

Eine Vorhilfe-Datenbank für Schulen



Vorhilfe ist besser als Nachhilfe!

Thomas Wolf
tomflow@gmx.de

Christoph Chi
c.chi@fu-berlin.de

Marten Schlüter
schluetermarten@googlemail.com

Ben Boettcher
boettcher.ben@t-online.de

3. Juli 2017

Inhaltsverzeichnis

1 Datenbank Vorhilfe Dokumentation	4
1.1 Miniwelt einer Vorhilfebörse	4
2 Integritätsbedingungen bzgl. der Datenbank 'Vorhilfe'	5
2.1 Statische Integritätsbedingungen	5
2.2 Dynamische Integritätsbedingungen	6
3 nichttriviale Funktionale Abhängigkeiten (d)	6
3.1 Relation Schüler	6
3.2 Relation Lehrer	6
4 Relationenmodell (e)	6
5 Notation von Integritätsbedingungen, die sich im Modell wiederfinden	7
5.1 Domain- bzw. Attribut-Bedingungen	7
6 Transformationsregeln am Beispiel unserer Miniwelt 'Vorhilfe'	8
6.1 Transformation von Entitätstypen	8
6.2 Transformation von n:m-Beziehungstypen	8
6.3 Problem einer einseitig 0...* optionalen Beziehung	8
6.4 Transformation von c:c-Beziehungstypen	8
6.5 Transformation von 1:n-Beziehungstypen	8
6.6 Transformation von 1:1-Beziehungstypen	9

7	Didaktisch-Methodisches	9
7.1	Relevanz von Datenbanken für die Gesellschaft und das Leben	9
7.2	Thema des RLPs	9
7.3	ER-Modellierung / Miniwelt	10
7.4	Transformation in ein Relationenmodell u. Regeln mit methodischen Überlegungen zur Normalform, Redundanz und Anomalien	10
7.5	Detaillierte method. did. Überlegungen der Umsetzung von differenzierten Anfragen	10
8	Kommentar zu den Views	11
8.1	View lehrer	12
8.2	View cafeteria	12
8.3	View schüler	12
9	Anfragen in RA, DRC und TRC	12
9.1	Auflistung aller Lehrerinnen	12
9.1.1	RA	12
9.1.2	DRC	12
9.1.3	TRC	12
9.1.4	DES	12
9.2	Ausgabe der Daten des Lehrers 'Czetö'	12
9.2.1	RA	12
9.2.2	DRC	12
9.2.3	TRC	13
9.3	Liste der Räume im 1.Stock	13
9.3.1	RA	13
9.3.2	DRC	13
9.3.3	TRC	13
9.3.4	DES	13
9.4	Ausgabe der Tabelle 'Ausstattung'	13
9.4.1	RA	13
9.4.2	DRC	13
9.4.3	TRC	13
9.5	Welche Räume haben einen Beamer?	13
9.5.1	RA	13
9.5.2	DRC	13
9.5.3	TRC	14
10	Relationen des ER-Modells	14
10.1	Entitäten	14
10.1.1	Tabelle: ausstattung	14
10.1.2	Tabelle: fach	14
10.1.3	Tabelle: konto	15
10.1.4	Tabelle: kurs	15

10.1.5	Tabelle: lehrer	16
10.1.6	Tabelle: methode	16
10.1.7	Tabelle: raum	17
10.1.8	Tabelle: schueler	17
10.2	Beziehungen	19
10.2.1	Tabelle: hat;	19
10.2.2	Tabelle: findet_statt;	20
10.2.3	Tabelle: beaufsichtigt;	20
10.2.4	Tabelle: eingeschrieben;	21
10.2.5	Tabelle: vermittelt;	22
10.2.6	Tabelle: beinhaltet;	22
10.3	Anfragen in aufsteigendem Schwierigkeitsgrad	22
10.3.1	Tabelle aus Vorhilfe-Anfrage 01: SELECT nname FROM lehrer; .	22
10.3.2	Tabelle aus Vorhilfe-Anfrage 02: SELECT * FROM lehrer WHERE nname = 'Czetö';	23
10.3.3	Tabelle aus Vorhilfe-Anfrage 03 : SELECT * FROM ausstattung; .	23
10.3.4	Tabelle aus Vorhilfe-Anfrage 04: SELECT * FROM schueler WHERE vname = 'Gabi';	24
10.3.5	Tabelle aus Vorhilfe-Anfrage 05: SELECT RNR FROM raum WHERE stock = 1;	24
10.3.6	Tabelle aus Vorhilfe-Anfrage 06 : SELECT vname,nname FROM schueler NATURAL JOIN konto WHERE kontostand = 0;	25
10.3.7	Tabelle aus Vorhilfe-Anfrage 07: SELECT vname,nname,kontostand FROM Schueler NATURAL JOIN konto WHERE kontostand > 0; .	25
10.3.8	Tabelle aus Vorhilfe-Anfrage 08: SELECT vname,nname,konr FROM schueler NATURAL JOIN konto WHERE vname = 'Frieder' AND nname = 'Bär';	26
10.3.9	Tabelle aus Vorhilfe-Anfrage 09a: SELECT nname, vname FROM schueler WHERE email LIKE '%luise.net';	26
10.3.10	Tabelle aus Vorhilfe-Anfrage-09b: SELECT nname,kuerzel FROM lehrer WHERE email LIKE '%vim.org';	27
10.3.11	Tabelle aus Vorhilfe-Anfrage 9c: SELECT nname,kuerzel,email FROM lehrer WHERE email LIKE '%todesstern.de';	27
10.3.12	Tabelle aus Vorhilfe-Anfrage 10 : SELECT vname,nname,kontostand FROM schueler NATURAL JOIN konto WHERE kontostand > 0 ORDER BY kontostand DESC;	27
10.3.13	Tabelle aus Vorhilfe-Anfrage 11 : SELECT s.vname,s.nname,s.lid,s.email FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,schueler AS s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';	28
10.3.14	Tabelle aus Vorhilfe-Anfrage 12: SELECT nname,vname,geb,klasse FROM schueler NATURAL JOIN eingeschrieben WHERE age(geb) < '16 year' AND KNR = 2;	29

