Funcions:

Realitza les següents funcions:

1. Se li passa com a paràmetre l'identificador d'un pilot i retorna quants punts te el pilot.

create or replace function ex1\_fun (identificador\_pilot int)

returns int

language

plpgsql

as

$$

declare

puntspilot integer:= 0;

begin

if(identificador\_pilot > 21) --Exception

then

raise notice 'No s ha trobat el codi pilot %', identificador\_pilot;

end if;

select DISTINCT(SUM(punts))

from punts pun INNER JOIN resultats re on pun.posicio = re.posicio

INNER JOIN pilots p on re.pilot = p.clave\_pilot

INTO puntspilot

WHERE clave\_pilot = identificador\_pilot;

return puntspilot;

end;

$$

select ex1\_fun(2)

2. Se li passa com a paràmetre l’identificador d’un pilot i retorna el % (valor amb 2 decimals) de carreres en les que ha puntuat.

select ex2\_func(1)

create or replace function repas2 (identificador\_pilot int)

returns decimal

language plpgsql

as

$$

declare

inf record;

puntuades decimal:= 0;

contador decimal:= 0;

total decimal:= 0;

begin

FOR inf IN (SELECT \*

FROM punts INNER JOIN resultats

on resultats.posicio = punts.posicio

INNER JOIN pilots ON clave\_pilot = pilot

WHERE clave\_pilot = identificador\_pilot)

LOOP inf.punts = inf.punts;

if inf.punts > 0

then

puntuades = puntuades + 1;

end if;

contador = contador +1;

end loop;

total = ROUND(AVG(COUNT(puntuades)\*100/contador,2));

return total;

end;

$$

3. Se li passa com a paràmetre l'identificador de la nació i retorna la quantitat de punts que han obtingut pilots d'eixa nacionalitat.

select ex3funcio('AUS')

create or replace function ex3funcio (identificador\_nacio varchar(3))

returns int

language plpgsql

as

$$

declare

puntsTotals integer:= 0;

begin

select SUM(punts.punts)

from punts JOIN resultats on punts.posicio = resultats.posicio JOIN pilots on clave\_pilot = resultats.pilot JOIN nacions on pilots.nacio = nacions.clave\_nacio

into puntsTotals

WHERE clave\_nacio = identificador\_nacio;

return puntsTotals;

end;

$$

4. Se li passa com a paràmetre l'identificador d'un equip i retorna els punts d'eixe equip.

select ex4\_fun(1)

create or replace function ex4\_fun (identificador\_equip integer)

returns int

language plpgsql

as

$$

declare

puntsTotals integer:= 0;

begin

select SUM(punts.punts)

from punts JOIN resultats on punts.posicio = resultats.posicio JOIN pilots on clave\_pilot = resultats.pilot JOIN nacions on pilots.nacio = nacions.clave\_nacio

into puntsTotals

WHERE pilots.equip = identificador\_equip;

return puntsTotals;

end;

$$

Procediments:

1. El procediment ha de guardar en la taula pilots el codi del pilot que es passa per paràmetre. També rep el número del pilot per paràmetre per identificar quin és el pilot que volem modificar. El codi serà les 3 primeres lletres del cognom. Per exemple, Kimi Räikkönnen porta el número 7 i volem guardar el codi RAI.

alter table pilots

add column codi varchar(3)

create or replace procedure ex1\_p (codi\_pilot integer, apodo varchar(3))

language plpgsql

as

$$

declare

vaux int;

begin

UPDATE pilots

set codi = apodo

where codi\_pilot = clave\_pilot;

raise notice 'Has afegit %', apodo;

SELECT count(\*)

into vaux

from pilots

where clave\_pilot=codi\_pilot;

if vaux <> 1 then

raise exception '%','Este pilot no existeix'

exception

when no\_data\_found

then

raise info 'No sha trobat el pilot';

when others

then

raise exception 'Error no especificat';

end;

$$

create or replace procedure ex1\_p (codi\_pilot integer, apodo varchar(3))

language plpgsql

as

$$

declare

vaux int;

begin

UPDATE pilots

set codi = apodo

where codi\_pilot = clave\_pilot;

raise notice 'Has afegit %', apodo;

SELECT count(\*)

into vaux

from pilots

where clave\_pilot=codi\_pilot;

if vaux <> 1 then

raise exception '%','Este pilot no existeix'

exception

when sqlstate 'POOO1'

then

raise info '%', SQLERRM;

when no\_data\_found

then

raise info 'No sha trobat el pilot';

when others

then

raise exception 'Error no especificat';

end;

$$

call ex1\_p (7, 'RAI')

