

# Modelos y bases de datos

## Integridad

CEIS

2018-02

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Modelo relacional

## Aspectos basicos

- ▶ **Aspecto estructural**

Los usuarios perciben la información de la base de datos como tablas y nada más que tablas

- ▶ **Aspecto de integridad**

Estas tablas satisfacen ciertas restricciones de integridad

- ▶ **Aspecto de manipulación**

Las operaciones disponibles permiten derivar tablas a partir de tablas

# Integridad

## El ideal: la regla de oro

Nunca debe permitirse una operación de actualización que deje cualquier tabla en un estado que viole su propio **predicado**.

## Un mecanismo: aspecto de integridad

Hacer que las tablas cumplan condiciones de integridad

# Agenda

## Integridad

Lo básico

**Mecanismos**

Nuevos envíos

## Integridad declarativa

## Acciones

Base de datos

Imperativo

Excepciones

## Integridad procedimental

Acciones referenciales

Disparadores

# Estrategias de integridad

## Declarativa

¿QUÉ? Indicar qué debe cumplirse

## Procedimental

¿ CÓMO? Indicar cómo debe cumplirse



# Estrategias de integridad

## Declarativa

¿QUÉ? Indicar qué debe cumplirse

- ▶ Restricción de atributo
- ▶ Restricción de tupla
- ▶ Restricción de tabla
- ▶ Restricción de base de datos

## Procedimental

¿ CÓMO? Indicar cómo debe cumplirse

# Estrategias de integridad

## Declarativa

¿QUÉ? Indicar qué debe cumplirse

- ▶ Restricción de atributo
- ▶ Restricción de tupla
- ▶ Restricción de tabla
- ▶ Restricción de base de datos

## Procedimental

¿CÓMO? Indicar cómo debe cumplirse

- ▶ Acciones de referencia
- ▶ Disparadores

# Agenda

## Integridad

Lo básico

Mecanismos

**Nuevos envíos**

## Integridad declarativa

## Acciones

Base de datos

Imperativo

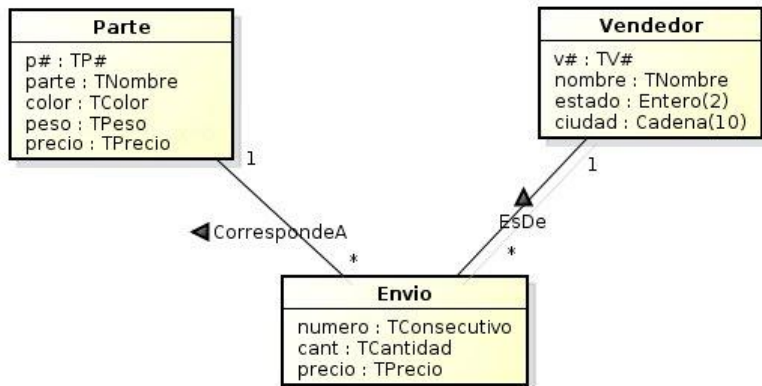
Excepciones

## Integridad procedimental

Acciones referenciales

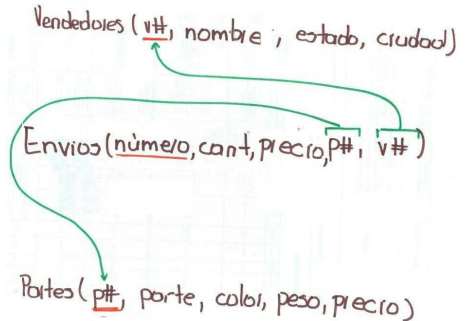
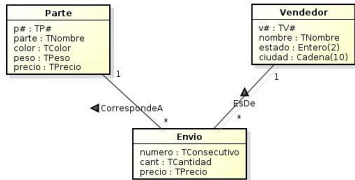
Disparadores

# Envíos extendido



¿A modelo relacional?

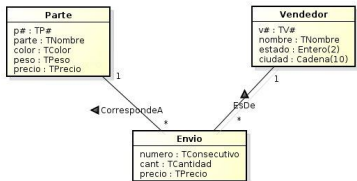
# Envios



## Regla de oro

- ¿Cuántos predicados debemos definir?
- ¿Predicado para ENVIOS?

# Envios



Vendedores (v#, nombre, estado, ciudad)

Envios (número, cant, precio, p#, v#)

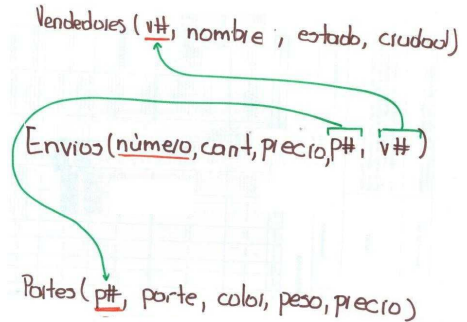
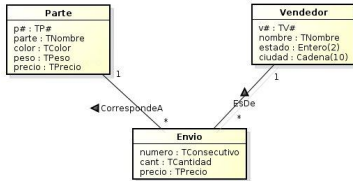
Partes (p#, parte, color, peso, precio)

## Reglas de integridad

Revisen el modelo relacional

- ¿Cuántas condiciones de integridad están definidas? ¿Cuáles?

