

SQL INTERMEDIO



INDICES




INDICES: DEFINICIÓN

- **LISTA O RELACIÓN ORDENADA CON ALGÚN CRITERIO.**
- **LOS ÍNDICES SE UTILIZAN PARA RECUPERAR DATOS DE LA BASE DE DATOS MUY RÁPIDO.**
- **LOS USUARIOS NO PUEDEN VER LOS ÍNDICES, SOLO SE UTILIZAN PARA ACELERAR LAS CONSULTAS.**
- **SE DENOMINA TAMBIÉN ÍNDICE DE ALMACÉN DE FILAS PORQUE ES UN ÍNDICE DE ÁRBOL B (AGRUPADO O NO AGRUPADO)**

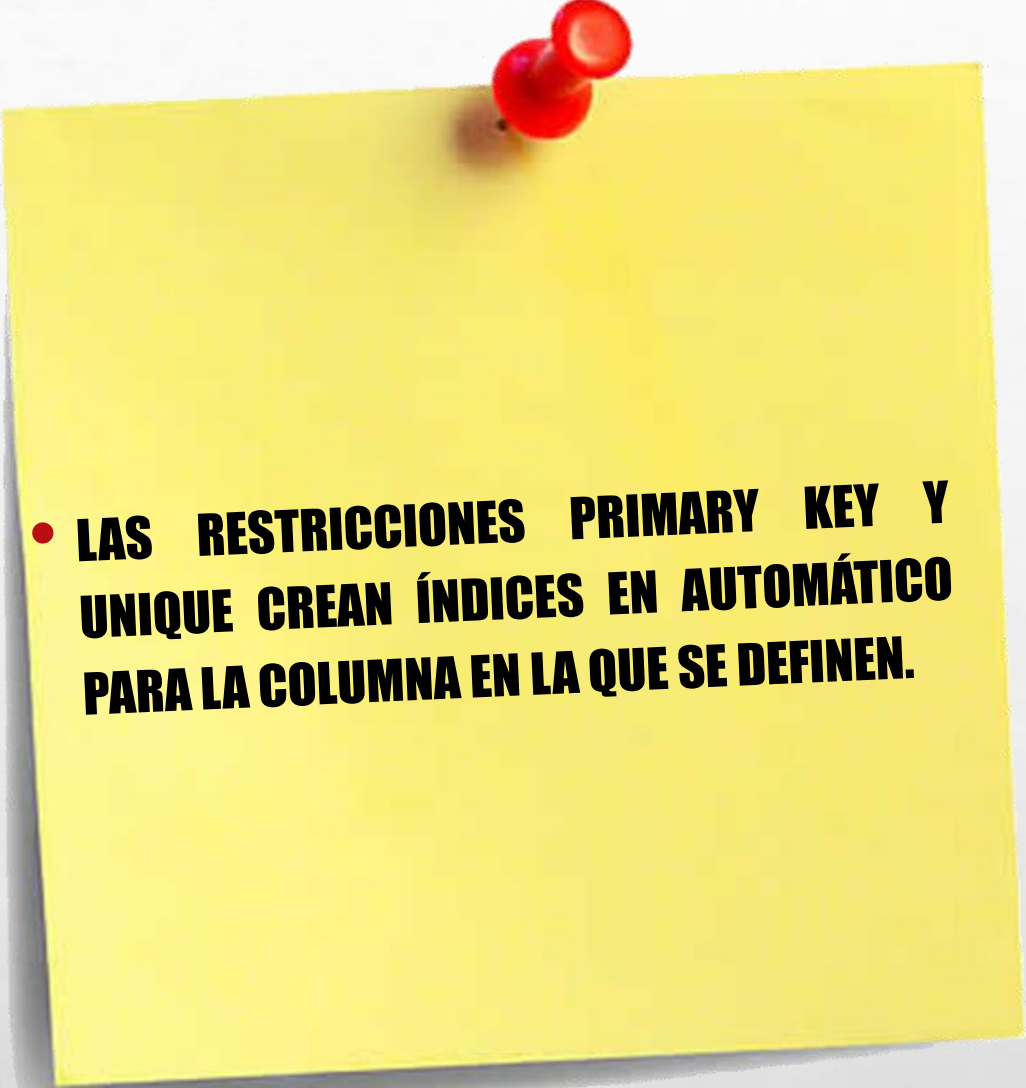
INDICES: TIPOS

- **CLÚSTER: ORDENA Y ALMACENA FILAS DE DATOS DE LA TABLA O VISTA DE ACUERDO CON LOS VALORES DE LA CLAVE DEL ÍNDICE. CUANDO LA TABLA LO CONTIENE SE DENOMINA TABLA AGRUPADA**
- **NO CLÚSTER (NO AGRUPADO): TIENEN UNA ESTRUCTURA SEPARADA DE LAS FILAS DE DATOS, CONTIENE PUNTERO A LA FILA DE DATOS QUE CONTIENE EL VALOR CLAVE Y SE LLAMA LOCALIZADOR DE FILA.**
- **EL VALOR DE LA COLUMNA ÍNDICE POR CADA FILA DEBE SER ÚNICO**

- 
- **ACTUALIZAR UNA TABLA CON ÍNDICES LLEVA MÁS TIEMPO QUE ACTUALIZAR UNA TABLA SIN ELLOS (PORQUE LOS ÍNDICES TAMBIÉN NECESITAN UNA ACTUALIZACIÓN). POR LO TANTO, SOLO CREE ÍNDICES EN COLUMNAS QUE SE BUSCARÁN CON FRECUENCIA**

INDICES: CREACIÓN

- **CREATE CLUSTERED INDEX IX_INDICE_PER ON DBO.PERSONA (CURP);**
- **CREATE NONCLUSTERED INDEX IX_INDICE_PER ON DBO.PERSONA (CURP);**
