

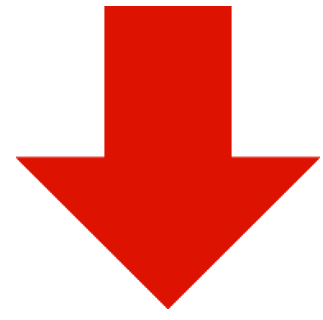
DATA WAREHOUSING: FRONT-END E OPERAZIONI OLAP

Vedremo



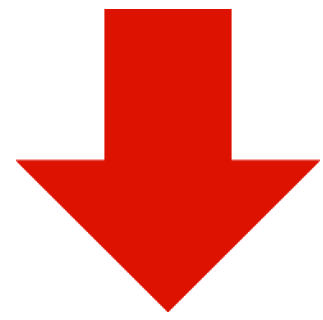
- **Gli strumenti per interagire con un Data Warehouse**
- **Le principali operazioni OLAP**
- **Le estensioni SQL per operazioni OLAP**

Strumenti per interagire con un Data Warehouse



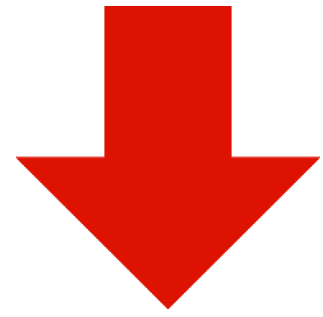
- Strumenti per la reportistica
- **Strumenti OLAP**
- Strumenti statistici
- Strumenti di Data Mining

Strumenti OLAP



- Utili quando l'uso di report predefiniti non è adeguato.
- Permettono di specificare operazioni OLAP arbitrarie che, in ambienti ROLAP, generano comandi SQL.
- Le operazioni vengono applicate in sequenza in una sessione di lavoro raffinando, passo dopo passo, il risultato ottenuto.
- I fatti vengono analizzati secondo punti di vista e livelli di dettaglio differenti.

Operazioni OLAP

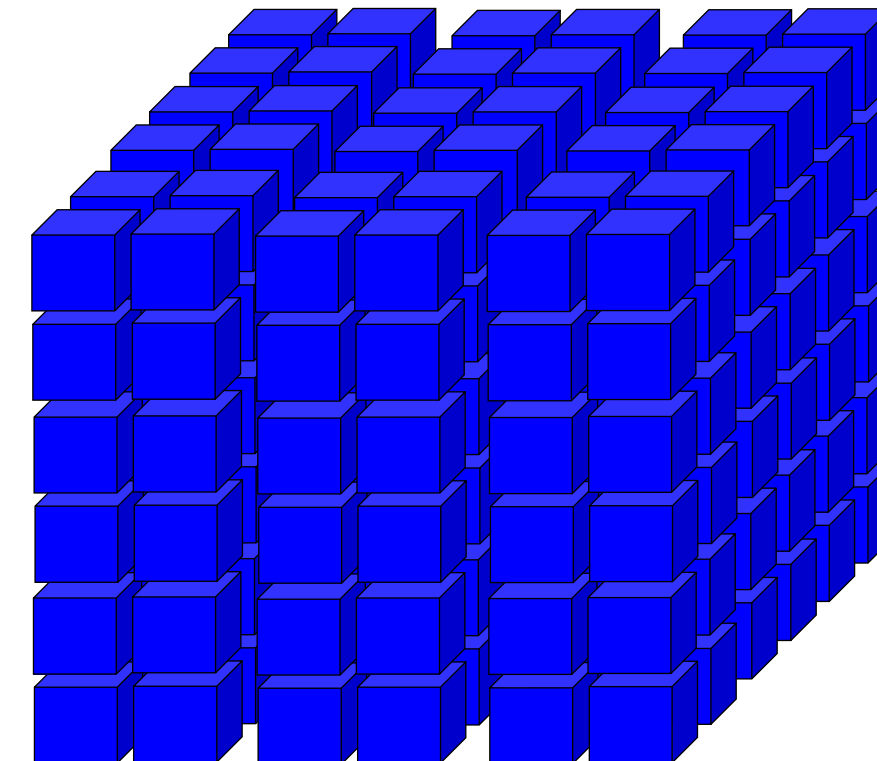


- **Calcolo di funzioni di aggregazione** rispetto a una o più dimensioni.
- **Operazioni di confronto** essenziali per comparare l'andamento dei fatti considerati.
- **Presentazione efficace** usando diverse modalità di rappresentazione degli stessi risultati.
- **Esplorazione profonda dei dati** secondo l'organizzazione gerarchica delle dimensioni.

