



Anwendungsentwicklung Datenbanken SQL

Stefan Goebel

- Structured Query Language
strukturierte Abfragesprache
von ANSI und ISO standardisiert
- deklarativ bedeutet *was* statt *wie*
SQL beschreibt, welche Daten abzurufen, zu löschen oder einzufügen sind und nicht, wie das zu geschehen hat.
- (Am Beispiel MySQL)

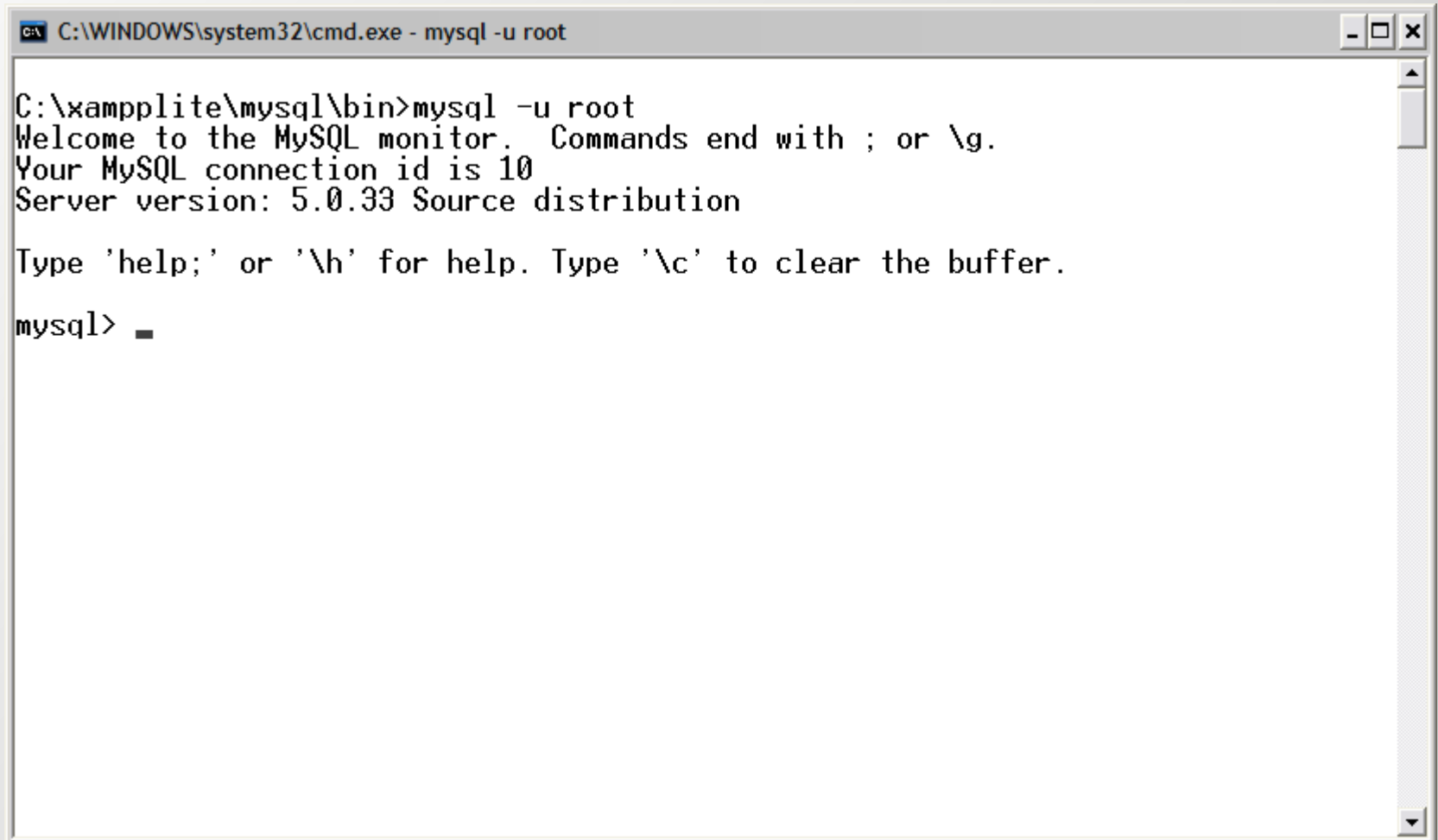
Vier Bereiche:

- **Data Definition Language (DDL)**
zur Definition des Datenbankschemas
Erstellen, ändern und löschen von
Datenbanken, Tabellen
- **Data Manipulation Language (DML)**
zur Datenmanipulation (lesen, schreiben,
ändern, etc.)
 - **Data Query Language (DQL)**
zur Datenabfrage
- **Data Control Language (DCL)**
zur Kontrolle der Datenbank-Ressourcen,
Berechtigungen

Zugriff auf MySQL-DB

- Konsole
- MySQL Workbench
- Phpmyadmin
- ODBC
- MySQL Connector/JDBC
- MySQL Connector/NET
- etc.

