

1. Caso No.1 Implementación de bloques PL/Sql.

```
Create Table Avion (  
  Piso number not null,  
  Asiento number not null,  
  Clase number not null,  
  Disponible char DEFAULT 'S' ,  
  constraint pk_Avion primary key (Piso, Asiento,Clase));
```

```
DECLARE  
v_Piso Avion.Piso%TYPE;  
v_Asiento Avion.Asiento%TYPE;  
v_Clase Avion.Clase%TYPE;  
v_Dispo Avion.Disponible%TYPE;  
BEGIN  
  For v_Piso IN 1..3 LOOP  
    IF v_Piso = 1 THEN  
      For v_Asiento IN 1..216 LOOP  
        v_Dispo := '& v_Dispo';  
        INSERT INTO Avion (Piso, Asiento, Clase, Disponible) VALUES(v_Piso,  
          v_Asiento,4, v_Dispo);  
      END LOOP;  
    ELSEIF v_Piso = 2 THEN  
      For v_Asiento IN 217..236 LOOP  
        v_Dispo := '& v_Dispo';  
        INSERT INTO Avion (Piso, Asiento, Clase, Disponible) VALUES(v_Piso,  
          v_Asiento,2, v_Dispo);  
      END LOOP;  
    For v_Asiento IN 237..277 LOOP  
      v_Dispo := '& v_Dispo';  
      INSERT INTO Avion (Piso, Asiento, Clase, Disponible) VALUES(v_Piso,  
        v_Asiento,3, v_Dispo);  
    END LOOP;  
    ELSE  
      For v_Asiento IN 278..288 LOOP  
        v_Dispo := '& v_Dispo';  
        INSERT INTO Avion (Piso, Asiento, Clase, Disponible) VALUES(v_Piso,  
          v_Asiento,1, v_Dispo);  
      END LOOP;  
    END IF;  
  END LOOP  
END;
```

2. Caso No.2 Implementación de programación almacenada de base de datos.

A. Creacion de tablas

```
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 pk_Temporadas primary key (nombre, mesinicio, diainicio),  
);
```

```
CREATE TABLE estadistichotel (  
Id_cadena      number not null,  
Nombre         varchar2(50),  
Habitacionesocupadas  number,  
Habitacionesdisponibles  number,  
Habitacionesreservadas  number,  
Constraint pk_estadistic 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 CHECK  
('Individual','Doble','Suite'),  
constraint fk_habitaciones foreign key(Id_cadena), references  
estadistichotel(Id_cadena));
```

```
Create Table reservas(  
Piso int,  
Habitacion varchar(2),  
Fechaentrada DEFAUT sysdate,  
Noches int,  
Constraint pk_ reservas primary key(Piso, Habitacion, Fechaentrada),  
constraint fk_reservas_pis foreign key (Piso) references (Piso),  
constraint fk_reservas_habi foreign key(Habitacion) references(Habitacion));
```

B. Procedimiento para la tabla temporadas

```
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
p_error out varchar2)AS
BEGIN
INSERT INTO (nombre, mesinicio, diainicio, mesfinalizacion, dia_finalizacion)
VALUES(P_nombre, P_mesinicio, P_diainicio, P_mesfin, P_diafin);
p_error := 'Registro Creado';
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
p_error := 'Temporada creada';
WHEN OTHERS THEN
p_error := 'No se creo el registro';
COMMIT;
END Crear_temporadas;
```

```
DECLARE
```

```
        v_error varchar2(100);
```

```
BEGIN
```

```
Crear_temporadas('Alta',6,1,8,30, v_error);
Crear_temporadas('Medio',1,1,31,5, v_error);
Crear_temporadas('Baja',1,9,31,12, v_error);
dbms_output.put_line(v_error);
END;
```

C. Tabla habitaciones

```
CREATE OR REPLACE PROCEDURE crear_habitaciones(
p_error out varchar2)AS
P_Piso habitaciones.piso%TYPE;
P_Habitacion habitaciones.Habitacion %TYPE;
BEGIN

p_error := 'Registro Creado';

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
p_error := 'Tipo de correo existente';
WHEN OTHERS THEN
p_error := 'No se creo el registro';
COMMIT;
END crear_habitaciones;

```

```

DECLARE

```

```

        v_error varchar2(100);

```

```

BEGIN

```

```
crear_habitaciones(v_error);  
END;
```

D. Habitaciones ocupadas.

```
CREATE OR REPLACE FUNCTION H_ocupadas(  
Habitacionesd estadisticashotel.habitacionesdisponibles%TYPE  
) RETURN NUMBER AS  
BEGIN  
RETURN 283-Habitacionesd;  
END H_ocupadas;  
CREATE OR REPLACE FUNCTION H_dispo(  
Habitacioneso estadisticashotel.Habitacionesocupadas%TYPE  
) RETURN NUMBER AS  
RETURN 283-Habitacioneso;  
END H_dispo;
```