

Taller de  
Base de datos procedure function 2

Alumno:

Jorge ivan Urueta ramos

Docente:

Mg. Luis García cuida

Universidad de Córdoba  
Facultad de ingeniería  
Programa de ingeniería de sistemas  
26-11-2015

Cree un procedimiento declarado en el esquema Clase (o donde tiene las tablas de clase). Inserte un registro en la tabla Salon\_clase.. Los datos de entrada son los requeridos por la tabla salón\_clase. Se debe validar que el salón\_clase a crear no este creado.

## SOLUCION

Creemos un procedimiento llamado EJERCICIO4 y le pedimos como parámetro los datos requeridos por la tabla que son tamaño número salón y nombre de la clase .

En el código están como tam ,num\_sal,nom\_cla y se les pone como tipo de datos el mismo que está en la tabla salón\_clase .luego se declaran 4 variables globales que son num1,num2,num3,num4 que nos servirán para comparar y validar si existen o no los datos ingresados por el usuario .

```
CREATE OR REPLACE PROCEDURE EJERCICIO4(  
TAM SALON_CLASE."TAMAÑO"%TYPE,  
NUM_SAL SALON_CLASE.NUMERO_SAL%TYPE,  
NOM_CLA SALON_CLASE.NOMBRE_CLA%TYPE  
) IS  
NUM1 INTEGER;  
NUM2 INTEGER;  
NUM3 INTEGER;  
NUM4 INTEGER;
```

En la sección del bloque begin se le asigna a las variables num1 un valor de cero("0") al igual que a la variable num2 con el objetivo de que tengan un valor inicial para poder comparar. Luego se hacen las respectivas consultas donde se cuenta cuantas veces se repite el dato ingresado. la primera consulta se cuenta la cantidad de veces que esta numero\_salon en la tabla salón con el objetivo de ver si existe el dato en la tabla salón .lo mismo pasa en la con la segunda consulta que cuenta cuantos nombres hay en el salón comparado con el nombre que el usuario a ingresado y así comprar si existe o no .

En las últimas dos consultas se mira ya la tabla salón\_clase lo que se hace es contar si los datos que ingreso el usuario existe .cada consulta se ha ingresado en las variables que anterior mente creamos en su respectivo orden num1 la primera consulta ,num2 la segunda consulta,num3 la tercera consulta y num4 la cuarta consulta con el objetivo de tener el resultado de la consulta que en este caso es un conteo en una variable .

```
BEGIN  
NUM1:=0;  
NUM2:=0;  
SELECT COUNT(*) INTO NUM1 FROM SALON WHERE NUMERO_SALON=NUM_SAL;  
SELECT COUNT(*) INTO NUM2 FROM CLASE WHERE NOMBRE=NOM_CLA;  
SELECT COUNT(*) INTO NUM3 FROM SALON_CLASE WHERE NOMBRE_CLA=NOM_CLA;  
SELECT COUNT(*) INTO NUM4 FROM SALON_CLASE WHERE NUMERO_SAL=NUM_SAL;
```

Después de hacer todas las consultas lo primero que se hace tomar las variables en donde está el conteo y se condiciona que si el dato num1 es igual a 0 quiere decir que en la tabla salón no existe tal salón y por ende no se puede ingresar los datos luego se hace lo mismo con la variable num2 y se condiciona que si la variable num2 es igual a 0 el dato del campo clase no existe, en la tercera condición ya se usan las variables num3 y num4 y se condiciona que si num3 y num4 son igual a 0 el salón clase no existe y por esta condición me permite que el dato se pueda ingresar el nuevo salón\_clase y en la cuarta y última condición se condicionan las mismas variables solo que la condición cambia que si num3 y num4 son distintos de 0 quiere decir que el salón\_clase ya fue ingresado y no se puede ingresar después.

```
IF(NUM1 = 0) THEN
DBMS_OUTPUT.PUT_LINE('EL DATO DEL CAMPO SALON NO EXITE ');
ELSIF(NUM2 =0) THEN
DBMS_OUTPUT.PUT_LINE('EL DATO DEL CAMPO CLASE NO EXITE ');
ELSIF(NUM3=0 AND NUM4=0) THEN
INSERT INTO SALON_CLASE(SALON_CLASE."TAMAÑO",SALON_CLASE.NUMERO_SAL,SALON_CLASE.NOMBRE_CLA) VALUES (TAM,NUM_SAL,NOM_CLA);
DBMS_OUTPUT.PUT_LINE('LOS DATOS ENSERTADOS');
ELSIF(NUM3!=0 AND NUM4!=0) THEN
DBMS_OUTPUT.PUT_LINE('LOS DATOS YA EXISTE');
END IF;
END;
```

Por último se ejecuta el procedure("proceso ") con el siguiente comando . y según los datos que se tenga en las distintas tablas ("salón","clase" y "salón\_clase"),se dará un resultado de ingreso ,que el dato no existe o que los datos ya se ingresaron .

```
execute EJERCICIO4(20,302,'MATE');
```

