

Tabelle LND

LANDSCHL	BEZEICHN
BM	Bermudas
DK	
E	
F	
GB	
GCA	Guatemala
I	
NL	
PNG	Papua Neuguinea
USA	

Tabelle KND

KUNDNR	KURZNAME	PKTOFIBU	LAND	KRELIMIT	PRIVAT	ANLDAT
1	ABZOCKER, BERLIN	10001		1000000,00	N	04.09.2006
2	MADGIRLS, LONDON	10002	GB		N	04.09.2006
3	STUPIDO, ROMA	10003	I	2,50	N	04.09.2006
4	FRAUDEUR, PARIS	10004	F	3000,00	N	04.09.2006
5	BRITNEY SPEARS, LA	10005	USA	1,00	J	04.09.2006
6	BRATT PITT, NY	10006	USA	1,50	J	04.09.2006

Tabelle BRA

BRANCHE
Bad/WC
Elektro
Holz

Tabelle UST

USTSCHL
Ausland
EU
Inland

Tabelle SPR

SPRACHE
Englisch
Franzoes

Tabelle LIE

LIEFNR	KURZNAME	PKTOFIBU	LAND	KRELIMIT	ANLDAT	USTSCHL	BRANCHE	NAME1	NAME2	STRASSE	PLZORT	SPRACHE
1	WEBER, PADERBORN	70001		10000,00	04.09.2006	Inland		*		*	*	
2	MILLER, HAMILTON	70002	BM		04.09.2006	Ausland		*		*	*	Englisch
3	LAHO, BAGNEUX	70012	F	5000,00	04.09.2006	EU		*		*	*	Franzoes
4	FOSTERS, BIRMINGHAM	70004	GB	7500,00	04.09.2006	EU		*		*	*	Englisch

Tabelle LAA

LIEFNR	ADNR	ADRSUCH	NAME1	NAME2	STRASSE	PLZORT
1	1	BIELEFELD	Weber Maschinenbau GmbH		Brunnenweg 7	33604 Bielefeld
1	2	HERFORD	Weber Maschinenbau GmbH	Zentrallager	Mindener Str. 5	32051 Herford

Tabelle LAP

LIEFNR	ADNR	PARTNR	PARTSUCH	VORNAME	FAMNAME	TELEFON
1	1	1	KLEINE	Heinz	Kleine	0521/3029-12
1	1	2	KRAUSE	Margit	Krause	0521/3029-26
1	2	1	KLEINE	Werner	Kleine	05221/34187-56
1	2	2	LORENZ	Konrad	Lorenz	05221/34187-54

Tabelle MEH

MENGEINH
Kg
Lit
Std
Stk

Tabelle ART

ARTIKEL	BEZEICHN	HPTLIE	MEH	PREIS1	PRS2AB	PREIS2
10.1001	Gartenpumpe	1	Stk	39,99	01.06.2006	49,99
10.1016	Hochdruckreiniger	1	Stk	99,99		
25.3282	Holzlasur	2	Lit	4,98	01.04.2006	3,98
56.7954	Fliesenkleber	3	Kg	0,40		
80.0001	Montageleistungen		Std	49,95		

Tabelle AFS

ARTIKEL	SPRACHE	BEZEICHN
10.1001	Englisch	garden-pump
10.1001	Franzoes	Pompent de jardin
10.1016	Englisch	high-pressure-cleaner
10.1016	Franzoes	Nettoyeuse Forte pression

Tabelle LAN

ARTIKEL	LIEFNR	LIEARTNR
10.1001	1	A-482.191
10.1001	4	128300
10.1016	1	B-452.957
10.1016	4	123853

Tabelle LSP

ARTIKEL	LIEFNR	MENGEV	PREIS1	PRS2AB	PREIS2
10.1001	1	1,00	27,95	01.06.2006	32,50
10.1001	1	10,00	26,50	01.06.2006	31,00
10.1001	1	100,00	24,50	01.06.2006	29,00
10.1001	4	1,00	42,50	01.07.2006	42,00
10.1016	1	1,00	78,50		
10.1016	1	10,00	75,00		
10.1016	4	1,00	79,00	01.07.2006	80,00
10.1016	4	10,00	75,00	01.07.2006	76,00
10.1016	4	100,00	69,00	01.07.2006	69,90

Tabelle ZIM

ZIMMER
101
102
103
104
105
201
202
203
204
205
301
302
303
304
305

Tabelle GST

GASTNR	TITEL	VORNAME	FAMNAME	STRASSE	PLZ	ORT	TELEFON	TELEFAX
1		Klaus	Meier	Mergelweg 7	33100	Paderborn	05251/319201	
2	Dr.	Jutta	Krause	Zur Linde 38	33689	Bielefeld	0521/2123512	0521/2118242

Tabelle BEL

ZIMMER	DATUM	GASTNR
101	10.11.2006	1
101	11.11.2006	1
101	12.11.2006	1
101	13.11.2006	1
101	14.11.2006	1
101	15.11.2006	1
102	10.11.2006	2
102	11.11.2006	2
102	12.11.2006	2
102	14.11.2006	

Aggregatfunktionen

Funktion	Ergebnis
MIN(expression)	Minimalwert
MAX(expression)	Maximalwert
AVG(expression)	Durchschnittswert
SUM(expression)	Summe
COUNT(*)	Anzahl der Datensätze
COUNT(expression)	Anzahl der Datensätze, in denen expression nicht NULL ist

Die Ergebnismenge einer Sql-Abfrage, deren Spalten ausschließlich Aggregatfunktionen beinhalten, enthält immer genau einen Satz. Die Verwendung der Klausel ORDER BY ist deshalb nicht sinnvoll.

