

## Laboration 4

```
--Tabeller
CREATE TABLE bilägare(
pnr VARCHAR2(13) PRIMARY KEY,
fnamn VARCHAR2(20),
enamn VARCHAR2(20),
bor_i VARCHAR2(20),
jobbar_i VARCHAR2(20));

CREATE TABLE fordon(
regnr VARCHAR2(6) PRIMARY KEY,
pnr REFERENCES bilägare(pnr),
tillverkare VARCHAR2(20),
modell VARCHAR2(20),
årsmodell NUMBER(4),
hk NUMBER(4),
datum DATE);

-----Insert Data-----
--Insert into Bilägare
INSERT INTO bilägare VALUES('19490321-7899','hans','rosenboll','borlänge','falun');
INSERT INTO bilägare VALUES('19540201-4428','tomas','kvist','gagnef','borlänge');
INSERT INTO bilägare VALUES('19650823-7999','roger','nyberg','borlänge','falun');
INSERT INTO bilägare VALUES('19710601-7799','lena','malm','borlänge','falun');
INSERT INTO bilägare VALUES('19690321-7898','ollas','bullas','falun','borlänge');

INSERT INTO bilägare VALUES('19590421-7199','tåmmy','dåmert','borlänge','falun');
INSERT INTO bilägare VALUES('19610321-4299','rollf','ekengren','borlänge','falun');
INSERT INTO bilägare VALUES('19810321-7199','maria','stjärnkvist','borlänge','falun');
INSERT INTO bilägare VALUES('19720721-7899','leyla','errstraid','borlänge','falun');
INSERT INTO bilägare VALUES('19380321-7799','arne','möller','borlänge','falun');

--Insert into Fordon
INSERT INTO fordon VALUES('ase456','19490321-7899','volvo','945',1998,160,to_date('2003-08-11','YYYY-MM-DD'));
INSERT INTO fordon VALUES('ptg889','19490321-7899','fiat','excel',1991,287,to_date('1998-05-19','YYYY-MM-DD'));
INSERT INTO fordon VALUES('bon666','19540201-4428','john deere','gris',1967,48,to_date('1989-06-28','YYYY-MM-DD'));

INSERT INTO fordon VALUES('rog589','19650823-7999','saab','900 talladega',1997,205,to_date('2003-05-11','YYYY-MM-DD'));
INSERT INTO fordon VALUES('ert456','19710601-7799','volvo','850',1997,150,to_date('2001-07-11','YYYY-MM-DD'));
INSERT INTO fordon VALUES('ola774','19690321-7898','mb','e420',1998,285,to_date('2000-08-11','YYYY-MM-DD'));

INSERT INTO fordon VALUES('thf345','19590421-7199','opel','kapitän',1968,105,to_date('1991-06-11','YYYY-MM-DD'));
INSERT INTO fordon VALUES('dde411','19610321-4299','saab','9000 aero',1998,225,to_date('2000-07-28','YYYY-MM-DD'));
INSERT INTO fordon VALUES('ser478','19810321-7199','audi','tt',2003,247,to_date('2004-07-05','YYYY-MM-DD'));

INSERT INTO fordon VALUES('fgt147','19720721-7899','volvo','66',1981,62,to_date('2003-05-11','YYYY-MM-DD'));
INSERT INTO fordon VALUES('tau444','19380321-7799','ford','taunus',1973,95,to_date('1975-08-23','YYYY-MM-DD'));
INSERT INTO fordon VALUES('pot333','19540201-4428','volvo','745',1989,93,to_date('1996-01-11','YYYY-MM-DD'));
COMMIT;
```

--Insert End

### Uppgift 1

Skapa ett anonymt PLSQL-block som skriver ut regnr, tillverkare och modell för det fordon som har ägare med personnummer (pnr) = '19650823-7999'. Du skall lösa uppgiften genom att deklarera tre lokala variabler. Variablerna skall vara av dynamisk datatyp, sk [anchored declaration](#). Du får **inte** lösa uppgiften med någon form av **cursor**.

```
declare
v_regnr fordon.regnr%type;
v_tillverkare fordon.tillverkare%type;
v_modell fordon.modell%type;

begin

select regnr, tillverkare, modell
into v_regnr, v_tillverkare, v_modell
from fordon
where pnr = '19650823-7999';
dbms_output.put_line('Regnr: '||v_regnr|| chr(10) || 'Tillverkare: '||v_tillverkare||
chr(10) || 'Modell: '||v_modell);
end;
```

Svar:

Regnr: ROG589  
Tillverkare: saab  
Modell: 900 talladega

### Uppgift 2

Återanvänd din kod från uppgift 1. **Ändra villkoret till** pnr = '19540201-4428', lägg till en felhanteringsdel där du fångar undantaget OTHERS och skriver ut en textsträng som talar om att något gick fel.

```
declare
v_regnr fordon.regnr%type;
v_tillverkare fordon.tillverkare%type;
v_modell fordon.modell%type;

begin

select regnr, tillverkare, modell
into v_regnr, v_tillverkare, v_modell
from fordon
where pnr = '19540201-4428';
dbms_output.put_line('regnr: '||v_regnr||','||' tillverkare: '||v_tillverkare||','||'
modell: '||v_modell);
exception
when others then
dbms_output.put_line('Något blev fel!');
end;
```

Svar:

Något blev fel!

### Uppgift 3

Återanvänd din kod från uppgift 2. Komplettera koden med ytterligare två variabler. Dessa skall lagra information från `SQLCODE` och `SQLERRM`. Förändra felhanteringsdelen i uppgift 2 till att skriva ut värdet av felkoden och felmeddelandet.

#### Varför uppstår ett fel i uppgifterna 2 och 3?

Felet uppstår i uppgift 2 och 3 för att det finns fler än ett fordon kopplad till personnumret '19540201-4428'.

Jag kan just nu med koden jag har återanvänt bara skriva ut en enda rad, med **Singleton select**. Om databasen hittar fler matchningar med personnumret som är kopplad till flera fordon så kommer exception hanteraren att kallas för att inte krascha databasen, men också ange vad för fel som har uppstått så att jag som programmerare lättare kan spåra felet och optimera koden. Jag behöver använda cursor för att kunna returnera fler än en rad.

Rätt utskrift:

```
declare
v_regnr fordon.regnr%type;
v_tillverkare fordon.tillverkare%type;
v_modell fordon.modell%type;

begin

select regnr, tillverkare, modell
into v_regnr, v_tillverkare, v_modell
from fordon
where pnr = '19540201-4428';
dbms_output.put_line('regnr: '||v_regnr||','||' tillverkare: '||v_tillverkare||','||'
modell: '||v_modell);
exception
when others then
dbms_output.put_line('Följande blev fel: ');
dbms_output.new_line;
dbms_output.put_line('Felkod: ' || sqlcode || chr(10) || 'Felmeddelande: ' ||
sqlerrm);
end;
```

Svar:

Följande blev fel:

Felkod: -1422

Felmeddelande: ORA-01422: Exakt hämtning returnerar fler än angivet antal rader

#### Uppgift 4

Skapa ett anonymt PLSQL-block som skriver ut förnamn, efternamn och ålder i år med en decimal för alla personer i tabellen bilägare, i mitt exempel var åldern rätt i september 2014. Använd valfri cursor, explicit eller implicit, och valfri loop för att lösa uppgiften. För-och efternamn skall skrivas ut med stor begynnelsebokstav. Tips: du kan använda funktionen `substr()` för att plocka ut del av en sträng. Du kommer endast att behöva de första åtta tecknen i kolumnen pnr. Tips: om du tar sysdate - ett datum kan du räkna ut rätt ålder.

**OBS!** Eftersom vi använder sysdate så får ni en annan ålder i er utskrift.

```
declare
ålderÅr integer;
ålderMånad integer;

födelseÅr integer;
födelseMånad integer;

nuvarandeÅr integer;
nuvarandeMånad integer;

cursor c_bilägare is select  bä.fnamn, bä.enamn, bä.pnr
from bilägare bä;

rec c_bilägare%rowtype;

begin

if not c_bilägare%isopen then
    open c_bilägare;
end if;
loop
    fetch c_bilägare
    into rec;
    exit when c_bilägare%notfound;

    nuvarandeÅr := extract(year from sysdate);
    nuvarandeMånad := extract(month from sysdate);

    födelseÅr := substr(rec.pnr, 0, LENGTH(rec.pnr) - 9);
    födelseMånad := substr(rec.pnr, 0, LENGTH(rec.pnr) - 7);
    födelseMånad := substr(födelseMånad,5);

    ålderÅr := nuvarandeÅr-födelseÅr;
    ålderMånad := nuvarandeMånad - födelseMånad;

    if(ålderMånad < 0) then
        ålderMånad := 12 + ålderMånad;
        ålderÅr := ålderÅr - 1;
    end if;

    dbms_output.put_line(INITCAP(rec.fnamn)|| ', ' || INITCAP(rec.enamn) || ', ' ||
ålderÅr || '.' || ålderMånad || ' år. ');
    end loop;
close c_bilägare;
end;
```

#### Utskrift:

Hans, Rosenboll, 65,6 år.  
Tomas, Kvist, 60,7 år.  
Roger, Nyberg, 49,1 år.  
Lena, Malm, 43,4 år.  
Ollas, Bullas, 45,6 år.  
Tåmmy, Dåmert, 55,5 år.  
Rollf, Ekengren, 53,6 år.

