

# Datenbank Woche 7

## 1. Anforderungen an die Datenbank Was muss gespeichert werden?

- Studenten
  - Matrikelnummer oder ID
  - Vorname
  - Nachname
  - Github Benutzername
  - Vorstellungs-Text (nullable)
  - Arbeitsproben (bspw. bereits bearbeitete Projekte)
- Gruppen
  - Gruppenname (unique constraint)
  - Gruppengründer
  - minimale Mitgliederanzahl
  - maximale Mitgliederanzahl
  - bei Gründung vereinbarte Regeln zur Abstimmung (enums?)
  - bei Gründung vereinbarte Frist für Verfall der Bewerbungen
- Teilnahme
  - Student
  - Gruppe
- Bewerbung
  - Student
  - Gruppe
  - Status
  - Ablaufszeit der Aufnahme
  - Nachricht (nullable)

## 2. Probleme

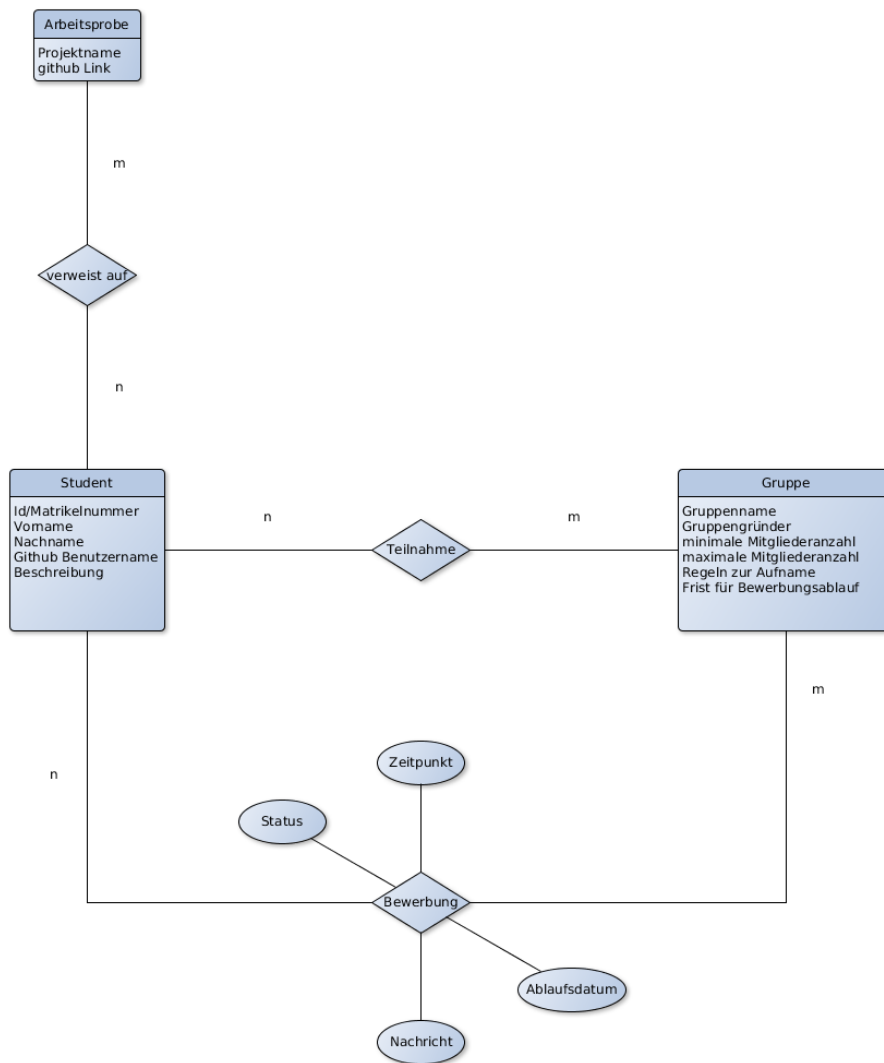
- Darstellung der Arbeitsproben:

Hier sind wahrscheinlich links zu github repositories sinnvoll. Allerdings ist dies unter Erhaltung der Atomarität/ersten NF problematisch. Möglich wäre hier als Alternative zu einem mehrwertigen Attribut eine eigenes Schema für die Arbeitsproben im Sinne einer n:m Beziehung zu erstellen. Der Nachteil dieser Methode ist, dass so bei jeder Darstellung eines einzelnen Studenten ein join mit diesem neuem Schema vollzogen werden muss (laufzeitaufwändig). Ein Vorteil dieser Methode ist, dass hier Optionalität dieser Beziehung ohne Gefährdung der Konsistenzbedingungen angeboten wird.

- Die minimale Mitgliederanzahl:  
Falls diese Zahl größer als 1 sein darf, kann keine valide Gruppe gegründet werden, ohne bei der Gründung direkt Mitglieder hinzuzufügen. Eine Lösung wäre ein Status-Attribut für Gruppen, welches jedoch für die Funktionalität der Gruppe unbedeutend wäre (da möglich sein muss der Gruppe beizutreten, um in den validen Status zu wechseln). Somit wäre dieses Attribut redundant. Eine andere Möglichkeit ist anzunehmen, dass die minimale Mitgliederzahl immer 1 beträgt.
- Zählen von Mitgliedern:  
Ein Attribut, welches die Anzahl der Gruppenmitglieder festhält ist redundante Information, würde jedoch die Notwendigkeit eines joins verhindern. In meinen Augen ist das hier nicht besonders wertvoll, da bei der Anzeige einer Gruppe neben der Anzahl der Mitglieder auch die Identität der Mitglieder interessant ist, wofür dieser join sowieso benötigt wird.
- Ablaufszeit der Aufnahmebestätigung:  
Wenn man die Aufnahmebestätigung nicht in einem eigenen Schema speichern möchte, ergibt sich das Problem, dass das Attribut im Laufe der Zeit seine Bedeutung ändert. Erst wäre es der Ablauf der Bewerbung, welche von der Gruppe bestimmt wird, dann nach der Bestätigung der Aufnahme, die Zeit in der Bewerber noch der Gruppe beitreten kann, bevor die Bestätigung verfällt. Da das Attribut jedoch in beiden Fällen die Lebensdauer der Entität beschreibt ist das in meinen Augen akzeptabel.
- Festlegen der Regeln für Aufnahme und Ausschluss:  
Es ist schwer dieses Attribut adäquat zu speichern. Meine einzige momentane Idee ist es, die Regeln durch Strings, enums oder durch eine Zahl zu kodieren.
- Datentyp der Frist für Bewerbungsverfall:  
Es gibt meines Wissens nach keinen passenden Datentyp für Zeitmaße. Man kann hier einen numerischen Wert verwenden und eine bestimmte Einheit als bekannt voraussetzen.

### 3. ER-Modell

Ein erster Lösungsentwurf.



#### 4. Sichten

Bisher ist mir nicht vollkommen klar wie viel uns Sichten nutzen und wie viel der Datenaggregation von der Anwendung übernommen werden kann. Die Sichten können bei Bedarf noch später hinzugefügt werden.