Hochschule München, Fakultät 07, SoSe 2016

Software-Architektur

Stichwortverzeichnis

Diana Irmscher

22. Juni 2016

Aufgabe 2

```
--------------------------------------------------------
--  DDL for Function MIN_MAX_SCALE
--------------------------------------------------------

  CREATE OR REPLACE FUNCTION ”MIN_MAX_SCALE”
  (min_old NUMBER, min_new NUMBER, max_old NUMBER, max_new
      NUMBER, v NUMBER)
RETURN NUMBER
IS
BEGIN
  RETURN (((v - min_old)/(max_old - min_old))*(max_new -
      min_new)) + min_new;
END;

/
```

```
--------------------------------------------------------
--  DDL for Procedure MIN_MAX_CALCULATOR
--------------------------------------------------------
-- Ergebnisse werden in neue Table eingetragen
  CREATE OR REPLACE PROCEDURE ”MIN_MAX_CALCULATOR”
  (min_new NUMBER,max_new NUMBER)
IS
min_old number;
BEGIN
 SELECT MIN(ZAHLEN) INTO min_old FROM NUMBERS;
  INSERT INTO NUMBERS_RESULT(
  SELECT MIN_MAX_SCALE(min_old, min_new, (SELECT MAX(ZAHLEN)
      FROM NUMBERS), max_new, ZAHLEN)
  FROM NUMBERS);
END;

/
```

```
-- Alternative: Update in gleicher Table
CREATE OR REPLACE PROCEDURE ”MIN_MAX_CALCULATOR”
  (min_new NUMBER,max_new NUMBER)
IS
min_old number;
BEGIN
 SELECT MIN(ZAHLEN) INTO min_old FROM NUMBERS;
```

```
  UPDATE NUMBERS SET ZAHLEN = MIN_MAX_SCALE(min_old, min_new, (
     SELECT MAX(ZAHLEN) FROM NUMBERS), max_new, ZAHLEN);
END;
```

```
--> Result:
EXECUTE min_max_calculator(0,10);

SELECT * FROM NUMBERS ORDER BY ZAHLEN ASC;
```

ZAHLEN
5
10
20
25
42
50
53
100
120
142
242
250
342
350
420

Aufgabe 3

```
--------------------------------------------------------
--  DDL for Table ANGESTELLTE
--------------------------------------------------------

  CREATE TABLE ”ANGESTELLTE” (
        ”A_NR” NUMBER(*,0),
        ”A_NAME” VARCHAR2(50 BYTE),
        ”A_GEBURTSDATUM” DATE,
        ”A_BERUFSBEZEICHNUNG” VARCHAR2(60 BYTE),
        ”A_MONATSGEHALT” NUMBER(*,0),
        ”A_GESCHLECHT” VARCHAR2(10 BYTE),
       PRIMARY KEY (”A_NR”));
```

```
--------------------------------------------------------
--  DDL for Table ARBEITER
--------------------------------------------------------
```

```
  CREATE TABLE "ARBEITER" (
          "A_NAME" VARCHAR2(30 BYTE),
          "A_VORNAME" VARCHAR2(30 BYTE),
          "A_GEBURTSMONAT" VARCHAR2(5 BYTE),
          "A_STUNDENLOHN" NUMBER(*,0),
          PRIMARY KEY ("A_NAME", "A_VORNAME"));
```

```
--------------------------------------------------------
--  DDL for Table BERUFE
--------------------------------------------------------

  CREATE TABLE "BERUFE" (
          "B_CODE" NUMBER(*,0),
          "B_TYPE" VARCHAR2(30 BYTE),
          PRIMARY KEY ("B_CODE"));
```

```
--------------------------------------------------------
--  DDL for Table GESCHLECHTER
--------------------------------------------------------

  CREATE TABLE "GESCHLECHTER" (
          "G_CODE" NUMBER(*,0),
          "G_TYPE" VARCHAR2(10 BYTE),
          PRIMARY KEY ("G_CODE"));
```

```
--------------------------------------------------------
--  DDL for Table PERSONAL
--------------------------------------------------------

  CREATE TABLE "PERSONAL" (
          "P_NR" NUMBER(*,0),
          "P_NAME" VARCHAR2(30 BYTE),
          "P_VORNAME" VARCHAR2(30 BYTE),
          "P_ALTER" NUMBER(*,0),
          "P_GESCHLECHT" NUMBER(*,0),
          "P_BERUFSCODE" NUMBER(*,0),
          "P_JAHRESEINKOMMEN" NUMBER(*,0),
          PRIMARY KEY ("P_NR"),
          FOREIGN KEY ("P_GESCHLECHT") REFERENCES "GESCHLECHTER"
              ("G_CODE"),
          FOREIGN KEY ("P_BERUFSCODE") REFERENCES "BERUFE" ("
              B_CODE"));
```

```
--------------------------------------------------------
--  DDL for Table ZUORDNUNG
--------------------------------------------------------
```

```
  CREATE TABLE "ZUORDNUNG" (
          "Z_NR" NUMBER(*,0) ,
          "Z_TABLE_OLD" VARCHAR2(30 BYTE) ,
          "Z_KEY_OLD" VARCHAR2(60 BYTE) ,
         PRIMARY KEY ("Z_NR") ,
         FOREIGN KEY ("Z_NR") REFERENCES "PERSONAL" ("P_NR"));
```

```
--------------------------------------------------------
--  INSERTS for Table GESCHLECHTER
--------------------------------------------------------
Insert into GESCHLECHTER (G_CODE,G_TYPE) values ('0','unbekannt
    ');
Insert into GESCHLECHTER (G_CODE,G_TYPE) values ('1','weiblich'
    );
Insert into GESCHLECHTER (G_CODE,G_TYPE) values ('2','maennlich
    ');
```

