

SQL-Grundlagen

Datentypen¹

CHAR(*n*) Zeichenreihe (String) einer festen Länge von *n* Zeichen

VARCHAR(*n*) Zeichenreihe variabler Länge (String) mit bis zu *n* Zeichen

INTEGER, SMALLINT Ganze Zahl (mit oder ohne Vorzeichen)

NUMERIC(*p,q*), DEC(*p,q*) Dezimalzahl mit *p* Ziffern insgesamt und *q* Ziffern nach dem Dezimalpunkt

FLOAT(*n*), FLOAT, REAL Gleitkommazahl mit Stelligkeit *n* oder systemabhängiger Genauigkeit

DATE, TIME, TIMESTAMP Datum und/oder Uhrzeit

BOOLEAN Boolean constants can be represented in SQL queries by the SQL key words TRUE, FALSE, and NULL

CREATE TABLE

```
1 CREATE TABLE Tabellennamen
2 (
3     Spaltenname1 Datentyp1 [Spaltenbedingung]
4     Spaltenname2 Datentyp2 [Spaltenbedingung]
5     [...]
6 );
```

Mengenoperationen²

Mengenoperation	SQL - Schlüsselwort
Vereinigung von Mengen	UNION
Schnitt von Mengen	INTERSECT
Differenz von Mengen	EXCEPT

Referentielle Integrität³

Unter Referentieller Integrität verstehen wir Bedingungen, die zur Sicherung der Datenintegrität bei Nutzung relationaler Datenbanken beitragen können. Demnach dürfen Datensätze über ihre Fremdschlüssel nur auf existierende Datensätze verweisen.

¹Winter, Lindner und Würdinger, *Einführung in relationale Datenbanksysteme & Datenmodellierung*, Seite 132.

²Winter, Lindner und Würdinger, *Einführung in relationale Datenbanksysteme & Datenmodellierung*, Seite 157.

³Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 14.

Danach besteht die Referentieller Integrität grundsätzlich aus zwei Teilen:

- (a) Ein neuer Datensatz mit einem Fremdschlüssel kann nur dann in einer Tabelle eingefügt werden, wenn in der referenzierten Tabelle ein Datensatz mit entsprechendem Wert im Primärschlüssel oder einem eindeutigen Alternativschlüssel existiert.
- (b) Eine Datensatzlöschung oder Änderung des Schlüssels in einem Primär-Datensatz ist nur möglich, wenn zu diesem Datensatz keine abhängigen Datensätze in Beziehung stehen.⁴

Integritätsbedingungen:⁵

NOT NULL Alle eingetragenen Tupel müssen an dieser Stelle einen definierten Wert haben.

UNIQUE Kennzeichnung von Schlüsselkandidaten. Gibt an, dass dieses Attribut eindeutig sein muss.

PRIMARY KEY Primärschlüssel-Kennzeichnung, impliziert automatisch **NOT NULL**, jede Tabelle benötigt einen Primärschlüssel!

```
1 CREATE TABLE Customers (  
2   Id INTEGER NOT NULL,  
3   Name VARCHAR(20) NOT NULL,  
4   PRIMARY KEY (Id, Name)  
5 );
```

FOREIGN KEY Kennzeichnung von Fremdschlüsseln, null-Werte möglich, wenn nicht explizit **NOT NULL** gesetzt (**UNIQUE FOREIGN KEY** um 1:1-Beziehung zu modellieren)

```
1 CREATE TABLE ist_Fachlehrkraft_von (  
2   Klasse INTEGER NOT NULL,  
3   Fach VARCHAR(20) NOT NULL,  
4   Lehrkraft INTEGER NOT NULL,  
5   Stundenzahl INTEGER NOT NULL,  
6   PRIMARY KEY (Klasse, Fach, Lehrkraft),  
7   FOREIGN KEY (Klasse) REFERENCES Klasse(Name),  
8   FOREIGN KEY (Fach) REFERENCES Fach(Name),  
9   FOREIGN KEY (Lehrkraft) REFERENCES Lehrkraft(PersNr)  
10 );
```

CHECK (Attributname IN / (BETWEEN) AND) Überprüfung der angegebenen Werte auf eine bestimmte Eigenschaft.

DEFAULT Wert setzt das Attribut auf einen bestimmten Standardwert, wenn kein Wert für das Attribut angegeben wird

ON DELETE / ON UPDATE SET NULL / SET DEFAULT / CASCADE für Fremdschlüssel, setzt im Fall des Updates bzw. Löschens des Primärschlüssels in der anderen Tabelle diesen Attributwert auf **NULL** / einen Defaultwert oder ändert / löscht ihn kaskadierend

⁴Wikipedia-Artikel „Referentielle Integrität“.

