

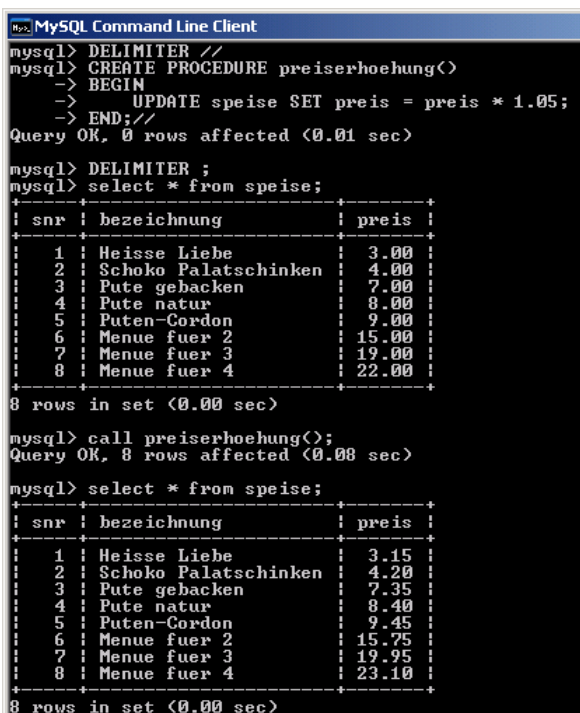
Vergleich: MySQL vs. PostgreSQL

Gespeicherte Routinen werden in MySQL und PostgreSQL unterschiedlich behandelt. Folgende Punkte sind in MySQL zu beachten:

- RETURNS VOID ist bei Funktionen unzulässig → als Prozedur formulieren
- Funktionen dürfen keine Anweisungen enthalten, die ein Result-Set erzeugen. Diese SELECT-Anweisungen müssen als SELECT ... INTO ... formuliert werden.
- Parameter müssen Name und Datentyp enthalten (d.h. so wie in modernen Programmiersprachen "üblich"), die Verwendung von \$1 etc. entfällt.
- Der Funktionskörper wird nicht mit
AS ' ... '
umschlossen, sondern mit
BEGIN ... END;
Zusätzlich muss (wie bei Triggern) das Abschluss-Zeichen (;) mit der Anweisung DELIMITER gesetzt bzw. geändert werden.

Beispiel:

```
DELIMITER //
CREATE PROCEDURE preiserhoehung()
BEGIN
    UPDATE speise SET preis = preis * 1.05;
END; //
DELIMITER ;
```



```
mysql> DELIMITER //
mysql> CREATE PROCEDURE preiserhoehung()
-> BEGIN
->     UPDATE speise SET preis = preis * 1.05;
-> END; //
Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;
mysql> select * from speise;
+----+-----+-----+
| snr | bezeichnung | preis |
+----+-----+-----+
| 1   | Heisse Liebe | 3.00  |
| 2   | Schoko Palatschinken | 4.00  |
| 3   | Pute gebacken | 7.00  |
| 4   | Pute natur | 8.00  |
| 5   | Puten-Cordon | 9.00  |
| 6   | Menue fuer 2 | 15.00 |
| 7   | Menue fuer 3 | 19.00 |
| 8   | Menue fuer 4 | 22.00 |
+----+-----+-----+
8 rows in set (0.00 sec)

mysql> call preiserhoehung();
Query OK, 8 rows affected (0.08 sec)

mysql> select * from speise;
+----+-----+-----+
| snr | bezeichnung | preis |
+----+-----+-----+
| 1   | Heisse Liebe | 3.15  |
| 2   | Schoko Palatschinken | 4.20  |
| 3   | Pute gebacken | 7.35  |
| 4   | Pute natur | 8.40  |
| 5   | Puten-Cordon | 9.45  |
| 6   | Menue fuer 2 | 15.75 |
| 7   | Menue fuer 3 | 19.95 |
| 8   | Menue fuer 4 | 23.10 |
+----+-----+-----+
8 rows in set (0.00 sec)
```