Aufgabe 3

```
--------------------------------------------------------
--  DDL for Sequence PNR_SEQUENCE
--------------------------------------------------------
   CREATE SEQUENCE  ”PNR_SEQUENCE” ;

/
```

```
--------------------------------------------------------
--  DDL for Table ANGESTELLTE
--------------------------------------------------------
  CREATE TABLE ”ANGESTELLTE”
   (        ”A_NR” NUMBER,
            ”A_NAME” VARCHAR2(50),
            ”A_GEBURTSDATUM” DATE,
            ”A_BERUFSBEZEICHNUNG” VARCHAR2(60),
            ”A_MONATSGEHALT” NUMBER,
            ”A_GESCHLECHT” VARCHAR2(10),
            PRIMARY KEY (”A_NR”)
   );

/
```

```
--------------------------------------------------------
--  DDL for Table ARBEITER
--------------------------------------------------------
  CREATE TABLE ”ARBEITER”
   (        ”A_NAME” VARCHAR2(30),
            ”A_VORNAME” VARCHAR2(30),
            ”A_GEBURTSMONAT” VARCHAR2(5),
            ”A_STUNDENLOHN” NUMBER,
            PRIMARY KEY (”A_NAME”, ”A_VORNAME”)
   );

/
```

```
--------------------------------------------------------
--  DDL for Table BERUFE
--------------------------------------------------------
  CREATE TABLE ”BERUFE”
   (        ”B_CODE” NUMBER,
            ”B_TYPE” VARCHAR2(30),
            PRIMARY KEY ("B_CODE")
    );

/
```

```
--------------------------------------------------------
--  DDL for Table GESCHLECHTER
--------------------------------------------------------

  CREATE TABLE "GESCHLECHTER"
   (    "G_NAME" VARCHAR2(15),
        "G_CODE" NUMBER,
            PRIMARY KEY ("G_NAME")
    );

/
```

```
--------------------------------------------------------
--  DDL for Table PERSONAL
--------------------------------------------------------

  CREATE TABLE "PERSONAL"
   (    "P_NR" NUMBER,
        "P_NAME" VARCHAR2(30),
        "P_VORNAME" VARCHAR2(30),
        "P_ALTER" NUMBER,
        "P_GESCHLECHT" NUMBER,
        "P_BERUFSCODE" NUMBER,
        "P_JAHRESEINKOMMEN" NUMBER,
            PRIMARY KEY ("P_NR"),
            FOREIGN KEY ("P_BERUFSCODE") REFERENCES "BERUFE" ("
                B_CODE")
    );

/
```

```
--------------------------------------------------------
--  DDL for Table ZUORDNUNG
--------------------------------------------------------


  CREATE TABLE "ZUORDNUNG"
   (    "Z_NR" NUMBER,
        "Z_TABLE_OLD" VARCHAR2(30),
        "Z_KEY_OLD" VARCHAR2(60),
            PRIMARY KEY ("Z_NR"),
        FOREIGN KEY ("Z_NR") REFERENCES "PERSONAL" ("P_NR")
    );

/
```

```
--------------------------------------------------------
--  Inserts in Table ANGESTELLTE
--------------------------------------------------------
Insert into ANGESTELLTE (A_NR,A_NAME,A_GEBURTSDATUM,
   A_BERUFSBEZEICHNUNG,A_MONATSGEHALT,A_GESCHLECHT) values ('1'
   ,'Fabian␣Uhlmann',to_date('03.11.88','DD.MM.RR'),'
   Informatiker','2000','m\E4nnlich');
Insert into ANGESTELLTE (A_NR,A_NAME,A_GEBURTSDATUM,
   A_BERUFSBEZEICHNUNG,A_MONATSGEHALT,A_GESCHLECHT) values ('2'
   ,'Diana␣Irmscher',to_date('01.01.90','DD.MM.RR'),'
   Informatiker','2001','weiblich');
Insert into ANGESTELLTE (A_NR,A_NAME,A_GEBURTSDATUM,
   A_BERUFSBEZEICHNUNG,A_MONATSGEHALT,A_GESCHLECHT) values ('3'
   ,'Alexandra␣Vogel',to_date('01.10.92','DD.MM.RR'),'