# Envios



## Reglas de integridad

- ▶ Implementen la tabla ENVIOS: estructura y restricciones de clave
- ▶ ¿Cuántas regla de integridad están definidas?
- ▶ ¿Otra regla que aporte al predicado de ENVIOS?

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

### Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores



# Integridad declarativa

## Mecanismos generales

## Mecanismos simplificados

1. Tipo de dato
2. Nulidad
3. Claves Primarias - Foráneas - Únicas

# Integridad declarativa

## Mecanismos generales

1. Definición de dominio (De atributo)
2. Condición de tupla (De atributo De tupla)
3. Aserción general (De tabla De base de datos)

## Mecanismos simplificados

1. Tipo de dato
2. Nulidad
3. Claves Primarias - Foráneas - Únicas

# Integridad declarativa

## Mecanismos generales

### 1. Definición de dominio (De atributo)

```
CREATE DOMAIN TCOLOR AS VARCHAR(10) CONSTRAINT  
CHECK (VALUE IN ('Rojo','Amarillo','Verde'))
```

### 2. Condición de tupla (De atributo De tupla)

### 3. Aserción general (De tabla De base de datos)

## Mecanismos simplificados

### 1. Tipo de dato

### 2. Nulidad

### 3. Claves Primarias - Foráneas - Únicas

# Integridad declarativa

## Mecanismos generales

### 1. Definición de dominio (De atributo)

```
CREATE DOMAIN TCOLOR AS VARCHAR(10) CONSTRAINT  
CHECK (VALUE IN ('Rojo','Amarillo','Verde'))
```

### 2. Condición de tupla (De atributo De tupla)

```
ALTER TABLE VENDEDORES ADD CONSTRAINT CK_VENDEDORES_V#  
CHECK (V LIKE 'V%')
```

### 3. Aserción general (De tabla De base de datos)

## Mecanismos simplificados

1. Tipo de dato
2. Nulidad
3. Claves Primarias - Foráneas - Únicas

# Integridad declarativa

## Mecanismos generales

### 1. Definición de dominio (De atributo)

```
CREATE DOMAIN TCOLOR AS VARCHAR(10) CONSTRAINT  
CHECK (VALUE IN ('Rojo','Amarillo','Verde'))
```

### 2. Condición de tupla (De atributo De tupla)

```
ALTER TABLE VENDEDORES ADD CONSTRAINT CK_VENDEDORES_V#  
CHECK (V LIKE 'V%')
```

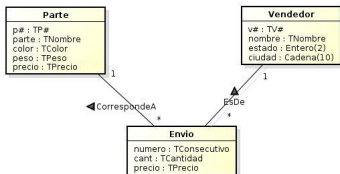
### 3. Aserción general (De tabla De base de datos)

```
CREATE ASSERTION REPARTIDO  
CHECK (NOT EXISTS (SELECT * FROM ENVIOS  
..... GROUP BY VENDEDOR HAVING COUNT(*) > 100))  
  
NOT EXISTS
```

## Mecanismos simplificados

1. Tipo de dato
2. Nulidad
3. Claves Primarias - Foráneas - Únicas

# Venderes y partes



Vendedores (v#, nombre, estado, ciudad)

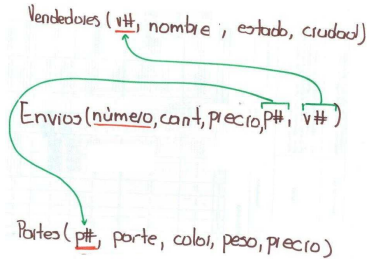
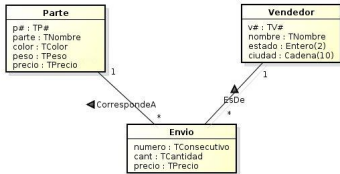
Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

¿?

- ▶ TPeso es un real, con dos decimales, que equivale al peso en kilogramos (0 ... 100]

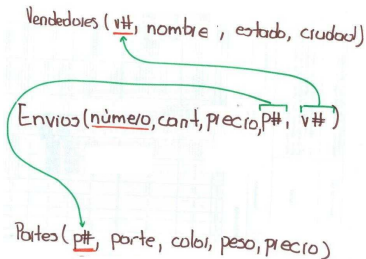
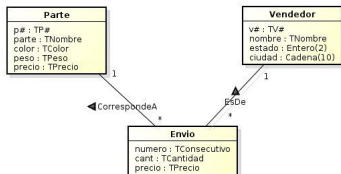
# Venderes y partes



¿?

- ▶ TPeso es un real, con dos decimales, que equivale al peso en kilogramos (0 ... 100]
- ▶ Los ciudades puede ser: BOGOTA, CALI, MEDELLIN

# Venderes y partes

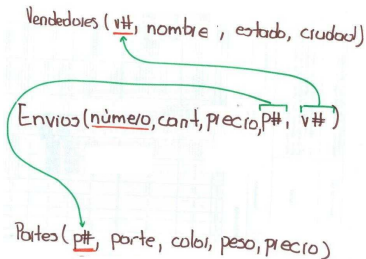
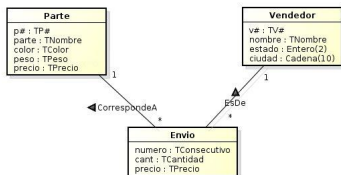


¿?