Als Funktionsparameter (expression) können Spaltennamen oder berechnete Ausdrücke eingesetzt werden.

Beispiele

```
SELECT MIN(PREIS1) AS MINIMUM FROM ART;
```

MINIMUM
0,40

Anmerkung: Das Umbenennen von Spalten mit Aggregatfunktionen ist in jedem Fall sinnvoll. Die Datenbankzugriffsklassen von Softwaresystemen erwarten generell Spaltennamen, die den Konventionen des jeweiligen Entwicklungssystems entsprechen.

```
SELECT MIN(PREIS1) AS MINIMUM,MAX(PREIS1) AS MAXIMUM FROM ART;
```

MINIMUM	MAXIMUM
0,40	99,99

Anmerkung: Eine Sql-Abfrage kann mehrere Spalten mit Aggregatfunktionen beinhalten.

```
SELECT MIN(PREIS1) AS MINIMUM,MAX(PREIS1) AS MAXIMUM
FROM ART WHERE ARTIKEL LIKE '10.%';
```

MINIMUM	MAXIMUM
39,99	99,99

Anmerkung: Eine Sql-Abfrage mit Aggregatfunktionen kann auch eine WHERE-Bedingung beinhalten.

Oracle

```
SELECT ROUND(AVG((PREIS1+PREIS2)/2),2) AS MITTELWT FROM ART;
```

Interbase

```
SELECT CAST(AVG(CAST((PREIS1+PREIS2)/2 AS DECIMAL(8,4))) AS DECIMAL(8,2)) AS
MITTELWT FROM ART;
```

MITTELWT
24,93

Anmerkung: Eine Aggregatfunktion kann auch einen berechneten Ausdruck als Parameter beinhalten. In Interbase wird durch die innere CAST-Funktion das Rechnen mit 4 Nachkommastellen erzwungen. Der berechnete Durchschnittswert wird dann in beiden Systemen auf 2 Nachkommastellen gerundet.

```
SELECT SUM(PREIS1) AS SUMME FROM ART;
```

SUMME
195,31

Anmerkung: Hier wird die Summe der Zeilenwerte für die Spalte PREIS1 der Tabelle ART ermittelt.

```
SELECT COUNT(*) AS ANZAHL FROM ART;
```

ANZAHL
5

Anmerkung: Die Funktion COUNT(*) liefert die Anzahl der Sätze in Tabelle ART zurück.

```
SELECT COUNT(*) AS ANZAHL FROM ART WHERE ARTIKEL > '80.0001';
```

ANZAHL

Anmerkung: Die Ergebnismenge enthält bei Aggregatfunktionen immer genau einen Satz. Dies gilt auch dann, wenn, wie in diesem Fall, kein Satz der Tabelle ART der WHERE-Bedingung entspricht.

```
SELECT COUNT(PRS2AB) AS ANZPRS2 FROM ART;
```

ANZPRS2
2

Anmerkung: Hier wird die Anzahl der Sätze in Tabelle ART ermittelt, in denen das Feld PRS2AB nicht NULL ist.

Übungsaufgaben

1. Ermitteln Sie die Anzahl der Artikel aus Tabelle ART, denen kein Hauptlieferant zugeordnet ist. Die Ausgabespalte soll mit dem Namen OHNELIEF bezeichnet werden. Die Abfrage darf keine WHERE-Klausel beinhalten.

Ausgabebeispiel

OHNELIEF
1

2. Ermitteln Sie das kleinste und das größte Anlagedatum aller ausländischen Lieferanten aus Tabelle LIE. Die Ausgabespalten sollen mit den Namen KLEINDAT und GROSSDAT gekennzeichnet werden.

Ausgabebeispiel (Hinweis: Ihr Ergebnis kann abweichende Datumsangaben beinhalten!)

KLEINDAT	GROSSDAT
04.09.2006	04.09.2006

3. Ermitteln Sie die Summe und das durchschnittliche Kreditlimit aller Kunden, denen ein Kreditlimit zugewiesen wurde. Die Ausgabespalten sollen mit den Namen ADDITION und DSCHNITT benannt werden.

Ausgabebeispiel

ADDITION	DSCHNITT
1003005,00	200601,00

Lösungen

1. SELECT COUNT(*)-COUNT(HPTLIE) AS OHNELIEF FROM ART;
2. SELECT MIN(ANLDAT) AS KLEINDAT,MAX(ANLDAT) AS GROSSDAT FROM LIE
WHERE LAND IS NOT NULL;
3. SELECT SUM(KRELIMIT) AS ADDITION,AVG(KRELIMIT) AS DSCHNITT FROM KND
WHERE KRELIMIT>0;

Outer Join

Tabelle TB1 (Vertreter)

```
CREATE TABLE TB1 (VERTRE NUMBER(2,0) NOT NULL,
                  KURZNAME VARCHAR2(20) NOT NULL,
                  PRIMARY KEY (VERTRE));
```

VERTRE	KURZNAME
1	Meier
2	Schulze
3	Weber
4	Schmidt

Tabelle TB2 (Kunden)

```
CREATE TABLE TB2 (KUNDNR NUMBER(6,0) NOT NULL,
                  KURZNAME VARCHAR2(20) NOT NULL,
                  VERTRE NUMBER(2,0),
                  FOREIGN KEY (VERTRE) REFERENCES TB1,
                  PRIMARY KEY(KUNDNR));
```

