Kapitel 3: Der SQL-Standard

Ein *Anfrageausdruck* in SQL besteht aus einer `SELECT`-Klausel, gefolgt von einer `FROM`-Klausel, gefolgt von einer `WHERE`-Klausel.

SFW-Ausdruck im Allgemeinen

```
SELECT A1,...,An  (...Attribute der Ergebnisrelation, bzw.
                     beliebige attributwertige SQL-Ausdrücke)
   FROM R1,...,Rm (...benötigte Relationen, bzw.
                     beliebige relationenwertige SQL-Ausdrücke)
   WHERE F        (...Auswahlbedingung, bzw.
                     beliebige boolesche SQL-Ausdrücke)
```

Mondial relationale Datenbank Teil 1

Land

LName	LCode	HStadt	Fläche
Austria	A	Vienna	84
Egypt	ET	Cairo	1001
France	F	Paris	547
Germany	D	Berlin	357
Italy	I	Rome	301
Russia	RU	Moscow	17075
Switzerland	CH	Bern	41
Turkey	TR	Ankara	779

Provinz

PName	LCode	Fläche
Baden	D	15
Bavaria	D	70,5
Berlin	D	0,9
Ile de France	F	12
Franken	D	null
Lazio	I	17

Stadt

SName	LCode	PName	Einwohner	LGrad	BGrad
Berlin	D	Berlin	3472	13,2	52,45
Freiburg	D	Baden	198	7,51	47,59
Karlsruhe	D	Baden	277	8,24	49,03
Munich	D	Bavaria	1244	11,56	48,15
Nuremberg	D	Franken	495	11,04	49,27
Paris	F	Ile de France	2125	2,48	48,81
Rome	I	Lazio	2546	12,6	41,8

Mondial relationale Datenbank Teil 2

Lage

LCode	Kontinent	Prozent
D	Europe	100
F	Europe	100
TR	Asia	68
TR	Europe	32
ET	Africa	90
ET	Asia	10
RU	Asia	80
RU	Europe	20

Mitglied

LCode	Organisation	Art
A	EU	member
D	EU	member
D	WEU	member
ET	UN	member
I	EU	member
I	NAM	guest
TR	UN	member
TR	CERN	observer

3.1 Nullwerte

Die Problematik

- Liegt zu einem Attribut kein Wert vor, so kann dies durch Verwendung des *Nullwerts* `null` ausgedrückt werden.
- Als mögliche Interpretationen eines Nullwertes können wir unterscheiden: *Wert existiert, jedoch zur Zeit unbekannt* - *Wert existiert erst in der Zukunft* - *Wert prinzipiell unbekannt* - oder auch *Attribut nicht anwendbar*.

Beispiel

Student

MatrNr	Name	Adresse	Semester	Exmatrikulationsdatum	Mutterschutz
1223	Hans Eifrig	null	2	null	null
3434	Lisa Lustig	Bergstraße 11	4	null	ja
1234	Maria Gut	Am Bächle 1	null	null	nein

Nullwerte und SQL

- SQL bietet die Prädikate `IS NULL` und `IS NOT NULL` an, um auf Existenz von Nullwerten prüfen zu können.
- In Ausdrücken der Form A+B, A+1, etc. ist das Resultat `null`, wenn einer der Operanden `null` ist.
- Ausdrücke mit Vergleichsoperatoren der Form A=B, A<>B, A<B, etc. haben den Wahrheitswert `UNKNOWN`, wenn mindestens einer der beteiligten Operanden den Wert `null` besitzt.
- SQL liegt eine dreiwertige Logik zugrunde. (`t=TRUE, f=FALSE, u=UNKNOWN`).

Bestimme alle Provinzen, zu denen die Fläche bekannt ist.

```
SELECT * FROM Provinz
   WHERE Fläche IS NOT NULL
```

Wahrheitswerte

AND	t	u	f
t	t	u	f
u	u	u	f
f	f	f	f

OR	t	u	f
t	t	t	t
u	t	u	u
f	t	u	f

NOT	
t	f
u	u
f	t

Welche Ausgabe liefert:

```
SELECT * FROM Provinz
   WHERE NOT (Fläche > 0)
```

Vermeide Nullwerte wann immer es geht!

