Software Engineering Praxis
SS 2003
17. Juli 2003

Bearbeitungszeit: 120 Minuten

Name:	
Vorname:	
Matrikelnummer:	
Studiengang:	

Aufgabe	Punkte	Bewertung
1 (DAV)	**16**	
2 (DAV)	**8**	
3 (DAV)	**12**	
4 (DAV)	**8**	
5 (DAV)	**8**	
6 (DAV)	**12**	
7	**16**	
Summe:	**80**	

Die Aufgaben 1-6 sind für den DAV-Schein relevant!

Bitte benutzen Sie für die Lösungen den freigelassenen Platz nach der jeweiligen Angabe; sollte dieser nicht ausreichen, verwenden Sie bitte die Rückseite, wobei die Zuordnung zur jeweiligen Aufgabe deutlich erkennbar sein muss!

Bitte nicht mit Bleistift und nicht mit Rotstift schreiben
Die Angabe besteht aus 9 Seiten (ohne Titelblatt)!

Viel Erfolg!!!

Aufgabe 1 **(16 Punkte)**

In der Abteilung Software-Test soll eine Fehlerdatenbank aufgebaut werden mit folgendem Hintergrund:

a) **Jeder** Fehler, der im Test aufgedeckt wird, erhält einen Schlüssel F_ID; zusätzlich werden Datum (DAT) und Uhrzeit (UHR) des Entdeckens vermerkt.

b) Die Fehler werden grundsätzlich Fehlerklassen zugeordnet, wobei ein Fehler durchaus mehreren Klassen zugeordnet werden kann; die Fehlerklassen werden identifiziert über einen Schlüssel FKL_ID und erhalten daneben eine eindeutige Bezeichnung.

c) Fehler haben ihre Ursachen in einem oder auch mehreren Modulen; dabei kann es vorkommen, dass auch die Spezifikation bzw. die Spezifikationen falsch sind.

d) Zu jedem Modul gibt es eine Spezifikation, eine Spezifikation kann sich auch auf mehrere Module beziehen. Die Zuordnung Modul – Spezifikation ist erst dann von Interesse, wenn ein Fehler gefunden wird, der eine Änderung im Modul oder in der Spezifikation verlangt; wenn also ein Fehler behoben wurde, so muss ermittelt werden können, welches Modul betroffen war und welche Spezifikation zu diesem gehört.

e) Module wie auch deren Spezifikationen sind immer Projekten zugeordnet – ohne ein Projekt (identifiziert durch P_ID, weiteres Attribut ist TITEL) können weder Module noch Spezifikationen existieren. Jedes Modul wie auch jede Spezifikation gehört zu genau einem Projekt.

f) Innerhalb eines Projektes werden Module durch eine von 1 an fortlaufende Nummer (M_ID) identifiziert, weitere Attribute sind AUTOR und LOC; Spezifikationen werden auf die gleiche Art (laufende Nummer S_ID) identifiziert und erhalten als weiteres Attribut nur den AUTOR.

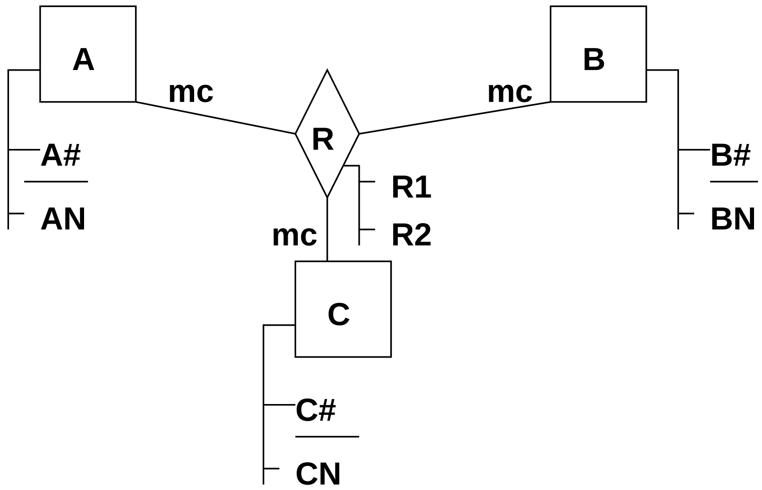
Erstellen Sie ein E/R-Diagramm inkl. Angabe aller Schlüssel, aller Komplexitätsgrade und der *weak entities*! Sollten Sie zusätzliche Annahmen treffen, so geben Sie diese bitte an!

