	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

1 Relationenalgebra

[8 P.]

Gegeben seien die folgenden Relationenschemata:

Person(PNR, Vorname, Nachname, Geburtsdatum, Wohnort → Ort.ONR)

Jugendherberge(JNR, Name, Ort → Ort.ONR, Manager → Person.PNR)

Ort(ONR, Ortsname, Buergermeister → Person.PNR)

Aufenthalt(Gast → Person.PNR, Jugendherberge → Jugendherberge.JNR, Preis)

Benutzen Sie zur Lösung der folgenden Aufgaben ausschließlich die in der Vorlesung vorgestellten Operatoren der Relationenalgebra!

- a) Geben Sie einen Relationenalgebra-Ausdruck an, der zu dem unten angegebenen SQL-Ausdruck äquivalent ist. [2 P.]

```
SELECT DISTINCT Preis, Name
FROM Person, Aufenthalt, Jugendherberge
WHERE PNR = Gast
AND Jugendherberge = JNR
AND Geburtsdatum <= '1919-03-24'
```

Lösungsvorschlag:

$$\pi_{\text{Preis, Name}} \left(\left(\sigma_{\text{Geburtsdatum} \leq \text{'1919-03-24'}} \left(\text{Person} \right) \right) \bowtie_{\text{PNR=Gast}} \left(\text{Aufenthalt} \right) \right) \bowtie_{\text{Jugendherberge=JNR}} \left(\text{Jugendherberge} \right)$$

- b) Geben Sie einen Relationenalgebra-Ausdruck an, der die Namen aller Jugendherbergen ausgibt, in denen sich schon mal ein Buergermeister aufgehalten hat. [2 P.]

Lösungsvorschlag:


$$\pi_{\text{Name}} \left(\left(\text{Aufenthalt} \right) \bowtie_{\text{Jugendherberge=JNR}} \left(\text{Jugendherberge} \right) \right) \bowtie_{\text{Gast=Buergermeister}} \left(\text{Ort} \right)$$

- c) Geben Sie einen Relationenalgebra-Ausdruck an, der die Vor- und Nachnamen aller Personen ausgibt, die keine Buergermeister sind. [2 P.]

Lösungsvorschlag:

$$\pi_{\text{Vorname, Nachname}} \left(\text{Person} \bowtie \left(\pi_{\text{PNR}} \left(\text{Person} \right) - \rho_{\text{PNR} \leftarrow \text{Buergermeister}} \left(\pi_{\text{Buergermeister}} \left(\text{Ort} \right) \right) \right) \right)$$

oder

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018


$\pi_{\text{Vorname, Nachname}}(Person \bowtie (\pi_{PNR}(Person) - \pi_{Buergermeister}(Ort)))$

- d) Geben Sie eine natürlichsprachliche Beschreibung der Ergebnismenge des folgenden Relationenalgebra-Ausdrucks an. [2 P.]

$\pi_{\text{Ortsname}}(Ort \underset{ONR=Ort}{\bowtie} Jugendherberge \bowtie (\rho_{\text{Manager} \leftarrow PNR}(\pi_{PNR}(\sigma_{\text{Vorname}="Tim"}(Person))))))$

Lösungsvorschlag:

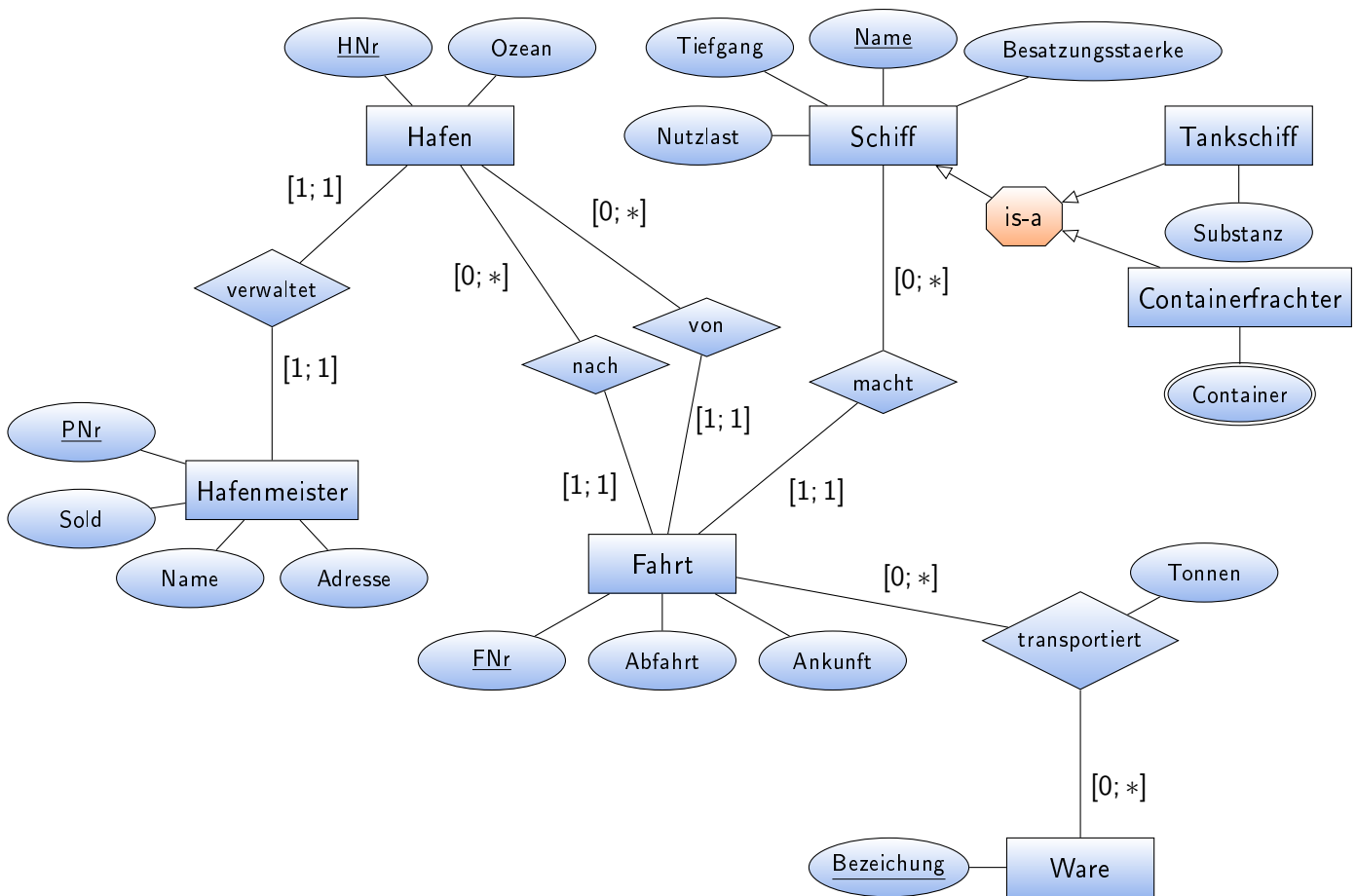
Die Ortsnamen aller Orte mit einer Jugendherberge, deren Manager den Vornamen „Tim“ hat.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

2 Schemadefinition

[12 P.]


Gegeben sei folgendes Entity-Relationship-Diagramm:



Um die Konsistenz der Daten sicherzustellen, sollen folgende Integritätsbedingungen gelten:

- IB1:** Die Kombination aus Name und Adresse eines Hafenmeisters ist eindeutig.
- IB2:** Die Besatzungsstaerke eines Schiffes ist optional. Alle anderen Attribute sind verpflichtend anzugeben.
- IB3:** Das Abfahrtsdatum einer Fahrt muss (sofern angegeben) kleiner als das aktuelle Datum (CURRENT_DATE) sein.
- IB4:** Der Sold eines Hafenmeisters muss größer als 50000 sein.