10.3.15	Tabelle aus Vorhilfe-Anfrage 13: SELECT COUNT(sid) AS Anzahl_eingeschrieben_in_Kurs_2 FROM eingeschrieben WHERE knr = 2;	29
10.3.16	Tabelle aus Vorhilfe-Anfrage 14: SELECT kuerzel, wtag, zeit FROM (kurs NATURAL JOIN beaufsichtigt NATURAL JOIN lehrer) ORDER BY kuerzel;	30
10.3.17	Tabelle aus Vorhilfe-Anfrage 15: SELECT * FROM raum WHERE anzP >= 30;	30
10.3.18	Tabelle aus Vorhilfe-Anfrage 16: SELECT COUNT(kurs.KNR) AS Kurse_am_Donnerstag FROM Kurs WHERE wtag= 'do';	31
10.3.19	Tabelle aus Vorhilfe-Anfrage 17: SELECT COUNT(*) AS Anzahl_schueler_Mathematik FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f, schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Mathematik';	31
10.3.20	Tabelle aus Vorhilfe-Anfrage 18: SELECT sk.knr, sk.vname,sk.nname AS Kursleitung_Mathematik ,sk.wtag FROM(schueler NATURAL JOIN kurs) AS sk,(vermittelt NATURAL JOIN fach) AS vf WHERE fname = 'Mathematik' AND sk.knr = vf.knr;	31
10.3.21	Tabelle aus Vorhilfe-Anfrage 19: SELECT COUNT(*) AS Raeume_mit_Beamer FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung GROUP BY geraete HAVING geraete ='Beamer';	32
10.3.22	Tabelle aus Vorhilfe-Anfrage 20a: SELECT rnr FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung WHERE geraete ='Beamer';	32
10.3.23	Tabelle aus Vorhilfe-Anfrage 20b: SELECT RNR FROM raum NATURAL JOIN hat WHERE ANR = 2;	32
10.3.24	Tabelle aus Vorhilfe-Anfrage 21a: SELECT vname,nname,email,fname FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';	33
10.3.25	Tabelle aus Vorhilfe-Anfrage 21b: SELECT MAX(f.maxP) - COUNT(fname) AS Freie_Plaetze_Informatik FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik'	33

1 Datenbank Vorhilfe Dokumentation

1.1 Miniwelt einer Vorhilfebörse

Vorhilfe ist besser als Nachhilfe! In einer Vorhilfeminiwelt gibt es SchülerInnen, die Vorhilfestunden suchen und SchülerInnen, die Vorhilfestunden anbieten. Ein(e) Lehrer(in) beaufsichtigt bis zu vier Vorhilfegruppen von SchülernInnen in einem Raum der Schule.

Die Räume haben unterschiedliche Ausstattungen. Die Vorhilfekurse finden im Mittagsblock der Schule an allen Wochentagen statt, ausnahmsweise gibt es auch eine Blocksession am Samstag. Für ihre Stunden bekommen die SchülerInnen für das Unterrichten 4 Punkte, für das Nehmen von Vorhilfe 1 Punkt, auf ihrem Konto gutgeschrieben. Es werden Kurse mit Fachbezug angeboten, es besteht weiterhin die Möglichkeit spezielle Prüfungsvorbereitungskurse (Abitur, MSA) zu belegen. Eine Kombination von beiden Möglichkeiten ist gestattet. SchülerInnen sollten nicht mehr als 4 SchülerInnen unterrichten.

Zusätzlich mögliche Integritätsbedingungen: SchülerInnen sollten maximal 2 Angebote in der Woche anbieten können. Ab einem SchülerInnen sollte der Kurs erst stattfinden. Maximal sollten vier Kurse gleichzeitig in einem Raum stattfinden können.

2 Integritätsbedingungen bzgl. der Datenbank 'Vorhilfe'

(c) Verbale Angabe von semantischen Bezügen, die sich im Modell nicht wiederfinden, durch Prädikate (Constraints, Integritätsbedingungen). Dokument: knapp formulierte Textprädikate in der Sprache des Modells (<db>-Semantik.txt).

2.1 Statische Integritätsbedingungen

- Tupelbedingungen: nicht vorhanden.
- Relationen-Bedingungen:
- Aggregatbedingungen: nicht vorhanden
- Rekursive Bedingungen: nicht vorhanden
- Wird z.B. ein Raum mit Beamer gesucht und wurde dieser Raum von einer anderen Gruppe schon gebucht, die auch einen Beamer braucht, kommt es zu einem Konflikt, weil mehr als eine Gruppe diesen Raum buchen könnte.
- Ein weiteres Problem stellt die mögliche zeitgleiche Teilnahme der SchülerInnen an mehreren Kursen dar, bzw. könnten die SchülerInnen auch zeitgleich zwei Kurse leiten. Ebenso wäre es möglich einen Kurs zu leiten und sich in diesen einzuschreiben.
- Zwei LehrerInnen könnten denselben Kurs beaufsichtigen.
- Max. 4 Kurse können zur selben Zeit im selben Raum stattfinden.
- Blockkurse werden nur über den Wochentag (Sa) abgefangen.
- Wenn keine SchülerInnen in einen Kurs eingeschrieben wurde, findet der Kurs nicht statt. - Wird nicht abgefangen, kann aber durch eine entsprechende Anfrage gelöst werden.

- Bewusste Inkaufnahme des Semantikverlustes bei der einseitig optionalen 0..* Beziehung bei den Relationen vermittelt und beinhaltet, damit bei Kurs Methoden oder Fächer kombiniert werden können. Damit ist es auch möglich, dass ein Kurs weder Methode noch Fach vermittelt bzw. beinhaltet.
- geprüfte Integrität: s. Datei 'RelMod.txt'

2.2 Dynamische Integritätsbedingungen

TRIGGER z.B. wenn alle SchülerInnen ein anderes Bewertungssystem bekommen sollten, müssten mittels Trigger die Konten ein Update bekommen.

3 nichttriviale Funktionale Abhängigkeiten (d)

3.1 Relation Schüler

$$F = \{nname, vname, geb\} \rightarrow Klasse, LID, email, KoNr$$

3.2 Relation Lehrer

$$F = email \rightarrow kuerzel$$

$$F = kuerzel \rightarrow email$$

4 Relationenmodell (e)

(e) Ein Relationenmodell dazu in dritter Normalform mit Angabe von Schlüsseln, Fremdschlüsseln und Notation von Integritätsbedingungen als logische Ausdrücke in einer Datei <db>-RelMod.txt.

Transformationsergebnis nach dem vorliegenden ER-Modell „Vorhilfe“: Als Ergebnis erhält man die folgenden 13 Relationen im Relationenmodell:

Bemerkung zur 3. NF: Außer der Relation Lehrer befinden sich alle Relationen in der 3.NF. Wir entschlossen uns diese so zu belassen, da in der Datenbank die Attribute der Relation „Lehrer“ immer nur zusammenhängend vorkommen.

