

## 1. Datenbankentwurf

**Gegeben in L1**

**Abgabe in: L3**

Denke an eine Anwendung, die eine Datenbank benötigt (keine Beispiele aus der Vorlesung). Modelliere die relationale Struktur der Daten und implementiere diese in einer SQL Server Datenbank. Berücksichtige dabei folgende minimale Anforderungen:

- Die Datenbank muss mindestens 10 Tabelle enthalten und mindestens eine one-to-many Beziehung (1:M) und eine many-to-many Beziehung (M:N) modellieren.
- Finde wenigstens eine numerische Spalte in einer Tabelle (außer IDs), damit ihr alle Abfragen in Lab2 schreiben könnt.
- Mindestens eine Tabelle muss einen zusammengesetzten Primärschlüssel haben.
- Alle Tabellen müssen Primärschlüssel haben.

Füge ein paar Daten in die Tabellen ein. Diese müssen möglichst „echte“ Daten sein, die man dann abfragen kann.

Zum Beispiel für die Tabelle `Vorlesung` (`VorlID`, `Titel`, `ECTS`) :

- (H1, asfj, 2) – ist kein gutes Tupel → **nicht erlaubt!**
- (BD1, Datenbanken, 5) – ein gutes Beispiel

Bevor ihr anfängt zu arbeiten, schreibe eine E-Mail an dem Tutor mit dem ausgewählten Thema für die Datenbank und eine kurze Beschreibung. **Jeder muss ein anderes Thema wählen.** Im Falle eines Konflikts gewinnt derjenige, der die E-Mail zuerst geschrieben hat, und der zweite muss eine andere Anwendung finden.

**Bem. In Lab2 müssen 2 Tabellen mit SQL Anweisungen erstellt werden. D.h. ihr könnt zwei der Tabellen direkt mit Anweisungen erstellen und die Anweisungen speichern (siehe Lab2).**

Sei die Beziehungsmenge ( $E_1, E_2$ )

- one-to-one (1:1) Beziehung – jede Entität aus  $E_1$  steht in Beziehung zu höchstens einer Entität aus  $E_2$
- one-to-many Beziehung (1:M) – alle Entitäten aus  $E_1$  stehen mit beliebig viele Entitäten aus  $E_2$  in Beziehung, aber jede Entität aus  $E_2$  steht mit maximal einer Entität aus  $E_1$  in Beziehung
- many-to-many Beziehung (M:N) – alle Entitäten aus  $E_1$  stehen mit beliebig viele Entitäten aus  $E_2$  in Beziehung und umgekehrt

