

1.-

Mostrar los datos de la receta que tiene más estrellas, junto con el nombre de su chef, en un formato listado para imprimir, con título y forma visual agradable

set serveroutput on

declare

v_estrellas evaluacion.estrellas%type;

cursor c2(var_estrellas evaluacion.estrellas%type) is

select distinct recetas.id_receta, recetas.nombre, recetas.calorias,

recetas.comensales, recetas.categoria, chefs.nombre as chef

from evaluacion

join recetas on evaluacion.id_receta = recetas.id_receta

join autoria on evaluacion.id_receta = autoria.id_receta

join chefs on autoria.id_chef = chefs.id_chef

where evaluacion.estrellas = var_estrellas;

begin

select max(estrellas) into v_estrellas

from evaluacion;

dbms_output.put_line('-----LAS MEJORES RECETAS-----');

dbms_output.put_line(' Id receta - Nombre receta - Calorias - Comensales - Categoria - Chef');

dbms_output.put_line('-----');

for c2_datos in c2(v_estrellas) loop

dbms_output.put_line(Lpad(c2_datos.id_receta,10) || ' - '

||Rpad(c2_datos.nombre,13) || ' - ' || Lpad(c2_datos.calorias,8) || ' - '

||Lpad(c2_datos.comensales,10) || ' - ' || Rpad(c2_datos.categoria,9) || ' - '

|| c2_datos.chef);

end loop;

end;

/

-----LAS MEJORES RECETAS-----					
Id receta	Nombre receta	Calorias	Comensales	Categoria	Chef
3	Paella	500	4	Arroces	Mari
4	Carne tierna	400	3	Carnes	Juan

2.-

Mostrar las 3 peores recetas, con sus ingredientes y nombre de chef, en un formato listado para imprimir, con título y forma visual agradable.

```
set serveroutput on
```

```
declare
```

```
cursor c1 is
```

```
select recetas.id_receta,evaluacion.estrellas, recetas.nombre, chefs.nombre as chef
from  evaluacion join recetas on evaluacion.id_receta = recetas.id_receta
join autoria on evaluacion.id_receta = autoria.id_receta
join chefs on autoria.id_chef = chefs.id_chef
order by evaluacion.estrellas;
```

```
cursor c2(n_id contiene.id_receta%type) is
```

```
select ingredientes.nombre
from ingredientes join contiene on contiene.id_ingredientes = ingredientes.id_ing
where contiene.id_receta = n_id;
```

```
i number(4):=1;
```

```
c1_datos c1%rowtype;
```

```
begin
```

```
dbms_output.put_line('-----LAS PEORES RECETAS-----');
```

```
open c1;
```

```
loop
```

```
fetch c1 into c1_datos;
```

```
exit when i >3;
```

```
dbms_output.put_line(' Id receta - Nombre receta - Estrellas - Chef');
```

```
dbms_output.put_line('-----');
```

```
dbms_output.put_line(Lpad(c1_datos.id_receta,10) || ' - '
```

```
|| Lpad(c1_datos.nombre,13) || ' - '
```

```
|| Lpad(c1_datos.estrellas,9) || ' - '
```

```
|| c1_datos.chef);
```

```
dbms_output.put_line('ingredientes: ');
```

```
for c2_datos in c2(c1_datos.id_receta) loop
```

```
dbms_output.put_line(c2_datos.nombre);
```

```
end loop;
```

```
dbms_output.put_line("");
```

```
i := i+1;
```

```
end loop;
```

```
close c1;
```

```
end;
```

```

/
-----LAS PEORES RECETAS-----
Id receta - Nombre receta - Estrellas - Chef
-----
          2 -          Guisado -          2 - Antone
ingredientes:
tomate
carne vacuna

Id receta - Nombre receta - Estrellas - Chef
-----
          8 -          Asado -          3 - Antone
ingredientes:
tomate
carne vacuna

```

3.-

Pedir al usuario una categoría, y que se le muestren todas las recetas de esa categoría.

set serveroutput on

ACCEPT v_teclado PROMPT 'Dime la categoría de las recetas que quieres ver: '

declare

```

cursor c1 is
select nombre
from recetas
where categoria='&v_teclado';

