

Einzelprüfung „Softwaretechnologie / Datenbanksysteme (nicht vertieft)“

Einzelprüfungsnummer 46116 / 2018 / Herbst

Thema 2 / Teilaufgabe 2 / Aufgabe 3

(Schuldatenbank)

Stichwörter: SQL, SQL mit Übungsdatenbank, CREATE TABLE, INSERT, ALTER TABLE, UPDATE, DELETE, VIEW, GROUP BY, DROP TABLE

Gegeben sei das folgende Datenbank-Schema, das für die Speicherung der Daten einer Schule entworfen wurde, zusammen mit einem Teil seiner Ausprägung. Die Primärschlüssel-Attribute sind jeweils unterstrichen.

Die Relation *Schüler* enthält allgemeine Daten zu den Schülerinnen und Schülern. Schülerinnen und Schüler nehmen an Prüfungen in verschiedenen Unterrichtsfächern teil und erhalten dadurch Noten. Diese werden in der Relation *Noten* abgespeichert. Prüfungen haben ein unterschiedliches Gewicht. Beispielsweise hat ein mündliches Ausfragen oder eine Extemporale das Gewicht 1, während eine Schulaufgabe das Gewicht 2 hat.

Schüler:

<u>SchülerID</u>	Vorname	Nachname	Klasse
1	Laura	Müller	4A
2	Linus	Schmidt	4A
3	Jonas	Schneider	4A
4	Liam	Fischer	4B
5	Tim	Weber	4B
6	Lea	Becker	4B
7	Emilia	Klein	4C
8	Julia	Wolf	4C

Noten:

<u>SchülerID</u> [<u>Schüler</u>]	Schulfach	Note	Gewicht	Datum
1	Mathematik	3	2	23.09.2017
1	Mathematik	1	1	03.10.2017
1	Mathematik	2	2	15.10.2017
1	Mathematik	4	1	11.11.2017

Additum: Übungsdatenbank

```
CREATE TABLE Schüler (  
  SchülerID INTEGER PRIMARY KEY NOT NULL,  
  Vorname VARCHAR(20),  
  Nachname VARCHAR(20),  
  Klasse VARCHAR(5)  
);
```

```
CREATE TABLE Noten (  
  SchülerID INTEGER NOT NULL,  
  Schulfach VARCHAR(20),
```

```

Note INTEGER,
Gewicht INTEGER,
Datum DATE,
PRIMARY KEY (SchülerID, Schulfach, Datum),
FOREIGN KEY (SchülerID) REFERENCES Schüler(SchülerID)
);

INSERT INTO Schüler VALUES
(1, 'Laura', 'Müller', '4A'),
(2, 'Linus', 'Schmidt', '4A'),
(3, 'Jonas', 'Schneider', '4A'),
(4, 'Liam', 'Fischer', '4B'),
(5, 'Tim', 'Weber', '4B'),
(6, 'Lea', 'Becker', '4B'),
(7, 'Emilia', 'Klein', '4C'),
(8, 'Julia', 'Wolf', '4C');

INSERT INTO Noten VALUES
(1, 'Mathematik', 3, 2, '23.09.2017'),
(1, 'Mathematik', 1, 1, '03.10.2017'),
(1, 'Mathematik', 2, 2, '15.10.2017'),
(1, 'Mathematik', 4, 1, '11.11.2017');

```

- (a) Geben Sie die SQL-Befehle an, die notwendig sind, um die oben dargestellten Tabellen in einer SQL-Datenbank anzulegen.

Lösungsvorschlag

```

CREATE TABLE IF NOT EXISTS Schüler (
  SchülerID INTEGER PRIMARY KEY NOT NULL,
  Vorname VARCHAR(20),
  Nachname VARCHAR(20),
  Klasse VARCHAR(5)
);

CREATE TABLE IF NOT EXISTS Noten (
  SchülerID INTEGER NOT NULL,
  Schulfach VARCHAR(20),
  Note INTEGER,
  Gewicht INTEGER,
  Datum DATE,
  PRIMARY KEY (SchülerID, Schulfach, Datum),
  FOREIGN KEY (SchülerID) REFERENCES Schüler(SchülerID)
);