Lösung nächste Seite

Lösung von Aufgabe 1:

Aufgabe 2 **(8 Punkte)**

Gegeben sei folgendes abstraktes E/R-Diagramm:



Weiterhin seien folgende funktionalen Abhängigkeiten bekannt:

$$(B\#, C\#) \rightarrow R1$$
$$(B\#, C\#) \rightarrow R2$$
$$(B\#, C\#) \rightarrow A\#$$

Geben Sie an, wie die Relation **R** in einem relationalen DBMS unter Berücksichtigung der 1.-3. Normalformen realisiert werden kann!

Aufgabe 3 **(12 Punkte)**

Gegeben ist eine Produktdatenbank mit den folgenden Tabellen:

Produkt	<u>PID</u>	Bezeichnung	Verkaufspreis

Teil	<u>TID</u>	Bezeichnung	Einkaufspreis

Enthaelt	<u>PID</u>	<u>TID</u>	Anzahl

a) (3 Punkte)

Suchen Sie mit einer SQL-Abfrage die Bezeichnungen aller Teile, die im Produkt mit der Bezeichnung „Alpha 2000" mindestens zweimal enthalten sind.

b) (3 Punkte)

Beschreiben Sie das Ergebnis der folgenden SQL-Abfrage mit Worten möglichst präzise:

```
SELECT Produkt.Bezeichnung,
       SUM(Teil.Einkaufspreis * Enthaelt.Anzahl)
FROM Produkt, Teil, Enthaelt
WHERE Produkt.PID = Enthaelt.PID
      AND Enthaelt.TID = Teil.TID
GROUP BY Produkt.PID;
```

c) (3 Punkte)

Bestimmen Sie durch eine SQL-Abfrage die Bezeichnungen aller Teile und dazu jeweils die Anzahl der Produkte, in der sie enthalten sind. Die Ausgabe soll sortiert nach der Anzahl der Produkte erfolgen.

d) (3 Punkte)

Ermitteln Sie mit einer SQL-Abfrage die Bezeichnungen und Nummern (TID) aller Teile, die gemeinsam mit dem Teil mit der Nummer „0815“ in einem Produkt enthalten sind.

Aufgabe 4 **(8 Punkte)**

Software gilt in allen ihren Ausprägungen (Quellcode, Dokumentation, . . .) als

schwer vermessbar.

Nennen und charakterisieren Sie kurz zwei Auswirkungen, die dies auf das Software-Projekt-Management hat.

Aufgabe 5 **(8 Punkte)**

In der Software-Entwicklung wird häufig folgende These genannt:

> Je später ein Fehler entdeckt wird, desto teurer ist dessen Behebung!

Begründen Sie diese These mit zwei nachvollziehbaren Argumenten!

Aufgabe 6 **(12 Punkte)**

Software-Inspektionen werden als effiziente Maßnahme zur Fehlerfindung bezeichnet.

a) (4 Punkte)

Ein Grund dafür liegt in der Rolle des Lesers. Geben Sie hierfür zwei nachvollziehbare Argumente, die dies belegen!

b) (8 Punkte)

Es gibt viele Gründe, die die eigentliche Prüfung (Inspektionssitzung) ineffizient / ineffektiv werden lassen. Nennen Sie vier nachvollziehbare Beispiele!

Aufgabe 7 **(16 Punkte)**

Eine Datei enthält Datensätze mit folgender Struktur:

- Datensatztrenner ist *newline*.
- Feldtrenner ist der Doppelpunkt.
- Das erste Feld enthält jeweils einen Produktnamen (identifiziert ein Produkt eindeutig).
- Das letzte Feld enthält eine ganze Zahl, die beschreibt, wieviele Einheiten des jeweiligen Produkts in einem bestimmten Zeitraum (dieser steht in einem anderen Feld) produziert wurden.
- Für ein Produkt können mehrere Einträge (Datensätze) existieren.

Erstellen Sie ein komplettes Perl-Programm, das diese Datei einliest (entweder über *stdin* oder als Argument übergeben), für jedes Produkt die gesamte Ausbringungsmenge ermittelt und die Liste, bestehend aus Produktnamen und Ausbringungsmenge aufsteigend sortiert nach Ausbringungsmenge nach *stdout* ausgibt!