


**Fachprüfung Crossmediale Datenbanken
 Sommersemester 2019 (1. Prüfungszeitraum)**

Studiengang: Bachelor Druck- und Medientechnik

Prüfungsdauer: 90 min.

Prüfungstermin: 11.07.2019

Letzter Prüfungsversuch?	<input type="checkbox"/> ja
	Punkte aus den Übungen: 40 / 50 Übung bestanden im WS/SS _____ Punkte aus der Prüfung: 37 / 50 Gesamtpunktzahl: 77 / 100 78,5
Die Klausur ist mit 25 Punkten bestanden. Bewertung der Leistungen für das Fach mit Note: 2,3	Unterschrift Prüfer: 

Folgende Hilfsmittel dürfen benutzt werden:

- Eine DIN A5 Seite, beidseitig beschriftet.

Andere Hilfsmittel sind nicht erlaubt!

Prüfungsteile:	Zu erreichende Punkte:	Erreichte Punkte:
I. Multiple Choice	14	10
II. Abfragen mit SQL	12	8
III. DB-Entwicklungsaufgabe	24	19

Viel Erfolg 😊

I. Multiple Choice: Es sind mehrere Antworten möglich. (Je 2P)

a) Welche Aussagen zum Phasenmodell für den Datenbankentwurf sind korrekt.

- ✓ Die Anforderungsanalyse ist die letzte Phase.
- ✓ Das ER-Modell ist Teil des Konzeptionellen Entwurfs.
- 1,5 ✓ Das Relationenmodell ist Teil des Konzeptionellen Entwurfs
- f Die Datendefinition erfolgt nach dem logischen Entwurf.

b) ER-Modell: Welche Aussagen sind korrekt?

- ✓ Entitäten werden als Rechtecke dargestellt.
- 1,5 ✓ Entitäten werden als Ovale dargestellt.
- f Relationen werden als Rauten dargestellt. ~~(ausgedr. Relation - Relationship)~~
Tabelle / hier Beziehung
- ✓ Nur Entitäten können Attribute haben.

c) Welche Aussagen treffen für die 2. Normalform zu?

- ✓ Ein eindeutiger Primärschlüssel muss vorhanden sein.
- 2 ✓ Es muss transitive Abhängigkeiten geben.
- ✓ Jedes Nichtschlüsselattribut muss voll funktional abhängig vom Primärschlüssel sein.
- ✓ Es darf nur ein Nichtschlüsselattribut geben.

d) Welche Kardinalitäten gibt es?

- 0,5 ✓ n:m
- ✓ 0:1
- ✓ 1:n
- f n:1

e) Welche der folgenden Aussagen stammen nicht aus den Regeln von Codd?

- 2 ✓ Variabilität der Tupelposition
- ✓ Information für den Benutzer ausschließlich durch Werte ausgedrückt
- ✓ Eindeutigkeit der Tupel muss gegeben sein.
- ✓ Abgeschlossenheit: Ergebnis einer Anfrage ist eine Relation und kann wieder als Eingabe für die nächste Anfrage verwendet werden.

f) Welche Aussagen zu Operationen der Relationenalgebra sind korrekt?

- 1,5 ✓ Um die Mengenoperationen "Vereinigung", "Differenz", "Durchschnitt" und "Symmetrische Differenz" auf zwei Relationen R und S anwenden zu können, müssen beide Relationen zueinander kompatibel sein.
- ✓ Mit der Selektion werden einzelne Tupel ausgewählt.
- ✓ Die Selektion bestimmt, welche Attribute dargestellt werden.
- ✓ Ein Join ist die sequentielle Ausführung von kartesischem Produkt und Selektion.

g) Welche Aussagen zur Schematransformation sind korrekt?