- ▶ `TPeso` es un real, con dos decimales, que equivale al peso en kilogramos (0 ... 100]
- ▶ Las ciudades puede ser: BOGOTA, CALI, MEDELLIN
- ▶ Las partes de menos de 1 kg deben tener costos inferiores a 100.000



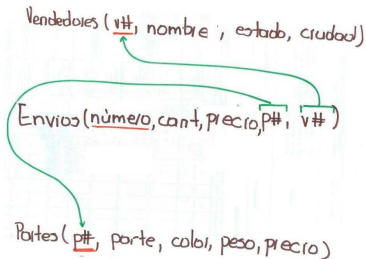
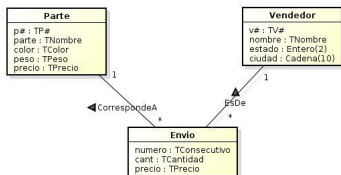
# Venderes y partes



¿?

- ▶ TPeso es un real, con dos decimales, que equivale al peso en kilogramos (0 ... 100]
- ▶ Las ciudades puede ser: BOGOTA, CALI, MEDELLIN
- ▶ Las partes de menos de 1 kg deben tener costos inferiores a 100.000
- ▶ Los precios de las partes en los envíos nunca deben bajar

# Venderes y partes



¿?

- ▶ TPeso es un real, con dos decimales, que equivale al peso en kilogramos (0 ... 100]
- ▶ Las ciudades puede ser: BOGOTA, CALI, MEDELLIN
- ▶ Las partes de menos de 1 kg deben tener costos inferiores a 100.000
- ▶ Los precios de las partes en los envíos nunca deben bajar
- ▶ Las partes pesadas (más de 50 kilogramos) deben venir de París

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

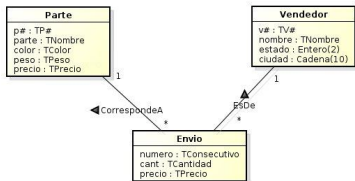
- Excepciones

## Integridad procedimental

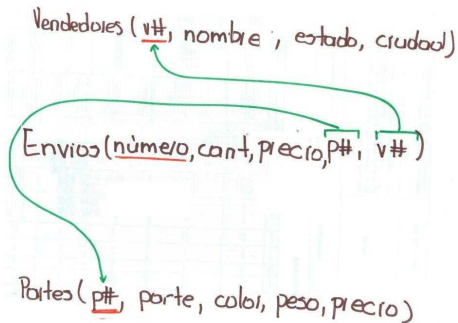
- Acciones referenciales

- Disparadores

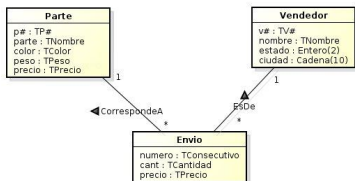
# Vendedores y partes



¿Las tres acciones básicas?



# Vendedores y partes



Vendedores (v#, nombre, estado, ciudad)

Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

¿Las tres acciones básicas? **INSERT DELETE UPDATE**

1. `INSERT INTO VENDEDORES (v#,proveedor,estado,ciudad) VALUES ('V1','Smith',20,'Londres');`
2. `DELETE FROM VENDEDORES WHERE v#='P2';`
3. `UPDATE VENDEDORES SET estado=2*estado, ciudad='Roma' WHERE ciudad='Paris';`

¿Qué hacen? ¿Errores?

# Partes y vendedores

Vendedores ( v#, nombre, estado, ciudad )

Envios ( número, cant, precio, p#, v# )

Partes ( p#, parte, color, peso, precio )

VENDEDORES

V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

ENVIOS

NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

PARTES

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Lava	Azul	12	200
P6	Engrane	Rojo	19	56

## Ejercicios

1. Adicionar un nuevo envío
2. Eliminar la parte que se puede eliminar
3. Incrementar el precio de todas las partes un porcentaje

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo**

- Excepciones

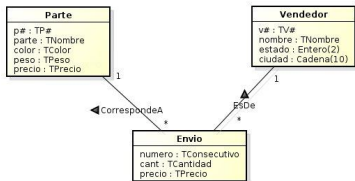
## Integridad procedimental

- Acciones referenciales

- Disparadores



# Venderes y partes



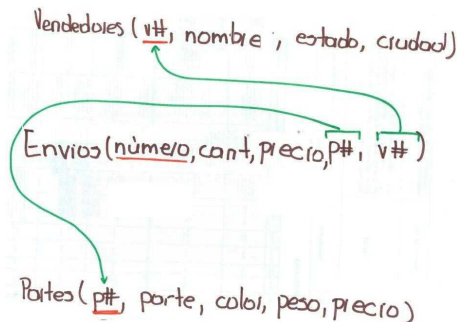
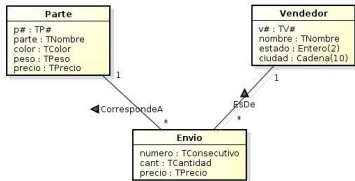
¿Las tres acciones básicas?