... denn sie verkomplizieren die Anfrageformulierung und ihre Semantik bzgl. der realen Welt ist nicht eindeutig.

.... eine sinnvolle Anwendung von Null-Werten: *äußerer Verbund* (engl. outer join).

Land

LName	LCode	HStadt	Fläche
Austria	A	Vienna	84
Egypt	ET	Cairo	1001
France	F	Paris	547
Germany	D	Berlin	357
Italy	I	Rome	301
Russia	RU	Moscow	17075
Switzerland	CH	Bern	41
Turkey	TR	Ankara	779

Stadt

SName	LCode	PName	Einwohner	LGrad	BGrad
Berlin	D	Berlin	3472	13,2	52,45
Freiburg	D	Baden	198	7,51	47,59
Karlsruhe	D	Baden	277	8,24	49,03
Munich	D	Bavaria	1244	11,56	48,15
Nuremberg	D	Franken	495	11,04	49,27
Paris	F	Ile de France	2125	2,48	48,81
Rome	I	Lazio	2546	12,6	41,8

Wieviel Einwohner haben die Hauptstädte der einzelnen Länder?

```
SELECT L.LName AS Land, L.HStadt, S.Einwohner
   FROM Land L LEFT OUTER JOIN Stadt S
   ON L.HStadt = S.SName
```

LName	HStadt	Einwohner
Austria	Vienna	null
Egypt	Cairo	null
France	Paris	2125
Germany	Berlin	3472
Italy	Rome	2546
Russia	Moscow	null
Switzerland	Bern	null
Turkey	Ankara	null

- Bei einem `Left OUTER JOIN` bleiben die Tupel der linken Relation erhalten; rechts werden bei fehlenden Verbundpartnern Nullwerte ergänzt,
- bei einem `RIGHT OUTER JOIN` werden analog gegebenenfalls links Nullwerte ergänzt,
- und ein `FULL OUTER JOIN` berechnet die Vereinigung des entsprechenden `LEFT OUTER JOIN` und `RIGHT OUTER JOIN`.

3.2 Anfragen mit Aggregierungsfunktionen

COUNT, MIN, MAX, SUM und AVG

Wieviele Länder gibt es in der Tabelle Land, wie groß ist die maximale, die minimale Fläche und die durchschnittliche Fläche aller Länder?

```
SELECT COUNT(LCode),MAX(Fläche),MIN(Fläche),AVG(Fläche)
   FROM Land
```

Wieviele Länder haben eine Mitgliedschaft bzgl. der EU?

```
SELECT COUNT(*) AS AnzEU
   FROM Mitglied
   WHERE Organisation = 'EU'
```

Wieviele unterschiedliche Organisationen werden in Mitglied aufgeführt?

```
SELECT COUNT(DISTINCT Organisation) FROM Mitglied
```

Besonderheiten

- `COUNT(*)` liefert die Anzahl Zeilen der Tabelle, die sich nach Auswerten der `FROM`- und `WHERE`-Klausel ergeben hat.
- ```
  SELECT LName, MAX(Fläche) FROM Land
  ```
  ist syntaktisch nicht zulässig, da ein Aggregierungsoperator eine Menge von Zeilen auf einen einzigen Wert reduziert. Zulässig wäre:
  ```
  SELECT MAX(Fläche) FROM Land
  ```
- Aggregierungsfunktionen ignorieren für ihre Berechnungen Nullwerte.
- Eine Ausnahme ist `COUNT(*)`; hier werden auch alle Zeilen, in denen alle Spalten `null` sind, mitgezählt.
  ```

3.3 Anfragen mit Gruppierungen

- Mittels einer *Gruppierung* können wir eine virtuelle Struktur über einer Tabelle definieren.
- Die Gruppierungsattribute fassen alle Zeilen der Tabelle jeweils zu einer Gruppe zusammen, die bezüglich aller Gruppierungsattribute gleiche Werte haben und zusätzlich die in einer optionalen `HAVING`-Klausel festgelegten Bedingungen erfüllt.
- Anfragen über einer gruppierten Tabelle betrachten die einzelnen Gruppen zusammen mit den Gruppierungsattributen analog zu einer Zeile einer Tabelle.

  Konsequenterweise dürfen Attribute, die nicht als Gruppierungsattribute verwendet wurden, nur als Parameter für Aggregierungsfunktionen verwendet werden.

Wie groß ist die durchschnittliche Einwohnerzahl der Städte der jeweiligen Länder?

