

Probeklausur

Datenbanksysteme

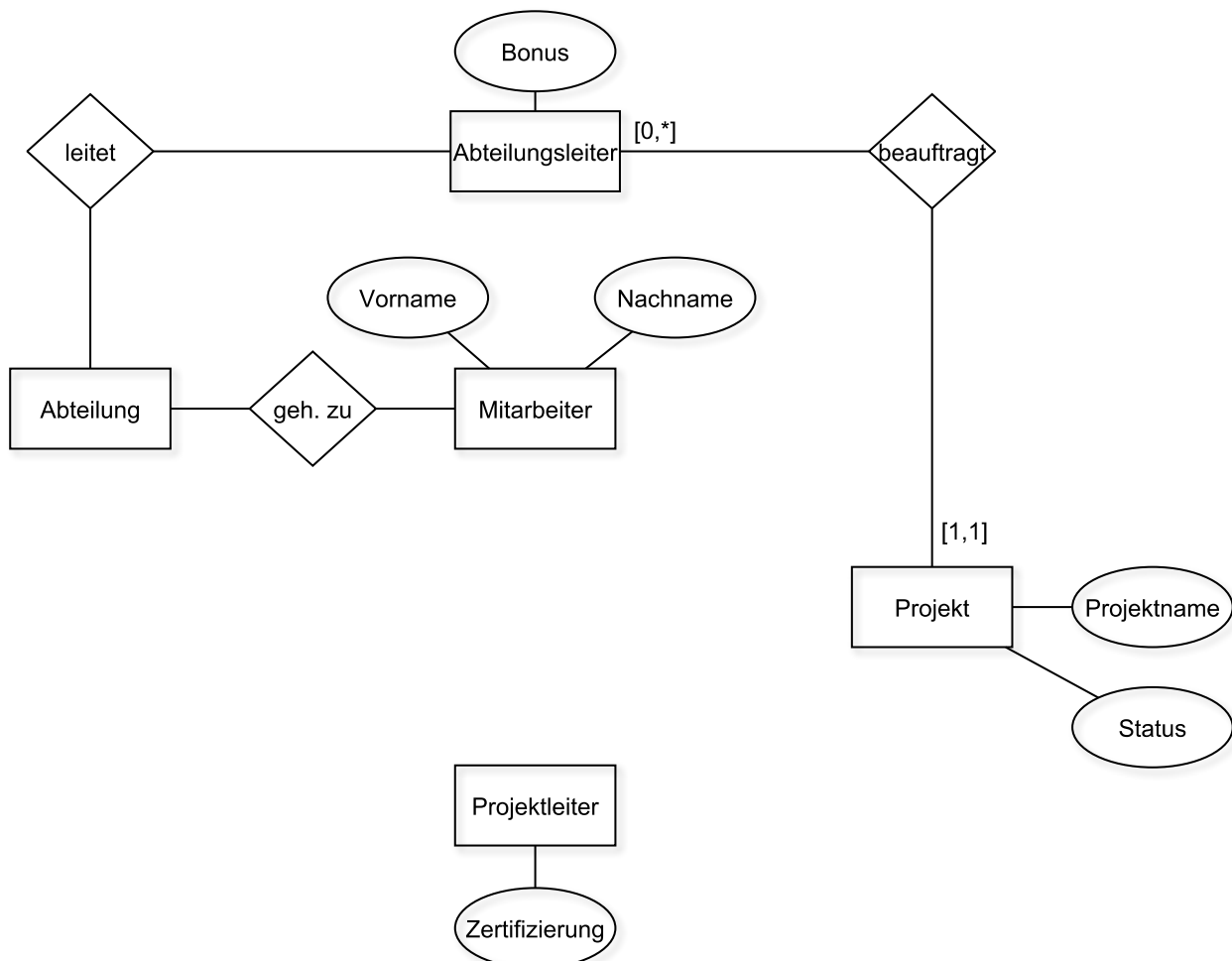
Besprechung am 31.01.2020!

Aufgabe 1 *ER-Modellierung*

(10 Punkte)

Im Folgenden wird ein Modell zur Projektverwaltung eines Unternehmens beschrieben. Vervollständigen Sie die unten aufgeführte ER-Modellierung unter Berücksichtigung der nachfolgenden Vorgaben. Geben Sie die Kardinalitäten in der [min, max]-Notation als Teilnahmezahl an und unterstreichen Sie die Schlüssel.