- **CREATE NONCLUSTERED INDEX IX_INDICE_PER ON DBO.PERSONA (CURP) WHERE ESTATUS = 'ACTIVO';**
- **CREATE NONCLUSTERED INDEX IX_INDICE_PER ON DBO.PERSONA (CURP) INCLUDE (NOMBRE, APELLIDO_PATERNO, CIUDAD);**

- 
- **LAS RESTRICCIONES PRIMARY KEY Y UNIQUE CREAN ÍNDICES EN AUTOMÁTICO PARA LA COLUMNA EN LA QUE SE DEFINEN.**

USUARIOS: DEFINICIÓN

USUARIOS BASADOS EN LOGINS EN MASTER

- **ES EL TIPO MÁS COMÚN QUE SE PUEDE DECLARAR**
 - **USUARIO BASADO EN UN LOGIN CON CUENTA DE WINDOWS ACTIVE DIRECTORY**
 - **USUARIOS BASADOS EN UN LOGIN BASADO EN UN GRUPO DE WINDOWS**
 - **USUARIO BASADO EN UN LOGIN USANDO AUTENTICACIÓN DE SQL SERVER**

USUARIOS: SINTAXIS DE CREACIÓN

CREATE USER [ACTIVEDIRETORY\AGUST];

CREATE USER [GRUPO_WINDOWS\AGUST];

CREATE USER AGUST;

USUARIOS: DEFINICIÓN

USUARIOS QUE SE AUTENTICAN EN UNA BASE

- **PERMITE CONTROLAR A DONDE TENDRÁ ACCESO EL USUARIO**
 - **USUARIO BASADO EN UN USUARIO WINDOWS SIN INGRESO**
 - **USUARIO BASADO EN UN GRUPO WINDOWS SIN INGRESO**
 - **USUARIO EN SQL BASADO EN UN USUARIO DE AZURE ACTIVE DIRECTORY**
 - **USUARIO DE BASE DE DATOS CON PASSWORD**

USUARIOS: SINTAXIS DE CREACIÓN

ASIGNADOS A UNA BASE DE DATOS ESPECÍFICA

CREATE USER [ACTIVEDIRECTORY\AGUST];

CREATE USER [GRUPO_WINDOWS\AGUST];