```
SELECT LCode, AVG(Einwohner) FROM Stadt
   GROUP BY LCode
```

In welchen Ländern ist die durchschnittliche Einwohnerzahl kleiner 2 Mio.?

```
SELECT LCode, AVG(Einwohner) FROM Stadt
   GROUP BY LCode
   HAVING AVG(Einwohner) < 2000
```

Bestimme die Einwohnerzahlen der drei größten Städte.

```
SELECT DISTINCT COUNT(*) AS Rang, A.Einwohner
   FROM Stadt A, Stadt B
   WHERE (A.Einwohner <= B.Einwohner)
   GROUP BY A.Einwohner
   HAVING COUNT(*) <= 3
   ORDER BY Rang
```

Bestimme die Einwohnerzahlen der drei größten Städte.

```
SELECT DISTINCT COUNT(*) AS Rang, A.Einwohner
   FROM Stadt A, Stadt B
   WHERE (A.Einwohner <= B.Einwohner)
   GROUP BY A.Einwohner
   HAVING COUNT(*) <= 3
   ORDER BY Rang
```

Bestimme die Einwohnerzahlen der drei größten Städte.

```
SELECT MAX(A.Einwohner) AS Rang1,
      MAX(B.Einwohner)AS Rang2,
      MAX(C.Einwohner) AS Rang3
   FROM Stadt A, Stadt B, Stadt C
   WHERE (A.Einwohner > B.Einwohner)
      AND (B.Einwohner > C.Einwohner)
```

SFW-Ausdruck

```
SELECT A_1,...,A_n        Liste der Attribute
   FROM R_1,...,R_m       Liste der Relationen
   WHERE F                Bedingung
   GROUP BY B_1,...,B_k   Liste der Gruppierungsattribute
   HAVING G               Gruppierungsbedingung
   ORDER BY H             Sortierordnung
```

Für die Auswertungsreihenfolge gilt: `FROM`-Klausel vor `WHERE`-Klausel vor `GROUP`-Klausel vor `HAVING`-Klausel vor `ORDER`-Klausel vor `SELECT`-Klausel.

3.4 Anfragen mit Mengenoperatoren

`UNION`, `INTERSECT` und `EXCEPT`.

- Die beteiligten Tabellen müssen zueinander *kompatible* Spaltentypen haben.
- Die Resultatspalte bekommt dann jeweils den allgemeineren Typ.

Welche Länder sind Teil von Europa und Asien?

```
SELECT LCode FROM Lage
   WHERE Kontinent = 'Europe'
INTERSECT
SELECT LCode FROM Lage
   WHERE Kontinent = 'Asia'
```

Welche Landcodes treten in der Relation Land oder der Relation Lage auf?

```
SELECT LCode, 'Stadt' AS Kategorie FROM Stadt
UNION
SELECT LCode, 'Lage' AS Kategorie FROM Lage
```

Welche Landcodes treten in der Relation Land und nicht in der Relation Lage auf?

```
SELECT LCode FROM Land
EXCEPT
SELECT LCode FROM Lage
```

Hinweis: Oracle kennt nicht EXCEPT, aber dafür MINUS!

Duplikate

- Duplikate werden berücksichtigt, sofern die Varianten `UNION ALL`, `INTERSECT ALL`, `EXCEPT ALL` verwendet werden. Anderenfalls wird standardmäßig `DISTINCT` angenommen.
- Im Falle einer Verwendung von `ALL` verhalten sich die Operatoren wie folgt. Hat der erste Operand n Duplikate einer Zeile und der zweite Operand m, wobei $0 \leq n, m$, dann hat das Ergebnis bei `UNION` $n + m$, bei `INTERSECT` $min(n, m)$, und bei `EXCEPT` $max(n - m, 0)$ Duplikate dieses Tupels.

3.5 Geschachtelte Anfragen

Eine Anfrage heißt *geschachtelt*, wenn sie in der `SELECT-`, `FROM-`, oder `WHERE-`, bzw. `HAVING`-Klausel selbst wieder eine SQL-Anfrage enthält.

Zum Testen des Ergebnisses einer Teilanfrage (engl. subquery) existieren die Operatoren: `IN`, `ANY`, `ALL`, `UNIQUE`, `EXISTS` und `NOT`.

Welche Länder befinden sich gemeinsam mit Russland auf einem Kontinent?