KUNDNR	KURZNAME	VERTRE
1	Kleine	
2	Schuster	1
3	Winkelmann	
4	Fischer	2
5	Vogel	
6	Genzmann	3

Aufgabe

Es sollen alle Kunden mit den evtl. zugeordneten Vertretern angezeigt werden. Die Ausgabe soll die Spalten TB2.KUNDNR, TB2.KURZNAME, TB2.VERTRE UND TB1.KURZNAME umfassen sowie aufsteigend nach Kundennummer (TB2.KUNDNR) sortiert sein. Die Spalte TB2.KURZNAME soll mit der Bezeichnung KUNDNAME, die Spalte TB1.KURZNAME soll mit der Bezeichnung VERTNAME ausgegeben werden.

```
SELECT TB2.KUNDNR, TB2.KURZNAME AS KUNDNAME, TB2.VERTRE, TB1.KURZNAME AS VERTNAME
FROM TB2, TB1 WHERE TB1.VERTRE=TB2.VERTRE ORDER BY TB2.KUNDNR;
```

Ergebnis

KUNDNR	KUNDNAME	VERTRE	VERTNAME
2	Schuster	1	Meier
4	Fischer	2	Schulze
6	Genzmann	3	Weber

Bewertung

Das Ergebnis entspricht leider nicht der Aufgabenstellung. Es sollten alle Kunden ausgegeben werden. Mit dieser Abfrage wurden aber nur diejenigen Kunden selektiert, denen auch ein Vertreter zugeordnet ist.

Die Schnittmenge aus Kunden und Vertretern wird mittels der Bedingung **TB1.VERTRE=TB2.VERTRE** gebildet. Bei fehlender Vertreterangabe ist die Schnittmenge leer. Folgerichtig wird auch keine Ergebniszeile erstellt.

Ein, der Aufgabenstellung entsprechendes Ergebnis, kann durch Formulierung eines **Outer Join** erzeugt werden.

Leider ist die Syntax des ANSI-Standards für Outer Joins in Oracle nicht implementiert. Kenntnisse bezüglich beider Varianten (Oracle und ANSI) sind zur Erlangung herstellerübergreifender Fertigkeiten jedoch unerlässlich.

Oracle

```
SELECT TB2.KUNDNR, TB2.KURZNAME AS KUNDNAME, TB2.VERTRE, TB1.KURZNAME AS VERTNAME
FROM TB2, TB1 WHERE TB1.VERTRE(+) = TB2.VERTRE ORDER BY TB2.KUNDNR;
```

ANSI

```
SELECT TB2.KUNDNR, TB2.KURZNAME AS KUNDNAME, TB2.VERTRE, TB1.KURZNAME AS VERTNAME
FROM TB2 LEFT OUTER JOIN TB1 ON TB1.VERTRE = TB2.VERTRE ORDER BY TB2.KUNDNR;
```

Ergebnis

KUNDNR	KUNDNAME	VERTRE	VERTNAME
1	Kleine		
2	Schuster	1	Meier
3	Winkelmann		
4	Fischer	2	Schulze
5	Vogel		
6	Genzmann	3	Weber

Anwendungsregeln

Oracle

Linke Seite	Operator	Rechte Seite
TB1.VERTRE(+)	=	TB2.VERTRE
Falls die Schnittmenge aus beiden beteiligten Tabellen leer ist, werden die Daten aus der Tabelle des mit (+) gekennzeichneten Feldes als NULL-Werte in die Ergebnismenge aufgenommen.		

ANSI

Linke Seite	Operator	Rechte Seite
TB2	LEFT OUTER JOIN	TB1
		Falls die Schnittmenge aus beiden beteiligten Tabellen leer ist, werden die Daten aus der hier genannten Tabelle als NULL-Werte in die Ergebnismenge aufgenommen.

Übungsaufgabe

Selektieren Sie alle Artikel mit den evtl. zugeordneten Hauptlieferanten. Das Ergebnis soll die Spalten ART.ARTIKEL, ART.BEZEICHN, ART.HPTLIE und LIE.KURZNAME enthalten sowie nach ART.ARTIKEL aufsteigend sortiert sein. Die Abfrage soll sowohl für Oracle als auch für Interbase (ANSI) formuliert werden.

Ausgabebeispiel

ARTIKEL	BEZEICHN	HPTLIE	KURZNAME
10.1001	Gartenpumpe	1	WEBER, PADERBORN
10.1016	Hochdruckreiniger	1	WEBER, PADERBORN
25.3282	Holzlasur	2	MILLER, HAMILTON
56.7954	Fliesenkleber	3	LAHO, BAGNEUX
80.0001	Montageleistungen		

Lösung

Oracle

```
SELECT ART.ARTIKEL,ART.BEZEICHN,ART.HPTLIE,LIE.KURZNAME
FROM ART,LIE WHERE LIE.LIEFNR(+)=ART.HPTLIE ORDER BY ART.ARTIKEL;
```

ANSI

```
SELECT ART.ARTIKEL,ART.BEZEICHN,ART.HPTLIE,LIE.KURZNAME
FROM ART LEFT OUTER JOIN LIE ON LIE.LIEFNR=ART.HPTLIE ORDER BY ART.ARTIKEL;
```

Unterabfragen

Unterabfragen sind Sql-Abfragen mit genau einem Rückgabewert bzw. Ergebnisfeld. Dieser Rückgabewert kann, ähnlich wie eine normale Spaltenangabe, in den verschiedenen Klauseln einer Sql-Select-Abfrage benutzt werden.

