

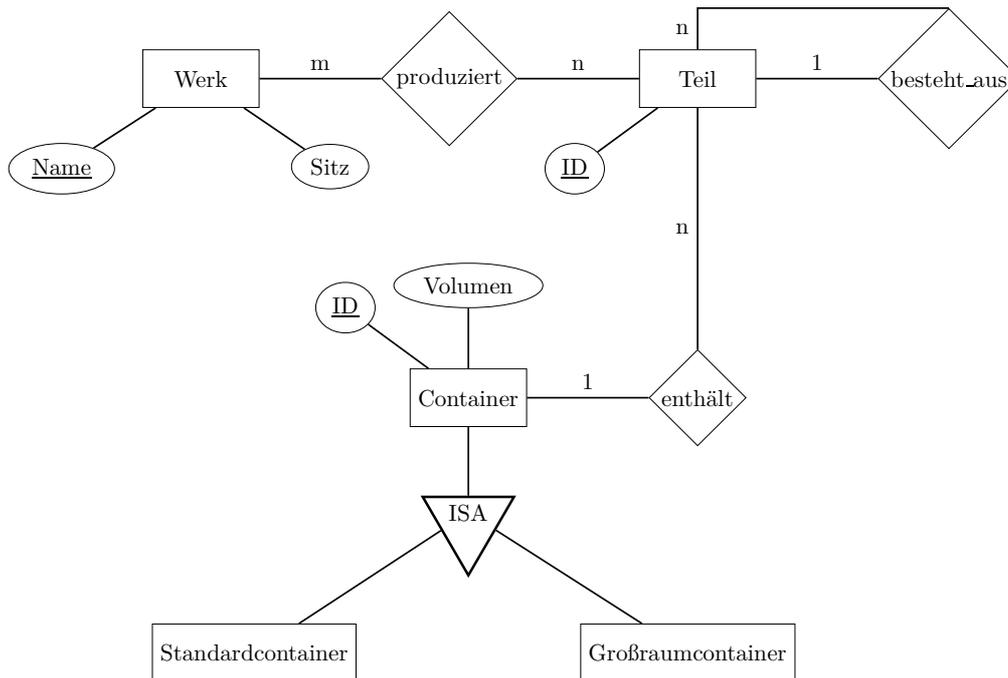


**FernUniversität in Hagen**

**Lösungsvorschläge  
zur Teilklausur  
01671 „Datenbanken I“**

**21.09.2022**

## Aufgabe 1 Entity-Relationship-Diagramm



Punkte: is-a-Relationship 2 Punkte, jeweils 1 Punkt für einen Entity- bzw. Beziehungstyp und Kardinalität (insg. 11 Punkte). Pro Schlüsselattribut wird ein 1 Punkt vergeben (insg. 3 Punkte), pro Nichtschlüsselattribut wird ein halber Punkt vergeben (insg. 1 Punkt).

## Aufgabe 2 Relationenalgebra

(a)

Die Vereinigungsoperation fasst Tupel aus zwei Relationen in einer neuen Relation zusammen. Aufgrund der Definition der Relationen als Mengen ist jedes Tupel nur einmal vorhanden. Für diese Operation müssen die Relationen vereinigungsverträglich sein (sie müssen die gleichen Attribute bzgl. Namen und Wertebereichen haben). Die gegebenen Relationen *GRUPPE\_1* und *GRUPPE\_2* sind nicht vereinigungsverträglich, da einige Attribute unterschiedliche Bezeichnungen haben (*Preis* und *Kaufpreis*). Mithilfe der Umbenennungsoperation kann die Vereinigungsverträglichkeit erreicht werden, indem z. B. *Kaufpreis* in *Preis* umbenannt wird.

(b)

$$GRUPPE_1 \cup (\rho_{Preis \leftarrow Kaufpreis}(GRUPPE_2))$$

## Aufgabe 3 Anfrageformulierung

(a)

```
SELECT DISTINCT Name, VertragsNr, Datum
FROM Kunde as k, bucht as b, Baustein as ba
WHERE k.KundenNr = b.KundenNr
      AND ba.Reisevertrag = b.VertragsNr
      AND ba.Kategorie = 'Kreuzfahrt'
```

Alternative Lösungen:

```
SELECT DISTINCT Name, VertragsNr, Datum
FROM Kunde as k
      JOIN bucht AS b ON k.KundenNr = b.KundenNr
      JOIN Baustein AS ba ON ba.Reisevertrag = b.VertragsNr
WHERE ba.Kategorie = 'Kreuzfahrt'
```

```
SELECT DISTINCT Name, VertragsNr, Datum
FROM Kunde as k
      NATURAL JOIN bucht AS b
      JOIN Baustein AS ba ON ba.Reisevertrag = b.VertragsNr
WHERE ba.Kategorie = 'Kreuzfahrt'
```

Hinweis: In den oberen Abfragen kann auch *Reisevertrag* statt *VertragsNr* verwendet werden.

(b)

```
SELECT Name, KundenNr
FROM Kunde
WHERE KundenNr NOT IN (
      SELECT KundenNr
      FROM bucht
)
```

Alternative Lösung:

```
SELECT Name, KundenNr
FROM Kunde k
      LEFT JOIN bucht b ON b.KundenNr = k.KundenNr
WHERE b.VertragsNr is NULL
```

Hinweis zur alternativen Lösung: In der WHERE-Klausel kann auch *Datum* statt *VertragsNr* verwendet werden.

(c)

```
SELECT COUNT(KundenNr)
FROM bucht
WHERE KundenNr = '1234'
```

(d)

```
SELECT Firmenname, SUM(Preis) AS GesPreis
FROM Baustein
GROUP BY Firmenname
HAVING (GesPreis >= 3500)
```

## Aufgabe 4 Normalisierung

(a)

$$F = \{LieferantenID \rightarrow Firmenort, LieferantenID \rightarrow Branchencode, \\ LieferantenID \rightarrow Branche, Branchencode \rightarrow Branche\}$$

(b)

Die Relation *LIEFERANT* befindet sich in der 2NF. Der einzige Schlüssel (Primärschlüssel) dieser Relation  $\{LieferantenID\}$  ist nicht zusammengesetzt und die Nichtschlüsselattribute sind voll funktional abhängig vom Schlüssel.

(c)

Die Relation *LIEFERANT* befindet sich nicht in der 3NF. Das Attribut *Branche* ist funktional abhängig vom Attribut *Branchencode*. *Branchencode* ist Nichtschlüssel und *Branche* ist Nichtschlüsselattribut. Diese Abhängigkeit verletzt die Bedingung für die 3NF. Daher wird die Relation wie folgt zerlegt:

*LIEFERANT\_NEU*(LieferantenID, Firmenort, Branchencode)

*BRANCHE*(Branchencode, Branche)

$\{Branchencode\}$  in *LIEFERANT\_NEU* ist ein Fremdschlüssel, welcher auf den Primärschlüssel  $\{Branchencode\}$  in Relation *BRANCHE* verweist.

(Punkte: 3 Punkte für die Begründung, warum das Ergebnis von (b) nicht der 3NF genügt, 2 Punkte für die Zerlegung)