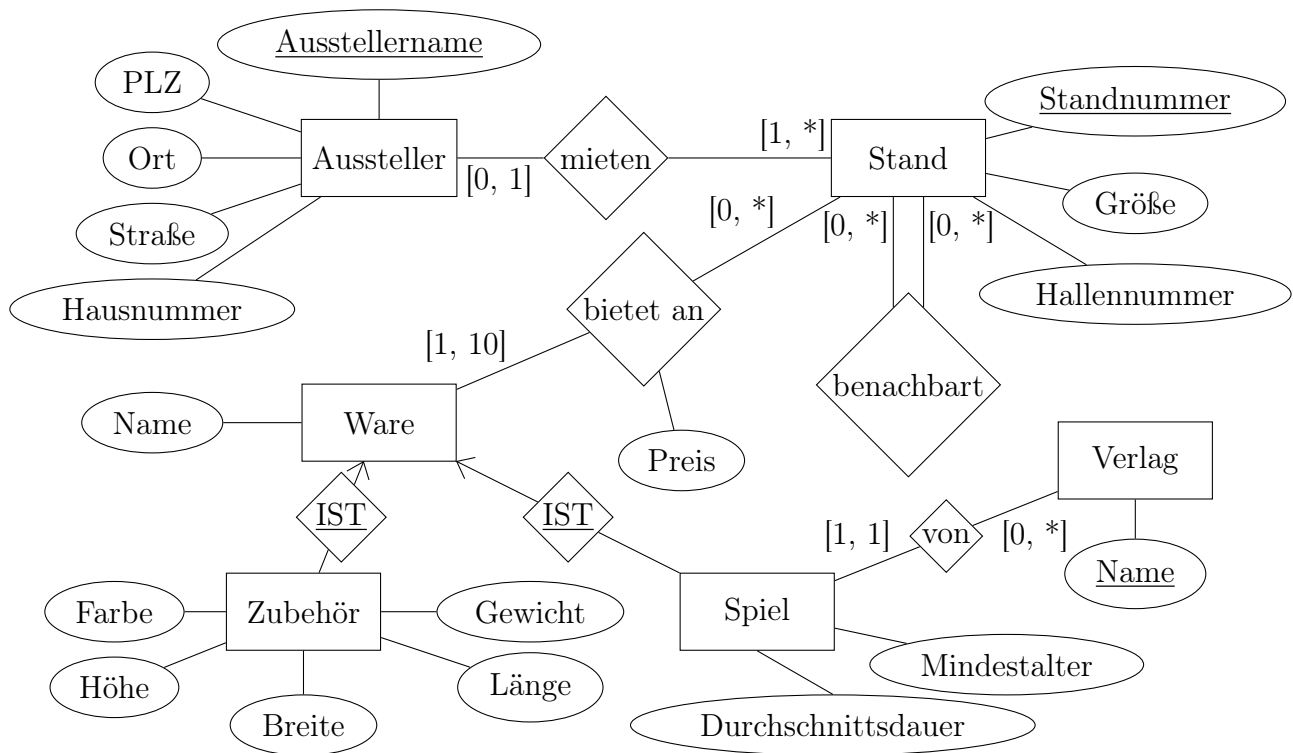
- Es gibt Mitarbeiter. Zu jedem Mitarbeiter werden Vor- und Nachname gespeichert.
- Jeder Mitarbeiter arbeitet in genau einer Abteilung. Nicht jede Abteilung hat Mitarbeiter.
- Jede Abteilung wird von genau einem Abteilungsleiter geleitet. Ein Abteilungsleiter ist ein Mitarbeiter und leitet genau eine Abteilung. Zusätzlich steht ihm ein variabler Bonus zu. Abteilungen haben einen eindeutigen Namen.
- Abteilungsleiter können beliebig viele Projekte in Auftrag geben. Ein Projekt wird von genau einem Abteilungsleiter in Auftrag gegeben.
- Projekte haben einen Projektnamen, einen Status und optionale Deadline.
- Weiterhin gibt es Projektleiter, die Mitarbeiter mit einer Zertifizierung sind.
- Projektleiter können beliebig viele Mitarbeiter für eine gewisse Stundenzahl für beliebig viele Projekte buchen. Dabei können Mitarbeiter für beliebig viele Projekte gebucht sein.
- Ein Projekt setzt sich aus beliebig vielen anderen Projekten zusammen.



Aufgabe 2 *Relationenmodell*

(10 Punkte)

Betrachten Sie das folgende ER-Modell für die Datenbank der diesjährigen Gesellschaftsspiele-Messe.



Überführen Sie das ER-Modell nun in das Relationenmodell. Beachten Sie dabei, folgende Punkte:

- Ihre Lösung sollte minimal sein. Führen Sie also Verschmelzungen durch, wo es möglich ist.
- Sie sollten Nullwerte im Relationenmodell vermeiden.
- Natürliche Schlüssel sind im ER-Modell unterstrichen dargestellt.
- Die Beziehung IST ist eine IST-Beziehung.
- Markieren Sie im Relationenmodell Primärschlüssel durch Unterstreichen und Fremdschlüssel durch Überstreichen.