El código completo queda de la siguiente manera .

```
CREATE OR REPLACE PROCEDURE EJERCICIO4(
TAM SALON_CLASE."TAMAÑO"%TYPE,
NUM_SAL SALON_CLASE.NUMERO_SAL%TYPE,
NOM_CLA SALON_CLASE.NOMBRE_CLA%TYPE
) IS
NUM1 INTEGER;
NUM2 INTEGER;
NUM3 INTEGER;
NUM4 INTEGER;
BEGIN
NUM1:=0;
NUM2:=0;
SELECT COUNT(*) INTO NUM1 FROM SALON WHERE NUMERO_SALON=NUM_SAL;
SELECT COUNT(*) INTO NUM2 FROM CLASE WHERE NOMBRE=NOM_CLA;
SELECT COUNT(*) INTO NUM3 FROM SALON_CLASE WHERE NOMBRE_CLA=NOM_CLA;
SELECT COUNT(*) INTO NUM4 FROM SALON_CLASE WHERE NUMERO_SAL=NUM_SAL;

IF(NUM1 = 0) THEN
DBMS_OUTPUT.PUT_LINE('EL DATO DEL CAMPO SALON NO EXITE ');
ELSIF(NUM2 =0) THEN
DBMS_OUTPUT.PUT_LINE('EL DATO DEL CAMPO CLASE NO EXITE ');
ELSIF(NUM3=0 AND NUM4=0) THEN
INSERT INTO SALON_CLASE(SALON_CLASE."TAMAÑO",SALON_CLASE.NUMERO_SAL,SALON_CLASE.NOMBRE_CLA) VALUES (TAM,NUM_SAL,NOM_CLA);
DBMS_OUTPUT.PUT_LINE('LOS DATOS ENSERTADOS');
ELSIF(NUM3!=0 AND NUM4!=0) THEN
DBMS_OUTPUT.PUT_LINE('LOS DATOS YA EXISTE');
END IF;
END;
```

Cree una función que devuelva el nombre de la clase, hora y fecha en que se

Reservar por parte del socio y total de horas (por todas las clases) que reserva dicho socio. Se ingresa como valor de entrada DNI\_SOCIO.

### Solución

Con la corrección que se hizo en clase .en el que el esquema clase que se tiene no se puede tener el total de hora que ha tenido el socio el ejercicio quedaría de la siguiente manera

Se crea un tipo de objeto con el nombre RESERVAS\_SOCIOS con los que se requiere en el ejercicio que son hora fecha y nombre de la clase a la que el socio hizo la reserva .

```
CREATE OR REPLACE TYPE RESERVAS_SOCIOS AS OBJECT(  
  HORA VARCHAR(30),  
  FECHA VARCHAR(30),  
  NOMBRECLASE VARCHAR(30)  
);|
```

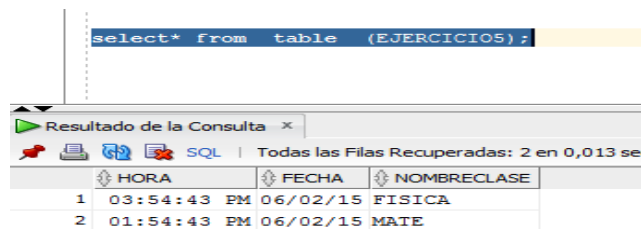
Luego se hace un objeto derivado que me servirá para meter los datos en el tipo objeto de la siguiente manera .

```
CREATE OR REPLACE TYPE DATOSRE AS TABLE OF RESERVAS_SOCIOS ;
```

Se crea la función que me va retornar el objeto derivado que me va a ayudar a ingresar los datos al tipo objeto. se crea una variable global que me servirá para meter los datos a tipo objeto reserva\_socio luego en el bloque begin se usa un select(cast(multiset que me permitirá meter toda la consulta que trae varios atributos en un solo tipo de dato que en este caso es un tipo objeto que tiene tres columnas . en la consulta que se hizo se utilizó un substr para fragmentar o partir un tipo de dato que en mi caso tengo un tipo de dato timestamp llamado hora\_fecha que me tiene la hora y la fecha en la que se hizo la reserva toda la consulta se metió en el objeto derivado con un "as" y se insertó en la variable cant .se usa la tabla dual que es una tabla temporal y por último se retorna el dato a la que le metió el objeto que fue cant y se retorna para poder mostrar la consulta .

```
CREATE OR REPLACE FUNCTION EJERCICIOS  
RETURN DATOSRE IS  
CANT DATOSRE;  
BEGIN  
  select  
  cast( multiset(  
  SELECT (SUBSTR(R.FECHA_HORA,9,9)||SUBSTR(R.FECHA_HORA,25,9)) "HORA",SUBSTR(R.FECHA_HORA,1,8) "FECHA",A.NOMBRECLASE FROM RESERVA R,ASISTE A  
  WHERE R.DNI_SOCIO2=A.DNISOCIO  
  ) as DATOSRE) INTO CANT  
  FROM DUAL;  
  RETURN CANT;  
END EJERCICIOS;
```

