

3.4.2 Procesamiento de Datos. Programación.

JESÚS TERINO RODRIGUEZ



PROGRAM A	ENTRADA/SALIDA + FUNCIÓN	ESTRUCTURAS DE DATOS	CONTROL DE FLUJO	PROCEDIMIENTO Y FUNCIONES	SELECT , INSERT,DELETE Y UPDATE	CURSORES
1	SI	SIMPLE	BUCLE	SI	SELECT	SI
2	SI	SIMPLE	DIRECTO	SI	INSERT	NO
3	NO	SIMPLE	DIRECTO	SI	UPDATE	NO
4	NO	SIMPLE	CONDICIONAL	SI	DELETE	NO
5	SI	SIMPLE	BUCLE	SI	SELECT	SI
6	SI	SIMPLE	CONDICIONAL	SI	INSERT	NO
7	NO	SIMPLE	BUCLE	SI	UPDATE Y SLECT	SI
8	SI	COMPUESTA	DIRECTO	SI	INSERT Y SELECT	NO
9	SI	SIMPLE	CONDICIONAL	SI	DELETE Y SELECT	NO
10	SI	SIMPLE	BUCLE	SI	SELECT	SI

Programa

```
DECLARE

    TYPE t_cursor IS REF CURSOR;

    c_jugadores t_cursor;

    v_dorsal JUGADORES.DORSAL%TYPE;

    v_nombre JUGADORES.NOMBRE%TYPE;

    v_apellidos JUGADORES.APELLIDOS%TYPE;

    v_max_salario JUGADORES.SALARIO%TYPE;

    FUNCTION obtener_max_salario RETURN NUMBER IS

        v_max_salario JUGADORES.SALARIO%TYPE;

    BEGIN

        SELECT MAX(SALARIO) INTO v_max_salario FROM JUGADORES;

        RETURN v_max_salario;

    END obtener_max_salario;
```

```
BEGIN

    v_max_salario := obtener_max_salario;

    OPEN c_jugadores FOR

        SELECT DORSAL, NOMBRE, APELLIDOS

        FROM JUGADORES

        WHERE SALARIO = v_max_salario;

    LOOP

        FETCH c_jugadores INTO v_dorsal, v_nombre, v_apellidos;

        EXIT WHEN c_jugadores%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_dorsal || ' | NOMBRE: ' || v_nombre || ' | APELLIDOS: ' || v_apellidos);

    END LOOP;

    CLOSE c_jugadores;

END;
```

/

CONSULTA CON FUNCIÓN Y CURSOR

ORACLE® Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

☒ Confirmación Automática Mostrar: 10

```
DECLARE
  TYPE t_cursor IS REF CURSOR;
  c_jugadores t_cursor;
  v_dorsal JUGADORES.DORSAL%TYPE;
  v_nombre JUGADORES.NOMBRE%TYPE;
  v_apellidos JUGADORES.APELLIDOS%TYPE;
  v_max_salario JUGADORES.SALARIO%TYPE;

  FUNCTION obtener_max_salario RETURN NUMBER IS
    v_max_salario JUGADORES.SALARIO%TYPE;
  BEGIN
    SELECT MAX(SALARIO) INTO v_max_salario FROM JUGADORES;
    RETURN v_max_salario;
  END obtener_max_salario;

BEGIN
  -- Almacenar el resultado de la función en una variable
  v_max_salario := obtener_max_salario;

  -- Usar la variable en la consulta
  OPEN c_jugadores FOR
    SELECT DORSAL, NOMBRE, APELLIDOS
    FROM JUGADORES
    WHERE SALARIO = v_max_salario;
```

UTILIZACION DE FUNCION

RECURSIVIDAD CON FOR

SALIDA DEL PROGRAMA

Resultados Explicar Describir SQL Guardado Historial

DORSAL: 12 | NOMBRE: JESUS | APELLIDOS: TERINO

sentencia procesada.

