**Laboratorio 6: Replicación Bidireccional**

1. Recordar qué objetos están creados en la base de datos (tanto de forma local como remota conectándose a SQL\*PLUS con el usuario respectivo):

COLUMN OBJECT\_NAME FORMAT A20;

COLUMN OBJECT\_TYPE FORMAT A20;

SELECT OBJECT\_NAME, OBJECT\_TYPE

FROM USER\_OBJECTS;

1. Ahora loguearse con el usuario local desde SQL\*PLUS y crear la tabla Promociones usando la siguiente transacción

CREATE TABLE PROMOCIONES

(CODIGOP CHAR (5)

CHECK (SUBSTR(CODIGOP, 3, 1) = 'P'),

DESCRIPCION CHAR (40),

STOCK NUMBER (10, 2),

FECHAINICIAL DATE,

FECHAFINAL DATE,

CONSTRAINT PK\_PROMOCIONES PRIMARY KEY (CODIGOP));

1. Desplegar los objetos y verificar la creación de la tabla Promociones

COLUMN OBJECT\_NAME FORMAT A20;

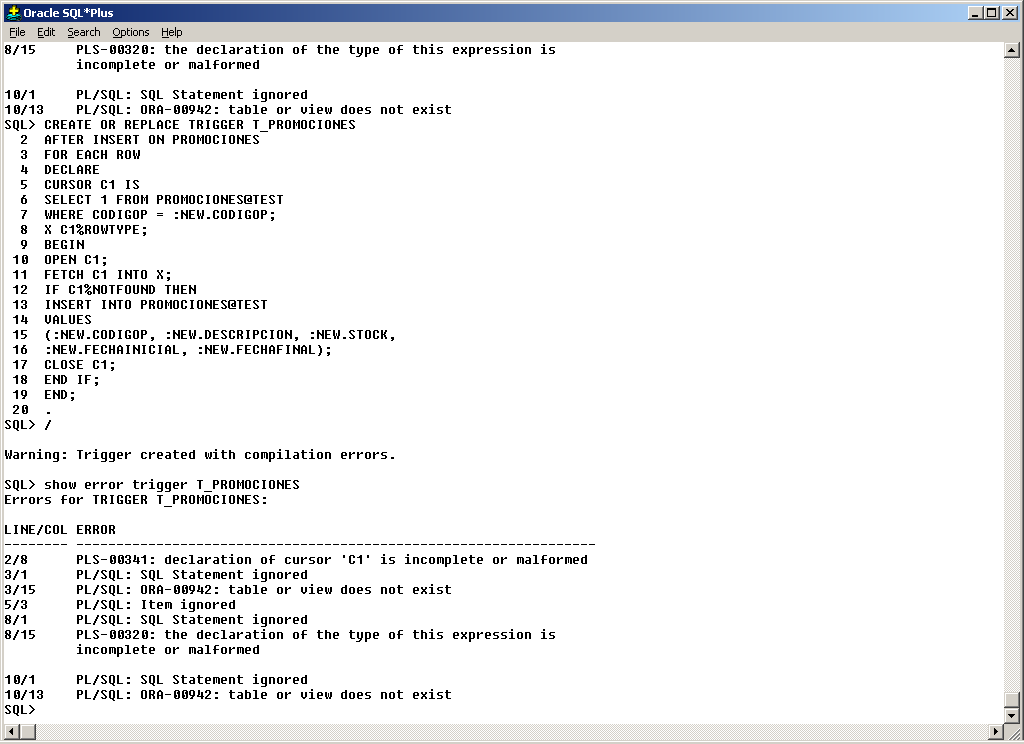
COLUMN OBJECT\_TYPE FORMAT A20;

SELECT OBJECT\_NAME, OBJECT\_TYPE

FROM USER\_OBJECTS;

1. Crear el siguiente trigger. En la imagen, en la línea 6 y línea 13 tener en cuenta el nombre de su base de datos remota. NOTA: Saldrá un Warning.

