

# MySQL – Informationsgewinnung

## Teil 3

Stefan Maihack Dipl. Ing. (FH)

Datum: 18.6.2013

# Mustervergleich

- Problem: Der gesuchte Datensatz ist nur teilweise bekannt
- Beispiel: alle Bücher zum Thema „MySQL“ → exakter Titel ist unbekannt  
alle Bücher eines Autors, dessen Name mit „D“ beginnt.
- Lösung: Vergleichsoperatoren für Mustervergleich
- Operatoren:
  - text1 = text2 zum Vergleich von zwei Texten
  - text LIKE muster zum Vergleich eines Textes mit einem Muster  
Muster und Text stimmen vollständig überein.
  - text REGEXP muster zum Vergleich eines Textes mit einem Muster  
das Muster ist irgendwo im Text enthalten
  - text LIKE muster

# Der LIKE Operator

## Funktionsweise

- Prinzip: - Muster aus Buchstaben und Platzhaltern wird mit einem Text verglichen  
- Keine Unterscheidung von Groß- und Kleinschreibung (Nicht Casesensitive)
- Syntax: text LIKE muster
- Platzhalter im Muster:                   % für eine beliebige Zeichenfolge (auch Leerzeichen)  
  \_ für ein beliebiges Zeichen
- Beispiele:                   „M%“ stimmt überein mit „Müller“, „Meier“, „M“  
                                  „M\_\_er“ stimmt überein mit „Müller“, „Meiser“  
                                  „M%er“ stimmt überein mit „Müller“, „Meister“, „Meier“  
                                  „\_ü%“ stimmt überein mit „Müller“, „Mühle“, „Güte“, „TÜV“
- Beispiele in MySQL: SELECT \* FROM autor WHERE nachname LIKE „Ki%“;  
  
    ➔ findet alles, was mit „Kl...“ beginnt, wie z.B. „King“

# Der LIKE Operator

## Optionen

- Groß und Kleinschreibung wird beachtet (Casesensitive): BINARY

Syntax: BINARY text LIKE muster

Beispiel: BINARY nachname LIKE „\_ü%“  
stimmt überein mit „Müller“ aber nicht mit „TÜV“

- Verneinung: NOT

Syntax: text NOT LIKE muster  
oder: NOT (text LIKE muster)

Beispiel: nachname NOT LIKE „\_ü%“  
stimmt überein mit „DÜbois“ aber nicht mit „Müller“

# Der LIKE-Operator

## Verwendung von Escape-Zeichen

- Problem: Die Platzhalter können auch als Zeichen im gesuchten Wort auftauchen
- Beispiel: gesucht wird der Text „100%“ → das Muster „100%“ liefert alle Texte, die mit der Zeichenfolge „100“ anfangen, z.B.: 1000
- Lösung: Definition eines Escape-Zeichens (Fluchtzeichen)

Syntax: text LIKE muster escape 'zeichen'

Beispiel: text LIKE „100#%“ escape '#'

stimmt überein mit „100%“ aber nicht mit „1000%“

Das Escape-Zeichen ist hier: #

Das Standard-Escapezeichen ist: \ (wie in Sprache C)

➔ Das Escape-Zeichen hebt die Sonderfunktion von % auf.

# Anwendungsbeispiel: nach dem Buchtitel suchen

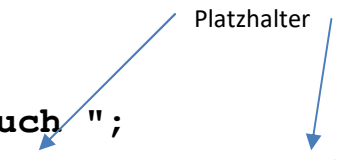
- Struktur:
  - Eingabemaske in Skript „nutzer\_pruefen.php“
  - Datensätze suchen in Skript „titel\_suchen.php“

- Eingabe in ein HTML-Formular im PHP-Skript nutzer\_pruefen.php:

```
printf("<form action=\"titel_suchen.php\" method=\"post\">");  
printf("<input type=\"text\" name=\"titelmuster\"> Titel <br>");  
printf("<input type=\"submit\" value=\"Nach Titeln suchen\">");  
printf("</form>");
```

- Suchbefehl im PHP-Skript titel\_suchen.php:

```
$kommando = "select titel from buch ";  
$kommando .= "where titel like \"%$titelmuster%\" ";  
$kommando .= "order by titel";  
$ergebnis = mysql_query($kommando);
```