1,00 segundos

Programa con:
Entrada salida
Estructura de
datos simple
Recursividad
Procedimientos/Fu
nciones
Selección
Cursores

Programa

```
DECLARE

    v_dorsal JUGADORES.DORSAL%TYPE;

    v_nombre JUGADORES.NOMBRE%TYPE := 'Carlos';

    v_apellidos JUGADORES.APELLIDOS%TYPE := 'López';

    v_salario JUGADORES.SALARIO%TYPE := 35000;

    v_duracion_contrato JUGADORES.DURACION_CONTRATO%TYPE := 3;
    -- Añade una duración de contrato

FUNCTION generar_dorsal RETURN NUMBER IS

    v_max_dorsal JUGADORES.DORSAL%TYPE;

BEGIN

    SELECT COALESCE(MAX(DORSAL), 0) INTO v_max_dorsal FROM
JUGADORES;

    RETURN v_max_dorsal + 1;

END generar_dorsal;

BEGIN

    v_dorsal := generar_dorsal;
```

```
BEGIN

    v_dorsal := generar_dorsal;

INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO,
DURACION_CONTRATO)

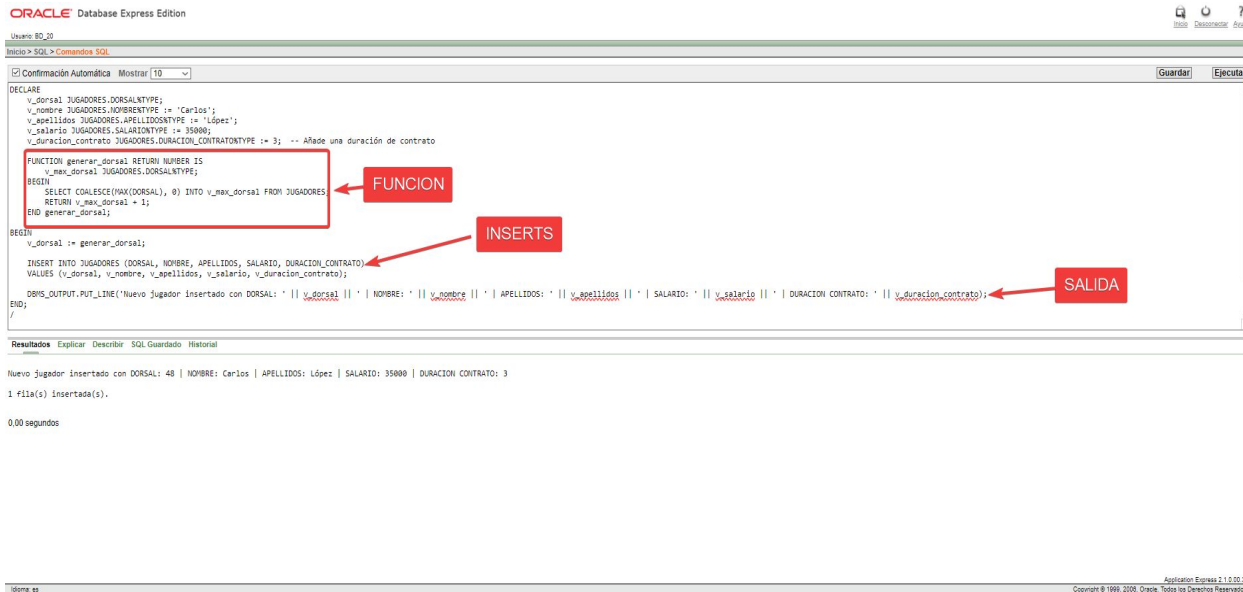
VALUES (v_dorsal, v_nombre, v_apellidos, v_salario, v_duracion_contrato);

    DBMS_OUTPUT.PUT_LINE('Nuevo jugador insertado con DORSAL: ' || v_dorsal || ' |
NOMBRE: ' || v_nombre || ' | APELLIDOS: ' || v_apellidos || ' | SALARIO: ' || v_salario
|| ' | DURACION CONTRATO: ' || v_duracion_contrato);

END;

/
```