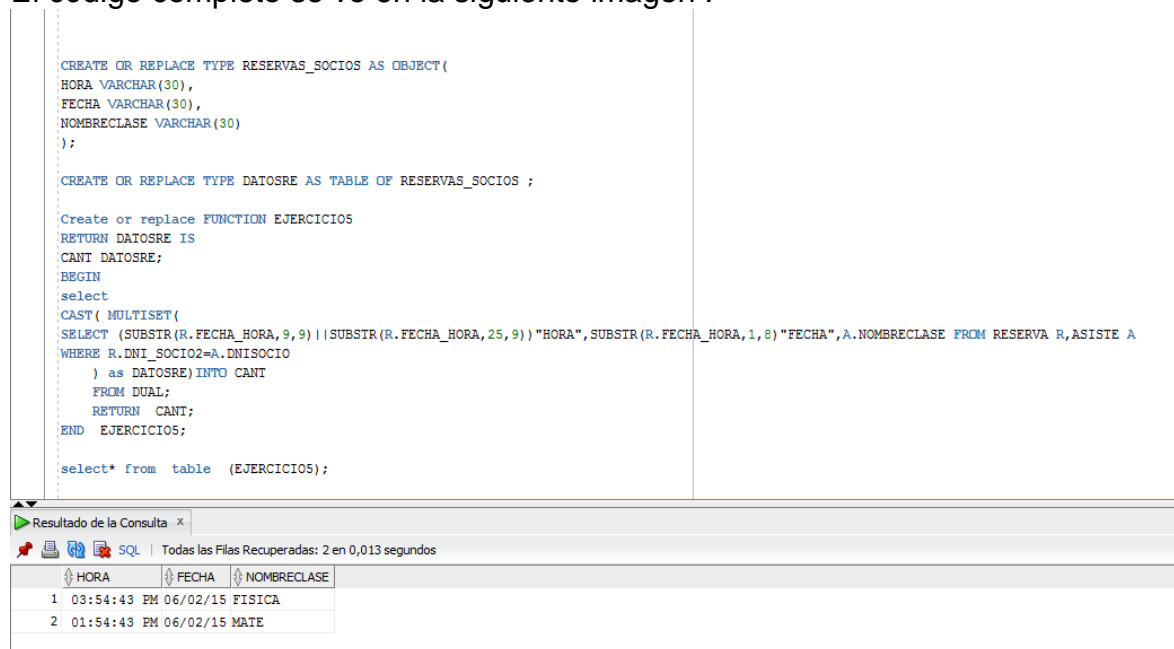
Se ejecuta la consulta y se muestra de la siguiente forma.



The screenshot shows a SQL query execution window. At the top, a query is entered: `select* from table (EJERCICIO5);`. Below the query, the window title is "Resultado de la Consulta x". The status bar indicates "Todas las Filas Recuperadas: 2 en 0,013 se". The result is displayed in a table with three columns: HORA, FECHA, and NOMBRECLASE. There are two rows of data.

	HORA	FECHA	NOMBRECLASE
1	03:54:43 PM	06/02/15	FISICA
2	01:54:43 PM	06/02/15	MATE

El código completo se ve en la siguiente imagen .



The screenshot shows a SQL script editor with the following code:

```
CREATE OR REPLACE TYPE RESERVAS_SOCIOS AS OBJECT(  
HORA VARCHAR(30),  
FECHA VARCHAR(30),  
NOMBRECLASE VARCHAR(30)  
);  
  
CREATE OR REPLACE TYPE DATOSRE AS TABLE OF RESERVAS_SOCIOS ;  
  
Create or replace FUNCTION EJERCICIO5  
RETURN DATOSRE IS  
CANT DATOSRE;  
BEGIN  
select  
CAST( MULTISET(  
SELECT (SUBSTR(R.FECHA_HORA,9,9) || SUBSTR(R.FECHA_HORA,25,9)) "HORA", SUBSTR(R.FECHA_HORA,1,8) "FECHA", A.NOMBRECLASE FROM RESERVA R, ASISTE A  
WHERE R.DNI_SOCIO2=A.DNISOCIO  
) as DATOSRE) INTO CANT  
FROM DUAL;  
RETURN CANT;  
END EJERCICIO5;  
  
select* from table (EJERCICIO5);
```

Below the script, the execution result is shown in a window titled "Resultado de la Consulta x". The status bar indicates "Todas las Filas Recuperadas: 2 en 0,013 segundos". The result is displayed in a table with three columns: HORA, FECHA, and NOMBRECLASE. There are two rows of data.

	HORA	FECHA	NOMBRECLASE
1	03:54:43 PM	06/02/15	FISICA
2	01:54:43 PM	06/02/15	MATE