- ergibt bei Eingabe von **MySQL** als Suchmuster das SQL-Kommando:

```
select titel from buch where titel like "%MySQL%" order by titel
```

# Der REGEXP Operator

- Prinzip:
  - Bildung von „regulären Ausdrücken“ (regular expression)
  - wie in UNIX-Kommandos „grep“ und „sed“
  - Unterscheidung von Groß- und Kleinschreibung
- Syntax:                   text REGEXP muster       oder       text RLIKE muster
- Reguläre Ausdrücke beschreiben Muster und definieren, wo diese Muster auftreten.
- Definition eines Musters „m“:

abcd	„m“ ist die Zeichenkette „abcd“
(abcd)	„m“ ist die Zeichenkette „abcd“ (als ein Element gruppiert)
[abcd]	„m“ ist die Zeichenkette „a“, „b“, „c“ und „d“
[a-g]	„m“ ist jedes Zeichen zwischen „a“ und „g“
.	„m“ ist jedes beliebige Zeichen

- Platzhalter                   Bedeutung
- %                               Eine Zeichenfolge aus null oder mehr Zeichen
- \_                               Ein einzelnes Zeichen
- [ ]                              Eine einzelnes Zeichen innerhalb eines angegebenen Bereichs
- [^]                             Ein einzelnes Zeichen, das nicht innerhalb des angegebenen Bereichs liegt

# Beispiele für REGEXP-Suchmuster

- Beispiele:
  - [y-z] passt zu dem Wort MySQL (wegen dem y)
  - (P.th.n) passt zu dem Wort Python (Punkte für ein beliebiges Zeichen)
  - (MySQL) passt zu dem Wort „MySQL“ aber nicht zu dem Wort „MYSQL“
  - [a] passt zu allen Wörtern in denen „a“ vorkommt

- Suchbefehl in einem PHP-Script:

```
$kommando = „SELECT titel FROM buch“;  
$kommando = „WHERE titel REGEXP \"(P.th.n)\" „;
```

- Ergibt das Kommando:

```
SELECT titel FROM buch WHERE titel REGEXP „(P.th.n)“;
```



# Weitere Angaben in REGEXP Suchmuster

- Für ein Muster „m“ kann der Ort spezifiziert werden:

<code>^m</code>	„m“ muss am Anfang des durchsuchenden Textes auftreten
<code>m\$</code>	„m“ muss am Ende des durchsuchenden Textes auftreten

- Beispiel: „`^MySQL`“ liefert nur Buchtitel die mit dem Wort „MySQL“ anfangen.

- Für ein Muster „m“ kann die Häufigkeit des Auftretens spezifiziert werden:

<code>m</code>	„m“ muss mindestens einmal auftreten
<code>m?</code>	„m“ darf höchstens einmal (oder überhaupt nicht) auftreten
<code>m*</code>	„m“ darf beliebig oft auftreten (oder auch gar nicht)
<code>m+</code>	„m“ darf beliebig oft auftreten (aber mindestens einmal)
<code>m{n1, n2}</code>	„m“ muss mindestens „n1“ mal, maximal aber „n2“ mal auftreten

- Beispiel:  
`MySQL{1, 2}` liefert den Buchtitel in denen „MySQL“ ein oder zweimal auftritt  
`^M.*L` liefert den Buchtitel in denen am Anfang ein „M“ und später ein „L“ auftritt

# Verknüpfung von Suchmuster

- Suchmuster können über einen oder-Operator „|“ verknüpft werden:

m1 | m2    Muster „m1“ oder „m2“ muss auftreten

- Beispiel: „MySQL | PHP“ liefert Bücher zu MySQL oder PHP

- Verknüpfung kann natürlich auch auf SQL-Ebene erfolgen:

```
SELECT titel FROM buch WHERE titel REGEXP „MySQL|PHP“
```

ist äquivalent zu:

```
SELECT titel FROM buch WHERE (titel REGEXP „MySQL“) OR (titel REGEXP „PHP“)
```

- REGEXP bietet noch viele weitere Verknüpfungsmöglichkeiten...