Prof. Dr. Stefan Brass
Institut für Informatik
Universität Gießen

12.06.2001

Vorlesung „Datenbankentwurf“
— Klausur —

Hinweise

- Die Bearbeitungszeit ist 1 Stunde, 40 Minuten (von 10^{20} bis 12^{00})
- Bitte schreiben Sie die Antworten in den dafür vorgesehenen Platz. Sollte der Platz nicht ausreichen, schreiben Sie die Antwort auf die Rückseite oder auf zusätzliche Blätter. Bitte vermerken Sie deutlich, daß es noch eine Fortsetzung gibt und schreiben Sie Ihren Namen und die Nummer der Aufgabe auf alle zusätzlichen Blätter.
- Bitte fragen Sie, falls die Aufgabe Ihnen unklar oder mehrdeutig erscheint.
- Diese Klausur hat 10 Seiten. Bitte prüfen Sie die Vollständigkeit.

Daten zur Erstellung des Übungsscheins:

Name: ______________________________

Geburtsdatum: ______________________________

Geburtsort: ______________________________

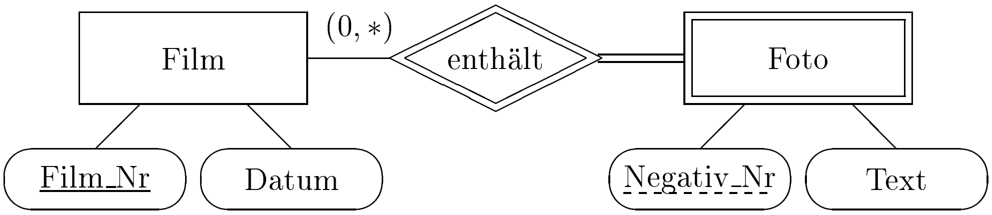
Ergebnisse:

Aufgabe	Punkte	Max. Punktzahl
1 (ER Model)		12
2 (Reverse Engineering, Oracle Designer)		12
3 (UML Klassendiagramme)		12
4 (BCNF)		4
Summe		40

Aufgabe 1 (ER Modell) 12 Punkte

Angenommen, Sie sollen eine kleine Datenbank zur Verwaltung von Fotos erstellen. Die Filme sind durchnumeriert. Außer der Film-Nummer ist das Datum der Entwicklung des Films abzuspeichern. Fotos gehören zu einem Film. Sie haben eine Negativnummer, die sie innerhalb des Filmes identifiziert (normalerweise 1..36). Nicht alle Fotos eines Filmes werden in der Datenbank eingetragen (nur die besseren). Zu jedem Foto in der Datenbank soll ein kurzer Text abgespeichert werden (eine Beschreibung des Fotos).

Aufgrund dieser Angaben haben Sie folgendes Datenbank-Schema erstellt, von dem hier vorausgesetzt werden soll, daß es korrekt ist:



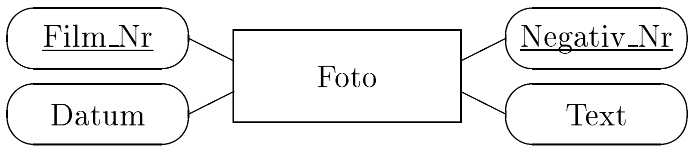
Nun haben weniger erfahrene Kollegen Ihres Teams aber noch andere Vorschläge gemacht. Welche dieser Vorschläge sind korrekt, welche nicht? Wenn ein Schema nicht korrekt ist, geben Sie bitte eine kurze Erklärung.

- Ein Schema ist bereits inkorrekt, wenn es zu allgemein ist, es also Datenbankzustände erlaubt, die keiner sinnvollen Situation in der realen Welt entsprechen. In diesem Fall wären zumindest zusätzliche Integritätsbedingungen nötig.
- Ein Schema ist inkorrekt auch dann, wenn es redundante Daten enthält, oder die Übersetzung ins Relationenmodell übliche Normalformen verletzen würde.
- Natürlich ist das Schema auch inkorrekt, wenn es nicht die geforderten Informationen enthält, oder nicht jede sinnvolle Zustand der realen Welt in einen Datenbankzustand abgebildet werden kann. Es soll hier aber angenommen werden, daß das obige Schema korrekt ist. Daher ist im wesentlichen die Äquivalenz der fraglichen Schemata zum gegebenen Schema zu prüfen.
- Schlechter Stil alleine soll hier nicht als inkorrekt gelten. Wenn Sie wollen, können Sie Stilfragen kommentieren, aber zusätzliche Punkte gibt es nur für ungewöhnlich gute Kommentare.