```

begin

```

dbms_output.put_line('CATEGORIA: ');
dbms_output.put_line('&v_teclado');
dbms_output.put_line('RECETAS: ');
for c1_datos in c1 loop
dbms_output.put_line(c1_datos.nombre);
end loop;

```

exception

When others then

```

dbms_output.put_line('Esa categoria no existe o esta mal escrita');
end;

```

/

```

CATEGORIA:
Carnes
RECETAS:
Guisado
Carne tierna
Carne porcina especial
Asado

```

4.-

Listar para cada CATEGORÍA las recetas que hay con la cantidad de ingredientes y unidad.

```
set serveroutput on
```

```
declare
```

```
cursor c1 is
```

```
select categoria, nombre, id_receta
```

```
from recetas;
```

```
cursor c2(v_rec contiene.id_receta%type) is
```

```
select ingredientes.nombre, contiene.cantidad, contiene.unidad
```

```
from ingredientes join contiene on ingredientes.id_ing = contiene.id_ingrediente
```

```
where contiene.id_receta = v_rec;
```

```
begin
```

```
dbms_output.put_line('Listado de Recetas por Categoria');
```

```
dbms_output.put_line('-----');
```

```
for c1_datos in c1 loop
```

```
dbms_output.put_line('CATEGORIA: '|| c1_datos.categoria || ' RECETA: '
```

```
||c1_datos.nombre);
```

```
dbms_output.put_line(' INGREDIENTES - CANTIDAD - UNIDAD');
```

```
for c2_datos in c2(c1_datos.id_receta) loop
```

```
dbms_output.put_line(Lpad(c2_datos.nombre,15) || ' - '|| Lpad(c2_datos.cantidad,8) || ' - '||  
c2_datos.unidad);
```

```
end loop;
```

```
dbms_output.put_line(' ');
```

```
end loop;
```

```
end;
```

```
/
```

```
Listado de Recetas por Categoria
```

```
-----  
CATEGORIA: Huevos RECETA: Omelet  
INGREDIENTES - CANTIDAD - UNIDAD  
huevo - 200 - gr  
leche - 100 - ml
```

```
CATEGORIA: Carnes RECETA: Guisado  
INGREDIENTES - CANTIDAD - UNIDAD  
carne vacuna - 50 - gr  
tomate - 70 - gr
```

```
CATEGORIA: Arroces RECETA: Paella  
INGREDIENTES - CANTIDAD - UNIDAD  
pollo - 100 - ml  
tomate - 150 - ml  
arroz - 500 - gr
```

5.-

Preguntar al usuario EL NOMBRE DE UN CHEF Y MOSTRAR:

Una cabecera: NOMBRE CHEF y su total de RECETAS

Para cada chef:

Numerar las recetas y mostrar el nombre de la RECETA y sus

ingredientes (nombre y tipo)Para cada chef, si tiene menos de tres

recetas mostrar mensaje 'tienes que ser más creativo'; y si tiene más de 3 recetas: 'Puedes pedir aumento de salario'.