   Informatiker','9999','weiblich');
Insert into ANGESTELLTE (A_NR,A_NAME,A_GEBURTSDATUM,
   A_BERUFSBEZEICHNUNG,A_MONATSGEHALT,A_GESCHLECHT) values ('4'
   ,'Alexander␣Boxhorn',to_date('27.07.82','DD.MM.RR'),'
   Logistiker','1375','m\E4nnlich');

/
```

```
--------------------------------------------------------
--  Inserts in Table GESCHLECHTER
--------------------------------------------------------
Insert into DBST47.GESCHLECHTER (G_NAME,G_CODE) values ('
   Alexandra','1');
Insert into DBST47.GESCHLECHTER (G_NAME,G_CODE) values ('Fabian
   ','2');

/
```

```
--------------------------------------------------------
--  DDL for Function GETAGE
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "GETAGE"
  (birthdate Date)
RETURN VARCHAR2
```

```
IS
BEGIN
  RETURN Trunc((months_between(sysdate, birthdate) /12),0);
END;

/
```

```
--------------------------------------------------------
--  DDL for Function GETAGE
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "GETAGE"
   (birthdate VARCHAR)
RETURN VARCHAR2
IS
BEGIN
  -- TODO: SELECT EXTRACT(MONTH FROM SYSDATE) FROM DUAL;
  -- TODO: SELECT TO_DATE('2012-06-05', 'YYYY-MM-DD') FROM dual
      ;
  RETURN Trunc((months_between(sysdate, birthdate) /12),0);
END;

/
```

```
--------------------------------------------------------
--  DDL for Function GETFIRSTNAME
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "DBST47"."GETFIRSTNAME"
   (fname VARCHAR2)
RETURN VARCHAR2
IS
BEGIN
  RETURN SUBSTR(fname,0, instr(fname,'␣')-1);
END;

/
```

```


--------------------------------------------------------
--  DDL for Function GETGENDER
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "GETGENDER"
   (vname VARCHAR2)
RETURN NUMBER
```

```
IS
gendercode NUMBER;
BEGIN
  SELECT G_CODE INTO gendercode FROM GESCHLECHTER WHERE G_NAME
      = vname;
   IF gendercode = NULL THEN gendercode := 0;
  END IF;
  RETURN gendercode;
END;

/
```

```
--------------------------------------------------------
--  DDL for Function GETGENDERCODE
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION ”GETGENDERCODE”
   (gender VARCHAR2, firstname VARCHAR2)
RETURN NUMBER
IS
CURSOR CGCODE IS
          SELECT G_CODE
          FROM Geschlechter
  WHERE G_NAME = firstname;
gendercode NUMBER;
tmp NUMBER;
BEGIN
   IF gender = ’maennlich’ THEN gendercode := 2;
   ELSIF gender = ’weiblich’ THEN gendercode := 1;
  ELSE gendercode := 0;
  END IF;
  OPEN CGCODE;
  FETCH CGCODE into tmp;
   IF CGCODE%NOTFOUND then
     INSERT INTO GESCHLECHTER (G_NAME, G_CODE) VALUES (firstname
         ,gendercode);
  END IF;
  RETURN gendercode;
END;
```

```
--------------------------------------------------------
--  DDL for Function GETGENDERCODEFROMNAME
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION ”GETGENDERCODEFROMNAME”
  (vname VARCHAR2)
RETURN NUMBER
IS
gendercode NUMBER;
BEGIN
  SELECT G_CODE INTO gendercode FROM GESCHLECHTER WHERE G_NAME
      = vname;
  IF gendercode = NULL THEN gendercode := 0;
  END IF;
  RETURN gendercode;
END;

/
```

```
--------------------------------------------------------
--  DDL for Function GETLASTNAME
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "DBST47"."GETLASTNAME"
  (lname VARCHAR2)
RETURN VARCHAR2
IS
BEGIN