- ✓ Ein Beziehungstyp der Kardinalität 1:1 wird über einen Fremdschlüssel abgebildet: Der Primärschlüssel eines (beliebigen) Entitätentyps, der an der Beziehung teilnimmt, wird als Fremdschlüssel in der Relation des anderen Entitätentyps verwendet.
- ✓ Ein Beziehungstyp der Kardinalität 1:n wird über einen Fremdschlüssel abgebildet: Der Primärschlüssel eines (beliebigen) Entitätentyps, der an der Beziehung teilnimmt, wird als Fremdschlüssel in der Relation des anderen Entitätentyps verwendet.
- 1P Ein Beziehungstyp der Kardinalität M:N wird auf eine Relation abgebildet: Die Primärschlüssel der Entitätentypen, die an der Beziehung teilnehmen, werden als Fremdschlüssel in diese Relation übergeben und bilden i.d.R. deren zusammengesetzten Primärschlüssel.
- Ein Beziehungstyp und seine elementaren Attribute werden auf eine Relation und deren Attribute abgebildet. ^{Tabelle?}

II. Abfragen mit SQL (je 4 Punkte)

Gegeben ist folgendes Relationenschema eines Ihnen bekannten Computerspiels:

Dorf (id, name, haeuptling_id)
Bewohner (id, name, dorf_id, geschlecht, beruf, gold, status)
Gegenstand (id, name, besitzer_id)

Formulieren Sie auf dem obigen Relationenschema folgende Anfragen als Ausdruck in SQL.

a) Zeigen Sie für jedes Dorf absteigend sortiert an, wieviel Vermögen die Bevölkerung der jeweiligen Dörfer zusammen haben.

b) Zeigen Sie alle Gegenstände, die den Bewohnerinnen gehören, die nicht in „Gurkendorf“ wohnen.

c) Lesen Sie alle Gegenstände aus, die keinem gehören und weisen Sie diese dem Hörbuchautoren zu.

3P a) ^{dorf_id} SELECT SUM (bewohner.gold) FROM Bewohner ~~id~~ GROUP BY ~~dorf_id~~ bewohner.dorf_id ORDER BY DESC SUM (bewohner.gold);

3 → 2P b) SELECT * FROM Gegenstand WHERE ~~besitzer_id~~ gegenstand.besitzer_id = Bewohner.dorf_id AND Bewohner.dorf_id = dorf_id AND Dorf.name ~~is~~ IS NOT = "Gurkendorf";

2P c) (SELECT * FROM Gegenstand WHERE besitzer_id IS NULL) und vorher auslesen
UPDATE Gegenstand SET besitzer_id = "Hörbuchautor" WHERE besitzer_id IS NULL;

oder: INSERT INTO Bewohner (id, name, dorf_id, geschlecht, beruf, gold, status) VALUES (20, "Hörbuchautor", 3, ...); ^{wird relevant}

dann UPDATE Gegenstand SET besitzer_id = 20 WHERE besitzer_id IS NULL

III. DB-Entwicklungsaufgabe

Gehen Sie bitte von nachfolgend dargestelltem Sachverhalt aus und beantworten Sie die Fragen:

- a) Bitte entwickeln Sie das Datenmodell für die Abbildung des Sachverhaltes. Stellen Sie das Datenmodell in Form eines **ER-Diagramms** dar. Machen Sie die Beziehungen entsprechend der in der Aufgabenstellung geforderten Kardinalitäten deutlich. (8 Punkte) *auf Zusatzblättern Seite 1*
- b) Leiten Sie aus dem ER-Diagramm (aus 3.1) normalisierte Relationen gemäß der Ihnen bekannten Transformationsregeln ab. Notieren Sie die Relationen in **Relationenschreibweise**. Definieren Sie zu jeder Relation Primär- und Fremdschlüssel. Machen Sie die Fremdschlüssel kenntlich (z.B. Unterstreichung oder Hervorhebung). Ordnen Sie bitte **alle** angesprochenen Attribute zu. (6 Punkte) *auf Zusatzblättern Seite 2*
- c) Erläutern Sie bitte die Abbildung von M:N-Beziehungen auf das Relationale Datenmodell unter Verwendung eines solchen Beziehungstyps aus Ihrer bisher erarbeiteten Lösung der vorhergegangenen Teilaufgaben. (3 Punkte) *auf Zusatzblättern Seite 2*
- d) Entwickeln Sie bitte für die Relationen „Schüler“ und „Bauteil“ die entsprechenden SQL DDL-Statements, mit Abbildung der Primär- und Fremdschlüssel. Verwenden Sie geeignete Datentypen und begründen Sie Ihre Entscheidung. Die weiteren Relationen sind nicht zu entwickeln! (5 Punkte) *auf Zusatzblättern Seite 2*
- e) Tragen Sie mittels SQL DML einen Beispieldatensatz in die Tabelle „Schüler“ ein. (2 Punkte) *auf Zusatzblättern Seite 3 (Rückseite Seite 2)*