Geben Sie die SQL-DDL-Anweisungen an, die notwendig sind, um das DB-Schema zu erstellen. Wählen Sie dabei geeignete SQL-Standard-Datentypen. Verwenden Sie **vertikale Partitionierung**, um evtl. Vererbungen abzubilden. Beachten Sie, dass die Kardinalitätsrestriktionen durch geeignete Constraints exakt abzubilden sind.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

Weiterhin ist bei 1:1-Beziehungen die Symmetrie sicherzustellen (Tipp: Fremdschlüssel in beiden Relationen). Testen Sie die SQL-Ausdrücke auf der Übungsdatenbank.

Lösungsvorschlag:

```

CREATE TABLE Hafen(
  HNr          int          PRIMARY KEY NOT NULL,
  Ozean        varchar(50)  NOT NULL
);

CREATE TABLE Hafenmeister(
  PNr          int          PRIMARY KEY NOT NULL,
  Sold         int          NOT NULL CHECK(Sold > 50000),
  Name         varchar(50)  NOT NULL,
  Adresse      varchar(50)  NOT NULL,
  Hafen        int          UNIQUE NOT NULL,
  CONSTRAINT fk_hafenmeister_hafen FOREIGN KEY (Hafen) REFERENCES Hafen(HNr),
  CONSTRAINT unique_hafenmeister_name_adresse UNIQUE (Name,Adresse)
);

ALTER TABLE Hafen
ADD CONSTRAINT fk_hafen_hafenmeister FOREIGN KEY (HNr) REFERENCES Hafenmeister(Hafen);

CREATE TABLE Schiff(
  Name         varchar(12)  PRIMARY KEY NOT NULL,
  Tiefgang     int          NOT NULL,
  Nutzlast     int          NOT NULL,
  Besatzungsstaerke int
);

CREATE TABLE Tankschiff(
  Name         varchar(12)  PRIMARY KEY NOT NULL,
  Substanz     varchar(50)  NOT NULL,
  CONSTRAINT fk_tankschiff_name FOREIGN KEY (Name) REFERENCES Schiff(Name)
);

CREATE TABLE Containerfrachter(
  Name         varchar(12)  PRIMARY KEY NOT NULL,
  CONSTRAINT fk_containerfrachter_name FOREIGN KEY (Name) REFERENCES Schiff(Name)
);

```



Lehrveranstaltung

Grundlagen von Datenbanken

WS 2018/19

Aufgabenzettel

4 (Lösungsvorschläge)

Gesamtpunktzahl

40

```
CREATE TABLE Container(  
  Frachter          varchar(12)  NOT NULL,  
  Container         int          NOT NULL,  
  CONSTRAINT pk_container PRIMARY KEY (Frachter, Container),  
  CONSTRAINT fk_container_frachter FOREIGN KEY (Frachter) REFERENCES Containerfrachter(Name)  
);  
  
CREATE TABLE Ware(  
  Bezeichnung      varchar(50)  PRIMARY KEY NOT NULL  
);  
  
CREATE TABLE Fahrt(  
  FNr              int          PRIMARY KEY NOT NULL,  
  Schiff           varchar(12)  NOT NULL,  
  Abfahrt          date        NOT NULL CHECK(Abfahrt < CURRENT_DATE),  
  Ankunft          date        NOT NULL,  
  nach             int          NOT NULL,  
  von              int          NOT NULL,  
  CONSTRAINT fk_fahrt_schiff FOREIGN KEY (Schiff) REFERENCES Schiff(Name),  
  CONSTRAINT fk_fahrt_nach FOREIGN KEY (nach) REFERENCES Hafen(HNr),  
  CONSTRAINT fk_fahrt_von FOREIGN KEY (von) REFERENCES Hafen(HNr)  
);  
  
CREATE TABLE transportiert(  
  Fahrt           int          NOT NULL,  
  Ware            varchar(50)  NOT NULL,  
  Tonnen          int          NOT NULL,  
  CONSTRAINT pk_transportiert PRIMARY KEY (Fahrt, Ware),  
  CONSTRAINT fk_transportiert_fahrt FOREIGN KEY (Fahrt) REFERENCES Fahrt(FNr),  
  CONSTRAINT fk_transportiert_ware FOREIGN KEY (Ware) REFERENCES Ware(Bezeichnung)  
);
```


3 SQL

[10 P.]