⁵Qualifizierungsmaßnahme Informatik - Datenbanksysteme 2, Seite 7.

Aggregatsfunktionen⁶

AVG() berechnet den arithmetischen Mittelwert einer Menge von Werten in einem bestimmten Feld einer Abfrage

COUNT() berechnet die Anzahl der von einer Abfrage zurückgegebenen Datensätze.

MAX()/ MIN() - gibt den größten / kleinsten Wert aus einer Reihe von Werten zurück, die in einem bestimmten Feld einer Abfrage enthalten sind.

SUM() - berechnet die Summe einer Menge von Werten in einem bestimmten Feld einer Abfrage.

Besonderheiten bei Aggregatoperationen

- SQL erzeugt pro Gruppe ein Ergebnistupel
- Deshalb müssen alle in der SELECT-Klausel aufgeführten Attribute - außer den aggregierten – auch in der GROUP BY-Klausel aufgeführt werden
- Nur so kann SQL sicherstellen, dass sich das Attribut nicht innerhalb der Gruppe ändert

Geschachtelte Anfragen in SQL

(Foliensatz Seite 26)

SQL erlaubt das Schachteln von Anfragen

Unterscheidung nach Anzahl der Rückgabetupel

Unterscheidung von Anfragen, die nur ein Tupel zurückgeben und Anfragen, die beliebig viele Tupel ergeben

Wenn nur ein Tupel mit nur einem Attribut zurückgeliefert wird, dann kann die Anfrage dort eingesetzt werden, wo ein skalarer Wert gefordert wird → vor allem in WHERE und SELECT-Klauseln

Korrelierte vs. Nicht-korrelierte Anfragen

Nicht-korrelierte Anfragen: verwenden lediglich „eigene“ Attribute, müssen nur einmal ausgewertet werden, zudem kann ihr Ergebnis materialisiert werden → wenn möglich ist dies die bessere Lösung!

```
1 SELECT *
2 FROM prüfen
3 WHERE Note < (SELECT AVG (Note) FROM prüfen);
```

Korrelierte Anfragen: greifen auf Attribute der umschließenden Anfrage zu, werden daher für jedes Tupel der umschließenden Anfrage neu berechnet

⁶Qualifizierungsmaßnahme Informatik - Datenbanksysteme 2, Seite 10.

```

1 SELECT PersNr, Name, (
2     SELECT SUM(SWS) AS Lehrbelastung
3     FROM Vorlesungen
4     WHERE gelesenVon = PersNr
5 )
6 FROM Professoren;

```

Verwertung der Ergebnismenge einer Unteranfrage

```

1 SELECT tmp.MatrNr, tmp.Name, tmp.VorlAnzahl
2 FROM (
3     SELECT s.MatrNr, s.Name, COUNT(*) AS VorlAnzahl
4     FROM Studenten s, hören h
5     WHERE s.MatrNr = h.MatrNr
6     GROUP BY s.MatrNr, s.Name
7 ) tmp
8 WHERE tmp.VorlAnzahl > 2;

```

Geht allerdings schöner: WITH → erlaubt Modularisierung, besonders nützlich, wenn ich das Ergebnis meiner Unterabfrage in einer Reihe weiterer SQL-Abfragen ebenfalls benötige

```

1 WITH tmp AS (
2     SELECT s.MatrNr, s.Name, COUNT(*) AS VorlAnzahl
3     FROM Studenten s, hören h
4     WHERE s.MatrNr = h.MatrNr
5     GROUP BY s.MatrNr, s.Name
6 )
7 SELECT tmp.MatrNr, tmp.Name, tmp.VorlAnzahl
8 FROM tmp
9 WHERE tmp.VorlAnzahl > 2;

```

Trigger

```

1 CREATE [OR REPLACE] TRIGGER <trigger_name>
2 [BEFORE|AFTER|INSTEAD OF] [INSERT|DELETE|UPDATE [OF <feld_name>]]
3 ON <table_name>
4 [REFERENCING [NEW AS <new_row_name>] [OLD AS <old_row_name>]]
5 [FOR EACH ROW [WHEN (<trigger_condition>)]]
6 <trigger_body>
7

```

Top-N-Query⁸

Ermittle die TOP(10) der Spitzenverdiener für jede Abteilung und gib den Rang und den Namen aus. (Alle Angestellten haben unterschiedliches Gehalt)

```

1 SELECT P.AbtNr, COUNT(*) AS Rang, P.Nachname
2 FROM Angestellte P, Angestellte A
3 WHERE P.AbtNr = A.AbtNr AND P.Gehalt <= A.Gehalt
4 GROUP BY P.PersNr, P.Nachname, P.AbtNr
5 HAVING COUNT(*) <= 10;

```

⁷Wikipedia-Artikel „Datenbanktrigger“.

