

**Prüfung**  
**Datenbanksysteme, DBS**  
**am: 09.02.2021**

Erstprüfer: Prof. Dr. K. Weidenhaupt  
Zweitprüfer: Prof. Dr. Ch. Dalitz  
Bearbeitungszeit: 90 Minuten + 15 Minuten für Einscannen/Hochladen der Lösung  
Anzahl der Aufgaben: 4

**Name:** \_\_\_\_\_

**Matrikelnummer:** \_\_\_\_\_

### Hinweise:

1. Schreiben Sie Ihren Namen und Ihre Matrikelnummer auf Ihre Lösung.
2. Die Lösung ist handschriftlich anzufertigen (auf Papier oder elektronisch per Stifteingabe).
3. Sie können die Lösung direkt in die Aufgabenblätter oder auf separate Lösungsblätter schreiben. In letzterem Fall geben Sie bitte zu jeder Lösung die jeweilige Aufgabennummer an.
4. Die komplette Lösung ist in Form eines pdf-Dokumentes im Aufgabenbereich des Moodle-Raums für diese Prüfung hochzuladen.
5. Die Prüfung ist ab 45 Punkten sicher bestanden, die maximale Punktzahl ist 90.
6. Es gibt 2 Bonusaufgaben mit je 5 Punkten, die zum Erreichen der vollen Punktzahl nicht unbedingt gelöst werden müssen. Sie können damit aber fehlende Punkte aus anderen Aufgaben wettmachen.
7. Klausureinsicht: wird per Email bekannt gegeben.

**Viel Erfolg!**

Aufgabe	1	2	3	4	Bonus- aufgaben	$\Sigma$
Maximalpunkte	22	39	10	19	10	90 (+10)
Erreichte Punkte						

<b>Bewertung</b>		
_____	_____	_____
Note	Unterschrift	Datum

## 1. Aufgabe - ER

Sie sollen ein Informationssystem zur Projektverwaltung in einer Softwarefirma erstellen. Zu verwaltende Basisdaten sind:

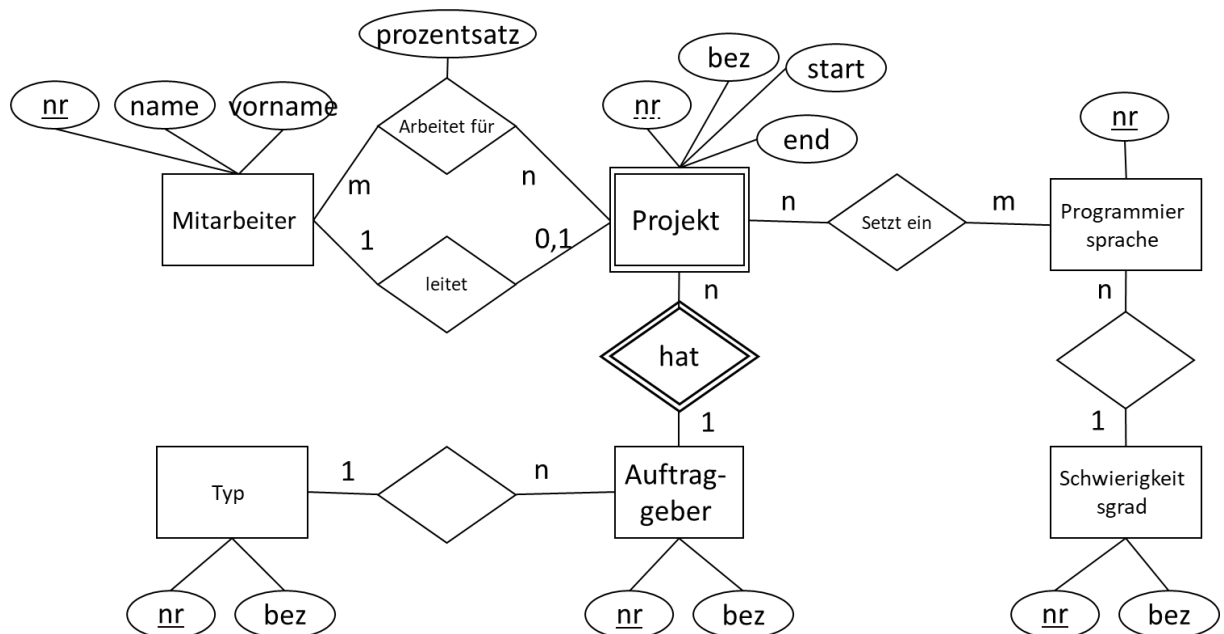
- Mitarbeiter mit Vorname und Nachname.
- Projekte mit Bezeichnung, Start- und Enddatum.
- Auftraggeber mit Bezeichnung und Typ (privat, öffentlich, intern).
- Programmiersprache mit Bezeichnung und Schwierigkeitsgrad (einfach, mittel, schwierig).