Para ver los errores usamos *show error trigger*



**Responda**: Qué hace el trigger?

1. Ahora loguearse con el usuario remoto y la cadena de conexión desde SQL\*PLUS a la base remota y crear también la tabla Promoción (porque dicha tabla no estaba creada y por eso nos salía el error del disparador) usando la siguiente transacción

CREATE TABLE PROMOCIONES

(CODIGOP CHAR (5)

CHECK (SUBSTR(CODIGOP, 3, 1) = 'P'),

DESCRIPCION CHAR (40),

STOCK NUMBER (10, 2),

FECHAINICIAL DATE,

FECHAFINAL DATE,

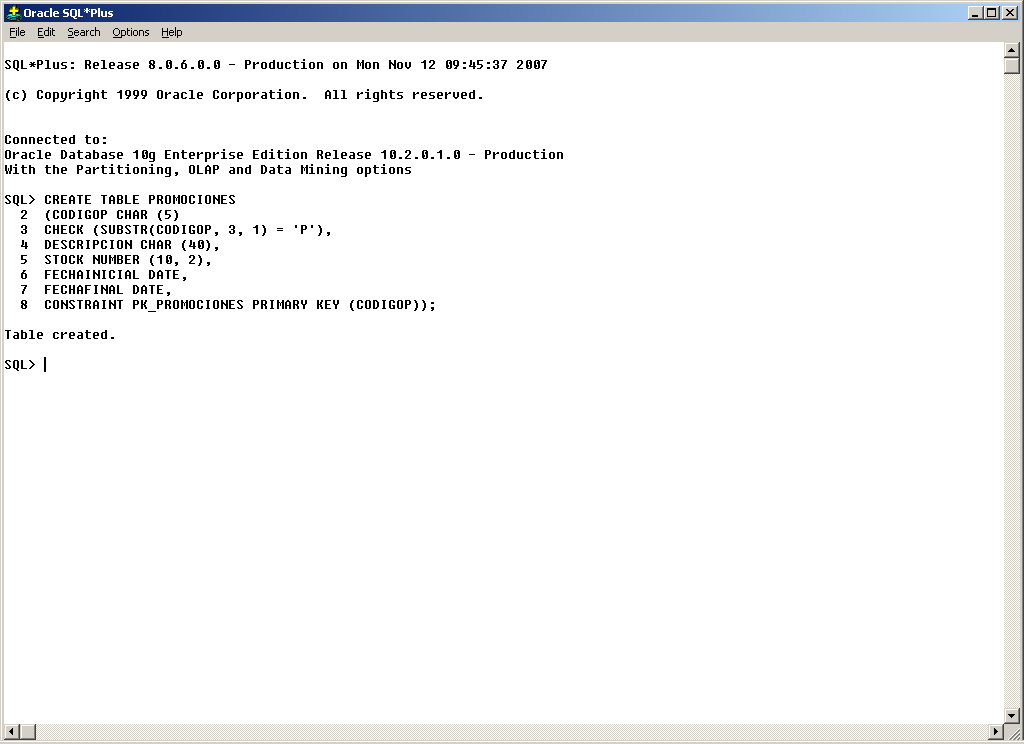
CONSTRAINT PK\_PROMOCIONES PRIMARY KEY (CODIGOP));

COLUMN OBJECT\_NAME FORMAT A20;

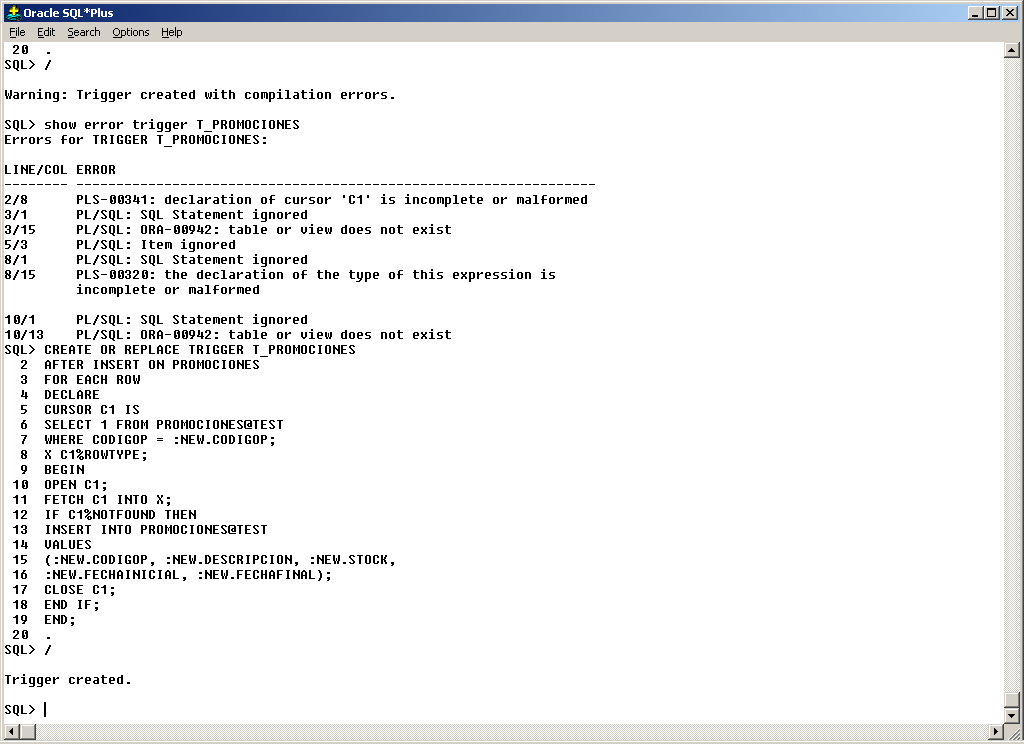
COLUMN OBJECT\_TYPE FORMAT A20;