MySQL Konsole



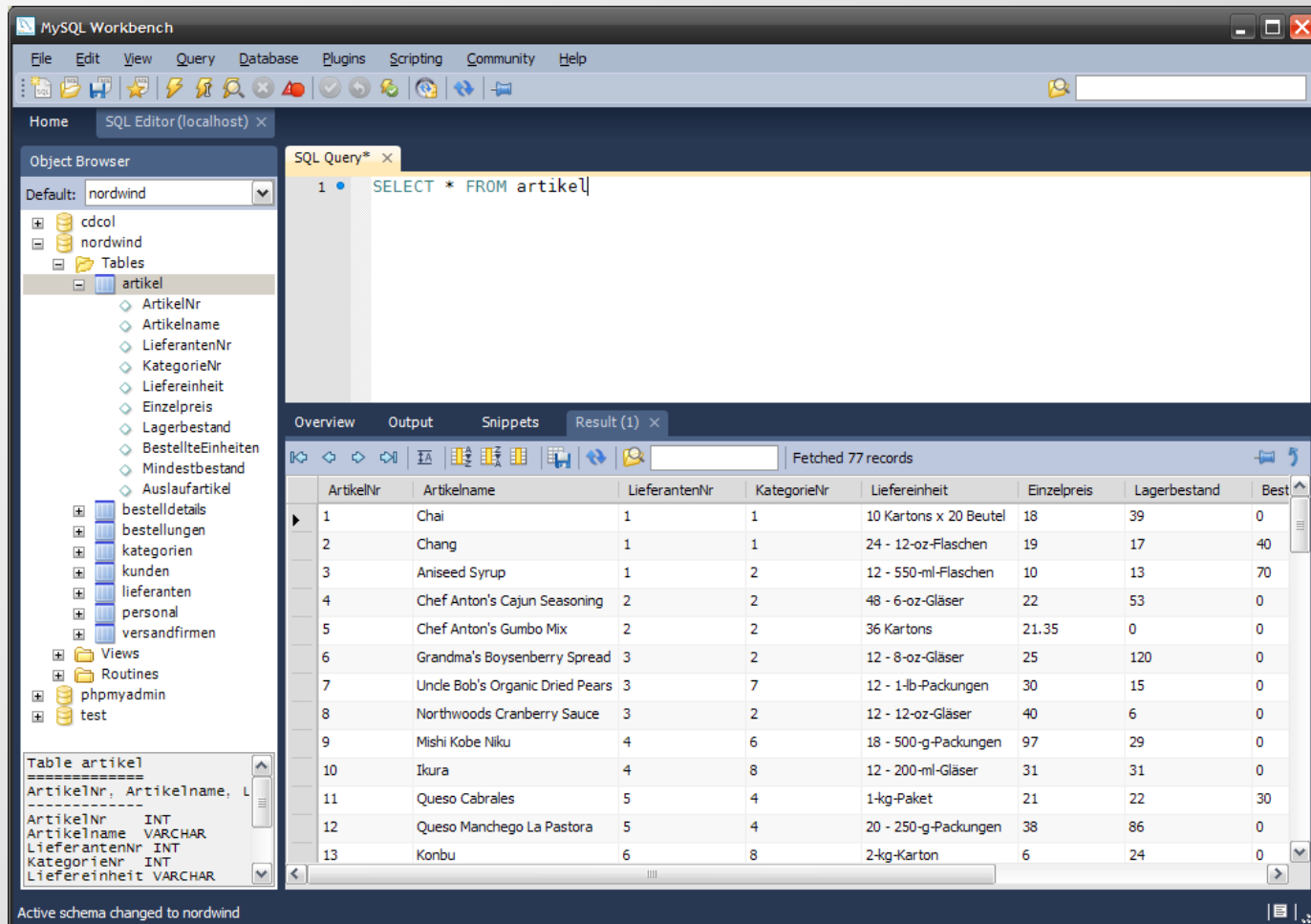
```
C:\WINDOWS\system32\cmd.exe - mysql -u root

C:\xampplite\mysql\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.33 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> _
```

MySQL Workbench



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `SELECT * FROM artikel`. The Results window displays 77 records from the 'artikel' table. The table structure is as follows:

