



Datenbanken 1

Datenbankabfragen (Teil 3)

Inhaltsübersicht

Prof. Dr. I. M. Saatz

Datenbanken 1

Fachbereich Informatik

2

1	Verbundoperation oder Unterabfrage?	2
2	Mengenoperationen und Unterabfragen	10
3	Mengenoperationen und Fallunterscheidungen	17
4	Pagination	21
5	Rekursive Abfragen	26

Welchen Lösungsansatz wählen?

Verschiedene SQL-Anfragen können semantisch äquivalent sein. Im SQLTuning wird durch geschickte Umformulierung der SQL-Anfrage eine Performanzsteigerung erreicht.

Liste die Nachnamen der Kunden mit Artikel im Warenkorb



```
SELECT DISTINCT Nachname  
FROM Kunde  
Natural Join Warenkorb
```



```
SELECT DISTINCT Nachname  
FROM Kunde  
WHERE Kundennummer IN  
(SELECT Kundennummer  
FROM Warenkorb)
```



Beispiel

Finde alle Artikel, die **keinen** Lagerplatz zugewiesen bekommen haben.

1. Verstehen der Aufgabenstellung → Nachfragen

2. Analyse der Aufgabenstellung

- Welche Tabellen werden benötigt?
- Welche Attribute werden projiziert?
- Welche Attribute werden selektiert?

3. Lösung strukturieren

4. Lösung formulieren

<https://vote.fh.do/ABST>



Join oder Subquery?

„Finde alle Artikel, die **keinen** Lagerplatz zugewiesen bekommen haben“

- Subquery mit IN-Operator

```
SELECT Artikelnummer FROM Artikel
WHERE Artikelnummer NOT IN
    (SELECT ANummer FROM Lager WHERE ANummer IS NOT NULL)
```

- Subquery mit Exists-Operator

```
SELECT Artikelname, Autor, Ausgabe FROM Artikel a
WHERE NOT EXISTS (SELECT ANummer From Lager
                    WHERE ANummer=a.Artikelnummer)
```

- Mit Left-Join

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

Beispiel

Finde alle Artikel, zu denen an (mind.) **einem** Standort **mehr als 2** Exemplare vorhanden sind

1. Verstehen der Aufgabenstellung

2. Analyse der Aufgabenstellung

- Welche Tabellen werden benötigt?
- Welche Attribute werden projiziert?
- Welche Attribute werden selektiert?

3. Lösung strukturieren

4. Lösung formulieren

<https://vote.fh.do/ABST>



Konzeption einer Anfrage

Liste alle volljährige Kunden
aufsteigend geordnet nach Alter
und sortiert nach Namen

Kunden- nummer	Anrede	Nach- name	Geburts- datum
2310	Frau	Meitner	17.11.1878
7562	Herr	Einstein	14.03.1879
8365	Frau	Curie	07.11.1867
8632	NULL	FB Informatik	NULL

Was wird gesucht?

In welchen Tabellen?

Auswahlbedingungen?

Sortierung?

Mengenoperationen?



Inhaltsübersicht

Prof. Dr. I. M. Saatz

Datenbanken 1

Fachbereich Informatik

10

1	Verbundoperation oder Unterabfrage?	2
2	Mengenoperationen und Unterabfragen	10
3	Mengenoperationen und Fallunterscheidungen	17
4	Pagination	21
5	Rekursive Abfragen	26

Die Vereinigung der Ergebnismengen zweier SQL-Anfragen:

- UNION - doppelte Tupel werden entfernt
- UNION **ALL** - doppelte Tupel werden **nicht** entfernt

SELECT Artikelname, Preis, Ausgabe FROM Artikel R WHERE ...

SQL

UNION ALL

SELECT Artikelname, Preis, Ausgabe FROM Artikel S WHERE ...

A

Artikel-name	Preis	Ausgabe
Datenbanken	20	gebunden
Märchen	5	broschiert

T

Artikel-name	Preis	Ausgabe
Datenbanken	20	gebunden
Märchen	5	broschiert
Skript	10	geheftet
Märchen	5	broschiert

B

Artikel-name	Preis	Ausgabe
Skript	10	geheftet
Märchen	5	broschiert

Duplikat wird bei UNION ALL nicht entfernt

Zusammenfassen von Abfrageergebnissen



Welches ist der
günstigste und
der teuerste
Artikel?