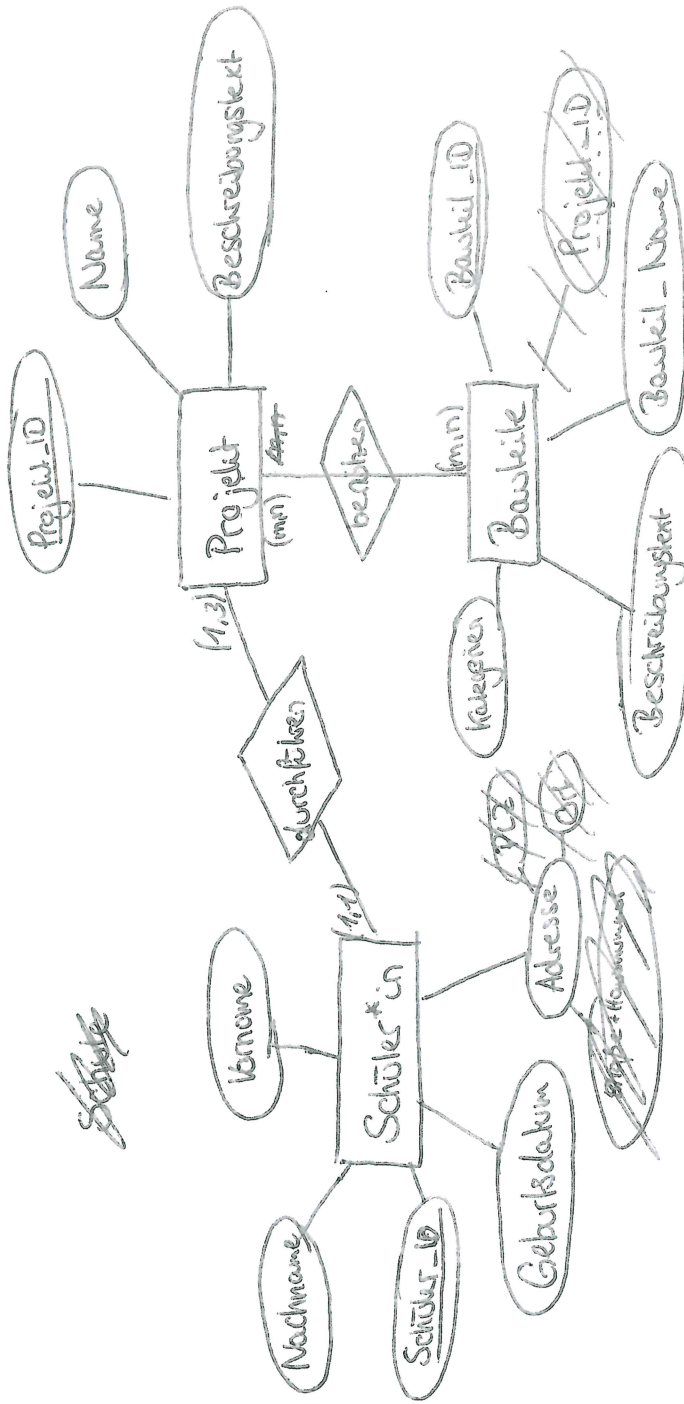
Gegenstand der vorliegenden Aufgabe:

Im Fach „Rechner in technische Prozesse einbinden“ an einer Schule, die Fachinformatiker*innen ausbildet, sollen Schüler*innen ein technisches Projekt erstellen. In einer Datenbank soll festgehalten werden, welche*r Schüler*in welches Projekt durchführt. Jede*r Schüler*in muss ein Projekt durchführen. Projekte können in einer Gruppe von bis zu drei Schüler*innen durchgeführt werden. Des Weiteren soll in einer Teilleiste eine Auflistung aller benötigten Bauteile für jedes Projekt erstellt werden. Eine Teilleiste soll folgende Informationen zur Verfügung stellen: Welches Bauteil wird in welchem Projekt wie oft benötigt?!

