

Übung 2

1. Erstelle eine Stored Function EMPCOUNT mit folgenden Parametern:
Eingabeparameter: DEPTNO
Zur DEPTNO soll die Anzahl der Angestellten ermittelt und zurückgegeben werden. Teste die Funktion.
2. Erstelle eine Stored Procedure EMPNAME mit folgenden Parametern:
Eingabeparameter: EMPNO
CODE (1 oder 2)
Zur EMPNO soll ENAME ermittelt und als Parameter werden.
Zusätzlich werden in Abhängigkeit vom Code folgende Werte ermittelt und ebenfalls als Parameter zurückgegeben:
wenn CODE 1, dann SAL+COMM,
wenn CODE 2, dann nur SAL.
Falls die Prozedur den Angestellten nicht finden kann, soll für ENAME 'Unbekannt' und für das Gehalt NULL ausgegeben werden.
Teste die Prozedur, indem Sie aus einem unnamed block aufgerufen wird.
3. Erstelle eine Stored Function SAL_COMM mit folgenden Parametern:
Eingabeparameter: EMPNO
Zur EMPNO soll das Verhältnis zwischen SAL und COMM (Wert SAL/COMM) berechnet und folgender String zurückgeliefert werden: 'Verhältnis: ...'.
Hat ein Angestellter keine Provision, so soll 'Keine Provision!' zurückgeliefert werden (Verwende dazu die eingebaute Exception ZERO_DIVIDE).
Teste die Funktion.
4. The purpose of the procedure sell is to accept the paint_id parameter passed and update the stock on hand quantity by the qty value parameter. If there is insufficient stock then we need to produce an error message.
Before we get started with the procedure we need to quickly create the table (paints) that our Procedure will update.
The desc(ription) of our table should be.

```
SQL> desc paints;
```

Name	Null?	Type
PT_ID		NUMBER (5)
PT_NAME		VARCHAR2 (30)
PT_STK_ON_HAND		NUMBER (3)