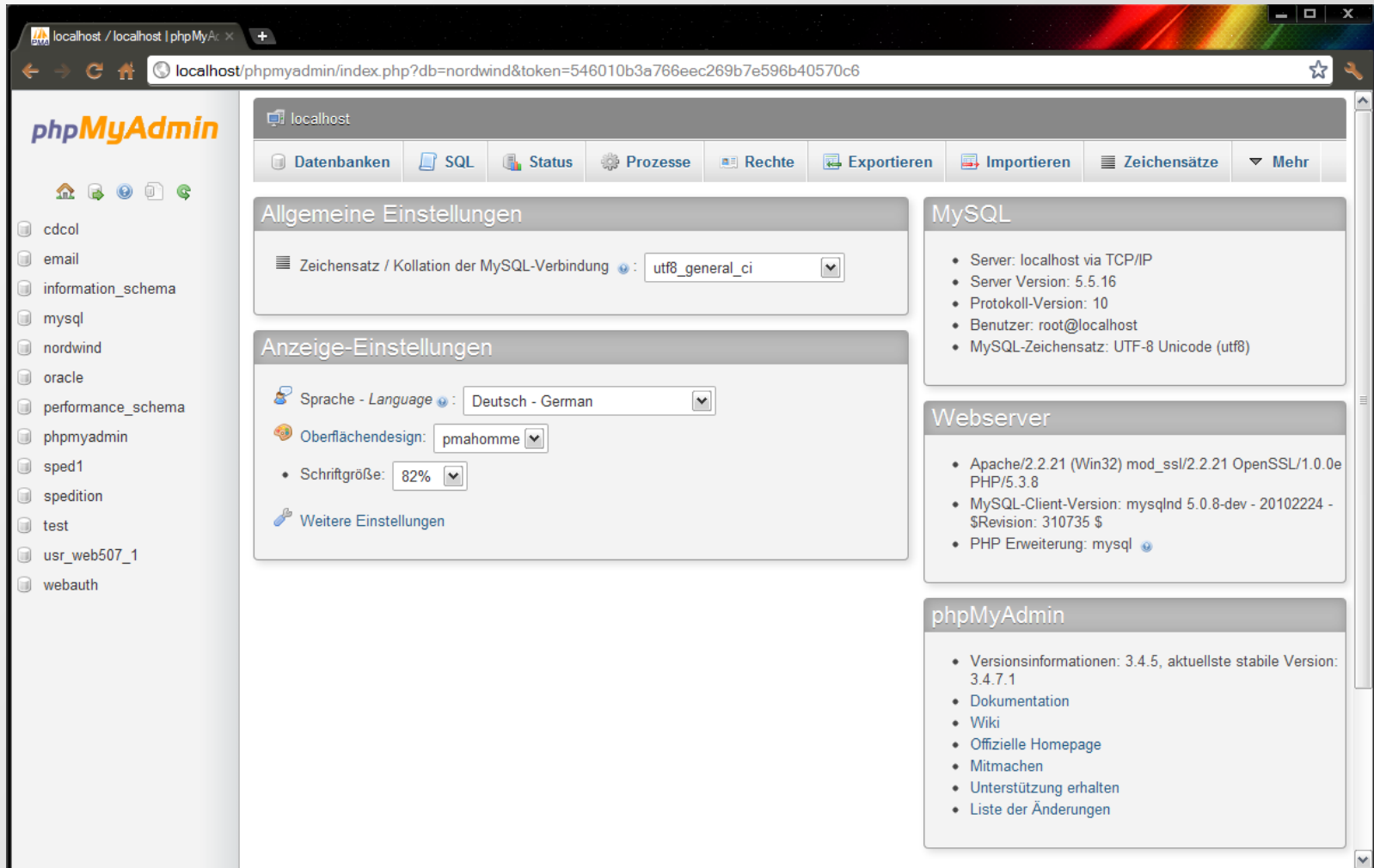
ArtikelNr	Artikelname	LieferantenNr	KategorieNr	Liefereinheit	Einzelpreis	Lagerbestand	Best
1	Chai	1	1	10 Kartons x 20 Beutel	18	39	0
2	Chang	1	1	24 - 12-oz-Flaschen	19	17	40
3	Aniseed Syrup	1	2	12 - 550-ml-Flaschen	10	13	70
4	Chef Anton's Cajun Seasoning	2	2	48 - 6-oz-Gläser	22	53	0
5	Chef Anton's Gumbo Mix	2	2	36 Kartons	21.35	0	0
6	Grandma's Boysenberry Spread	3	2	12 - 8-oz-Gläser	25	120	0
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1-lb-Packungen	30	15	0
8	Northwoods Cranberry Sauce	3	2	12 - 12-oz-Gläser	40	6	0
9	Mishi Kobe Niku	4	6	18 - 500-g-Packungen	97	29	0
10	Ikura	4	8	12 - 200-ml-Gläser	31	31	0
11	Queso Cabrales	5	4	1-kg-Paket	21	22	30
12	Queso Manchego La Pastora	5	4	20 - 250-g-Packungen	38	86	0
13	Konbu	6	8	2-kg-Karton	6	24	0

Table artikel
=====

```

ArtikelNr, Artikelname, L
-----
ArtikelNr    INT
Artikelname  VARCHAR
LieferantenNr INT
KategorieNr  INT
Liefereinheit VARCHAR
  
```

Active schema changed to nordwind



The screenshot shows the phpMyAdmin web interface in a browser window. The address bar displays the URL: localhost/phpmyadmin/index.php?db=nordwind&token=546010b3a766eec269b7e596b40570c6. The interface includes a navigation menu on the left with options like 'cdcol', 'email', 'information_schema', 'mysql', 'nordwind', 'oracle', 'performance_schema', 'phpmyadmin', 'sped1', 'spedition', 'test', 'usr_web507_1', and 'webauth'. The main content area is divided into several sections: 'Allgemeine Einstellungen' (General Settings) with a dropdown for 'Zeichensatz / Kollation der MySQL-Verbindung' set to 'utf8_general_ci'; 'Anzeige-Einstellungen' (Display Settings) with a dropdown for 'Sprache - Language' set to 'Deutsch - German', a dropdown for 'Oberflächendesign' set to 'pmahomme', and a dropdown for 'Schriftgröße' set to '82%'; 'MySQL' status information showing 'Server: localhost via TCP/IP', 'Server Version: 5.5.16', 'Protokoll-Version: 10', 'Benutzer: root@localhost', and 'MySQL-Zeichensatz: UTF-8 Unicode (utf8)'; 'Webserver' information showing 'Apache/2.2.21 (Win32) mod_ssl/2.2.21 OpenSSL/1.0.0e PHP/5.3.8', 'MySQL-Client-Version: mysqlnd 5.0.8-dev - 20102224 - \$Revision: 310735 \$', and 'PHP Erweiterung: mysql'; and 'phpMyAdmin' version information showing 'Versionsinformationen: 3.4.5, aktuellste stabile Version: 3.4.7.1', along with links for 'Dokumentation', 'Wiki', 'Offizielle Homepage', 'Mitmachen', 'Unterstützung erhalten', and 'Liste der Änderungen'.