1. Konto (KoNR, Kontostand)
2. Lehrer (LID, nname, email, kuerzel)
3. Schueler (SID, nname, vname, geb, klasse, LID (Beziehung 'ist Klassenlehrer' geht hier auf), email, KoNR (Ref. Beziehung 'besitzt'))
4. Raum (RNR, stock, anzP)
5. Ausstattung (ANR, geraete)
6. Kurs (KNR, SID (Beziehung 'bietet_an' geht hier auf) wtag, zeit, maxP)

7. Fach (FNR, fname)
8. Methode (MNR, mname)
9. hat (RNR (Ref. Raum (RNR), ANR Ref. Ausstattung (ANR))
10. findet_statt (KNR (Ref. Kurs (KNR), RNR (Ref. Raum (RNR))
11. eingeschrieben (SID (Ref. Schueler (SID)), KNR (Ref. Kurs (KNR)), zeitS)
12. vermittelt (KNR (Ref. Kurs (KNR), FNR Ref. Fach (FNR))
13. beinhaltet (KNR (Ref. Kurs (KNR), MNR (Ref. Methode (MNR))
14. beaufsichtigt (KNR Ref. Kurs (KNR), LID (Ref. Lehrer (LID))

5 Notation von Integritätsbedingungen, die sich im Modell wiederfinden

5.1 Domain- bzw. Attribut-Bedingungen

Diese Bedingungen sichern Beschränkungen des Wertebereiches einzelner Attribute:
Beispiel:

```
CREATE TABLE kurs (
    KNR SERIAL, --KursNr
    --> hier geht die Beziehung schueler -<bietet an>- kurs in kurs auf
    Sid INTEGER REFERENCES schueler(sid),
    wtag wotag,
    zeit TIME,          -- Kurszeit
    maxP INTEGER -- Maximale Teilnehmerzahl (Integritätsbedingung)
CHECK (maxP BETWEEN 2 AND 5) DEFAULT 4,
PRIMARY KEY (KNR)
);
```

Relationen-Bedingungen Schlüsselbedingungen wie z.B. in der folgenden Form, in der die Bedingung an das Attribut angehängt wird. Finden sich z.B. in folgender Form wieder:

```
CREATE TABLE fach (
    FNR SERIAL,
    fname VARCHAR (50),
PRIMARY KEY (FNR)
);
```

Referentielle Bedingungen und ihre Problem in unserer Modellierung: Referentielle Bedingungen stellen semantische Bezüge zwischen Paaren von Relationen sicher. Eine Form ist z. B. die Fremdschlüsselbeziehung als Teilmengenprädikat.

```
--> hier geht die Beziehung schueler -<bietet an>- kurs in kurs auf  
Sid INTEGER REFERENCES schueler(sid),
```

Problematisch, wenn z.B. mehr als ein(e) Schüler(in) einen Kurs leiten möchte.

6 Transformationsregeln am Beispiel unserer Miniwelt 'Vorhilfe'

6.1 Transformation von Entitätstypen

Ein Entitätstyp des ER-Modells wird in einen Relationstyp des Relationenmodells mit gleicher Attributmenge transformiert.

Beispiel:

Raum (*RNR*, *stock*, *anzP*)

6.2 Transformation von n:m-Beziehungstypen

Der ursprüngliche Beziehungstyp im Relationenmodell wird zur eigenständigen Relation und enthält sowohl die Schlüssel der beteiligten Entitäten (als Fremdschlüssel) als auch -falls vorhanden- die Attribute des alten Beziehungstyps selbst. Als Schlüssel wird die Kombination der beteiligten Fremdschlüssel verwendet.

Beispiel:

hat (*RNR*(*Ref.Raum*(*RNR*), *ANRRef.Ausstattung*(*ANR*))

6.3 Problem einer einseitig 0...* optionalen Beziehung

Sie ist nicht unterscheidbar von 1...* (Semantikverlust)

Beispiel:

vermittelt (*KNR*, *FNR*)

6.4 Transformation von c:c-Beziehungstypen

(0...1) : (0...1) In unserem Modell nicht vorhanden.

6.5 Transformation von 1:n-Beziehungstypen

"Bei der Transformation von 1:n-Beziehungstypen werden die ER-Relationsattribute zusammen mit dem Fremdschlüssel der Entität der 1-Seite der transformierten Entität der n-Seite hinzugefügt. Als Schlüssel dient der jeweilige Primärschlüssel des Entitätstyps." (Siehe Schäferskript 07 S. 51)

Semantikverlust ist mgl., da die Kardinalitäten der beteiligten Entitäten verloren gehen. Aus diesem Grund haben wir teilweise Integritätsbedingungen hinzugefügt.

Beispiel (s.o.):


```

CREATE TABLE kurs (
    KNR SERIAL, --KursNr
    --> hier geht die Beziehung schueler -<bietet an>- kurs in kurs auf
    Sid INTEGER REFERENCES schueler(sid),
    wtag wotag,
    zeit TIME,          -- Kurszeit
    maxP INTEGER -- Maximale Teilnehmerzahl (Integritätsbedingung)
CHECK (maxP BETWEEN 2 AND 5) DEFAULT 4,
PRIMARY KEY (KNR)
);

```

6.6 Transformation von 1:1-Beziehungstypen

Der Primärschlüssel einer der beiden Tabellen wird als Fremdschlüssel der anderen Relationen in eine zusätzliche Spalte aufgenommen.

Beispiel: Relation 'besitzt'

```

CREATE TABLE schueler (
    SID SERIAL, -- automatischer Zähler
    nname VARCHAR (100),
    vname VARCHAR (100),
    geb DATE,
    klasse VARCHAR (3), --Klassenstufe
    LID INTEGER REFERENCES lehrer (LID), --> Hier geht die Beziehung
    email VARCHAR (100), -- 'ist Klassenlehrer' auf
    KoNR INTEGER REFERENCES konto (KoNR), --> hier geht 'besitzt' auf
    PRIMARY KEY (SID)
);

```

Schueler (SID, nname, vname, geb, klasse, LID (Ref. Beziehung 'ist Klassenlehrer'), email, KoNR (Ref. Beziehung 'besitzt'))

7 Didaktisch-Methodisches

7.1 Relevanz von Datenbanken für die Gesellschaft und das Leben

Im Alltag trifft man mehr oder weniger bewusst auf Datenbanken. Hierfür soll ein Bewusstsein geschaffen werden. Methodische Überlegung: Unterschied zwischen Tabellen und DB vermitteln.

7.2 Thema des RLPs

„Am Beispiel der Entwicklung eines Datenbanksystems führen die Schülerinnen und Schüler alle Phasen des Problemlöseprozesses von der Analyse der Ausgangssituation

zur Erfassung der Daten über die Modellierung einer Datenbank bis hin zu ihrer praktischen Umsetzung in einem Datenbankmanagementsystem selbstständig durch.“ Rahmenlehrplan Sek.II Seite 19.

7.3 ER-Modellierung / Miniwelt

Umsetzungsmöglichkeiten: Ausgehend von der Beschreibung der Miniwelt wird ein gemeinsames ER-Modell entwickelt, wobei die Kardinalitäten hierbei sinnvoll eingearbeitet/thematisiert werden. Bei kürzerem Zeitumfang besteht die Möglichkeit das vorgegebene ER-Modell nachzuvollziehen und im Anschluss in ein Relationenmodell umzuwandeln.

7.4 Transformation in ein Relationenmodell u. Regeln mit methodischen Überlegungen zur Normalform, Redundanz und Anomalien

Nach der Umwandlung der Entitäten in ein Relationenmodell werden die Beziehungen umgewandelt. Hierbei sollten insbesondere die Kardinalitäten und deren „Auflösung“ in Relationen thematisiert werden. Anomalien und Redundanzen könnten aufgegriffen werden, um den Vorteil von Normalformen herauszuarbeiten. Semantikverlust kann inhaltlich erörtert werden (siehe Semantik.txt).

