Estudiante: Guadalupe Castillo 8-929-2252 Grupo: 1IF-131

Caso 1

Create table Avion(

piso number,

asiento number,

clase number,

disponible char,

constraint pk\_avion primary key (piso, asiento, clase));

set serveroutput on;

DECLARE

V\_piso Avion.piso%TYPE;

V\_asiento Avion.asiento%TYPE;

V\_clase Avion.clase%TYPE;

V\_disponible Avion.disponible%TYPE;

V\_error out varchar;

BEGIN

dbms\_output.put\_line(‘Ingresa el número de piso: ’);

v\_piso:= ‘&v\_piso’

If v\_piso=1 Then

insert into table Avion(piso, asiento, clase, disponible) values(1, 1, 4, ‘s’);

elseif v\_piso=2 Then

insert into table Avion(piso, asiento, clase, disponible) values(2, 218, 2, ‘n’);

insert into table Avion(piso, asiento, clase, disponible) values(2, 240, 3, ‘s’);

else

insert into table Avion(piso, asiento, clase, disponible) values(3, 280, 1, ‘n’);

END IF;

EXCEPTION

WHEN OTHERS THEN

v\_error:= 'El dato no existe';

COMMIT;

END;

Caso 2

A. Creación de tablas

Create table Habitaciones(

idcadena number,

piso number not null,

habitacion varchar(2) not null,

tipo varchar(10) default ‘Doble’ not null,

Contraint Check (‘Individual’, ‘Doble’, ‘Suite’),

constraint pk\_habitacion primary key (piso, habitacion),

constraint fk\_cadena\_est foreign key (idcadena) references Estadisticahotel(idcadena));

Create table Reservas(

piso number,

habitacion varchar(2),

fechaentrada date default sysdate,

noches number,

constraint pk\_reservas primary key (piso, habitacion, fechaentrada),

constraint fk\_habitaciones foreign key (piso, habitacion) references Habitaciones(piso, habitacion));

Create table Temporadas(

nombre varchar(10) not null,

mesinicio number not null,

diainicio number not null,

mesfinalizacion number not null,

diafinalizacion number not null,

constraint pk\_temporada primary key (nombre, mesinicio, diainicio));

Create table Estadisticashotel(

idcadena number,

nombre varchar2(50),

habitacionesocupadas number,

habitacioesdisponibles number,

habitacionesreservas number,

constraint pk\_Esthotel primary key (idcadena));

B. Procedimiento para insertar en la tabla Temporadas

Create or Replace Procedure P\_Temporadas(

p\_nombre Temporadas.nombre%TYPE,

p\_mesinicio Temporadas.mesinicio%TYPE,

p\_diainicio Temporadas.diainicio%TYPE,

p\_mesfinalizacion Temporadas.mesfinalizacion%TYPE,

p\_diafinalizacion Temporadas.diafinalizacion%TYPE

)AS

BEGIN

insert into Temporadas(nombre, mesinicio, diainicio, mesfinalizacion, diafinalizacion)

values(p\_nombre, p\_mesinicio, p\_diainicio, p\_mesfinalizacion, p\_diafinalizacion);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END P\_Temporadas;

BEGIN

Add\_P\_Temporadas(‘Alta’, 6, 1, 8, 30);

Add\_P\_Temporadas(‘Media’, 1, 1, 31, 5);

Add\_P\_Temporadas(‘Baja’, 1, 9, 31, 12);

END;

C. Procedimiento para llenar la tabla Habitaciones

Create or Replace Procedure P\_Habitaciones(

p\_idcadena Habitaciones.idcadena%TYPE,

p\_piso Habitaciones.piso%TYPE,

p\_habitacion Habitaciones.habitacion%TYPE,

p\_tipo Habitaciones.tipo%TYPE

)AS

BEGIN

insert into Habitaciones(idcadena, piso, habitacion, tipo)

values(p\_idcadena, p\_piso, p\_habitacion, p\_tipo);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END P\_Habitaciones;

BEGIN

Add\_P\_Temporadas(4, 2, 8, 18);

Add\_P\_Temporadas(4, 3, 12, 7);

Add\_P\_Temporadas(8, 10, 8, 23);

END;

D. Procedimiento que permita actualizar las habitaciones que han sido ocupadas.

create or replace function c\_hab (

f\_idcadena Habitaciones.idcadena%TYPE,

f\_piso Habitaciones.piso%TYPE,

f\_habitacion Habitaciones.habitacion%TYPE,

f\_tipo Habitaciones.tipo%TYPE

)

RETURN NUMBER AS

BEGIN

Return ;

END ;

Create or Replace Procedure P\_Update\_Habitaciones(

p\_idcadena Habitaciones.idcadena%TYPE,

p\_piso Habitaciones.piso%TYPE,

p\_habitacion Habitaciones.habitacion%TYPE,

p\_tipo Habitaciones.tipo%TYPE

)AS

BEGIN

insert into Habitaciones(idcadena, piso, habitacion, tipo)

values(p\_idcadena, p\_piso, p\_habitacion, p\_tipo);

UPDATE Habitaciones

SET habitacion

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No se encontro la tabla');

COMMIT;

END P\_Update\_Habitaciones;