

Manual Scripts 3 2022

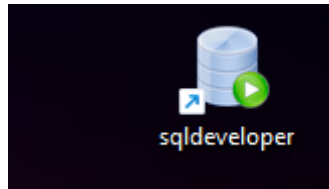
6 diciembre

Creado por: Henry Asdrubal Rodriguez Morales

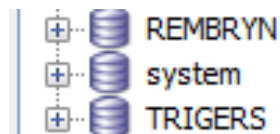


Paso 1

Ejecutar el sql developer.

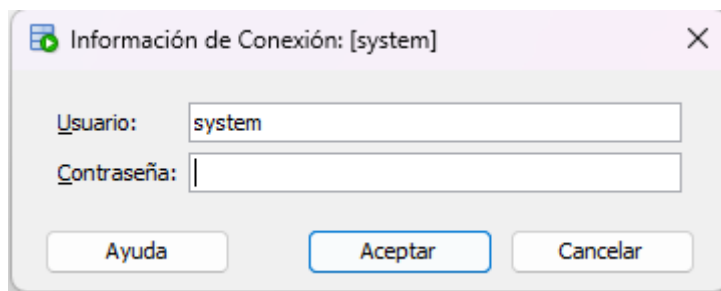


Al ingresar Logearse con el usuario administrador, el usuario que se registro al momento de instalar el programa.

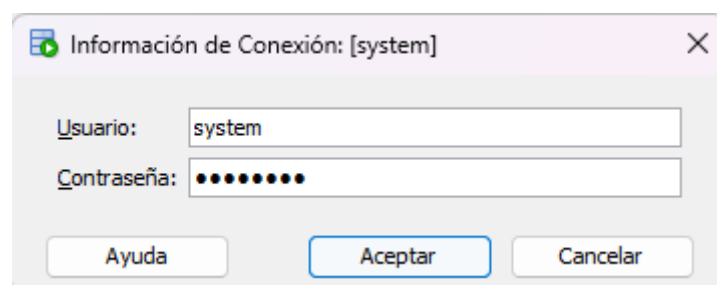


En este caso el usuario administrador se llama system.

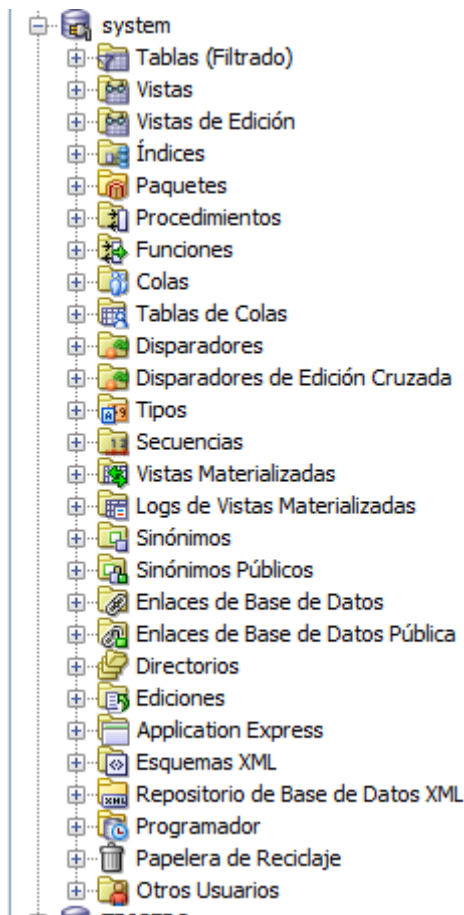
Al darle click al simbolo + aparece una ventan para ingresar la contraseña del administrador registrada anteriormente al momento de instalacion del gestor de base de datos.



Escribir su contraseña y hacer click en aceptar.

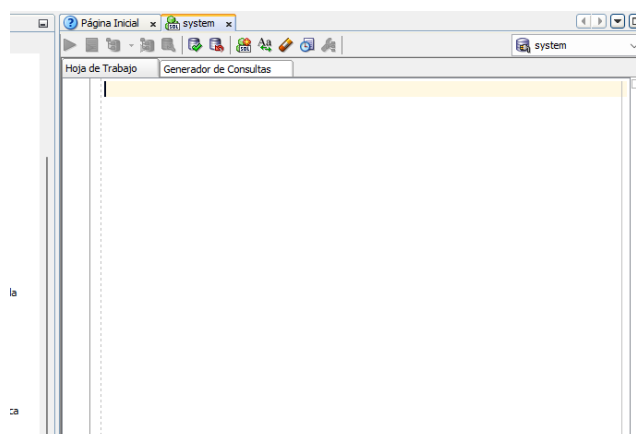


Al ingresar al usuario administrador se despliegan diferentes menus.



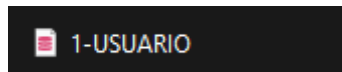
Paso 2

Al ingresar se vera la consola para poder escribir los scripts que se requieran



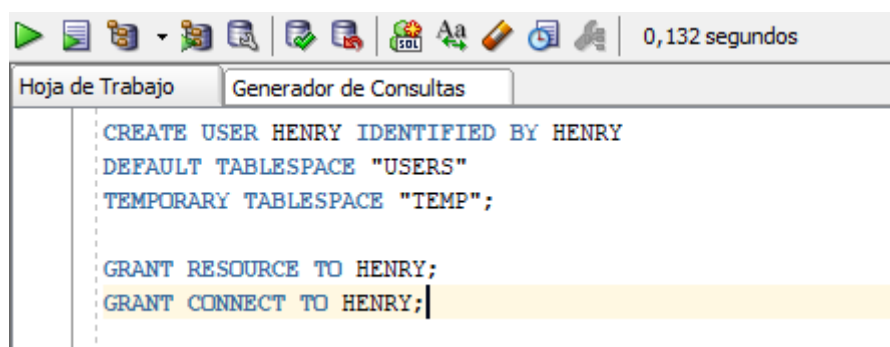
Allí se copiará el script para la creación de un usuario