Für jede Teilaufgabe gibt es zwei Punkte (einen für das richtige Kreuz und einen für die Erklärung). Schemata können mehrfache Fehler enthalten. Es reicht aber, nur einen Fehler pro Teilaufgabe zu erklären.

Hinweis: Es könnte nützlich sein, sich die Übersetzung der ER-Schemata ins Relationenmodell zu überlegen.

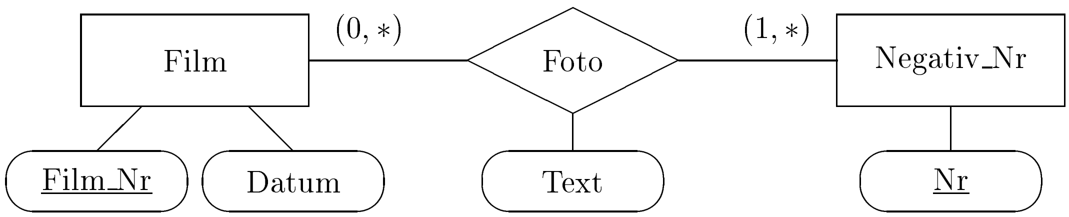
a) Kollege A würde alle Informationen in eine Tabelle stecken:



☐ Korrekt
☐ Falsch, Begründung: ____

____

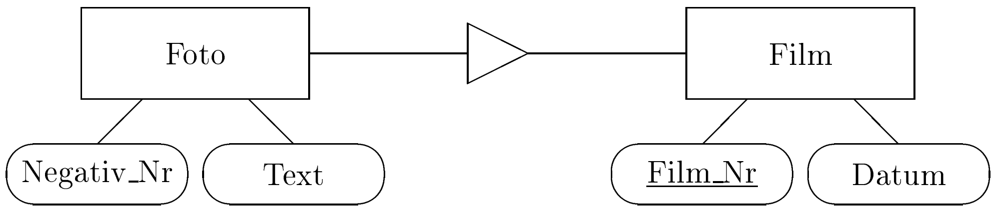
b) Kollege B sieht das Foto als Relationship zwischen Film und der Negativ-Nummer:



☐ Korrekt
☐ Falsch, Begründung: ____

____

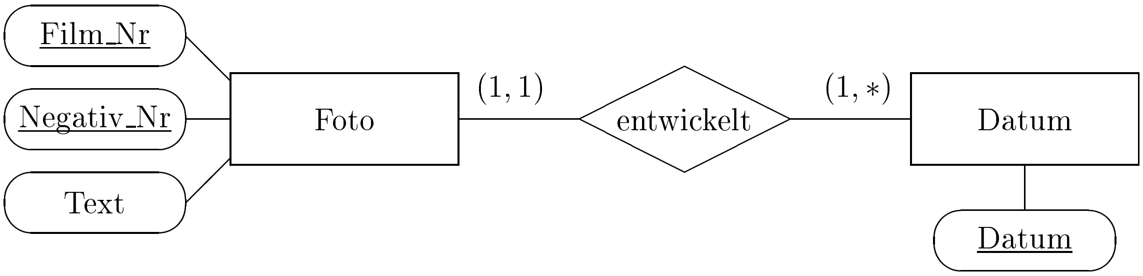
c) Kollege C meint, hier wäre eine Spezialisierung angebracht:



☐ Korrekt
☐ Falsch, Begründung: ____

____

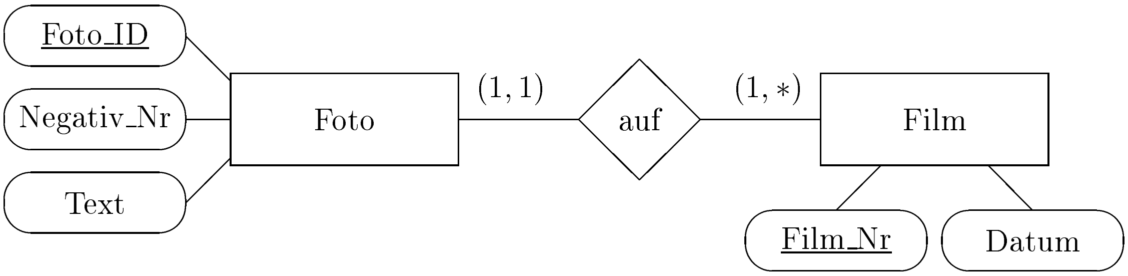
d) Kollege D hat eigentlich eine ähnliche Lösung wie Kollege A, aber er besteht darauf, daß „aus Normalisierungsgründen" das Datum in ein eigenes Entity ausgelagert werden muß:



☐ Korrekt
☐ Falsch, Begründung: ______________________________

____

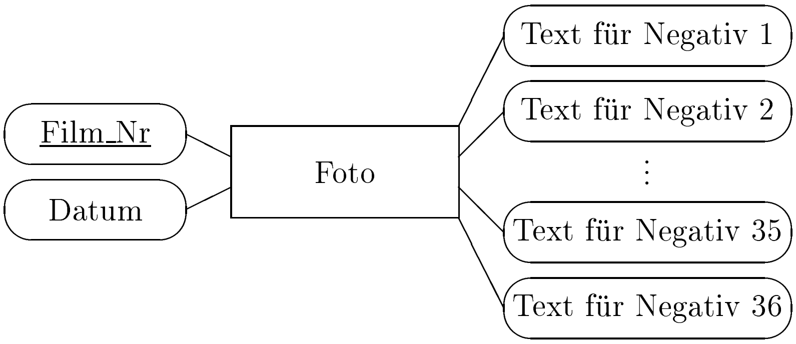
e) Kollege E lehnt zusammengesetzte Schlüssel ab, und schlägt vor, eine laufende Nummer für Fotos einzuführen:



☐ Korrekt
☐ Falsch, Begründung: ______________________________

____

f) Kollege F schließlich würde für jeden Film nur ein Entity vorsehen, und dann für jede mögliche Negativnummer ein Attribut für den Text. Es sei hier davon ausgegangen, daß es tatsächlich nur die Negativnummern 1 bis 36 geben kann:



☐ Korrekt

☐ Falsch, Begründung: ______________________________

____

Aufgabe 2 (Reverse Engineering, Oracle Designer) 12 Punkte

Die folgende relationale Datenbank wird von einem großen Software-Hersteller für Online-Support genutzt:

- Die Tabelle PRODUCTS enthält einen Eintrag für jede Version jedes Programmes, das momentan unterstützt wird:

  PRODUCTS(PRODNO, PRODNAME, VERSION)

- Die Tabelle EMPLOYEES enthält einen Eintrag für jeden Angestellen der Support-Abteilung:

  EMPLOYEES(EMPNO, FIRST_NAME, LAST_NAME, EMAIL)

- In der Tabelle EXPERT_FOR wird abgespeichert, welcher Angestelle Fragen bezüglich welchen Produktes beantworten kann:

  EXPERT_FOR(EMPNO→EMPLOYEES, PRODNO→PRODUCTS)

- In der Tabelle CUST_CONTACTS sind alle Kunden aufgelistet, die aufgrund von Wartungsverträgen Fragen stellen und an Diskussionen teilnehmen dürfen:

  CUST_CONTACTS(CONTNO, FIRST_NAME, LAST_NAME, EMAIL)

- Häufig müssen die Experten erst noch zurückfragen, so daß es mit einem einmaligen Austausch von Emails nicht getan ist. Außerdem können auch andere Kunden sich an einer laufenden Diskussion beteiligen. In der Tabelle THREADS gibt es einen Eintrag für jede Diskussion:

  THREADS(THREADNO, DATE_STARTED, STARTED_BY→CUST_CONTACTS, DATE_CLOSED^o)

- Schließlich werden die einzelnen Nachrichten in der Tabelle MESSAGES abgespeichert:

  MESSAGES(THREADNO→THREADS, MSG_NO, FROM_CUST^o→CUST_CONTACTS, FROM_EMP^o→EMPLOYEES, DATE, TEXT)

  Es gibt einen CHECK-Constraint, daß FROM_CUST und FROM_EMP nicht gleichzeitig null sein können. Ein anderer CHECK-Constraint besagt, daß eines von beiden Attributen null sein muß.

Bitte übersetzen Sie dieses relationale Schema in ein Schema im ER-Modell. Verwenden Sie dazu die Notation von Oracle Designer.

