# DBI²

# Einführung:

* Transaktionen
* PL/SQL, Procedure, Trigger, Function, Package
* ADO.NET, C#
* Normalformen, Modellierung
* DB-Admin:
  + Architektur
  + Backup/Recovery
  + Userverwaltung
* SQL

## Übung:

Bewegung

Konto

Konto(K#, Stand)

Bewegung(B#, Betrag, K#)

K# Stand B# Betrag K#

1. 5000 100, -1000,1
2. 6000 101, +2000, 1

Welche Konten haben keine Bewegung?

Select \* from Konto where

K# not in (select k#from bewegung);

## Operatoren:

IN

Vergleichsoperatoren: ><=…

Log Op: NOT, AND, OR

BETWEEN

LIKE

IS NULL

EXISTS (überprüft ob das sub select ein Ergebnis liefert oder nicht)

ANY, ALL

MINUS, INTERSECT (nur idente Reihen ausgegeben), UNION [ALL]

JOIN

* OUTER JOIN ()
* LEFT
* RIGHT
* FULL

**Gleiche Aufgabe wie früher:**

SELECT \* from konto k where NOT EXISTS

(select \* from bewegung b where b.K# = k.K#);

SELECT \* K# from konto

MINUS select K# from bewegung;

Welches Konto hat die wenigsten Bewegungen?

Select K# , count(B#)

From Konto left join bewegung b

Group by k#

Having count(b#)=

(select Min(count(b#)) from -“-);

# PL/SQL Procedure

## Cursor

Für Iteration durch Ergebnisreihen eines select – Kommando:

Select \* from tab; 🡪 Ergebnismenge 🡪 Cursor

Select count(\*) from tab; 🡪 skalares Ergebnis select … into …

Create or replace PROCEDURE xyz AS

DECLARE

…

Pl/SQL BLOCK

BEGIN

… unbenannter SQL block

EXCEPTION

…

END;

### Umgang mit Cursor:

DECLARE CURSOR ( in DECLARE Sektion)

CURSOR cursor\_name AS select – Kommando

Select Kommando 🡪 kann Bindevariablen enthalten : where x = my\_x

**OPEN Cursor:**

Führt select – Kommandos aus; Bindevariablen vorher versorgen!

**FETCH Cursor**

Holt Ergebnisreihe, auf die der Cursor verweist und setzt Cursor auf nächste Reihe

FETCH cursor\_name into Buffer Buffer 🡪 ist vom Cursor – Typ Cursor %ROWTYPE

**CLOSE Cursor**

Gibt belegte Speicherbereiche wieder frei

**Cursor – Attribute**

%FOUND 🡪 true, wenn Reihe vorhanden

%NOTFOUND 🡪 true, wenn Reihe nicht vorhanden

%ISOPEN

**Cursor – Schleifen**

Implizite Schleife:

For cursor IN cursor\_rec loop

…

End loop;

Entspricht der impliziten Form:

Cursor c AS select \* from tab;

C\_rec c%ROWTYPE;

Begin

OPEN c;

FETCH c in c\_rec;

While c%FOUND LOOP

…

FETCH c in c\_rec;

End loop;

Close C;

Ausgabe

Set serveroutput ON;

Begin

My\_function();

End;

DBMS\_OUTPUT.PUT\_LINE(‚xyz‘ || x )

**Aphrodite 4**

**192.168.128.152**

**D4a01**

**D4a**

Aufgabe:

Report über die Verkaufszahlen pro Land

Cursor C as select \* from verkauf

Order by B\_land;

C\_rec C%ROWTYPE;

Begin

Open c;

Fetch c into c\_rec;

BL:= c\_ rec.Bundesland;

While C% FOUND loop

If \_\_

# Transaktionen

(im Sinne von Sql)

Lock table in … mode;

X 🡪 EXCL