

Semesteraufgabe

1. SZENARIO

Datenbankmodellierung einer Schulverwaltung

Der Gegenstand unseres Semesterprojektes ist die Erstellung eines Entity Relation Models (ER Modell) sowie die anschließende Implementierung der Datenbank für eine Grundschule.

2. VORGEHEN

1. Aufnehmen der Anforderungen

Zu Beginn des Projektes wurden die Anforderungen der Grundschule gesammelt. Unter anderem soll die Datenbank in der Lage sein, Lehrer, Schüler, sowie Klassen abzubilden. Darüber hinaus soll die Möglichkeit bestehen einer Klasse, abgesehen von Schülern, auch einen Klassensprecher und einen Klassenlehrer zuzuweisen. Lehrer wiederum können verschiedene Klassen unterrichten und Fächer lehren. Wie für Grundschulen übrig, sollen die einzelnen Klassen spezifische Klassenräume besitzen. Räume sollen zudem in einer Raumbtabelle erfasst werden. Diese beispielhaften Anforderungen wurden anschließend in Entitäten und Relationen übersetzt und in ein ER Modell überführt. Dieses befindet sich im Anhang: Abb. 1

2. Entitätstypen

Die Überführung in Entitäten ergab beispielsweise die Entitätstypen Lehrer und Schüler. Neben einem spezifischen LehrerID Attribut, welches als Primärschlüssel dient, besitzen Lehrer die Attribute Vorname und Nachname. Schüler hingegen besitzen ebenfalls ein SchülerID Attribut als Primärschlüssel. Zusätzlich besitzen sie noch einen Fremdschlüssel KlassenID, damit man sie einer Entität vom Typ Schulklasse zuordnen kann. Zur weiteren Identifizierung der Schüler während Ihrer Schullaufbahn wird das Attribut Schuljahr ergänzt.

3. Relationen

Zur Abbildung des Schulbetriebs werden Relationen eingeführt. Schüler und Lehrer sind direkt über die Relation Prüfen verbunden. Diese Relation ist eine N:M-Relation, da Schüler von mehreren Lehrern geprüft werden können und einzelne Lehrer auch mehrere Schüler prüfen können. Zusätzlich verfügt die Relation über die eigenen Attribute Fach und Note. Fach ist hierbei ein Fremdschlüssel, um zu spezifizieren, in welchem Fach eine Prüfung stattfindet.

4. Tabellen

Anschließend erfolgte die Übertragung der Entitätstypen und Relationen in eine Tabellenform, um sie in eine Datenbank implementieren zu können. Die erzeugten Tabellen werden im Anhang dargestellt. Zur beispielhaften Veranschaulichung dient der Entitätstyp Lehrer:

Lehrer			
<u>lehrer_id</u>	vorname	nachname	email

Die Relationen wurden ebenfalls in Tabellen überführt. Beispielhaft dargestellt durch die N:M-Relation Prüfen. Diese besitzt keinen Primärschlüssel, jedoch drei Fremdschlüssel, um auf den prüfenden Lehrer, den zu prüfenden Schüler sowie das zu prüfende Fach zu verweisen.

pruefen			
<u>lehrer_id</u>	<u>schueler_id</u>	<u>fach_id</u>	note

5. Implementierung

Anschließend erfolgte die Implementierung. Hierfür wurde MariaDB als Datenbank und NodeJS als serverseitige Laufzeitumgebung verwendet. Durch die Integration des ExpressJS Frameworks werden die Client-seitigen HTTP Requests abgefangen und von der Datenbank verarbeitet. Das Bootstrap Framework dient zur effizienten Implementierung einer nutzerorientierten Bedienoberfläche. Zur weiteren effizienten Gestaltung des Produktionsprozess wird die Template Engine pug verwendet, welche die dynamisch erzeugten HTML Seiten rendert.

6. Funktionsweise

Beispielhaft werden nachfolgend zwei Relationale Algebra und SQL Befehle aufgezeigt.

Der erste zeigt die Relation »unterrichtet« mit den passenden lesbaren Namen der beteiligten Entitäten.

$$\pi_{\text{lehrer.l_nachname, fach.fachname, schulklasse.jahrgang, schulklasse.bezeichner}}((($$

$$\text{unterrichtet} \bowtie_{\text{unterrichtet.lehrer_id=lehrer.id}} \text{lehrer})$$

$$\bowtie_{\text{unterrichtet.fach_id=fach.id}} \text{fach})$$

$$\bowtie_{\text{unterrichtet.klassen_id=schulklasse.id}} \text{schulklasse})$$

```

1 SELECT lehrer.l_nachname, fach.fachname, schulklasse.jahrgang, schulklasse.bezeichner
2 FROM unterrichtet
3 INNER JOIN lehrer
4 ON unterrichtet.lehrer_id = lehrer.id
5 INNER JOIN fach
6 ON unterrichtet.fach_id = fach.id
7 INNER JOIN schulklasse
8 ON unterrichtet.klassen_id = schulklasse.id

```

Diese Funktion zählt alle Lehrer und gibt die Anzahl in einer neuen Spalte aus

$$\pi_{\text{anzahl_lehrer}} \gamma_{(\text{COUNT}(\text{lehrer.id}) \rightarrow \text{anzahl_lehrer})} \text{lehrer}$$

```

1 SELECT COUNT(lehrer.id) as anzahl_lehrer
2 FROM lehrer

```

Lehrer			
<u>lehrer_id</u>	vorname	nachname	email

Schüler					
<u>schueler_id</u>	s_vorname	s_nachname	<u>klassen_id</u>	s_schuljahr	s_email

Fach	
<u>fach_id</u>	fachname

Raum			
<u>raum_id</u>	raumart	sitzplaetze	raumbezeichner

schulklasse				
<u>klassen_id</u>	<u>klassenlehrer_id</u>	<u>klassensprecher_id</u>	jahrgang	bezeichner

lehrt			
<u>lehrer_id</u>	<u>fach_id</u>	wochenstunden	schuljahr

pruefen			
<u>lehrer_id</u>	<u>schueler_id</u>	<u>fach_id</u>	note

pruefen			
<u>lehrer_id</u>	<u>schueler_id</u>	<u>fach_id</u>	note

raumverwendung	
<u>klassen_id</u>	<u>raum_id</u>

unterrichtet		
<u>fach_id</u>	<u>klassen_id</u>	<u>fach_id</u>

Abb.1