SELECT OBJECT\_NAME, OBJECT\_TYPE

FROM USER\_OBJECTS;



1. Ahora, ya creada la tabla en el servidor remoto vamos a crear el mismo trigger para lograr crear la replicación bidireccional. Además, ejecutar el trigger nuevamente en la base de datos local y verificar que ya funciona el trigger que antes tenía errores.



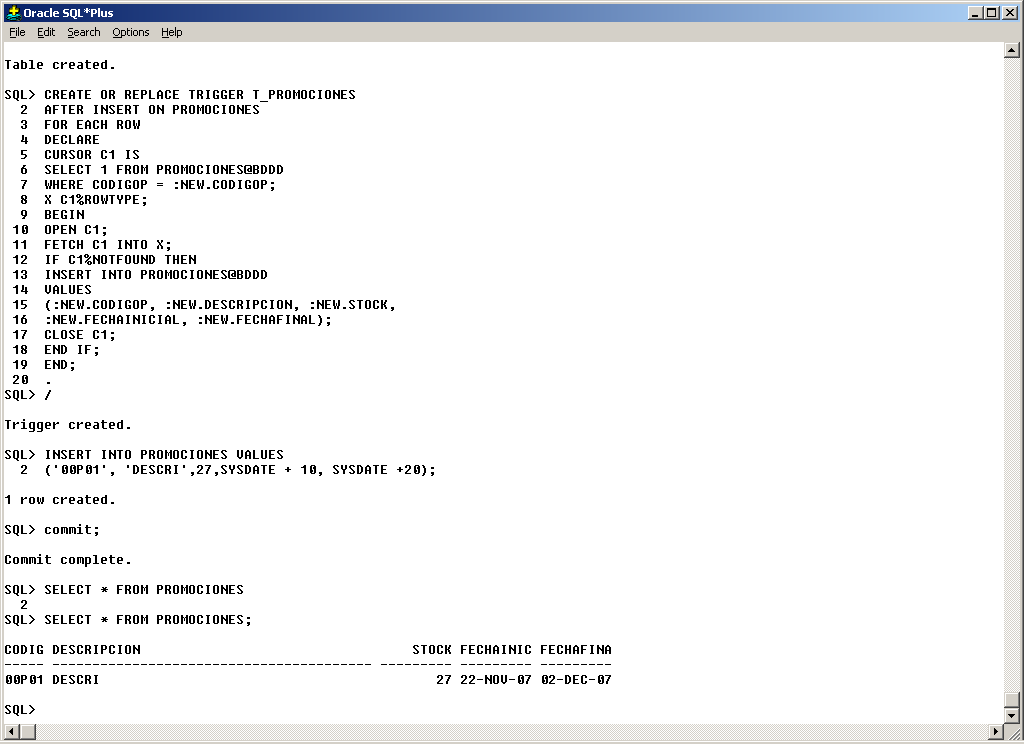
1. Ahora vamos a insertar una promoción, (en la base de datos local) para lo cual ingresamos los siguientes campos de la tabla *promociones*:

* Código
* Descripción
* Cantidad de Stock
* Fecha en que se Inicia la promoción
* Fecha en que termina la promoción

Para nuestro ejemplo ingresaremos la siguiente transacción.

