

Zusammenfassende Ausarbeitung

(B) Modelldokumente

1. Miniweltbeschreibung DB-Projektwoche

Die Datenbank verwaltet die Kurse im Rahmen der jährlichen Projektwoche.

Jeder Kurs ist genau einem Unterrichtsfach zugeordnet, nicht alle Fächer bieten jedoch Kurse an.

Jedes Fach hat genau einen Fachverantwortlichen. Jeder Fachverantwortliche ist nur für ein Fach zuständig. Jede Lehrkraft unterrichtet genau zwei Fächer.

Ein Kurs wird von einer Lehrkraft durchgeführt, nicht alle Lehrkräfte bieten Kurse an.

Die Kurse finden von Montag bis Freitag statt und können an unterschiedlichen Tagen zu unterschiedlichen Zeiten beginnen und enden.

Zwei Lehrkräfte werden in den Pausen zudem an verschiedenen Stellen im Schulhaus zur Aufsicht eingesetzt. Die Aufsicht kann sich auch auf die Begleitung von Exkursionen der Kurse beziehen.

Alle Schüler besuchen genau einen Kurs. Für das Zustandekommen eines Kurses sind mindestens zehn Teilnehmende notwendig. Einige Schülerinnen und Schüler sind verantwortlich für einen Kurs und unterstützen die Lehrkräfte bei der Bewerbung.

2. ER-Modell DB-Projektwoche

Das ER-Modell findet sich in der Datei *prowo-er-modell.jpg*

3. Integritätsbedingungen

Es muss sichergestellt sein, dass bei Exkursionen sowohl die begleitenden Lehrer als auch der Kurs gleichzeitig am Ort zur gleichen Zeit sind.

Weiterhin muss sichergestellt sein, dass ein Lehrer nicht zeitgleich zu einem Kurs und einer Exkursion (die nicht zu seinem Kurs gehört) eingeteilt ist.

Auch die Bedingung, dass jeder Kurs einen verantwortlichen Schüler benötigt, muss sichergestellt werden.

Zudem wird die Kapazität des Ortes nicht abgebildet, d. h. es muss sichergestellt sein, dass ein Ort nicht durch mehrere Kurse gleichzeitig belegt wird.

Es muss sichergestellt werden, dass die Mindestanzahl zum Zustandekommen eines Kurses zehn Schüler beträgt (aus Gründen der Übersichtlichkeit wurde dies in der Beispielpopulation nicht berücksichtigt).

Folgendes lässt sich zudem nicht abbilden.

Wenn zwei Kurse gleichzeitig eine Exkursion vornehmen und für beide Kurse zusätzliche begleitende Lehrer als Aufsicht eingeteilt werden, lässt sich nicht zuordnen, welcher Lehrer Kurs 1 bzw. Kurs 2 begleitet.

4. Funktionale Abhängigkeiten

Wesentliche funktionale Abhängigkeiten:

(1) Kuerzel -> {Lehrer.Vorname,Lehrer.Nachname}

(2) FachID -> {Fach.Name}

(3) KursID -> {Kurs.Name,'(2)'}

(4) SchuelerID -> {Schueler.Vorname,Schueler.Nachname,Klasse,istVerantwortlich,'(3)'}

Und damit: '(4)' -> '(3)' -> '(2)'

(5) OrtID -> {Ort.Name}

Triviale Abhängigkeiten der Form $AB \rightarrow A$ werden nicht mit angegeben.

- (6) KursID, Termin.Tag, Termin.Start $\rightarrow \{ \text{'(5)', Termin.Ende} \}$
- (7) OrtID, KursID, Termin.Tag, Termin.Start $\rightarrow \{ \text{Termin.Ende} \}$
- (8) OrtID, KursID, Aufsicht.Tag, Aufsicht.Start $\rightarrow \{ \text{Aufsicht.Ende} \}$