```
SELECT DISTINCT LCode FROM Lage
   WHERE Kontinent IN
      (SELECT Kontinent FROM Lage WHERE LCode = 'RU')
```

Welche Länder haben eine größere Fläche als mindestens ein anderes Land?

```
SELECT LName FROM Land
   WHERE Fläche > ANY
      (SELECT Fläche FROM Land)
```

Welche Länder haben eine größere Fläche als mindestens ein anderes europäisches Land?

```
SELECT LName FROM Land
   WHERE Fläche > ANY
      (SELECT Fläche FROM Land, Lage
         WHERE Land.LCode = Lage.LCode AND
            Lage.Kontinent = 'Europe')
```

Welche Länder haben eine größere Fläche als alle anderen Länder?

```
SELECT LName FROM Land L1
   WHERE Fläche > ALL
      (SELECT Fläche FROM Land L2
         WHERE L1.LCode <> L2.LCode)
```

Bei Verwendung von *Korrelationsvariablen, (- namen)* wird die Teilanfrage pro möglicher Wertekombination der Korrelationsvariablen ihrer übergeordneten Anfragen genau einmal ausgeführt.

Bei Verwendung von *Korrelationsvariablen* wird die Teilanfrage pro möglicher Wertekombination der Korrelationsvariablen ihrer übergeordneten Anfragen genau einmal ausgeführt.

Zu welchen Ländern ist genau ein Kontinent bekannt?

```
SELECT LName FROM Land L1
   WHERE UNIQUE
      (SELECT L2.Kontinent FROM Lage L2
         WHERE L1.LCode = L2.LCode)
```

Zu welchen Ländern ist genau ein Kontinent bekannt?

```
SELECT LName FROM Land L1
   WHERE 1 =
      (SELECT COUNT(*) FROM Lage L2
         WHERE L1.LCode = L2.LCode)
```

Division

Beschreiben Sie das Ergebnis der folgenden Anfrage:

```
SELECT DISTINCT LCode
   FROM Mitglied M
   WHERE NOT EXISTS
      ((SELECT Organisation FROM Mitglied
         WHERE LCode = 'A')
      EXCEPT
      (SELECT Organisation FROM Mitglied
         WHERE LCode = M.LCode))
```

Es werden die Länder berechnet, die Mitglied in denselben Organisationen wie Österreich sind.

Anfragen von dieser Struktur bezeichent man als *Division* in Analogie zur Division auf den ganzen Zahlen.

- Seien a, b ganze Zahlen. Dann ist $a \div b$ die größte Zahl q, so dass $q * b \leq a$.
- Seien A, B Mengen (Relationen). Dann ist $A \div B$ die größte Relation Q, so dass $Q \times B \subseteq A$.

Welche Länder sind Mitglied in denselben Organisationen wie Österreich?

```
SELECT DISTINCT LCode
   FROM Mitglied M
   WHERE NOT EXISTS
      ((SELECT Organisation FROM Mitglied
         WHERE LCode = 'A')
      EXCEPT
      (SELECT Organisation FROM Mitglied
         WHERE LCode = M.LCode))
```

```
(SELECT Organisation FROM Mitglied WHERE LCode = 'A')
```

{ EU }

```
(SELECT Organisation FROM Mitglied WHERE LCode = M.LCode)
```

M.LCode	Organisationen
A	{ EU }
D	{ EU, WEU }
ET	{ UN }
I	{ EU, NAM }
TR	{ UN, CERN }

$\Longrightarrow$ {A, D, I}

Division alternativ A.

Welche Länder sind Mitglied in denselben Organisationen wie Österreich?