Sämtliche Spaltenwerte der in der FROM-Klausel der Hauptabfrage benannten Tabellen können zur Formulierung von Bedingungen in Unterabfragen genutzt werden.

Unterabfragen als Spaltenersatz in der SELECT-Klausel

Je Artikel soll der geringste und der höchste Einkaufspreis aus Tabelle LSP ausgegeben werden. Das Ergebnis soll die Spalten ART.ARTIKEL, ART.BEZEICHN, den geringsten Einkaufspreis mit der Spaltenbezeichnung MINPREIS und den höchsten Einkaufspreis mit der Spaltenbezeichnung MAXPREIS enthalten sowie aufsteigend nach Artikelnummer (ART.ARTIKEL) sortiert sein.

```
SELECT ART.ARTIKEL,ART.BEZEICHN,
(SELECT MIN(PREIS1) FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL) AS MINPREIS,
(SELECT MAX(PREIS1) FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL) AS MAXPREIS
FROM ART ORDER BY ART.ARTIKEL;
```

ARTIKEL	BEZEICHN	MINPREIS	MAXPREIS
10.1001	Gartenpumpe	24,50	42,50
10.1016	Hochdruckreiniger	69,00	79,00
25.3282	Holzlasur		
56.7954	Fliesenkleber		
80.0001	Montageleistungen		

Hinweise

1. Die Unterabfrage wird durch runde Klammern begrenzt.
2. Dem Ergebnis der Unterabfrage wird mit dem Schlüsselwort **AS** ein Spaltenname zugewiesen.
3. Die Bedingung kann mit den Spaltenwerten der Hauptabfrage (hier: ART.ARTIKEL) formuliert werden.
4. Die Ergebnisspalte der Unterabfrage muss der Rückgabewert einer Aggregatfunktion sein (genau ein Satz).

Übungsaufgabe

Je Umsatzsteuerschlüssel (UST.USTSCHL) soll die Anzahl der Verwendungen in der Lieferantentabelle ermittelt werden. Das Ergebnis soll die Spalte USTSCHL und die Anzahl der Verwendungen mit der Spaltenbezeichnung ANZAHL enthalten sowie absteigend nach Anzahl sortiert sein.

Ausgabebeispiel

USTSCHL	ANZAHL
EU	2
Inland	1
Ausland	1

Lösung

```
SELECT USTSCHL,
(SELECT COUNT(*) FROM LIE WHERE LIE.USTSCHL=UST.USTSCHL) AS ANZAHL
FROM UST ORDER BY 2 DESC;
```

Die Funktion EXISTS

Syntax: bool EXISTS(string Query)

Das Ergebnis der Funktion EXISTS ist **false**, falls die als Parameter angegebene Sql-Abfrage (Query) eine leere Ergebnismenge zurückliefert. Andernfalls wird **true** zurückgegeben.

Wie bei allen Unterabfragen können Bedingungen mit den Spaltenwerten der Hauptabfrage formuliert werden.

Beispiel

Es sollen sämtliche Lieferanten mit Lieferantenummer (LIE.LIEFNR) und Kurzname (LIE.KURZNAME) angezeigt werden, für die keinerlei Preiseinträge in der Lieferantenpreistabelle LSP vorhanden sind. Das Ergebnis soll aufsteigend nach Lieferantenummer sortiert sein.

```
SELECT LIEFNR, KURZNAME FROM LIE
WHERE NOT EXISTS(SELECT LIEFNR FROM LSP WHERE LSP.LIEFNR=LIE.LIEFNR)
ORDER BY LIEFNR;
```

LIEFNR	KURZNAME
2	MILLER, HAMILTON
3	LAHO, BAGNEUX

Hinweis

Das Ergebnis der Abfrage in der Funktion EXISTS darf nicht als Aggregatfunktion (MIN,MAX,AVG,SUM,COUNT) formuliert sein, da diese Funktionen immer eine Ergebnismenge mit genau einem Satz erzeugen.

Die Abfrage

```
SELECT LIEFNR, KURZNAME FROM LIE
WHERE NOT EXISTS(SELECT COUNT(*) FROM LSP WHERE LSP.LIEFNR=LIE.LIEFNR)
ORDER BY LIEFNR;
```

liefert eine leere Ergebnismenge für die Hauptabfrage, da die Funktion COUNT(*) auch bei den Lieferanten 2 und 3 den Wert 0 zurückliefert und dadurch immer genau ein Satz in der Ergebnismenge der Unterabfrage enthalten ist. Der Rückgabewert der Funktion EXISTS ist damit für jeden Satz der Lieferantentabelle **true**. Das Ergebnis der Hauptabfrage ist, aufgrund der Negation (**NOT**), für jeden Satz der Lieferantentabelle **false**.

Übungsaufgabe

Erzeugen Sie eine Liste aller Länderschlüssel (LND.LANDSCHL), die weder in der Kundentabelle (KND) noch in der Lieferantentabelle (LIE) verwendet werden. Das Ergebnis soll aufsteigend sortiert sein.