CREATE USER AGUST;

CREATE USER [ACTIVEDIRECTORY\AGUST] FROM EXTERNAL PROVIDER;

CREATE USER AGUST WITH PASSWORD = '***';**

USUARIOS: DEFINICIÓN

USUARIO BASADO EN CREDENCIALES WINDOWS QUE SE CONECTAN MEDIANTE UN GRUPO DE LOGINS DE WINDOWS

- **USUARIO BASADO EN UN USUARIO DE WINDOWS QUE NO TIENE LOGIN, PERO PUEDE CONECTARSE AL MOTOR DE BASE DE DATOS A TRAVÉS DE LA MEMBRESÍA EN UN GRUPO WINDOWS.**
- **USUARIO BASADO EN UN GRUPO DE WINDOWS QUE NO TIENE LOGIN, PERO PUEDE CONECTARSE AL MOTOR DE BASE DE DATOS A TRAVÉS DE LA MEMBRESÍA EN UN GRUPO DE WINDOWS DIFERENTE.**

USUARIOS: DEFINICIÓN

USUARIOS QUE NO PUEDEN AUTENTICARSE

- **ESTOS USUARIOS NO PUEDEN INGRESAR A SQL**
 - **USUARIOS SIN LOGIN, NO PUEDEN INGRESAR PERO SE LES CONCEDEN PERMISOS**
 - **USUARIOS BASADOS EN UN CERTIFICADO, NO PUEDEN INGRESAR PERO PUEDEN CONCEDER PERMISOS Y PUEDEN FIRMAR MÓDULOS**
 - **USUARIO BASADO EN UNA LLAVE ASIMÉTRICA. NO PUEDE INGRESAR PERO PUEDE CONCEDER PERMISOS Y PUEDE FIRMAR MÓDULOS**

USUARIOS: SINTAXIS DE CREACIÓN

CREATE USER AGUST WITHOUT LOGIN;

CREATE USER AGUST FOR CERTIFICATE CARNATIONPRODUCTION50;

CREATE USER AGUST FROM ASYMMETRIC KEY PACIFICSALES09;

USUARIOS SIN LOGIN

SE CONFIGURAN PARA ASIGNAR PERMISOS ESPECIALES QUE LOS USUARIOS NO TIENEN. DE TAL FORMA QUE INGRESAN MEDIANTE UNA SUPLANTACIÓN DE USUARIO PARA REALIZAR LA OPERACIÓN.

LA SUPLANTACIÓN SE REALIZA CON EL SIGUIENTE CÓDIGO:

GRANT IMPERSONATE ON USER:: [USUARIO_SIN_LOGIN] TO [USUARIO]

EL SIGUIENTE PUNTO ES EJECUTAR COMO..

EXECUTE AS USER = 'USUARIO_SIN_LOGIN'

DISPARADORES: DEFINICIÓN

UN TRIGGER (O DISPARADOR) ES UNA CLASE ESPECIAL DE PROCEDIMIENTO ALMACENADO QUE SE EJECUTA AUTOMÁTICAMENTE CUANDO SE PRODUCE UN EVENTO EN EL SERVIDOR DE BASES DE DATOS.

DISPARADORES: TIPOS

DML: SE EJECUTAN CON OPERACIONES DE INSERT, UPDATE O DELETE EN UNA TABLA (SOBRE DATOS)

DDL: SE EJECUTAN CON OPERACIONES DE ALTER Y DROP (SOBRE OBJETOS)

DISPARADORES: SINTAXIS CREACIÓN DML

```
CREATE TRIGGER <NOMBRE_TRIGGER>  
ON <NOMBRE_TABLA>  
AFTER <INSERT,DELETE,UPDATE>  
AS  
BEGIN  
SET NOCOUNT ON;  
-- CÓDIGO DEL TRIGGER  
END
```

```
ALTER TRIGGER [DBO].[ISEXISTENCIA] ON [DBO].[LISTA_PEDIDO]  
FOR INSERT  
AS  
DECLARE @TOTAL INT  
SELECT @TOTAL = EXISTENCIAS FROM VEXISTENCIAS INNER JOIN  
INSERTED  
ON VEXISTENCIAS.IDARTICULO = INSERTED.IDARTICULO  
IF @TOTAL < 0  
BEGIN  
    ROLLBACK TRAN  
    RAISERROR('YA NO HAY DISPONIBILIDAD DEL  
PRODUCTO',2,1)  
END
```