Vamos al **archivo** USUARIO

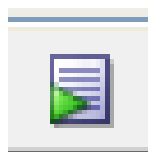


```
1-USUARIO.SQL
1  CREATE USER HENRY IDENTIFIED BY HENRY
2  DEFAULT TABLESPACE "USERS"
3  TEMPORARY TABLESPACE "TEMP";
4
5  GRANT RESOURCE TO HENRY;
6  GRANT CONNECT TO HENRY;
```

Lo copiaremos y pegaremos y queda de la siguiente forma:



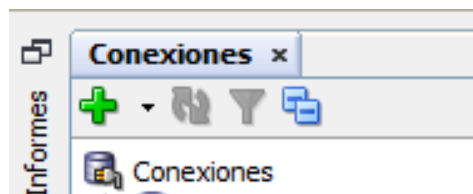
Y le damos ejecutar en el simbolo de la parte superior



Al ejecutarlo nos debe aparecer el mensaje de éxito en la creación del usuario en la parte inferior.



En la parte superior izquierda se encuentra el símbolo **+** que significa añadir nueva conexión con el usuario creado



Nos aparece una ventana

En nombre de Conexión le vamos a dar el nombre que queramos en nuestro caso le pondremos HENRY,

En usuario le pondremos el usuario anteriormente creado, que también le pusimos HENRY igualmente la contraseña.

Nombre de Conexión	HENRY
Usuario	HENRY
Contraseña	•••••

Al llenar los campos con nuestro usuario y contraseña le hacemos click en probar, en la parte inferior.

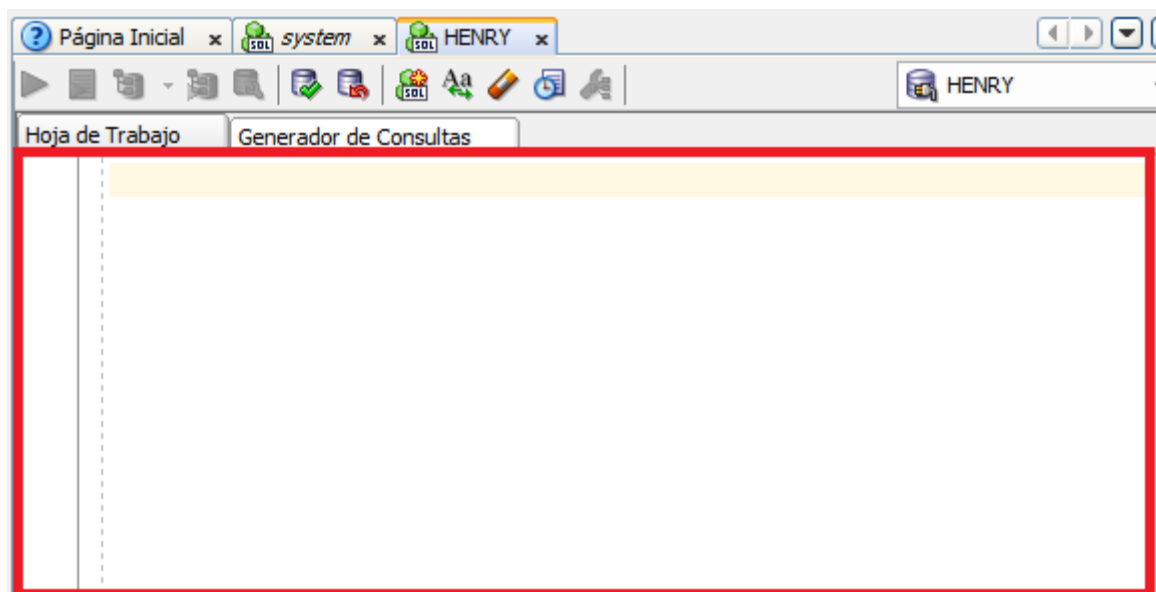
Nombre de Conexión	HENRY		
Usuario	HENRY		
Contraseña	•••••		
<input type="checkbox"/> Guardar Contraseña	<input checked="" type="checkbox"/> Color de Conexión		
Oracle			
Tipo de Conexión	Básico	Rol	valor por defecto
Nombre del Host	localhost		
Puerto	1521		
<input checked="" type="radio"/> SID	xe		
<input type="radio"/> Nombre del Servicio			
<div>Guardar Borrar Probar Conectar Cancelar</div>			

Al hacerle click podemos ver el estado en la parte inferior izquierda

Estado: Correcto					
Ayuda	Guardar	Borrar	Probar	Conectar	Cancelar

Si nos aparece correcto le podemos hacer click en Conectar.

Nos aparecera una nueva base de dato en la parte izquierda llamada HENRY y nos aparece la consola para escribir nuestros scripts.



Paso 3

Allí vamos a escribir nuestro scripts DDL para crear nuestras tablas.

Vamos al archivo DDL y encontraremos el script para crear nuestras tablas.