Operazioni OLAP (segue)



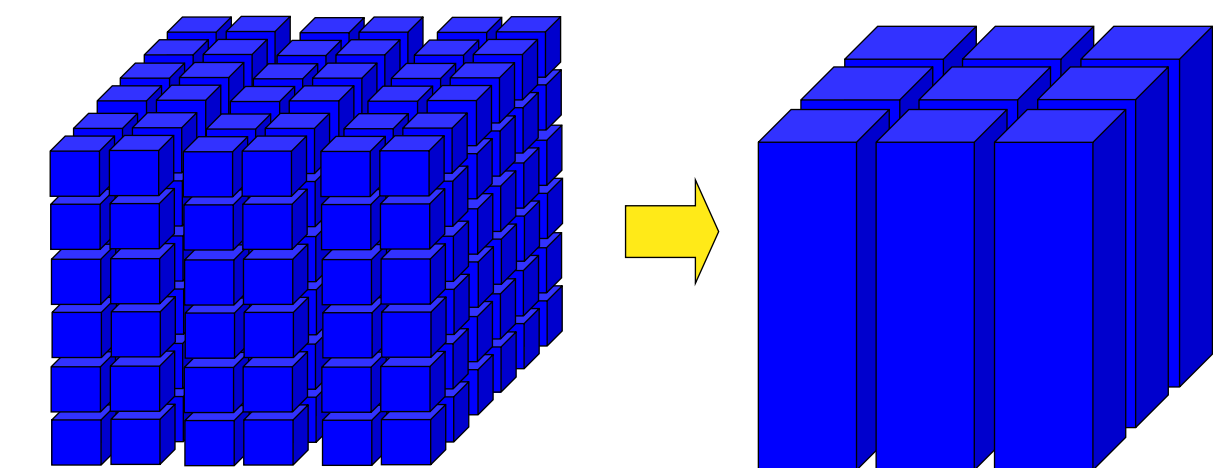
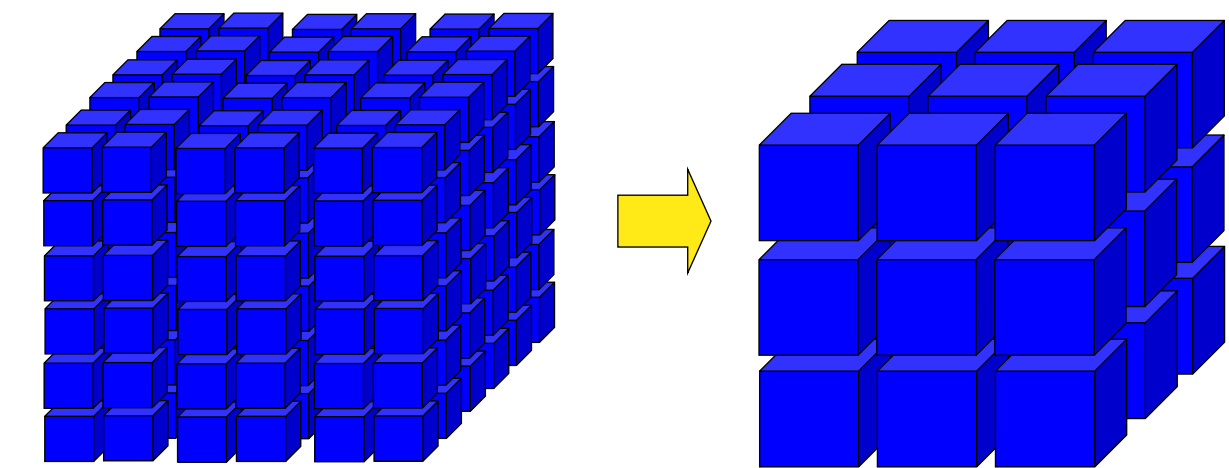
- Roll up
- Drill down
- Slice and dice
- Pivoting
- Sorting
- Drill across