<https://vote.fh.do/ABST>



ARTIKELNUMMER	ARTIKELNAME	AUTOR	PREIS	AUSGABE	KATEGORIEKRZ
1	4812 Datenbanksysteme	Elmasri	29, 95	gebunden	FBU
2	4811 Datenbanksysteme	Kemper	28, 9	broschiert	DB
3	4813 Märchen von Beedle dem Barden	Rowling	12, 9	broschiert	FAN
4	4814 Anatomie-interaktiv	Schattauer	19, 95	CD-Rom	MED
5	4815 Anatomie	Sobotta	69, 95	gebunden	MED
6	4816 Anatomie-Atlas	Smith	24, 95	gebunden	MED
7	4820 Datenbank-Skript	FH	5, 95	geheftet	SQL

Zusammenfassen von Anfrageergebnissen



Welches ist der
günstigste und
der teuerste
Artikel?

SELECT Artikelname, Least(Preis), Greatest(Preis)
FROM Artikel



Semantisch nicht korrekt

Least (1, 2, 27, -1) = -1

Greatest (1, 2, 27, -1) = 27

Least (29,95) = 29,95

ARTIKELNAME	LEAST(PREIS)	GREATEST(PREIS)
1 Datenbanksysteme	29,95	29,95
2 Datenbanksysteme	28,9	28,9
3 Märchen von Beedle dem Barden	12,9	12,9
4 Anatomie-interaktiv	19,95	19,95
5 Anatomie	69,95	69,95
6 Anatomie-Atlas	24,95	24,95
7 Datenbank-Skript	5,95	5,95
8 Harry Potter Band 20	(null)	(null)

Zusammenfassen von Anfrageergebnissen



Welches ist der
günstigste und
der teuerste
Artikel?

```
SELECT Artikelname, Min(Preis), Max(Preis)  
FROM Artikel;
```



Fehlermeldung!

```
SELECT Artikelname, Min(Preis), Max(Preis)  
FROM Artikel  
WHERE Preis=Min(Preis) OR Preis=Max(Preis)  
GROUP by Artikelname
```



Fehlermeldung!

```
SELECT Artikelname, Min(Preis), Max(Preis)  
FROM Artikel  
GROUP by Artikelname, Preis  
HAVING Preis=Min(Preis) OR Preis=Max(Preis)
```



Semantisch nicht korrekt!

Zusammenfassen von Anfrageergebnissen



Welches ist der
günstigste und
der teuerste
Artikel?

Idee: Lösung durch mehrere einfache
Anfragen, die mit Union zusammen-
gefasst werden.

ARTIKELNAME	PREIS
1 Anatomie	69,95
2 Datenbank-Skript	5,95

```
SELECT Artikelname, Min(Preis) Preis  
FROM Artikel  
GROUP BY Artikelname  
FETCH FIRST ROW ONLY  
UNION  
SELECT Artikelname, Max(Preis) Preis  
FROM Artikel  
GROUP BY Artikelname  
FETCH FIRST ROW ONLY
```

X
Funktioniert nicht!

Weshalb?

```
SELECT Artikelname, Preis  
FROM (SELECT * FROM Artikel  
      ORDER BY Preis ASC)  
WHERE ROWNUM =1  
UNION  
SELECT Artikelname, Preis  
FROM (SELECT * FROM Artikel  
      WHERE Preis IS NOT NULL  
      ORDER BY Preis DESC)  
WHERE ROWNUM =1
```



Zusammenfassen von Anfrageergebnissen



Welches ist der
günstigste und
der teuerste
Artikel?

Idee: Lösung durch mehrere einfache
Anfragen, die mit Union zusammen-
gefasst werden.

