

# BD 2 - Trigger in Postgres-

Luca Cosmo

Università Ca' Foscari Venezia



Università  
Ca' Foscari  
Venezia

# Esercizio 1

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Nessun produttore di PC può anche produrre laptop

- Quali operazioni possono violare l'invariante?
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

# Esercizio 1

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Nessun produttore di PC può anche produrre laptop

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE su Product**
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

# Esercizio 1

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Nessun produttore di PC può anche produrre laptop

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE su Product**
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

# Esercizio 1

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Nessun produttore di PC può anche produrre laptop

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE su Product**
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante? **Bloccare l'operazione (BEFORE)**

# Soluzione

Nessun produttore di PC può anche produrre laptop:

```
CREATE TRIGGER NoPCLaptop
BEFORE INSERT OR UPDATE
ON Product
FOR EACH ROW
EXECUTE FUNCTION no_pc_laptop();
```

```
CREATE FUNCTION no_pc_laptop() RETURNS trigger AS $$
BEGIN
    IF (NEW.type = 'pc' AND NEW.maker IN (SELECT maker
        FROM Product WHERE type = 'laptop')) THEN
        RETURN NULL;
    END IF;
    IF (NEW.type = 'laptop' AND NEW.maker IN (SELECT maker
        FROM Product WHERE type = 'pc')) THEN
        RETURN NULL;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

## Esercizio 2

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Se un laptop ha più memoria di un PC, allora deve anche costare di più di tale PC

- Quali operazioni possono violare l'invariante?
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

## Esercizio 2

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Se un laptop ha più memoria di un PC, allora deve anche costare di più di tale PC

- Quali operazioni possono violare l'invariante? **INSERT** o **UPDATE** su Laptop o su PC
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?



## Esercizio 2

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Se un laptop ha più memoria di un PC, allora deve anche costare di più di tale PC

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE su Laptop o su PC**
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

## Esercizio 2

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Se un laptop ha più memoria di un PC, allora deve anche costare di più di tale PC

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE su Laptop o su PC**
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante? **Bloccare l'operazione (BEFORE)**

# Soluzione

Se un laptop ha più memoria di un PC, allora deve anche costare di più:<sup>1</sup>

```
CREATE TRIGGER NoCheapLaptop
BEFORE INSERT OR UPDATE ON Laptop
FOR EACH ROW EXECUTE FUNCTION no_cheap_laptop();
```

```
CREATE FUNCTION no_cheap_laptop() RETURNS trigger AS $$
BEGIN
    IF EXISTS(SELECT * FROM pc WHERE NEW.ram > ram AND
              NEW.price <= price) THEN
        RETURN NULL;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

---

<sup>1</sup>Per esercizio: definizione dell'altro trigger su PC

## Esercizio 3

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Il modello di un prodotto deve essere anche il modello di un PC, di un laptop oppure di una stampante

- Quali operazioni possono violare l'invariante?
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

## Esercizio 3

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Il modello di un prodotto deve essere anche il modello di un PC, di un laptop oppure di una stampante

- Quali operazioni possono violare l'invariante? **INSERT** o **UPDATE** su **Product**, **DELETE** sulle altre tabelle
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no?
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

## Esercizio 3

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Il modello di un prodotto deve essere anche il modello di un PC, di un laptop oppure di una stampante

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE** su **Product**, **DELETE** sulle altre tabelle
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante?

## Esercizio 3

```
Product(maker, model, type)
PC(model*, speed, ram, hd, price)
Laptop(model*, speed, ram, hd, screen, price)
Printer(model*, color, type, price)
```

Il modello di un prodotto deve essere anche il modello di un PC, di un laptop oppure di una stampante

- Quali operazioni possono violare l'invariante? **INSERT o UPDATE** su **Product**, **DELETE** sulle altre tabelle
- Il mantenimento dell'invariante può essere controllato per ogni riga coinvolta dall'operazione oppure no? **Sì, è un controllo locale**
- Cosa bisogna fare prima o dopo dell'operazione per garantire il mantenimento dell'invariante? **Bloccare l'operazione (BEFORE) nel primo caso, cancellare anche da Product (AFTER) nel secondo caso**

# Soluzione

Il modello di un prodotto deve essere anche il modello di un PC, di un laptop oppure di una stampante:<sup>2</sup>

```
CREATE TRIGGER Wellformed
BEFORE INSERT OR UPDATE ON Product
FOR EACH ROW EXECUTE FUNCTION check_wellformed();
```

```
CREATE FUNCTION check_wellformed() RETURNS trigger AS $$
BEGIN
    IF NEW.model IN (SELECT model FROM Laptop UNION
                     SELECT model from PC UNION
                     SELECT model from Printer) THEN
        RETURN NEW;
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

Si noti che il controllo poteva essere reso più preciso, guardando anche il valore di `NEW.type` e controllando la rispettiva tabella.

---

<sup>2</sup>Per esercizio: definizione degli altri tre trigger



# Dipendenze Circolari

Uh, oh! Con l'ultimo trigger siamo finiti in uno scenario delicato:

- per inserire nella tabella PC dobbiamo avere già quel modello in Product, poiché model è chiave esterna di PC
- per inserire un computer nella tabella Product dobbiamo avere già quel modello in PC, come richiesto dal trigger

Come risolvere questo problema?

- 1 workaround: rimuovere il vincolo FOREIGN KEY, inserire il prodotto prima su PC e poi su Product, reinserire infine il vincolo
- 2 più elegante: **posticipare** il controllo dei vincoli (ne parleremo)

# Posticipare Vincoli e Trigger

In situazioni come la precedente, è possibile **differire** il controllo di alcuni vincoli di integrità alla fine di una transazione (sequenza di istruzioni SQL eseguite atomicamente).

Ciascun vincolo può appartenere ad una fra tre categorie:

- **NOT DEFERRABLE**: viene sempre controllato dopo ogni operazione
- **DEFERRABLE INITIALLY IMMEDIATE**: viene controllato dopo ogni operazione della transazione, ma è possibile rilassarlo per farlo controllare solo prima del commit
- **DEFERRABLE INITIALLY DEFERRED**: viene controllato solo prima del commit, ma è possibile rafforzarlo per farlo controllare dopo ogni operazione della transazione

I vincoli differibili possono essere configurati tramite **SET CONSTRAINTS**. Solo i **CONSTRAINT TRIGGER** possono essere dichiarati differibili (link).

## Alcuni esercizi:

Preparate i vostri laptop e create lo schema dei prodotti di elettronica utilizzato in queste slide. Definite poi i seguenti trigger Postgres:

- 1 Quando il prezzo di un PC viene alzato, verificare che non vi sia già un laptop più veloce di prezzo inferiore o uguale.
- 2 Quando vengono inseriti dei nuovi laptop, controllare che il prezzo medio dei laptop per ciascun produttore sia almeno 1500.
- 3 Quando vengono aggiornati dei laptop, controllare che il prezzo medio dei laptop per ciascun produttore sia almeno 1500.
- 4 Quando viene inserita una nuova stampante a colori, essa deve costare di più di qualsiasi stampante in bianco e nero.

Riflettete bene sul tipo di trigger corretto da utilizzare! E' consigliato l'uso di un'interfaccia grafica come pgAdmin, phpPgAdmin o DBeaver.