7.5 Detaillierte method. did. Überlegungen der Umsetzung von differenzierten Anfragen

Anfangs sollte eine Datenbank, befüllt mit Populationen, vorgegeben werden. Erst wenn das Erstellen von Anfragen in allen drei Schwierigkeitsstufen beherrscht wird, sollten die Schüler und Schülerinnen selbst Datenbanken erstellen dürfen.

Um im Unterricht eine ausreichende Differenzierung gewährleisten zu können, haben wir Anfragen mit unterschiedlichen Schwierigkeitsstufen entwickelt. Den Grundlagen der Progression folgend steigt der Schwierigkeitsgrad mit der Anzahl der Sterne, die unsere vier Schwierigkeitsstufen repräsentieren. In der Einführungsphase beschränken sich die Ein-Sterneanfragen hauptsächlich auf die Operatoren SELECT, FROM und WHERE. Anfragen mit eineinhalb Sternen enthalten zusätzlich NATURAL JOIN Verbindungen. Diese Anfragen eignen sich für sehr leistungsstarke Schüler und Schülerinnen, denen die einfachen Anfragen in der Einführungsphase zu leicht erscheinen. Denkbar ist ebenfalls, die Schüler und Schülerinnen selbst einfache Anfragen entwickeln zu lassen, die nur mit diesen genannten Operatoren auskommen dürfen.

Nach der Einführungsphase folgen die komplexeren Abfragen. Die Zwei-Sterneanfragen, die den Großteil unserer Anfragen bilden, sollten mindestens einen zusätzlichen Befehl wie beispielsweise: ORDER BY, AS, LIKE, ASC, WHERE NOT oder NOT enthalten; jedoch üblicherweise immer nur EINEN neuen Befehl zusätzlich. Ausnahmsweise kann auch schon mal ein COUNT Befehl verwendet werden. Leistungsstarke Schüler und Schülerinnen können sich die Befehle über die Hilfefunktion bzw. im Internet mit Hilfe von Syntaxbeispielen selbst erarbeiten. In der Regel aber, sollten die Bedeutung und Verwendung aller unbekannten Operatoren vom Lehrer oder Lehrerin vorgeführt werden.

Dagegen zeichnen sich die Drei-Sterneanfragen im Allgemeinen durch die Verwendung mehrerer JOINS aus bzw. beziehen in einzelnen Fällen Aggregatfunktionen (u.a. MAX, COUNT) mit ein. Auch Unteraanfragen sollten möglich sein. Diese Anfragen sind schon sehr komplex und möglicherweise nicht von jedem Schüler bzw. jeder Schülerin selbstständig zu lösen. Jedoch sollte jeder mind. eine Drei-Sterneanfrage ohne zusätzliche Hilfe bauen können.

Die Vier-Sterneanfrage ist nur sehr leistungsstarken Schülern bzw. Schülerinnen vorbehalten.

Insgesamt liegt der Schwerpunkt auf den Zwei-Sterneanfragen, daher haben wir für diese Schwierigkeitsstufe die meisten Anfragen erstellt. An dieser Stelle werden auch die meisten neuen Befehle eingeführt.

Nicht alle Anfragen sind für ein didaktisch sinnvolles Vorgehen im Unterricht notwendig. Möglicherweise erscheinen einigen Schülern und Schülerinnen die Ein-Sterneanfragen zu einfach und von den fünf angebotenen kann eine Auswahl bearbeitet werden. Ebenfalls sollte die Vier-Sterneanfrage fakultativ sein, da die Komplexität entsprechend hoch ist. Aufgrund der Variabilität und Vielfältigkeit sollten von den verbleibenden Schwierigkeitsstufen (sowohl Zwei- als auch Drei-Sterneanfragen) keine Anfragen weglassen.

Eine mögliche Erweiterung wäre die selbstständige Erstellung einer Datenbank nach einem vorgegebenen ER-Modell. Entsprechend der unterschiedlichen Lerngruppen bzw. Stundenzahl (Wahlpflichtkurs, Oberstufenkurse u.ä.) kann die Arbeit mit der Datenbank aufgrund der aufbauenden Schwierigkeitsgrade zeitlich erweitert bzw. ergänzt werden. Sollte nur ein kurzer Einblick in die Arbeit mit einer Datenbank erfolgen, genügen die fünf Anfragen mit dem geringsten Schwierigkeitsgrad (Ein-Sterneanfragen). Ein solcher Zugang ist auch für eine Einführung in SQL ab der 10. Klasse geeignet.

Spielraum für eine unterschiedliche Methodik bietet sich dann, wenn die Schüler und Schülerinnen selbst Anfragen für ihren Banknachbarn entwickeln sollen. Dazu bieten sich Gruppenarbeiten an, die arbeitsteilig (Anfragen, Erstellung einer DB, skizzieren eines ER Modells) vorgehen. Frontaler wäre die Vorgabe der dargestellten Anfragen durch den Lehrer, die jeder Schüler bzw. jede Schülerin sukzessive lösen soll. Die Lösungen könnten anschließend diskutiert bzw. verglichen werden. Langfristige Projekte könnten die bestehende Datenbank erweitern, um sie auf die speziellen Gegebenheiten der jeweiligen Schule anzupassen und auch dort einsatzbereit zu machen.

Vorhilfe ist besser als Nachhilfe

8 Kommentar zu den Views

Die Go-Datei views.go implementiert einen Webserver auf dem Port (localhost:) 8080, der über einen beliebigen Browser aufgerufen werden kann. Nach Aufruf des Wurzelverzeichnisses wird eine Auswahlmaske gezeigt, über die die einzelnen Optionen aufgerufen werden kann. Sie enthält auch ein Formular, in das beliebige Anfragen eingegeben werden können, die dann als html-formatierte Tabelle ausgegeben werden. Die Anfragen dürfen nicht mehr als 12 Attribute ausgeben, sonst werden sie nicht dargestellt.

Danach wird das SVG-Bild des ER-Modells gezeigt, das Links als Anfragen enthält.

8.1 View lehrer

Diese View ist für den aufsichtführenden Lehrer bestimmt. Der Lehrer kann über diese View den Kontostand eines Schülers nach Angabe der Kontonummer erhöhen.

8.2 View cafeteria

In der Cafeteria haben die Bediensteten Zugriff auf diese View. Nach Eingabe der Kontonummer wird dem Schüler ein Punkt vom Konto abgezogen. Er kann sich dafür einen Tee kaufen.

8.3 View schüler

Mit dieser View kann ein Schüler nach einem Raum suchen, der bestimmte Ausstattungsmerkmale besitzt.