ARTIKELNAME	PREIS
1 Anatomie	69,95
2 Datenbank-Skript	5,95

```
SELECT Artikelname, Preis  
FROM Artikel  
WHERE Preis = (Select Min(Preis)  
                FROM Artikel)  
UNION  
SELECT Artikelname, Preis  
FROM Artikel  
WHERE Preis = (Select Max(Preis)  
                FROM Artikel)
```



```
SELECT Artikelname, Preis  
FROM (SELECT * FROM Artikel  
      ORDER BY Preis ASC)  
WHERE ROWNUM =1  
UNION  
SELECT Artikelname, Preis  
FROM (SELECT * FROM Artikel  
      WHERE Preis IS NOT NULL  
      ORDER BY Preis DESC)  
WHERE ROWNUM =1
```



Inhaltsübersicht

Prof. Dr. I. M. Saatz

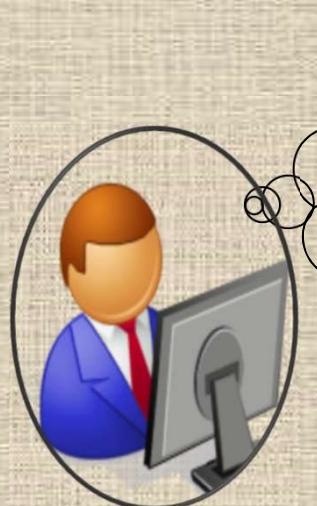
Datenbanken 1

Fachbereich Informatik

17

1	Verbundoperation oder Unterabfrage?	2
2	Mengenoperationen und Unterabfragen	10
3	Mengenoperationen und Fallunterscheidungen	17
4	Pagination	21
5	Rekursive Abfragen	26

Fallunterscheidungen treffen



Artikel mit Standort und Stockwerk ausgeben

Standort	Stockwerk
INF	2. OG
MED	1. OG
FAN	EG

	ANUMMER	STANDORT	STOCKWERK
1	4811	INF	2. OG
2	4812	INF	2. OG
3	4813	FAN	EG
4	4814	MED	1. OG
5	4815	MED	1. OG

Vorsicht Falle: Spaghetti-Abfrage

SELECT LehrangebotImStundenplan.Kuerzel, LehrangebotImStundenplan.Modulnummer, LehrangebotImStundenplan.Hinweise, LehrangebotImStundenplan.Kennung & LehrangebotImStundenplan.Modulnummer AS Kennung, LehrangebotImStundenplan.Dozent as Dozent, LehrangebotImStundenplan.Veranstaltung as Veranstaltung
FROM LehrangebotImStundenplan
UNION SELECT LehrangebotImStundenplan.Kuerzel, LehrangebotImStundenplan.Modulnummer, LehrangebotImStundenplan.Hinweise, LehrangebotImStundenplan.Kennung & LehrangebotImStundenplan.Modulnummer AS Kennung, LehrangebotImStundenplan.Dozent as Dozent, LehrangebotImStundenplan.Veranstaltung as Veranstaltung
FROM LehrangebotImStundenplan
Where LehrangebotImStundenplan.AltModulnummer is not null
UNION SELECT LehrangebotWahlpflicht.Kuerzel, LehrangebotWahlpflicht.Modulnummer AS Kennung, LehrangebotWahlpflicht.Hinweise, LehrangebotWahlpflicht.Modulnummer AS Kennung, LehrangebotWahlpflicht.Dozent as Dozent, LehrangebotWahlpflicht.Veranstaltung as Veranstaltung
FROM LehrangebotWahlpflicht
UNION SELECT LehrangebotWahlpflicht.Kuerzel, LehrangebotWahlpflicht.Modulnummer AS Kennung, LehrangebotWahlpflicht.Hinweise, LehrangebotWahlpflicht.Modulnummer AS Kennung & LehrangebotWahlpflicht.Veranstaltung as Veranstaltung
FROM LehrangebotWahlpflicht
Where LehrangebotWahlpflicht.AltModulnummer is not null
UNION SELECT LehrangebotGeplant.Kuerzel, LehrangebotGeplant.Modulnummer AS Kennung, LehrangebotGeplant.Hinweise, LehrangebotGeplant.Modulnummer AS Kennung & LehrangebotGeplant.Veranstaltung as Veranstaltung
FROM LehrangebotGeplant
Where LehrangebotGeplant.AltModulnummer is not null
UNION SELECT LehrangebotGeplant.Kuerzel, LehrangebotGeplant.Modulnummer AS Kennung, LehrangebotGeplant.Hinweise, LehrangebotGeplant.Modulnummer AS Kennung & LehrangebotGeplant.Veranstaltung as Veranstaltung
FROM LehrangebotGeplant
Where LehrangebotGeplant.AltModulnummer is not null
UNION SELECT LehrangebotAltModule.Kuerzel, LehrangebotAltModule.Modulnummer AS Kennung, LehrangebotAltModule.Hinweise, LehrangebotAltModule.Modulnummer AS Kennung & LehrangebotAltModule.Veranstaltung as Veranstaltung
FROM LehrangebotAltModule
Where LehrangebotAltModule.AltModulnummer is not null
UNION SELECT LehrangebotAltModule.Kuerzel, LehrangebotAltModule.Modulnummer AS Kennung, LehrangebotAltModule.Hinweise, LehrangebotAltModule.Modulnummer AS Kennung & LehrangebotAltModule.Veranstaltung as Veranstaltung
FROM LehrangebotAltModule
Where LehrangebotAltModule.AltModulnummer is not null
UNION SELECT Distinct Studiengaenge FROM Studiengaenge
UNION SELECT Distinct Studiengaenge FROM Studiengaenge
= "Semesteranzahl" = "1", DLookup("Dozent, " " as Veranstaltung
FROM Studiengaenge;

