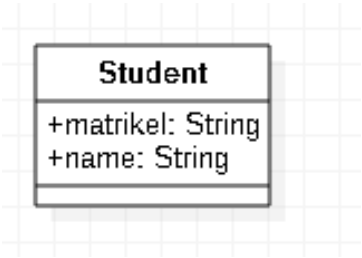
Persistenz (Speichern)

1. Relationale Datenbanken (SQL)
2. (Text)Dateien
    1. JSON
    2. XML
    3. csv
3. NOSQL
    1. Dokumentbasierte DB: CouchDB, MongoDB
    2. Key-Value-Datenbanken
    3. Wide Column DB (Casandra) (Facebook)

O/R-Mapping (Objekt-Relationales-Mapping)

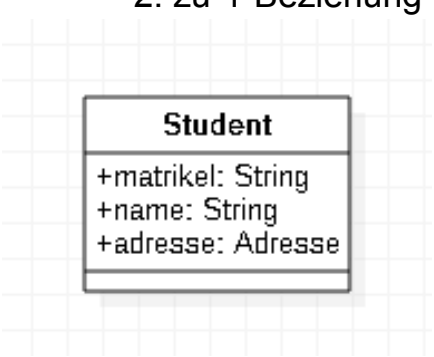
1. Einzelne Klasse

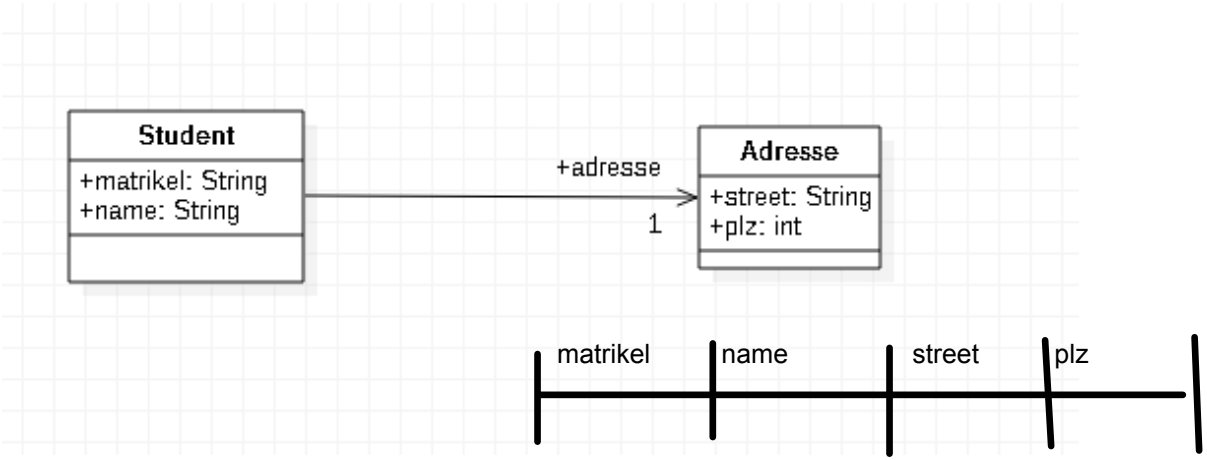


Eine Tabelle pro Klasse

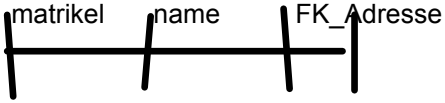
Jedes Attribut eine Spalte

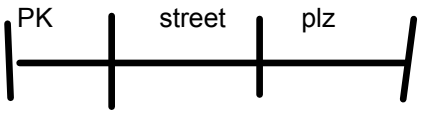
2: zu-1-Beziehung



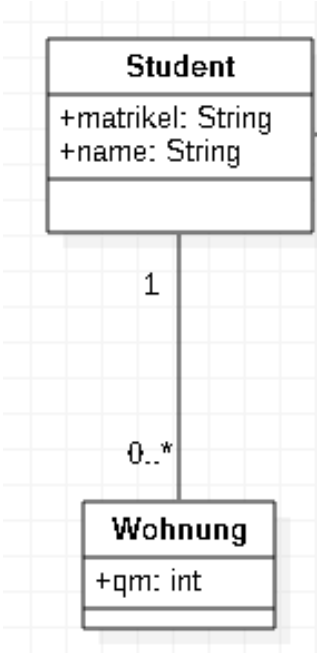


1. Möglichkeit: Eine Tabelle, die alle Attribute enthält
2. Möglichkeit: Zwei Tabellen, wobei die Tabelle (hier Student) einen FK auf die andere hat





