

Aufgabe 1 Entity-Relationship-Diagramm

17 Punkte

Modellieren Sie das Entity-Relationship-Modell (ER-Modell), um Aufgabenbereiche von Angestellten eines Autohauses zu beschreiben.

Angestellte haben eine eindeutige ID und einen Namen. Ein Angestellter ist entweder ein Berater oder ein Verkäufer. Ein Berater berät verschiedene Kunden. Ein Kunde hat jedoch einen festen Berater, von dem er sich zu einem bestimmten Thema beraten lassen kann. Ein Kunde wird durch eine KundenID eindeutig identifiziert. Zu einem Kunden soll ebenfalls sein Name erfasst werden. Kunden können beliebig viele Autos bei verschiedenen Verkäufern kaufen. Verkäufer verkaufen somit beliebig viele Autos an verschiedene Kunden. Das Kaufdatum soll hierbei mit erfasst werden. Ein Auto besitzt eine eindeutige Fahrzeug-Identifizierungsnummer (FIN). Zudem werden für jedes Auto sein Modell und Jahr erfasst.

Wählen Sie in Ihrem ER-Modell zudem geeignete Primärschlüssel aus und geben Sie die Komplexität der Beziehungstypen an.

Aufgabe 2 Relationenalgebra und Optimierung

6 Punkte

Gegeben seien die folgenden Relationenschemata:

Kunde (KundenID, Name, gewerblich)

Konto (IBAN, KundenID, AnzahlTransaktionenImMonat)

Gegeben sei außerdem die folgende Anfrage in Relationenalgebra:

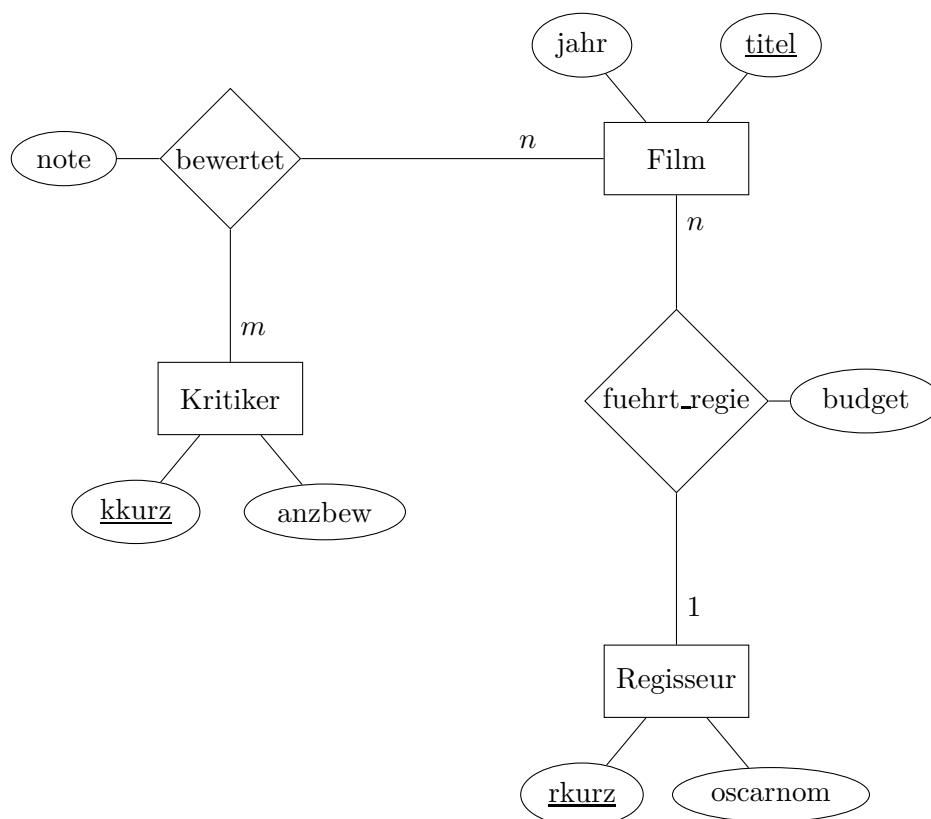
$$\pi_{Name}(\sigma_{\text{gewerblich}='nein' \wedge \text{AnzahlTransaktionenImMonat} < 10} (Kunde \bowtie Konto))$$

- (a) Formulieren Sie die obere Anfrage in Relationenalgebra in eine äquivalente SQL-Anfrage um. 2 Punkte
- (b) Bitte begründen Sie, ob die oben angegebene Anfrage in Relationenalgebra Optimierungspotenzial hat, um die Anfrage effizienter durchzuführen, oder nicht (in diesem Fall ist die Anfrage bereits optimal). Falls die Anfrage Optimierungspotenzial hat, dann modifizieren Sie diese und geben die modifizierte Anfrage an. Begründen Sie, inwieweit Ihre Modifikationen zu einer Optimierung beigetragen haben. 4 Punkte

