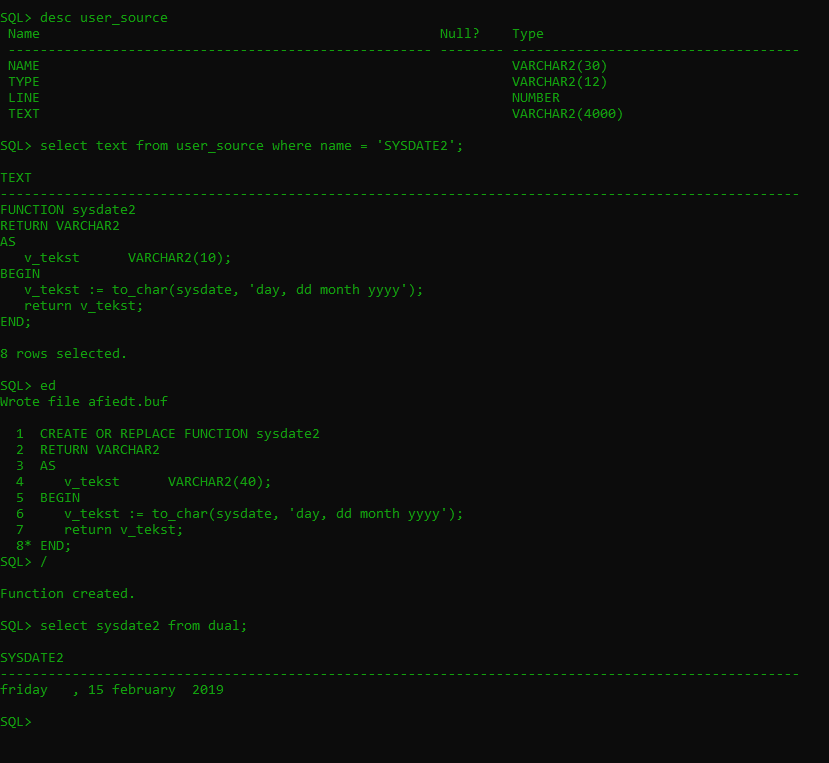
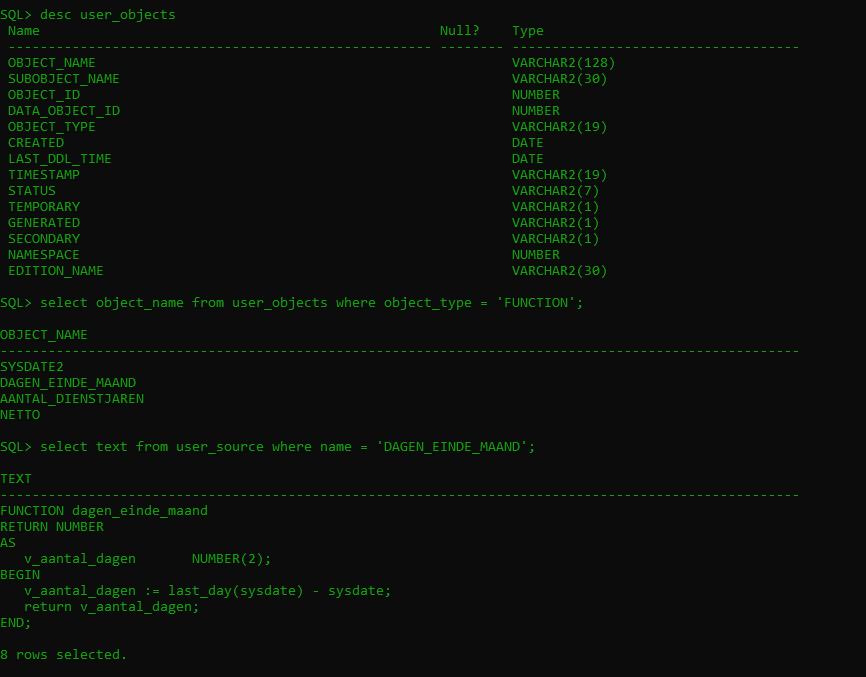
Fuctie aanmaken:

Voor aan te passen als je het niet hebt opgeslaan:





Hoofdstuk 2 oefeningen blackboard

Opgave1:

CREATE OR REPLACE FUNCTION dagen\_einde\_maand

RETURN NUMBER

AS

v\_aantal\_dagen NUMBER(2);

BEGIN

v\_aantal\_dagen := last\_day(sysdate) - sysdate;

return v\_aantal\_dagen;

END;

En daarna het volgende voor het aantal dagen te krijgen

select dagen\_einde\_maand from dual;

opgave 2:

CREATE OR REPLACE FUNCTION aantal\_dienstjaren

(p\_date IN date)

RETURN NUMBER

AS

BEGIN

RETURN trunc(months\_between(sysdate, p\_date) / 12);

END;

select aantal\_dienstjaren(hire\_date), last\_name from employees;

opgave 3a:

CREATE OR REPLACE FUNCTION netto

(p\_salary employees.salary%type)

