

SET SERVEROUTPUT ON FORMAT;

CREATE TABLE

Clave alternativa

```
Constraint nombreClave unique (campo_1, campo_n);
```

Referencia a otra tabla

Desde la clave primaria

```
Constraint pk_nombreclave primary key(campos),
```

```
Constraint fk_nombreclave reference otraTabla;
```

Como clave ajena

```
Campo tipoCampo fk_nombredeclave reference otraTabla;
```

Check

Para un campo concreto

```
Campo tipoCampo ch_nombreClave check (condicion)
```

p.Ej

```
Campo tipoCampo ch_nombreClave check (Campo > 0)
```

```
Campo tipoCampo ch_nombreClave check (Campo IN('hola','adios'))
```

ALTER TABLE

ALTER TABLE tabla

```
ADD campo TipoCampo constraint NombreConst condicion,
```

```
MODIFY Campo Constraint NombreConst condición,
```

```
MODIFY Campo NOT NULL,
```

```
DROP Campo;
```

FUNCTION

```
Create function nombre (I_var1 campo%TYPE, I_var2 campo%TYPE) return campo%TYPE
```

Is

```
--Declaracion de variables
```

```
Var_temp1 campo%TYPE;
```

```
Var_temp2 campo%TYPE;
```

Begin

```
--Solucion lógica
```

```
Select c1, c2 into Var_temp1, Var_temp2
```

```
From tabla
```

```
Where
```

```
Campo1 = I_var1 and Campo2 = I_var2;
```

```
--Las sumo por ejemplo
```

```
Return (Var_temp1 + Var_temp2);
```

EXCEPTION

```
WHEN NO_DATA_FOUND THEN...
```

```
WHEN ZERO_DIVIDE THEN
```

```
WHEN OTHERS THEN
```

End;

PROCEDURE

```
Create procedure nombreDelProcedimiento Is
    --declaracion de variables
    Var1 tipo;
    Var2 Campo%TYPE;
    --Declaro también algún cursor
    Cursor nombreCursor IS
    Select campos from tabla quere condición;
Begin

    --recorrer el cursor
    For registro in nombreCursor
    Loop
        --Hago cosas con los registros del cursor
        -- como guardalo en otra tabla
        Insert into tabla registro;

        --o sacar por pantalla alguno de sus campos
        Dbms_output.put_line(registro.campo || ' <- algo de texto');

    End loop;
EXCEPTION
    WHEN NO_DATA_FOUND      THEN...
    WHEN ZERO_DIVIDE        THEN
    WHEN OTHERS              THEN

End;
```

TRIGGER

```
CREATE or REPLACE Trigger NombreDelTrigger (aft/bef)(Ins/upd/del) OF campo ON Tabla

FOR EACH ROW

    DECLARE      variable      TipoVariable;

BEGIN
    IF (:NEW.campo = :OLD.campo) THEN

        --hacer cosas

    END IF;
END;
```

OTROS

```
RAISE_APPLICATION_ERROR(-20100,MENSAJE DE ERROR);

EXCEPTION
WHEN NO_DATA_FOUND THEN HacerCosas;

Exec Procedimiento();

Select Funcion(p) from dual;
```