Aufgabe 3 Anfrageformulierung

14 Punkte

Gegeben sei das folgende Entity-Relationship-Modell, welches die Zusammenhänge einer Filmkritik darstellt:



Einige Erläuterungen:

- Ein Regisseur führt Regie bei unterschiedlichen Filmen und wird eindeutig durch das Attribut *rkurz* identifiziert (*rkurz* steht für Kurzform eines Namens eines Regisseurs). *oscarnom* gibt an, wie häufig ein Regisseur für den Oscar „Beste Regie“ nominiert worden ist.
- Zu jedem Regie-Projekt wird das Budget des jeweiligen Films festgehalten.
- Ein Film wird durch den Titel eindeutig identifiziert.
- Kritiker bewerten verschiedene Filme und werden eindeutig durch das Attribut *kkurz* (Kurzform eines Namens eines Kritikers) identifiziert. *anzbew* gibt an, wie viele Filme ein Kritiker bereits bewertet hat.
- Jeder durch einen Kritiker bewertete Film erhält eine Note (*note* nimmt die Werte 1-5 an, wobei 5 die Bestnote ist). Der gleiche Film kann mehrere Bewertungen von verschiedenen Kritikern erhalten.

Aus diesen Zusammenhängen lassen sich die folgenden Relationen erstellen:

Film (titel, jahr)

Regisseur (rkurz, oscarnom)

fuehrt_regie (rkurz, film, budget)

PRIMARY KEY (rkurz, film)

FOREIGN KEY (rkurz) REFERENCES Regisseur (rkurz)

FOREIGN KEY (film) REFERENCES Film (titel)

Kritiker (kkurz, anzbew)

bewertet (kkurz, film, note)

PRIMARY KEY (kkurz, film)

FOREIGN KEY (kkurz) REFERENCES Kritiker (kkurz)

FOREIGN KEY (film) REFERENCES Film (titel)

Formulieren Sie jeweils eine SQL-Anfrage, um die folgenden Informationen zu erhalten (Wenn nachfolgend von Filmen, Regisseuren oder Kritikern gesprochen wird, dann sind Filmtitel und die Kurzformen der Namen gemeint):

- (a) Listen Sie die Filme und deren Regisseure auf, die der Kritiker mit der Kurzform „k1“ bewertet hat. 3 Punkte
- (b) Welche Filme haben noch keine Bewertung erhalten? 3 Punkte
- (c) Listen Sie die Filme auf, deren Regisseure mindestens 3 Oscar-Nominierungen erhalten haben. 3 Punkte
- (d) Welche Filme haben mehr als drei Bewertungen erhalten? 5 Punkte

Aufgabe 4 Normalformen und Zerlegungen

13 Punkte

Das relationale Modell einer Spielerverwaltung eines Vereines enthält das folgende Relationenschema:

SPIELER(SpielerNr, SpielerName, AnzahlSpiele, Team, TrainerName)

Die Nummer eines Spielers (*SpielerNr*) und sein *Team* identifizieren eindeutig den Namen des Spielers (*SpielerName*), die Anzahl der Spiele, die ein Spieler für sein Team absolviert hat (*AnzahlSpiele*), sowie den Trainer (*TrainerName*). Die Durchnummerierung der Spieler erfolgt pro *Team*, sodass die *SpielerNr* mehrfach vorkommen kann. Der Trainer wird zudem durch das *Team* eindeutig festgelegt.

Die Relation *SPIELER* enthält folgende Beispieleinträge:

<u>SpielerNr</u>	SpielerName	AnzahlSpiele	<u>Team</u>	TrainerName
01	Heike	104	A	Moritz
02	Maik	17	A	Moritz
⋮	⋮	⋮	⋮	⋮
01	Heike	34	B	Anne
02	Peter	29	B	Anne
⋮	⋮	⋮	⋮	⋮

Die Relation genügt der 1. Normalform.

- (a) Geben Sie die Menge funktionaler Abhängigkeiten (Fd-Menge) der Relation *SPIELER* an. Überprüfen Sie, ob die Relation *SPIELER* der 2. Normalform (2NF) genügt. Falls nicht, begründen Sie warum und überführen Sie die Relation in die 2NF (Geben Sie dazu die resultierenden Relationenschemata an). 6 Punkte
- (b) Überprüfen Sie, ob das Ergebnis von (a) der 3. Normalform (3NF) genügt. Begründen Sie, warum bzw. warum nicht und überführen Sie die Relationenschema(ta) ggf. in die 3NF. 3 Punkte
- (c) Betrachten Sie die folgende Relation R und ihre Zerlegung in R1 und R2: 4 Punkte

R		
A	B	C
2	3	4
8	3	9
5	6	7

R1	
A	B
2	3
8	3
5	6

R2	
B	C
3	4
3	9
6	7

Begründen Sie, ob diese Zerlegung verlustfrei ist, oder nicht.