Datenbankstruktur

localhost / localhost / nordwind

localhost/phpmyadmin/index.php?db=nordwind&token=546010b3a766eec269b7e596b40570c6

phpMyAdmin

nordwind

- artikel
- bestelldetails
- bestellungen
- kategorien
- kunden
- lieferanten
- personal
- versandfirmen

Erzeuge Tabelle

Struktur SQL Suche Abfrageeditor Exportieren Importieren Rechte Operationen Mehr

Tabelle	Aktion	Zeilen	Typ	Kollation	Größe	Überhang
<input type="checkbox"/> artikel		77	InnoDB	latin1_swedish_ci	16,0 KiB	-
<input type="checkbox"/> bestelldetails		2,155	InnoDB	latin1_swedish_ci	128,0 KiB	-
<input type="checkbox"/> bestellungen		830	InnoDB	latin1_swedish_ci	160,0 KiB	-
<input type="checkbox"/> kategorien		8	InnoDB	latin1_swedish_ci	144,0 KiB	-
<input type="checkbox"/> kunden		91	InnoDB	latin1_swedish_ci	16,0 KiB	-
<input type="checkbox"/> lieferanten		29	InnoDB	latin1_swedish_ci	16,0 KiB	-
<input type="checkbox"/> personal		9	InnoDB	latin1_swedish_ci	304,0 KiB	-
<input type="checkbox"/> versandfirmen		3	InnoDB	latin1_swedish_ci	16,0 KiB	-
8 Tabellen	Gesamt	3,202	InnoDB	latin1_swedish_ci	800,0 KiB	0 Bytes

Alle auswählen / Auswahl entfernen markierte:

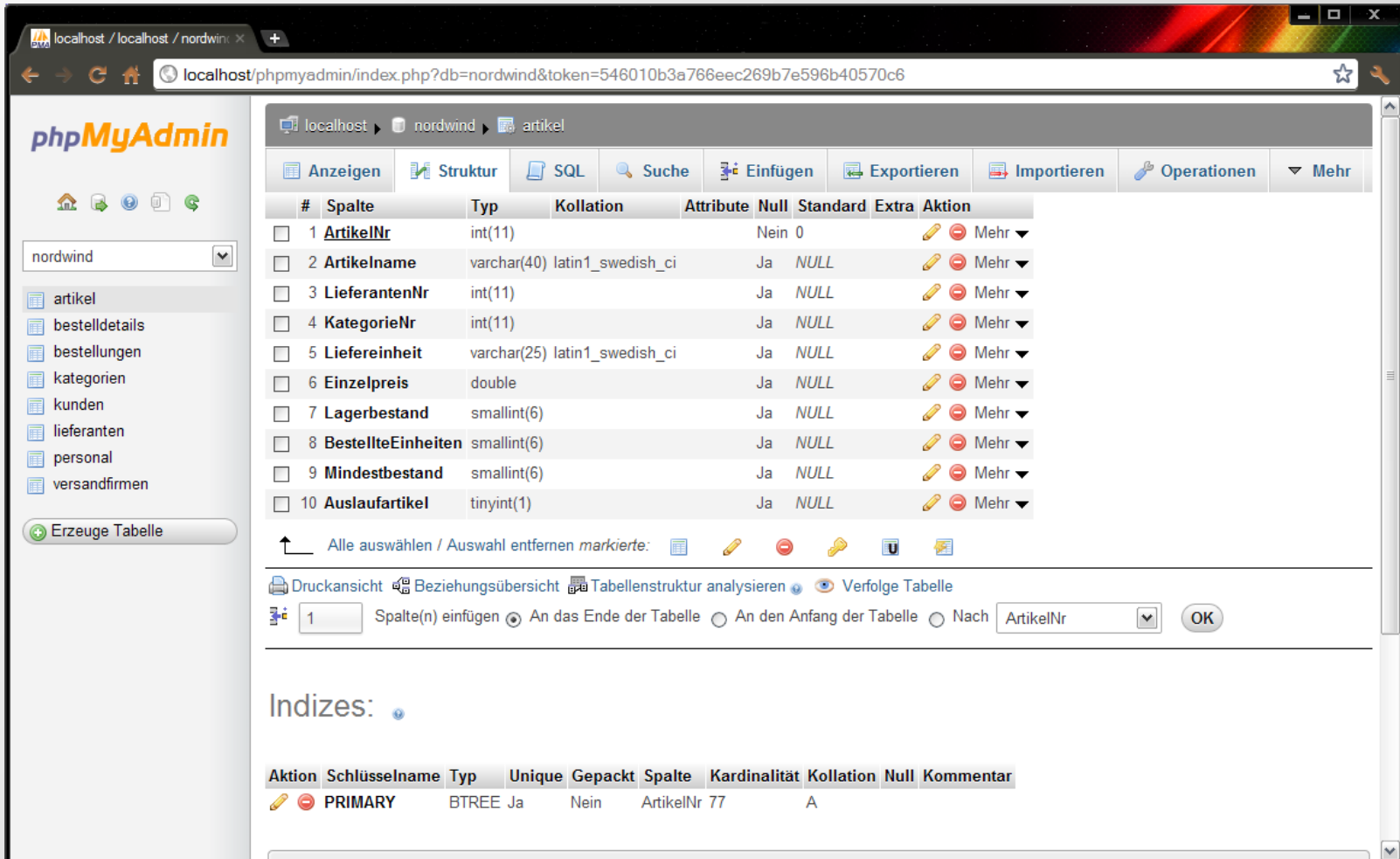
Druckansicht Strukturverzeichnis

Neue Tabelle in Datenbank nordwind erstellen

Name: Anzahl der Spalten:

OK

Tabellenstruktur



localhost / localhost / nordwind

localhost/phpmyadmin/index.php?db=nordwind&token=546010b3a766eec269b7e596b40570c6

phpMyAdmin

nordwind

- artikel
- bestelldetails
- bestellungen
- kategorien
- kunden
- lieferanten
- personal
- versandfirmen

