DECLARE

x NUMBER := 100;

BEGIN

FOR i IN 0..N LOOP

IF MOD(i,2) = 0 THEN -- i is even

INSERT INTO CONDICIONES\_TECNICAS VALUES (i, 'Luces, sillas movibles, conexiones');

ELSIF MOD(i,3) = 0 THEN -- i is even

INSERT INTO CONDICIONES\_TECNICAS VALUES (i, 'Amplificacion de sonido, instalaciones propias');

ELSIF MOD(i,4) = 0 THEN -- i is even

INSERT INTO CONDICIONES\_TECNICAS VALUES (i, 'Amplificacion de sonido, instalaciones propias');

ELSE

INSERT INTO CONDICIONES\_TECNICAS VALUES (i, 'Luces de colores, efectos 3D');

END IF;

x := x + 100;

END LOOP;

END;

Este loop se utilizó para generar los datos en la tabla condiciones técnicas, usando el resultado de la función modulo sobre el id para asignar aleatoriamente un valor.

BEGIN

FOR i IN 1..N LOOP

IF MOD(i,2) = 0 THEN -- i is even

INSERT INTO ABONOS VALUES (i, mod(i,2), i+10);

ELSIF MOD(i,3) = 0 THEN -- i is even

INSERT INTO ABONOS VALUES (i, mod(i,3), i+10);

ELSIF MOD(i,4) = 0 THEN -- i is even

INSERT INTO ABONOS VALUES (i, mod(i,4), i+10);

ELSE

INSERT INTO ABONOS VALUES (i, mod(i,5),i+10);

END IF;

x := x + 100;

END LOOP;

END;

Esta función es similar a la anterior y se utilizo en abonos, esta usa múltiples funciones modulo para la asignación de datos para garantizar que algunos clientes se repitieran para simular la compra de múltiples boletas en un solo abono. N es el número de valores generados

insert into CLIENTES(nombre, id, email, rol, preferencia, password, es\_abonado)

select 'Nombre ' || to\_char(rownum), to\_char(rownum), 'nombre@email ' || to\_char(rownum), 'Rol ' || to\_char(rownum), 'pref ' || to\_char(rownum), 'pass' || to\_char(rownum),

IF EXISTS ( SELECT \* FROM ABONOS WHERE rownum=id\_cliente)

‘T’

END IF;

‘F’

from dual

connect by level <= N;

Esta es la function que se utilize para generar los datos en la tabla clientes, usando una tabla del sistema para generar los números, un IF que revisa si el id actual del cliente existe en la tabla de abonos para agregar el atributo correcto de si es abonado o no, y el N es el número de datos generados.

insert into Usuarios\_no\_registrados(nombre, id, email, rol)

select 'Nombre ' || to\_char(rownum), to\_char(rownum), 'nombre@email ' || to\_char(rownum), 'Rol ' || to\_char(rownum) from dual

connect by level <=N;

Similar a la tabla anterior esta función genera datos var char aleatorios siguiendo un esquema de agregar un número a diferentes Strings para similar los datos.

insert into Sillas(codigo, numcolumna, numfila, localidad, codentrada)

select rownum, mod(rownum,2) , mod(rownum,2), 'localidad ' || to\_char(rownum), mod(rownum,2)

from dual

connect by level <= N;

Esta función genera datos aleatorios con patrones repitentes para similar el hecho de que se compartirán escenarios y el número de filas no va cambiar.

insert into Entradas(codigo, numcolumna, numfila, precio, codpersona, codcliente, codfuncion)

select rownum, mod(rownum,2) , mod(rownum,2), rownum+1000, mod(rownum,2), mod(rownum,2), mod(rownum,2)

from dual

connect by level <= N;

Esta función es similar a la anterior en que genera datos con códigos y filas repitentes para asegurase que el número de sillas se mantenga consistentes con la vida real.

insert into Funciones(codigo, fecha, hora, realizada, entradasDisponibles, entradasVendidas, codEspectaculo, codEscenario, idCompania)

select rownum, 'fecha ' || to\_char(rownum), 'fecha ' || to\_char(rownum), 'Rol ' || to\_char(rownum), 'numEntradas ' || to\_char(rownum), 'numVendidad ' || to\_char(rownum), 'codigo ' || to\_char(rownum))

from dual

connect by level <= N;

Esta funcion genera datos para la tabla funciones basándose en las funciones anteriores, utiliza un generador de números secuenciales para garantizar que los números y fechas siempre sean distintas y se incluye la funcion modulo para asegurarse que algunas horas se repitan para simular un dia concurrido en el festical.