Ausgabebeispiel

LANDSCHL
DK
E
GCA
NL
PNG

Lösung

```
SELECT LANDSCHL FROM LND
WHERE NOT EXISTS(SELECT LAND FROM KND WHERE KND.LAND=LND.LANDSCHL)
AND NOT EXISTS(SELECT LAND FROM LIE WHERE LIE.LAND=LND.LANDSCHL)
ORDER BY LANDSCHL;
```

Die Funktionen ANY und ALL

Mit den Funktionen **ANY** und **ALL** wird ein Spaltenwert der Hauptabfrage mit den Werten der Ergebnisspalte jedes einzelnen Satzes einer Unterabfrage verglichen.

```
Syntax: bool [(| Value {<=<|=|<>>|>=>} ALL (string Query) []]
        bool [(| Value {<=<|=|<>>|>=>} ANY(string Query) []]
```

Die Funktion **ANY** liefert **true** zurück, falls die Vergleichsbedingung auf mindestens einen Satz der Unterabfrage zutrifft. Falls kein Satz in der Ergebnismenge der Unterabfrage enthalten ist, wird **false** zurückgeliefert.

Die Funktion **ALL** liefert **true** zurück, falls die Vergleichsbedingung auf alle Sätze der Unterabfrage zutrifft. Falls kein Satz in der Ergebnismenge der Unterabfrage enthalten ist, wird **true** zurückgeliefert.

Beispiel 1 (ANY)

Es sollen alle Artikel aus Tabelle ART mit den Spalten ARTIKEL und BEZEICHN selektiert werden, bei denen der Verkaufspreis (PREIS1) kleiner oder gleich irgendeinem entsprechenden Einkaufspreis (PREIS1) aus Tabelle LSP ist. Das Ergebnis soll nach Artikelnummer aufsteigend sortiert sein.

```
SELECT ARTIKEL, BEZEICHN FROM ART
WHERE PREIS1 <= ANY(SELECT PREIS1 FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL)
ORDER BY ARTIKEL;
```

ARTIKEL	BEZEICHN
10.1001	Gartenpumpe

Erläuterungen

Für jeden einzelnen Satz der Hauptabfrage (hier: alle Sätze der Tabelle ART) wird die in der Funktion ANY genannte Unterabfrage ausgeführt. Falls das Feld PREIS1 aus dem Satz der Hauptabfrage kleiner oder gleich dem Inhalt des Feldes PREIS1 aus irgendeinem Satz der Unterabfrage ist, wird **true** als Ergebnis des Gesamtvergleichs zurückgegeben.

Der Verkaufspreis (PREIS1) für den Artikel 10.0001 beträgt 39.99.
In der Tabelle LSP sind die nachfolgenden Einkaufspreise gespeichert.

LIEFNR	MENGEV	PREIS1
1	1,00	27,95
1	10,00	26,50
1	100,00	24,50
4	1,00	42,50

Eine entsprechende Prüfschleife könnte z.B. wie folgt programmiert sein:

```
bool result = false;
for(int i=0; i<LSP.Count(); i++)
{
    if(ART.PREIS1<=LSP.PREIS1[i])
    {
        result = true;
        break;
    }
}
```

Beim letzten Satz der Ergebnismenge der Unterabfrage trifft die Prüfbedingung zu. Das Ergebnis wird auf **true** gesetzt und danach wird die Schleife beendet. Der Satz mit der Artikelnummer 10.0001 wird daher in die Ergebnismenge der Hauptabfrage aufgenommen.

Der Verkaufspreis (PREIS1) für den Artikel 10.1016 beträgt 99.99.
In der Tabelle LSP sind die nachfolgenden Einkaufspreise gespeichert.

LIEFNr	MENGEV	PREIS1
1	1,00	78,50
1	10,00	75,00
4	1,00	79,00
4	10,00	75,00
4	100,00	69,00

Die Vergleichsbedingung ist für keinen Satz aus der Ergebnismenge der Unterabfrage erfüllt.
Der Artikel 10.1016 wird deshalb nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Für die Artikel 25.3282, 56.7954 und 80.0001 erzeugt die Unterabfrage eine leere Ergebnismenge.
Der Rückgabewert ist **false**. Diese Sätze werden deshalb ebenfalls nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Beispiel 2 (ANY)

Es sollen alle Artikel aus Tabelle ART mit den Spalten ARTIKEL und BEZEICHN selektiert werden, bei denen der Verkaufspreis (PREIS1) größer als irgendein entsprechender Einkaufspreis (PREIS1) aus Tabelle LSP ist. Das Ergebnis soll nach Artikelnummer aufsteigend sortiert sein.

```
SELECT ARTIKEL, BEZEICHN FROM ART
WHERE PREIS1 > ANY(SELECT PREIS1 FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL)
ORDER BY ARTIKEL;
```

ARTIKEL	BEZEICHN
10.1001	Gartenpumpe
10.1016	Hochdruckreiniger

Erläuterungen

Der Verkaufspreis (PREIS1) des Artikels 10.1001 ist größer als jeder Einkaufspreis (PREIS1) des Lieferanten 1.
Der Satz mit der Artikelnummer 10.1001 wird deshalb in die Ergebnismenge der Hauptabfrage aufgenommen.

Der Verkaufspreis des Artikels 10.1016 übersteigt alle entsprechenden Preise in der Lieferantenpreistabelle.
Dieser Artikel wird deshalb ebenfalls in die Ergebnismenge der Hauptabfrage aufgenommen.