INSERT INTO PROMOCIONES VALUES

('00P01', 'DESCRI',27,SYSDATE + 10, SYSDATE +20);



**Responda**: Qué hace SYSDATE?

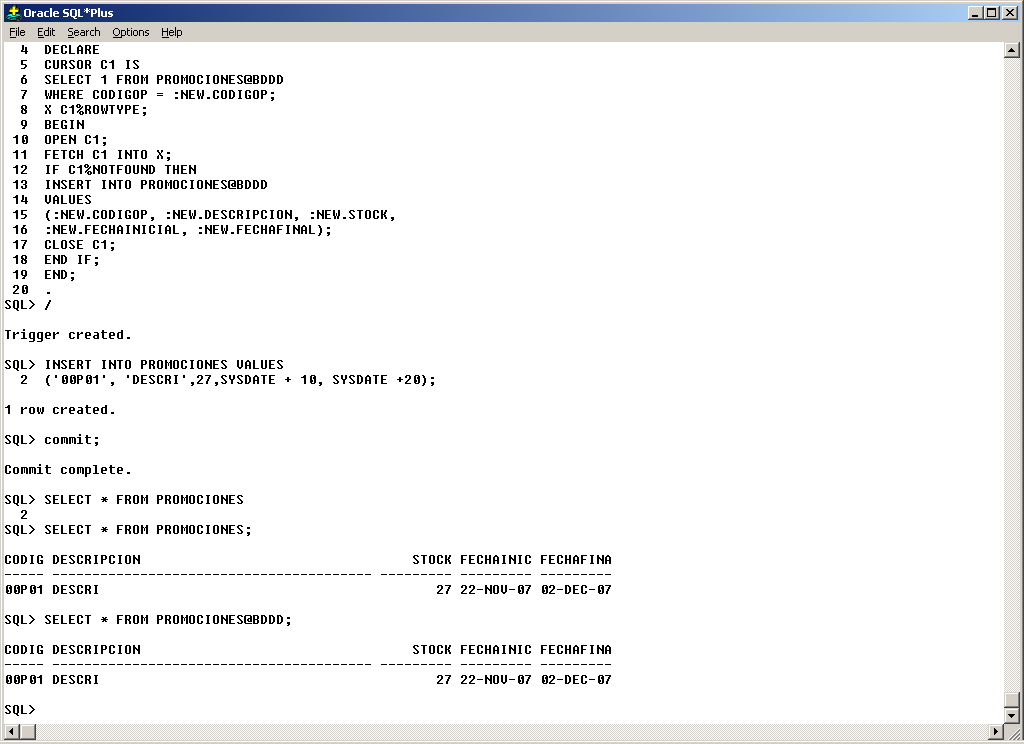
1. Vemos la promoción que acabamos de ingresar usando

SELECT \* FROM PROMOCIONES;

1. Para comprobar si funcionó la implementación de replicación bidireccional debemos revisar si los datos ingresados se replicaron en la base remota, para esto usamos la consulta

SELECT \* FROM PROMOCIONES**@baseRemota**;

Si está correcto veremos un resultado como el que se refleja en la siguiente figura.