INSERCIÓN CON FUNCIÓN



The screenshot shows the Oracle Database Express Edition interface. The main window displays a PL/SQL script. A red box highlights the function definition, with a red arrow pointing to it from a label 'FUNCION'. Another red arrow points from a label 'INSERTS' to the INSERT statement. A third red arrow points from a label 'SALIDA' to the DBMS_OUTPUT.PUT_LINE statement. The script defines a function to generate a unique dorsal number and inserts a player record. The results pane at the bottom shows the successful execution of the script.

```
DECLARE
  v_dorsal JUGADORES.DORSAL%TYPE;
  v_nombre JUGADORES.NOMBRE%TYPE := 'Carlos';
  v_apellidos JUGADORES.APELLIDOS%TYPE := 'López';
  v_salario JUGADORES.SALARIO%TYPE := 35000;
  v_duracion_contrato JUGADORES.DURACION_CONTRATO%TYPE := 3; -- Añade una duración de contrato

  FUNCTION generar_dorsal RETURN NUMBER IS
    v_max_dorsal JUGADORES.DORSAL%TYPE;
  BEGIN
    SELECT COALESCE(MAX(DORSAL), 0) INTO v_max_dorsal FROM JUGADORES;
    RETURN v_max_dorsal + 1;
  END generar_dorsal;
BEGIN
  v_dorsal := generar_dorsal;

  INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO, DURACION_CONTRATO)
  VALUES (v_dorsal, v_nombre, v_apellidos, v_salario, v_duracion_contrato);

  DBMS_OUTPUT.PUT_LINE('Nuevo jugador insertado con DORSAL: ' || v_dorsal || ' | NOMBRE: ' || v_nombre || ' | APELLIDOS: ' || v_apellidos || ' | SALARIO: ' || v_salario || ' | DURACION CONTRATO: ' || v_duracion_contrato);
END;
```

Results

Results	Explicar	Describir	SQL Guardado	Historial
Nuevo jugador insertado con DORSAL: 48 NOMBRE: Carlos APELLIDOS: López SALARIO: 35000 DURACION CONTRATO: 3				
1 fila(s) insertada(s).				
0.00 segundos				

Programa con:
Entrada salida
Estructura de
datos simple
Procedimientos/Fu
nciones
Inserción

Programa

```
DECLARE
```

```
    PROCEDURE actualizar_salario(p_dorsal IN NUMBER, p_nuevo_salario IN NUMBER) IS
```

```
    BEGIN
```

```
        UPDATE JUGADORES
```

```
        SET SALARIO = p_nuevo_salario
```

```
        WHERE DORSAL = p_dorsal;
```

```
        DBMS_OUTPUT.PUT_LINE('Salario actualizado.');
```

```
    COMMIT;
```

```
END actualizar_salario;
```

```
BEGIN
```

```
    actualizar_salario(1, 5000);
```

```
END;
```

```
/
```

ACTUALIZACIÓN CON PROCEDIMIENTO

```
ORACLE Database Express Edition
Usuario: BD_20
Inicio > SQL > Comandos SQL
[ ] Confirmación Automática  Mostrar 10
DECLARE
PROCEDURE actualizar_salario(p_dorsal IN NUMBER, p_nuevo_salario IN NUMBER) IS
BEGIN
  UPDATE JUGADORES
  SET SALARIO = p_nuevo_salario
  WHERE DORSAL = p_dorsal;
  DBMS_OUTPUT.PUT_LINE('Salario actualizado.');
```

UPDATE

SALIDA

```
  COMMIT;
END actualizar_salario;
BEGIN
  actualizar_salario(1, 5000);
END;
/

Resultados  Explicar  Describir  SQL Guardado  Historial

Salario actualizado.
1 fila(s) actualizada(s).

0,00 segundos
```

Programa con:
Estructura de
datos simple
Procedimientos/Fu
nciones
Modificación
Cursores

Programa

DECLARE

PROCEDURE eliminar_jugador(p_dorsal IN NUMBER) IS

BEGIN

DELETE FROM JUGADORES WHERE DORSAL = p_dorsal;

