**Parcial N°2**

Nombre: Reynaldo Rojas Cédula: 8-950-792 Grupo: 1IF131

**I PROBLEMA**

create table AVION (

piso number not null,

asiento number not null,

clase number not null,

disponible char default 'S',

constraint p\_a\_c\_pk primary key (piso, asiento, clase)

);

DECLARE

p\_piso avion.piso%TYPE,

p\_asiento avion.asiento%TYPE,

p\_clase avion.clase%TYPE,

p\_disponible avion.disponible%TYPE;

BEGIN

FOR p\_piso IN 1..3 LOOP

IF p\_piso=1 THEN

FOR p\_asiento IN 1..216 LOOP

INSERT INTO AVION (piso, asiento, clase, disponible)

VALUES (p\_piso, p\_asiento, 4, p\_disponible);

END LOOP;

ELSE IF p\_piso = 2 THEN

FOR p\_asiento IN 217..236 THEN

INSERT INTO AVION (piso, asiento, clase, disponible)

VALUES (p\_piso, p\_asiento, 2, p\_disponible);

END LOOP;

FOR p\_asiento IN 237..277 THEN

INSERT INTO AVION (piso, asiento, clase, disponible)

VALUES (p\_piso, p\_asiento, 3, p\_disponible);

END LOOP;

ELSE

FOR p\_asiento IN 278..288 THEN

INSERT INTO AVION (piso, asiento, clase, disponible)

VALUES (p\_piso, p\_asiento, 1, p\_disponible);

END LOOP;

END IF;

END LOOP;

EXCEPTION

WHEN

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('No hay datos');

END;

/

DECLARE

**II PROBLEMA**

create table TEMPORADAS (

nombre varchar(10) not null,

mesinicio int not null,

diainicio int not null,

mesfinalizacion int not null,

dia\_finalizacion int not null,

constraint temp\_pk primary key (nombre, mesinicio, diainicio)

);

CREATE TABLE ESTADISTICAHOTEL (

Id\_cadena number,

Nombre varchar2(50),

Habitacionesocupadas number,

Habitacionesdisponibles number,

Habitacionesreservadas number,

Constraint estadisticahotel\_pk primary key (id\_cadena)

);

create table HABITACIONES (

id\_cadena number,

piso int not null,

habitacion varchar(2) not null,

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

constraint habitacion\_pk primary key (piso, habitacion),

constraint habitacion\_fk foreign key (id\_cadena) references estadisticahotel (id\_cadena)

);

create table RESERVAS (

piso int not null,

habitacion varchar(2),

fechaentrada default sysdate,

noches int,

constraint p\_h\_f\_pk primary key (piso, habitacion, fechaentrada),

constraint fk\_reservas\_piso foreign key (piso) references habitaciones (piso),

constraint fk\_reservas\_habitacion foreign key (habitacion) references habitaciones (habitacion)

);

CREATE OR REPLACE PROCEDURE add\_season(

V\_nombre Temporadas.nombre%TYPE,

V\_mesinicio Temporadas.mesinicio%TYPE,

V\_diainicio Temporadas.diainicio%TYPE,

V\_mes\_finalizacion Temporadas. mesfinalizacion%TYPE,

V\_dia\_finalizacion Temporadas.dia\_finalizacion%TYPE

) AS

BEGIN

INSERT INTO (nombre, mesinicio, diainicio, mesfinalizacion, dia\_finalizacion)

VALUES(v\_nombre, v\_mesinicio, v\_diainicio, v\_mes\_finalizacion, v\_dia\_finalizacion);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

COMMIT;

END add\_season;

BEGIN

Add\_season('Alta',6,1,8,30);

Add\_season('Medio',1,1,31,5);

Add\_season('Baja',1,9,31,12);

END;

CREATE OR REPLACE PROCEDURE add\_room(

C\_piso habitaciones.piso%TYPE,

C\_habitacion habitaciones.habitacion%TYPE,

C\_tipo habitaciones.tipo%TYPE

) AS

BEGIN

FOR c\_piso IN 1..12 LOOP

IF c\_piso <12 THEN

FOR i IN 1..25 THEN

IF i <=20 THEN

INSERT INTO HABITACIONES (piso, habitacion)

VALUES (c\_piso, i.c\_habitacion);

ELSE

INSERT INTO HABITACIONES (piso, habitacion, tipo)

VALUES (c\_piso, i.c\_habitacion, 'Sencilla');

END IF;

FOR y IN 1..8 LOOP

INSERT INTO HABITACIONES (piso, habitacion, tipo)

VALUES (c\_piso, y.c\_habitacion, 'Suite');

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

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

COMMIT;

END add\_room;

/

BEGIN

Add\_room ();

END;

/

CREATE OR REPLACE FUNCTION ocupado (

Ocupado estadisticashotel.habitacionesocupadas%TYPE,

Available estadisticashotel.habitacionesdisponibles%TYPE,

Reserva estadisticashotel.habitacionesreservadas%TYPE

)

RETURN NUMBER AS

BEGIN

Calculo := (ocupado + available + reserve) – available;

RETURN calculo;

END ocupado;

CREATE OR REPLACE FUNCTION disponible (

Ocupado estadisticashotel.habitacionesocupadas%TYPE,

Available estadisticashotel.habitacionesdisponibles%TYPE,

Reserva estadisticashotel.habitacionesreservadas%TYPE

)

RETURN NUMBER AS

BEGIN

Calculo := (ocupado + available + reserve) – ocupada;

RETURN calculo;

END disponible;

CREATE OR REPLACE FUNCTION reserva (

Ocupado estadisticashotel.habitacionesocupadas%TYPE,

Available estadisticashotel.habitacionesdisponibles%TYPE,

Reserva estadisticashotel.habitacionesreservadas%TYPE

)

RETURN NUMBER AS

BEGIN

Calculo := (ocupado + available + reserve) – ocupado-available;

RETURN calculo;

END reserva;