

## Meilenstein 1a: Anforderungsanalyse News-Portal

Das Startup *UNITALK*, das es Studierenden ermöglichen möchte anonym zu kommunizieren, hat mich mit der Erstellung einer Datenbank beauftragt.

Die Datenbank soll der Verwaltung veröffentlichter Artikel (eindeutige Artikelnummer, Veröffentlichungsdatum, ist live / aufrufbar), welche eindeutig einer Universität (Name, Anzahl Studierende) zugeordnet werden, dienen. Personen (Vorname, Nachname, DOB\*) können entweder Autoren (Autorennummer, tätig seit, Kürzel) oder Mitarbeiter (eindeutige Sozialversicherungsnummer, Lohn, Einstellungsdatum) sein. Autoren sprechen eine oder mehrere Sprachen (eindeutiges Kürzel, Name, Zeichen, RTL\*\*), in welchen sie sich an Artikeln beteiligen können. Autoren schreiben nur für genau eine Universität, welcher sie untergeordnet sind.

Artikel werden in einer Sprache und für eine Universität veröffentlicht. Es können auch mehrere Autoren derselben Universität an einem Artikel beteiligt sein, jedoch hat jeder Artikel einen Hauptautor, welcher die Verantwortung für den Inhalt übernimmt. Artikel können maximal einen anderen Artikel referenzieren, jedoch kann ein Artikel von beliebig vielen Artikeln referenziert werden.

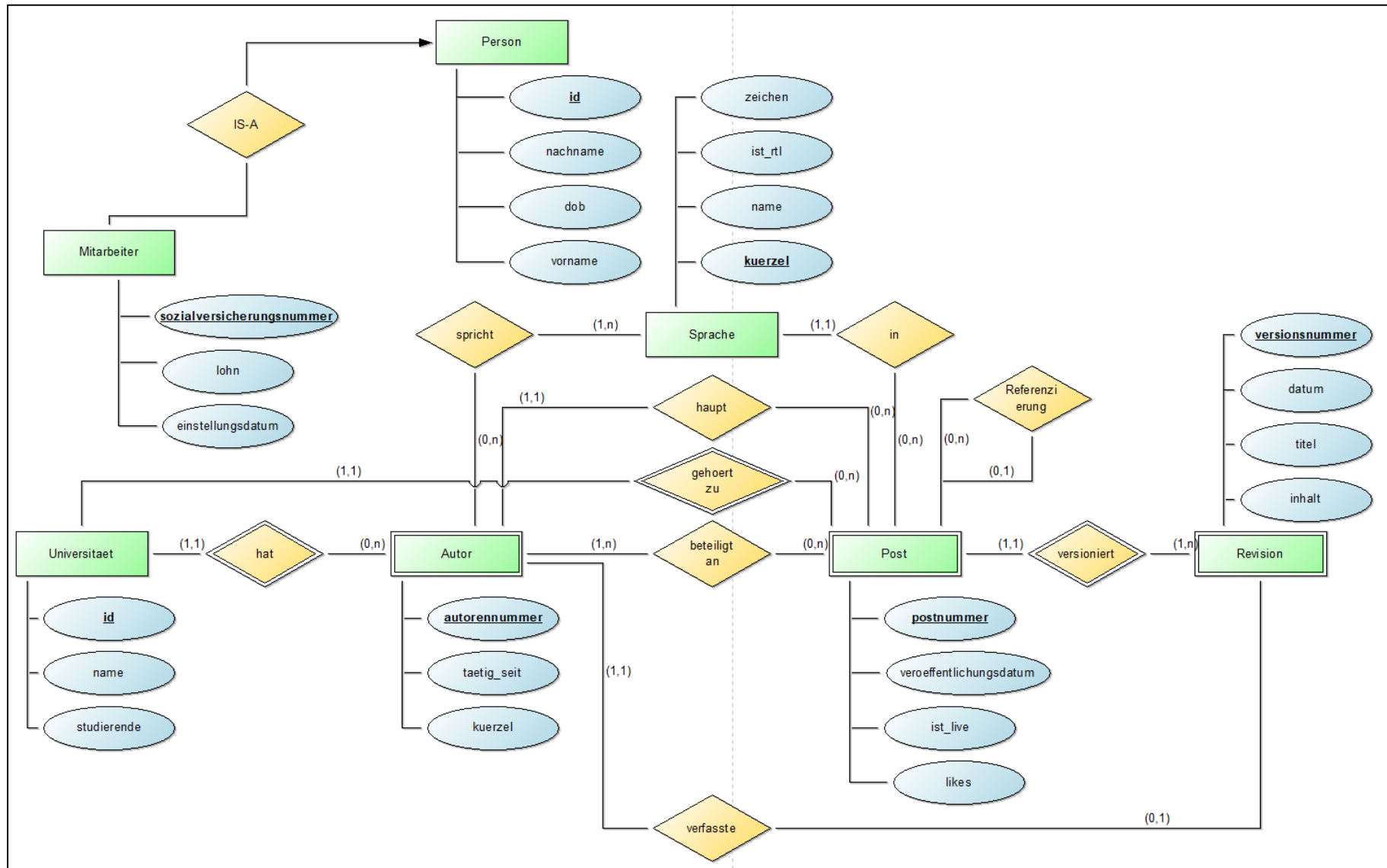
Weiter sollen Revisionen (eindeutige Versionsnummer, Titel, Inhalt, Datum) von Artikeln verwaltet werden, welche Auskunft darüber geben, welcher Autor zu welcher Zeit einen Artikel erstellt oder aktualisiert hat.