Auf der nächsten Seite gibt es Platz für die Lösung.

Lösung von Aufgabe 2

Aufgabe 3 (UML Klassendiagramme) 12 Punkte

Entwickeln Sie ein UML Klassendiagramm als Datenbank-Schema für folgende Anwendung:

- Es soll eine Datenbank für Fehler in einer Klausur entwickelt werden. Die Datenbank ist nur für eine einzige Klausur gedacht, und es sind keine Daten über die Klausur abzuspeichern.
- Die Klausur besteht aus einer Reihe von Aufgaben. Für jede Aufgabe ist die Aufgabennummer, die Überschrift und die Punktzahl abzuspeichern. Ein Teil der Aufgaben sind Multiple-Choice-Aufgaben. Für diese ist die Nummer der richtigen Antwort abzuspeichern (es ist immer genau eine Antwort korrekt).
- Für jeden Studierenden, der die Klausur geschrieben hat, sind Vor- und Nachname, sowie ggf. eine EMail-Adresse abzuspeichern. Nicht jeder Studierende hat eine Email-Adresse.
- Bei jeder Aufgabe wurden bestimmte Fehler gemacht. Für jeden Fehler sind ein Text abzuspeichern (Fehlerbeschreibung, wird den Studierenden ausgedruckt, die diesen Fehler gemacht haben) sowie der Punktabzug für diesen Fehler. Jeder Fehler gehört zu genau einer Aufgabe. Im Fall von Fehlern zu Multiple-Choice-Aufgaben ist die Nummer der jeweiligen falschen Antwort abzuspeichern (die der Studierende fälschlicherweise angekreuzt hat).
- Es ist abzuspeichern, welcher Studierende bei welcher Aufgabe welchen Fehler gemacht hat. Es ist möglich, daß ein Studierender bei der gleichen Aufgabe mehrere Fehler macht.

Sie brauchen außer den beschriebenen Attributen nur eine einzige Operation zu deklarieren, nämlich eine, die für jeden Studierenden die erreichte Punktzahl ausrechnet. Natürlich brauchen Sie die dazugehörige Methode nicht zu definieren.

Lösung zu Aufgabe 3

Aufgabe 4 (BCNF) 4 Punkte

Das Data Dictionary eines fiktiven DBMS enthält eine Relation

`KEYS(ID, TAB_NAME, COL_NO, COL_NAME)`

Dabei identifiziert `ID` den Schlüssel. `TAB_NAME` ist die Tabelle, auf die sich der Schlüssel bezieht. `KEYS` hat einen Eintrag für jedes Attribut `COL_NAME` des Schlüssels (Schlüssel können eine Attribut-Kombination sein). `COL_NO` ist die Nummer des Attributes innerhalb dieses Schlüssels. Es gelten folgende funktionalen Abhängigkeiten:

- `ID` → `TAB_NAME`
- `ID, COL_NO` → `COL_NAME`
- `ID, COL_NAME` → `COL_NO`

a) Was sind die (minimalen) Schlüssel dieser Relation? Kreuzen Sie alle minimalen Schlüssel an. Es können mehrere Antworten korrekt sein.

- ☐ `ID`
- ☐ `ID, TAB_NAME`
- ☐ `ID, COL_NO`
- ☐ `ID, COL_NAME`
- ☐ `ID, COL_NO, COL_NAME`
- ☐ `TAB_NAME, COL_NAME`

b) Ist die Relation in BCNF?

- ☐ Ja.
- ☐ Nein. Die funktionale Abhängigkeit „`ID` → `TAB_NAME`“ verletzt BCNF.
- ☐ Nein. Die funktionale Abhängigkeit „`ID, COL_NO` → `COL_NAME`“ verletzt BCNF.
- ☐ Nein. Die funktionale Abhängigkeit „`ID, COL_NAME` → `COL_NO`“ verletzt BCNF.

c) Angenommen, eine Data Dictionary Relation verletzt BCNF (nicht notwendigerweise diese Relation). Es handelt sich dabei aber tatsächlich um eine Sicht und keine abgespeicherte Relation. Wäre die BCNF-Verletzung dann schlecht? Sollte der DBMS-Hersteller die Definition der Sicht so modifizieren, daß BCNF nicht verletzt wird?

- ☐ Ja. Normalformen sind auch für Sichten wichtig.
- ☐ Nein.