IF SQL%ROWCOUNT > 0 THEN

DBMS_OUTPUT.PUT_LINE('Jugador eliminado.');

ELSE

DBMS_OUTPUT.PUT_LINE('Jugador no encontrado.');

END IF;

COMMIT;

END eliminar_jugador;

BEGIN

eliminar_jugador(2);

END;

/

ELIMINACION CON PROCEDIMIENTO Y CONTROL DE FLUJO

ORACLE® Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

☒ Confirmación Automática Mostrar 10

```
DECLARE
PROCEDURE eliminar_jugador(p_dorsal IN NUMBER) IS
BEGIN
    DELETE FROM JUGADORES WHERE DORSAL = p_dorsal;
    IF SQL%ROWCOUNT > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Jugador eliminado.');
```

```
    ELSE
        DBMS_OUTPUT.PUT_LINE('Jugador no encontrado.');
```

```
    END IF;
    COMMIT;
END eliminar_jugador;

BEGIN
    eliminar_jugador(17);
END;
/
```

Resultados Explicar Describir SQL Guardado Historial

Jugador eliminado.
1 fila(s) suprimida(s).

0,00 segundos

Programa con:
Estructura de
datos simple
Eliminación
Procedimientos/Fu
nciones

Programa

DECLARE

TYPE t_cursor IS REF CURSOR;

c_jugadores t_cursor;

v_dorsal JUGADORES.DORSAL%TYPE;

v_nombre JUGADORES.NOMBRE%TYPE;

v_salario JUGADORES.SALARIO%TYPE;

PROCEDURE listar_jugadores(p_min_salario IN NUMBER) IS

BEGIN

OPEN c_jugadores FOR

SELECT DORSAL, NOMBRE, SALARIO

FROM JUGADORES

WHERE SALARIO >= p_min_salario;

LOOP

FETCH c_jugadores INTO v_dorsal, v_nombre, v_salario;

EXIT WHEN c_jugadores%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_dorsal || ' | NOMBRE: ' || v_nombre || ' |
SALARIO: ' || v_salario);

END LOOP;

CLOSE c_jugadores;

END listar_jugadores;

BEGIN

listar_jugadores(3000);

END;

/

CURSOS CON PARAMETRO Y CONTROL DE FLUJO

ORACLE Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

☒ Confirmación Automática Mostrar 10

```
DECLARE
  TYPE t_cursor IS REF CURSOR;
  c_jugadores t_cursor;
  v_dorsal JUGADORES.DORSAL%TYPE;
  v_nombre JUGADORES.NOMBRE%TYPE;
  v_salario JUGADORES.SALARIO%TYPE;

  PROCEDURE listar_jugadores(p_min_salario IN NUMBER) IS
  BEGIN
    OPEN c_jugadores FOR
      SELECT DORSAL, NOMBRE, SALARIO
      FROM JUGADORES
      WHERE SALARIO >= p_min_salario;

    LOOP
      FETCH c_jugadores INTO v_dorsal, v_nombre, v_salario;
      EXIT WHEN c_jugadores%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_dorsal || ' NOMBRE: ' || v_nombre || ' SALARIO: ' || v_salario);
    END LOOP;

    CLOSE c_jugadores;
  END listar_jugadores;
```

BEGIN

Resultados Explicar Describir SQL Guardado Historial

DORSAL: 48	NOMBRE: Carlos	SALARIO: 35000
DORSAL: 12	NOMBRE: JESUS	SALARIO: 34568
DORSAL: 15	NOMBRE: PACO	SALARIO: 3458
DORSAL: 16	NOMBRE: JUAN	SALARIO: 3468
DORSAL: 19	NOMBRE: PEPE	SALARIO: 7456
DORSAL: 7	NOMBRE: AKSEL	SALARIO: 4576
DORSAL: 27	NOMBRE: ENZO	SALARIO: 3458
DORSAL: 47	NOMBRE: IKER	SALARIO: 4958
DORSAL: 11	NOMBRE: OTTI	SALARIO: 4868
DORSAL: 23	NOMBRE: XELAI	SALARIO: 4668

Sentencia procesada.