9 Anfragen in RA, DRC und TRC

9.1 Auflistung aller Lehreramen

9.1.1 RA

$$\pi_{nname}(schueler)$$

9.1.2 DRC

$$\{nname | (\exists SID)(\exists vname)(\exists geb)(\exists klasse)(\exists LID)(schueler(SID, vname, geb, klasse, LID))\}$$

9.1.3 TRC

$$\{t | \exists s(schueler(s)) \wedge t[nname] = s[nname]\}$$

9.1.4 DES

$$project\ nname(lehrer)$$

9.2 Ausgabe der Daten des Lehrers 'Czetö'

9.2.1 RA

$$\sigma_{nname='Czetö'}(lehrer)$$

9.2.2 DRC

$$\{LID, nname, email, kuerzel | (lehrer(LID, nname, email, kuerzel, nname = 'Czetö'))\}$$

9.2.3 TRC

$$\{t | (\text{lehrer}(t)) \wedge t[nname] = 'Czetö'\}$$

9.3 Liste der Räume im 1.Stock

9.3.1 RA

$$\pi_{RNR}(\sigma_{stock=1}(raum))$$

9.3.2 DRC

$$\{RNR | (\exists stock)(\exists anzP)(raum(RNR, stock, anzP) \wedge stock = 1)\}$$

9.3.3 TRC

$$\{t | \exists r(raum(r) \wedge t[RNR] = r[RNR] \wedge r[stock] = 1)\}$$

9.3.4 DES

$$project\ RNR(select\ stock = 1(raum))$$

9.4 Ausgabe der Tabelle 'Ausstattung'

9.4.1 RA

$$ausstattung$$

9.4.2 DRC

$$\{ANR, geraete | (ausstattung(ANR, geraete))\}$$

9.4.3 TRC

$$\{t | ausstattung(t)\}$$

9.5 Welche Räume haben einen Beamer?

9.5.1 RA

$$\pi_{RNR}(\sigma_{geraete='Beamer'}(raum \bowtie (hat \bowtie ausstattung)))$$

9.5.2 DRC

$$\begin{aligned} &\{RNR | (\exists stock)(\exists anzP)(\exists ANR)(\exists geraete) \\ & (raum(RNR, stock, anzP) \wedge hat(RNR, ANR) \wedge \\ & ausstattung(ANR, geraete) \wedge geraete = 'Beamer')\} \end{aligned}$$

9.5.3 TRC

$$\{t | (\exists r)(\exists h)(\exists a)(raum(r) \wedge hat(h) \wedge ausstattung(a)$$

$$\wedge t[RNR] = r[RNR] \wedge a[geraete] = 'Beamer')\}$$

10 Relationen des ER-Modells

10.1 Entitäten

10.1.1 Tabelle: ausstattung

ausstattung	
anr	geraete
1	Tafel
2	Beamer
3	Computer
4	OH-Projektor
5	Fernseher
6	Whiteboard
7	Smartboard
8	Waschbecken
9	Sitzecke

10.1.2 Tabelle: fach

fach	
fnr	fname
1	keine
2	Informatik
3	Mathematik
4	Englisch
5	Französisch
6	Deutsch

10.1.3 Tabelle: konto

konto	
konr	kontostand
1000	0
1001	10
1002	3
1003	1
1004	3
1005	4
1006	1
1007	0
1008	2
1009	3
1010	8
1011	0
1012	10
1013	2
1014	8
1015	4

10.1.4 Tabelle: kurs

kurs				
knr	sid	wtag	zeit	maxp
1	1	mo	0000-01-01T12:30:00Z	4
2	1	di	0000-01-01T12:15:00Z	4
3	2	mo	0000-01-01T12:30:00Z	4
4	2	mi	0000-01-01T12:30:00Z	4
5	3	do	0000-01-01T12:00:00Z	4
6	3	di	0000-01-01T13:00:00Z	4
7	4	mi	0000-01-01T12:15:00Z	4
8	4	fr	0000-01-01T13:30:00Z	4
9	4	di	0000-01-01T14:00:00Z	4
10	7	di	0000-01-01T14:00:00Z	4
11	8	mi	0000-01-01T14:30:00Z	4

10.1.5 Tabelle: lehrer

lehrer			
lid	nname	email	kuerzel
1	Schaefer	os@vim.org	sf
2	Czetö	cz@aiko.net	cz
3	Bond	james@mi5.org	bd
4	Cooper	high@noon.us	cp
5	Klemt	winnie2@ado.de	kl
6	Schmitt	schmitt@gmx.de	sc
7	Schmidt	darthvader@todesstern.de	si
8	Boettcher	benniboe@zedat.de	boe
9	Wolf	tflow@gmx.de	wf
10	Chi	themroc@musichi.net	chi
11	Schlüter	ms@ado.de	ms

10.1.6 Tabelle: methode

methode	
mnr	mname
1	keine
2	ABI
3	MSA
4	Präsentation

10.1.7 Tabelle: raum

raum		
rnrr	stock	anzp
400	4	8
401	4	13
402	4	19
403	4	21
300	3	20
301	3	32
200	2	20
201	2	32
202	2	30
203	2	13
100	1	16
101	1	25
102	1	28
103	1	10
1	0	23
-101	-1	25

10.1.8 Tabelle: schueler

schueler							
sid	nname	vname	geb	klasse	lid	email	konr
1	Bär	Frieder	2002-11-11T00:00:00Z	11	1	go@luise.net	1000
2	Schmidt	Harald	2002-10-10T00:00:00Z	11	1	aha@luise.net	1001
3	Wolf	Hermine	2002-10-01T00:00:00Z	11	1	wo@luise.net	1002
4	Waal	Fritz	2002-09-11T00:00:00Z	11	1	wale@luise.net	1003
5	Gammel	Gabi	2000-08-01T00:00:00Z	12	2	gaga@luise.net	1004
6	Schlau	Gabi	2003-07-01T00:00:00Z	10	3	schlau@luise.net	1005
7	Gans	Gunter	2003-07-01T00:00:00Z	10	3	gg@luise.net	1006
8	Kannicht	Thomas	2004-03-03T00:00:00Z	09	4	konzert@luise.net	1007
9	Kurz	Susanne	2004-03-28T00:00:00Z	09	4	shorty@luise.net	1008
10	Braumeister	Thomas	2004-04-03T00:00:00Z	09	4	lame@luise.net	1009
11	Sabbel	Klara	2005-12-04T00:00:00Z	08	5	schlau@luise.net	1010
12	Kunze	Max	2005-12-05T00:00:00Z	08	6	maumau@luise.net	1011
13	Kolbe	Theresa	2005-06-07T00:00:00Z	08	6	prince@luise.net	1012
14	Kunze	Constanze	2005-12-05T00:00:00Z	08	6	babo@luise.net	1013
15	Witzel	Merve	2000-04-09T00:00:00Z	12	6	knaller@luise.net	1014
16	Sonnenschein	Clara	2000-12-30T00:00:00Z	12	7	mist@luise.net	1015