Erzeuge Tabelle

localhost ▶ nordwind ▶ artikel

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Operationen Mehr

#	Spalte	Typ	Kollation	Attribute	Null	Standard	Extra	Aktion
<input type="checkbox"/>	1 <u>ArtikelNr</u>	int(11)			Nein	0		Mehr ▼
<input type="checkbox"/>	2 ArtikelName	varchar(40)	latin1_swedish_ci		Ja	NULL		Mehr ▼
<input type="checkbox"/>	3 LieferantenNr	int(11)			Ja	NULL		Mehr ▼
<input type="checkbox"/>	4 KategorieNr	int(11)			Ja	NULL		Mehr ▼
<input type="checkbox"/>	5 Liefereinheit	varchar(25)	latin1_swedish_ci		Ja	NULL		Mehr ▼
<input type="checkbox"/>	6 Einzelpreis	double			Ja	NULL		Mehr ▼
<input type="checkbox"/>	7 Lagerbestand	smallint(6)			Ja	NULL		Mehr ▼
<input type="checkbox"/>	8 BestellteEinheiten	smallint(6)			Ja	NULL		Mehr ▼
<input type="checkbox"/>	9 Mindestbestand	smallint(6)			Ja	NULL		Mehr ▼
<input type="checkbox"/>	10 Auslaufartikel	tinyint(1)			Ja	NULL		Mehr ▼

Alle auswählen / Auswahl entfernen markierte:

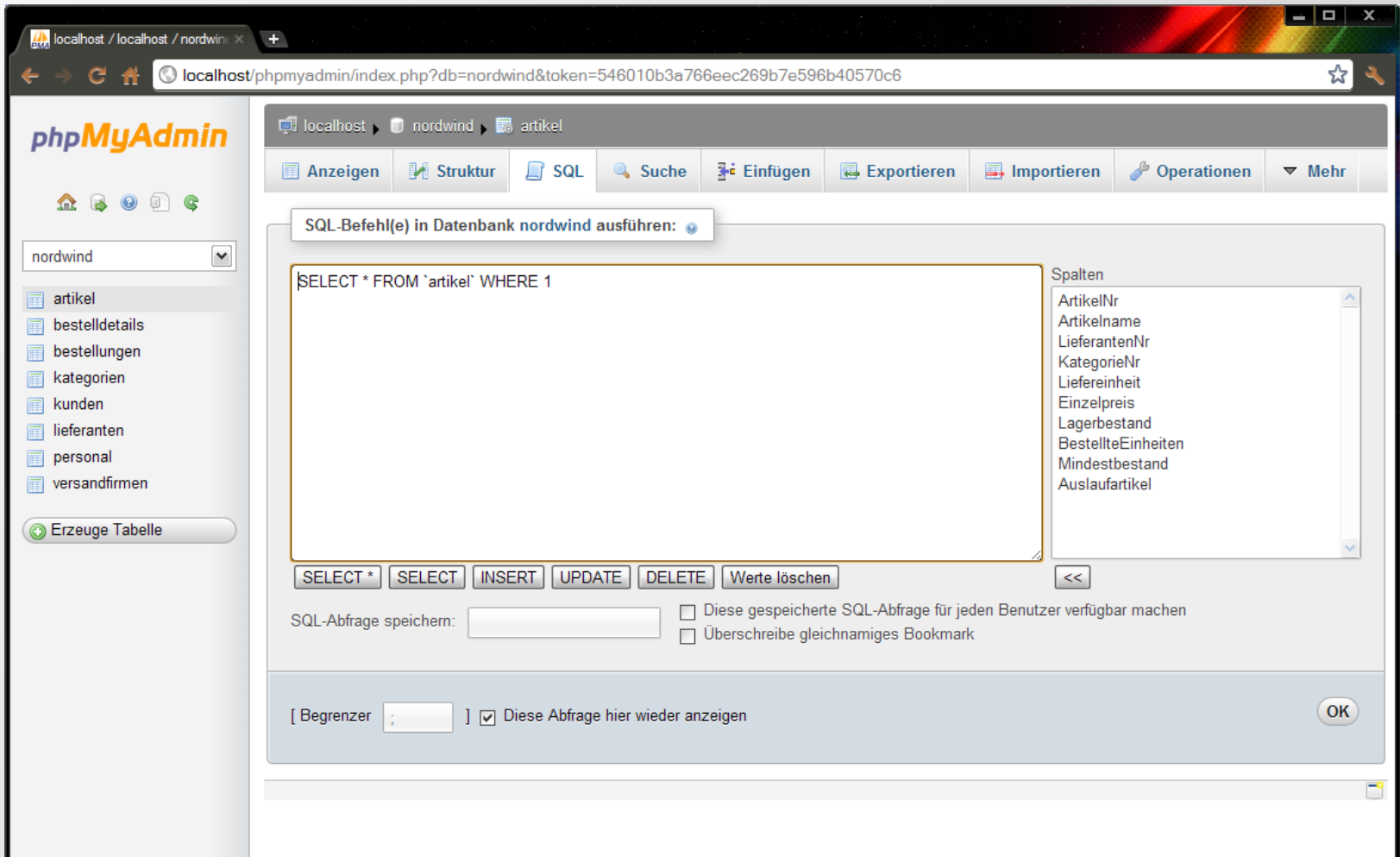
Druckansicht Beziehungsübersicht Tabellenstruktur analysieren Verfolge Tabelle

Spalte(n) einfügen An das Ende der Tabelle An den Anfang der Tabelle Nach

Indizes:

Aktion	Schlüsselname	Typ	Unique	Gepackt	Spalte	Kardinalität	Kollation	Null	Kommentar
	PRIMARY	BTREE	Ja	Nein	ArtikelNr	77	A		