Regeln:

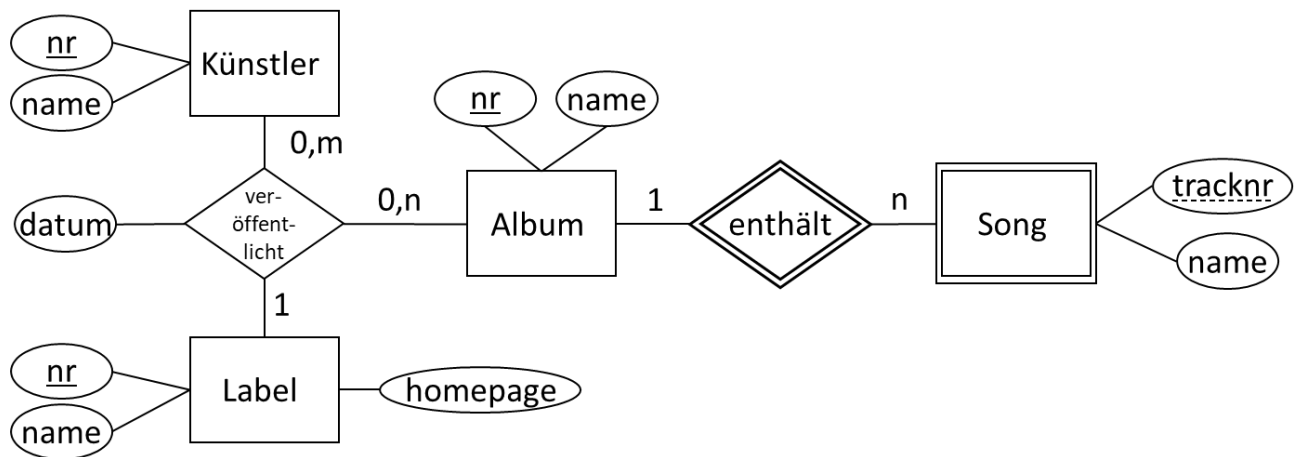
- Ein Mitarbeiter ist zu einem bestimmten Prozentsatz in einem Projekt tätig. Ein Mitarbeiter ist in mindestens einem Projekt tätig, kann aber höchstens ein Projekt leiten.
- In einem Projekt arbeitet mindestens ein Mitarbeiter. Ein Projekt wird von genau einem Mitarbeiter geleitet.
- Ohne einen alleinigen Auftraggeber kann es ein Projekt nicht geben. Ein Auftraggeber kann mehrere Projekte in Auftrag geben.
- In einem Projekt werden eine oder mehrere Programmiersprachen eingesetzt. Eine Programmiersprache kann in verschiedenen Projekten eingesetzt werden.

