

# **BASES DE DATOS**

**Triggers**

# Agenda

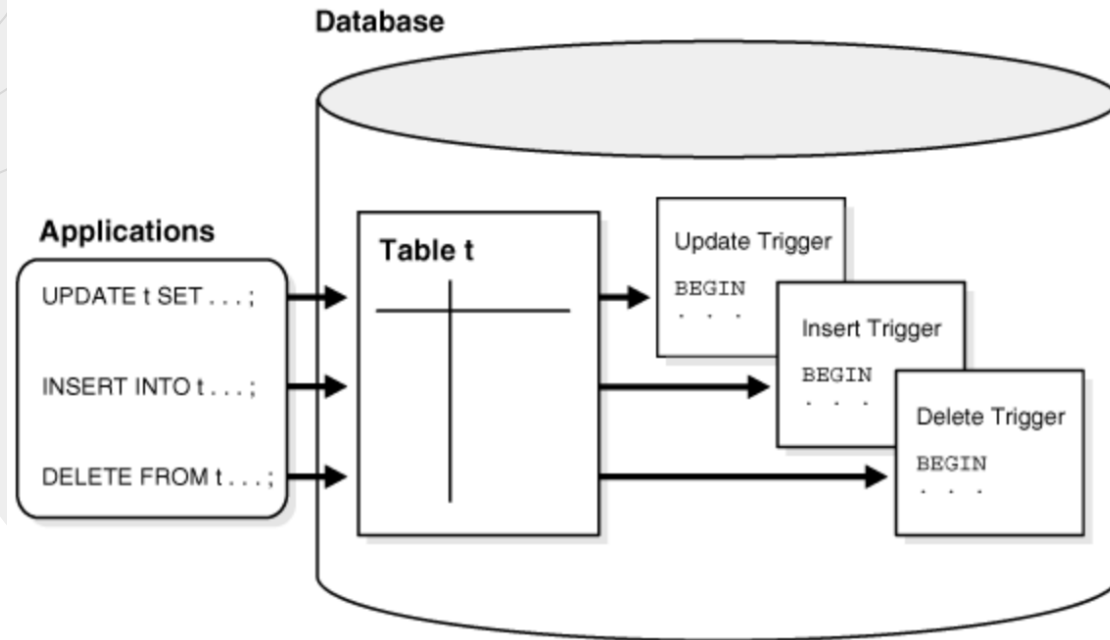


- ▣ Triggers
- ▣ Definición
- ▣ Tipos

# TRIGGERS

“

Es un objeto de BD compuesto por código que se ejecuta automáticamente ante ciertos eventos.



# TRIGGERS



Dependiendo del DBMS existen diferentes tipos de triggers:  
Eventos ...

- DML: Insert, Delete, Update (sobre tablas y views)
- DDL: Create, Alter, Drop, Grant, Revoke
- Sistema: Logon, Logoff, Startup, Shutdown.

# TRIGGERS

- Forman parte de la misma transacción que disparó el trigger.
- Los triggers de DMLs están asociados a tablas o vistas.

Ademas del evento y el objeto se define el momento de ejecución del trigger.

Dependiendo del DBMS existen diferentes momentos:

- Before (en Sql Server no existe)
- Instead of
- After

# TRIGGERS

- Se almacenan en las tablas del catálogo.

""

```
select * from sys.objects where type = 'TR'
```

```
select * from sys.triggers
```

```
select * from sys.all_sql_modules  
where object_id = 727673640
```

# TRIGGERS

Motivos para utilizar Triggers.

- Generar valores de columnas derivados.
- Prevenir operaciones inválidas.
- Forzar autorizaciones de seguridad.
- Forzar la integridad referencial entre diferentes BD.
- Implementar reglas de negocio.
- Proveer auditoría.
- Mantener replicas entre tablas.
- Generar estadísticas de operaciones.
- Modificar datos cuando las DML son ejecutadas contra Vistas.
- Implementar reglas de integridad.

Cuidado con los triggers en cascada !!

# Triggers

CREATE TRIGGER *nombre*  
ON *tabla*

Objeto sobre  
el que se  
dispara

[*BEFORE/INSTEAD OF/AFTER*] [*operacion*] AS

Acción que  
dispara el  
evento

Begin

...

...

end;

Momento en  
que se dispara



# Triggers – Pseudotablas

Pertenecen a SqlServer

Son tablas que se generan automaticamente cuando usamos triggers.