Für ein Projekt soll der Name des Projekts und eine kurzer Beschreibungstext gespeichert werden. Von Schüler*innen benötigen wir neben dem Vor- und dem Nachnamen, eine eindeutige ID-Nummer, das Geburtsdatum und Daten für die postalische Adresse. Für die Bauteile müssen der Name und ein Beschreibungstext erfasst werden können. Bauteile können mehrfach in einem oder mehreren Projekten vorkommen. Projekte können aus einem oder mehreren Bauteilen bestehen. Bauteile werden in Kategorien (z.B. Sensoren, Kondensatoren, Mikrocontroller) unterteilt.

III. a)

Schüler



- Teilleiste
- Kard. (m, n)?

JP

koriert auf
6,5 P

Projekt

Projekt-ID	Name	Beschreibungstext	Bausteil-ID	Schüler-ID
------------	------	-------------------	-------------	------------

Schüler*in

Schüler-ID	Vorname	Nachname	Adresse	Geburtsdatum	Projekt-ID
------------	---------	----------	---------	--------------	------------

4P

m:n nicht
abgebildet

Bausteil

Bausteil-ID	Bausteil-Name	Beschreibungstext	Kategorie	Projekt-ID
-------------	---------------	-------------------	-----------	------------

Adresse

Straße + Hausnummer		PLZ	Ort	Schüler-ID
---------------------	--	-----	-----	------------

Wachst

b) Schüler* in (Schüler-ID, Vorname, Nachname, Adresse, Geburtsdatum, Projekt-ID) - Primärschlüssel = _____
 Fremdschlüssel = - - - - -

Projekt (Projekt-ID, Name, Beschreibungstext, Bauteil-ID, Schüler-ID) -

c) Bauteil (Bauteil-ID, Bauteil-Name, Beschreibungstext, Kategorie, Projekt-ID)

aus m:n Beziehungen wird ein neuer zusammengesetzter Primärschlüssel gebildet, die Primärschlüssel werden in neuer Tabelle als Fremdschlüssel eingehen.
 würde bedeuten aus Projekt & Bauteile entsteht neue Relation mit Bspw. ein Namen Bauteil-Projekt (ID, Projekt-ID, Bauteil-ID)



d) CREATE TABLE Schüler* in (_____ mit z.B. nach Anrede etc

- Schüler-ID INTEGER, // ← wenn Char möglich → am besten ID nur aus Zahlen Char (10) auch möglich wenn wie bei Buchtitelnummer Buchstabe dazu
- Vorname VARCHAR(50), → 50 Buchstaben (inkl. Leerzeichen) für Vornamen sollten ausreichen et sein
- Nachname VARCHAR(50), → // _____ für Nachname
- Adresse CHAR(200), → aufpas auch gelesen, dass in neue Tabelle eintragen were, so alles möglich eig dort aber auch wieder eine ID wenn nicht schon direkt beim Namen (FK)
- Geburtsdatum DATE, → Datum für Geburtsdatum
- Projekt-ID INTEGER, // ← auch Char möglich → gleiches wie Schüler-ID

PRIMARY KEY (Schüler-ID)

FOREIGN KEY (Projekt-ID) REFERENCES Projekt (Projekt-ID)

JP

CREATE TABLE Bauteil (

- Bauteil-ID INTEGER, → ISBN z.B. davon ausgehend dass nur Zahlen, gleiches wie bei Schüler-ID möglich
- Bauteil-Name CHAR(50), → nur CHAR weil in manchen Berechnungs Zahlen integriert sind (Länge etc.)
- Beschreibungstext CHAR(1000), → gleiches wie Bauteil-Name + 1000 Zeichen müssen für Beschreibung reichen
- Kategorie VARCHAR(50), → Kategorie selbst zu wählen daher festgelegt nur Buchstaben
- Projekt-ID ~~INTEGER~~ INTEGER, → immer gleichen Datentyp wie Projekt-ID in Schüler, wenn dort Char, hier auch möglich
- PRIMARY KEY (Bauteil-ID), → Begründung gleiches wie Schüler-ID, Projekt-ID
- FOREIGN KEY (Projekt-ID) REFERENCES Projekt (Projekt-ID)

);

III e) INSERT INTO Schüler* in (Schüler_ID, Vorname, Nachname, Adresse, Geburtsdatum, Projekt_ID) VALUES (5, "Klaus", "Müller", "Musterstraße 3 18379 Musterstadt", "03.05.2000", 7)