**Schlechte
Performanz**



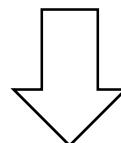
Master Informatik (MPO 2013)

Modul-Nr.	Themenbereiche / Module	Semester	LP	SS 2013
	Formale Strukturen	1.-3.	15	
	3 aus 6 zu wählen (insg. 15 LP)			
INPM 46865	Formale Sprachen und Compilerbau		5	s. Stundenplan Master INF
INPM 46866	Berechenbarkeit und Komplexitätstheorie		5	s. Stundenplan Master INF
INPM 46801	Angewandte Statistik		5	s. Stundenplan Master INF
INPM 46867	Angewandte Mathematik		5	geplant WS 2017/18
INPM 46839	Maschinelles Lernen (vorm. Neuroinformatik)		5	s. Stundenplan Master INF
INPM 46800	Mathematische Grundlagen der Verschlüsselungstechnik		5	geplant WS 2017/18
	Informatik	1.-3.	15	
	3 aus 5 zu wählen (insg. 15 LP)			
INPM 46859	Formale Methoden der Softwaretechnik		5	in Englisch, s. Stundenplan Master INF
INPM 46848	System- und Softwarequalitätssicherung		5	geplant WS 2017/18
INPM 46857	IT-Sicherheit		5	geplant WS 2017/18
INPM 46910	Requirements Engineering		5	in Englisch, geplant WS 2017/18
INPM 46833	IT-Netze		5	geplant WS 2017/18
	Managementqualifikation	1.-3.	10	
	2 aus 4 zu wählen (insg. 10 LP)			
INPM 46911	Fortgeschrittene BWL		5	s. Stundenplan Master INF
INPM 46858	Projektmanagement		5	geplant WS 2017/18
INPM 47723	Personalführung		5	s. Stundenplan Master INF
INPM 46877	Organisatorisch/rechtliche Aspekte von IT-Beschaffungen		5	s. Stundenplan Master INF

Fallunterscheidungen

Spaghetti-Anfrage sind nicht performant, da bei jeder UNION SELECT-Abfragen zusätzlich sortiert wird, um Duplikate zu entfernen. Besser ist es, UNION SELECT durch geschickte CASE WHEN-Abfragen umzuformulieren.

```
SELECT ANummer, '2. OG' as Stockwerk FROM Lager WHERE Standort='INF'  
UNION  
SELECT ANummer, '1. OG' as Stockwerk FROM Lager WHERE Standort='MED'  
UNION  
SELECT ANummer, 'EG' as Stockwerk FROM Lager WHERE Standort NOT IN ('INF', 'MED')
```



```
Select ANummer,  
CASE Standort  
    WHEN 'INF' THEN '2. OG'  
    WHEN 'MED' THEN '1. OG'  
    ELSE 'EG'  
END AS Stockwerk  
From Lager
```