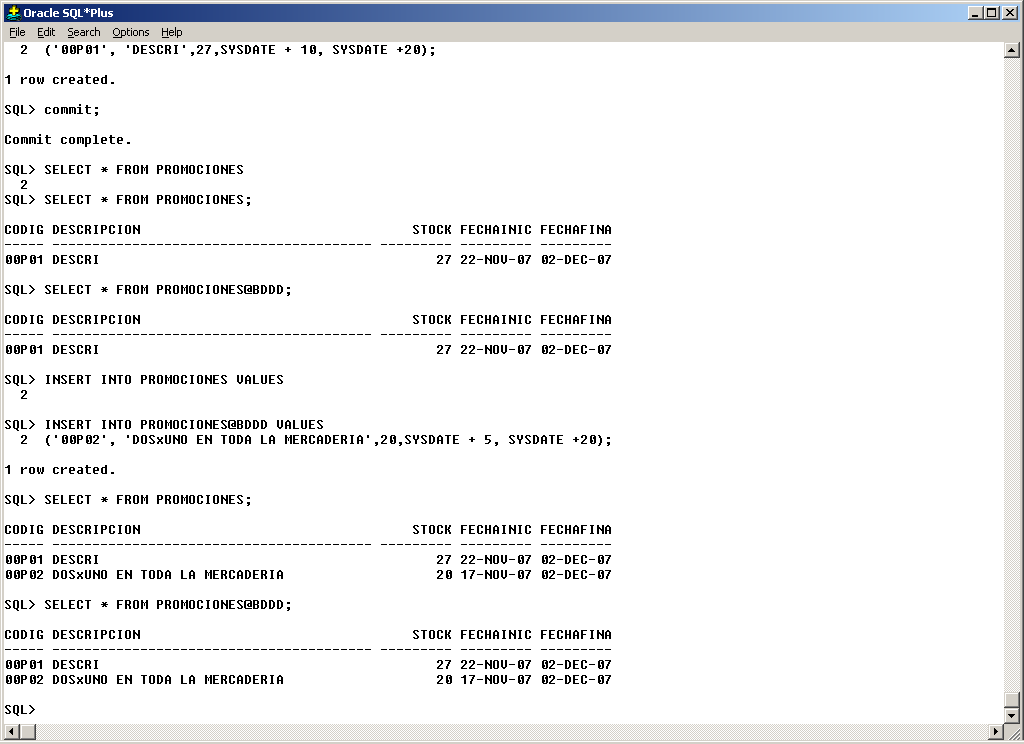


1. Ahora vamos a insertar en la base remota desde el nodo local, para eso usamos la transacción

INSERT INTO PROMOCIONES**@baseRemota** VALUES

('00P03', 'POR SUS COMPRAS RECLAME UN PELUCHE DE RENO',100,SYSDATE + 8, SYSDATE +24);

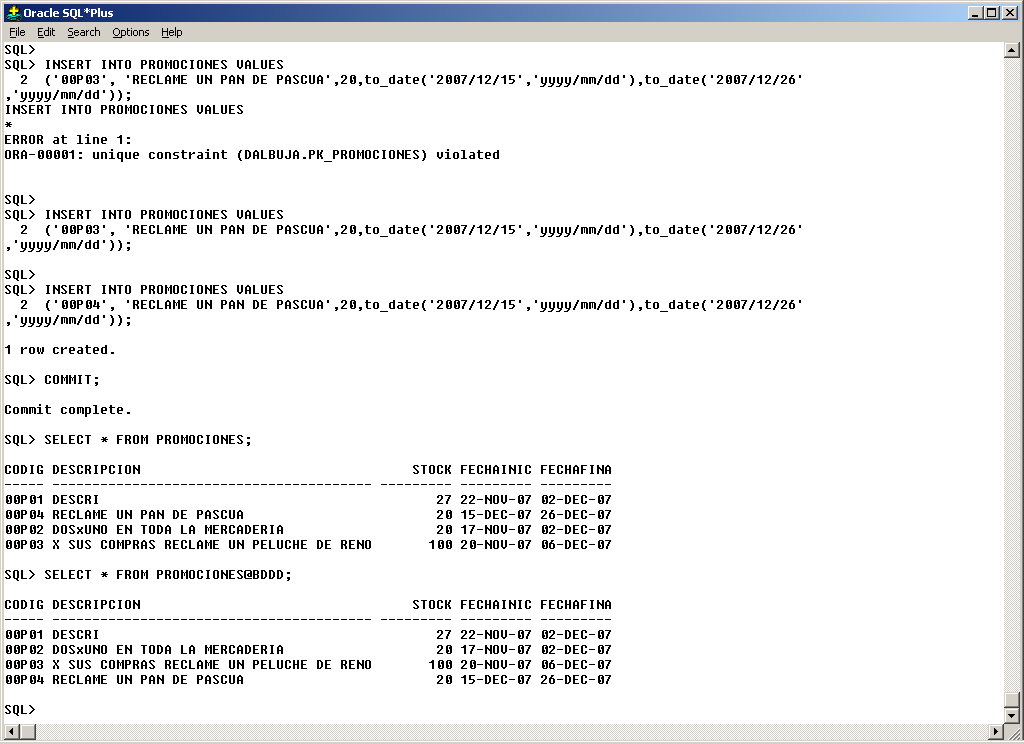
Y luego volvemos a realizar la comprobación realizada en el literal anterior.



1. Ahora vamos a crear una promoción que empiece en una fecha determinada y termine de igual manera, para esto usamos el comando *to\_date*.

INSERT INTO PROMOCIONES VALUES

('00P04', 'RECLAME UN PAN DE PASCUA',20,to\_date('2007/12/15','yyyy/mm/dd'),to\_date('2007/12/26','yyyy/mm/dd'));



**Responda:** Realice más inserciones tanto en la réplica local como en la réplica remota y verifique que esas inserciones se vean reflejadas en ambas bases.