0,02 segundos

PROCEDURE

CURSOS
y BUCLE

SALIDA

SALIDA

Programa con:
Entrada salida
Estructura de
datos simple
Recursividad
Procedimientos/Fu
nciones
Selección
Cursos

Programa

DECLARE

FUNCTION verificar_jugador(p_dorsal IN NUMBER) RETURN BOOLEAN IS

v_existe NUMBER;

BEGIN

SELECT COUNT(*)

INTO v_existe

FROM JUGADORES

WHERE DORSAL = p_dorsal;

RETURN v_existe > 0;

END verificar_jugador;

BEGIN

IF NOT verificar_jugador(5) THEN

INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO,
DURACION_CONTRATO)

VALUES (5, 'Ana', 'Lopez', 3500, 3);

DBMS_OUTPUT.PUT_LINE('Jugador insertado.');

ELSE

DBMS_OUTPUT.PUT_LINE('Jugador ya existe.');

END IF;

COMMIT;

END;

/

SELECCION E INSERCIÓN CON FUNCIÓN Y CONTROL DE FLUJO

ORACLE Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

Confirmación Automática Mostrar 10 Guardar Ejecutar

```
DECLARE
FUNCTION verificar_jugador(p_dorsal IN NUMBER) RETURN BOOLEAN IS
v_existe NUMBER;
BEGIN
SELECT COUNT(*)
INTO v_existe
FROM JUGADORES
WHERE DORSAL = p_dorsal;
RETURN v_existe > 0;
END verificar_jugador;

BEGIN
IF NOT verificar_jugador(5) THEN
INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO, DURACION_CONTRATO)
VALUES (5, 'Ana', 'Lopez', 3500, 3);
ELSE
DBMS_OUTPUT.PUT_LINE('Jugador insertado.');
```

FUNCIÓN

SELECCIÓN

ITERACIÓN

SALIDA

I

Resultados Explicar Describir SQL Guardado Historial

Jugador insertado.
1 fila(s) insertada(s).
0.00 segundos

Idioma: es Application Express 2.1.0.0.39 Copyright © 1999, 2006, Oracle. Todos los Derechos Reservados.

Programa con:
Entrada salida
Estructura de
datos simple
Iteración
Procedimientos/Fu
nciones
Selección

Programa

DECLARE

TYPE t_cursor IS REF CURSOR;

c_estadisticas t_cursor;

v_n_partido ESTADISTICAS.N_PARTIDO%TYPE;

v_goles ESTADISTICAS.GOLES%TYPE;

PROCEDURE actualizar_goles(p_min_goles IN NUMBER) IS

BEGIN

OPEN c_estadisticas FOR

SELECT N_PARTIDO, GOLES

FROM ESTADISTICAS

WHERE GOLES < p_min_goles;

LOOP

FETCH c_estadisticas INTO v_n_partido, v_goles;

EXIT WHEN c_estadisticas%NOTFOUND;

UPDATE ESTADISTICAS

SET GOLES = p_min_goles

WHERE N_PARTIDO = v_n_partido;

END LOOP;

CLOSE c_estadisticas;

DBMS_OUTPUT.PUT_LINE('Goles actualizados.');

COMMIT;

END actualizar_goles;

BEGIN

actualizar_goles(10);

END;

/

PROCEDIMIENTO CON CURSOR PARA ACTUALIZACIÓN

ORACLE Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

☒ Confirmación Automática Mostrar 10

```
DECLARE
  TYPE t_cursor IS REF CURSOR;
  c_estadisticas t_cursor;
  v_n_partido ESTADISTICAS.N_PARTIDO%TYPE;
  v_goles ESTADISTICAS.GOLES%TYPE;

  PROCEDURE actualizar_goles(p_min_goles IN NUMBER) IS
  BEGIN
    OPEN c_estadisticas FOR
      SELECT N_PARTIDO, GOLES
      FROM ESTADISTICAS
      WHERE GOLES < p_min_goles;

    LOOP
      FETCH c_estadisticas INTO v_n_partido, v_goles;
      EXIT WHEN c_estadisticas%NOTFOUND;
      UPDATE ESTADISTICAS
      SET GOLES = p_min_goles
      WHERE N_PARTIDO = v_n_partido;
    END LOOP;

    CLOSE c_estadisticas;
    DBMS_OUTPUT.PUT_LINE('Goles actualizados: ');
    COMMIT;
  END actualizar_goles;