Inhaltsübersicht

Prof. Dr. I. M. Saatz

Datenbanken 1

Fachbereich Informatik

21

1	Verbundoperation oder Unterabfrage?	2
2	Mengenoperationen und Unterabfragen	10
3	Mengenoperationen und Fallunterscheidungen	17
4	Pagination	21
5	Rekursive Abfragen	26

Bereichsanfragen (Pagination)



Die Pseudospalte ROWNUM kann genutzt werden, um einen Bereich abzufragen. Die ROWNUM gibt die Nummer des Tupels in der Lesereihenfolge aus der Tabelle an. Eine Sortierung erfolgt erst nach der Festlegung der ROWNUM.

```
SELECT Preis, ROWNUM
FROM Artikel
ORDER BY Preis;
```

	PREIS	ROWNUM
1	5,95	7
2	12,9	3
3	19,95	4
4	24,95	6
5	28,9	2
6	29,95	1
7	69,95	5
8	(null)	8

Bereichsanfrage
Auswahl Tupel 41 - 60:

```
SELECT *
FROM ( SELECT Artikelnummer, Artikelname
      FROM Artikel
      ORDER BY Artikelnummer
    )
WHERE ROWNUM >=41 AND ROWNUM <=60
```



Um einen Bereich abzufragen kann in **Oracle 12c** der Result-Offset-Klausel nach der Order-By Klausel verwendet werden.

Syntax der Result-Offset-Klausel

```
OFFSET { integer-literal | ? } {ROW | ROWS}  
FETCH { FIRST | NEXT } [integer-literal | ? ]  
{ROW | ROWS} ONLY
```

Auswahl Tupel 41 - 60:

```
SELECT Artikelnummer, Artikelname  
FROM Artikel  
ORDER BY Artikelnummer  
OFFSET 40 ROWS FETCH NEXT 20 ROWS ONLY
```

Beispiel



Was liefert die Abfrage?

Oracle 12c

```
SELECT Artikelname FROM Artikel  
ORDER BY Preis DESC  
FETCH FIRST ROW ONLY
```

<https://vote.fh.do/ABST>



ARTIKELNAME	PREIS
1 Datenbanksysteme	29,95
2 Anatomie	69,95
3 Harry Potter Band 20	(null)



Bei der aufsteigenden Sortierung folgen NULL-Werte nach den größten Werten.

Oracle 12c

```
SELECT Artikelname FROM Artikel  
ORDER BY Preis DESC  
FETCH FIRST ROW ONLY
```

Harry Potter Band 20

Oracle 11g

```
SELECT * FROM (SELECT Artikelname FROM Artikel  
ORDER BY Preis DESC)  
WHERE ROWNUM=1;
```

Harry Potter Band 20

Artikel gibt es nicht! ☹
Der Preis ist daher NULL!

Inhaltsübersicht

Prof. Dr. I. M. Saatz

Datenbanken 1

Fachbereich Informatik

26

1	Verbundoperation oder Unterabfrage?	2
2	Mengenoperationen und Unterabfragen	10
3	Mengenoperationen und Fallunterscheidungen	17
4	Pagination	21
5	Rekursive Abfragen	26

Strukturierung von Anfragen

Eine Unterabfrage in der FROM-Klausel stellt eine temporäre Benutzersicht (Inline View) dar. Diese ist nur für diese Anfrage verfügbar. Mit der WITH-Klausel kann die Unterabfrage zu Beginn der Abfrage deklariert werden.

Unstrukturierte Abfrage

```
SELECT Lagernummer
FROM Lager
WHERE Standort = 'INF'
AND Lagerbestand IS NULL AND ANummer IS NULL
```

Mit Unterabfrage
strukturiert:

```
SELECT frei.Lagernummer
FROM (SELECT * FROM Lager
      WHERE Lagerbestand IS NULL
      AND ANummer IS NULL) frei
WHERE Standort = 'INF'
```

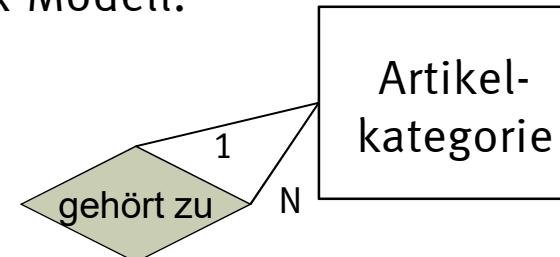
Mit WITH-Präfix:
(*Sub-Query Factoring*)

```
WITH frei AS (SELECT * FROM Lager
              WHERE Lagerbestand IS NULL
              AND ANummer IS NULL)
SELECT Lagernummer FROM frei
WHERE Standort = 'INF'
```