```

- (b) Entscheiden Sie jeweils, ob folgende Einfügeoperationen vom gegebenen Datenbanksystem (mit der angegebenen Ausprägungen) erfolgreich verarbeitet werden können und begründen Sie Ihre Antwort kurz.

```

INSERT INTO Schüler
(SchülerID, Vorname, Nachname, Klasse)
VALUES
(6, 'Johannes', 'Schmied', '4C');

```

Lösungsvorschlag

Nein. Ein/e Schüler/in mit der ID 6 existiert bereits. Primärschlüssel müssen eindeutig sein.

```
INSERT INTO Noten VALUES (6, 'Chemie', 1, 2, '1.4.2020');
```

Lösungsvorschlag

Nein. Ein *Datum* ist zwingend notwendig. Da *Datum* im Primärschlüssel enthalten ist, darf es nicht NULL sein. Es gibt auch keine/n Schüler/in mit der ID 9. Der/die Schüler/in müsste vorher angelegt werden, da die Spalte *SchülerID* von der Tabelle *Noten* auf den Fremdschlüssel *SchülerId* aus der Schülertabelle verweist.

- (c) Geben Sie die Befehle für die folgenden Aktionen in SQL an. Beachten Sie dabei, dass die Befehle auch noch bei Änderungen des oben gegebenen Datenbankzustandes korrekte Ergebnisse zurückliefern müssen.

- Die Schule möchte verhindern, dass in die Datenbank mehrere Kinder mit dem selben Vornamen in die gleiche Klasse kommen. Dies soll bereits auf Datenbankebene verhindert werden. Dabei sollen die Primärschlüssel nicht verändert werden. Geben Sie den Befehl an, der diese Änderung durchführt.

Lösungsvorschlag

```
ALTER TABLE Schüler  
ADD CONSTRAINT eindeutiger_Vorname UNIQUE (Vorname, Klasse);
```

- Der Schüler *Tim Weber* (SchülerID: 5) wechselt die Klasse. Geben Sie den SQL-Befehl an, der den genannten Schüler in die Klasse „4C“ überführt.

Lösungsvorschlag

```
UPDATE Schüler  
SET Klasse = '4C'  
WHERE  
  Vorname = 'Tim' AND  
  Nachname = 'Weber' AND  
  SchülerID = 5;
```

- Die Schülerin *Laura Müller* (SchülerID: 1) zieht um und wechselt die Schule. Löschen Sie die Schülerin aus der Datenbank. Nennen Sie einen möglichen Effekt, welcher bei der Verwendung von Primär- und Fremdschlüsseln auftreten kann.

Lösungsvorschlag

Alle Noten von *Laura Müller* werden gelöscht, falls `ON DELETE CASCADE` gesetzt ist. Oder es müssen erst alle Fremdschlüsselverweise auf diese *SchülerID* in der Tabelle *Noten* gelöscht werden

```
DELETE FROM Noten  
WHERE SchülerID = 1;
```

- Erstellen Sie eine View „DurchschnittsNoten“, die die folgenden Spalten beinhaltet: *Klasse, Schulfach, Durchschnittsnote*

Hinweis: Beachten Sie die Gewichte der Noten.

Lösungsvorschlag

```
CREATE VIEW DurchschnittsNoten AS (
  (SELECT s.Klasse, n.Schulfach, (SUM(n.Note * n.Gewicht) / SUM(n.Gewicht))
  ↪ AS Durchschnittsnote
  FROM Noten n, Schüler s
  WHERE s.SchülerID = n.SchülerID
  GROUP BY s.Klasse, n.Schulfach)
);

SELECT * FROM DurchschnittsNoten;
```

klasse	schulfach	durchschnittsnote
4A	Mathematik	2

(1 row)

- Geben Sie den Befehl an, der die komplette Tabelle „Noten“ löscht.

Lösungsvorschlag

```
DROP TABLE Noten;
```

- (d) Formulieren Sie die folgenden Anfragen in SQL. Beachten Sie dabei, dass sie SQL-Befehle auch noch bei Änderungen der Ausprägung die korrekten Anfrageergebnisse zurückgeben sollen.

