CREATE TABLE AVION (

Piso number not null,

Asiento number not null,

Clase number not null,

Disponible char not null,

Constraint t\_avion primary key (piso, asiento, clase),

);

DECLARE

v\_piso avión.piso%type,

v\_asiento avión.asiento%type,

v\_clase avión.clase%type,

v\_disponible avión.disponible%type Default ‘S’

BEGIN

FOR p IN 1..3 LOOP

IF p = 1 THEN

FOR a IN 1..216 LOOP

V\_disponible = ‘&v\_disponible’;

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

VALUES (p, a, 4, v\_disponible)

END LOOP;

ELSIF p = 2 THEN

FOR a IN 217..236 LOOP

V\_disponible = ‘&v\_disponible’;

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

VALUES (p, a, 1, v\_disponible)

END LOOP;

FOR a IN 237..277 LOOP

V\_disponible = ‘&v\_dispo’;

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

VALUES (p, a, 2, v\_disponible)

END LOOP;

ELSE

FOR a IN 278..288 LOOP

V\_disponible = ‘&v\_disponible’;

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

VALUES (p, p3, 4, v\_disponible)

END LOOP;

END IF;

END LOOP;

END;

create table TEMPORADAS (

nombre varchar(10) not null,

mesinicio int not null,

diainicio int not null,

mesfinalizacion int not null,

día\_finalizacion int not null,

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

);

CREATE TABLE ESTADISTICAHOTEL (

Id\_cadena number not null,

Nombre varchar2(50),

Habitacionesocupadas number,

Habitacionesdisponibles number,

Habitacionesreservadas number,

Constraint id\_estadistica\_fk primary key (id\_cadena)

);

CREATE TABLE HABITACIONES (

Id\_cadena number not null,

Piso int not null,

Habitación varchar(2) not null,

Tipo varchar(10) Default ‘Doble’,

Constraint habitaciones\_pk primary key (piso, habitacion),

Constraint id\_cadena\_fk foreign key (id\_cadena) references estadisticahotel,

Constraint chk\_Tipo CHECK( Tipo = 'Individual' Tipo = 'Doble' Tipo = 'Suite'),

(id\_cadena);

);

CREATE TABLE RESERVAS (

piso int,

habitacion varchar(2),

fechaentrada default sysdate,

noches int,

primary key (piso, habitacion, fechaentrada),

constraint reservas\_fk foreign key (piso) references habitaciones (piso),

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

);

CREATE OR REPLACE PROCEDURE Crear\_temporadas(

P\_nombre Temporadas.nombre%TYPE,

P\_mesinicio Temporadas.mesinicio%TYPE,

P\_diainicio Temporadas.diainicio%TYPE,

P\_mesfin Temporadas. mesfinalizacion %TYPE,

P\_diafin Temporadas.dia\_finalizacion %TYPE

)AS

BEGIN

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

VALUES(P\_nombre, P\_mesinicio, P\_diainicio, P\_mesfin, P\_diafin);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS.OUTPUT.PUT\_LINE (‘Dato Existente’);

WHEN OTHERS THEN

DBMS.OUTPUT.PUT\_LINE (‘No se creo el registro’);

COMMIT;

END Crear\_temporadas;

BEGIN

Crear\_temporadas('Alta', 6, 1, 8, 30);

Crear\_temporadas('Medio', 1, 1, 31, 5);

Crear\_temporadas('Baja',1, 9, 31, 12);

END;

CREATE OR REPLACE PROCEDURE crear \_habitaciones(

p\_error out varchar2

)AS

P\_Piso habitaciones.piso%TYPE;

P\_Habitacion habitaciones.Habitacion %TYPE;

BEGIN

FOR P\_Piso IN 1..12 LOOP

IF P\_Piso >=1 AND P\_Piso<=11

FOR x IN 1..25 LOOP

IF P\_Habitacion = 20 THEN

P\_Tipo := 'Dobles';

ELSE

P\_Tipo := 'Sencilla';

END IF;

INSERT INTO habitaciones( piso, Habitacion, Tipo)

VALUES(P\_Piso,x, P\_Tipo);

END LOOP;

--Inicio para las habitaciones del piso 12

ELSE

FOR y IN 1..8 LOOP

P\_Tipo := 'Suites';

INSERT INTO habitaciones( piso, Habitacion, Tipo)

VALUES(P\_Piso,y, P\_Tipo);

END LOOP;

END IF;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS.OUTPUT.PUT\_LINE (‘Dato Existente’);

WHEN OTHERS THEN

DBMS.OUTPUT.PUT\_LINE (‘No se creo el registro’);

COMMIT;

END crear \_habitaciones;

DECLARE

v\_error varchar2(100);

BEGIN

crear \_habitaciones(v\_error);

END;

CREATE OR REPLACE FUNCTION hab\_ocup (

Habitación habitaciones.habitacion%TYPE,

Habitacionesdisponibles estadisticashotel.habitacionesdisponibles%TYPE

)

RETURN NUMBER AS

BEGIN

RUTURN habitaciones - habitacionesdisponibles;

END hab\_ocup;

CREATE OR REPLACE FUNCTION hab\_disp (

Habitación habitaciones.habitacion%TYPE,

Habitacionesocupadas estadisticashotel.habitacionesocupadas%TYPE

)

RETURN NUMBER AS

BEGIN

RUTURN habitaciones - habitacionesocupadas;

END hab\_disp;