---

\*DOB (date of birth) steht für das Geburtsdatum

\*\*RTL (right to left) bezeichnet die Leserichtung von Text

## Meilenstein 1b: Konzeptioneller Entwurf



## Meilenstein 2: Logischer Entwurf

<b>person</b> ( <u>id</u> , nachname, vorname, dob) PK: id → einziger Schlüsselkandidat
<b>mitarbeiter</b> ( <u>snr</u> , person_id, lohn, einst_datum) PK: snr → einziger Schlüsselkandidat FK: person_id ◇ person
<b>sprache</b> ( <u>kuerzel</u> , ist_rtl, name, zeichen) PK: kuerzel → einziger Schlüsselkandidat
<b>universitaet</b> ( <u>id</u> , name, studierende) PK: id → einziger Schlüsselkandidat
<b>autor</b> ( <u>id</u> , person_id, uni_id, taetig_seit, kuerzel) PK: id → einziger Schlüsselkandidat <del>FK: person_id ◇ person</del> FK: uni_id ◇ universitaet
<b>autor_hat_sprache</b> ( <u>autor_id</u> , <u>sprache_kuerzel</u> ) PK: autor_id, sprache_id → einziger Schlüsselkandidat FK: autor_id ◇ autor FK: sprache_kuerzel ◇ sprache
<b>post</b> ( <u>id</u> , <u>uni_id</u> , hauptautor_id, ref_artikel_id, sprache_kuerzel, datum, ist_live) PK: id, uni_id → einziger Schlüsselkandidat FK: hauptautor_id ◇ autor FK: ref_post_id ◇ post FK: sprache_kuerzel ◇ sprache FK: uni_id ◇ universitaet
<b>autor_hat_post</b> ( <u>autor_id</u> , <u>post_id</u> ) PK: autor_id, post_id, uni_id → einziger Schlüsselkandidat FK: autor_id ◇ autor FK: post_id ◇ post
<b>revision</b> ( <u>post_id</u> , <u>versnr</u> , uni_id, autor_id, datum, titel, inhalt) PK: post_id, versnr → einziger Schlüsselkandidat FK: uni_id ◇ universitaet FK: autor_id ◇ autor

## **Meilenstein 4: Implementierung**

1. Datenbank:  
Die Datenbank wird mit Oracle SQL Developer erstellt werden, dazu dient das ER-Diagramm.
2. Dummy Data:  
Ein Java-Programm befüllt die Datenbank mit Test-Daten. Davor leert es die Datenbank immer. Dieses Programm lasse ich lokal, auf meinem Rechner laufen.
3. API:  
Wird in PHP programmiert und ist dafür zuständig, dem Frontend die Interaktion mit der Datenbank zu ermöglichen.
4. Frontend / Geschäftslogik:  
Soll in Angular programmiert werden. Es soll sich um eine sehr simple Applikation halten.
5. Deployment:  
Das Frontend, sowie die API sollen über Git auf Almighty deployed werden (git pull). Ich werde versuchen, einen GitHub WebHook einzurichten, jedoch kann dies aus Berechtigungsgründen fehlschlagen.  
<http://wwwlab.cs.univie.ac.at/~christophb77/dbs/>