Rekursive Beziehungen (Adjacency List pattern)



ER-Modell:



Relationales Modell:

KATEGORIEKRZ	BEZEICHNUNG	OBERKATEGORIE
1 AK	Alle Kategorien	(null)
2 BUE	Bücher	AK
3 FBU	Fremdsprachige Bücher	AK
4 INF	Computer und Internet	BUE
5 DB	Datenbanken	INF
6 SQL	SQL	DB
7 FAN	Fantasy (Deutsch)	BUE
8 FAE	Fantasy (Fremdsprachig)	FBU
9 FIN	Computer and Internet (Fremdsprachig)	FBU

Unterkategorien abfragen - Unterabfragen



Problem:

Bei der Nutzung von Unterabfragen muss explizit die Tiefe der Rekursion festgelegt werden.

```
SELECT '1. Ebene', Bezeichnung
FROM Artikelkategorie
WHERE KategorieKrz='AK'
UNION
SELECT '2. Ebene', Bezeichnung
FROM Artikelkategorie
WHERE Oberkategorie ='AK'
UNION
SELECT '3. Ebene', Bezeichnung
FROM Artikelkategorie
WHERE Oberkategorie IN
```

```
(SELECT KategorieKrz
FROM Artikelkategorie
WHERE Oberkategorie ='AK')
```

'1. EBENE'	BEZEICHNUNG
1	1. Ebene Alle Kategorien
2	2. Ebene Bücher
3	2. Ebene Fremdsprachige Bücher
4	3. Ebene Computer and Internet (Fremdsprachig)
5	3. Ebene Computer und Internet
6	3. Ebene Fantasy (Deutsch)
7	3. Ebene Fantasy (Fremdsprachig)

Unterkategorien abfragen - Verbund



Problem:

Bei der Nutzung von Verbundoperationen muss explizit die Tiefe der Rekursion festgelegt werden.

Mit Verbundoperation

```
SELECT ak1.*, ak2.*, ak3.*  
FROM Artikelkategorie ak1  
    LEFT OUTER JOIN Artikelkategorie ak2  1. Ebene  
        ON ak1.KategorieKrz = ak2.Oberkategorie  
    LEFT OUTER JOIN Artikelkategorie ak3  2. Ebene  
        ON ak2.KategorieKrz = ak3.Oberkategorie  
WHERE ak1.KategorieKrz='AK';  
                                              3. Ebene
```

1. Ebene

2. Ebene

3. Ebene

BEZEICHNUNG	BEZEICHNUNG_1	BEZEICHNUNG_2
1 Alle Kategorien Bücher		Computer und Internet
2 Alle Kategorien Bücher		Fantasy (Deutsch)
3 Alle Kategorien Fremdsprachige Bücher	Fantasy (Fremdsprachig)	
4 Alle Kategorien Fremdsprachige Bücher	Computer and Internet (Fremdsprachig)	

Rekursive Abfragen – WITH-Rekursiv



Einige DBMS unterstützen hierarchische Abfragen.

$$2^n = \begin{cases} 1, & \text{für } n = 0 \quad \text{Induktionsanfang} \\ 2 \cdot 2^{n-1}, & \text{für } n > 0 \quad \text{Induktionsschritt} \end{cases}$$

```
WITH zweierpotenzen (n, potenz)
AS
(SELECT 0,1 FROM dual      Induktionsanfang
 UNION ALL
 SELECT n+1, 2*potenz      Induktionsschritt
 FROM zweierpotenzen
 WHERE n<9
)
SELECT * FROM zweierpotenzen
```

N	FACT
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512

ab Oracle 11g

Rekursive Abfragen – WITH-Rekursiv



Einige DBMS unterstützen hierarchische Abfragen.