10.2 Beziehungen

10.2.1 Tabelle: hat;

hat;	
nr	anr
400	1
400	3
400	4
401	1
401	8
402	7
402	3
402	4
402	6
403	1
300	1
300	2
300	3
301	1
301	4
200	3
200	9
200	8
201	1
202	1
203	1
100	5
100	4
100	9
101	5
101	4
101	9
102	7
102	3
103	7
103	3
1	1
1	9
-101	1
-101	2
-101	4
-101	8
-101	9

10.2.2 Tabelle: findet_statt;

findet_statt;	
knr	rnr
1	300
2	300
3	301
4	301
5	201
6	301
7	301
8	200
9	403
10	1
11	-101

10.2.3 Tabelle: beaufsichtigt;

beaufsichtigt;	
knr	lid
1	1
2	8
3	3
4	9
5	10
6	11
7	5
8	6
9	6
10	2
11	7

10.2.4 Tabelle: eingeschrieben;

eingeschrieben;		
sid	knr	zeits
4	1	0000-01-01T14:57:59.421899Z
5	1	0000-01-01T14:57:59.421899Z
6	1	0000-01-01T14:57:59.421899Z
9	5	0000-01-01T14:57:59.421899Z
8	5	0000-01-01T14:57:59.421899Z
5	6	0000-01-01T14:57:59.421899Z
6	6	0000-01-01T14:57:59.421899Z
5	2	0000-01-01T14:57:59.468874Z
7	2	0000-01-01T14:57:59.468874Z
8	2	0000-01-01T14:57:59.468874Z
9	2	0000-01-01T14:57:59.468874Z
9	3	0000-01-01T14:57:59.468874Z
4	3	0000-01-01T14:57:59.468874Z
5	4	0000-01-01T14:57:59.468874Z
4	4	0000-01-01T14:57:59.468874Z
8	7	0000-01-01T14:57:59.468874Z
10	7	0000-01-01T14:57:59.468874Z
12	7	0000-01-01T14:57:59.468874Z
8	8	0000-01-01T14:57:59.468874Z
11	8	0000-01-01T14:57:59.468874Z
13	8	0000-01-01T14:57:59.468874Z
8	9	0000-01-01T14:57:59.468874Z
6	9	0000-01-01T14:57:59.468874Z
5	9	0000-01-01T14:57:59.468874Z
8	10	0000-01-01T14:57:59.468874Z
5	10	0000-01-01T14:57:59.468874Z
15	10	0000-01-01T14:57:59.468874Z
8	11	0000-01-01T14:57:59.468874Z
5	11	0000-01-01T14:57:59.468874Z

10.2.5 Tabelle: vermittelt;

vermittelt;	
knr	fmr
1	2
2	3
3	4
4	6
5	5
6	3
7	6
8	4
9	1
10	3
11	5

10.2.6 Tabelle: beinhaltet;

beinhaltet;	
knr	mnr
1	1
2	2
3	2
4	2
5	3
6	3
7	3
8	3
9	3
9	4
10	1
11	1

10.3 Anfragen in aufsteigendem Schwierigkeitsgrad

10.3.1 Tabelle aus Vorhilfe-Anfrage 01: SELECT nname FROM lehrer;

```
-- 01. Auflistung aller Lehrer-Nachnamen. *  
SELECT nname  
FROM lehrer;
```

SELECT nname FROM lehrer;
nname
Schaefer
Czetö
Bond
Cooper
Klemt
Schmitt
Schmidt
Boettcher
Wolf
Chi
Schlüter

10.3.2 Tabelle aus Vorhilfe-Anfrage 02: SELECT * FROM lehrer WHERE nname = 'Czetö';

```
-- 02. Ausgabe der Daten des Lehrers "Czetö". *
SELECT *
FROM lehrer
WHERE nname = 'Czetö';
```

SELECT * FROM lehrer WHERE nname = 'Czetö';			
lid	nname	email	kuerzel
2	Czetö	cz@aiko.net	cz

10.3.3 Tabelle aus Vorhilfe-Anfrage 03 : SELECT * FROM ausstattung;

```
-- 03. Ausgabe der Tabelle Ausstattung. *
SELECT *
FROM ausstattung;
```

SELECT * FROM ausstattung;	
anr	geraete
1	Tafel
2	Beamer
3	Computer
4	OH-Projektor
5	Fernseher
6	Whiteboard
7	Smartboard
8	Waschbecken
9	Sitzecke

10.3.4 Tabelle aus Vorhilfe-Anfrage 04: SELECT * FROM schueler WHERE vname = 'Gabi';

```
-- 04. Auflistung aller Schüler mit Vornamen 'Gabi'. *
SELECT *
FROM schueler
WHERE vname = 'Gabi';
```

SELECT * FROM schueler WHERE vname = 'Gabi';							
sid	nname	vname	geb	klasse	lid	email	konr
5	Gammel	Gabi	2000-08-01T00:00:00Z	12	2	gaga@luisse.net	1004
6	Schlau	Gabi	2003-07-01T00:00:00Z	10	3	schlau@luisse.net	1005

10.3.5 Tabelle aus Vorhilfe-Anfrage 05: SELECT RNR FROM raum WHERE stock = 1;

```
-- 05. Liste der Räume im "1. Stock". *
SELECT RNR
FROM raum
WHERE stock = 1;
```

SELECT RNR FROM raum WHERE stock = 1;
rnr
100
101
102
103

10.3.6 Tabelle aus Vorhilfe-Anfrage 06 : SELECT vname,nname FROM schueler NATURAL JOIN konto WHERE kontostand = 0;

```
-- 06. Welche Schüler haben kein Guthaben? (Kontostand) * 1/2
SELECT vname,nname
FROM schueler NATURAL JOIN konto
WHERE kontostand = 0;
```

```
SELECT vname,nname FROM schueler NATURAL JOIN konto WHERE kontostand = 0;
```

vname	nname
Frieder	Bär
Thomas	Kannicht
Max	Kunze

10.3.7 Tabelle aus Vorhilfe-Anfrage 07: SELECT vname,nname,kontostand FROM Schueler NATURAL JOIN konto WHERE kontostand > 0;

```
-- 07. Welche Schüler haben Guthaben * 1/2
SELECT vname,nname,kontostand
FROM schueler NATURAL JOIN konto
WHERE kontostand > 0;
```

```
SELECT vname,nname,kontostand FROM Schueler
NATURAL JOIN konto WHERE kontostand > 0;
```

vname	nname	kontostand
Harald	Schmidt	10
Hermine	Wolf	3
Fritz	Waal	1
Gabi	Gammel	3
Gabi	Schlau	4
Gunter	Gans	1
Susanne	Kurz	2
Thomas	Braumeister	3
Klara	Sabbel	8
Theresa	Kolbe	10
Constanze	Kunze	2
Merve	Witzel	8
Clara	Sonnenschein	4

10.3.8 Tabelle aus Vorhilfe-Anfrage 08: SELECT vname,nname,konr FROM schueler NATURAL JOIN konto WHERE vname = 'Frieder' AND nname = 'Bär';

```
-- 08. Wie ist die Kontonummer der Schülerin Frieder Bär? * 1/2
SELECT vname,nname,konr
FROM schueler NATURAL JOIN konto
WHERE vname = 'Frieder' AND nname = 'Bär';
```