Vendedores (v#, nombre, estado, ciudad)

Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

# Venderes y partes



¿Las tres acciones básicas? **Asignar Leer Escribir**

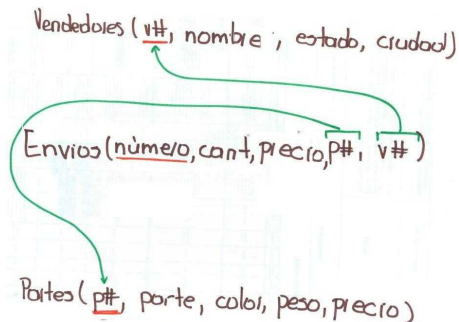
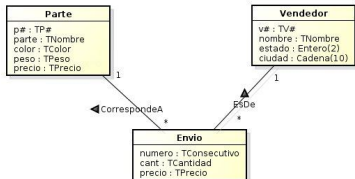
1. `SELECT estado INTO xEstado FROM VENEDORES WHERE v#='V2';`

2.

`CURSOR datos IS SELECT estado, ciudad FROM VENEDORES WHERE estado>10;`

¿Qué hacen? **Errores?** **Funciones?**

# Venderes y partes



¿Las tres acciones básicas? **Asignar Leer Escribir**

```
1. SELECT estado INTO xEstado FROM VENDEDORES WHERE v#='V2';
```

**NO\_DATA\_FOUND (100), TOO\_MANY\_ROWS (-1422)**

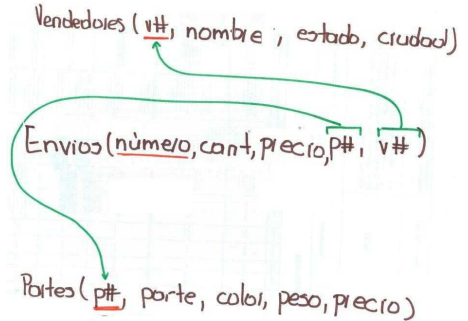
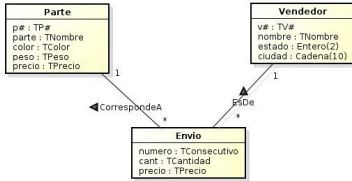
2.

```
CURSOR datos IS SELECT estado, ciudad FROM VENDEDORES WHERE estado>10;
```

datos%FOUND,datos%NOTFOUND

¿Qué hacen? **Errores?** **Funciones?**

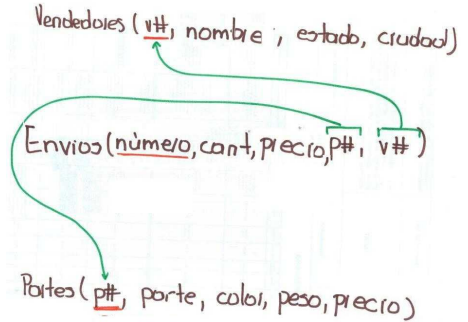
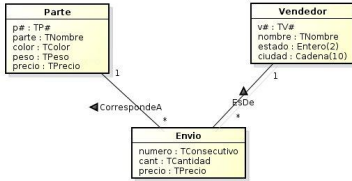
# Partes y vendedores



## Ejercicios

1. Calcular el número a asignar al siguiente envío

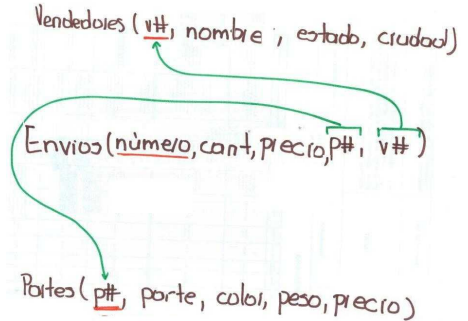
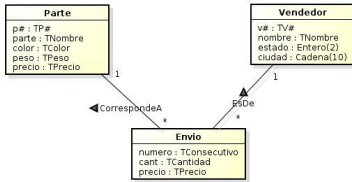
# Partes y vendedores



## Ejercicios

1. Calcular el número a asignar al siguiente envío
2. Calcular el precio de una parte

# Partes y vendedores



## Ejercicios

1. Calcular el número a asignar al siguiente envío
2. Calcular el precio de una parte
3. Tener la información del número de envíos de cada ciudad

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

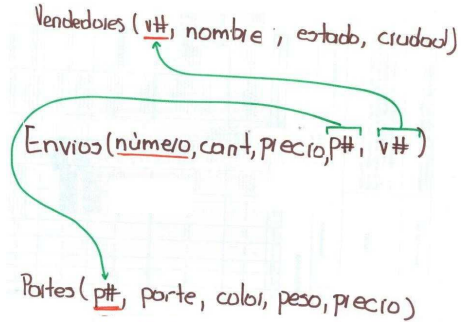
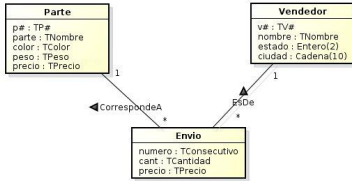
- Excepciones**