Roll up



- Permette di ridurre il livello di aggregazione dei fatti mediante:
 - la riduzione del livello di dettaglio di una o più dimensioni, navigando le gerarchie
 - group by (store, month) → group by (city, month)
 - l'eliminazione di una dimensione
 - group by (product, city) → group by (product)
- Si riduce il numero di aggregati da calcolare e aumenta l'insieme dei fatti su cui calcolare ciascun aggregato.

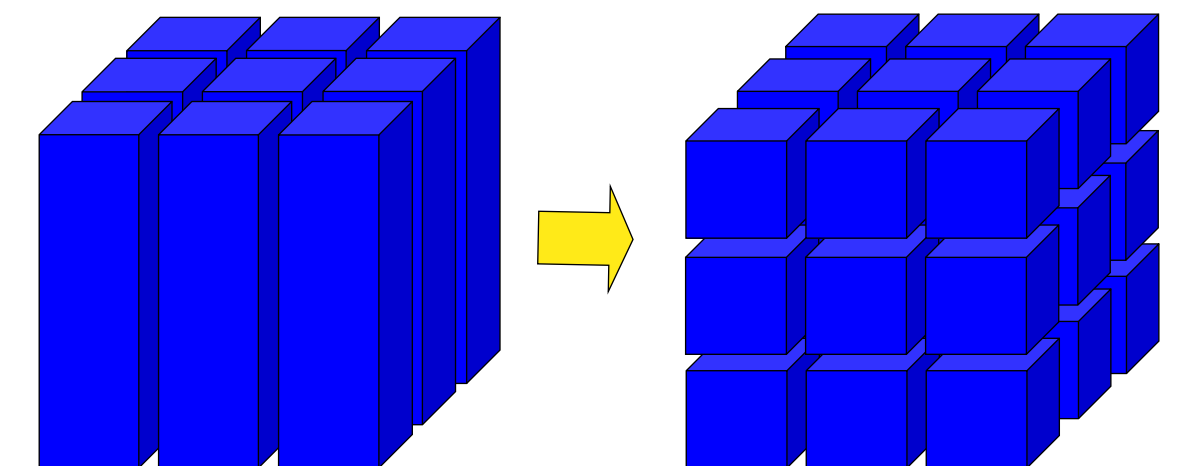
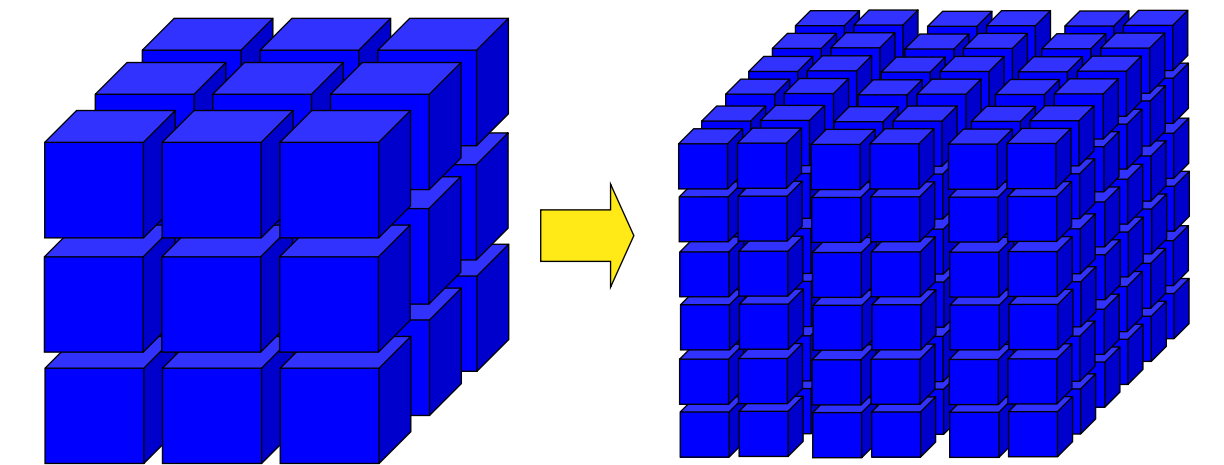