Für die Artikel 25.3282, 56.7954 und 80.0001 erzeugt die Unterabfrage eine leere Ergebnismenge.
Diese Sätze werden deshalb nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Beispiel 3 (ALL)

Es sollen alle Artikel aus Tabelle ART mit den Spalten ARTIKEL und BEZEICHN selektiert werden, bei denen der Verkaufspreis (PREIS1) größer ist als alle entsprechenden Einkaufspreise (PREIS1) aus Tabelle LSP.
Das Ergebnis soll nach Artikelnummer aufsteigend sortiert sein.

```
SELECT ARTIKEL, BEZEICHN FROM ART
WHERE PREIS1 > ALL(SELECT PREIS1 FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL)
ORDER BY ARTIKEL;
```

ARTIKEL	BEZEICHN
10.1016	Hochdruckreiniger
25.3282	Holzlasur
56.7954	Fliesenkleber
80.0001	Montageleistungen

Eine Prüfschleife für die Funktion ALL könnte z.B. wie folgt programmiert sein:

```
bool result = true;
for(int i=0;i<LSP.Count();i++)
{
    if(!(ART.PREIS1>LSP.PREIS1[i]))
    {
        result = false;
        break;
    }
}
```

Erläuterungen

Falls alle Sätze in der Ergebnismenge der Unterabfrage der Bedingung entsprechen oder falls die Ergebnismenge der Unterabfrage leer ist, wird **true** zurückgegeben.

Der Verkaufspreis des Artikels 10.1016 übersteigt alle entsprechenden Preise in der Lieferantenpreistabelle. Das Prüfergebnis ist **true**. Dieser Artikel wird deshalb ebenfalls in die Ergebnismenge der Hauptabfrage aufgenommen.

Für die Artikel 25.3282, 56.7954 und 80.0001 erzeugt die Unterabfrage eine leere Ergebnismenge. Bei einer leeren Ergebnismenge ist der Rückgabewert **true**. Diese Sätze werden deshalb, im Gegensatz zur Funktion ANY, in die Ergebnismenge der Hauptabfrage aufgenommen.

Die zum Artikel 10.1001 gespeicherten Einkaufspreise entsprechen nur teilweise (3 von 4) der Bedingung. Das Prüfergebnis ist deshalb **false**. Dieser Artikel wird nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Beispiel 4 (ALL)

Es sollen alle Artikel aus Tabelle ART mit den Spalten ARTIKEL und BEZEICHN selektiert werden, bei denen der Verkaufspreis (PREIS1) kleiner oder gleich allen entsprechenden Einkaufspreisen (PREIS1) aus Tabelle LSP ist. Das Ergebnis soll nach Artikelnummer aufsteigend sortiert sein.

```
SELECT ARTIKEL, BEZEICHN FROM ART
WHERE PREIS1 <= ALL(SELECT PREIS1 FROM LSP WHERE LSP.ARTIKEL=ART.ARTIKEL)
ORDER BY ARTIKEL;
```

ARTIKEL	BEZEICHN
25.3282	Holzlasur
56.7954	Fliesenkleber
80.0001	Montageleistungen

Erläuterungen

Die zum Artikel 10.1001 gespeicherten Einkaufspreise entsprechen nur teilweise (3 von 4) der Bedingung. Dieser Artikel wird deshalb nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Keiner der zum Artikel 10.1016 gespeicherten Einkaufspreise entspricht der Bedingung. Dieser Artikel wird deshalb ebenfalls nicht in die Ergebnismenge der Hauptabfrage aufgenommen.

Für die Artikel 25.3282, 56.7954 und 80.0001 erzeugt die Unterabfrage eine leere Ergebnismenge. Diese Sätze werden, wie im vorigen Beispiel, in die Ergebnismenge der Hauptabfrage aufgenommen.

Übungsaufgaben (Gebäudereinigung Petra Siffich)

1. Selektieren Sie alle Leistungen aus Tabelle LST mit der Spalte LEISTG, bei denen der Standardpreis (LST.PREIS1) kleiner ist als sämtliche Preiseinträge in der Tabelle der objektspezifischen Leistungspreise (OLP.PREIS1).

Ausgabebeispiel (Leere Ergebnismenge)

LEISTG

2. Selektieren Sie alle Leistungen aus Tabelle LST mit der Spalte LEISTG, bei denen der Standardpreis (LST.PREIS1) kleiner ist als irgendein beliebiger Preiseintrag aus der Tabelle der objektspezifischen Leistungspreise (OLP.PREIS1).

Ausgabebeispiel

LEISTG
Toilettenreinigung

3. Selektieren Sie alle Leistungen aus Tabelle LST mit der Spalte LEISTG, bei denen der Standardpreis (LST.PREIS1) größer oder gleich einem beliebigen Preiseintrag aus der Tabelle der objektspezifischen Leistungspreise (OLP.PREIS1) ist.

Ausgabebeispiel

LEISTG
Bodennassreinigung
Fensterreinigung
Toilettenreinigung
Unterhaltsreinigung

4. Selektieren Sie alle Leistungen aus Tabelle LST mit der Spalte LEISTG, bei denen der Standardpreis (LST.PREIS1) größer oder gleich sämtlichen Preiseinträgen in der Tabelle der objektspezifischen Leistungspreise (OLP.PREIS1) ist.

Ausgabebeispiel

LEISTG
Bodennassreinigung
Fensterreinigung
Unterhaltsreinigung

Anmerkung

Betrachten Sie die Tabelleninhalte und überprüfen Sie die Ergebnismengen!

