

(DB 21)

1a)

• $NULL \text{ IS } NULL \text{ AND } 0 \leq NULL \Rightarrow u$

• $NOT(1 > 2) \text{ OR } NULL = 3 \Rightarrow T$

T: True, L: False, u: unknown

b)

$R = R \bowtie R$ für jede Relation R?

Nein. Das Problem sind NULL Werte

z.B.

R	
A	B
-	-

\neq

R \bowtie R	
A	B

Relation ohne Tupel

Relation mit
einem Tupel

(NULL, NULL)

c)

R		
A	B	C
1	2	0
2	2	0
3	2	6
4	-	4

$\gamma(R, B, \text{avg}, C)$	
B	avg(C)
2	3
-	4

γ arbeitet mit Mengen statt

Multimengen $\text{avg}\{0, 0, 6\} = \text{avg}\{0, 6\}$
 $= 3$

d) $\{B \rightarrow AC, D \rightarrow AB, A \rightarrow D\} \vdash A \rightarrow C$?

L: Ja, denn $A \rightarrow D \rightarrow AB \rightarrow B \rightarrow AC \rightarrow C$

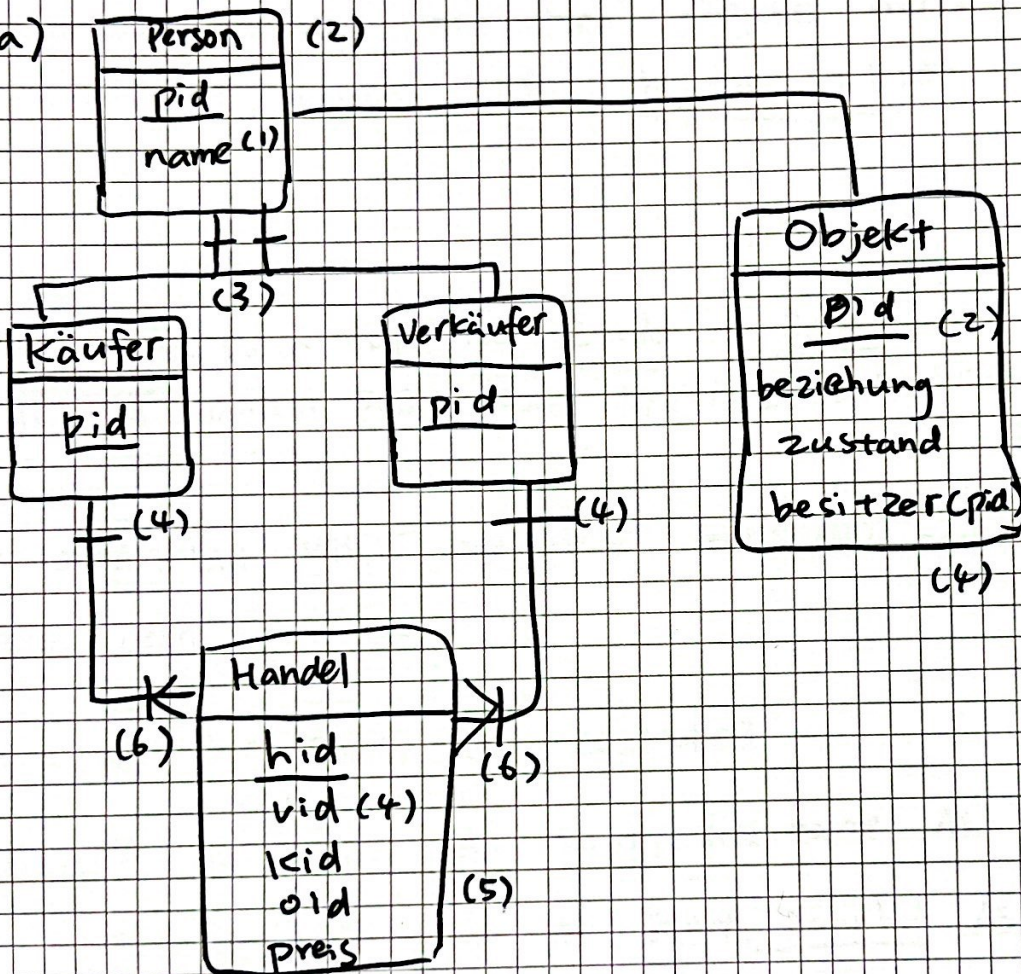
-|-

e) (L. lid, R. rid)

Beachte: Ein Schlüssel kann auch aus mehreren Attributen bestehen.