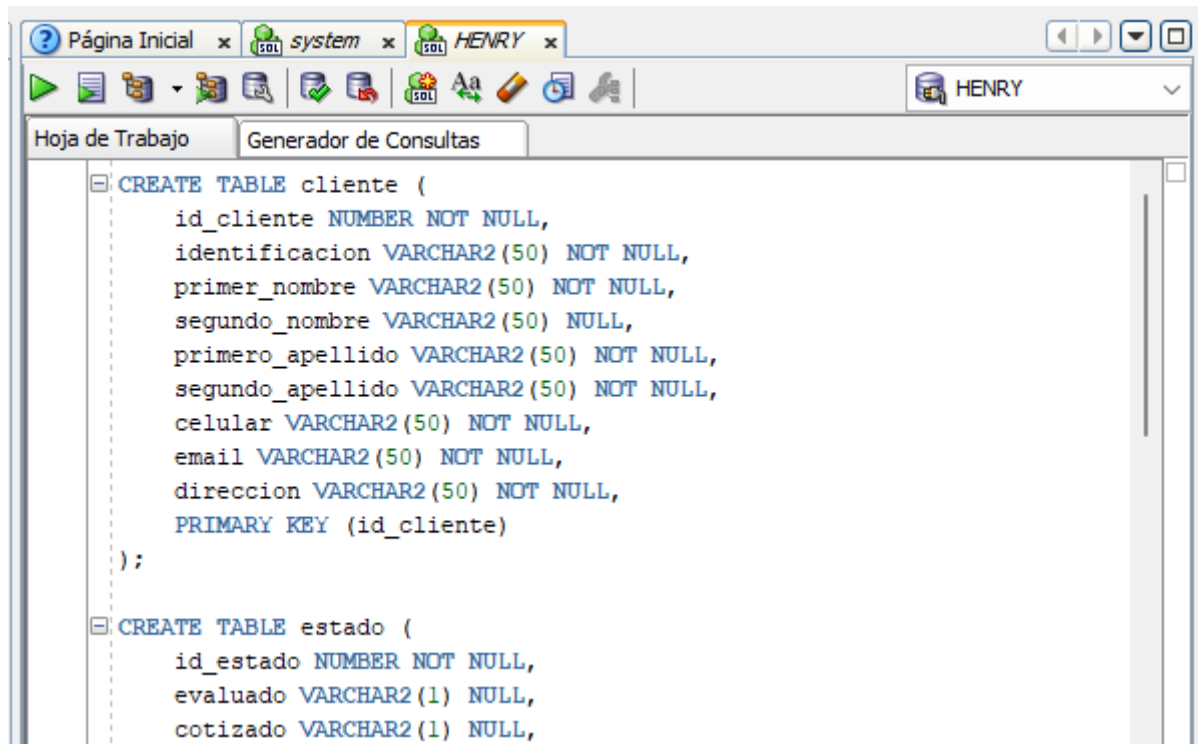
2-DDL

Allí encontraremos las líneas de código y las copiaremos todas usando control+A

```
2-DDL.sql x
2-DDL.sql
1 CREATE TABLE cliente (
2   ...id_cliente NUMBER NOT NULL,
3   ...identificacion VARCHAR2(50) NOT NULL,
4   ...primer_nombre VARCHAR2(50) NOT NULL,
5   ...segundo_nombre VARCHAR2(50) NULL,
6   ...primero_apellido VARCHAR2(50) NOT NULL,
7   ...segundo_apellido VARCHAR2(50) NOT NULL,
8   ...celular VARCHAR2(50) NOT NULL,
9   ...email VARCHAR2(50) NOT NULL,
10  ...direccion VARCHAR2(50) NOT NULL,
11  ...PRIMARY KEY (id_cliente)
12 );
13
14 CREATE TABLE estado (
15   ...id_estado NUMBER NOT NULL,
16   ...evaluado VARCHAR2(1) NULL,
```

Copiamos todas las líneas anteriores y las pegamos en la consola del usuario HENRY,

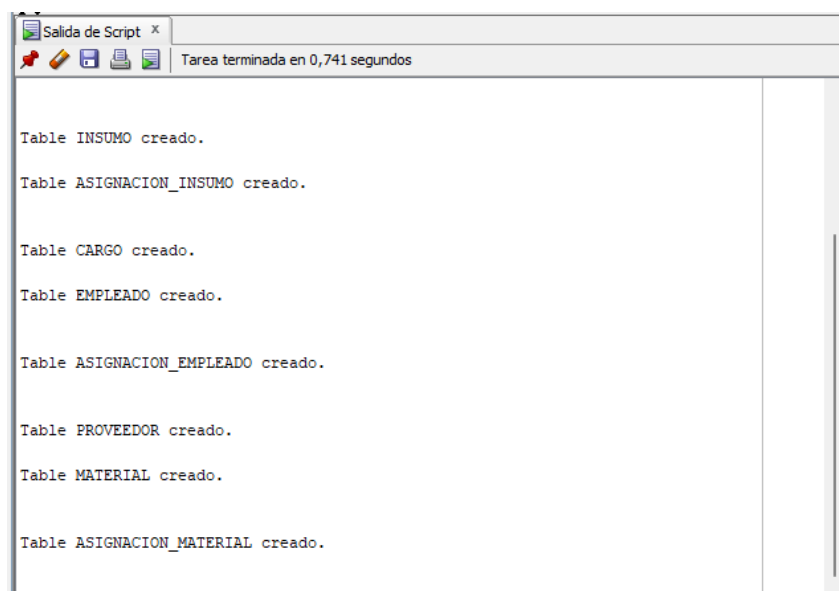
nos debe quedar de la siguiente manera.



```
CREATE TABLE cliente (  
    id_cliente NUMBER NOT NULL,  
    identificacion VARCHAR2(50) NOT NULL,  
    primer_nombre VARCHAR2(50) NOT NULL,  
    segundo_nombre VARCHAR2(50) NULL,  
    primero_apellido VARCHAR2(50) NOT NULL,  
    segundo_apellido VARCHAR2(50) NOT NULL,  
    celular VARCHAR2(50) NOT NULL,  
    email VARCHAR2(50) NOT NULL,  
    direccion VARCHAR2(50) NOT NULL,  
    PRIMARY KEY (id_cliente)  
);  
  
CREATE TABLE estado (  
    id_estado NUMBER NOT NULL,  
    evaluado VARCHAR2(1) NULL,  
    cotizado VARCHAR2(1) NULL,
```

Al tenerlos todos en la consola, le damos click en el simbolo  en la parte superior para ejecutar toda la consola.