Elias Posluk  
h16elipo  
GIK23M

Maria, Stjärnkvist, 33,5 år.  
Leyla, Errstraid, 42,2 år.  
Arne, Möller, 76,6 år.

## Uppgift 5

Skapa ett anonymt PLSQL-block som skriver ut personnummer, förnamn, efternamn samt hur många bilar respektiver person äger. **Observera** att du måste ta hänsyn till singular och plural. Det heter **en bil** respektive **flera bilar** på korrekt svenska. Lös uppgiften genom att använda valfri cursor och loop.

Utskrift:

**declare**

```
antal_bilar int;  
sträng_bilar string(5);
```

```
cursor c_bilägare is select co.pnr, co.fnamn, co.enamn, count(f.regnr) as antal_bilar  
from bilägare co, fordon f  
where co.pnr = f.pnr  
group by co.pnr, co.fnamn, co.enamn  
order by pnr asc;
```

```
rec c_bilägare%rowtype;
```

**begin**

```
  if not c_bilägare%isopen then  
    open c_bilägare;  
  end if;
```

**loop**

```
    fetch c_bilägare  
    into rec;  
    exit when c_bilägare%notfound;
```

```
      if rec.antal_bilar = 1  
      then  
        sträng_bilar := 'bil';  
      elsif rec.antal_bilar > 1  
      then  
        sträng_bilar := 'bilar';  
      end if;
```

```
      dbms_output.put_line( rec.pnr || ', ' || INITCAP(rec.fnamn) || ', ' ||  
INITCAP(rec.enamn) || ' äger: ' || rec.antal_bilar || ' ' || sträng_bilar);
```

```
    end loop;
```

```
  close c_bilägare;
```

**end**;

Utskrift:

```
19380321-7799, arne, möller, owns: 1 bil  
19490321-7899, hans, rosenboll, owns: 2 bilar  
19540201-4428, tomas, kvist, owns: 2 bilar  
19590421-7199, tåmmy, dāmert, owns: 1 bil  
19610321-4299, rollf, ekengren, owns: 1 bil  
19650823-7999, roger, nyberg, owns: 1 bil  
19690321-7898, ollas, bullas, owns: 1 bil  
19710601-7799, lena, malm, owns: 1 bil  
19720721-7899, leyla, errstraid, owns: 1 bil  
19810321-7199, maria, stjärnkvist, owns: 1 bil
```

## Uppgift 6

Skapa ett anonymt PLSQL-block som kopierar över data från tabellerna bilägare och fordon till tabellen fartdåre. Ta bara med data där fordonets motorstyrka är högre än 200 hästkrafter. Kolumnen **fordon.hk** innehåller information om antal hästkrafter. Lös uppgiften genom att deklarera en **explicit cursor**. Programmet skall skriva ut ett meddelande om att kopieringen är klar. Transaktionerna skall ha gjort **commit** i programmet **innan** utskriften **inträffar**.

```
set serveroutput on

declare
cursor c_fartdåre is select bä.fnamn, bä.enamn, bä.pnr, f.regnr, f.tillverkare,
f.modell
from bilägare bä, fordon f
where bä.pnr = f.pnr
and f.hk > 200;

v_rec c_fartdåre%rowtype;

begin

  for v_rec in c_fartdåre loop

    insert into fartdåre(fnamn, enamn, pnr, regnr, tillverkare, modell)
    values(v_rec.fnamn, v_rec.enamn, v_rec.pnr, v_rec.regnr, v_rec.tillverkare,
v_rec.modell);

  end loop;

commit;

  dbms_output.put_line('Kopieringen är klar!');
end;

select *
from fartdåre;
```

## Utskrift:

Kopieringen är klar!

PNR	FNAMN	ENAMN	REGNR	TILLVERKARE	MODELL
19490321-7899	hans	rosenboll	ptg889	fiat	excel
19650823-7999	roger	nyberg	rog589	saab	900 talladega
19690321-7898	ollas	bullas	ola774	mb	e420
19610321-4299	rollf	ekengren	dde411	saab	9000 aero
19810321-7199	maria	stjärnkvist	ser478	audi	tt