Drill down:



Permette di aumentare il livello di aggregazione dei fatti mediante:

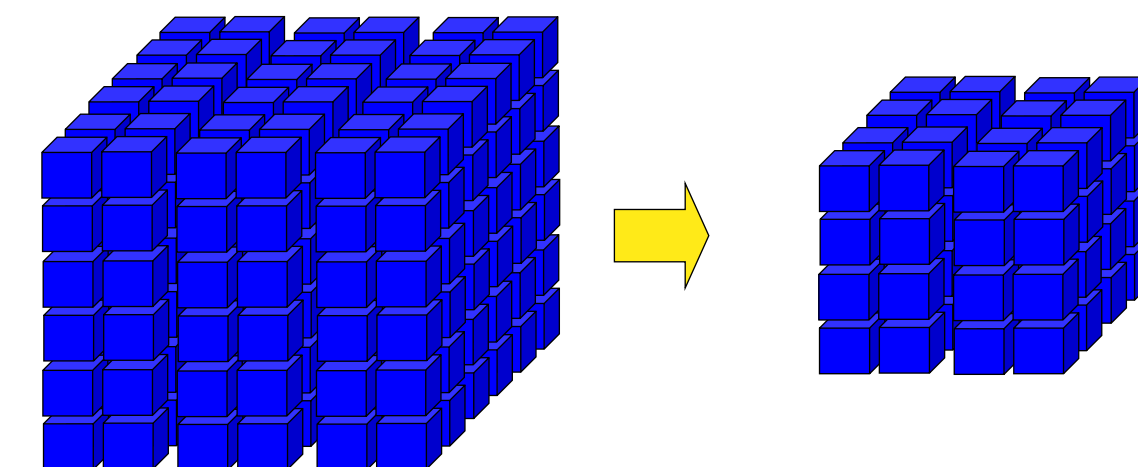
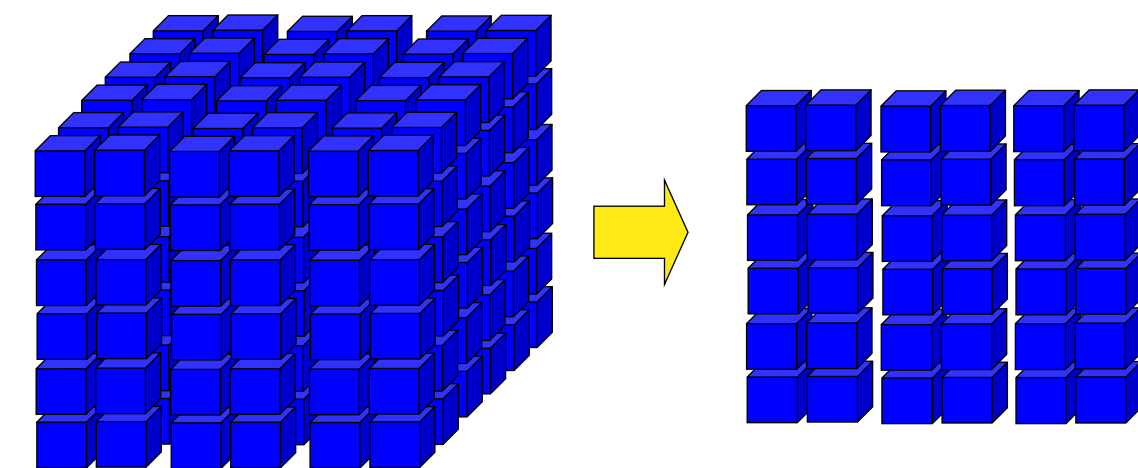
- l'incremento del livello di dettaglio di una o più dimensioni navigando le gerarchie
 - group by (city, month) → group by (store, month)
- l'aggiunta di una nuova dimensione
 - group by (city) → group by (city, product)
- Aumenta il numero di aggregati da calcolare e si riduce l'insieme dei fatti su cui calcolare ciascun aggregato.



