

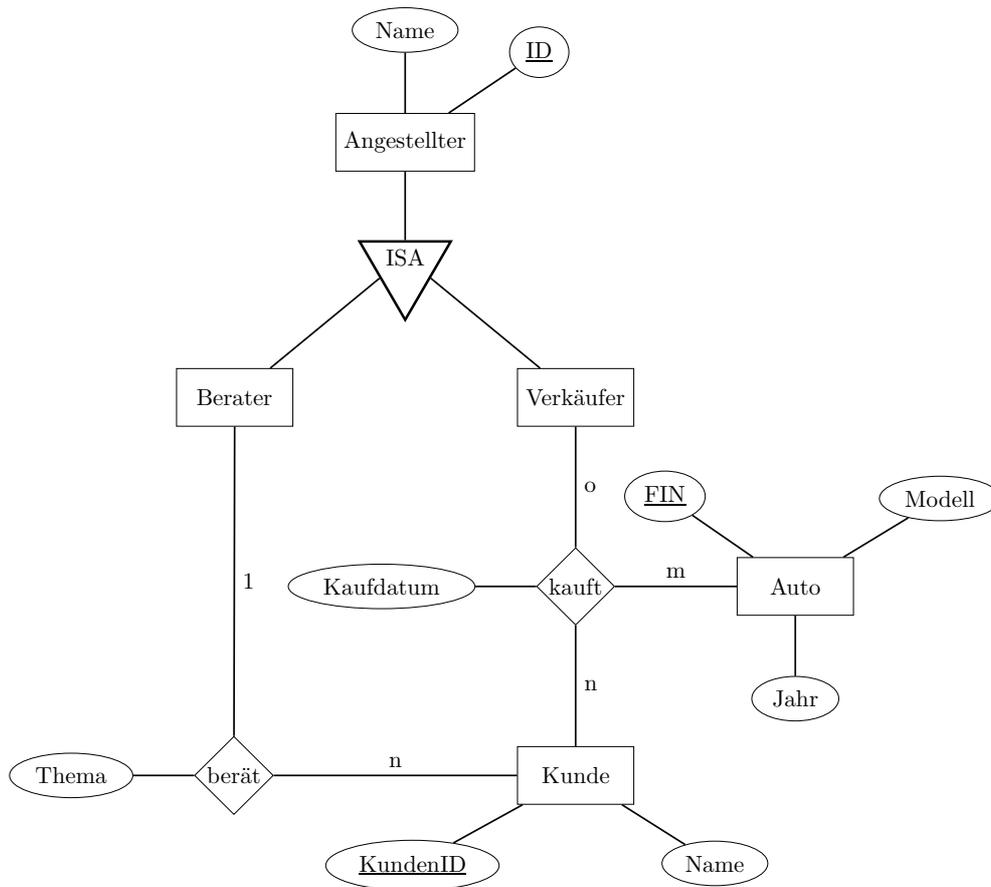


**FernUniversität in Hagen**

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

**23.03.2022**

## Aufgabe 1 Entity-Relationship-Diagramm



Punkte: is-a-Relationship 1 Punkt, jeweils 1 Punkt für einen Entity- bzw. Beziehungstyp. Alternative Lösungen, ohne einen ternären Beziehungstyp sind richtig (ein binärer n:m-Beziehungstyp zwischen Verkäufer und Auto in Verbindung mit einem binären n:m-Beziehungstyp zwischen Kunde und Auto). Für alle Entity- / Beziehungstypen werden insg. 7 Punkte vergeben. Pro Schlüsselattribut und Kardinalität wird ein 1 Punkt vergeben (Kardinalitäten: 1 Punkt für den 1:n-Beziehungstyp zwischen Berater und Kunde und insg. 2 Punkte für die n:m-Beziehungstypen zwischen Verkäufer, Auto und Kunde), pro Nichtschlüsselattribut wird ein halber Punkt vergeben (insg. 3 Punkte).