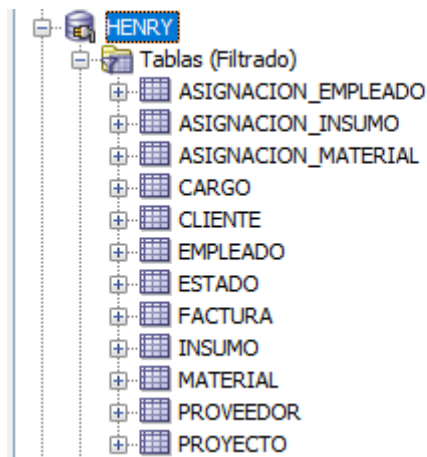
Al hacerle click nos debe aparecer los mensajes de existo al crear las tablas.



```
Salida de Script x  
Tarea terminada en 0,741 segundos  
  
Table INSUMO creado.  
Table ASIGNACION_INSUMO creado.  
  
Table CARGO creado.  
Table EMPLEADO creado.  
  
Table ASIGNACION_EMPLEADO creado.  
  
Table PROVEEDOR creado.  
Table MATERIAL creado.  
  
Table ASIGNACION_MATERIAL creado.
```


En la parte izquierda nos aparecera la base de datos y dentro nos aparece las tablas

creadas.




Despues de crear las tablas vamos a ejecutar el script de DML la inserción de datos a esas tablas que hemos creado, copia todas las lineas que a continuación te presentan.

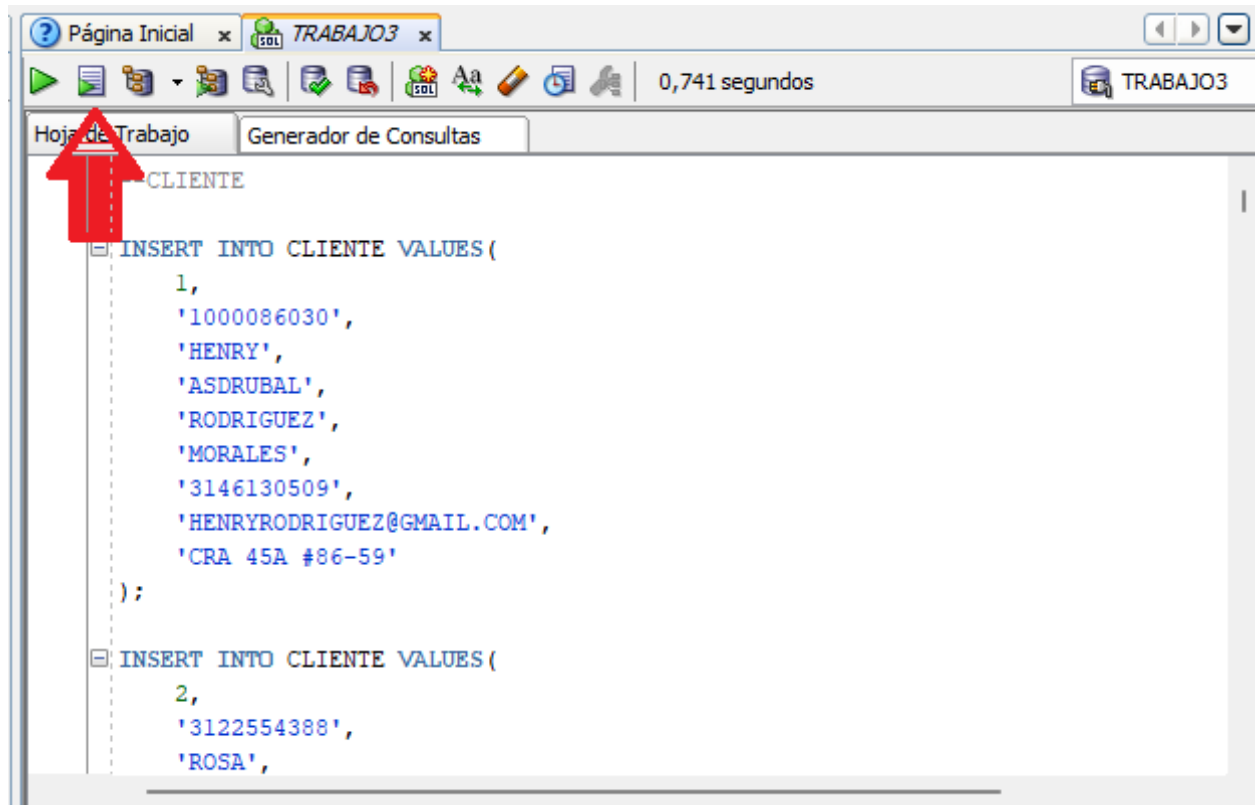
Paso 4

En el **archivo** DML se encuentran las lineas que se deben copias todas de igual manera control + A y pegar en la consola.