SELECT vname,nname,konr FROM schueler NATURAL JOIN konto WHERE vname = 'Frieder' AND nname = 'Bär';		
vname	nname	konr
Frieder	Bär	1000

10.3.9 Tabelle aus Vorhilfe-Anfrage 09a: SELECT nname, vname FROM schueler WHERE email LIKE '%luise.net';

```
-- 09a. Liste der Schüler, die bei luise.net sind. **
SELECT nname, vname
FROM schueler
WHERE email LIKE '%luise.net';
```

SELECT nname, vname FROM schueler WHERE email LIKE '%luise.net';	
nname	vname
Bär	Frieder
Schmidt	Harald
Wolf	Hermine
Waal	Fritz
Gammel	Gabi
Schlau	Gabi
Gans	Gunter
Kannicht	Thomas
Kurz	Susanne
Braumeister	Thomas
Sabbel	Klara
Kunze	Max
Kolbe	Theresa
Kunze	Constanze
Witzel	Merve
Sonnenschein	Clara

10.3.10 Tabelle aus Vorhilfe-Anfrage-09b: SELECT nname,kuerzel FROM lehrer WHERE email LIKE '%vim.org';

```
-- 09b. Liste der Lehrer (Kuerzel), die bei vim.org sind. - Alternative **
SELECT nname,kuerzel
FROM lehrer
WHERE email LIKE '%vim.org';
```

SELECT nname,kuerzel FROM lehrer WHERE email LIKE '%vim.org';	
nname	kuerzel
Schaefer	sf

10.3.11 Tabelle aus Vorhilfe-Anfrage 9c: SELECT nname,kuerzel,email FROM lehrer WHERE email LIKE '%todesstern.de';

```
-- 09c. Liste der Lehrer (Name,Kuerzel,email), die bei todesstern.de sind. - Alternative **
SELECT nname,kuerzel,email
FROM lehrer
WHERE email LIKE '%todesstern.de';
```

SELECT nname,kuerzel,email FROM lehrer WHERE email LIKE '%todesstern.de';		
nname	kuerzel	email
Schmidt	si	darthvader@todesstern.de

10.3.12 Tabelle aus Vorhilfe-Anfrage 10 : SELECT vname,nname,kontostand FROM schueler NATURAL JOIN konto WHERE kontostand > 0 ORDER BY kontostand DESC;

```
-- 10. Welche Schüler haben Guthaben (nach Größe sortiert - Beginn höchster)? **
SELECT vname,nname,kontostand
FROM schueler NATURAL JOIN konto
WHERE kontostand > 0 ORDER BY kontostand DESC;
```

SELECT vname,nname,kontostand FROM schueler NATURAL JOIN konto WHERE kontostand > 0 ORDER BY kontostand DESC;		
vname	nname	kontostand
Harald	Schmidt	10
Theresa	Kolbe	10
Klara	Sabbel	8
Merve	Witzel	8
Clara	Sonnenschein	4
Gabi	Schlau	4
Gabi	Gammel	3
Thomas	Braumeister	3
Hermine	Wolf	3
Susanne	Kurz	2
Constanze	Kunze	2
Fritz	Waal	1
Gunter	Gans	1

**10.3.13 Tabelle aus Vorhilfe-Anfrage 11 : SELECT s.vname,s.nname,s.lid,s.email
FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS
f,schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr =
e.knr AND fname = 'Informatik';**

```
-- 11. Kursliste:
-- Alle Schüler mit Vor-, Nachname,Klassenlehrer und E-Mail,
-- die das Fach Informatik belegen **
SELECT s.vname,s.nname,s.lid,s.email
FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs ) AS f,
      schueler as s, eingeschrieben AS e
      WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';
```

SELECT s.vname,s.nname,s.lid,s.email FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f, schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';			
vname	nname	lid	email
Fritz	Waal	1	wale@luise.net
Gabi	Gammel	2	gaga@luise.net
Gabi	Schlau	3	schlau@luise.net
Thomas	Kannicht	4	konzert@luise.net
Gabi	Gammel	2	gaga@luise.net
Merve	Witzel	6	knaller@luise.net
Thomas	Kannicht	4	konzert@luise.net
Gabi	Gammel	2	gaga@luise.net

10.3.14 Tabelle aus Vorhilfe-Anfrage 12: SELECT nname,vname,geb,klasse FROM schueler NATURAL JOIN eingeschrieben WHERE age(geb) < '16 year' AND KNR = 2;

```
-- 12. Welcher Schüler ist in Kurs 2 unter 16 Jahre alt? **
SELECT nname,vname,geb,klasse
FROM schueler NATURAL JOIN eingeschrieben
WHERE age(geb) < '16 year' AND KNR = 2;
```

SELECT nname,vname,geb,klasse FROM schueler NATURAL JOIN eingeschrieben WHERE age(geb) < '16 year' AND KNR = 2;			
nname	vname	geb	klasse
Gans	Gunter	2003-07-01T00:00:00Z	10
Kannicht	Thomas	2004-03-03T00:00:00Z	09
Kurz	Susanne	2004-03-28T00:00:00Z	09

10.3.15 Tabelle aus Vorhilfe-Anfrage 13: SELECT COUNT(sid) AS Anzahl_eingeschrieben_in_Kurs_2 FROM eingeschrieben WHERE knr = 2;

```
-- 13. Wie viele Schüler sind im Kurs 2 eingeschrieben? **
SELECT COUNT(sid) AS Anzahl_eingeschrieben_in_Kurs_2
FROM eingeschrieben
WHERE knr = 2;
```

SELECT COUNT(sid) AS Anzahl_eingeschrieben_in_Kurs_2 FROM eingeschrieben WHERE knr = 2;
anzahl_eingeschrieben_in_kurs_2
4

10.3.16 Tabelle aus Vorhilfe-Anfrage 14: SELECT kuerzel, wtag, zeit FROM (kurs NATURAL JOIN beaufsichtigt NATURAL JOIN lehrer) ORDER BY kuerzel;

```
-- 14. Welcher Lehrer macht wann Aufsicht? **
SELECT kuerzel, wtag, zeit
FROM (kurs NATURAL JOIN beaufsichtigt NATURAL JOIN lehrer) ORDER BY kuerzel;
```

SELECT kuerzel, wtag, zeit FROM (kurs NATURAL JOIN beaufsichtigt NATURAL JOIN lehrer) ORDER BY kuerzel;		
kuerzel	wtag	zeit
bd	mo	0000-01-01T12:30:00Z
boe	di	0000-01-01T12:15:00Z
chi	do	0000-01-01T12:00:00Z
cz	di	0000-01-01T14:00:00Z
kl	mi	0000-01-01T12:15:00Z
ms	di	0000-01-01T13:00:00Z
sc	fr	0000-01-01T13:30:00Z
sc	di	0000-01-01T14:00:00Z
sf	mo	0000-01-01T12:30:00Z
si	mi	0000-01-01T14:30:00Z
wf	mi	0000-01-01T12:30:00Z

