



# INTELIGENCIA DE NEGOCIOS

## TABLAS

# TABLAS

Cuando se crea una tabla debe especificarse

- Nombre de la tabla,
- Nombres de columnas
- Tipos de datos de las columnas.

Los nombres de las columnas deben ser:

- Únicas para la misma tabla,
- Se puede utilizar mismo nombre de columnas en diferentes tablas,
- Se debe especificar un tipo de dato por cada columna.



# TABLAS

## Consideraciones para crear tablas.

### 1. Collation:

Usados con tipos de datos carácter para dictar el tipo de código de página y sus correspondientes caracteres que pueden ser representados y el orden en que se presentan.

### 2. Nulabilidad:

Se puede especificar que una columna de una tabla acepte valores nulos (no se capture un valor en ellos)



# TABLAS

## 3. Tipos especiales de columnas:

- **Columnas Calculadas** (Computed columns): Columna virtual que no esta físicamente almacenada en la tabla. (Formula)
- **Columnas Identity:** Columnas que contienen un valor secuencial generado de forma automático por SQL Server.
- **Columnas Timestamp:** Contienen un valor que indica cuando fue modificado los registros por ultima vez.
- **Columnas Dispersas:** son columnas normales que disponen de un almacenamiento optimizado para los valores NULL. Reducen los requisitos de espacio de los valores NULL a costa de una mayor sobrecarga a la hora de recuperar valores no NULL.



# TABLAS

## Ejemplo de Creación de Tablas.

```
CREATE TABLE Sales.CustomerOrders  
(OrderID int identity NOT NULL,  
OrderDate datetime NOT NULL,  
CustomerID int NOT NULL,  
Notes nVarchar(200) NULL)
```

**Nota: Crear Antes el schema SALES**



# TABLAS

Ejemplo de modificación de Tablas.

```
ALTER TABLE Sales.CustomerOrders  
ADD SalesPersonID int NOT NULL  
GO
```

```
ALTER TABLE Sales.CustomerOrders  
ALTER COLUMN Notes nVarchar(200) Not Null  
GO
```



# TABLAS

Ejemplo de Eliminación de Tablas.

```
DROP TABLE Sales.CustomerOrders
```



# ALIAS DE TIPOS DE DATOS

Se debe crear un alias de tipo de datos cuando se requiere definir un elemento que se usa comúnmente con un específico formato.

## Ejemplo.

Tipo de dato: sexo char(1)

```
CREATE TYPE dbo.sexo FROM  
Char(1) NULL
```



# REGLA

- Sirve para definir valores específicos que debe tomar un tipo de dato.

## Ejemplo:

REGLA. Tipo de datos SEXO debe aceptar solo los valores **F** (Femenino) y **M** (Masculino).

```
CREATE RULE sexo as  
@sexo in ('M','F')
```

# LLAVE FORÁNEA

- Una llave foránea es una columna o combinación que es usada para establecer y forzar una liga entre los datos de dos tablas.
- La llave foránea genera integridad referencial.
- La llave foránea define una referencia a una columna con una llave primaria o constraint UNIQUE en la misma u otra tabla



# CUANDO UTILIZAR UNA LLAVE FORÁNEA

- Los datos de una o mas columnas pueden contener solo valores que se encuentran en ciertas columnas en la misma o en otra tabla.
- Los registros en la tabla no deben ser eliminadas mientras los registros en alguna otra tabla dependen de ellos.



# LLAVE FORÁNEA

Mtro_Ventas			
folio_vta	Cliente	pares	Importe
00001	90876756	4	500
00003	90876766	2	240

Det_Ventas			
folio_vta	clave_producto	pares	Precio
00001	000234	1	100
00001	000456	2	150
00001	000222	1	100
00003	003467	2	120

Cat_productos			
clave_producto	precio_vta	precio_costo	proveedor
000234	100	80	9898767909
000456	150	120	9898763423
000222	100	78	9898767909
003467	120	100	9546767909

## Ejercicio\_Fkey.SQL



# LLAVE FORÁNEA

Ejemplo:

```
ALTER TABLE dbo.det_ventas ADD CONSTRAINT  
    FK_det_ventas_cat_productos FOREIGN KEY  
    (  
        clave_producto  
    ) REFERENCES dbo.cat_productos  
    (  
        clave_producto  
    ) ON UPDATE NO ACTION  
    ON DELETE NO ACTION
```

GO

\* Eliminar primero el detalle y luego el maestro.



# PRACTICA

## CREACION DE TABLAS