Aufgaben:

- a) Entwickeln Sie für diese Anforderungen ein ER-Diagramm inklusive aller Schlüsselattribute und Kardinalitäten. Die Wahl der Primärschlüssel ist Ihnen überlassen, fügen Sie ggf. künstliche Schlüsselattribute hinzu. (15)



- b) Transformieren Sie das folgende ER-Modell für eine Musikdatenbank in ein relationales Schema. Machen Sie im relationalen Schema alle Schlüsselattribute und Fremdschlüsselverweise kenntlich! (10)



Künstler(nr, name)

Album(nr, name)

Label(nr, name, homepage)

Song(albumnr, tracknr, name)

Veröffentlichung(künstlernr, albumnr, labelnr, datum)

FKs:

Song.albumnr → Album.nr

Veröffentlichung.albumnr → Album.nr

Veröffentlichung.künstlernr → Künstler.nr

Veröffentlichung.labelnr → Label.nr

## 2. Aufgabe – SQL-DML

Ein Copy-Shop verwaltet in einem RDBMS Informationen über Kopiervorgänge an seinen Kopierern. Das Schema der Datenbank besteht aus folgenden Relationen:

- Kopierer(CID, CName, Hersteller, Kaufdatum)
- Kunde(KID, KName, Plz, Ort, Straße)
- Kopiervorgang(KID, CID, Datum, Anzahl)

Kopiervorgang.KID und Kopiervorgang.CID sind Fremdschlüssel-Attribute auf Kunde.KID bzw. Kopierer.CID; Alle IDs seien vom Typ INTEGER, die Attribute Kaufdatum und Datum seien vom Typ DATE. Anzahl ist vom Typ INTEGER und muss einen Wert zwischen minimal 1 und maximal 500 annehmen.

Formulieren Sie folgende Sachverhalte als SQL-Statements:

- a) Legen Sie die Tabelle **Kopiervorgang** inklusive aller Constraints an. (5)

```
CREATE TABLE Kopiervorgang (  
  KID INTEGER REFERENCES Kunde.KID,  
  CID INTEGER REFERENCES Kopierer.CID,  
  Datum Date,  
  Anzahl INTEGER CHECK (Anzahl BETWEEN 1 AND 500),  
  PRIMARY KEY (KID, CID, Datum)  
);
```

- b) Fügen Sie folgende Information in die Datenbank ein: Der Kunde mit der KID 4711 hat am 09.02.2021 am Kopierer mit der CID 14 hundert Kopien angefertigt. (4)

```
INSERT INTO Kopiervorgang  
VALUES ('4711', '14', to_date('09.02.2021', 'DD.MM.YYYY'), '100');
```

*Damit Sie nicht blättern müssen, ist das Schema hier nochmal angegeben:*

Kopierer(CID, CName, Hersteller, Kaufdatum)

Kunde(KID, KName, Plz, Ort, Straße)

Kopiervorgang(KID, CID, Datum, Anzahl)

- c) Alle Kopiervorgänge, die vor dem 1.1.2021 an Kopierern des Herstellers 'Xorex' durchgeführt wurden, sollen aus der Datenbank gelöscht werden. **(5)**

```
DELETE FROM Kopiervorgang
WHERE Datum < to_date('01.01.2021', 'DD.MM.YYY') AND CID IN
(SELECT CID FROM Kopierer WHERE Hersteller = 'Xorex');
```

- d) Bei allen Kopiervorgängen, bei denen das Datum unbekannt ist, soll das Datum auf den 1.1.2020 gesetzt werden. **(4)**

```
UPDATE Kopiervorgang SET Datum = to_date('01.01.2020', 'DD.MM.YYYY') WHE-
RE Datum IS NULL;
```

*Damit Sie nicht blättern müssen, ist das Schema hier nochmal angegeben:*

Kopierer(CID, CName, Hersteller, Kaufdatum)

Kunde(KID, KName, Plz, Ort, Straße)

Kopiervorgang(KID, CID, Datum, Anzahl)

- e) Welche Kunden (KID) fertigten am 15.01.2021 Kopien an Kopierern des Herstellers 'Conan' an? (6)

```
SELECT KID
```

```
FROM Kopiervorgang JOIN Kopierer ON Kopiervorgang.CID = Kopierer.CID
```

```
WHERE Datum = to_date('15.01.2021', 'DD.MM.YYY') AND Kopierer.Hersteller = 'Conan';
```

