

# Kapitel 4 Der SQL-Standard

## 4.1 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					
<u>MatrNr</u>	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.

Bestimme alle Provinzen, zu denen die Fläche bekannt ist.

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

- ▶ 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).

## Wahrheitswerte

AND	t	u	f	OR	t	u	f	NOT	
t	t	u	f	t	t	t	t	t	f
u	u	u	f	u	t	u	u	u	u
f	f	f	f	f	t	u	f	f	t

### Was liefern:

```
SELECT * FROM Provinz
  WHERE (Fläche > 0)
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.

## 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

Wie viele 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 AND L.LCode = S.LCode
```

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

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

### Dieselbe Anfrage mit RIGHT OUTER JOIN

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

LName	HStadt	Einwohner
France	Paris	2125
Germany	Berlin	3472
Italy	Rome	2546
null	null	198
null	null	277
null	null	1244
null	null	495

## Varianten des äußeren Verbunds

- ▶ 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.
- ▶ Ein `FULL OUTER JOIN` berechnet die Vereinigung des entsprechenden `LEFT OUTER` und `RIGHT OUTER JOIN`.

## 4.2 Aggregation und Gruppierung

### Aggregierungsfunktionen: COUNT, MIN, MAX, SUM und AVG

Wie viele 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(Flaeche),MIN(Flaeche),AVG(Flaeche)
FROM Land
```

COUNT(LCode)	MAX(Flaeche)	MIN(Flaeche)	AVG(Flaeche)
8	17075	41	2523,125

Wie viele Länder haben eine Mitgliedschaft bzgl. der EU?

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

AnzEU
-------

3
---

### Wie viele unterschiedliche Organisationen werden in Mitglied aufgeführt?

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

AnzMitglied
5

## Besonderheiten

- ▶ COUNT(\*) liefert die Anzahl Zeilen der Tabelle, die sich nach Auswerten der FROM- und WHERE-Klausel ergeben hat.
- ▶ Aggregierungsfunktionen ignorieren für ihre Berechnungen Nullwerte.
- ▶ Eine Ausnahme ist COUNT(\*); hier werden auch alle Zeilen, in denen alle Spalten null sind, mitgezählt.

- ▶ SELECT LCode, AVG(Einwohner) FROM Stadt

ist syntaktisch nicht zulässig, da ein Aggregierungsoperator eine Menge von Zeilen auf einen einzigen Wert reduziert. Zulässig wäre:

```
SELECT AVG(Einwohner) FROM Stadt
```



# Gruppierung

## GROUP BY

- ▶ 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 die gleichen 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
```

LCode	AVG(Einwohner)
D	1137,2
F	2125
I	2546

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

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

LCode	AVG(Einwohner)
D	1137,2

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

```
SELECT LCode, PName, AVG(Einwohner), COUNT(*) AS Anzahl FROM Stadt  
GROUP BY (LCode, PName)
```

LCode	PName	AVG(Einwohner)	Anzahl
D	Berlin	3472	1
D	Baden	237,5	2
D	Bavaria	1244	1
D	Franken	495	1
F	Ile de France	2125	1
I	Lazio	2546	1

Was ist die größte Einwohnerzahl der Städte in einer Provinz?

```
SELECT LCode, PName, MAX(Einwohner) FROM Stadt  
GROUP BY (LCode, PName)
```

LCode	PName	MAX(Einwohner)
D	Berlin	3472
D	Baden	277
D	Bavaria	1244
D	Franken	495
F	Ile de France	2125
I	Lazio	2546

# Auswertungsreihenfolge

## SFW-Ausdruck (erweitert)

SELECT $A_1, \dots, A_n$	Liste der Attribute
FROM $R_1, \dots, R_m$	Liste der Relationen
WHERE $F$	Bedingung
GROUP BY $B_1, \dots, B_k$	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.

## 4.3 Mengenoperatoren

### UNION, INTERSECT und EXCEPT (MINUS)

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

Welche Ländercodes treten in der Relation Stadt oder der Relation Lage auf?

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

LCode	Kategorie
D	Lage
D	Stadt
F	Lage
F	Stadt
TR	Lage
ET	Lage
RU	Lage
I	Stadt

Welche Länder sind Teil von Europa und Asien?

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

LCode
RU
TR

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

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

LCode
A
CH
I

Hinweis: Oracle verwendet das Schlüsselwort MINUS anstelle von EXCEPT!

## 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.

Hinweis: Oracle unterstützt nur UNION ALL!

## 4.4 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, EXISTS und NOT.

Welche Länder befinden sich im gleichen Kontinent wie Russland?

```
SELECT DISTINCT L2.LCode
  FROM Lage L1, Lage L2
 WHERE L1.Kontinent = L2.Kontinent AND L1.LCode = 'RU'
```

```
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)
```

LName
Austria
Egypt
France
Germany
Italy
Russia
Turkey

Welche Länder haben eine größere Fläche als alle anderen Länder?

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

FALSCH! Alle *anderen* Länder!

## Korrelationsvariablen

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

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)
```

LName
Russia

Zu welchen Ländern ist mindestens ein Kontinent bekannt? (Variante 1)

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

LName
Germany
Egypt
France
Russia
Turkey

Zu welchen Ländern ist mindestens ein Kontinent bekannt? (Variante 2)

```
SELECT LName FROM Land L1
WHERE 1 <=
  (SELECT COUNT(L2.Kontinent) 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 /*MINUS*/
    (SELECT Organisation FROM Mitglied
     WHERE LCode = M.LCode))
```

Es werden die Länder berechnet, die in denselben Organisationen wie Österreich vertreten sind (member, guest oder observer).

Anfragen von dieser Struktur implementieren die relationale *Division*.  
Zur Erinnerung:

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

$\pi[LCode, Organisation]Mitglied \div \pi[Organisation](\sigma[LCode = 'A']Mitglied)$

Welche Länder sind in denselben Organisationen wie Österreich vertreten?

```
SELECT DISTINCT LCode
  FROM Mitglied M
 WHERE NOT EXISTS
    ((SELECT Organisation FROM Mitglied
      WHERE LCode = 'A')
  MINUS
  (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	$\Rightarrow \{A, D, I\}$
A	{ EU }	
D	{ EU, WEU }	
ET	{ UN }	
I	{ EU, NAM }	
TR	{ UN, CERN }	

## Division (alternative Variante)

Welche Länder sind in denselben Organisationen wie Österreich vertreten?

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

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

# Gleichheit

Welche Länder sind in GENAU denselben Organisationen wie Österreich vertreten?

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

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

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

# 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

---

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