Nos permiten identificar las filas que estan involucradas en las operaciones.

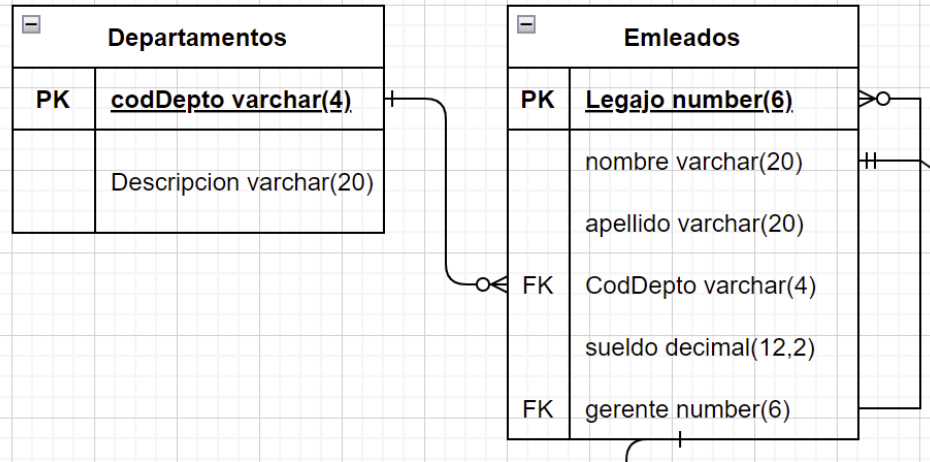
Tienen el mismo formato de la tabla/View base

Se generan según las operaciones ...

	INSERTED	DELETED
Insert	X	
Delete		X
Update	X	X

# Triggers – Ejemplo 1

“



CREATE VIEW empDeptoView AS

```
select e.legajo, e.nombre, e.sueldo, e.codDepto, d.descripcion
from empleados e join departamentos d
on e.docDepto = d.codDepto;
```

# Triggers – Ejemplo 1

```
CREATE TRIGGER insertaEmpTR
```

```
    ON empDeptoView
```

```
    INSTEAD OF INSERT AS
```

```
Begin
```

```
    declare @codDepto varchar(5);
```

```
    select @codDepto = codDepto from inserted;
```

```
    if not EXISTS (select 1 from departamentos
```

```
                    where codDepto. = @codDepto)
```

```
        Insert into departamentos (codDepto, Descripcion)
```

```
            select i.codDepto, i.Descripcion from inserted i;
```

```
    Insert into empleados(legajo, nombre, sueldo, codDepto)
```

```
        select i.legajo, i.nombre, i.sueldo, i.codDepto from inserted i;
```

```
end;
```

CUIDADO !!  
Ejemplo para un  
solo registro !!

# Triggers – Ejemplo 2

```
CREATE TRIGGER auditorTR
```

```
ON empleados
```

```
AFTER INSERT, UPDATE AS
```

```
Begin
```

```
    Insert into empAuditoria(legajo, sueldoAnt, sueldoNue, usuario, fecha)
```

```
        select i.legajo, d.sueldo, i.sueldo, SYSTEM_USER, GETDATE()
```

```
        from inserted i LEFT join deleted d on i.legajo = d.legajo;
```

```
end;
```

# Triggers – Ejemplo 3

```
CREATE TRIGGER sumaSaldo ON facturas_det
AFTER INSERT AS
--
declare detalleCur CURSOR FOR
    Select factura_num, renglon, producto_cod, cantidad from INSERTED;
declare @factura_num int, @renglon smallint,
        @producto_cod smallint, @cantidad smallint
Begin
    open detalleCur
    FETCH detalleCur INTO @factura_num, @renglon, @producto_cod, @cantidad
    ...
```

# Triggers – Ejemplo 3 (cont.)

```
WHILE (@@FETCH_STATUS = 0)
BEGIN
    if not exists (select 1 from productosVendidos where producto_cod = @producto_cod)
        insert into productosVendidos (producto_cod, cantidad)
            values (@producto_cod, @cantidad);
    else
        update productosVendidos
            set cantidad += @cantidad
            where producto_cod = @producto_cod;
    FETCH detalleCur INTO @factura_num, @renglon, @producto_cod, @cantidad
END
CLOSE detalleCur
DEALLOCATE detalleCur
END
```



# Preguntas ?