

# Variables

```
DECLARE @variable varchar(30);
```

```
SET @find = 'Man%';
```

Declarar dando  
nombre y tipo,  
siempre comienzan  
con @

Asignación de valor con SET

```
DECLARE @variable varchar(30) = 'Man%';
```

También inicializando su valor

**Se pueden usar en las cláusulas SELECT**

```
SELECT @variable  
PRINT @variable
```

Para visualizarlo

# IF

```
--procedimiento almacenado con IF
if object_id('ver_texto') is not null
    drop procedure ver_texto;
go
create procedure ver_texto
    @texto varchar(20)
as
IF @texto like 'A%'
    BEGIN
        PRINT @texto + ' comienza con A'
    END
ELSE
    BEGIN
        PRINT @texto + ' no comienza con A'
    END
GO
exec ver_texto 'hola'
exec ver_texto 'antes'
```

```
--procedimiento detectar pares
if object_id('detecta_pares') is not null
    drop procedure detecta_pares;
go
create procedure detecta_pares
    @numero int
as
    IF  (@numero % 2)=0
        BEGIN
            PRINT cast(@numero AS varchar) + ' es par'
        END
    ELSE
        BEGIN
            PRINT cast(@numero AS varchar) + ' es impar'
        END
GO
exec detecta_pares 44
exec detecta_pares 33
exec detecta_pares -1
```

```
-- procedimiento almacenado IF con subconsulta
if object_id('libros') is not null
    drop table libros;
go
create table libros(
    titulo varchar(40),
    autor varchar(30),
    editorial varchar(15),
    precio float,
    cantidad integer
);
go
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('El aleph','Borges','Emece',25.50,100);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Alicia en el pais de las maravillas','Lewis
Carroll','Atlantida',10,200);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Matematica estas ahi','Paenza','Siglo XXI',18.8,200);
go
```

```
if object_id('actualizar_autor') is not null
    drop procedure actualizar_autor;
go
create procedure actualizar_autor
    @autoranterior varchar(30), @autornuevo varchar(30)
as
IF
    EXISTS(SELECT titulo, autor, editorial, precio, cantidad FROM libros
           WHERE autor = @autoranterior)
    BEGIN
        update libros
        set autor = @autornuevo
        where autor=@autoranterior
    END
else
    print 'El autor indicado no está'
go
SELECT titulo, autor, editorial, precio, cantidad FROM libros
exec actualizar_autor 'Borges', 'Jorge Luis Borges'
SELECT titulo, autor, editorial, precio, cantidad FROM libros
exec actualizar_autor 'Cervantes', 'Miguel de Cervantes'
```

Aquí lo  
ejecutamos

# While

```
--ejemplo de while
if object_id('proc_contador') is not null
    drop procedure proc_contador;
go

create procedure proc_contador
as
    DECLARE @contador int
    SET @contador = 1
    WHILE (@contador <= 20)
    BEGIN
        PRINT 'Iteracion del bucle ' + cast(@contador AS varchar)
        SET @contador = @contador + 1
    END
    print 'Final'
go
exec proc_contador
```

# set nocount y @@rowcount

```
-- uso de set nocount y @@rowcount
if object_id('libros') is not null
    drop table libros;
go
create table libros(
    titulo varchar(40),
    autor varchar(30),
    editorial varchar(15),
    precio float,
    cantidad integer
);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('El aleph','Borges','Emece',25.50,100);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Alicia en el pais de las maravillas','Lewis
Carroll','Atlantida',10,200);
insert into libros (titulo,autor,editorial,precio,cantidad)
values ('Matematica estas ahi','Paenza','Siglo XXI',18.8,200);
go
```

Set nocount on no muestra el nº de filas afectadas, al contrario que colocando off.

@@rowcount contiene el nº de registros afectados por la sentencia anterior.

```
if object_id('ver_autores') is not null
    drop procedure ver_autores;
go
create procedure ver_autores
as
    set nocount on
    select distinct autor from libros
go
```



```
exec ver_autores
```

```
if object_id('ver_libros') is not null
    drop procedure ver_libros;
go
create procedure ver_libros
    @autor varchar(30)
as
    set nocount on --off
    select titulo from libros
    where autor like @autor
    print cast(@@rowcount as varchar) + ' registros'
go
```



```
exec ver_libros 'Cervantes'
exec ver_libros 'Borges%'
```



# Return

La instrucción "return" sale de una consulta o procedimiento y todas las instrucciones posteriores no son ejecutadas.

```
create procedure pa_libros_autor
    @autor varchar(30)=null
as
if @autor is null
begin
    select 'Debe indicar un autor'
    return
end;
select titulo from libros where autor = @autor;
go
```

# Case

La función "case" compara 2 o más valores y devuelve un resultado.