- f) Geben Sie für jeden Kopierer (CID, CName) die Gesamtzahl von Kopien an, die auf dem Kopierer gemacht wurden. Es sollen auch Kopierer aufgelistet werden, an denen noch nie kopiert wurde. (6)

```
SELECT CID, CName, Sum(Anzahl)
```

```
FROM Kopierer LEFT JOIN Kopiervorgang ON Kopierer.CID = Kopiervorgang.CID
```

```
GROUP BY Kopierer.CID;
```

*Damit Sie nicht blättern müssen, ist das Schema hier nochmal angegeben:*

Kopierer(CID, CName, Hersteller, Kaufdatum)

Kunde(KID, KName, Plz, Ort, Straße)

Kopiervorgang(KID, CID, Datum, Anzahl)

g) Welche Kopierer (alle Attribute) wurden zuletzt angeschafft? (6)

SELECT \*

FROM Kopierer

WHERE Kaufdatum = (SELECT max(Kaufdatum) FROM Kopierer);

**Die folgende Aufgabe ist eine Bonusaufgabe und muss zum Erreichen der vollen Punktzahl nicht unbedingt gelöst werden. Sie können damit aber fehlende Punkte aus anderen Aufgaben wettmachen.**

h) Welche Kunden (KID, KName) kopieren grundsätzlich *nur* an Kopierern von 'Conan'? (5)

SELECT KID, KName FROM Kunde

WHERE KID NOT IN (

SELECT KID FROM Kopiervorgang NATURAL JOIN Kopierer

WHERE Hersteller <> 'Conan');

### 3. Aufgabe – Trigger und referentielle Integrität

Ihre Datenbank enthält die Tabellen X und Y sowie darauf operierende Trigger gemäß folgender Definition

```
CREATE TABLE X (
  A INT PRIMARY KEY,
  B INT
);
CREATE TABLE Y (
  C INT PRIMARY KEY,
  D INT REFERENCES X(A)
);

CREATE OR REPLACE FUNCTION func1() RETURNS TRIGGER AS '
BEGIN
  DELETE FROM Y WHERE Y.D = old.A;
  RETURN new;
END;
' LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION func2() RETURNS TRIGGER AS '
BEGIN
  UPDATE Y SET D = NULL WHERE Y.D = old.A;
  RETURN old;
END;
' LANGUAGE 'plpgsql';

CREATE TRIGGER t1 BEFORE UPDATE ON X FOR EACH ROW EXECUTE PROCEDURE func1();
CREATE TRIGGER t2 BEFORE DELETE ON X FOR EACH ROW EXECUTE PROCEDURE func2();
```

Ihre Tabellen X und Y enthalten folgende Tupel.

X	
A	B
1	42
2	4711

Y	
C	D
17	1
18	1
19	2

a) Welche Tupel enthalten X und Y nach Ausführung der folgenden Befehle? (6)

```
UPDATE X SET A = 3 WHERE A = 1;
DELETE FROM X WHERE A = 2;
```

X	
A	B
3	42
2	4711

Y	
C	D
17	4
18	4
19	NULL



- b) Den Effekt von Trigger t2 kann man auch auf einfachere Weise erreichen. Wie müsste man dazu die Tabellendefinition von X und/oder Y modifizieren? (2)

```
CREATE TABLE Y (  
  C INT PRIMARY KEY,  
  D INT REFERENCES X(A) ON DELETE SET NULL  
);
```

- c) Das Konzept der referentiellen Integrität ist nicht nur im Datenbankbereich wichtig. Geben Sie aus Ihrer Alltagserfahrung ein Beispiel für eine Verletzung der referentiellen Integrität. (2)

404 Error bei Aufruf einer Webseite, die unter dem Link nicht mehr erreichbar ist

Kein Anschluss unter dieser Nummer (Telefon)

#### 4. Aufgabe - Normalisierung

Der Internet-Händler Nozama speichert Daten über Bücher, Kunden und Buchbewertungen durch Kunden in einer Relation mit dem folgenden Schema:

Nozama( ISBN, Autor, Titel, Genre, KdNr, KdName, KdAdr, KdEmail, Bewertung)

Folgende Randbedingungen seien bekannt:

- Ein Buch wird durch eine ISBN identifiziert, jedes Buch hat jeweils genau einen Titel und gehört zu einem Genre, hat aber möglicherweise mehrere Autoren;
- Jeder Autor hat sich auf genau ein Genre spezialisiert (allerdings gibt es natürlich mehrere Autoren pro Genre);
- Kunden werden über eine KdNr identifiziert; sie haben einen Namen (KdName), eine postalische Adresse (KdAdr) und möglicherweise mehrere email-Adressen (KdEmail).
- Ein Kunde kann pro Buch maximal eine Bewertung abgeben;

a) Welche nicht-trivialen funktionalen Abhängigkeiten ergeben sich hieraus? (8)

ISBN  $\rightarrow$  Titel, Genre

Autor  $\rightarrow$  Genre

KdNr  $\rightarrow$  KdName, KdAdr

ISBN, KdNr  $\rightarrow$  Bewertung

b) Geben Sie einen Primärschlüssel der Relation Nozama an. (3)

ISBN, Autor, KdNr, KdEmail

c) Für das Relationenschema  $R(\underline{A}, \underline{B}, \underline{C}, D, E, F, G, H)$  sind folgende funktionalen Abhängigkeiten bekannt:

- $\{A\} \rightarrow \{G, H\}$
- $\{B, C\} \rightarrow \{E, F\}$
- $\{E\} \rightarrow \{D\}$
- $\{G\} \rightarrow \{H\}$

Führen Sie eine Normalisierung bis zur 3. Normalform durch. Geben Sie bei den einzelnen Zwischenschritten an, welche Normalform aufgrund welcher funktionalen Abhängigkeit verletzt wurde. (8)

$\{A\} \rightarrow \{G, H\}$  verletzt 2NF in R  
führt zu  $R1(\underline{A}, G, H)$  und  $R2(\underline{A}, \underline{B}, \underline{C}, D, E, F)$

$\{B, C\} \rightarrow \{E, F\}$  verletzt 2NF in R2  
führt zu  $R21(\underline{B}, \underline{C}, E, F, D)$  und  $R22(\underline{A}, \underline{B}, \underline{C})$

$\{E\} \rightarrow \{D\}$  verletzt 3NF in R21  
führt zu  $R211(\underline{E}, D)$  und  $R212(\underline{B}, \underline{C}, E, F)$

$\{G\} \rightarrow \{H\}$  verletzt 3NF in R1  
führt zu  $R11(\underline{G}, H)$  und  $R12(\underline{A}, G)$

Normalisiertes Relationenschema besteht aus:

$R11(\underline{G}, H)$   
 $R12(\underline{A}, G)$   
 $R211(\underline{E}, D)$   
 $R212(\underline{B}, \underline{C}, E, F)$   
 $R22(\underline{A}, \underline{B}, \underline{C})$

**Die folgende Aufgabe ist eine Bonusaufgabe und muss zum Erreichen der vollen Punktzahl nicht unbedingt gelöst werden. Sie können damit aber fehlende Punkte aus anderen Aufgaben wettmachen.**

d) Beweisen Sie mithilfe der Armstrong-Regeln IR1 - IR3 (Reflexivität, Augmentation und Transitivität) folgende Inferenzregel:  $X \rightarrow Y$  und  $X \rightarrow Z \Rightarrow X \rightarrow Y \cup Z$  (5)

$X \rightarrow Y$  mit  $X$  augmentieren (IR2):  $X = X \cup X \rightarrow Y \cup X$  (1)

$X \rightarrow Z$  mit  $Y$  augmentieren (IR2):  $X \cup Y \rightarrow Z \cup Y$  (2)

(1) und (2) mit Transitivität (IR3):  $X \rightarrow Y \cup Z$