

Trigger

Basi di dati attive

- **Trigger:** regole che specificano azioni attivate automaticamente dal DBMS al verificarsi di determinati eventi
- Fanno parte dello standard a partire da SQL-99
 - Prima, ogni DBMS li implementava seguendo una propria sintassi
- Negli esempi che vedremo faremo riferimento alla sintassi seguita in Oracle, molto vicina alla sintassi SQL-99

Basi di dati attive

- La specifica di una regola è generalmente effettuata attraverso il modello **ECA**:

—**E**vent-**C**ondition-**A**ction

Modello EventConditionAction

- L'**evento** può corrispondere
 - ad una operazione di aggiornamento del DB
 - ad un evento sistematico (ogni giorno alle 21.30)
 - ad un evento esterno (un utente che si collega al DBMS)
- Al verificarsi dell'evento viene attivata la regola

Modello EventConditionAction

- **Condizione**

- Definisce una condizione da verificarsi prima di procedere effettivamente all'esecuzione della regola
- La condizione è facoltativa:
 - Se non viene specificata, la regola viene eseguita ogni volta che si verifica l'evento
 - Se è specificata, la regola viene eseguita solo se la condizione è TRUE

Modello EventConditionAction

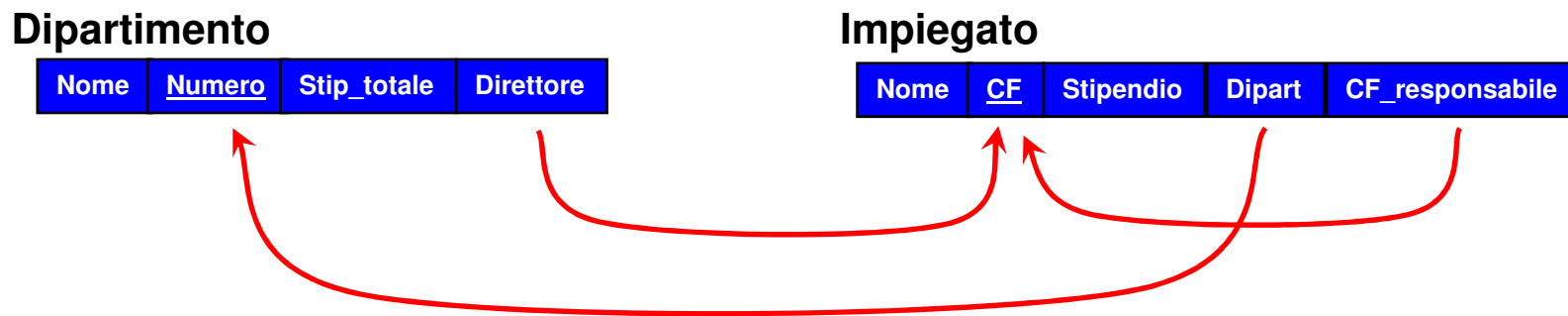
- **Azione**

- Di solito è una sequenza di comandi SQL (o PL/SQL), ma può essere anche una transazione sul DB o un programma esterno
- In PostgreSQL l'azione è specificata all'interno di una funzione (che viene chiamata dal trigger)

Trigger

- Per illustrare la definizione ed il funzionamento dei trigger consideriamo il seguente DB Impiegato-Dipartimento
 - CF_responsabile è una chiave esterna ricorsiva
 - Dipart può essere NULL per quegli impiegati non ancora affiliati ad alcun dipartimento
 - Stip_totale è un **attributo derivato** che deve riflettere la somma degli stipendi degli impiegati affiliati

Come calcolare il valore dell'attributo derivato ?



Trigger

- Per illustrare la definizione ed il funzionamento dei trigger consideriamo il seguente DB Impiegato-Dipartimento
 - CF_responsabile è una chiave esterna ricorsiva
 - Dipart può essere NULL per quegli impiegati non ancora affiliati ad alcun dipartimento
 - Stip_totale è un **attributo derivato** che deve riflettere la somma degli stipendi degli impiegati affiliati

UPDATE Dipartimento d SET Stip_totale = (SELECT sum(stipendio)
FROM impiegato WHERE dipart = d.numero);

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Questa istruzione di UPDATE è necessaria per inizializzare il valore di Stip_totale
- Tuttavia mantenere aggiornato il valore di Stip_totale attraverso tale istruzione può essere inefficiente
 - Per esempio, perché ricalcolare il valore degli stipendi di **tutti** i dipartimenti se a cambiare è stato lo stipendio di **un solo** impiegato?

UPDATE Dipartimento d SET Stip_totale = (SELECT sum(stipendio)
FROM impiegato WHERE dipart = d.numero);

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Questa istruzione di UPDATE è necessaria per inizializzare il valore di Stip_totale
- Tuttavia mantenere aggiornato il valore di Stip_totale attraverso tale istruzione può essere inefficiente
 - Per esempio, perché ricalcolare il valore degli stipendi di **tutti** i dipartimenti se a cambiare è stato lo stipendio di **un solo** impiegato?