DISPARADORES: SINTAXIS CREACIÓN DDL

```
CREATE TRIGGER <NOMBRE_TRIGGER>  
ON DATABASE  
FOR <DROP_TABLE, ALTER_TABLE>  
AS  
BEGIN  
END
```

```
CREATE TRIGGER TR_SEGURIDAD  
ON DATABASE FOR DROP_TABLE, ALTER_TABLE  
AS  
BEGIN  
RAISERROR ('NO ESTÁ PERMITIDO BORRAR NI MODIFICAR  
TABLAS', 16, 1)  
ROLLBACK TRANSACTION  
END
```

DISPARADORES: SINTAXIS

-- DESACTIVA EL TRIGGER ISEXISTENCIA

DISABLE TRIGGER ISEXISTENCIA ON LISTA_PEDIDO

GO

-- ACTIVA EL TRIGGER ISEXISTENCIA

ENABLE TRIGGER ISEXISTENCIA ON LISTA_PEDIDO

GO

-- DESACTIVA TODOS LOS TRIGGER DE LA TABLA LISTA_PEDIDO

ALTER TABLE LISTA_PEDIDO DISABLE TRIGGER ALL

GO

-- ACTIVA TODOS LOS TRIGGER DE LA TABLA LISTA_PEDIDO

ALTER TABLE LISTA_PEDIDO ENABLE TRIGGER ALL

DISPARADORES: TABLAS AUXILIARES

INSERTED: INSERT Y UPDATE, GUARDA DATOS INSERTADOS (ANTES DE HACER LOS CAMBIOS EN LA TABLA)

DELETED: UPDATE Y DELETE, GUARDA EL DATO A REEMPLAZAR O A ELIMINAR (DESPUES DE HACER LA OPERACIÓN)

CURSORES: DEFINICIÓN

- **ELEMENTO QUE REPRESENTARÁ A UN CONJUNTO DE DATOS DETERMINADO POR UNA CONSULTA T-SQL, EL CURSOR PERMITIRÁ RECORRER FILA A FILA, LEER Y EVENTUALMENTE MODIFICAR DICHO CONJUNTO DE RESULTADOS.**
- **→ ES UNA ESTRUCTURA DE DATOS CREADA EN MEMORIA RAM PRODUCTO DE UNA SENTENCIA SELECT Y QUE NOS PERMITE NAVEGAR DENTRO DE LAS FILAS PARA OBTENER LA INFORMACIÓN.**

CURSORES SIMPLES: ESTRUCTURA

- **1) DECLARACIÓN:**

DECLARE CSR_PERSONA CURSOR FOR SELECT ID, NOMBRE FROM PERSONA

- **2) APERTURA**

OPEN CSR_PERSONA

CURSORES SIMPLES: ESTRUCTURA

- **3) RECORRIDO Y ACCESO A LOS DATOS**

```
FETCH NEXT FROM CSR_PERSONA INTO @ID_PERSONA, @NOMBRE
```

```
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
```

```
    FETCH NEXT FROM CSR_PERSONA INTO @ID_PERSONA, @NOMBR
```

```
END
```

CURSORES SIMPLES: ESTRUCTURA

- **4) CIERRE Y LIBERACIÓN DE RECURSO**

CLOSE CSR_PERSONA

DEALLOCATE CSR_PERSONA

CURSORES SIMPLES: VALOR DE @@FETCH_STATUS

Valor	Descripción Description
0	La instrucción FETCH se ejecutó correctamente.
-1	La instrucción FETCH no se ejecutó correctamente o la fila estaba más allá del conjunto de resultados.
-2	Falta la fila capturada.