Lösungen

1. SELECT LEISTG FROM LST
WHERE PREIS1<ALL(SELECT PREIS1 FROM OLP WHERE OLP.LEISTG=LST.LEISTG)
ORDER BY LEISTG;
2. SELECT LEISTG FROM LST
WHERE PREIS1<ANY(SELECT PREIS1 FROM OLP WHERE OLP.LEISTG=LST.LEISTG)
ORDER BY LEISTG;
3. SELECT LEISTG FROM LST
WHERE PREIS1>=ANY(SELECT PREIS1 FROM OLP WHERE OLP.LEISTG=LST.LEISTG)
ORDER BY LEISTG;
4. SELECT LEISTG FROM LST
WHERE PREIS1>=ALL(SELECT PREIS1 FROM OLP WHERE OLP.LEISTG=LST.LEISTG)
ORDER BY LEISTG;

Die Klausel GROUP BY ... HAVING ...

Mit dieser Klausel kann das Ergebnis der Aggregatfunktionen MIN, MAX, AVG, SUM und COUNT auf andere Spaltenwerte, die keine Aggregatfunktion beinhalten, bezogen werden. Falls eine WHERE-Klausel vorhanden ist, muss die Klausel GROUP BY im Anschluß an die WHERE-Bedingung formuliert werden.

Beispiel (GROUP BY ...)

Je Zimmer soll die Anzahl der Belegungstage durch Gäste ermittelt werden. Die Anzahl der Belegungstage soll mit der Spaltenbezeichnung ANZAHL ausgegeben werden. Das Ergebnis soll aufsteigend nach der Zimmernummer sortiert sein.

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL GROUP BY ZIMMER ORDER BY ZIMMER;
```

ZIMMER	ANZAHL
101	6
102	3

Durch die Klausel GROUP BY ZIMMER wird festgelegt, dass sich das Ergebnis der Aggregatfunktion auf jedes einzelne Zimmer beziehen soll. Da die Aggregatfunktion COUNT(GASTNR) die Anzahl der Sätze zählt, in denen das Feld GASTNR ungleich NULL ist, erhalten wir, auch ohne Angabe einer WHERE-Bedingung, die geforderten Belegungstage durch Gäste.

Die Gesamtanzahl der Belegungstage je Zimmer (Gastbelegungen und Belegungen für Renovierungsarbeiten etc.) kann mit folgender Sql-Abfrage ermittelt werden:

```
SELECT ZIMMER,COUNT(*) AS ANZAHL FROM BEL GROUP BY ZIMMER ORDER BY ZIMMER;
```

ZIMMER	ANZAHL
101	6
102	4

Bedingungen bezüglich des Ergebnisses von Aggregatfunktionen werden in der Klausel HAVING formuliert.

Beispiel (GROUP BY ... HAVING ...)

Je Zimmer soll die Anzahl der Belegungstage durch Gäste ermittelt werden. Die Anzahl der Belegungstage soll mit der Spaltenbezeichnung ANZAHL ausgegeben werden. Die Ausgabe soll nur solche Zimmer umfassen, die an maximal 5 Tagen belegt sind. Das Ergebnis soll aufsteigend nach Zimmernummer sortiert sein.

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL  
GROUP BY ZIMMER HAVING COUNT(GASTNR)<=5 ORDER BY ZIMMER;
```

ZIMMER	ANZAHL
102	3

Die Formulierung von Bedingungen bezüglich des Ergebnisses von Aggregatfunktionen innerhalb der Klausel HAVING muss durch Wiederholung der Aggregatfunktion (COUNT(GASTNR)<=5) erfolgen. Die Angabe einer evtl. zugewiesenen Spaltenbezeichnung ist nicht zulässig.

Die folgende Abfrage wird deshalb als fehlerhaft vom Datenbanksystem abgewiesen:

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL
GROUP BY ZIMMER HAVING ANZAHL<=5 ORDER BY ZIMMER;
```

Sortierung

Eine Sortierung der Ergebnismenge nach dem Inhalt von Spalten mit Aggregatfunktionen ist unter Angabe der entsprechenden Spaltennummer in allen relevanten Datenbanksystemen möglich.

Beispiel

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL GROUP BY ZIMMER ORDER BY 2,ZIMMER;
```

oder

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL GROUP BY ZIMMER ORDER BY 2,1;
```

ZIMMER	ANZAHL
102	3
101	6

In Oracle können die Aggregatfunktionen und auch die evtl. zugewiesenen Spaltenbezeichnungen zusätzlich als Sortierkriterium benutzt werden.

Beispiel (Oracle)

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL GROUP BY ZIMMER
ORDER BY COUNT(GASTNR) ,ZIMMER;
```

oder

```
SELECT ZIMMER,COUNT(GASTNR) AS ANZAHL FROM BEL GROUP BY ZIMMER
ORDER BY ANZAHL ,1;
```

Merke: Alle Spalten in der SELECT-Klausel, die keine Aggregatfunktion beinhalten, müssen in die Klausel GROUP BY aufgenommen werden.

Die Sortierung der Ergebnismenge nach dem Inhalt einer Aggregatfunktion ist durch Angabe der Spaltennummer als Sortierkriterium immer möglich.

Reihenfolge: SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ...

Übungsaufgabe

Ermitteln Sie je Artikel den durchschnittlichen Einkaufspreis (LSP.PREIS1). Die Ausgabe soll die Artikelnummer, die Artikelbezeichnung und den durchschnittlichen Einkaufspreis mit der Spaltenbezeichnung DPREIS beinhalten. Einschränkend sollen nur Artikel mit einem Durchschnittspreis von mehr als 50 € ausgegeben werden. Sortieren Sie das Ergebnis absteigend nach dem Durchschnittspreis und dann aufsteigend nach der Artikelnummer.