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Partes y vendedores



## Del sistema

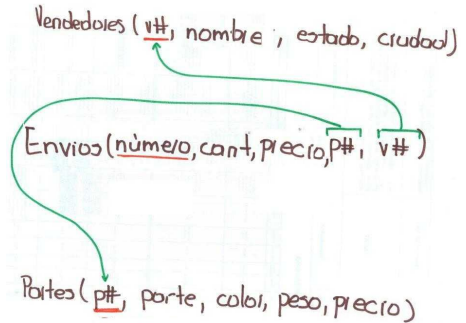
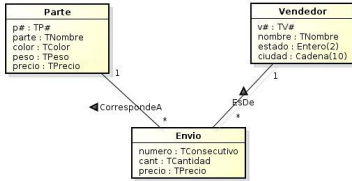
SELECT INTO

NO\_DATA\_FOUND (100), TOO\_MANY\_ROWS (-1422)

Excepción: mensaje + número



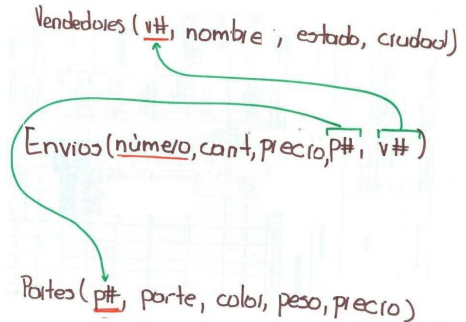
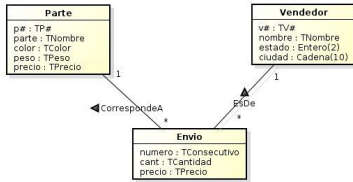
# Partes y vendedores



## Excepciones personalizadas

1. `RAISE_APPLICATION_ERROR(numero [-20001,-20999],mensaje)`

# Partes y vendedores



## Excepciones personalizadas

1. `RAISE_APPLICATION_ERROR(numero [-20001,-20999],mensaje)`
2. Lanzar error indicando que no se pueden eliminar los envíos

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

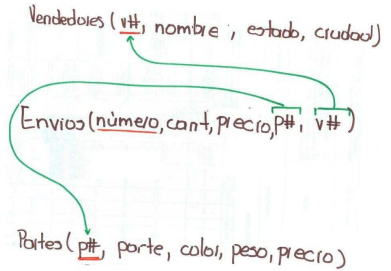
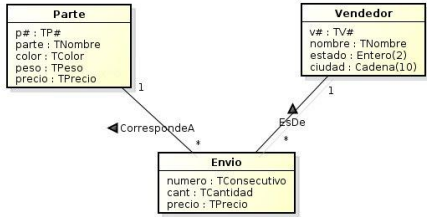
- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Partes y proveedores



¿Podemos?

1. Borrar una parte
2. Borrar un vendedor
3. Modificar el código de una parte

# Partes y proveedores

VENDEDORES			
V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrase	Rojo	19	56

ENVIOS				
NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

Vendedores (v#, nombre, estado, ciudad)

Envios (número, cant, precio, p#, v#)

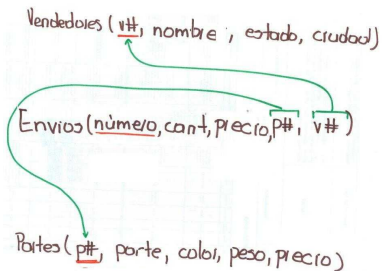
Partes (p#, parte, color, peso, precio)

¿Podemos?

1. Borrar una parte ¿Cuáles si?
2. Borrar un vendedor ¿Cuáles si?
3. Modificar el código de una parte ¿Cuáles si?

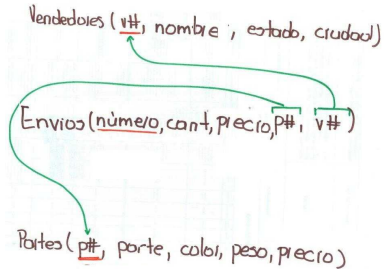
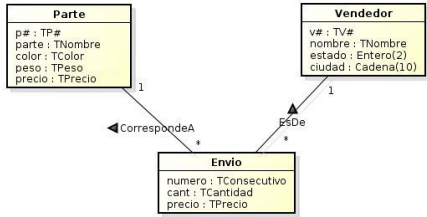
# Actualizar

```
CONSTRAINT FK_<nombre_restriccion> FOREIGN KEY (<nombre_atributo>, ...)
REFERENCES <nombre_tabla> (<nombre_atributo>, ...)
[ON DELETE (RESTRICT | CASCADE | SET NULL | NO ACTION)]
[ON UPDATE (RESTRICT | CASCADE | SET NULL | NO ACTION)]
```



- ▶ **CASCADE**: propaga la acción, si es necesario.
- ▶ **RESTRICT** o **NO ACTION** **PENDIENTE**
- ▶ **SET NULL**: deja en NULL, si es necesario

# Partes y vendedores



## Propuestas

1. Dejar borrar una parte. Al borrarla se borran sus envíos.
2. Dejar borrar vendedores. Al borrarlo queda en desconocido sus envíos.
3. Dejar modificar el código de una parte. Deja todo con el mismo código.



# Partes y vendedores

VENEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

Vendedores (v#, nombre, estado, ciudad)

Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Propuestas

1. Dejar borrar una parte. Al borrarla se borran sus envíos.
2. Dejar borrar vendedores. Al borrarlo queda en desconocido sus envíos.
3. Dejar modificar el código de una parte. Deja todo con el mismo código.