5. Relationenmodell

- 1. Lehrer (Kuerzel:VARCHAR[4],Vorname:VARCHAR[],Nachname:VARCHAR[])
- 2. Kurs (KursID:INTEGER,Kuerzel:VARCHAR[4],Name:VARCHAR[])
- 3. Fach (FachID:INTEGER,Name:VARCHAR[])
- 4. unterrichtet (FachID:INTEGER,Kuerzel:VARCHAR[4])
- 5. Ort (OrtID:INTEGER,Name:VARCHAR[])
- 6. Schueler (SchuelerID:INTEGER,KursID:INTEGER,Vorname:VARCHAR[],
Nachname:VARCHAR[], Klasse:VARCHAR[],istVerantwortlich:INTEGER)
- 7. Termin (KursID:INTEGER,OrtID:INTEGER,Tag:INTEGER,Start:INTEGER,Ende:INTEGER)
- 8. Aufsicht (Kuerzel:VARCHAR[4],OrtID:INTEGER,Tag:INTEGER,Start:INTEGER,
Ende:INTEGER)
- 9. Zugehoerigkeit (FachID:INTEGER,KursID:INTEGER)

6. Transformation des ER-Modells in das Relationenmodell

1. Schritt: Transformation der Entitätstypen

Alle Entitäten des ER-Modells mit ihren Attributen werden in Relationen des Relationsmodells transformiert.

Daraus entstehen zunächst folgende Relationen:

- | | |
|--|--------------------------------|
| a) Lehrer (Kuerzel,Vorname,Nachname) | -> Primärschlüssel: Kuerzel |
| b) Kurs (KursID,Name) | -> Primärschlüssel: KursID |
| c) Fach (FachID,Name) | -> Primärschlüssel: FachID |
| d) Ort (OrtID,Name) | -> Primärschlüssel: OrtID |
| e) Schueler (SchuelerID,Vorname,Nachname,Klasse) | -> Primärschlüssel: SchuelerID |

2. Schritt: Transformation der Beziehungstypen

- | | |
|--|----------------------------------|
| 1:1-Beziehungstyp: 'besucht' | -> angehängt zu 'Kurs' |
| 1:1-Beziehungstyp: 'verantwortet' | -> angehängt zu 'Fach' |
| 1:1-Beziehungstyp: 'findet statt' | -> neue Entität 'Termin' |
| // Die Relation hat die Attribute 'Tag','Start' und 'Ende', daher wird eine neue Entität benötigt. | |
| 1:n-Beziehungstyp: 'beaufsichtigt' | -> neue Entität 'Aufsicht' |
| // Die Relation hat die Attribute 'Tag','Start' und 'Ende', daher wird eine neue Entität benötigt. | |
| 1:n-Beziehungstyp: 'belegt' | -> angehängt zu 'Schueler' |
| 1:n-Beziehungstyp: 'gehört zu' | -> neue Entität 'Zugehoerigkeit' |
| m:n-Beziehungstyp: 'unterrichtet' | -> neue Entität 'unterrichtet' |

3. Schritt: Transformationsergebnis

- | | |
|---|-----------------------------|
| 1. Lehrer (Kuerzel,Vorname,Nachname) | -> Primärschlüssel: Kuerzel |
| 2. Kurs (KursID,Kuerzel,Name) | |
| -> Primärschlüssel: KursID; Fremdschlüssel: Kuerzel | |
| 3. Fach (FachID,Name) | |
| -> Primärschlüssel: FachID | |

- | | |
|---|--|
| 4. unterrichtet (FachID,Kuerzel) | -> Primärschlüssel: Kuerzel,FachID |
| 5. Ort (OrtID,Name) | -> Primärschlüssel: OrtID |
| 6. Schueler (SchuelerID,KursID,Vorname,Nachname,Klasse,istVerantwortlich) | -> Primärschlüssel: SchuelerID; Fremdschlüssel: KursID |
| 7. Termin (KursID,OrtID,Tag,Start,Ende) | -> Primärschlüssel: KursID,OrtID,Tag,Start |
| 8. Aufsicht (Kuerzel,OrtID,Tag,Start,Ende) | -> Primärschlüssel: Kuerzel,OrtID,Tag,Start |
| 9. Zugehoerigkeit (FachID,KursID) | -> Primärschlüssel: FachID,KursID |

7. Daten Beispielpopulation

Die einzelnen Tabellen der Beispielpopulation finden sich in der Datei: *prowo-<table>.csv*.