Aufgabe 3 *Anfragesprachen*

(20 Punkte)

Gegeben sei das folgende Relationenmodell des Webportals *gesund-und-glücklich-speisen.de*:

REZEPT (RezeptID, Name, Zubereitungsbeschreibung, vegan, glutenfrei, laktosefrei, eingestelltVon)
ZUTAT (ZutatID, Name)
BENUTZER (Benutzername, Vorname, Nachname, Geburtsjahr, Email)
BEINHALTET (RezeptID, ZutatID, Mengenangabe)
BEWERTUNG (RezeptID, Benutzername, Punkte, Kommentar)

Im Relationenmodell gilt: Primärschlüssel sind unterstrichen, Fremdschlüssel sind überstrichen. Der Fremdschlüssel eingestelltVon bezieht sich auf den Primärschlüssel BENUTZER.Benutzername. Die Attribute *vegan*, *glutenfrei* und *laktosefrei* sind vom Typ Boolean, d.h. können den Wert *true* oder *false* annehmen. Das Attribut *Mengenangabe* ist vom Typ Text (String).

3.1 Formulieren Sie die folgenden Anfragen in der Relationenalgebra:

- (a) Geben Sie die Namen aller glutenfreien Rezepte aus.

- (b) Geben Sie alle Rezepte (RezeptID) aus, die die Zutat *Goji-Beere* enthalten.

- (c) Welche Benutzer (alle Attribute) haben noch kein Rezept eingestellt?

- (d) Welche Rezepte (alle Attribute) beinhalten beide Zutaten: *Kokosöl*, *Hanfsamen*?

3.2 Formulieren Sie die folgenden Anfragen im Tupelkalkül:

- (a) Geben Sie alle Benutzer (alle Attribute) im Alter zwischen 18 und 40 Jahren aus.
- (b) Geben Sie die Bewertungskommentare zu dem Rezept mit dem Namen *Grüner Frühstückssmoothie* aus.
- (c) Welche Rezepte (alle Attribute) wurden noch nie bewertet?

REZEPT (RezeptID, Name, Zubereitungsbeschreibung, vegan, glutenfrei, laktosefrei, *eingestelltVon*)
ZUTAT (ZutatID, Name)
BENUTZER (Benutzername, Vorname, Nachname, Geburtsjahr, Email)
BEINHALTET (RezeptID, ZutatID, Mengenangabe)
BEWERTUNG (RezeptID, Benutzername, Punkte, Kommentar)

(d) Welche Benutzer (alle Attribute) haben nur vegane Rezepte eingestellt?

3.3 Formulieren Sie die folgenden Anfragen in SQL:

(a) Listen Sie alle veganen Rezepte (alle Attribute) auf.

(b) Geben Sie alle Rezepte (alle Attribute) aus, die *Chia-Samen* enthalten.

(c) Geben Sie die Namen der bewerteten Rezepte mit ihrer Gesamtpunktezah aus. Das Ergebnis soll nach der Gesamtpunktezah absteigend sortiert sein.

- (d) Welche Benutzer (alle Attribute) haben weder ein Rezept eingestellt noch bewertet?

3.4 Formulieren Sie die folgenden Änderungsoperationen in SQL:

- (a) Der Benutzer *friedrich12* hat festgestellt, dass sein Porridge viel besser schmeckt, wenn er die getrockneten Früchte Physalis (ZutatID *2016*) weglässt. Ändern Sie sein Rezept (RezeptID *25*) so ab, dass die Physalis Frucht in seinem Rezept nicht mehr vorkommt.
- (b) *Hilda Heine* (Benutzername *hildhe87*) möchte ihr Lieblingssmoothie *Weizen-gras-Himbeer-Traum* einstellen. Die Zubereitungsbeschreibung des *veganen, glutenfreien* und *laktosefreien* Getränks lautet *Weizen-gras und Himbeeren mit Wasser mixen*. Fügen Sie dieses Getränk in die Datenbank ein. Die notwendigen bereits in der Datenbank vorhandenen Zutaten sind Weizen-gras (ZutatID *13*) und Himbeere (ZutatID *31*), die in der Menge *eine Handvoll* und *100 g* notwendig sind. Nehmen Sie an, dass diesem neuen Rezept die RezeptID *8426* vom Datenbanksystem zugewiesen wird.