```
SELECT DISTINCT LCode
   FROM Mitglied M1
   WHERE NOT EXISTS (
      SELECT M2.Organisation FROM Mitglied M2
      WHERE M2.LCode = 'A' AND NOT EXISTS (
         SELECT Organisation FROM Mitglied M3
         WHERE M3.LCode = M1.LCode AND
            M3.Organisation = M2.Organisation))
```

$A \setminus B = \{x \mid x \in A \wedge x \notin B\}$

Welche Länder sind Mitglied in denselben Organisationen wie Österreich?

```
SELECT DISTINCT LCode
   FROM Mitglied M1
   WHERE NOT EXISTS (
      SELECT M2.Organisation FROM Mitglied M2
      WHERE M2.LCode = 'A' AND NOT EXISTS (
         SELECT Organisation FROM Mitglied M3
         WHERE M3.LCode = M1.LCode AND
            M3.Organisation = M2.Organisation))
```

M1	M2	M3	NOT EXISTS innen	NOT EXISTS außen
A	A	A	false	true
D	A	D	false	true
ET	A	ET	true	false
I	A	I	false	true
TR	A	TR	true	false

Division alternativ B.

Welche Länder sind Mitglied in denselben Organisationen wie Österreich?

```
SELECT DISTINCT LCode
   FROM Mitglied M1
   WHERE NOT EXISTS (
      SELECT M2.Organisation FROM Mitglied M2, Mitglied M3
      WHERE M3.LCode = 'A' AND M2.LCode = M1.LCode AND
         M3.Organisation NOT IN (
         SELECT M4.Organisation FROM Mitglied M4
         WHERE M4.LCode = M2.LCode)
   )
```

`NOT IN` ist problematisch bei Nullwerten (s. Kapitel 3.**??**(3)).

Division zum Letzten.

Welche Länder sind Mitglied in denselben Organisationen wie Österreich?

```
SELECT DISTINCT M1.LCode
FROM Mitglied M1, Mitglied M2
WHERE M2.LCode = 'A' AND
   M1.Organisation = M2.Organisation
GROUP BY M1.LCode
HAVING COUNT(M1.Organisation) = (
   SELECT COUNT(M3.Organisation)
   FROM Mitglied M3 WHERE M3.LCode = 'A' )
```

Wie verhalten sich die unterschiedlichen Varianten bei einer Division durch eine leere Tabelle, bei Existenz von Duplikaten?

Gleichheit!

Welche Länder sind Mitglied in GENAU denselben Organisationen wie Österreich?

```
SELECT DISTINCT LCode FROM Mitglied M WHERE
   NOT EXISTS
   ((SELECT Organisation FROM Mitglied WHERE LCode = 'A')
      EXCEPT
   (SELECT Organisation FROM Mitglied WHERE LCode = M.LCode))
   AND NOT EXISTS
   ((SELECT Organisation FROM Mitglied WHERE LCode = M.LCode)
      EXCEPT
   (SELECT Organisation FROM Mitglied WHERE LCode = 'A'))
```

Mengen A, B sind gleich genau dann, wenn $A \subseteq B$ und $A \supseteq B$;

und $A \subseteq B$ genau dann, wenn $A - B = \emptyset$.

A

```
SELECT LCode FROM Lage
   WHERE Lage.Kontinent = 'Europe'
```

B

```
SELECT LCode FROM Mitglied
   WHERE Organisation = M.Organisation
```

3.6 empfohlene Lektüre

SEQUEL: A STRUCTURED ENGLISH QUERY LANGUAGE

by

Donald D. Chamberlin
Raymond F. Boyce

IBM Research Laboratory
San Jose, California

ABSTRACT: In this paper we present the data manipulation facility for a structured English query language (SEQUEL) which can be used for accessing data in an integrated relational data base. Without resorting to the concepts of bound variables and quantifiers SEQUEL identifies a set of simple operations on tabular structures, which can be shown to be of equivalent power to the first order predicate calculus. A SEQUEL user is presented with a consistent set of keyword English templates which reflect how people use tables to obtain information. Moreover, the SEQUEL user is able to compose these basic templates in a structured manner in order to form more complex queries. SEQUEL is intended as a data base sublanguage for both the professional programmer and the more infrequent data base user.

1

[1] In: Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control. Kann gegoogled werden.