



UTN.BA FACULTAD
REGIONAL
BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de
e-Learning**

Diplomatura en Bases de Datos

Módulo 1 - Unidad 3



www.sceu.frba.utn.edu.ar/e-learning

Agenda: Objetos de las Bases de Datos

- *Tablas*
- Índices
- Vistas
- Procedimientos almacenados.
- Funciones.
- Triggers
- Estructuras de control
- Cursores

Tablas

- *Sirven para guardar los datos*
- CREATE
- DROP
- ALTER
- TRUNCATE

Tablas: CREATE (sintaxis)

```
CREATE TABLE [dueño de la tabla].[nombre de la tabla]
(
[nombre del campo] [tipo del campo](longitud) NULL,
...
)
```

Tablas: CREATE (ejemplo)

```
CREATE TABLE dbo.clientes  
(  
  IdCliente INT NOT NULL,  
  FechaAlta DATE NOT NULL,  
  RazonSocial VARCHAR(50) NOT NULL,  
  Vendedor VARCHAR(50) NULL  
)
```

Tablas: ALTER (sintaxis) = CREATE

Tablas: DROP (sintaxis)

DROP [Nombre de la tabla]

Índices

- *Sirven para acelerar las búsquedas*
- CREATE
- DROP
- ALTER

Índices: CREATE (sintaxis)

CREATE CLUSTERED INDEX [nombre del índice] ON [dueño de la
tabla].[nombre de la tabla]

(
[lista de campos] [orden]
)

CREATE NONCLUSTERED INDEX [nombre del índice] ON [dueño de
la tabla].[nombre de la tabla]

(
[lista de campos] [orden]
)

Índices: CREATE (ejemplo)

```
CREATE CLUSTERED INDEX MiIndice ON dbo.cursos  
(  
    IdCurso  
)
```

```
CREATE NONCLUSTERED INDEX MiIndice2 ON  
    dbo.cursos  
(  
    Descripcion  
)
```

Índices: ALTER (sintaxis) = CREATE

Índices: DROP (sintaxis)

DROP [Nombre del índice]

Vistas

- *Sirven para mostrar los datos con una organización distinta a la de las tablas sin persistencia*
- CREATE
- DROP
- ALTER

Vistas: CREATE (sintaxis)

***CREATE VIEW [nombre de la vista]
AS [sentencia de selección]***

Vistas: CREATE (ejemplo)

CREATE VIEW Clientes2018

AS

SELECT * FROM dbo.clientes WHERE
YEAR(FechaAlta) = 2018

Vistas: ALTER (sintaxis) = CREATE

Vistas: DROP (sintaxis)

DROP [Nombre de la vista]

Procedimientos Almacenados

- *Sirven para guardar código que queramos ejecutar en bloque.*

- CREATE
- DROP
- ALTER

Procedimientos Almacenados: CREATE (sintaxis)

CREATE PROCEDURE [dueño].[nombre del
procedimiento]

[parametro 1 tipo de dato del parámetro 1],

...

[parametro n tipo de datos del parámetro n]

AS

BEGIN

[paquete de sentencias que queremos que se ejecuten]

END

Procedimientos almacenados: CREATE (ejemplo)

```
CREATE PROCEDURE [dbo].[ClientesPepe]  
AS  
BEGIN  
select * from clientes where vendedor = 'PEPE'  
END
```

***Procedimientos Almacenados: ALTER
(sintaxis) = CREATE***

***Procedimientos Almacenados: DROP
(sintaxis)***

DROP [Nombre del procedimiento]

Funciones

- *Sirven para guardar código que querramos ejecutar en bloque devolviendo un resultado*
- CREATE
- DROP
- ALTER

Funciones: CREATE (sintaxis)

```
CREATE FUNCTION [dueño de la función].[nombre de la función]
(
  [@nombre del parámetro 1 tipo de dato del parámetro 1],
  ...
  [@nombre del parámetro n tipo de dato del parámetro n]
)
RETURNS [tipo de dato que la función devuelve]
AS
BEGIN
  [código de la función]
RETURN [@valor que devuelve la función]
END
```

Funciones: CREATE (ejemplo)

```
CREATE FUNCTION dbo.[sumame]
(
    @a int,
    @b int
)
RETURNS int
AS
BEGIN
    declare @c int
    set @c = @a+@b
    RETURN @c
END
```

Funciones: ALTER (sintaxis) = CREATE

Funciones: DROP (sintaxis)

DROP [Nombre de la función]

Triggers

- *Sirven para disparar acciones cuando se insertan, borran o modifican registros*
- CREATE
- DROP
- ALTER

Triggers: CREATE (sintaxis)

```
CREATE TRIGGER [nombre del trigger]
ON [nombre de la tabla]
AFTER [insert, delete o update]
AS
BEGIN
[sentencias que se ejecutarán]
END
```


Triggers: CREATE (ejemplo)

CREATE TRIGGER BorraCurso

ON Cursos

AFTER delete

AS

BEGIN

delete from cursadas where idCurso in (select
IdCurso from Deleted)

END

Triggers: ALTER (sintaxis) = CREATE

Triggers: DROP (sintaxis)

DROP [Nombre del trigger]

Estructuras de control en SQL

- *Sirven para controlar el curso de la ejecución de las sentencias SQL*
- IF
ELSE
- WHILE

Ejecución condicional (sintaxis)

IF [condición]

BEGIN

[código SQL a ejecutar si la condición es verdadera]

END

ELSE

BEGIN

[código SQL a ejecutar si la condición es falsa]

END

Ejecución condicional (ejemplo)

IF @k<5

BEGIN

select @k,'Menor que 5'

END

ELSE

BEGIN

select @k, 'Mayor o igual que 5'

END

Ciclos (sintaxis)

WHILE [condición]

BEGIN

[conjunto de sentencias que se repiten]

END

Ciclos (ejemplo)

declare @s int

declare @k int

declare @l int

set @s=0

set @k = 1

set @l = 100

WHILE @k<@l

BEGIN

 set @s = @s+@k

 set @k = @k+1

END

select @s

Cursores

- *Sirven para recorrer en forma secuencial un conjunto de datos*
- DECLARE
- OPEN
- FETCH
- CLOSE
- DEALLOCATE



UTN.BA FACULTAD
REGIONAL
BUENOS AIRES
SECRETARÍA DE EXTENSIÓN UNIVERSITARIA FRBA UTN

**Centro de
e-Learning**

¿Alguna pregunta en el tintero?



www.sceu.frba.utn.edu.ar/e-learning



www.sceu.frba.utn.edu.ar/e-learning