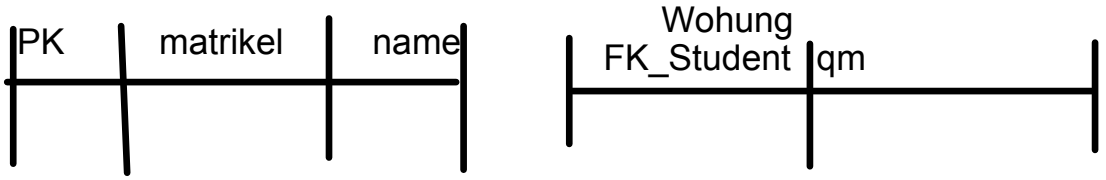
3. 1:n-Beziehungen



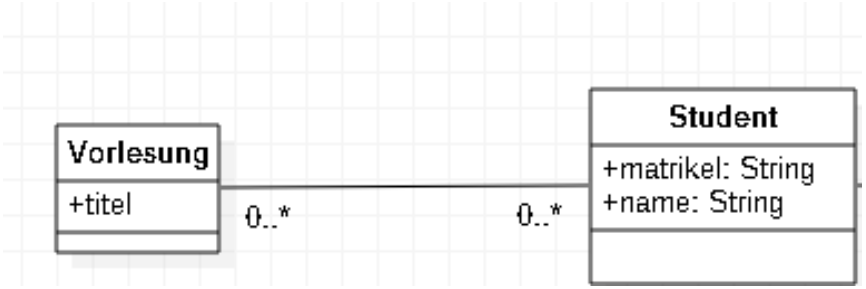
Zwei Tabellen (Student, Wohnung)

Fremdschlüssel auf der Wohnung

Student



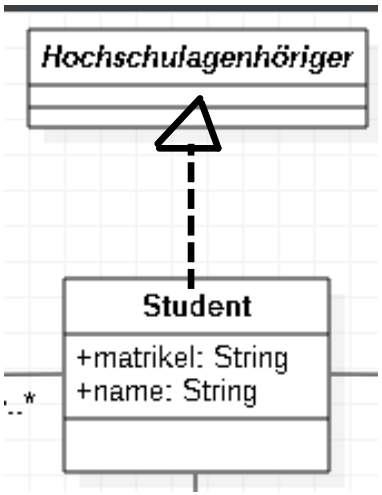
4. n:m Beziehungen



3 Tabellen

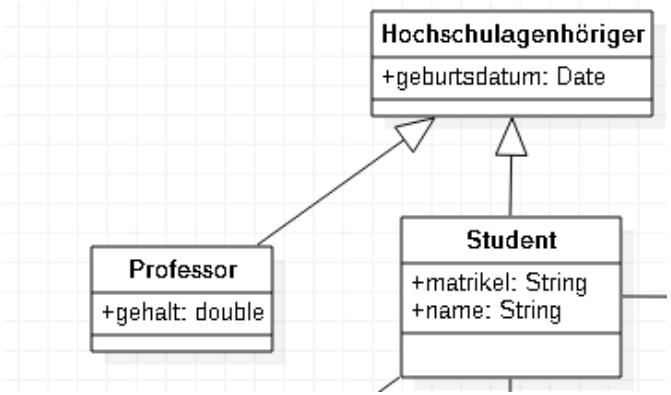
1. Student (PK, matrikel, name)
2. Vorlesung (PK, titel)
3. Schlüsseltabelle (FK_Student, FK_Vorlesung)

5. Interfaces

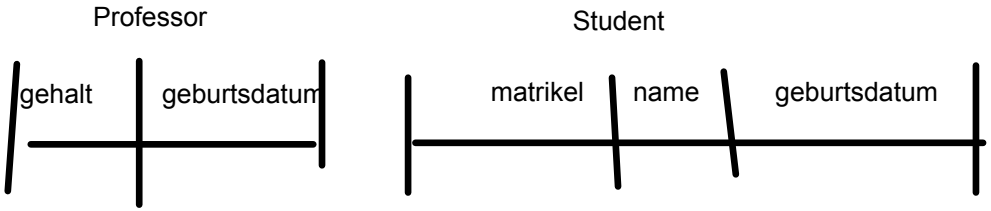


Interfaces werden nicht in der DB repräsentiert

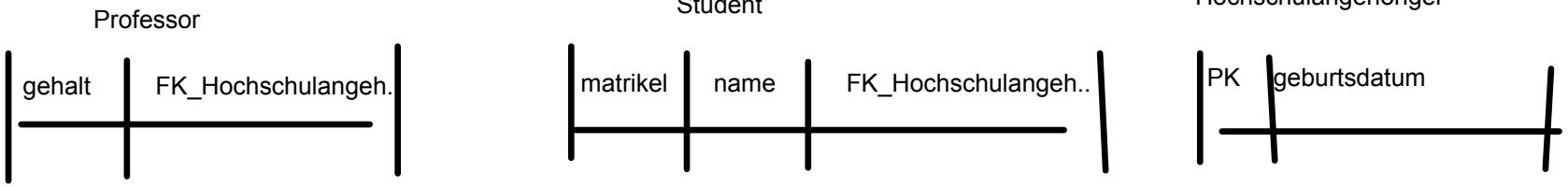
6. Vererbungsbeziehungen



1. Variante: 2 Tabellen



2. Variante: 3 Tabellen



3. Variante: 1 Tabelle

geburtsdatum	gehalt	matrikel	name	rolle
1.1.1999	--	123	Alex	Student
31.12.1876	3 Mio.	---	Richy	Professor

	Vorteile	Nachteile
1 Tabelle	keine Joins, schneller	keine DB-Sicherheit wie z.B. null-Prüfungen (muss im Code gemacht werden, Wartbarkeit?) Zusätzliche Spalte hier "Rolle""
2 Tabellen	Tabellen sind anhand der Klassen zu identifizieren	Wartbarkeit: jede Änderung der vererbende Tabelle muss für jede erbetende Tabelle realisert werden
3 Tabellen	Null-Prüfungen, O/R-Mapping ist (1 Tabelle pro Klasse)	Performanz (viele Joins)