Datenbankabfragen

Unterabfragen (Teil 1)

Fachhochschule
Dortmund
University of Applied Sciences

Fragestellung

Wie können SQL-Anfragen verschachtelt werden?

Hauptabfrage

Lager-nummer	Standort	ANummer	Lager-bestand	Artikelname
12345	FAN	4813	0	

Unterabfrage

4813	Märchen von Beedle dem Barden	Rowling	12.90	broschiert

Gebundene Unterabfrage

Aufgabe: Liefere den Standort, Artikelnamen und Lagerbestand

1. Schritt – Ermittlung der Artikelnummern

```
SELECT   ANummer,

         Lagerbestand
FROM     Lager
WHERE    ANummer = 4813
```

2. Schritt - Artikelnamen zum Artikel 4813 ermitteln

```
SELECT Artikelname
FROM   Artikel
WHERE  Artikelnummer = 4813
```

Gebundene Unterabfrage

Gebundene Unterabfrage (Subquery):
- Schachtelung der Anfragen mit Bezug auf äußere Variable
- Eindeutigkeit der Variablenbezüge wird durch Aliasnamen hergestellt

3. Schritt: Zusammenführen der Teilschritte 1 & 2

```
SELECT    ANummer,
          (SELECT Artikelname
          FROM    Artikel
          WHERE   Artikelnummer = 4813   ℓ.ANummer
          ),
          Lagerbestand
FROM      Lager  ℓ
```

Das Attribut ℓ.ANummer ist über den Alias ℓ eine gebundene Variable. Die innere Abfrage referenziert einen Wert der äußeren Abfrage des jeweils bearbeiteten Tupels. Die Abarbeitung der SQL-Abfrage erfolgt von Außen nach innen.

Ungebundene Unterabfrage

Ungebundene Unterabfrage: Schachtelung der Anfragen ohne Bezug auf eine äußere Variable.

1. Äußere Abfrage - Liste mit Abweichung vom mittleren Lagerbestand

```
SELECT Standort, ANummer,
          Lagerbestand - ‹mittlerer Lagerbestand›
FROM     Lager
WHERE  ANummer IS NOT NULL
```

2. Innere Abfrage: Ermittlung des mittleren Lagerbestandes

```
SELECT AVG(Lagerbestand)
FROM     Lager
```

3. Zusammengesetzt Abfrage

```
SELECT Standort, ANummer,
          Lagerbestand - ( SELECT AVG(Lagerbestand)
                             FROM     Lager)
FROM     Lager
WHERE  ANummer IS NOT NULL
```

Zusammenfassung

Eine gebundene Unterabfrage referenziert Attributwerte der äußeren Abfrage. Sie wird von Außen nach Innen abgearbeitet.

```
SELECT    ANummer,
          (SELECT Artikelname
          FROM    Artikel
          WHERE   Artikelnummer = 4813   ANummer
          ),
          Lagerbestand
FROM      Lager
```

Eine ungebundene Unterabfrage besitzt keinen Bezug zur äußeren Abfrage. Sie wird von Innen nach Außen abgearbeitet.

```
SELECT Standort, ANummer,
          Lagerbestand - ( SELECT AVG(Lagerbestand)
                           FROM    Lager)
FROM     Lager
WHERE  ANummer IS NOT NULL
```

Unterabfragen – Verwendung und Ergebnismengen

Die Unterabfrage liefert

… einen einzelnen Wert

- **SELECT**-Clause
  - als Spaltenangabe
- **WHERE**-Clause
  - einer SELECT-Abfrage
  - einer DELETE-Anweisung
  - einer UPDATE-Anweisung
- **SET**-Clause
  - einer UPDATE-Anweisung

… eine Menge von Tupeln

- **FROM-Clause**
- **WHERE**-Clause
  - einer SELECT-Abfrage
  - einer DELETE-Anweisung
  - einer UPDATE-Anweisung

Teil 2

Datenbankabfragen

Unterabfragen (Teil 2)

Fachhochschule
Dortmund
University of Applied Sciences

Fragestellung

Finde die Artikel, die (k)einen Lagerplatz besitzen …

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken
4899	Harry Potter Bd. 20

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	NULL	NULL

… mit einer Unterabfrage.

SQL-Abfragen

1. Artikel im Lager

```
SELECT ANummer
FROM   Lager
```

2. Alle Artikel

```
SELECT Artikelnummer, Artikelname
FROM   Artikel
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken
4899	Harry Potter Bd. 20

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	NULL	NULL

SQL-Abfragen

Finde die Artikel, die einen Lagerplatz besitzen

```
SELECT Artikelnummer, Artikelname
FROM   Artikel
WHERE Artikelnummer IN (SELECT ANummer FROM Lager)
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✓
4899	Harry Potter Bd. 20 ✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	NULL	NULL

SQL-Abfragen

Finde die Artikel, die keinen Lagerplatz besitzen

```
SELECT Artikelname, Autor, Ausgabe
FROM   Artikel
WHERE Artikelnummer NOT IN        (4812, NULL)
```