Gegeben seien die aus Aufgabe 1 bekannten Relationenschemata:

Person(PNR, Vorname, Nachname, Geburtsdatum, Wohnort → Ort.ONR)

Jugendherberge(JNR, Name, Ort → Ort.ONR, Manager → Person.PNR)

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

Ort(ONR, Ortsname, Buergermeister → Person.PNR)

Aufenthalt(Gast → Person.PNR, Jugendherberge → Jugendherberge.JNR, Preis)

Formulieren Sie entsprechende SQL-Anweisungen für die in den nachfolgenden Teilaufgaben angeführten natürlichsprachlich formulierten Mengenbeschreibungen. **Verwenden Sie den in der Vorlesung verwendeten SQL-Standard.** Das SQL-Schlüsselwort JOIN darf dabei nicht verwendet werden.

- a) Die Vornamen und Nachnamen aller Buergermeister (ohne Duplikate in absteigender Reihenfolge nach Nachname sortiert), die Manager einer Jugendherberge sind, in der sie bereits einen Aufenthalt als Gast hatten. [2 P.]

Lösungsvorschlag:


```
SELECT DISTINCT p.Vorname, p.Nachname
FROM Ort o, Person p, Aufenthalt a, Jugendherberge j
WHERE o.Buergermeister = p.PNR
AND p.PNR = a.Gast
AND a.Jugendherberge = j.JNR
AND j.Manager = p.PNR
ORDER BY p.Nachname DESC
```

- b) Die PNRs und die Nachnamen aller Manager sowie die Anzahl der von ihnen jeweils geleiteten Jugendherbergen. [2 P.]

Lösungsvorschlag:

```
SELECT j.Manager, p.Nachname, COUNT(*) AS AnzahlJugendherbergen
FROM Jugendherberge j, Person p
WHERE j.Manager = p.PNR
GROUP BY j.Manager, p.Nachname
```

Gruppierung allein nach j.Manager oder nach p.PNR ist auch akzeptabel.

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

c) Die PNR aller Manager, die weniger als 5 Jugendherbergen leiten.

[2 P.]

Lösungsvorschlag:

```
SELECT Manager
FROM Jugendherberge
GROUP BY Manager
HAVING COUNT(*) < 5
```

d) Die Vornamen aller Personen (ohne Duplikate), die denselben Nachnamen haben wie der Manager einer von ihnen besuchten Jugendherberge, jedoch nicht selbst dieser Manager sind.

[2 P.]

Lösungsvorschlag:

```
SELECT DISTINCT gast.Vorname
FROM Aufenthalt a, Person gast, Person manager, Jugendherberge j
WHERE gast.PNR = a.Gast
AND a.Jugendherberge = j.JNR
AND j.Manager = manager.PNR
AND gast.Nachname = manager.Nachname
AND gast.PNR <> manager.PNR
```

e) PNRs, Vornamen und Nachnamen aller Personen, die keine Manager sind.


[2 P.]

Lösungsvorschlag:

```
SELECT p.PNR, p.Vorname, p.Nachname
FROM Person p
WHERE NOT EXISTS(
  SELECT *
  FROM Jugendherberge j
  WHERE p.PNR = j.Manager
)
```

oder

```
SELECT p.PNR, p.Vorname, p.Nachname
FROM Person p
WHERE p.PNR NOT IN(
  SELECT Manager
```

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

```
FROM Jugendherberge
)
```

4 Optimierung

[10 P.]