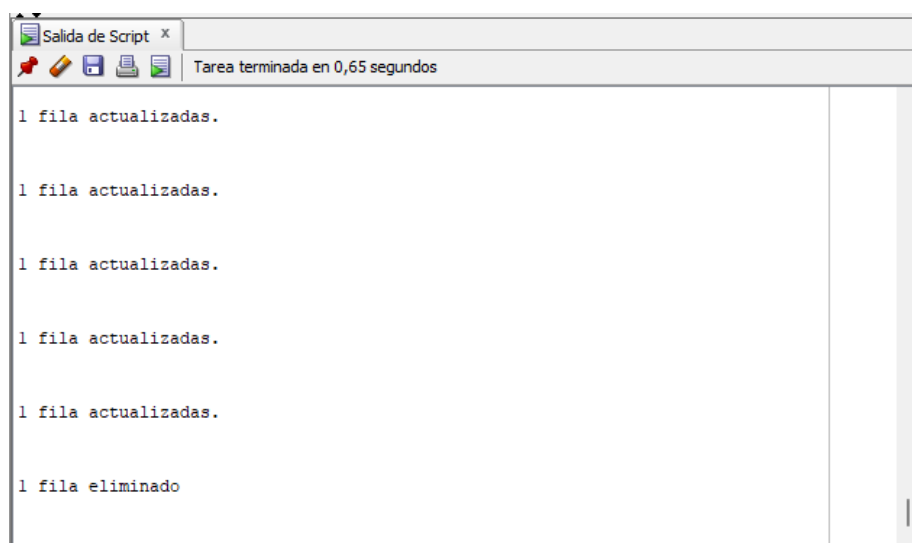
3-DML

```
3-DML.SQL X
3-DML.SQL
1  --CLIENTE
2
3  INSERT INTO CLIENTE VALUES(
4      1,
5      '1000086030',
6      'HENRY',
7      'ASDRUBAL',
8      'RODRIGUEZ',
9      'MORALES',
10     '3146130509',
11     'HENRYRODRIGUEZ@GMAIL.COM',
12     'CRA 45A #86-59'
13 );
14
```

Le damos en el simbolo de  ejecutar todo el script que tenemos pegado en nuestra consola de la siguiente manera.



De igual manera al hacerle click podemos observar los mensajes de éxito que salen cuando ejecutamos el script.




Paso 5

Despues de generar nuestros registros y tablas es hora de generar nuestras funciones y procedimientos para manejar la información.

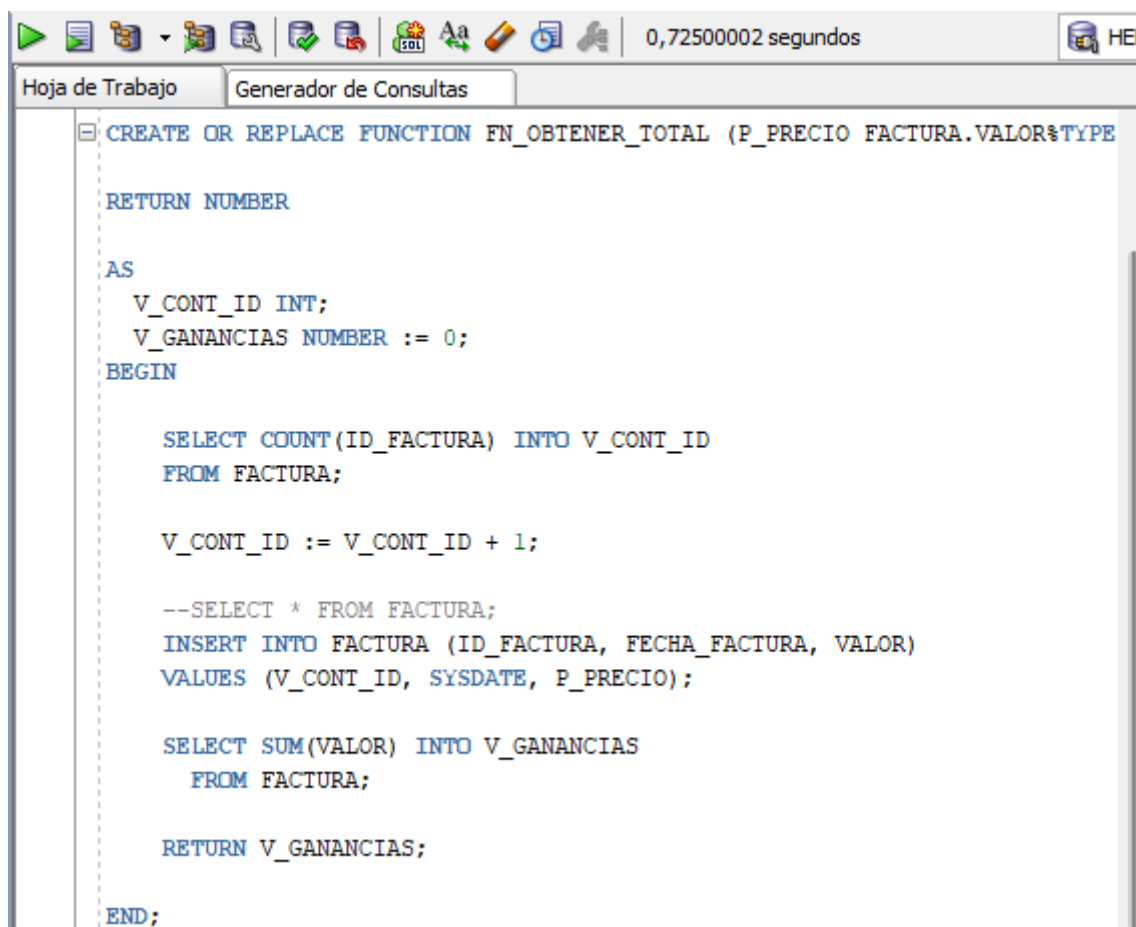
En la consola de escritura de scripts vamos a pegar las siguientes funciones:

Vamos al archivo ARTEFACTOS y copiamos uno por uno

 4-ARTEFACTOS

```
-- *****
-- FUNCIÓN PARA SABER EL TOTAL DE DINERO QUE LA EMPRESA TIENE AL MOMENTO MAS LA ULTIMA
-- *****

CREATE OR REPLACE FUNCTION FN_OBTENER_TOTAL (P_PRECIO FACTURA.VALOR%TYPE)
RETURN NUMBER
AS
  V_CONT_ID INT;
  V_GANANCIAS NUMBER := 0;
BEGIN
  -- SELECT COUNT(ID_FACTURA) INTO V_CONT_ID
  -- FROM FACTURA;
  -- V_CONT_ID := V_CONT_ID + 1;
  -- --SELECT * FROM FACTURA;
  -- INSERT INTO FACTURA (ID_FACTURA, FECHA_FACTURA, VALOR)
  -- VALUES (V_CONT_ID, SYSDATE, P_PRECIO);
  -- SELECT SUM(VALOR) INTO V_GANANCIAS
  -- FROM FACTURA;
  -- RETURN V_GANANCIAS;
END;
```



```
CREATE OR REPLACE FUNCTION FN_OBTENER_TOTAL (P_PRECIO FACTURA.VALOR%TYPE)
RETURN NUMBER
AS
    V_CONT_ID INT;
    V_GANANCIAS NUMBER := 0;
BEGIN
    SELECT COUNT(ID_FACTURA) INTO V_CONT_ID
    FROM FACTURA;

    V_CONT_ID := V_CONT_ID + 1;

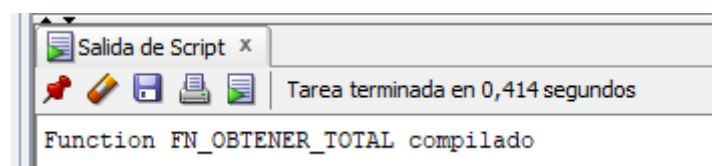
    --SELECT * FROM FACTURA;
    INSERT INTO FACTURA (ID_FACTURA, FECHA_FACTURA, VALOR)
    VALUES (V_CONT_ID, SYSDATE, P_PRECIO);

    SELECT SUM(VALOR) INTO V_GANANCIAS
    FROM FACTURA;

    RETURN V_GANANCIAS;
END;
```

Cuando lo peguemos le vamos a dar en el simbolo de ejecutar consola de scripts 

Y nos debe aparecer nuestro mensaje de éxito de ejecución



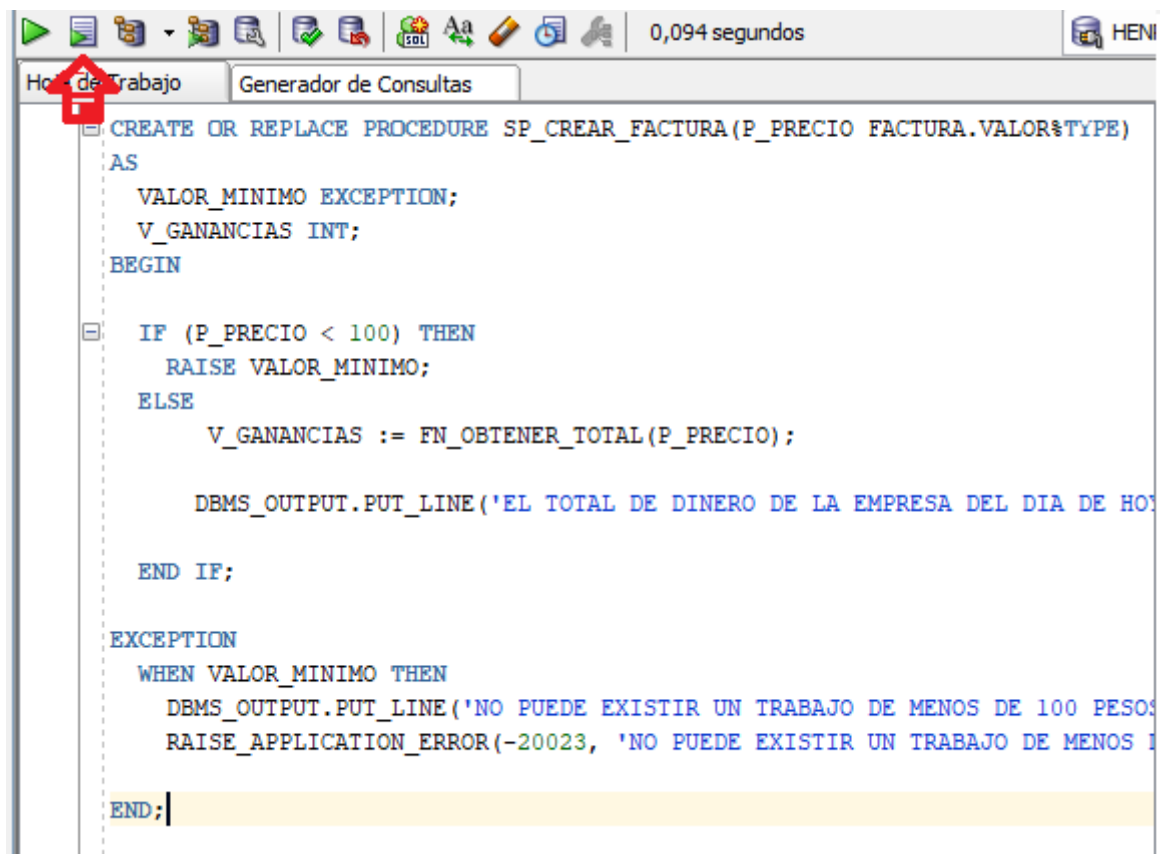
Crearemos nuestro procedimiento que llama nuestra funcion creada anteriormente.

