid (\text{d3}(\text{titel}, \text{kosten}, \text{udatum}, \text{uhrzeit}) \text{ and kosten}>100.00)\}$;

%Anfrage d5

%Gesucht sind alle Teilnehmer (Vor-, Nachname) der Fahrt nach Orlando.


```
%Relationenalgebra:  $\pi_{vname,nname}(\sigma_{ziel='Orlando'}(Fahrt) \bowtie F\ddot{a}hrtmit \bowtie Teilnehmer)$ 
%DRC:      {vname,nname|( $\exists teilnnr$ )( $\exists mobilnr$ )( $\exists geschlecht$ )( $\exists adresse$ )( $\exists notfallnr$ )( $\exists besonderes$ )
( $\exists fahrtnr$ )( $\exists fahrtname$ )( $\exists ziel$ )( $\exists von$ )( $\exists bis$ )( $\exists iban$ )(fahrt(fahrtnr,fahrtname,ziel,von,bis,iban)       $\wedge$ 
ziel='Orlando'  $\wedge$  faehrtmit(teilnnr,fahrtnr)  $\wedge$  teilnehmer (teilnnr, mobilnr, geschlecht, adresse,
notfallnr, vname, nname, besonderes))}
```

%DRC in DES:

```
d5 := {vname,nname | (exists teilnnr, mobilnr, geschlecht, adresse, notfallnr, besonderes, fahrtnr,
fahrtname, ziel, von, bis, iban) (fahrt (fahrtnr, fahrtname, ziel, von, bis, iban) and ziel = 'Orlando'
and faehrtmit (teilnnr, fahrtnr) and teilnehmer (teilnnr, mobilnr, geschlecht, adresse, notfallnr, vna-
me, nname, besonderes)))};
```

Beschreibung und Motivation des View

Ein View ermöglicht dem Anwender eine Eingabe, ohne Kenntnisse von SQL zu haben. Die Schüler müssen sich in die Lage eines Anwenders hinein versetzen, um eine möglichst einfache und intuitive Oberfläche zu nutzen. Hierzu dient der eigens erstellte Layout-ADT, welcher den Schülern zur Verfügung gestellt wird.

Des Weiteren wird ein ADO für die Erstellung von Buttons zur Verfügung gestellt. Die Schüler sollte eine gute Programmiererfahrung haben und die Funktionen eines Grafikpaketes beherrschen.

Basismöglichkeiten und Erweiterung des Views

Die einfachste Anwendung ist die Anfrage von Ergebnissen. Hierfür sind in dem beigegeführten Arbeitsauftrag zwei Aufgaben. Es öffnet sich nur ein Fenster, in denen Anfragen eingegeben und ausgegeben werden.

Schwieriger ist die Eingabe von Daten in die Datenbank, da hier Vorprüfungen erfolgen müssen.

Des Weiteren lässt sich der View dadurch erweitern, dass mehrere Anfragen und Eingaben möglich sind, die vom Nutzer per Klick ausgewählt werden. Siehe Aufgabe 3 des folgenden Arbeitsauftrages.

Textvorlage: Aufgaben bzgl. des Views

Aufg. 1

Szenario:

Der Schulleiter möchte von seinem Computer aus abfragen können, welche Schüler oder Schülerinnen bei einer bestimmten Fahrt mitfahren.

a) Entwirf ein Programm, das die Eingabe einer Fahrtennummer erwartet. Die Ausgabe soll Folgendes beinhalten:

Vorname, Nachname, Geschlecht, Adresse, Mobilnummer, Notfallnummer, Besonderes
!! Beachte den Sonderfall "Besonderes": soll nur dann erscheinen, wenn etwas eingetragen wurde. !!

b) Die Abfrage soll eine eigene Funktion sein und in der Main-Funktion aufgerufen wird.

Verwende die Pakete, die als Hilfsmittel angegeben sind.

Aufg. 2

Der Schulleiter möchte auch die Möglichkeit haben, einen bestimmten Schüler zu suchen. Folgendes soll angezeigt werden:

- a) Wenn der Schüler an einer Fahrt teilnimmt, dann soll die Fahrt mit den Informationen Fahrtnummer, Fahrtname, Ziel, Von, Bis, IBAN angezeigt werden.
- b) Wenn der Schüler nicht (!) an einer Fahrt teilnimmt, erscheint: „Der Schüler nimmt an keiner Fahrt teil.“

+++++++ ZUSATZ - AUFGABE ++++++

Aufg. 3

- a) Erstelle die Möglichkeit, dass der Schulleiter Schüler zu einer bestehenden Fahrt hinzufügen kann.
- b) Füge die 2 Anfragen und die Eingabe zu einer GUI zusammen.

Hilfsmittel:

- gfx Spezifikation (bekannt)
- SQL Spezifikation (bekannt)
- Formulare Spezifikation (bekannt)
- eckigebutton Spezifikation (siehe Anhang)
- Layout Spezifikation (siehe Anhang)