```

```
BEGIN
  actualizar_goles(10);

```

Resultados Explicar Describir SQL Guardado Historial

Goles actualizados.

1 fila(s) actualizada(s).

0.02 segundos

Programa con:
Estructura de
datos simple
Recursividad
Procedimientos/Fu
nciones
Selección y
modificación
Cursores

Programa

DECLARE

TYPE t_jugador IS RECORD (

dorsal JUGADORES.DORSAL%TYPE,

nombre JUGADORES.NOMBRE%TYPE,

apellidos JUGADORES.APELLIDOS%TYPE,

salario JUGADORES.SALARIO%TYPE,

duracion_contrato JUGADORES.DURACION_CONTRATO%TYPE

);

v_nuevo_jugador t_jugador;

PROCEDURE insertar_y_listar_jugador(p_jugador IN t_jugador) IS

v_jugador t_jugador;

BEGIN

INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO,
DURACION_CONTRATO)

VALUES (p_jugador.dorsal, p_jugador.nombre, p_jugador.apellidos,
p_jugador.salario, p_jugador.duracion_contrato);

DBMS_OUTPUT.PUT_LINE('Jugador insertado: ' || p_jugador.nombre);

SELECT DORSAL, NOMBRE, APELLIDOS, SALARIO,
DURACION_CONTRATO INTO v_jugador
FROM JUGADORES WHERE DORSAL = p_jugador.dorsal;

DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_jugador.dorsal || ' |
NOMBRE: ' || v_jugador.nombre || ' | APELLIDOS: ' || v_jugador.apellidos
|| ' | SALARIO: ' || v_jugador.salario || ' | DURACION_CONTRATO: ' ||
v_jugador.duracion_contrato);

COMMIT;

END insertar_y_listar_jugador;

BEGIN

v_nuevo_jugador.dorsal := 6;

v_nuevo_jugador.nombre := 'Luis';

v_nuevo_jugador.apellidos := 'Martinez';

v_nuevo_jugador.salario := 4000;

v_nuevo_jugador.duracion_contrato := 24; -- Asigna un valor para la
duración del contrato

insertar_y_listar_jugador(v_nuevo_jugador);

END;

/

PROCEDIMIENTO PARA INSERCIÓN Y SELECCIÓN CON ESTRUCTURAS DE DATOS

```
ORACLE Database Express Edition
User: TO_20
Menu > SQL > Comandos SQL
Confirmación Automática Mostrar 10 Guardar Ejecutar

-- Jugador - Jugador --
TYPE t_jugador IS RECORD (
  dorsal JUGADORES.DORSAL%TYPE,
  nombre JUGADORES.NOMBRE%TYPE,
  apellidos JUGADORES.APELLIDOS%TYPE,
  salario JUGADORES.SALARIO%TYPE,
  duracion_contrato JUGADORES.DURACION_CONTRATO%TYPE
);

PROCEDURE Insertar_y_listar_jugador(p_jugador IN t_jugador) IS
  v_jugador t_jugador;
BEGIN
  INSERT INTO JUGADORES (DORSAL, NOMBRE, APELLIDOS, SALARIO, DURACION_CONTRATO)
  VALUES (p_jugador.dorsal, p_jugador.nombre, p_jugador.apellidos, p_jugador.salario, p_jugador.duracion_contrato);
  DBMS_OUTPUT.PUT_LINE('Jugador insertado: ' || p_jugador.nombre);
  SELECT DORSAL, NOMBRE, APELLIDOS, SALARIO, DURACION_CONTRATO INTO v_jugador
  FROM JUGADORES WHERE DORSAL = p_jugador.dorsal;
  DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_jugador.dorsal || ' | NOMBRE: ' || v_jugador.nombre || ' | APELLIDOS: ' || v_jugador.apellidos || ' | SALARIO: ' || v_jugador.salario || ' | DURACION_CONTRATO: ' || v_jugador.duracion_contrato);
END Insertar_y_listar_jugador;

BEGIN
  v_nuevo_jugador.dorsal := 6;
  Insertar_y_listar_jugador(v_nuevo_jugador);
END;
```