KATEGORIEKrz	OBERKATEGORIE
1 AK	(null)
2 BUE	AK
3 FAN	BUE
4 INF	BUE
5 DB	INF
6 SQL	DB
7 FBU	AK
8 FAE	FBU
9 FIN	FBU

```
WITH ak(KategorieKrz, Oberkategorie) AS (
  -- Start-Kategorie.
  SELECT KategorieKrz,
    Oberkategorie
  FROM Artikelkategorie
  WHERE KategorieKrz = 'AK'
  UNION ALL
  -- Unterkategorie
  SELECT child.KategorieKrz, child.Oberkategorie
  FROM Artikelkategorie child, ak
  WHERE child.Oberkategorie = ak.KategorieKrz
)
SEARCH DEPTH FIRST BY KategorieKrz SET order1
SELECT KategorieKrz, Oberkategorie
FROM ak
ORDER BY order1;
```

ab Oracle 11g

Ober- und Unterkategorien abfragen

Mit der Oracle-spezifischen Erweiterung **CONNECT BY PRIOR** können rekursive hierarchische Abfragen durchgeführt werden.

Alle Unterkategorien

```
SELECT * FROM Artikelkategorie
START WITH Oberkategorie='BUE'
CONNECT BY PRIOR KategorieKrz=Oberkategorie
```

KATEGORIEKRZ	BEZEICHNUNG	OBERKATEGORIE
1 FAN	Fantasy (Deutsch)	BUE
2 INF	Computer und Internet	BUE
3 DB	Datenbanken	INF
4 SQL	SQL	DB

Alle Oberkategorien

```
SELECT * FROM Artikelkategorie
START WITH KategorieKrz='SQL'
CONNECT BY PRIOR Oberkategorie=KategorieKrz
```

KATEGORIEKRZ	BEZEICHNUNG	OBERKATEGORIE
1 SQL	SQL	DB
2 DB	Datenbanken	INF
3 INF	Computer und Internet	BUE
4 BUE	Bücher	AK
5 AK	Alle Kategorien	(null)

Unterkategorien abfragen – Oracle-Spezifisch

Mit der Oracle-spezifischen Erweiterung **CONNECT BY PRIOR** können rekursive hierarchische Abfragen durchgeführt werden.

```
SELECT Level, Oberkategorie, KategorieKrz,
       SYS_CONNECT_BY_PATH(KategorieKrz, '/') Pfad,
       CONNECT_BY_ISLEAF istBlatt,
       CONNECT_BY_ISCYCLE istZyklus,
       CONNECT_BY_ROOT KategorieKrz Wurzel
  FROM Artikelkategorie
 START WITH Oberkategorie IS NULL
 CONNECT BY NOCYCLE PRIOR KategorieKrz=Oberkategorie;
```

} Pseudospalten

LEVEL	OBERKATEGORIE	KATEGORIEKrz	PFAD	ISTBLATT	ISTZYKLUS	WURZEL
1	1 (null)	AK	/AK	0	0	AK
2	2 AK	BUE	/AK /BUE	0	0	AK
3	3 BUE	FAN	/AK /BUE/FAN	1	0	AK
4	3 BUE	INF	/AK /BUE/INF	0	0	AK
5	4 INF	DB	/AK /BUE/INF/DB	0	0	AK
6	5 DB	SQL	/AK /BUE/INF/DB /SQL	1	0	AK
7	2 AK	FBU	/AK /FBU	0	0	AK
8	3 FBU	FAE	/AK /FBU/FAE	1	0	AK
9	3 FBU	FIN	/AK /FBU/FIN	1	0	AK

Fragen zur Selbstreflexion

- 1) Wie können TOP-N und Bereichsanfragen formuliert werden?
- 2) Wie können rekursive Anfragen formuliert werden und welche Schwierigkeit tritt hierbei auf?
- 3) Wie kann eine komplexe Abfrage strukturiert werden?
- 4) Wie können Spaghetti-Abfragen vermieden werden?

- ILIAS → 09. Woche

- Inhalte

- Lernmodul und Praktikum Datenbankabfragen (Teil 3)
 - Wochentest Datenbankabfragen (Teil 3)
 - Reservierung eines Abnahmetermins für die Projektaufgabe
→ Terminbuchung in ILIAS (Ordner Projektaufgabe)

**Vielen Dank
für Ihre Aufmerksamkeit**