DELETE FROM PARTES WHERE p#='P1' ¿Qué pasa?

# Partes y vendedores

VENEDORES			
V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

ENVIOS				
NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

Vendedores (v#, nombre, estado, ciudad)

Envios (número, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Propuestas

1. Dejar borrar una parte. Al borrarla se borran sus envíos.
2. Dejar borrar vendedores. Al borrarlo queda en desconocido sus envíos.
3. Dejar modificar el código de una parte. Deja todo con el mismo código.

DELETE FROM VENEDORES WHERE v#='V1' ¿Qué pasa?

# Partes y vendedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

Vendedores (v#, nombre, estado, ciudad)

Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Propuestas

1. Dejar borrar una parte. Al borrarla se borran sus envíos.
2. Dejar borrar vendedores. Al borrarlo queda en desconocido sus envíos.
3. Dejar modificar el código de una parte. Deja todo con el mismo código.

UPDATE PARTES SET p#='P0' WHERE P#='P5' ¿Qué pasa?

# Agenda

## Integridad

- Lo básico

- Mecanismos

- Nuevos envíos

## Integridad declarativa

## Acciones

- Base de datos

- Imperativo

- Excepciones

## Integridad procedimental

- Acciones referenciales

- Disparadores

# Disparadores

**SUBPROGRAMA** que se ejecuta  
en **MOMENTOS** específicos

asociado a **EVENTOS**,

sobre una **TABLA**.

Se ejecuta una o más **VECES**

si se cumple una **CONDICIÓN**

Se pueden declarar variables

y las acciones a realizar

```
CREATE TRIGGER nombre
[ BEFORE | AFTER | INSTEAD OF ]
[ DELETE | INSERT | UPDATE [ OF nombre columnas ] ]
ON nombre Tabla
[ FOR EACH ROW [ WHEN (condición) ] ]
DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```

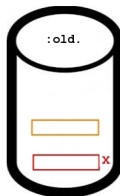


:new.

INSERT

UPDATE

DELETE



# Disparadores

**SUBPROGRAMA** que se ejecuta  
en **MOMENTOS** específicos

**AFTER BEFORE INSTANCE OF**

asociado a **EVENTOS**,

**INSERT DELETE UPDATE**

sobre una **TABLA**.

**ON**

Se ejecuta una o más **VECES**

**FOR EACH ROW**

si se cumple una **CONDICIÓN**

**WHEN**

Se pueden declarar variables

**DECLARE**

y las acciones a realizar

**AFTER BEGIN END**

```
CREATE TRIGGER nombre
[ BEFORE | AFTER | INSTEAD OF ]
[ DELETE | INSERT | UPDATE [ OF nombre columnas ] ]
ON nombre Tabla
[ FOR EACH ROW [ WHEN (condición) ] ]
DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```

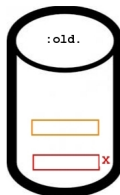


**INSERT**

**UPDATE**

**DELETE**

**:new.**



# Partes y proveedores

```
CREATE OR REPLACE TRIGGER UP_PARTESCOLOR
BEFORE UPDATE OF COLOR ON PARTES
FOR EACH ROW
DECLARE
    VECES NUMBER;
BEGIN
    IF :OLD.COLOR IN ('AMARILLO', 'AZUL', 'ROJO') THEN
        :NEW.COLOR := :OLD.COLOR;
    ELSE
        SELECT COUNT(*) INTO VECES FROM PARTES WHERE COLOR=:NEW.COLOR;
        IF (VECES > 30) THEN
            :NEW.COLOR:= 'GENERICO';
        END IF;
    END IF;
END;
```

```
CREATE TRIGGER nombre
[ BEFORE | AFTER | INSTEAD OF ]
[DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]
DECLARE
    -- Definición de variables
BEGIN
    -- Instrucciones del disparador
END nombre;
```



## Mo: Actualizar el color de una parte

1. ¿Qué pasa si queremos cambiar el color de una parte ROJA a VERDE? ¿Por qué?

# Partes y proveedores

```
CREATE OR REPLACE TRIGGER UP_PARTESCOLOR
BEFORE UPDATE OF COLOR ON PARTES
FOR EACH ROW
DECLARE
    VECES NUMBER;
BEGIN
    IF :OLD.COLOR IN ('AMARILLO', 'AZUL', 'ROJO') THEN
        :NEW.COLOR := :OLD.COLOR;
    ELSE
        SELECT COUNT(*) INTO VECES FROM PARTES WHERE COLOR=:NEW.COLOR;
        IF (VECES > 30) THEN
            :NEW.COLOR:= 'GENERICO';
        END IF;
    END IF;
END;
```

```
CREATE TRIGGER nombre
[ BEFORE | AFTER | INSTEAD OF ]
[DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]
DECLARE
    -- Definición de variables
BEGIN
    -- Instrucciones del disparador
END nombre;
```



## Mo: Actualizar el color de una parte

1. ¿Qué pasa si queremos cambiar el color de una parte ROJA a VERDE? ¿Por qué?
2. ¿Qué pasa si queremos cambiar el color de una parte NARANJA a VERDE? Es la primera parte verde. ¿Por qué?