SQL eingeben



The screenshot shows the phpMyAdmin interface in a browser window. The address bar displays the URL: localhost/phpmyadmin/index.php?db=nordwind&token=546010b3a766eec269b7e596b40570c6. The interface is for the 'nordwind' database, and the 'artikel' table is selected. The main area contains a text input field with the SQL query: `SELECT * FROM `artikel` WHERE 1`. To the right of the input field is a list of columns: ArtikelNr, Artikelname, LieferantenNr, KategorieNr, Liefereinheit, Einzelpreis, Lagerbestand, BestellteEinheiten, Mindestbestand, and Auslaufartikel. Below the input field are buttons for SQL operations: SELECT *, SELECT, INSERT, UPDATE, DELETE, and Werte löschen. There are also checkboxes for saving the query and a checkbox for displaying the query again. An OK button is located at the bottom right of the main area.

Vier Bereiche:

- **Data Definition Language (DDL)**
zur Definition des Datenbankschemas
Erstellen, ändern und löschen von
Datenbanken, Tabellen
- **Data Manipulation Language (DML)**
zur Datenmanipulation (lesen, schreiben,
ändern, etc.)
 - **Data Query Language (DQL)**
zur Datenabfrage
- **Data Control Language (DCL)**
zur Kontrolle der Datenbank-Ressourcen,
Berechtigungen

Erstellen von Datenbanken

- Datenbank verbinden
- CONNECT Datenbankname;
- CREATE DATABASE Datenbankname;

Kleine Auswahl

- Zahlen
 - INT
 - DECIMAL [(m,d)]
- Zeichenketten
 - VARCHAR(n)
- Datum
 - DATE
 - TIME
 - DATETIME

Erstellen von Tabellen

```
CREATE TABLE Tabellename (  
    t_id INT PRIMARY KEY AUTO_INCREMENT,  
    Spalte1 Typ NOT NULL,  
    Spalte2 Typ UNIQUE,  
    anrede CHAR CHECK (anrede IN('m', 'w')),  
    datum TIMESTAMP DEFAULT NOW(),  
    a_id INT,  
    FOREIGN KEY(a_id) REFERENCES Tab(a_id)  
)
```

ALTER TABLE TabName

- Feld hinzufügen, Typ angeben
ADD COLUMN (Spalte Typ, ...);
- Feldtyp ändern
MODIFY (Spalte NeuerTyp, ...);
- Feld löschen, Daten gehen verloren
DROP COLUMN Spalte;

Vier Bereiche:

- **Data Definition Language (DDL)**
zur Definition des Datenbankschemas
Erstellen, ändern und löschen von
Datenbanken, Tabellen
- ***Data Manipulation Language (DML)***
*zur Datenmanipulation (lesen, schreiben,
ändern, etc.)*
 - **Data Query Language (DQL)**
zur Datenabfrage
- **Data Control Language (DCL)**
zur Kontrolle der Datenbank-Ressourcen,
Berechtigungen

Einfügen von Daten

INSERT INTO Tabellename
(Spalte1, Spalte2, ...) . . .

- VALUES ('Wert 1', 'Wert 2', ...),
('Wert 3', 'Wert 4', ...),
(...);
- SELECT spalte1, spalte2, ... FROM ...;

Ändern von Daten

UPDATE TabName SET Spalte = Wert

WHERE Bedingung

Analog zu SELECT

ERST TESTEN!!

Löschen von Daten

DELETE FROM TableName

WHERE Bedingung

Analog zu SELECT

ERST TESTEN!!

Vier Bereiche:

- **Data Definition Language (DDL)**
zur Definition des Datenbankschemas
Erstellen, ändern und löschen von
Datenbanken, Tabellen
- **Data Manipulation Language (DML)**
zur Datenmanipulation (lesen, schreiben,
ändern, etc.)
 - **Data Query Language (DQL)**
zur Datenabfrage
- **Data Control Language (DCL)**
zur Kontrolle der Datenbank-Ressourcen,
Berechtigungen

Alle Spalten ausgeben:

```
SELECT * FROM Tabellennamen
```

Bestimmte Spalten ausgeben:

```
SELECT Spalte1, Spalte2, ...  
FROM Tabellennamen
```

Nur eindeutige Zeilen ausgeben:

```
SELECT DISTINCT Spalte1, Spalte2, ...  
FROM TabName
```

Abfrage: einschränken

Nur Zeilen ausgeben, die eine Bedingung erfüllen

```
SELECT ... FROM ...
```

```
...WHERE Feld1 = 'Müller';
```

```
...WHERE Feld2 = 123;
```

```
...WHERE Feld1 LIKE 'Mü%'
```

```
...WHERE Feld1 LIKE 'Me_er'
```

Abfragen: logische Operatoren

Boolsche Algebra wird unterstützt

WHERE Spalte1 = 'Müller'

AND (Spalte2 = 123 **OR** Spalte2 = 500);

Hierbei gelten auch die Klammerregeln

Oder auch:

WHERE Spalte1 = 'Müller'

AND Spalte2 **IN** (123,500);

Abfragen: Datum

Format bei der Abfrage eines Datums

WHERE Datum = 'jjjj-mm-tt'

Datumsbereich

WHERE Datum BETWEEN
'jjjj-mm-tt' AND 'jjjj-mm-tt'

Aber auch für andere Bereiche.