Aufgabe 4 *Normalformen*

(10 Punkte)

Gegeben sei die Relation $T(M,N,O,P,Q,R)$ mit der folgenden Menge S von funktionalen Abhängigkeiten:

$$S = \{ \begin{array}{lll} Q & \rightarrow & P, \\ MN & \rightarrow & R, \\ N & \rightarrow & OQ, \\ MR & \rightarrow & N \end{array} \}$$

- (a) Bestimmen Sie für die Relation T alle Schlüsselkandidaten. Begründen Sie, warum es sich hier um Schlüsselkandidaten handelt und warum es keine weiteren geben kann.
- (b) Sei die 1. Normalform gegeben. Geben Sie für alle in Aufgabenteil (a) bestimmten Schlüsselkandidaten jeweils an, in welcher Normalform T ist und warum T nicht in der nächsthöheren Normalform ist.
- (c) Bestimmen Sie die Hülle der Attributmenge $\{R,N\}$ bzgl. S .

Aufgabe 5 *Transaktionen*

(15 Punkte)

5.1 Überprüfen Sie, ob die folgenden Schedules konfliktserialisierbar sind. Zeichnen Sie hierzu den Konfliktgraphen jedes Schedules und geben Sie zu jeder Kante des Konfliktgraphens ein Konfliktpaar an.

Wenn der Schedule serialisierbar ist, dann geben Sie alle äquivalenten seriellen Schedules an.

1. $S_1 := r_1(c)r_2(b)r_1(a)w_1(c)r_2(a)w_2(b)r_3(b)w_1(a)w_3(b)$

2. $S_2 := r_1(a)r_2(b)r_3(a)w_3(a)r_2(a)w_2(b)r_1(b)w_1(b)w_1(a)$

3. $S_3 := r_1(a)r_2(a)w_2(a)r_4(c)r_4(b)r_2(b)w_4(b)r_1(c)r_3(a)w_4(c)w_3(a)$

5.2 Gegeben seien die folgenden Transaktionen T_1 und T_2 :

```
 $T_1$ : read(X);  
      read(Y);  
       $X := X * (1.00 + Y)$ ;  
      write(X);  
  
 $T_2$ : read(Y);  
      read(X);  
      if  $X > 5000$  then  $Y := 0.5 * Y$ ;  
      write(Y);
```

Die Transaktion T_1 aktualisiert die Mitarbeitergehälter X entsprechend der Leistungszulage Y . Die Transaktion T_2 halbiert die Leistungszulage Y derjenigen Mitarbeiter, die mehr als 5000 verdienen.

1. Fügen Sie *Read-Lock*-, *Write-Lock*- und *Unlock*-Anweisungen in die Transaktionen T_1 und T_2 ein, so dass sie dem Zwei-Phasen-Sperrprotokoll entsprechen.

2. Könnte eine nicht serielle Ausführung der Transaktionen T_1 und T_2 eine Verklemmung (engl. Deadlock) produzieren? Falls ja, geben Sie ein kurzes Beispiel an! Falls nein, begründen Sie ihre Antwort!

3. Ist eine nicht serielle Ausführung der Transaktionen T_1 und T_2 gemäß dem konservativen strikten Zwei-Phasen-Sperrprotokoll (CS2PL) möglich? Falls ja, geben Sie ein kurzes Beispiel an! Falls nein, begründen Sie ihre Antwort!

Aufgabe 6 *PHP-Cookies und SQL-Injections*

(10 Punkte)

Nehmen Sie an, dass Sie eine Webseite besuchen, die Zustände mit Hilfe von PHP-Cookies realisiert. Daten, die in den Cookies gespeichert werden, werden für SQL-Anfragen verwendet. Die Webseite bietet ein Login an. Nach dem erfolgreichen Login mit dem Benutzer *Gast*, und dem Passwort *geheim* werden folgende Funktionsaufrufe durchgeführt:

```
setcookie("Benutzer", "Gast", time() + 3600, "/login");  
setcookie("Passwort", "geheim", time() + 3600, "/login");
```

Hinweis: Die angegebenen Zeitangaben sind in der Einheit Sekunden gegeben.

1. Nach wievielen Minuten laufen die Cookies ab?
2. Bei der Verwendung der PHP-Cookies argumentiert der Programmierer, dass diese vom Server gesetzt werden und deshalb nicht weiter überprüft werden müssen.
Warum liegt der Programmierer falsch?
3. Sie wissen, dass es einen Benutzer mit dem Benutzernamen *Administrator* gibt. Wie müssten die Cookies aussehen, damit Sie sich als Administrator einloggen können, ohne das entsprechende Passwort zu kennen?

4. Der Programmierer der Webseite sieht ein, dass seine Lösung Sicherheitsmängel birgt. Er schlägt die Verwendung einer Hashfunktion vor:

```
setcookie("Benutzer", "Gast", time() + 3600, "/login");  
setcookie("Passwort", hash('sha2', 'geheim'), time() + 3600, "/login");
```

Erläutern Sie den Nutzen einer (beliebigen) Hashfunktion. Behebt der Vorschlag des Programmierers das bestehende Problem? Begründen Sie Ihre Antwort!