10.3.17 Tabelle aus Vorhilfe-Anfrage 15: SELECT * FROM raum WHERE anzP >= 30;

```
-- 15. Liste aller Räume, die min 30 Plätze haben. **
SELECT *
FROM raum
WHERE anzP >= 30;
```

SELECT * FROM raum WHERE anzP >= 30;		
rn timer	stock	anzp
301	3	32
201	2	32
202	2	30

10.3.18 Tabelle aus Vorhilfe-Anfrage 16: SELECT COUNT(kurs.KNR) AS Kurse_am_Donnerstag FROM Kurs WHERE wtag= 'do';

```
-- 16. Wie viele Kurse werden am Donnerstag angeboten? *** AGG
SELECT COUNT(kurs.KNR) AS Kurse_am_Donnerstag
FROM Kurs
WHERE wtag= 'do';
```

Vorhilfe-Anfrage 16
kurse_am_donnerstag
1

10.3.19 Tabelle aus Vorhilfe-Anfrage 17: SELECT COUNT(*) AS Anzahl_schueler_Mathematik FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f, schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Mathematik';

```
-- 17. Wie viele Schüler haben 'Mathematik' belegt? *** AAG
SELECT COUNT(*) AS Anzahl_schueler_Mathematik
FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,
      schueler as s, eingeschrieben AS e
WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Mathematik';
```

Vorhilfe-Anfrage 17
anzahl_schueler_mathematik
6

10.3.20 Tabelle aus Vorhilfe-Anfrage 18: SELECT sk.knr, sk.vname,sk.nname AS Kursleitung_Mathematik ,sk.wtag FROM(schueler NATURAL JOIN kurs) AS sk,(vermittelt NATURAL JOIN fach) AS vf WHERE fname = 'Mathematik' AND sk.knr = vf.knr;

```
-- 18. Wer leitet den Kurs 'Mathematik' an welchem Tag? ***
SELECT sk.knr, sk.vname,sk.nname AS Kursleitung_Mathematik ,sk.wtag
FROM(schueler NATURAL JOIN kurs) AS sk,(vermittelt NATURAL JOIN fach) AS vf
WHERE fname = 'Mathematik' AND sk.knr = vf.knr;
```

Vorhilfe-Anfrage 18			
knr	vname	kursleitung_mathematik	wtag
2	Frieder	Bär	di
6	Hermine	Wolf	di

10.3.21 Tabelle aus Vorhilfe-Anfrage 19: SELECT COUNT(*) AS Raeume_mit_Beamer FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung GROUP BY geraete HAVING geraete ='Beamer';

```
-- 19. Wie viele Räume haben einen Beamer? ***
SELECT COUNT( * ) AS Raeume_mit_Beamer
FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung
GROUP BY geraete HAVING geraete ='Beamer';
```

Vorhilfe-Anfrage 19
raeume_mit_beamer
2

10.3.22 Tabelle aus Vorhilfe-Anfrage 20a: SELECT rnr FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung WHERE geraete ='Beamer';

```
-- 20a. Welche Räume haben einen Beamer? ***
SELECT rnr
FROM raum NATURAL JOIN hat NATURAL JOIN ausstattung
WHERE geraete ='Beamer';
```

Vorhilfe-Anfrage 20a
rnr
300
-101

10.3.23 Tabelle aus Vorhilfe-Anfrage 20b: SELECT RNR FROM raum NATURAL JOIN hat WHERE ANR = 2;


```
-- 20b. alternativ (wenn Ausstattungsnummer von Beamer bekannt ist): **
SELECT RNR
FROM raum NATURAL JOIN hat
WHERE ANR = 2;
```

SELECT RNR FROM raum NATURAL JOIN hat WHERE ANR = 2;	
rnr	
300	
-101	

10.3.24 Tabelle aus Vorhilfe-Anfrage 21a: SELECT vname,nname,email,fname FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';

```
-- 21. Wieviele Plätze sind im Kurs "Informatik" noch frei? **** AGG
-- a) Kursliste Informatik ***
SELECT vname,nname,email,fname
FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs ) AS f,
      schueler as s, eingeschrieben AS e
WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';
```

Vorhilfe-Anfrage 21a			
vname	nname	email	fname
Fritz	Waal	wale@luise.net	Informatik
Gabi	Gammel	gaga@luise.net	Informatik
Gabi	Schlau	schlau@luise.net	Informatik
Thomas	Kannicht	konzert@luise.net	Informatik
Gabi	Gammel	gaga@luise.net	Informatik
Merve	Witzel	knaller@luise.net	Informatik
Thomas	Kannicht	konzert@luise.net	Informatik
Gabi	Gammel	gaga@luise.net	Informatik

10.3.25 Tabelle aus Vorhilfe-Anfrage 21b: SELECT MAX(f.maxP) - COUNT(fname) AS Freie_Plaetze_Informatik FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs) AS f,schueler as s, eingeschrieben AS e WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik'

```
-- b) Plätze in Informatik ****
SELECT MAX(f.maxP) - COUNT(fname) AS Freie_Plaetze_Informatik
```

```
FROM ((fach NATURAL JOIN vermittelt) NATURAL JOIN kurs ) AS f,
      schueler AS s, eingeschrieben AS e
WHERE e.sid = s.sid AND f.knr = e.knr AND fname = 'Informatik';
```

Vorhilfe-Anfrage 21b
freie_plaetze_informatik
1

Listings

Create-Vorhilfe.sql	7
Create-Vorhilfe.sql	7
Create-Vorhilfe.sql	8
Create-Vorhilfe.sql	9
Create-Vorhilfe.sql	9
./query/Vorhilfe-query_01.txt	22
./query/Vorhilfe-query_02.txt	23
./query/Vorhilfe-query_03.txt	23
./query/Vorhilfe-query_04.txt	24
./query/Vorhilfe-query_05.txt	24
./query/Vorhilfe-query_06.txt	25
./query/Vorhilfe-query_07.txt	25
./query/Vorhilfe-query_08.txt	26
./query/Vorhilfe-query_09a.txt	26
./query/Vorhilfe-query_09b.txt	27
./query/Vorhilfe-query_09c.txt	27
./query/Vorhilfe-query_10.txt	27
./query/Vorhilfe-query_11.txt	28
./query/Vorhilfe-query_12.txt	29
./query/Vorhilfe-query_13.txt	29
./query/Vorhilfe-query_14.txt	30
./query/Vorhilfe-query_15.txt	30
./query/Vorhilfe-query_16.txt	31
./query/Vorhilfe-query_17.txt	31
./query/Vorhilfe-query_18.txt	31
./query/Vorhilfe-query_19.txt	32
./query/Vorhilfe-query_20a.txt	32
./query/Vorhilfe-query_20b.txt	33
./query/Vorhilfe-query_21a.txt	33
./query/Vorhilfe-query_21b.txt	33