El formato del listado debe ser cuidado , para poderse imprimir podéis poner líneas decorativas o separadoras tipo '====' o '-----'

```
set serveroutput on
accept v_teclado prompt 'Inserta el nombre del chef: '
declare
cursor c1(v_tec chefs.nombre%type)
is select recetas.id_receta, recetas.nombre as nom_receta
from chefs
join autoria on chefs.id_chef = autoria.id_chef
join recetas on autoria.id_receta = recetas.id_receta
where chefs.nombre = v_tec;

cursor c2(v_id contiene.id_receta%type)
is select ingredientes.nombre, ingredientes.tipo
from contiene
join ingredientes on contiene.id_ingredient = ingredientes.id_ing
where contiene.id_receta = v_id;

chef_rec number(2);
c1_datos c1%rowtype;
num_receta number(2):=1;
num_ing number(3):=1;

begin
select count(autoria.id_receta) into chef_rec
from chefs
join autoria on chefs.id_chef = autoria.id_chef
where chefs.nombre = '&v_teclado';

dbms_output.put_line(' =====');
dbms_output.put_line('|| NOMBRE DEL CHEF - TOTAL RECETAS ||');
dbms_output.put_line('|| '||Lpad('&v_teclado',15)||' - '||Lpad(chef_rec,7)||Lpad(' ||',9));
dbms_output.put_line(' =====');

open c1('&v_teclado');
loop fetch c1 into c1_datos;
exit when c1%notfound;
```

```

dbms_output.put_line(num_receta||'º Receta: '||c1_datos.nom_receta);
dbms_output.put_line('Ingredientes:');
for c2_datos in c2(c1_datos.id_receta)
loop
dbms_output.put_line(num_ing||'.- '||c2_datos.nombre||' Tipo: '||c2_datos.tipo);
num_ing:= num_ing+1;
end loop;
num_ing:=1;
num_receta := num_receta+1;
dbms_output.put_line('-----');
end loop;
if num_receta>3
then dbms_output.put_line('Puedes pedir un aumento de salario');
end if;
if num_receta<=3 then
dbms_output.put_line('tienes que ser mas creativo');
end if;
close c1;
exception
when no_data_found then
dbms_output.put_line('Ese chef no existe!!');
when others then
dbms_output.put_line('Vuelve a ejecutar y asegurate de poner bien el nombre del chef!!');
end;
/

```

```

=====
|| NOMBRE DEL CHEF - TOTAL RECETAS ||
||      Antone -      4      ||
=====
1º Receta: Omelet
Ingredientes:
1.- huevo Tipo: huevo
2.- leche Tipo: lacteo
-----
2º Receta: Guisado
Ingredientes:
1.- tomate Tipo: vegetal
2.- carne vacuna Tipo: vacuno
-----
3º Receta: Carne porcina especial
Ingredientes:
1.- tomate Tipo: vegetal
2.- carne porcina Tipo: porcino
-----
4º Receta: Asado
Ingredientes:
1.- tomate Tipo: vegetal
2.- carne vacuna Tipo: vacuno
-----
Puedes pedir un aumento de salario

```

6. Insertar un nuevo ingrediente, con un id que sea una unidad más del último id y resto de los datos pregunte al usuario.

```

set serveroutput on
accept v_nombre prompt 'Dime el ingrediente'
accept v_tipo prompt 'Dime el tipo'
declare

```

```

n_ing number(4);

```

```

begin
  select max(id_ing) into n_ing
  from ingredientes;
  n_ing:= n_ing+1;

  INSERT INTO ingredientes
  VALUES(n_ing,'&v_nombre','&v_tipo');
end;
/

```

7.(Trigger) Realizar un log de las altas/bajas/modificaciones que se han hecho en la tabla INGREDIENTES, que conste el usuario y la fecha y hora de la modificación.

```

drop table auditoria;
create table auditoria(
estado varchar2 (20),
nombre varchar2 (30),
usuario varchar2 (20),
fecha date,
hora varchar2 (10)
);

```

```

create or replace trigger log_auditoria
before insert or update or delete of nombre on ingredientes
for each row

```

```

begin
  if inserting then
    insert into auditoria
    values('alta', :new.nombre , (select user from dual), sysdate, (select
to_char(current_timestamp,'HH-MI-SS') from dual));
    end if;

  if deleting then
    insert into auditoria
    values('baja',:old.nombre, (select user from dual), sysdate, (select
to_char(current_timestamp,'HH-MI-SS') from dual));
    end if;

  if updating then
    insert into auditoria
    values('modificacion',:old.nombre, (select user from dual), sysdate, (select
to_char(current_timestamp,'HH-MI-SS') from dual));
    end if;

```

```
end;  
/
```

	ESTADO	NOMBRE	USUARIO	FECHA	HORA
1	alta	yuca	HR	11-MAY-20	10-09-10
2	modificacion	yuca	HR	11-MAY-20	10-09-10
3	baja	camote	HR	11-MAY-20	10-09-10