Abfragen: sortieren

Suchergebnis sortiert ausgeben

```
SELECT ... FROM ... WHERE ...
```

```
ORDER BY Spalte1, Spalte2, ... ASC |  
DESC;
```

Datumsfunktionen

CURDATE() - aktuelles Datum

DATEDIFF(von, bis) -
Anzahl Tage zwischen zwei Daten

DAYOFYEAR() - Tag im Jahr

DAYOFWEEK(datum) - Wochentag
(1=Sonntag)

DAYNAME(datum) - Wochentag- Name

DATE_FORMAT(datum, 'Formatstring')
%W, %d.%m.%Y (%u.KW)
Saturday, 17.03.2017 (11.KW)

Aggregatfunktionen

- Anzahl
SELECT COUNT(*) FROM Tab;
- Größter/kleinster Wert
MAX(Spalte) / MIN(Spalte)
- Durchschnitt
AVG(Spalte)
- Summe
SUM(Spalte)

Abfragen: gruppieren

```
SELECT Spalte1, COUNT(Spalte2)  
FROM ... WHERE ...
```

- GROUP BY Spalte1
gruppiert nach Spalte1
- HAVING COUNT(Spalte2) \geq 3
einschränken der Ausgabe mit Ergebnis
der Funktion
- z.B.: SELECT KdNr, COUNT(BestellNr)
FROM bestellungen GROUP BY KdNr

Abfragen mit Bedingungen

```
SELECT Spalte1,  
       IF( Bedingung, Wenn_TRUE,  
          Wenn_FALSE)  
FROM ... WHERE ...
```

Bedingungen:

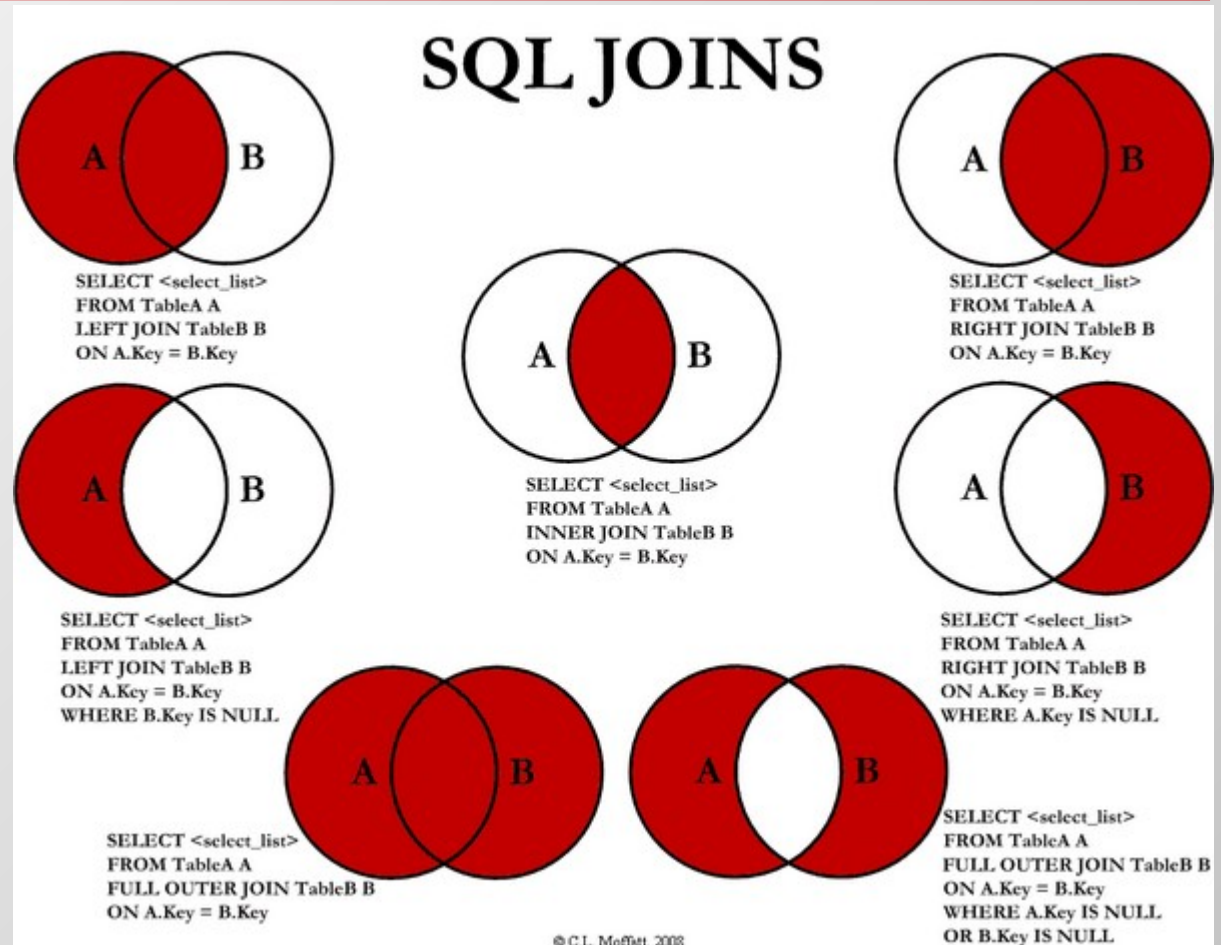
Spalte1 > Spalte2

Spalte1 = 42

Spalte3 = 'Text'

Mehrere Tabellen abfragen

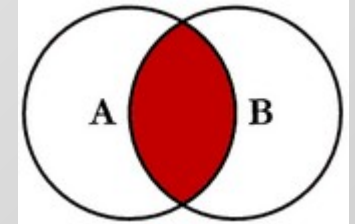
- Join
- Inner Join
- Outer Join
 - Left Join
 - Right Join
 - Full Outer
- Self Join