	NICHT
W	F
F	W
NULL	NULL

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✗
4899	Harry Potter Bd. 20 ✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	**NULL**	NULL

Der IN-Operator

Finde die Artikel, die keinen Lagerplatz besitzen

```
SELECT Artikelname, Autor, Ausgabe
FROM   Artikel
WHERE Artikelnummer NOT IN (SELECT ANummer FROM Lager
                            WHERE ANummer IS NOT NULL)
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✗
4899	Harry Potter Bd. 20 ✓

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	NULL	NULL

Der IN-Operator

- Die Vergleichsoperator IN prüft, ob ein Wert in einer Menge enthalten ist. Dafür wird der = Operator zum Wertevergleich genutzt. Ein Vergleich mit NULL wird durch den = Operator nicht durchgeführt und die entsprechenden Tupel übersprungen.

```
SELECT ANummer
FROM   Lager
WHERE  Artikelnummer IN (SELECT ANummer FROM Lager)
```

- Bei NULL-Werten in der Vergleichsmenge der Unterabfrage werden keine Tupel zurück geliefert. Bei der Verwendung von NOT IN müssen NULL-Werte aus der Ergebnismenge der Unterabfrage entfernt werden.

```
SELECT Artikelname, Autor, Ausgabe
FROM   Artikel
WHERE Artikelnummer NOT IN (SELECT ANummer FROM Lager
                            WHERE ANummer IS NOT NULL)
```

Der IN-Operator

Finde die Artikel, die keinen Lagerplatz besitzen

Verbundoperator

```
SELECT DISTINCT Artikelname, Autor, Ausgabe, ANummer
FROM   Artikel a  LEFT JOIN  Lager l
          ON a.Artikelnummer = l.ANummer
WHERE ANummer IS  NULL
```

IN-Operator

```
SELECT Artikelname, Autor, Ausgabe
FROM   Artikel
WHERE Artikelnummer NOT IN (SELECT ANummer FROM Lager
                            WHERE ANummer IS NOT NULL)
```

Mengen-Quantor EXISTS

Teil 3

Datenbankabfragen

Mengen-Quantoren Exist, All und Any

Fachhochschule
Dortmund
University of Applied Sciences

Fragestellung

Wie können Bedingungen auf die Ergebnismenge einer Unterabfrage formuliert werden?

EXISTS ALL ANY

Fragestellung

Finde die Artikel, für die (k)ein Lagerplatz existiert …

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken
4820	Datenbank-Skript
4899	Harry Potter Bd. 20

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

… mit einer Unterabfrage.

Der Mengen-Quantor **Exists (‹menge›)** prüft, ob eine Menge nicht-leer ist.

Finde die Artikel, für die ein Lagerplatz existiert …

```
SELECT Artikelnummer FROM  Artikel a
WHERE        EXISTS (SELECT Lagernummer FROM Lager
                     WHERE ANummer= a.Artikelnummer)
```

Artikel

Artikel-nummer	Artikelname	
4812	Datenbanken	✓
4820	Datenbank-Skript	✓
4899	Harry Potter Bd. 20	✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

Der Mengen-Quantor **Exists (‹menge›)** prüft, ob eine Menge nicht-leer ist.

Finde die Artikel, für die **kein** Lagerplatz existiert …

```
SELECT Artikelnummer FROM  Artikel a
WHERE NOT EXISTS (SELECT Lagernummer FROM Lager
                  WHERE ANummer= a.Artikelnummer)
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✗
4820	Datenbank-Skript ✗
4899	Harry Potter Bd. 20 ✓

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

Fragestellung

Finde die Artikel, für die an allen Lagerplätzen
ein Lagerbestand von mindestens 4 vorhanden ist …

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✓
4820	Datenbank-Skript ✗
4899	Harry Potter Bd. 20 ✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

… mit einer Unterabfrage.

ALL

ALL liefert true, wenn die Bedingung von **allen** Tupeln der Menge erfüllt wird. Die Bedingung wird erfüllt, wenn die Ergebnismenge **leer** ist.

Finde die Artikel, für die an allen Lagerplätzen ein Lagerbestand von mindestens 4 vorhanden ist …

```
SELECT Artikelnummer, Artikelname
FROM Artikel a
WHERE 4 <= ALL(SELECT Lagerbestand FROM Lager
               WHERE  ANummer=a.Artikelnummer)
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✓
4820	Datenbank-Skript ✗
4899	Harry Potter Bd. 20

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

Finde die Artikel, für die an allen Lagerplätzen
ein Lagerbestand von mindestens 4 vorhanden ist …