Für die übersichtliche Darstellung der gesamten Beispielpopulation siehe *prowo-Daten.pdf*

8. Text zu Anfragen

Die einzelnen Anfragen sind den Dateien *prowo-query<nn>.txt* zu entnehmen

9. Übersicht der Anfragen

Anfragen ohne Aggregatsfunktionen

1. "Liste aller Kursnamen"
2. "Schuelernamen der Klasse 8a"
3. "Lehrerkuerzel + Fach aller Fachverantwortlichen mit Nawi-Fach"
4. "Alle Fachverantwortlichen mit jeweiligem zugehoerigen Fach"
5. "Alle Raeume (und Exkursions- und Aufsichtsorte), die am Montag nicht belegt sind"
6. "Die Kuerzel der Lehrer, die wenigstens einmal in der Woche arbeiten (Kurs, Exkursion, Aufsicht)"
7. "Alle Lehrer, die nicht in der Projektwoche eingesetzt werden (nicht einmal eine Aufsicht)"
8. "Alle Lehrer, die im Hof Aufsicht fuehren"
9. "Eine Liste aller Schueler der Klasse 7b mit ihrem Aufenthaltsort am Donnerstag"
10. "Alle Lehrer, die irgendwann in der Woche im Museumsdorf sind (Entweder als Kursleiter oder als Begleitung)"

Anfragen mit Aggregatsfunktionen

11. "Anzahl aller Kurse"
12. "Anzahl aller Kurse bestimmen und diese in 'Anzahl aller Kurse' umbennennen"
13. "Klasse(n) mit mehr als 5 Schülern"
14. "Übersicht der Lehrerkürzel, inklusive Anzahl der Aufsichten (ohne Exkursionsbegleitungen)"
15. "Anzahl der Exkursionsbegleitungen der Lehrer"
16. "Anzahl der SuS in den jeweiligen Kursen, mit Angabe der Kursnamen"
17. "Anzahl der Schüler im 'Mittelalterkurs'"
18. "Übersicht der Kurse mit der Anzahl der verantwortlichen SuS"
19. "Durchschnittlicher Kursstart pro Tag"
20. "Anzahl der Sportlehrer"

(C) RALG (DES)

1. Relationenschema

siehe Datei *Prowo-schema.ra*

2. Anfragen und Ergebnisse [RALG;TRC;DRC]

Für die Relationale Algebra, siehe Datei *prowo_query_erg_1ralg.ra*,

für das TRC siehe Datei *prowo_query_erg_2trc.ra*

für das DRC siehe Datei *prowo_query_erg_3drc.ra*

für alle Anfrageergebnisse der ersten zehn Anfragen siehe Datei *prowo-query_erg.txt*

(D) SQL (PostgreSQL)

für die Create-Datei siehe *Create-prowo.sql*

für die Drop-Datei siehe *Drop-prowo.sql*

für die Insert-Datei siehe *Insert-prowo.sql*

für die Delete-Datei siehe *Delete-prowo.sql*

[die Insert-Dateien für jede Tabelle finden sich im Ordner: Inserts-Tabellen]

für alle Anfragen siehe jeweilige Datei *prowo-<n>.sql*

für alle gesammelten Anfragen in SQL siehe *prowo-query.sql*

für alle Anfragen inklusive Ergebnisse siehe *prowo-query_SQL_erg.txt*

(E) Didaktisch-Methodisches

Die Verwaltung einer schulinternen Projektwoche ist schülernah und ermöglicht es den Lernenden, sich dadurch leichter in die Thematik der Datenbank einzuarbeiten.