Il valore corretto dell'attributo derivato può essere mantenuto attraverso un trigger

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Quali eventi ***possono*** causare un cambiamento dell'attributo derivato Stip_totale ?
 - Inserimento di (una o più) tuple di nuovi impiegati
 - Cambiamento di stipendio di (uno o più) impiegati esistenti
 - Cambiamento di affiliazione di impiegati da un dipartimento all'altro
 - Eliminazione di (una o più) tuple di impiegati

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Inserimento di (una o più) tuple di nuovi impiegati
 - Se i nuovi impiegati hanno campo Dipart=NULL non bisogna far nulla
 - Altrimenti bisogna aggiornare il valore Stip_totale per i dipartimenti interessati
 - In questo caso, esiste una condizione che, se verificata, inibisce l'esecuzione della regola

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Inserimento di (una o più) tuple di nuovi impiegati

```
CREATE TRIGGER STIPTOTALE1  
  AFTER INSERT ON Impiegato  
  FOR EACH ROW  
  WHEN (NEW.Dipart IS NOT NULL)  
    UPDATE Dipartimento  
    SET Stip_totale = Stip_totale + :NEW.Stipendio  
    WHERE Numero = :NEW.Dipart;
```

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Cambiamento di stipendio di (uno o più) impiegati esistenti
- Aggiornare il valore Stip_totale per i dipartimenti interessati

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Cambiamento di stipendio di (uno o più) impiegati esistenti

CREATE TRIGGER **STIPTOTALE2**

AFTER UPDATE OF **Stipendio** ON **Impiegato**

FOR EACH ROW

WHEN (NEW.**Dipart** IS NOT NULL)

UPDATE **Dipartimento**

SET **Stip_totale** = **Stip_totale** - :OLD.**Stipendio** + :NEW.**Stipendio**

WHERE **Numero** = :NEW.**Dipart**;

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Cambiamento di affiliazione di impiegati da un dipartimento all'altro
- Aggiornare il valore Stip_totale per i dipartimenti interessati

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Cambiamento di affiliazione di impiegati da un dipartimento all'altro

```
CREATE TRIGGER STIPTOTALE3
```

```
AFTER UPDATE OF Dipart ON Impiegato
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    UPDATE Dipartimento
```

```
    SET Stip_totale = Stip_totale + :OLD.Stipendio
```

```
    WHERE Numero = :NEW.Dipart;
```

```
    UPDATE Dipartimento
```

```
    SET Stip_totale = Stip_totale - :OLD.Stipendio
```

```
    WHERE Numero = :OLD.Dipart;
```

```
END
```

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Eliminazione di (una o più) tuple di impiegati
- Aggiornare il valore Stip_totale per i dipartimenti interessati

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Eliminazione di (una o più) tuple di impiegati

```
CREATE TRIGGER STIPTOTALE4  
AFTER DELETE ON Impiegato  
FOR EACH ROW  
    UPDATE Dipartimento  
    SET Stip_totale = Stip_totale - :OLD.Stipendio  
    WHERE Numero = :OLD.Dipart;
```

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- La clausola (facoltativa) FOR EACH ROW provoca l'applicazione della regola ad ogni riga, separatamente
 - Si parla di **trigger a livello di riga (row-level)**: Le variabili NEW ed OLD consentono di accedere selettivamente al valore di ogni singola riga cambiata
- Se la clausola viene omessa la regola viene applicata una volta per ogni istruzione
 - Si parla di **trigger a livello di istruzione (statement-level)**: L'uso delle variabili NEW ed OLD non è consentito

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



- Trigger a livello di statement

```
CREATE TRIGGER STIPTOTALE2  
  AFTER UPDATE OF Stipendio ON Impiegato  
    UPDATE Dipartimento d  
      SET Stip_totale = (SELECT sum(stipendio)  
                          FROM impiegato  
                          WHERE dipart=d.numero);
```

Trigger

- Abbiamo un'operazione di aggiornamento che aumenta del 10% lo stipendio degli impiegati del dipartimento 5

UPDATE **Impiegato**

SET **Stipendio** = **Stipendio***1.10

WHERE **Dipart**=5;

- Questa istruzione potrebbe eseguire l'aggiornamento simultaneo di tutti i valori di Stipendio delle tuple interessate...

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

```
UPDATE Impiegato  
SET Stipendio = Stipendio*1.10  
WHERE Dipart=5;
```

- ...tuttavia, attivando il trigger STIPTOTALE2 che è di tipo FOR EACH ROW, l'aggiornamento causerà l'attivazione del trigger per tante volte quanti sono gli impiegati del dipartimento 5

Dipartimento

Nome	<u>Numero</u>	Stip_totale	Direttore
------	---------------	-------------	-----------

Impiegato

Nome	<u>CF</u>	Stipendio	Dipart	CF_responsabile
------	-----------	-----------	--------	-----------------



Trigger

- Trigger a livello di istruzione
 - Velocità d'esecuzione
 - Non si possono usare i costrutti OLD e NEW
- Trigger a livello di riga
 - Riducono la velocità d'esecuzione dei comandi di aggiornamento (ma sono più veloci se l'aggiornamento riguarda poche tuple)
 - Attraverso i costrutti OLD e NEW si può fare riferimento ai valori di ciascuna singola tupla modificata

Trigger

- La differenza tra trigger **after** e **before** è che quest'ultimo è eseguito prima che i nuovi valori (variabile NEW) siano memorizzati nel database.
- Criterio guida:
 - Se una riga inserita o modificata (NEW) può essere cambiata all'interno dell'azione allora è necessario un **trigger before**
 - Se invece una riga inserita o modificata non può essere cambiata all'interno dell'azione allora è sufficiente un **trigger after**