Quelle: http://www.codeproject.com/KB/database/Visual_SQL_Joins/Visual_SQL_JOINS_orig.jpg

Inner Join

Join - Bedingung angeben



```
SELECT .. FROM TabA INNER JOIN TabB
```

entweder: `ON TabA.Key = TabB.Key`

oder: `USING(Key)`

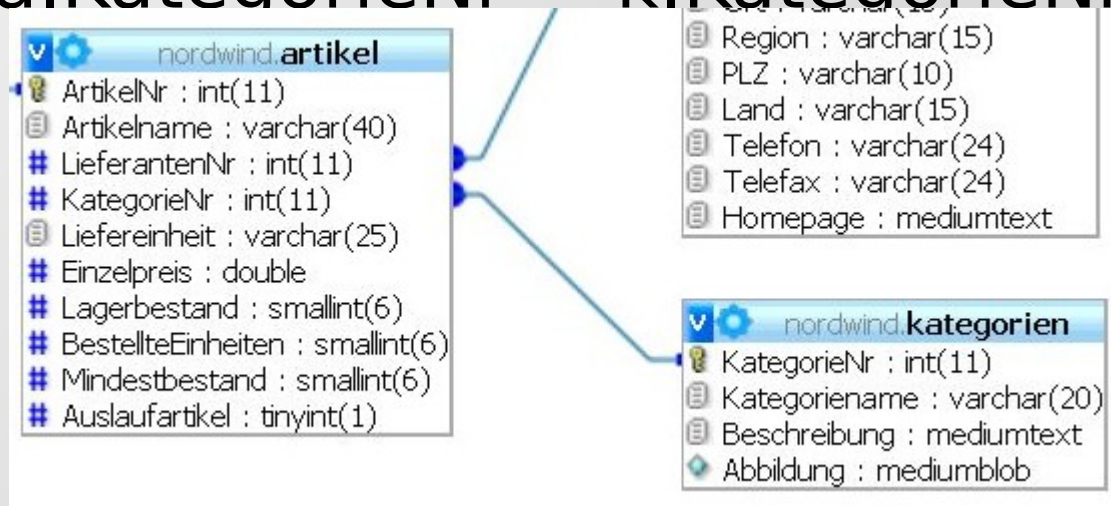
Ergebnis: nur die Datensätze, die in beiden Tabellen enthalten sind.

Ebenfalls möglich in WHERE-Klausel:

```
SELECT ... FROM TabA, TabB  
WHERE TabA.Key = TabB.Key
```

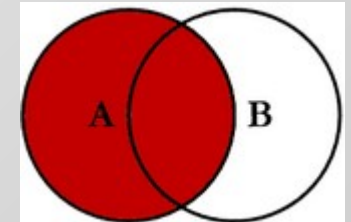
Inner Join-Beispiel

```
SELECT * FROM artikel a INNER JOIN  
kategorien k  
... USING(KategorieNr)  
... ON a.KategorieNr = k.KategorieNr
```



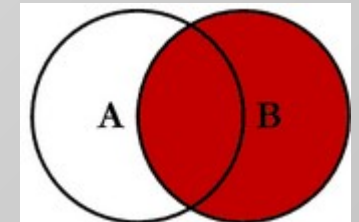
Outer Left Join

```
SELECT ... FROM TabA LEFT JOIN TabB  
USING(Key)
```



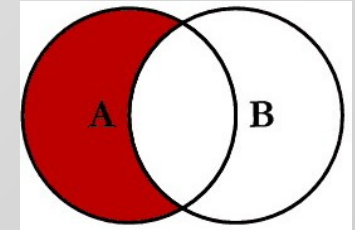
Ergebnis: alle Datensätze aus der Tabelle,
die **links** vom Join steht und die
dazugehörigen aus der **rechten** Tabelle

Right Join analog

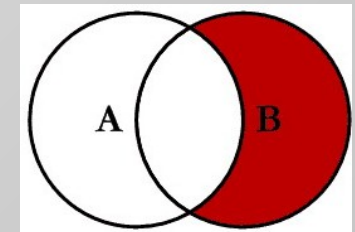


Left Excluding JOIN

```
SELECT ... FROM TabA LEFT JOIN TabB  
USING(Key)  
WHERE TabB.Key IS NULL
```



Ergebnis: alle Datensätze aus der Tabelle, die **links** vom Join steht, für die es keine entsprechenden Datensätze in der **rechten** Tabelle gibt.



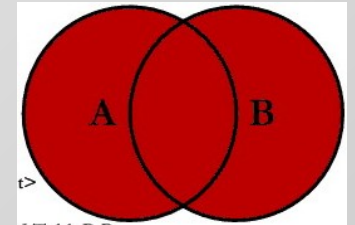
Right Join analog

FULL OUTER JOIN

SELECT ... FROM TabA LEFT JOIN TabB
USING(key)

UNION

SELECT ... FROM TabA RIGHT JOIN TabB
USING(key)



Ergebnis: alle Datensätze aus beiden
Tabellen.

```
SELECT ... FROM Tab AS AL1 LEFT  
JOIN Tab AS AL2 ON AL1.Key1 =  
AL2.Key2
```

Ergebnis: Verbindung einer Tabelle mit sich selbst.

<u>PersonalNr</u>	Name	VorgesetzterNr
45	Klaus	42
83	Dieter	42
42	Chef	0

SubSELECT

```
SELECT ... FROM ... WHERE Spalte1 IN ...  
... (Wert1, Wert2, ...)  
... (SELECT ... FROM ... )
```