Ejecutamos todo el siguiente procedimiento que llama nuestra funcion:

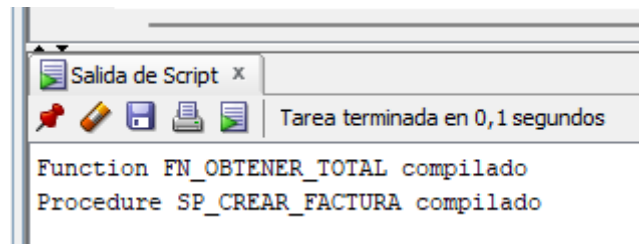
```

CREATE OR REPLACE PROCEDURE SP_CREAR_FACTURA(P_PRECIO FACTURA.VALOR%TYPE)
AS
  VALOR_MINIMO EXCEPTION;
  V_GANANCIAS INT;
BEGIN
  IF (P_PRECIO < 100) THEN
    RAISE VALOR_MINIMO;
  ELSE
    V_GANANCIAS := FN_OBTENER_TOTAL(P_PRECIO);
    DBMS_OUTPUT.PUT_LINE('EL TOTAL DE DINERO DE LA EMPRESA DEL DIA DE HOY ' || SYSDATE || ' ES ');
  END IF;
EXCEPTION
  WHEN VALOR_MINIMO THEN
    DBMS_OUTPUT.PUT_LINE('NO PUEDE EXISTIR UN TRABAJO DE MENOS DE 100 PESOS COLOMBIANOS');
    RAISE_APPLICATION_ERROR(-20023, 'NO PUEDE EXISTIR UN TRABAJO DE MENOS DE 100');
END;

```



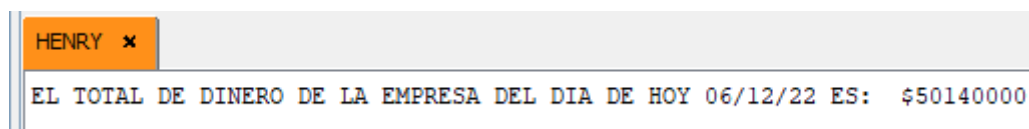
Nos debe mostrar el mensaje de éxito de creación



Tenemos 2 artefactos dentro de nuestra base de datos, una funcion y un procedimiento. Ahora vamos a ejecutarlos y ver su función.

```
--  
*****  
*****  
--NUEVA FACTURA DE 15000000  
EXECUTE SP_CREAR_FACTURA(15000000);
```

Ejecución:



Juego

Vamos al archiv JUEGO



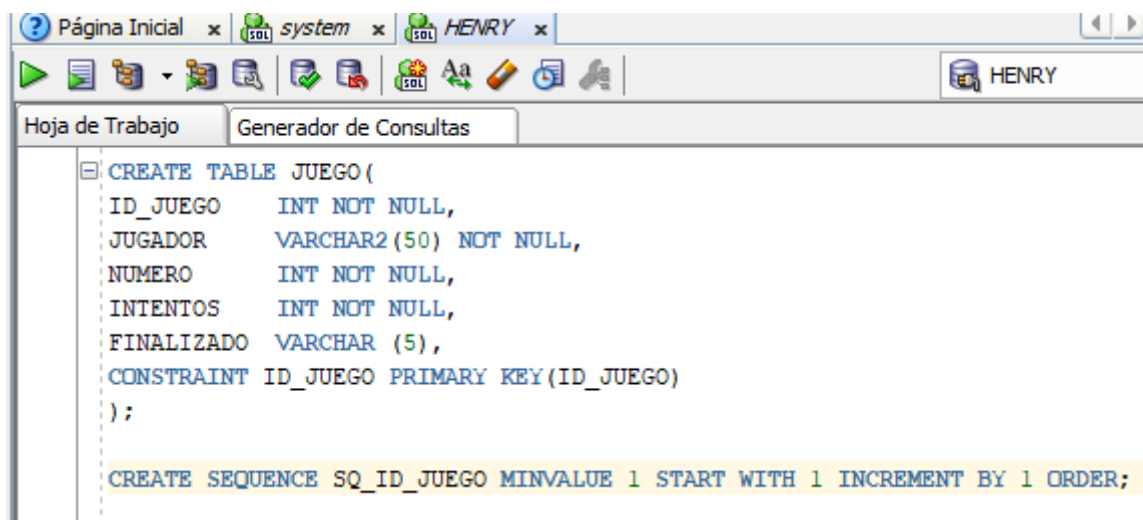
Tendremos nuestro script para realizar el siguiente juego. Crearemos nuestras tablas que aparecen en el archivo

```

JUEGO.SQL X
JUEGO.SQL
1  CREATE TABLE JUEGO(
2  ID_JUEGO      INT NOT NULL,
3  JUGADOR       VARCHAR2(50) NOT NULL,
4  NUMERO        INT NOT NULL,
5  INTENTOS      INT NOT NULL,
6  FINALIZADO    VARCHAR (5),
7  CONSTRAINT ID_JUEGO PRIMARY KEY(ID_JUEGO)
8  );
9
10 CREATE SEQUENCE SQ_ID_JUEGO MINVALUE 1 START WITH 1 INCREMENT BY 1 ORDER;
11