```
SELECT Artikelnummer, Artikelname
FROM Artikel a
WHERE 4 <= ALL(SELECT Lagerbestand FROM Lager
                WHERE  ANummer=a.Artikelnummer)
AND      EXISTS (SELECT Lagerbestand FROM Lager
                WHERE  ANummer=a.Artikelnummer)
```

Artikel

Artikel-nummer	Artikelname	
4812	Datenbanken	✓
4820	Datenbank-Skript	✗
4899	Harry Potter Bd. 20	✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

Fragestellung

Finde die Artikel, für die an (mindestens) einem Lagerplätzen ein Lagerbestand von mindestens 4 vorhanden ist …

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken
4820	Datenbank-Skript
4899	Harry Potter Bd. 20

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

… mit einer Unterabfrage.

ANY

ANY liefert true, wenn die Bedingung von **ein** Tupel der Menge erfüllt wird. Die Bedingung wird **nicht** erfüllt, wenn die Ergebnismenge **leer** ist.

Finde die Artikel, für die an mindestens einem Lagerplatz ein Lagerbestand von mindestens 4 vorhanden ist …

```
SELECT Artikelnummer, Artikelname
FROM Artikel a
WHERE 4 <= ANY(SELECT Lagerbestand FROM Lager
                    WHERE  ANummer=a.Artikelnummer)
```

Artikel

Artikel-nummer	Artikelname
4812	Datenbanken ✓
4820	Datenbank-Skript ✓
4899	Harry Potter Bd. 20 ✗

Lager

Lager-nummer	ANummer	Lager-bestand
27135	4812	18
27136	4820	3
27137	4820	5

Vorsicht Falle!

Der Vergleich mit einer leeren Menge liefert bei ALL true und bei ANY false.
ALL und ANY liefern daher unterschiedliche Ergebnisse bei leeren Unterabfragen.

ANY

```
SELECT * from Artikel
WHERE 0 < ANY(SELECT Preis from Artikel where Preis <0);
```

iptausgabe | Abfrageergebnis

SQL | Alle Zeilen abgerufen:0 in 0,003 Sekunden

ARTIKELN...	ARTIKELN...	AUTOR	PREIS	AUSGABE	KATEGOR...

ALL

```
SELECT * from Artikel
WHERE 0 < ALL(SELECT Preis from Artikel where Preis <0);
```

ausgabe | Abfrageergebnis

SQL | Alle Zeilen abgerufen:9 in 0,005 Sekunden

ARTIKELNUMMER	ARTIKELNAME	AUTOR	PREIS	AUSGABE	KATEG
4812	Datenbanksysteme	Elmasri	29,95	gebunden	FBU
4811	Datenbanksysteme	Kemper	28,9	broschiert	DB
4813	Märchen von Beedle dem Barden	Rowling	12,9	broschiert	FAN

ALL liefert alle Tupel der Tabelle Artikel

Hintergrund: Query-Optimierer

Durch den Query-Optimierer werden der ALL- und der ANY-Operator intern umgeformt und auf eine Abfrage mit dem Exists-Operator zurückgeführt.

ANY

0‹ANY (SELECT Preis FROM Artikel WHERE Preis ‹0)

⇩

EXISTS (SELECT Preis FROM Artikel WHERE Preis ‹0 AND 0‹Preis)

leere Menge

⇨ ✗

Hintergrund: Query-Optimierer

Durch den Query-Optimierer werden der ALL- und der ANY-Operator intern umgeformt und auf eine Abfrage mit dem Exists-Operator zurückgeführt.

ALL

0‹ALL (SELECT Preis FROM Artikel WHERE Preis ‹0)

⇩

NOT (0 ›= ANY (SELECT Preis FROM Artikel WHERE Preis ‹0))

⇩

NOT EXISTS (SELECT Preis FROM Artikel WHERE Preis ‹0
AND 0›=Preis)

⇨ ✓

leere Menge

Zusammenfassung

Die Mengen-Quantoren EXISTS, ALL und ANY werden verwendet, um Bedingungen auf einer Menge, z.B. der Ergebnismenge einer Unterabfrage, zu formulieren.

Quantor		Erläuterung
EXISTS	1	Liefert True, wenn die Ergebnismenge der Unterabfrage nicht leer ist, also Tupel vorhanden sind (existieren).
ALL	2	Liefert True, wenn die Bedingung für **alle** Tupel der Unterabfrage erfüllt ist.
ANY	3	Liefert True, wenn die Bedingung für **mindestens ein** Tupel der Unterabfrage erfüllt ist.

Der Vergleich mit einer leeren Menge liefert bei ALL true und bei ANY false. ALL und ANY liefern daher unterschiedliche Ergebnisse, wenn die Unterabfrage eine leere Ergebnismenge zurückliefert.

Alternativen

Finde alle Artikel, mit mindestens 4 Exemplaren an mindestens einem Lagerplatz

- Unterabfrage

```
SELECT Artikelnummer, Artikelname
FROM  Artikel a
WHERE 4 <= ANY (SELECT Lagerbestand  FROM Lager
                WHERE ANummer=a.Artikelnummer)
```

- Verbundoperation

```
SELECT DISTINCT Artikelnummer, Artikelname
FROM  Artikel a JOIN Lager l ON a.Artikelnummer=l.ANummer
WHERE Lagerbestand >= 4
```