Diagram annotations:

- PROCEDIMIENTO** points to the entire procedure block.
- INSERTION** points to the `INSERT INTO` statement.
- SELECCION** points to the `SELECT INTO` statement.
- SALIDA** points to the `DBMS_OUTPUT.PUT_LINE` statements.

Results:

```
Results Explicit Describe SQL Quando Historial
Jugador insertado: Luis
DORSAL: 6 | NOMBRE: Luis | APELLIDOS: Martinez | SALARIO: 4000 | DURACION_CONTRATO: 24
1 fila(s) insertada(s).
0.01 segundos
```

Programa con:
Entrada salida
Estructura de
datos compuesta
Procedimientos/Fu
nciones
Selección y
Inserción

Programa

DECLARE

FUNCTION verificar_eliminar_entrenador(p_id IN NUMBER) RETURN BOOLEAN IS

 v_existe NUMBER := 0;

BEGIN

 SELECT COUNT(*)

 INTO v_existe

 FROM ENTRENADOR

 WHERE N_IDENTIFICADOR = p_id;

 IF v_existe > 0 THEN

 DELETE FROM ENTRENADOR WHERE N_IDENTIFICADOR = p_id;

 DBMS_OUTPUT.PUT_LINE('Entrenador eliminado.');

 RETURN TRUE;

 ELSE

 DBMS_OUTPUT.PUT_LINE('Entrenador no encontrado.');

 RETURN FALSE;

 END IF;

 END verificar_eliminar_entrenador;

BEGIN

 IF verificar_eliminar_entrenador(3) THEN

 COMMIT;

 ELSE

 ROLLBACK;

 END IF;

END;

/

ELIMINAR CON FUNCIÓN Y CONDICIONES

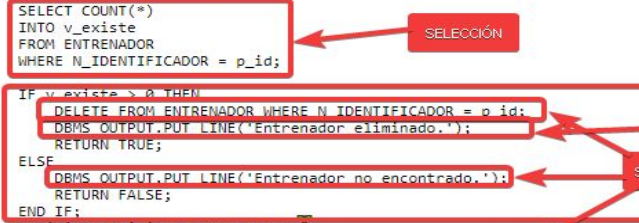
ORACLE® Database Express Edition

Usuario: BD_20

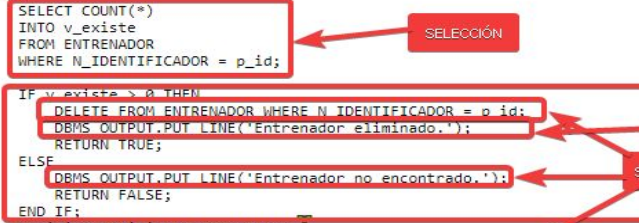
Inicio > SQL > Comandos SQL

☒ Confirmación Automática Mostrar: 10

```
DECLARE
FUNCTION verificar_eliminar_entrenador(p_id IN NUMBER) RETURN BOOLEAN IS
  v_existe NUMBER := 0;
BEGIN
  SELECT COUNT(*)
  INTO v_existe
  FROM ENTRENADOR
  WHERE N_IDENTIFICADOR = p_id;
  IF v_existe > 0 THEN
    DELETE FROM ENTRENADOR WHERE N_IDENTIFICADOR = p_id;
    DBMS_OUTPUT.PUT_LINE('Entrenador eliminado.');
```



```
    RETURN TRUE;
  ELSE
    DBMS_OUTPUT.PUT_LINE('Entrenador no encontrado.');
```



```
    RETURN FALSE;
  END IF;
END verificar_eliminar_entrenador;
BEGIN
  IF verificar_eliminar_entrenador(3) THEN
    COMMIT;
  ELSE
    ROLLBACK;
  END IF;
```

Resultados Explicar Describir SQL Guardado Historial

Entrenador no encontrado.

