Übung 3: PL/SQL

# Aufgabe 3.2 Realisierung einer min-max Skalierung mit Funktionen / Prozeduren

Die Skalierung selbst ist als **Funktion** implementiert. Sie benötigt als Parameter das alte/neue Minimum und Maximum zudem noch einen Wert der den aktuellen Wert darstellt. Sie gibt den skalierten Wert zurück.

create or replace FUNCTION scaling

(v NUMBER, min\_value NUMBER, max\_value NUMBER, new\_min NUMBER, new\_max NUMBER)

RETURN NUMBER

IS

BEGIN

RETURN ((v - min\_value) / (max\_value - min\_value)) \* (new\_max - new\_min) + new\_min;

END;

/

Die Funktion wird von einer **Prozedur** aufgerufen. Sie ermittelt den minimalen und maximalen Wert der bisherigen Attributwerte.

create or replace PROCEDURE calculateScaling

(min\_new NUMBER, max\_new NUMBER)

IS

min\_old NUMBER;

BEGIN

SELECT Min(Zahlen) INTO min\_old FROM Numbers;

**UPDATE Numbers SET ZAHLEN = SCALING (**

**Zahlen, min\_old, (SELECT MAX(Zahlen) FROM Numbers), min\_new, max\_new)**

);

END;

Diese Implementierung der Prozedur ersetzt alle Werte durch die skalierten Werte. Man könnte sich die skalierten Werte auch in einer eigenen Tabelle abspeichern.

create or replace PROCEDURE calculateScaling

(min\_new NUMBER, max\_new NUMBER)

IS

min\_old NUMBER;

BEGIN

SELECT Min(Zahlen) INTO min\_old FROM Numbers;

**INSERT INTO SCALINGTABLE (**

**SELECT SCALING(Zahlen, min\_old, (SELECT MAX(Zahlen) FROM Numbers), min\_new, max\_new)**

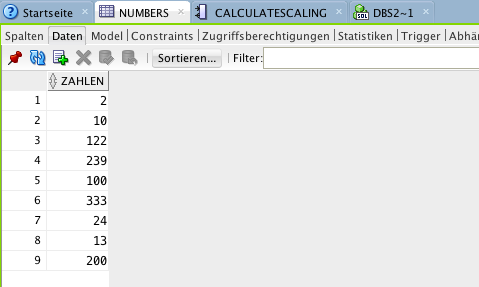
**FROM Numbers**

);

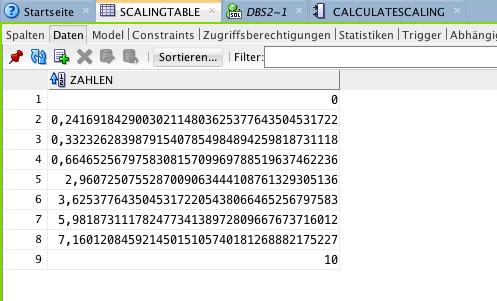
END;

Die Prozedur wird mit folgendem Befehl ausgeführt:

EXECUTE calculateScaling(min,max);



**Abbildung 1**: Tablle mit vorher belegten Zahlen



**Abbildung 2:** Werte nach Ausführung der Prozedur mit min = 0, max = 10

# Aufgabe 3.3 Migration über Prozeduren / Funktionen