# Partes y proveedores

```
CREATE OR REPLACE TRIGGER UP_PARTESCOLOR
BEFORE UPDATE OF COLOR ON PARTES
FOR EACH ROW
DECLARE
    VECES NUMBER;
BEGIN
    IF :OLD.COLOR IN ('AMARILLO', 'AZUL', 'ROJO') THEN
        :NEW.COLOR := :OLD.COLOR;
    ELSE
        SELECT COUNT(*) INTO VECES FROM PARTES WHERE COLOR=:NEW.COLOR;
        IF (VECES > 30) THEN
            :NEW.COLOR:= 'GENERICO';
        END IF;
    END IF;
END;
```

```
CREATE TRIGGER nombre
[ BEFORE | AFTER | INSTEAD OF ]
[ DELETE | INSERT | UPDATE [ OF nombre columnas ] ]
ON nombre Tabla
[ FOR EACH ROW [ WHEN (condición) ] ]
DECLARE
    -- Definición de variables
BEGIN
    -- Instrucciones del disparador
END nombre;
```

## Mo: Actualizar el color de una parte

1. ¿Qué pasa si queremos cambiar el color de una parte ROJA a VERDE? ¿Por qué?
2. ¿Qué pasa si queremos cambiar el color de una parte NARANJA a VERDE? Es la primera parte verde. ¿Por qué?
3. ¿Qué pasa si queremos cambiar el color de una parte NARANJA a VERDE? Ya hay treinta verdes. ¿Por qué?

# Partes y proveedores

VENDEDORES

V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

PARTES

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

ENVIOS

NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

Vendedores (v#, nombre, estado, ciudad)

Envios (numero, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Ad: Adicionar un envío

1. ¿Instrucción actual para adicionar un nuevo envío de 25 partes de P1 del proveedor V1?

# Partes y proveedores

VENDEDORES

V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

PARTES

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

ENVIOS

NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

Vendedores (v#, nombre, estado, ciudad)

Envios (número, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Ad: Adicionar un envío

1. ¿Instrucción actual para adicionar un nuevo envío de 25 partes de P1 del proveedor V1?
2. ¿Instrucción deseada con datos mínimos? ¿Qué puede calcular el sistema?

# Partes y proveedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

Vendedores (v#, nombre, estado, ciudad)

Envios (número, cant, precio, p#, v#)

Partes (p#, parte, color, peso, precio)

## Ad: Adicionar un envío

1. ¿Instrucción actual para adicionar un nuevo envío de 25 partes de P1 del proveedor V1?
2. ¿Instrucción deseada con datos mínimos? ¿Qué puede calcular el sistema?
3. ¿Cómo lograrlo?

# Partes y proveedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Lava	Azul	12	200
P6	Engrane	Rojo	19	56

NUMERO	V#	P#	CANT	PRECIO
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
|
| [FOR EACH ROW [WHEN (condición)]]
|
DECLARE
-- Definición de variables
|
BEGIN
-- Instrucciones del disparador
|
END nombre;
```



Ad: Adicionar un envío

1. Disparador ENVIOS\_AD : Adicionar un envío con datos mínimos

# Partes y proveedores

VENDEDORES				ENVÍOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Lava	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
|
| [FOR EACH ROW [WHEN (condición)]]
|
DECLARE
-- Definición de variables
|
BEGIN
-- Instrucciones del disparador
|
END nombre;
```



## Ad: Adicionar un envío

1. Disparador ENVIOS\_AD : Adicionar un envío con datos mínimos
2. Seguimiento a adicionar un nuevo envío de 25 partes de P1 del proveedor V1

# Partes y vendedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```



Mo: Actualizar el precio de una parte, no deben bajar

1. Instrucción para subir el precio de todas las partes a 100

# Partes y vendedores

VENDEDORES

V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark		Londres
V5	Adams	3	Atenas

ENVIOS

NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

PARTES

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```



Mo: Actualizar el precio de una parte, no deben bajar

1. Instrucción para subir el precio de todas las partes a 100
2. ¿Qué pasa? ¿Qué desearíamos que pasara?



## Partes y vendedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160
				6	V2	P1	300	20
				7	V2	P2	400	110
				8	V3	P2	200	110
				9	V4	P3	200	60
				10	V4	P4	300	40
				11	V4	P5	400	200

PARTES				
P#	PARTIE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
| BEFORE | AFTER | INSTEAD OF |
| DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```

Mo: Actualizar el precio de una parte, no deben bajar

1. Instrucción para subir el precio de todas las partes a 100
2. ¿Qué pasa? ¿Qué deseábamos que pasara?
3. Disparador PARTES\_MO\_PRECIO : Modificar el precio de una parte

## Partes y vendedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160
				6	V2	P1	300	20
				7	V2	P2	400	110
				8	V3	P2	200	110
				9	V4	P3	200	60
				10	V4	P4	300	40
				11	V4	P5	400	200

PARTES				
P#	PARTIE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
| BEFORE | AFTER | INSTEAD OF |
| DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```

Mo: Actualizar el precio de una parte, no deben bajar

1. Instrucción para subir el precio de todas las partes a 100
2. ¿Qué pasa? ¿Qué desearíamos que pasara?
3. Disparador PARTES\_M0\_PRECIO : Modificar el precio de una parte
4. Seguimiento