2 II. Modellierung

a)



* keine Angabe
also
beliebig

* Nummer
repräsentativ

* Primärschlüssel müssen eingefügt werden

z.B. oid bei Objekt. Zur Kennzeichnung unterstreichen!

Unter der Annahme, dass jemand nie dasselbe Objekt an dieselbe Person verkauft, ist auch (vid, lcid, oid) ein Schlüssel für Handel.

b) Verkäufer darf \neq gleich Käufer sein ($vid \neq kid$)

- oid muss als Objekt existieren.

vid und kid müssen als Person existieren.

vid muss entweder bereits in verkäufer existieren oder in Verkäufer eingefügt werden (kid analog)

- Der Besitzer des verkauften Objekts muss aktualisiert werden. (Set besitzer = kid)

3. SQL

3a An welchen Daten für Peer Zauder \rightarrow Bern?

```
SELECT datum FROM Kunde NATURAL JOIN Reise INNER JOIN
    Bahnhof
ON zbid = bid WHERE vorname = 'Peer' AND nachname = 'Zauder'
AND 'Bern'
```

(b) Reisen von Peer Zauder am 20210603.

```
SELECT start.name AS start, ziel.name AS ziel FROM
    Kunde, Reise, Bahnhof AS start, Bahnhof AS ziel WHERE
    Kunde.pid = Reise.pid AND sbid = start.bid AND zbid = ziel.bid
AND vorname = 'Peer' AND nachname = 'Zauder' AND datum =
    20210603.
```


(3c) INSERT INTO Kunde (vorname, nachname) VALUES ('Peter', 'Müller');
pid ist von Typ SERIAL, für aid ist per default NULL.

(3d) Anzahl Kunden ohne Abo.

```
SELECT count(*) FROM Kunde WHERE aid IS NULL;
```

(3e) Zu jedem Abo die Anzahl Kunden, die dieses besitzen.

```
SELECT name, count(pid) FROM Kunde NATURAL JOIN Abo  
GROUP BY aid;
```

(3f) Ausgaben pro Kunde

WITH Abokosten AS (

```
SELECT pid, vorname, nachname, CASE WHEN preis  
IS NULL THEN 0 ELSE preis END AS abokosten  
FROM Kunde LEFT JOIN Abo  
)
```

Reisekosten AS (

```
SELECT pid, CASE WHEN sum(preis) IS NULL THEN  
0 ELSE sum(preis) END AS reisekosten  
FROM Kunde LEFT JOIN Reise GROUP BY pid
```

)

```
SELECT vorname, nachname, abokosten + reisekosten AS kosten  
FROM Abokosten NATURAL JOIN Reisekosten;
```


4.a $\pi_{A, R.B} (\sigma_{L.B = R.B \wedge C = 4} (L \times R))$

4.b 1. Selektionsprädikate zerlegen

$$\pi_{A, R.B} (\sigma_{L.B = R.B} (\sigma_{C = 4} (L \times R)))$$

2. Selektionen nach innen schieben

$$\pi_{A, R.B} (\sigma_{L.B = R.B} (L \times \sigma_{C = 4} (R)))$$

3. \bowtie -Joins

$$\pi_{A, R.B} (L \bowtie_{L.B = R.B} \sigma_{C = 4} (R))$$

4. Reihenfolge der Joins

nur 1 Join, keine andere Reihenfolge

5. Projektionen nach innen / zusätzlich

$$\pi_{A, R.B} (L \bowtie_{L.B = R.B} \pi_B (\sigma_{C = 4} (R)))$$

4c

$$\pi_{A, R.B} (\sigma_{L.B = R.B, C = 4} (\underbrace{L}_{2 \times 100} \times \underbrace{R}_{2 \times 100}))$$

$$\underbrace{\hspace{10em}}_{4 \times 10}$$

$$\underbrace{\hspace{10em}}_{2 \times 10}$$

$$\pi_{A, R.B} (L \bowtie_{L.B = R.B} \pi_B (\sigma_{C = 4} (R)))$$

$$\underbrace{\underbrace{\pi_B (\sigma_{C = 4} (R))}_{2 \times 100}}_{2 \times 10}$$

$$7 \times 10$$

$$\underbrace{\underbrace{2 \times 10}_{-5-} 3 \times 10}_{7 \times 10}$$