Der Umfang der Miniwelt mit fünf Entitäten und daraus resultierenden neun Relationen gewährleistet die Übersichtlichkeit. Auf ternäre Beziehungen wurde mit Blick auf die Komplexität bewusst verzichtet. Die Modellierung eines ER-Modells ist von den Schülerinnen und Schülern (mit Einhilfen) leistbar, die gewählten Kardinalitäten berücksichtigen alle Beziehungstypen (1:1, 1:n und m:n) und sind aufgrund der Nähe zur Lebenswelt der Schüler im Allgemeinen intuitiv erfassbar. Zur Spezifikation wurde auf die Min-Max-Notation zurückgegriffen, da Informationen umfassender als mit der Chen-Notation dargestellt werden können.

Die Beispielpopulationen der Tabellen sind realitätsnah, aber dennoch exemplarischen Charakters. Trotz des geringen Umfangs wurde mit Blick auf die Anfragen bewusst auf Variationen geachtet. Dies betrifft z.B. die Anzahl der Lernenden in den Klassen, die beteiligten Fächer oder die unterschiedlichen Orte oder Zeiten, an denen die Kurse stattfinden.

Die gewählten Anfragen besitzen eine Relevanz für die Realität, die die Datenbank abbilden soll. Es wurde darauf geachtet, bei jeder weiteren Anfrage eine Progression anzustreben. Zunächst wurde ohne Aggregatsfunktionen nur innerhalb einer Tabelle gearbeitet. Schrittweise wurden WHERE-Bedingungen und anschließend verschiedene JOINS eingeführt und der Schwierigkeitsgrad sukzessive erhöht. Anschließend wurde dieses Vorgehen mit Aggregatsfunktionen wiederholt. Aufgrund der geringen Anzahl numerischer Attribute wurde auf komplexe verschachtelte Anfragen mit Aggregatsfunktionen verzichtet.

Die realisierte Miniwelt kann in Teilen in der Sekundarstufe I (Wahlthemenfeld 3.7: Datenbanken) verwendet werden, um z.B. Attribute mit Name, Typ und Wert sowie Datensätze zu unterscheiden oder Datensätze zu verändern. Zudem können Planung und Realisierung der

Datenbank in Ansätzen behandelt werden. In der Sekundarstufe II (4.1. Datenbanken) können mit der Projektwochenverwaltung ergänzend u.a. das Konzept des relationalen Datenbankschemas und verschiedene Abfragen (Projektion, Selektion, JOINS) thematisiert werden. Es ist denkbar, dass die Schülerinnen und Schüler unterschiedliche Miniwelten als Datenbanken realisieren. Die Auseinandersetzung mit der Miniwelt Projektwoche im gesamten Kurs ist vom Betreuungsaufwand allerdings deutlich geringer. Individuelle Lösungswege und Alternativen bieten sich auch mit der Projektwochen-Miniwelt an. Unterschiedliche Kleingruppen können dabei verschiedene Ausschnitte der Miniwelt fokussieren (z.B. Möglichkeit von Exkursionen, verantwortliche Schüler für Kurse etc.). Die Modellierung und Implementierung der Datenbank sollte sich über mehrere Wochen erstrecken.

(F) Implementierung eines Views in Go/PHP (Auswahl)

1. Beschreibung und Motivation des Views

Das Formular01 ermöglicht es für einen beliebigen Schüler neben der Angabe seiner Klasse auch seinen belegten Kurs, den Kursleiter sowie ggf. dessen Verantwortlichkeit für den Kurs anzuzeigen. Dies kann zum Beispiel in der Realität zur Verwendung kommen, wenn ein "vergesslicher" Schüler seine Kursbelegung erfragen möchte.

Das Formular02 ermöglicht es für jeden gewählten Ort eine Übersicht der Lehrkräfte anzuzeigen, die wenigstens einmal an diesem Ort im Rahmen der Projektwoche anwesend waren. Dies kann zum Beispiel dafür genutzt werden, um bei etwaigen Problemen die verantwortliche Lehrkraft zu ermitteln.

Basismöglichkeit

Eine Möglichkeit ist die benutzerfreundliche und schnelle Möglichkeit von Anfragen bezüglich der Datenbank.

Erweiterte Möglichkeit

Eine erweiterte Möglichkeit ist die benutzerfreundliche Möglichkeit beliebige Daten zu manipulieren.