# Partes y vendedores

VENDEDORES

V#	NOMBRE	ESTADO	CIUDAD
V1	Smith	2	Londres
V2	Jones	1	Paris
V3	Blake	3	Paris
V4	Clark	3	Londres
V5	Adams		Atenas

ENVÍOS

NUMERO	V#	P#	CANT	PRECIO
1	V1	P1	300	20
2	V1	P2	200	100
3	V1	P3	400	40
4	V1	P4	200	40
5	V1	P5	100	160
6	V2	P1	300	20
7	V2	P2	400	110
8	V3	P2	200	110
9	V4	P3	200	60
10	V4	P4	300	40
11	V4	P5	400	200

PARTES

P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```



El: No se deben poder eliminar los envíos

1. Instrucción para eliminar los envíos de 'P4'

## Partes y vendedores

VENDEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NÚMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark	3	Londres	4	V1	P4	200	40
V5	Adams		Atenas	5	V1	P5	100	160
			6	V2	P1	300	20	
			7	V2	P2	400	110	
			8	V3	P2	200	110	
			9	V4	P3	200	60	
			10	V4	P4	300	40	
			11	V4	P5	400	200	

PARTES				
P#	PARTIE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
| BEFORE | AFTER | INSTEAD OF |
| DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
[FOR EACH ROW [WHEN (condición)]]

DECLARE
-- Definición de variables

BEGIN
-- Instrucciones del disparador

END nombre;
```

El: No se deben poder eliminar los envíos

1. Instrucción para eliminar los envíos de 'P4'
2. ¿Qué pasa? ¿Qué desearíamos que pasara?

# Partes y vendedores

VENEDORES				ENVIOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

PARTES				
P#	PORTE	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
|
| [FOR EACH ROW [WHEN (condición)]]
|
DECLARE
-- Definición de variables
|
BEGIN
-- Instrucciones del disparador
|
END nombre;
```



El: No se deben poder eliminar los envíos

1. Instrucción para eliminar los envíos de 'P4'
2. ¿Qué pasa? ¿Qué desearíamos que pasara?
3. Disparador ENVIOS\_EL : Eliminar un envío

# Partes y vendedores

VENDEDORES				ENVÍOS				
V#	NOMBRE	ESTADO	CIUDAD	NUMERO	V#	P#	CANT	PRECIO
V1	Smith	2	Londres	1	V1	P1	300	20
V2	Jones	1	Paris	2	V1	P2	200	100
V3	Blake	3	Paris	3	V1	P3	400	40
V4	Clark		Londres	4	V1	P4	200	40
V5	Adams	3	Atenas	5	V1	P5	100	160

PARTES				
P#	PARTI	COLOR	PESO	PRECIO
P1	Tuerca	Rojo	12	20
P2	Perno	Verde	17	110
P3	Tornillo	Azul	17	60
P4	Tornillo	Rojo	14	40
P5	Leva	Azul	12	200
P6	Engrane	Rojo	19	56

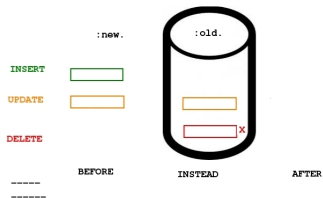
```
CREATE TRIGGER nombre
|
| [ BEFORE | AFTER | INSTEAD OF ]
| [DELETE|INSERT|UPDATE [OF nombre columnas]]
ON nombre Tabla
|
| [FOR EACH ROW [WHEN (condición)]]
|
DECLARE
-- Definición de variables
|
BEGIN
-- Instrucciones del disparador
|
END nombre;
```



El: No se deben poder eliminar los envíos

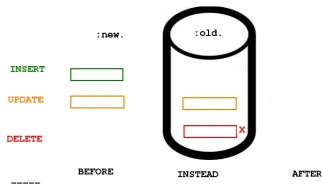
1. Instrucción para eliminar los envíos de 'P4'
2. ¿Qué pasa? ¿Qué desearíamos que pasara?
3. Disparador ENVIOS\_EL : Eliminar un envío
4. Seguimiento

# Triggers



¿Para qué?

# Triggers

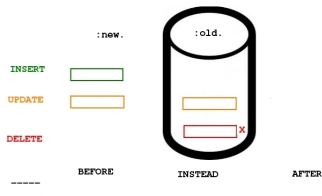


¿Para qué?

1. Implementar reglas de integridad
2. Mover lógica de aplicación a datos



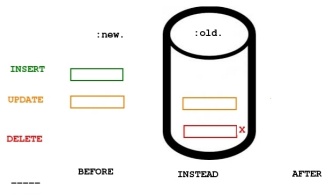
# Triggers



## ¿Para qué?

1. Implementar reglas de integridad
  - ▶ Evitando acciones no deseadas
  - ▶ Automatizando acciones deseadas
2. Mover lógica de aplicación a datos

# Triggers



## ¿Para qué?

1. Implementar reglas de integridad
  - ▶ Evitando acciones no deseadas
  - ▶ Automatizando acciones deseadas
2. Mover lógica de aplicación a datos
  - ▶ Centralizando acciones
  - ▶ Aumentando la eficiencia