⁸Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 8.

Rekursive SQL-Abfragen

Weiterführende Literatur:

- Kemper und Eickler, *Datenbanksysteme*, Seite 136
- *Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3*, Seite 27-31
- *10 SQL-Tricks, die du nicht für möglich gehalten hättest*

```
1 WITH RECURSIVE t(v) AS (  
2   SELECT 1      -- Seed Row  
3   UNION ALL  
4   SELECT v + 1 -- Recursion  
5   FROM t  
6 )  
7 SELECT v  
8 FROM t  
9 LIMIT 5
```

9

Was ist Rekursion? Grundsätzliche Idee: Bestimmte Vorgänge werden auf ein Produkt, das sie bereits hervorgebracht haben, von neuem angewandt. Dabei entstehen unendliche Abläufe, die mit Hilfe bestimmter Bedingungen abgebrochen werden.

Wann ist das erforderlich?

Typisches Beispiel: In einem Unternehmen sollen Hierarchien erfasst werden. Wir möchten für einen Chef alle Untergebenen ermitteln, aber eben nicht nur auf der Hierarchieebene direkt unter ihm, sondern ALLE. Dabei scheitert eine klassische Anweisung!

Basisaufbau:

```
1 WITH tabelle (spaltenliste) AS  
2 ( UrsprungsSELECT  
3 UNION ALL  
4 RekursionsSELECT )  
5 SELECT spaltenliste FROM tabelle WHERE ...
```

Hilfreiche Fragen zur Erstellung:

Welche Spalten sollen in meiner Ergebnismenge auftauchen und/oder werden für die Rekursionsbedingung benötigt?

Wie lautet der SELECT für den Satz, von dem die Rekursion ausgehen soll?

Wie lautet die Rekursionsbedingung?

Ansichten (VIEWS)

Aus dem SQL92 Standard:

```
<view definition> ::=  
  CREATE VIEW <table name> [ <left paren> <view column list>  
  <right paren> ]  
  AS <query expression>  
  [ WITH [ <levels clause> ] CHECK OPTION ]
```

⁹10 SQL-Tricks, die du nicht für möglich gehalten hättest.

```

<levels clause> ::=
    CASCADED | LOCAL

<view column list> ::= <column name list>

```

10
Beispiele¹¹

```

1 CREATE VIEW StudProf (Sname, Semester, Titel, Pname) AS
2 SELECT s.Name, s.Semester, v.Titel, p.Name
3 FROM Studenten s, hören h, Vorlesungen v, Professoren p
4 WHERE s.Matr.Nr = h.MatrNr AND
5       h.VorlNr=v.VorlNr AND
6       v.gelesenVon = p.PersNr;

```

12

Literatur

- [1] 10 SQL-Tricks, die du nicht für möglich gehalten hättest. <https://jaxenter.de/10-sql-tricks-39974>. aufgerufen 2020-06-25.
- [2] Database Language SQL - July 30, 1992. (Second Informal Review Draft) ISO/IEC 9075:1992. <https://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>.
- [3] Alfons Kemper und André Eickler. *Datenbanksysteme. eine Einführung*. 2013.
- [4] Qualifizierungsmaßnahme Informatik - Datenbanksysteme 2. Einführung SQL, Relationale Algebra. https://www.studon.fau.de/file2480903_download.html.
- [5] Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3. SQL Vertiefung, Integrität, Physische Datenorganisation. https://www.studon.fau.de/file2480904_download.html.
- [6] Wikipedia-Artikel „Datenbanktrigger“. <https://de.wikipedia.org/wiki/Datenbanktrigger>.
- [7] Wikipedia-Artikel „Referentielle Integrität“. https://de.wikipedia.org/wiki/Referentielle_Integrität.
- [8] Stefan Winter, Annabel Lindner und Markus Würdinger. *Einführung in relationale Datenbanksysteme & Datenmodellierung*. https://www.studon.fau.de/file2686598_download.html.

¹⁰Database Language SQL - July 30, 1992, Seite 296.

¹¹Kemper und Eickler, *Datenbanksysteme*, Seite 140.

¹²Qualifizierungsmaßnahme Informatik - Datenbanksysteme 3, Seite 11.