  RETURN SUBSTR(lname,INSTR(lname,' ')+1);
END;

/
```

```
--------------------------------------------------------
--  DDL for Function GETJOBCODE
--------------------------------------------------------
  CREATE OR REPLACE FUNCTION "GETJOBCODE"
  (jobname VARCHAR2)
RETURN NUMBER
IS
jobcode NUMBER;
BEGIN
  SELECT B_CODE INTO jobcode FROM BERUFE WHERE B_TYPE = jobname
      ;
  IF jobcode = NULL THEN
        SELECT max(B_CODE) INTO jobcode FROM BERUFE;
        jobcode := jobcode + 1;
        INSERT INTO BERUFE (B_CODE, B_TYPE) VALUES (jobcode,
            jobname);
  END IF;
  RETURN jobcode;
END;

/
```

```

--------------------------------------------------------------
--  DDL for Function GETMONEY
--------------------------------------------------------------

  CREATE OR REPLACE FUNCTION ”GETMONEY”
   (monthmoney NUMBER)
RETURN NUMBER
IS
BEGIN
  RETURN (monthmoney * 12);
END;

/
```

```
--------------------------------------------------------------
--  DDL for Procedure TRANSFORMATION_ANGESTELLTE
--------------------------------------------------------------

  CREATE OR REPLACE PROCEDURE ”TRANSFORMATION_ANGESTELLTE”
IS
a_nr NUMBER;
p_nr NUMBER;
p_name VARCHAR2(30);
p_vorname VARCHAR2(30);
p_age DATE;
p_geschlecht VARCHAR2(10);
p_job VARCHAR(50);
p_money NUMBER;
CURSOR CANGST IS
          SELECT A_Nr, A_Name, A_Geburtsdatum,
               A_Berufsbezeichnung, A_Monatsgehalt, A_Geschlecht
          FROM Angestellte;
BEGIN
  OPEN CANGST;
  LOOP
    FETCH CANGST INTO a_nr, p_name, p_age, p_job, p_money,
        p_geschlecht;
    EXIT WHEN CANGST%NOTFOUND;
    SELECT pnr_sequence.nextval INTO p_nr FROM DUAL;
    SELECT GETFIRSTNAME(p_name) INTO p_vorname FROM DUAL;
        INSERT INTO PERSONAL(p_nr,p_name,p_vorname,p_alter,
            p_geschlecht,p_berufscode,p_jahreseinkommen) VALUES
            (p_nr,GETLASTNAME(p_name),p_vorname,GETAGE(p_age),
            GETGENDERCODE(p_geschlecht,p_vorname),GETJOBCODE(
            p_job),GETMONEY(p_money));
    INSERT INTO ZUORDNUNG (Z_NR, Z_TABLE_OLD, Z_KEY_OLD) VALUES
        (p_nr, 'Angestellte', TO_CHAR(a_nr, '9999'));
  END LOOP;
  CLOSE CANGST;
END;

/
```

```
--------------------------------------------------------
--  DDL for Procedure TRANSFORMATION_ARBEITER
--------------------------------------------------------

  CREATE OR REPLACE PROCEDURE "TRANSFORMATION_ARBEITER"
IS
p_nr NUMBER;
p_name VARCHAR2(30);
p_vorname VARCHAR2(30);
p_age DATE;
p_geschlecht VARCHAR2(10);
p_job VARCHAR(50);
p_money NUMBER;
CURSOR CANGST IS
        SELECT A_Name, A_Vorname, A_Geburtsmonat, A_Stundenlohn
        FROM Arbeiter;
BEGIN
  OPEN CARB;
  LOOP
        FETCH CARB INTO p_name, p_vorname, p_age, p_money;
        EXIT WHEN CANGST%NOTFOUND;
        p_nr := pnr_sequence.nextval

        INSERT INTO PERSONAL(p_nr,p_name,p_vorname,p_alter,
            p_geschlecht) VALUES (p_nr,p_name,p_vorname,GETAGE(
            p_age),GETGENDERCODE(p_geschlecht,p_vorname),
            GETJOBCODE(p_job),GETMONEY(p_money));
        -- TODO: CONCAT(CONCAT(p_name, ', '),p_vorname) -->
            arb_nr erstellen
        INSERT INTO ZUORDNUNG (Z_NR, Z_TABLE_OLD, Z_KEY_OLD)
```

```
            VALUES (p_nr, 'Arbeiter', arb_nr);
  END LOOP;
  CLOSE CANGST;
END;

/
```