2. Rep com a paràmetre l’identificador d’una nació i com a segon paràmetre un indicador si volem trobar circuits o pilots. Mostrarà un missatge indicant els circuits o els pilots de la nació introduïda, en funció del 2on paràmetre que li hem passat.

call ex2\_p('AUS','Pilot')

CREATE or replace procedure ex2\_p(clave\_nacio varchar,pilotOcircuit varchar)

language plpgsql

as

$$

declare

identificador varchar;

begin

if pilotOcircuit = 'Pilot' then

SELECT nom

into identificador

FROM pilots;

raise notice '%',identificador;

elsif pilotOcircuit = 'Circuit' then

SELECT nom

into identificador

FROM circuits;

raise notice '%',identificador;

end if;

exception

when no\_data\_found then

raise exception 'No sha trobat el pilot / circuit';

end;

$$

NO HEM FUNCIONA:

CREATE or replace procedure ex2\_p(clave\_nacio varchar,pilotOcircuit varchar)

language plpgsql

as

$$

declare

identificador varchar;

begin

if pilotOcircuit = 'Pilot' then

SELECT COUNT(clave\_pilot)

into identificador

FROM pilots;

raise notice '%',identificador;

elsif pilotOcircuit = 'Circuit' then

SELECT COUNT(clave\_circuit)

into identificador

FROM circuits;

raise notice '%',identificador;

end if;

end;

$$

3. Se li passa com a paràmetre el codi d'un pilot i retorna un missatge indicant si és millor que el seu company, o bé si és pitjor, depenent de si ha aconseguit més o menys punts que el seu company d'equip.

call ex3\_pro(3)

create or replace procedure ex3\_pro (identificador\_pilot integer)

language plpgsql

as

$$

declare

pilot1 integer:= ex1\_fun(identificador\_pilot);

pilot2 integer:= 0;

equip integer:= 0;

begin

SELECT pilots.equip

FROM pilots

into equip

WHERE identificador\_pilot = clave\_pilot;

SELECT punts

into pilot2

FROM punts INNER JOIN pilots

on clave\_pilot = pilot1 INNER JOIN equips on pilots.equip = clave\_equip;

SELECT punts.punts

FROM punts INNER JOIN pilots

on clave\_pilot = pilot2 INNER JOIN equips on pilots.equip = clave\_equip

into pilot1

WHERE clave\_pilot != pilot2;

if (pilot1 < pilot2)

then

raise notice 'El pilot1 es millor que el pilot2 %', punts.punts;

elsif pilot2 < pilot1

then

raise notice 'El pilot2 es millor que el pilot1 %',punts.punts;

else

raise notice 'Els dos pilots tenen la mateixa puntuacio %',punts.punts;

end if;

exception

when no\_data\_found then

raise exception'No s ha trobat el pilot %', identificador\_pilot;

end;

$$

4. Se li passa com a paràmetre el nom d'una escuderia i retorna una cadena indicant quantes vegades ha fet pòdium. Per exemple: Ferrari ha quedat 3 vegades en primer lloc, 7 en segon i 9 vegades en tercer lloc.

call ex4\_p('Ferrari')

create or replace procedure ex4\_p (escuderia varchar)

language plpgsql

as $$

declare

inf record;

contador int:= 0;

primer int:= 0;

segon int:= 0;

tercer int:= 0;

resto int:= 0;

begin

for inf IN (SELECT \* FROM equips INNER JOIN pilots ON clave\_equip = equip INNER JOIN resultats ON clave\_pilot = pilot

WHERE nomconstructor = escuderia)

loop inf.posicio:= inf.posicio;

IF inf.posicio = 1

then primer = primer + 1;

elsif inf.posicio = 2

then segon = segon + 1;

elsif inf.posicio = 3

then tercer = tercer + 1;

else

resto = resto + 1;

end if;

contador = contador + 1;

end loop;

raise notice '1r %', primer;

raise notice '2n %', segon;

raise notice '3r %', tercer;

raise notice 'altres %', resto;

exception

when no\_data\_found then

raise exception'No s ha trobat la escuderia %', escuderia;

end;

$$

Cursor:

do $$

declare

vble\_registre record; --variable del cursor

cur\_f1 cursor for select \* from pilots --crear cursor

begin

open cur\_f1; --obrir el cursor

fetch cur\_f1 into vble\_registre; --almacenes el registre dintre de la variable amb el fetch el qual serveix com a contador si vols ficar-ne mes d’1.

raise notice 'Es mostraran els noms pilots, % juntament amb la seva clau, %', vble\_registre.nom, vble\_registre.clave\_pilot;

close cur\_f1;

end; $$