

Zur Definition von Textmustern können Platzhalter (Wildcards) verwendet werden:

Platzhalter	Erläuterung
%	Platzhalter für beliebige Zeichen in beliebiger Anzahl (einschließlich der Anzahl 0).
_ (underscore)	Platzhalter für genau ein beliebiges Zeichen.

Beispiele für den Vergleich eines Spaltennamens mit Textmustern für Maria DB und PostgreSQL:

Ort LIKE 'U%'	Vergleicht das Feld Ort mit einem Textmuster. Im Beispiel werden alle Orte berücksichtigt, die mit U beginnen und beliebig lang sind.
Ort LIKE '%au'	Berücksichtigt alle Orte, die mit beliebiger Zeichenfolge beginnen und mit den Buchstaben au enden, z. B. Blumenau.
Ort LIKE 'L%au'	Ergebnis: Orte, die mit L beginnen und au enden, z. B. Lindau, Langenau.
Ort LIKE 'L__au'	Ergebnis: Orte, die mit L beginnen und au enden und dazwischen genau drei beliebige Buchstaben enthalten, z. B. Lindau (nicht aber Langenau).
Ort LIKE '%au%'	Ergebnis: Orte, in deren Namen die Buchstabenfolge au an beliebiger Stelle vorkommt, z. B. Blaubeuren, Laupheim.

Beispiel:

Mit einer SELECT-Anweisung werden alle Kunden ausgegeben, deren Wohnort mit dem Buchstaben U beginnt.

Lösung:

```
SELECT Nachname, Vorname, Straße, PLZ, Ort, Telefon
FROM Kunden
WHERE Ort LIKE 'U*';
```

Beim Vergleich eines Textfeldes mit einem Textmuster stellt der Operator LIKE klar, dass das Textfeld mit einem Textmuster verglichen werden soll und somit im Beispiel alle Orte mit dem Anfangsbuchstaben U ausgegeben werden sollen.

Um nach Textmustern zu suchen, in denen die Zeichen % oder _ vorkommen (z. B. Passwort LIKE 'password%'), müssen diese als tatsächliche Zeichen kenntlich gemacht werden. Dies geschieht durch einen vorgestellten Escape-Strich \ (Backslash).

Beispiel:

```
SELECT *
FROM User
WHERE password LIKE '%\%';
```

Das erste Zeichen % wird als Wildcard für beliebig viele Zeichen interpretiert. Durch den Backslash \ wird das nächste Zeichen als Prozentzeichen interpretiert. Als gefunden werden somit alle Passwörter angezeigt, die als letztes Zeichen das Zeichen % enthalten, z. B. 'password%' und 'HjuZtJ_k556L%'.

Beispiel:

Eine Abfrage soll alle Passwörter ausgeben, die mit dem Zeichen 1 beginnen und mit dem Zeichen % enden.

Lösung:

```
SELECT *
FROM User
WHERE password LIKE '1%\%';
```

Platzhalter in Jet-SQL (MS Access):

In Access stehen folgende Platzhalter zur Verfügung:

Zeichen	Beschreibung	Beispiel
*	Dieses Zeichen entspricht einer beliebigen Anzahl von Zeichen. Sie können den Stern (*) an einer beliebigen Stelle in einer Zeichenfolge verwenden.	wo* findet wohin und woher, aber nicht warum.
?	Dieses Zeichen entspricht einem einzelnen Buchstaben an einer bestimmten Stelle.	B?llen findet Bullen, Ballen und Bellen.
#	Dieses Zeichen entspricht einer einzelnen Ziffer.	1#3 findet 103, 113 und 123, nicht aber 1223
[]	Für die Zeichen in der Klammer wird nach einer Entsprechung gesucht.	B[ae]llen findet Ballen und Bellen, aber nicht Bullen.
!	Mit diesem Zeichen werden Zeichen in den Klammern ausgeschlossen.	B[!ae]llen findet Bollen und Bullen, aber nicht Ballen und Bellen.
-	Dieses Zeichen bestimmt einen Zeichenbereich. Er muss in aufsteigender Reihenfolge angegeben werden (A-Z, nicht Z-A).	b[a-c]d findet bad, bbd und bcd.

Beispiel:

Es sollen Daten zurückgegeben werden, die mit dem Buchstaben „P“ beginnen, gefolgt von einem beliebigen Buchstaben von A bis F und drei Ziffern.

Lösung:

```
Like 'P[A-F]###'
```

Um nach Sonderzeichen zu suchen, werden sie im Textmuster in eckige Klammern eingeschlossen.

Beispiel:

Das Textmuster 'a[*]a' findet die Zeichenfolge 'a*a', nicht aber 'aaa'. Die Argumente können kombiniert werden:

Beispiel:

Analysieren Sie den folgenden LIKE-Operator:

```
... LIKE 'a[!b-m]#'
```

Ergebnis:

Das erste Zeichen muss 'a' oder 'A' sein, das zweite Zeichen ist beliebig außer der Buchstaben b bis m, das dritte Zeichen muss eine Ziffer sein. Das Textmuster findet z. B. 'An9', 'az0', 'a99', nicht aber 'abc' oder 'aj0'.