```

Copiamos nuestro script en nuestra consola y lo ejecutamos.



```

CREATE TABLE JUEGO(
  ID_JUEGO      INT NOT NULL,
  JUGADOR       VARCHAR2(50) NOT NULL,
  NUMERO        INT NOT NULL,
  INTENTOS      INT NOT NULL,
  FINALIZADO    VARCHAR (5),
  CONSTRAINT ID_JUEGO PRIMARY KEY(ID_JUEGO)
);

CREATE SEQUENCE SQ_ID_JUEGO MINVALUE 1 START WITH 1 INCREMENT BY 1 ORDER;

```

Nos debe mostrar nuestro mensaje de éxito.

Despues copiamos nuestra funcion

```

12 CREATE OR REPLACE PROCEDURE SP_FINALIZAR_JUEGO(P_JUGADOR JUEGO.ID_JUEGO%TYPE)
13
14 AS
15   V_FINALIZADO JUEGO.FINALIZADO%TYPE := 'S';
16
17 BEGIN
18
19   --UPDATE JUEGO
20   --SET FINALIZADO = V_FINALIZADO
21   --WHERE ID_JUEGO = P_JUGADOR;
22   DBMS_OUTPUT.PUT_LINE('JUEGO FINALIZADO');
23
24 END;
25
26

```

Y la ejecutamos, despues la otra funcion obtener numero secreto

```
CREATE OR REPLACE FUNCTION FN_OBTENER_NUMERO_SECRETO(P_JUGADOR JUEGO.ID_JUEGO%TYPE)
RETURN NUMBER
AS
  V_NUMERO NUMBER;
BEGIN
  SELECT NUMERO INTO V_NUMERO
  FROM JUEGO J
  WHERE J.ID_JUEGO = P_JUGADOR;
  RETURN V_NUMERO;
END;
```

Y la ejecutamos.

Luego nuestra otra funcion

```
CREATE OR REPLACE FUNCTION FN_INCREMENTAR_INTENTO (P_JUGADOR JUEGO.ID_JUEGO%TYPE)
RETURN NUMBER
AS
  V_INTENTOS NUMBER;
BEGIN
  SELECT INTENTOS INTO V_INTENTOS
  FROM JUEGO
  WHERE ID_JUEGO = P_JUGADOR;
  V_INTENTOS := V_INTENTOS + 1;
  UPDATE JUEGO
  SET INTENTOS = V_INTENTOS
  WHERE ID_JUEGO = P_JUGADOR;
  RETURN V_INTENTOS;
END;
```

Y la ejecutamos

Luego la otra funcion FN_OBTENER_JUEGO

```
CREATE OR REPLACE FUNCTION FN_OBTENER_JUEGO(P_JUGADOR VARCHAR)

RETURN JUEGO.ID_JUEGO%TYPE

AS
    V_JUEGO JUEGO.ID_JUEGO%TYPE;
    V_RANGO_MIN NUMBER := 1;
    V_RANGO_MAX NUMBER := 100;
    V_RANDOM NUMBER;

BEGIN

    SELECT J.ID_JUEGO INTO V_JUEGO
    FROM JUEGO J
    WHERE J.JUGADOR = P_JUGADOR AND J.FINALIZADO = 'N';

    RETURN V_JUEGO;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        V_RANDOM := ROUND(SYS.dbms_random.VALUE(V_RANGO_MIN, V_RANGO_MAX));
        SELECT SQ_ID_JUEGO.NEXTVAL INTO V_JUEGO FROM DUAL;

        INSERT INTO JUEGO (ID_JUEGO, JUGADOR, NUMERO, INTENTOS, FINALIZADO)
        VALUES (V_JUEGO, P_JUGADOR, V_RANDOM, 0, 'N');

        RETURN V_JUEGO;

END;
```

Luego copiamos la otra funcion FN_OBTENER_JUEGO

```

CREATE OR REPLACE FUNCTION FN_OBTENER_JUEGO(P_JUGADOR VARCHAR)
RETURN JUEGO.ID_JUEGO%TYPE
AS
  V_JUEGO JUEGO.ID_JUEGO%TYPE;
  V_RANGO_MIN NUMBER := 1;
  V_RANGO_MAX NUMBER := 100;
  V_RANDOM NUMBER;
BEGIN
  SELECT J.ID_JUEGO INTO V_JUEGO
  FROM JUEGO J
  WHERE J.JUGADOR = P_JUGADOR AND J.FINALIZADO = 'N';

  RETURN V_JUEGO;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
    V_RANDOM := ROUND(SYS.dbms_random.VALUE(V_RANGO_MIN, V_RANGO_MAX));
    SELECT SQ_ID_JUEGO.NEXTVAL INTO V_JUEGO FROM DUAL;

    INSERT INTO JUEGO (ID_JUEGO, JUGADOR, NUMERO, INTENTOS, FINALIZADO)
    VALUES (V_JUEGO, P_JUGADOR, V_RANDOM, 0, 'N');

    RETURN V_JUEGO;

END;

```

Y la ejecutamos.

Y por ultimo nuestro procedimiento el que llama todas las otras funciones

```
EXECUTE SP_JUGAR('HENRY',61);
```

```
El número ingresado (61) es mayor que el número secreto
```

Podemos seguir ingresando numeros hasta encontrar el indicado.

```
EXECUTE SP_JUGAR('HENRY',55);
```

```
El número ingresado (55) es mayor que el número secreto
```