1 fila(s) suprimida(s).

0,00 segundos

Programa con:
Entrada salida
Estructura de
datos simple
Iteración
Procedimientos/Fu
nciones
Selección y
Eliminación

Programa

DECLARE

TYPE t_cursor IS REF CURSOR;

c_jugadores t_cursor;

v_dorsal JUGADORES.DORSAL%TYPE;

v_nombre JUGADORES.NOMBRE%TYPE;

v_apellidos JUGADORES.APELLIDOS%TYPE;

FUNCTION obtener_jugadores_por_salario(p_salario_min IN NUMBER)
RETURN t_cursor IS

v_cursor t_cursor;

BEGIN

OPEN v_cursor FOR

SELECT DORSAL, NOMBRE, APELLIDOS

FROM JUGADORES

WHERE SALARIO >= p_salario_min;

RETURN v_cursor;

END obtener_jugadores_por_salario;

BEGIN

c_jugadores := obtener_jugadores_por_salario(4000);

LOOP

FETCH c_jugadores INTO v_dorsal, v_nombre, v_apellidos;

EXIT WHEN c_jugadores%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_dorsal || ' | NOMBRE: ' ||
v_nombre || ' | APELLIDOS: ' || v_apellidos);

END LOOP;

CLOSE c_jugadores;

END;

/

PROCEDIMIENTO CON CURSOR Y FUNCION PARA CONSULTAS

ORACLE Database Express Edition

Usuario: BD_20

Inicio > SQL > Comandos SQL

Confirmación Automática Mostrar: 10

```
DECLARE
  TYPE t_cursor IS REF CURSOR;
  c_jugadores t_cursor;
  v_dorsal JUGADORES.DORSAL%TYPE;
  v_nombre JUGADORES.NOMBRE%TYPE;
  v_apellidos JUGADORES.APELLIDOS%TYPE;

  FUNCTION obtener_jugadores_por_salario(p_salario_min IN NUMBER) RETURN t_cursor IS
    v_cursor t_cursor;
  BEGIN
    OPEN v_cursor FOR
      SELECT DORSAL, NOMBRE, APELLIDOS
      FROM JUGADORES
      WHERE SALARIO >= p_salario_min;
    RETURN v_cursor;
  END obtener_jugadores_por_salario;

BEGIN
  c_jugadores := obtener_jugadores_por_salario(4000);

  LOOP
    FETCH c_jugadores INTO v_dorsal, v_nombre, v_apellidos;
    EXIT WHEN c_jugadores%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('DORSAL: ' || v_dorsal || ' | NOMBRE: ' || v_nombre || ' | APELLIDOS: ' || v_apellidos);
  END LOOP;
```

SELECCIÓN

FUNCION

SALIDA

CURSOR

Resultados Explicar Describir SQL Guardado Historial

DORSAL: 48	NOMBRE: Carlos	APELLIDOS: López
DORSAL: 6	NOMBRE: Luis	APELLIDOS: Martínez
DORSAL: 12	NOMBRE: JESUS	APELLIDOS: TERINO
DORSAL: 19	NOMBRE: PEPE	APELLIDOS: GARCIA
DORSAL: 7	NOMBRE: AKSEL	APELLIDOS: ZUZUNAGA
DORSAL: 47	NOMBRE: IKER	APELLIDOS: GARZA
DORSAL: 11	NOMBRE: OTTI	APELLIDOS: SANDEMETRIO
DORSAL: 23	NOMBRE: XELAI	APELLIDOS: URRIAGA

Sentencia procesada.

0.00 segundos

Programa con:
Entrada salida
Estructura de
datos simple
Procedimientos/Fu
nciones
Selección
Cursores