



UNIVERSIDAD TÉCNOLOGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

Base de Datos II

Teoría SQL – Parte 4

ING. ARIEL HERRERA



Contenido

Control de flujo.....	2
-----------------------	---



Control de flujo

Existen palabras especiales que pertenecen al lenguaje de control de flujo que controlan la ejecución de las sentencias, los bloques de sentencias y procedimientos almacenados.

Nosotros veremos Begin... end, If... Else y Case

"begin... end" encierran un bloque de sentencias para que sean tratados como unidad.

"if... else": testean una condición; se emplean cuando un bloque de sentencias debe ser ejecutado si una condición se cumple y si no se cumple, se debe ejecutar otro bloque de sentencias diferente.

```
DECLARE @Number int
```

```
SET @Number = 150
```

```
IF @Number > 100
```

```
    PRINT 'El número es grande.'
```

```
ELSE
```

```
    BEGIN
```

```
        IF @Number < 10
```

```
            PRINT 'El número es pequeño.'
```

```
        ELSE
```

```
            PRINT 'El número es mediano.'
```

```
    END
```

```
GO
```



También se puede ejecutar dentro de un procedimiento almacenado...

```
create procedure XXX
```

```
@Number int
```

```
As
```

```
IF @Number > 100
```

```
    PRINT 'El número es grande.'
```

```
ELSE
```

```
    IF @Number < 10
```

```
        PRINT 'El número es pequeño.'
```

```
    ELSE
```

```
        PRINT 'El número es mediano.'
```

```
GO
```

Veamos otro ejemplo. Tenemos nuestra tabla "Articulos"; queremos mostrar todos los artículos que no estén disponibles (cantidad=0), y si de todos hubiera unidades, mostrar un mensaje indicando tal situación:

```
if exists (select * from Articulos where Cantidad=0)
```

```
    (select Articulo from Articulos where cantidad=0)
```

```
else
```

```
    Select 'Hay stock de todos los artículos' as Mensaje
```

```
-- o: PRINT 'Hay stock de todos los artículos'
```

Podemos consultar también por la existencia (o no) de una tabla en nuestra base de datos. Este procedimiento suele utilizarse para la creación y borrado de tablas temporales.



```
IF not exists (select * from INFORMATION_SCHEMA.TABLES where  
TABLE_NAME='ssx')
```

```
begin
```

```
create table ssx  
(  
    ssx_id      int identity(1,1) primary key,  
    ssx_desc varchar(40) not null  
)
```

```
End
```

O por la inversa

```
IF exists (select * from INFORMATION_SCHEMA.TABLES where  
TABLE_NAME='ssx')
```

```
begin
```

```
    drop table ssx  
    create table  
    ssx  
    (  
        ssx_id int identity(1,1) primary key,  
        ssx_desc varchar(40) not null  
    )
```

```
end
```

La expresión CASE evalúa una lista de condiciones y devuelve una de las varias expresiones de resultado posibles.

La expresión CASE compara una expresión con un conjunto de expresiones sencillas para determinar el resultado, con argumentos WHEN... THEN, admitiendo un argumento ELSE opcional.

CASE se puede utilizar en cualquier instrucción o cláusula que permite una expresión válida. Por ejemplo, puede utilizar CASE en instrucciones como SELECT, UPDATE, DELETE y SET, y en cláusulas como WHERE, ORDER BY y



HAVING.

USE Clase09

GO

SELECT CodArticulo, Articulo, 'Rango de Precios' =

CASE

WHEN PrecioUnitario < 50 THEN 'Menos de \$50'

WHEN PrecioUnitario >= 50 and PrecioUnitario < 250 THEN 'Menos de \$250'

WHEN PrecioUnitario >= 250 and PrecioUnitario < 1000 THEN 'Menos de
\$1000'

ELSE 'Más de \$1000'

END

FROM Articulos

ORDER BY CodArticulo

GO