RETURN VARCHAR2

AS

BEGIN

RETURN to\_char(p\_salary \* 0.6, '999,999,999.00') || ' euro';

END;

select netto(salary) from employees;

opgave 3b:

CREATE OR REPLACE FUNCTION netto

(p\_bruto employees.salary%type)

RETURN VARCHAR2

AS

v\_netto employees.salary%type;

BEGIN

IF p\_bruto <= 10000 then

v\_netto := p\_bruto \* 0.6;

ELSIF p\_bruto <= 160000 then

v\_netto := 6000 + (p\_bruto - 10000) \* 0.5;

ELSE

v\_netto:= 6000 + 3000 + (p\_bruto - 16000) \* 0.4;

END IF;

RETURN to\_char(v\_netto, '999,999,999.00') || ' euro';

END;

select netto(salary) from employees;

opgave 4b:

CREATE OR REPLACE FUNCTION schrikkeljaar

(p\_jaar NUMBER)

RETURN VARCHAR2

AS

BEGIN

IF mod(p\_jaar , 400) = 0 OR ( mod(p\_jaar, 4) = 0 AND mod(p\_jaar, 100) != 0) THEN

RETURN p\_jaar || ' is een schrikkeljaar';

ELSE

RETURN p\_jaar || ' is geen schrikkeljaar';

END IF;

END;

/

Opgave 5:

CREATE OR REPLACE FUNCTION volgende\_vergadering

RETURN DATE

AS

v\_datum date;

BEGIN

v\_datum := next\_day(last\_day(sysdate), 'monday');

IF to\_char(v\_datum, 'ddmm') = '0101' OR to\_char(v\_datum, 'ddmm') = '0105' THEN

v\_datum := v\_datum + 1;

END IF;

RETURN v\_datum;

END;

select volgende\_vergadering from dual;

Opgave 6:

CREATE OR REPLACE FUNCTION next\_employee\_id

RETURN number

AS

v\_id number(3);

v\_max number(3);

BEGIN

select max(employee\_id)

into v\_max

FROM employees;

v\_id := v\_max + 1;

return v\_id;

END;

Select next\_employee\_id from dual;

Opgave 7:

CREATE OR REPLACE FUNCTION berekening\_bruto\_jaarloon

(p\_emp\_id employees.employee\_id%type)

RETURN number

AS

v\_bruto employees.salary%type;

BEGIN

select(salary \* NVL(commission\_pct + 1,1))\*12

into v\_bruto

from employees

where employee\_id = p\_emp\_id;

return v\_bruto;

END;

select berekening\_bruto\_jaarloon (100) from dual;

Opgave 8:

CREATE OR REPLACE FUNCTION manager\_meeste\_werknemers

RETURN varchar2

AS

v\_voornaam varchar2(40);

v\_naam varchar2(40);

BEGIN

select d.department\_id, count(e.employee\_id)

into v\_depid, v\_countemp

from departments d join employees e

group by d.department\_id

having count(e.employee\_id) = (select max(count(department\_id)) from employees group by department\_id);

select first\_name, last\_name

into v\_voornaam, v\_naam

from departments d join employees e

return v\_voornaam || ' ' || v\_naam;

END;

/

Hoofdstuk 3 oefeningen blackboard

**Oefening 1:**

CREATE OR REPLACE PROCEDURE toon\_laatste\_emp

IS v\_employee\_id employees.employee\_id%type;

v\_employee\_first\_name employees.first\_name%type;

v\_employee\_last\_name employees.last\_name%type;

v\_employee\_hire\_date employees.hire\_date%type;

BEGIN

SELECT employee\_id, first\_name, last\_name, hire\_date

INTO v\_employee\_id, v\_employee\_first\_name, v\_employee\_last\_name, v\_employee\_hire\_date

FROM employees

WHERE hire\_date = (SELECT(MAX(hire\_date)) FROM employees);

DBMS\_OUTPUT.PUT\_LINE(v\_employee\_id || ' ' || v\_employee\_first\_name || ' ' || v\_employee\_last\_name || ' ' || v\_employee\_hire\_date);

END toon\_laatste\_emp;

/

exec toon\_laatste\_emp

**oefening2:**

CREATE OR REPLACE PROCEDURE grootste\_dept

IS v\_dept\_name departments.department\_name%type;

v\_emp\_first\_name employees.first\_name%type;

v\_emp\_last\_name employees.last\_name%type;

v\_emp\_salary employees.salary%type;

BEGIN

SELECT d.department\_name, e.last\_name, e.first\_name, e.salary

INTO v\_dept\_name, v\_emp\_last\_name, v\_emp\_first\_name, v\_emp\_salary

FROM employees e

JOIN departments d on e.department\_id = d.department\_id

WHERE e.department\_id = (SELECT department\_id

FROM employees

GROUP BY department\_id

HAVING count(department\_id) = (SELECT MAX(COUNT(department\_id)) FROM employees GROUP BY department\_id))

AND e.salary = (SELECT MAX(salary)

FROM employees

WHERE department\_id = e.department\_id);