- Gesucht ist die durchschnittliche Note, die im Fach Mathematik vergeben wird.
Hinweis: Das Gewicht ist bei dieser Anfrage nicht relevant

Lösungsvorschlag

```
SELECT AVG(Note)
FROM Noten
WHERE Schulfach = 'Mathematik';
```

- Berechnen Sie die Anzahl der Schüler, die im Fach Mathematik am 23.09.2017 eine Schulaufgabe (d. h. Gewicht=2) geschrieben haben.

Lösungsvorschlag

```
SELECT COUNT(*) AS Anzahl_Schüler
FROM Noten
WHERE Datum = '23.09.2017' AND Gewicht = 2 AND Schulfach = 'Mathematik';
```

anzahl_schüler
1

(1 row)

- Geben Sie die *SchülerID* aller Schüler zurück, die im Fach Mathematik mindestens drei mal die Schulnote 6 geschrieben haben.

Lösungsvorschlag

```

SELECT SchülerID
FROM Noten
WHERE Schulfach = 'Mathematik' AND Note = 6
GROUP BY SchülerID
HAVING COUNT(*) >= 3;

schülerid
-----
(0 rows)

```

- Gesucht ist der Notendurchschnitt bezüglich jedes Fachs der Klasse „4A“.

Lösungsvorschlag

```

SELECT n.Schulfach, AVG(n.Note)
FROM Schüler s, Noten n
WHERE s.SchülerID = n.SchülerID AND s.Klasse = '4A'
GROUP BY n.Schulfach;

schulfach |      avg
-----+-----
Mathematik | 2.5000000000000000
(1 row)

```

- (e) Geben Sie jeweils an, welchen Ergebniswert die folgenden SQL-Befehle für die gegebene Ausprägung zurückliefern.

```

SELECT COUNT(DISTINCT Klasse)
FROM
Schüler NATURAL JOIN Noten;

```

Lösungsvorschlag

```

count
-----
1
(1 row)

```

4A von Laura Müller. Ohne **DISTINCT** wäre das Ergebnis 4.

```

SELECT COUNT(ALL Klasse)
FROM
Noten, Schüler;

```

Lösungsvorschlag

```

count
-----
32
(1 row)

```

Es entsteht das Kreuzprodukt ($8 \cdot 4 = 32$).

```

SELECT COUNT(Note)
FROM
Schüler NATURAL LEFT OUTER JOIN Noten;

```

Lösungsvorschlag

```
SELECT * FROM Schüler NATURAL LEFT OUTER JOIN Noten;
```

ergibt:

schülerid	vorname	nachname	klasse	schulfach	note	gewicht	datum
1	Laura	Müller	4A	Mathematik	3	2	2017-09-23
1	Laura	Müller	4A	Mathematik	1	1	2017-10-03
1	Laura	Müller	4A	Mathematik	2	2	2017-10-15
1	Laura	Müller	4A	Mathematik	4	1	2017-11-11
2	Linus	Schmidt	4A				
5	Tim	Weber	4B				
8	Julia	Wolf	4C				
6	Lea	Becker	4B				
4	Liam	Fischer	4B				
3	Jonas	Schneider	4A				
7	Emilia	Klein	4C				

(11 rows)

```
count
-----
      4
(1 row)
```

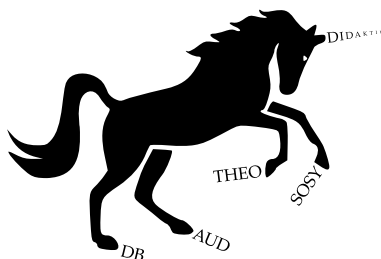
COUNT zählt die **{NULL}**-Werte nicht mit. Die *Laura Müller* hat 4 Noten.

```
SELECT COUNT(*)
FROM
Schüler NATURAL LEFT OUTER JOIN Noten;
```

Lösungsvorschlag

```
count
-----
     11
(1 row)
```

Siehe Zwischenergebnistabelle in der obenstehenden Antwort. Alle Schüler und die Laura 4-mal, weil sie 4 Noten hat.



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/46116/2018/09/Thema-2/Teilaufgabe-2/Aufgabe-3.tex>