9.(Trigger) Realizar un trigger, que bloquee los Inserts e update's en la tabla de evaluación para que no se pueda hacer evaluaciones para uno de los cocineros que elijais.

```
create or replace trigger block_evaluacion  
before insert or update of id_receta on evaluacion  
for each row  
declare  
id_rec number(4);  
  
begin  
SELECT count(recetas.id_receta) into id_rec  
  FROM recetas  
  join autoria on autoria.id_receta=recetas.id_receta  
  join chefs on chefs.id_chef=autoria.id_chef  
  where chefs.nombre='Antone' and :new.id_receta=recetas.id_receta;  
  
  if inserting then  
    if id_rec >0 then  
      raise_application_error(-20001, 'No puedes modificar la evaluacion de las recetas del  
chef Antone');  
    end if;  
  end if;  
  
  if updating then  
    if id_rec >0 then  
      raise_application_error(-20001, 'No puedes modificar la evaluacion de las recetas del  
chef Antone');  
    end if;  
  end if;  
end;  
/
```



```
Error starting at line 31 in command:  
insert into evaluacion
```

```
values(3,1,4)  
Error report:  
SQL Error: ORA-20001: No puedes modificar la evaluacion de las recetas del chef A  
ORA-06512: at "HR.BLOCK_EVALUACION", line 13  
ORA-04088: error during execution of trigger 'HR.BLOCK_EVALUACION'
```

10.Crear una función, un procedimiento y un trigger para la tabla que os inventasteis (o una nueva).

```
create table restaurante(  
id_rest number(4),  
nombre varchar2(20),  
id_chef number(4),  
telefono number(10)  
);
```

```
insert into restaurante  
values (1,'Ratatouille',1,640702245);
```

```
insert into restaurante  
values (2,'Pasion de la carne',3,648342245);
```

```
insert into restaurante  
values (3,'Pollos Copacabana',4,640712345);
```

```
insert into restaurante  
values (4,'Rincon del vegano',2,645602245);
```

```
insert into restaurante  
values (5,'Chefsito',1,641202245);
```

FUNCION

```
create or replace function recetas_disponibles (idch number) return number  
as  
num_rec number(4);
```

```
begin  
select count(*) into num_rec  
from recetas  
join autoria on autoria.id_receta=recetas.id_receta  
where autoria.id_chef= idch;  
  
return num_rec;
```

```
end;
```

```
select nombre, recetas_disponibles (1) from restaurante  
where id_chef=1;
```

PROCEDURE

```
create or replace procedure cambio_telf (nomres in varchar2, telf in number)  
is  
begin
```

```
    update restaurante set telefono= telf  
    where nombre=nomres;  
end;
```

```
set serveroutput on  
declare  
v_name restaurante.nombre%type;  
v_telf restaurante.telefono%type;  
begin  
v_name:='Chefsito';  
v_telf:=6999999990;  
  
cambio_telf(v_name,v_telf);  
dbms_output.put_line(v_name||' cambio de telefono al: '||v_telf);  
end;  
/
```

TRIGGER

```
drop table audi_rest;
```

```
create table audi_rest(  
estado varchar2 (20),  
nombre varchar2 (30),  
usuario varchar2 (20),  
fecha date,  
hora varchar2 (10)
```

```
);
```

```
create or replace trigger log_audi_rest  
before insert or update or delete of nombre on restaurante  
for each row  
begin
```

```

    if inserting then
        insert into audi_rest
            values('añadido', :new.nombre , (select user from dual), sysdate, (select
to_char(current_timestamp,'HH24-MI-SS') from dual));
    end if;

    if deleting then
        insert into audi_rest
            values('eliminado',:old.nombre, (select user from dual), sysdate, (select
to_char(current_timestamp,'HH24-MI-SS') from dual));
    end if;

    if updating then
        insert into audi_rest
            values('modificado',:old.nombre||' a '||:new.nombre, (select user from dual), sysdate,
(select to_char(current_timestamp,'HH24-MI-SS') from dual));
    end if;
end;

/

begin
INSERT INTO restaurante VALUES
    (6,'Ajayu',2,225185098);

update restaurante
    set nombre = 'La sanguijuela'
    where id_rest = 6;

delete from restaurante
    where id_rest = 6;
end;

/

```