Ausgabebeispiel

ARTIKEL	BEZEICHN	DPREIS
10.1016	Hochdruckreiniger	75,30

Lösung

```
SELECT LSP.ARTIKEL,ART.BEZEICHN,AVG(LSP.PREIS1) AS DPREIS FROM LSP,ART
WHERE ART.ARTIKEL=LSP.ARTIKEL
GROUP BY LSP.ARTIKEL,ART.BEZEICHN HAVING AVG(LSP.PREIS1)>50
ORDER BY 3 DESC,1;
```

UNION [ALL]

Die Resultate mehrerer Sql-Abfragen können zu einer Ergebnismenge zusammengefasst werden. Die Ergebnisspalten der beteiligten Sql-Abfragen müssen insgesamt identische Eigenschaften aufweisen. Standardmäßig werden doppelte Zeileninhalte im Gesamtergebnis unterdrückt (ähnlich DISTINCT). Die Ausgabe doppelter Zeileninhalte wird durch Angabe des optionalen Schlüsselworts ALL erzwungen.

Beispiel

Die Standardartikelbezeichnungen aus Tabelle ART sollen mit den fremdsprachlichen Bezeichnungen aus Tabelle AFS zu einer Ergebnismenge kombiniert werden. Die Sortierung soll aufsteigend nach Artikelnummer und Sprache erfolgen.

Die Spalte Sprache ist in der Tabelle ART nicht vorhanden. Da die Ergebnisspalten beider SQL-Abfragen identische Merkmale aufweisen müssen, d.h. auch die Anzahl der Spalten muss identisch sein, wird für die Tabelle ART ein Leerzeichen als Ersatz für die fehlende Sprachenspalte eingesetzt. Diese Pseudospalte wird mit dem Namen SPRACHE gekennzeichnet, da die Spaltenbezeichnungen immer aus der ersten beteiligten Sql-Abfrage übernommen werden.

In Interbase wird der Vergleich der Spalteneigenschaften gegenüber Oracle äußerst restriktiv gehandhabt. Sämtliche Spaltenmerkmale der beteiligten Sql-Abfragen, d.h. auch die Längenangaben, müssen in Interbase identisch sein. In Oracle ist es ausreichend, wenn die Datentypen identisch sind bzw. aus konstanten Ausdrücken abgeleitet werden können.

Zugewiesene Spaltennamen (AS ...) sind in Interbase als Sortierkriterium unzulässig. Ersatzweise werden hier die entsprechenden Spaltennummern als Sortierkriterium eingesetzt. Die Angabe von Spaltennummern ist in jedem Fall, auch in Oracle, zulässig. Die Klausel ORDER BY gilt für alle beteiligten Sql-Abfragen. Diese darf deshalb nur einmalig zum Ende der Abfragenkombination, nicht aber in den Einzelabfragen, vorhanden sein.

Oracle

```
SELECT ARTIKEL, ' ' AS SPRACHE,BEZEICHN FROM ART
UNION
SELECT ARTIKEL,SPRACHE,BEZEICHN FROM AFS
ORDER BY ARTIKEL,SPRACHE;
```

Interbase

```
SELECT ARTIKEL,CAST(' ' AS VARCHAR(10)) AS SPRACHE,BEZEICHN FROM ART
UNION
SELECT ARTIKEL,SPRACHE,BEZEICHN FROM AFS
ORDER BY 1,2;
```

ARTIKEL	SPRACHE	BEZEICHN
10.1001		Gartenpumpe
10.1001	Englisch	garden-pump
10.1001	Franzoes	Pompent de jardin
10.1016		Hochdruckreiniger
10.1016	Englisch	high-pressure-cleaner
10.1016	Franzoes	Nettoyeuse Forte pression
25.3282		Holzlasur
56.7954		Fliesenkleber
80.0001		Montageleistungen

Übungsaufgabe

Erzeugen Sie eine kombinierte Ergebnismenge aus den Hauptadressen und den Niederlassungsadressen aller Lieferanten. Die Ausgabe soll die Spalten LIEFNR, ADRNR, NAME1, NAME2, STRASSE und PLZORT umfassen. Als Ersatz für die fehlende Adressnummer (ADRNR) in der Lieferantentabelle LIE soll die Zahl 0 eingesetzt werden. Das Ergebnis soll nach Lieferantenummer (LIEFNR) und Adressnummer (ADRNR) aufsteigend sortiert sein.

Entwickeln Sie die notwendigen Sql-Abfragen sowohl für Oracle als auch für Interbase.

Ausgabebeispiel

LIEFNR	ADRNR	NAME1	NAME2	STRASSE	PLZORT
1		*		*	*
1	1	Weber Maschinenbau GmbH		Brunnenweg 7	33604 Bielefeld
1	2	Weber Maschinenbau GmbH	Zentrallager	Mindener Str. 5	32051 Herford
2		*		*	*
3		*		*	*
4		*		*	*

Lösung

Oracle

```
SELECT LIEFNR,0 AS ADRNR,NAME1,NAME2,STRASSE,PLZORT FROM LIE
UNION
SELECT LIEFNR,ADRNR,NAME1,NAME2,STRASSE,PLZORT FROM LAA
ORDER BY LIEFNR,ADRNR;
```

Interbase

```
SELECT LIEFNR,CAST(0 AS DECIMAL(3,0)) AS ADRNR,NAME1,NAME2,STRASSE,PLZORT FROM LIE
UNION
SELECT LIEFNR,ADRNR,NAME1,NAME2,STRASSE,PLZORT FROM LAA
ORDER BY 1,2;
```