DBMS\_OUTPUT.PUT\_LINE(v\_dept\_name || ' ' || v\_emp\_last\_name || ' ' || v\_emp\_first\_name || ' ' || v\_emp\_salary);

END grootste\_dept;

/

**Na code van slide 17: (anoniem blok)**

DECLARE

v\_aantal number(3);

BEGIN

raise\_salary\_2\_dept('Administration', 5, v\_aantal);

DMBS\_OUTPUT.PUT\_LINE('aangepaste records: ' || v\_aantal);

END;

/

variable b\_aantal number

exec raise\_salary\_2\_dept('Administration', 8, :b\_aantal)

print b\_aantal

desc raise\_salary\_2\_dept

**oefening3:**

CREATE OR REPLACE PROCEDURE minimumlonen

(p\_landnaam IN countries.country\_name%TYPE

, p\_nieuw\_minimumloon IN employees.salary%TYPE)

AS

BEGIN

UPDATE employees

SET salary = p\_nieuw\_minimumloon

WHERE department\_id IN (SELECT e.department\_id

FROM employees e join departments d

on e.department\_id = d.department\_id

join locations l

on d.location\_id = l.location\_id

join countries c

on l.country\_id = c.country\_id

WHERE country\_name = p\_landnaam)

AND salary < p\_nieuw\_minimumloon;

END;

/

exec minimumlonen('Canada', 6250);

**extra opgave:**

CREATE OR REPLACE PROCEDURE testloop

AS

BEGIN

FOR emprec IN ( SELECT department\_id, first\_name, last\_name

FROM employees

WHERE department\_id BETWEEN 10 AND 50

ORDER BY department\_id) LOOP

DBMS\_OUTPUT.PUT\_LINE(emprec.department\_id || ' ' || emprec.last\_name || ' '|| emprec.first\_name);

END LOOP;

END;

exec testloop

**oefening4:**

CREATE OR REPLACE PROCEDURE minimumlonen

(p\_landnaam IN countries.country\_name%TYPE

, p\_nieuw\_minimumloon IN employees.salary%TYPE

, p\_countrows OUT NUMBER)

AS

BEGIN

UPDATE employees

SET salary = p\_nieuw\_minimumloon

WHERE department\_id IN (SELECT e.department\_id

FROM employees e join departments d

on e.department\_id = d.department\_id

join locations l

on d.location\_id = l.location\_id

join countries c

on l.country\_id = c.country\_id

WHERE country\_name = p\_landnaam)

AND salary < p\_nieuw\_minimumloon;

p\_countrows := sql%rowcount;

END;

/

**Oproepen via anoniem block**

**Eerst:** eenmalig: set autoprint on

declare

v\_aantal number;

begin

minimumlonen('Canada', 6270, v\_aantal);

dbms\_output.put\_line(v\_aantal);

end;

/

**Via bind variable:**

variable b\_aantal NUMBER

exec minimumlonen ('Canada', 6280, :b\_aantal)

Hoofdstuk 4 oefeningen blackboard

bus -> before update statement

**oefening 1:**

CREATE OR REPLACE TRIGGER bus\_jobs

BEFORE UPDATE

ON jobs

BEGIN

IF USER in ('STUDENT', 'BEZOEKER') THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Je hebt onvoldoende rechten om deze actie uit te voeren!');

ELSE

DBMS\_OUTPUT.PUT\_LINE('ben je zeker? anders doe ROLLBACK');

END IF;

END;

/

update jobs set job\_title='qksdjfqdksjfllkqdf';

**oefening 2:**

CREATE table log\_history(

log\_user varchar2(20),

log\_tijdstip date,

log\_actie varchar2(20))

CREATE OR REPLACE TRIGGER auids\_jobhis

AFTER UPDATE OR INSERT OR DELETE

ON job\_history

DECLARE

v\_actie varchar2(20);

BEGIN

IF UPDATING THEN

v\_actie := 'update';

ELSIF DELETING THEN

v\_actie := 'delete';

ELSIF INSERTING THEN

v\_actie := 'insert';

END IF;

INSERT INTO log\_history

VALUES(USER, sysdate, v\_actie);

END;

/

**oefening 3:**

CREATE OR REPLACE TRIGGER buid

BEFORE UPDATE OR INSERT OR DELETE of salary

ON employees

BEGIN

IF UPDATING AND NOT(TO\_CHAR(sysdate, 'hh24') >= 9 AND TO\_CHAR(sysdate, 'hh24') < 17 AND TO\_CHAR(sysdate, 'DAY') = 'MONDAY') THEN

RAISE\_APPLICATION\_ERROR(-20000,'Je mag enkel maandag tussen 9u00 - 17u00 aanpassen');

END IF;

IF DELETING OR INSERTING AND TO\_CHAR(sysdate, 'DAY') IN ('SATURDAY', 'SUNDAY') THEN

RAISE\_APPLICATION\_ERROR(-20001,'Je kan in het weekend geen employees verwijderen of toevoegen');

END IF;

END;

/