Gegeben seien die aus Aufgabe 1 bekannten Relationenschemata:

Person(PNR, Vorname, Nachname, Geburtsdatum, Wohnort → *Ort.ONR*)

Jugendherberge(JNR, Name, Ort → *Ort.ONR*, Manager → *Person.PNR*)

Ort(ONR, Ortsname, Buergermeister → *Person.PNR*)


Aufenthalt(Gast → *Person.PNR*, *Jugendherberge* → *Jugendherberge.JNR*, Preis)

- a) Für die nachfolgende Anfrage soll eine algebraische Optimierung durchgeführt werden. Zeichnen Sie dafür als erstes den entsprechenden Operatorbaum für die vorgegebene Anfrage und optimieren Sie diesen anschließend anhand der in der Vorlesung eingeführten Regeln. (Projektionen sollen dabei jedoch **nicht** nach unten gezogen werden.) [7 P.]
- b) Bewerten Sie die Operatorbäume mit den Kardinalitäten der Zwischenergebnisse. [3 P.]

Für die zugehörige Datenbank werden folgende Kardinalitäten angenommen:

$Card(Person) = 100$, $Card(Ort) = 50$ und $Card(Aufenthalt) = 400$.

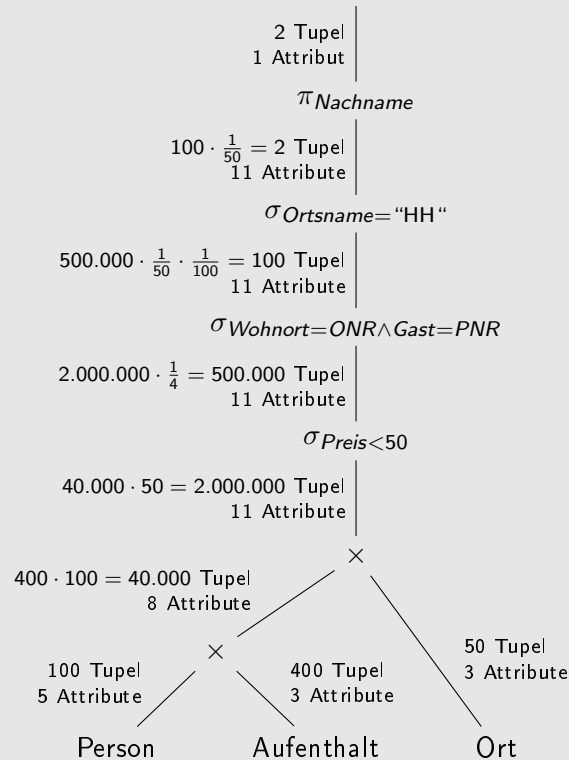
Jeder Ort hat einen eindeutigen Namen, zudem liegt der Preis eines Aufenthaltes immer zwischen 40 und 80 Euro (Nehmen sie eine Gleichverteilung an).

	Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
	Aufgabenzettel	4 (Lösungsvorschläge)		
	Gesamtpunktzahl	40		
	Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

$\pi_{\text{Nachname}}(\sigma_{\text{Ortsname}=\text{"Hamburg"}}(\sigma_{\text{Wohnort}=\text{ONR} \wedge \text{Gast}=\text{PNR}}(\sigma_{\text{Preis}<50}((\text{Person} \times \text{Aufenthalt}) \times \text{Ort))))$

Lösungsvorschlag:

nicht optimiert:



optimiert:

$\pi_{\text{Nachname}}(\sigma_{\text{Preis}<50}(\text{Aufenthalt}) \bowtie_{\text{Gast}=\text{PNR}} (\text{Person} \bowtie_{\text{Wohnort}=\text{ONR}} \sigma_{\text{Ortsname}=\text{"Hamburg"}}(\text{Ort))))$



Lehrveranstaltung	Grundlagen von Datenbanken		WS 2018/19
Aufgabenzettel	4 (Lösungsvorschläge)		
Gesamtpunktzahl	40		
Ausgabe	Mi. 28.11.2018	Abgabe	Fr. 14.12.2018