## Aufgabe 2 Relationenalgebra und Optimierung

(a)

```
SELECT Name
FROM Kunde
    NATURAL JOIN Konto
WHERE gewerblich='nein'
    AND AnzahlTransaktionenImMonat < 10
```

Alternative Lösung:

```
SELECT Name
FROM Kunde, Konto
WHERE Kunde.KundenID = Konto.KundenID
    AND gewerblich='nein'
    AND AnzahlTransaktionenImMonat < 10
```

(b)

Um Join-Operationen möglichst effizient auszuführen, sollte überprüft werden, ob die zugehörigen Tupelmengen vor dem Join reduziert werden können. Selektionen, die sich auf eine Relation beziehen, werden daher vor dem Join ausgeführt, um die Anzahl der Tupel zu reduzieren. In der gegebenen Anfrage in Relationenalgebra beziehen sich die Selektionen auf die Relationen *Kunde* und *Konto*. Sie werden jedoch nach dem Join ausgeführt. Die gegebene Anfrage hat daher Optimierungspotenzial. Die Anfrage wird dahingehend modifiziert, dass die Selektionen vor dem Join ausgeführt werden, um die Tupelmengen zu reduzieren. Die modifizierte Anfrage sieht wie folgt aus:

$$\pi_{Name}((\sigma_{\text{gewerblich}='nein'}(Kunde)) \bowtie (\sigma_{\text{AnzahlTransaktionenImMonat}<10}(Konto)))$$

## Aufgabe 3 Anfrageformulierung

(a)

```
SELECT film, rkurz
FROM bewertet NATURAL JOIN fuehrt_regie
WHERE kkurz = 'k1'
```

Alternative Lösung:

```
SELECT film, rkurz
FROM bewertet, fuehrt_regie
WHERE kkurz = 'k1'
    AND bewertet.film = fuehrt_regie.film
```

(b)

```
SELECT titel
FROM Film
WHERE titel NOT IN (
    SELECT film
    FROM bewertet
)
```

(c)

```
SELECT film
FROM fuehrt_regie
    NATURAL JOIN Regisseur
WHERE oscarnom >= 3
```

Alternative Lösung:

```
SELECT film
FROM fuehrt_regie, Regisseur
WHERE fuehrt_regie.rkurz = Regisseur.rkurz
    AND oscarnom >= 3
```

(d)

```
SELECT film
FROM bewertet
    GROUP BY film
    HAVING COUNT(kkurz) > 3
```

## Aufgabe 4 Normalformen und Zerlegungen

(a)

$$F = \{ \{SpielerNr, Team\} \rightarrow Name, \{SpielerNr, Team\} \rightarrow AnzahlSpiele, \\ \{SpielerNr, Team\} \rightarrow TrainerName, Team \rightarrow TrainerName \}$$

Die Relation *SPIELER* genügt nicht der 2NF, da die Nichtschlüsselattribute (hier *TrainerName*) nicht voll funktional von beiden Schlüsselattributen abhängen. Daher wird die Relation wie folgt zerlegt:

*SPIELER\_NEU*(SpielerNr, SpielerName, AnzahlSpiele, Team)

*TRAINER*(Team, TrainerName)

(Punkte: 2 Punkte für die Fd-Menge; 3 Punkte für die Begründung, warum die Relation nicht der 2NF genügt; 1 Punkt für die Überführung in die 2NF)

(b)

Das Ergebnis von (a) genügt der 3NF, da keine funktionale Abhängigkeit  $X \rightarrow A$  mit  $X$  als Nichtschlüssel und  $A$  als Nichtschlüsselattribut existiert.

(Punkte: 3 Punkte für die Begründung, warum das Ergebnis von (a) der 3NF genügt)

(c)

Die Zerlegung ist nicht verlustfrei. Zerlegungen sind dann verlustfrei, wenn das Attribut, über das der natürliche Join für die Wiederherstellung der ursprünglichen Relation durchgeführt wird, ein Schlüssel einer der beiden Teilrelationen ist. Das Attribut  $B$  ist in keiner Teilrelation ein Schlüssel.

(Punkte: 1 Punkt für die Feststellung, ob die Zerlegung verlustfrei ist oder nicht; 3 Punkte für die Begründung)