La sintaxis es la siguiente:

```
case VALORACOMPARAR
when VALOR1 then RESULTADO1
when VALOR2 then RESULTADO2
...
else RESULTADO3
end
```

```
select nombre,nota, resultado=
case nota
when 0 then 'libre'
when 1 then 'libre'
when 2 then 'libre'
when 3 then 'libre'
when 4 then 'regular'
when 5 then 'regular'
when 6 then 'regular'
when 7 then 'promocionado'
when 8 then 'promocionado'
when 9 then 'promocionado'
when 10 then 'promocionado'
end
from alumnos;
```

```
select nombre, nota, condicion=
case
when nota<4 then 'libre'
when nota >=4 and nota<7 then 'regular'
when nota>=7 then 'promocionado'
else 'sin nota'
end
from alumnos;
```

```
use empresasinformaticas  
go
```

```
select clave,descripcion,precio  
from componente;
```

```
select clave,descripcion,precio,  
case  
    when precio<10 then 'Barato'  
    when precio between 10 and 50 then 'Medio'  
    else 'Caro'  
end as nivelprecio  
from componente;
```

```
select count(*),  
case  
    when precio<10 then 'Barato'  
    when precio between 10 and 50 then 'Medio'  
    else 'Caro'  
end as nivelprecio  
from componente  
group by  
case  
    when precio<10 then 'Barato'  
    when precio between 10 and 50 then 'Medio'  
    else 'Caro'  
end  
;
```

## Concatenar cadenas de caracteres

Con las tablas de ejemplos anteriores...

```
if object_id('concatenar_autores') is not null
    drop procedure concatenar_autores;
go
create procedure concatenar_autores
as
    declare @delimitador varchar, @autores varchar(max)
    set @delimitador = ','
    set @autores = ''
    select @autores=@autores +@delimitador+autor
    from libros
    select @autores as autores
go

exec concatenar_autores
select autor from libros
```

## Cargar datos en Tablas temporales

Con el procedimiento anterior

```
-- ver resultado de procedimiento en tabla temporal
create table #t1
(
    autores varchar(100)
)
insert #t1 exec concatenar_autores
select autores from #t1
```

# Control de errores

```
--try catch
if object_id('libros') is not null
    drop table libros;
go
create table libros(
    idlibro int primary key,    titulo varchar(40),    autor varchar(30),
    editorial varchar(15),    precio float,    cantidad integer
);
go
if object_id('insertar_libros') is not null
    drop procedure insertar_libros;
go
create procedure insertar_libros
    @idlibro integer,    @idtitulo varchar(40),    @autor varchar(30),
    @editorial varchar(15),    @precio float,    @cantidad integer
as
begin try
set nocount on
insert into libros (idlibro,titulo,autor,editorial,precio,cantidad)
    values (    @idlibro,    @idtitulo,    @autor,
        @editorial,    @precio,    @cantidad)
select 'Registro insertado correctamente'
end try
begin catch
    SELECT ERROR_NUMBER() AS 'Nº de error',ERROR_MESSAGE() as 'Mensaje de error'
end catch;
go
exec insertar_libros 2,'Matematica estas aqui','Paenza','Siglo XXI',18.8,200
exec insertar_libros 1,'Alicia en el pais de las maravillas','Lewis Carroll','Atlantida',10,200
exec insertar_libros 2,'El aleph','Borges','Emece',25.50,100
```

## Sp\_executesql

Procedimiento para  
ejecutar una  
cadena de  
caracteres que  
contiene una  
sentencia SQL

```
-- hacer select * de tabla especificada
if object_id('proc_vertabla') is not null
    drop procedure proc_vertabla;
go

create procedure proc_vertabla
    @tabla nvarchar(100)
as
    set nocount on
    if object_id(@tabla) is null
        begin
            print N'La tabla ' + @tabla + N' no existe'
        end
    else
        begin
            declare @sentencia nvarchar(1000)
            set @sentencia=N'select * from ' + @tabla
            exec sp_executesql @sentencia
        end
go

exec proc_vertabla 'pruebalibro1'
go
exec proc_vertabla 'pruebalibro2'
go
exec proc_vertabla 'pruebalibro3'
go
```



Igual sin nvarchar  
mediante  
exec(@sentencia)

```
if object_id('proc_vertabla2') is not null
    drop procedure proc_vertabla2;
go

create procedure proc_vertabla2
    @tabla varchar(100)
as
    set nocount on
    if object_id(@tabla) is null
        begin
            print 'La tabla ' + @tabla + ' no existe'
        end
    else
        begin
            declare @sentencia varchar(1000)
            set @sentencia='select * from ' + @tabla
            exec (@sentencia)
        end
    go

exec proc_vertabla2 'pruebalibro1'
go
exec proc_vertabla2 'pruebalibro2'
go
exec proc_vertabla2 'pruebalibro3'
go
```