Slide and dice



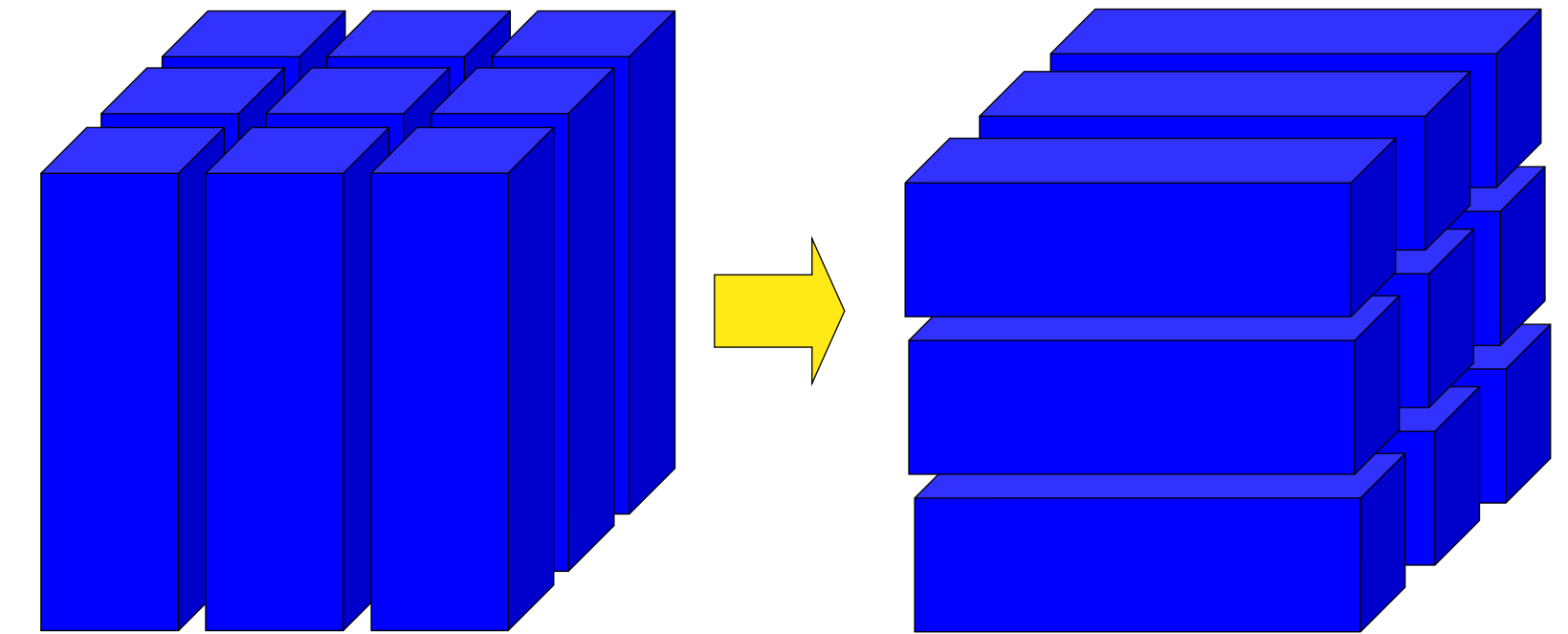
- Permette di ridurre il volume dei fatti da analizzare selezionandone un sottoinsieme.
- **slice**: singolo predicato
 - vendite del 2021
- **dice**: combinazione di predicati
 - vendite del 2021 a Genova.



Pivoting



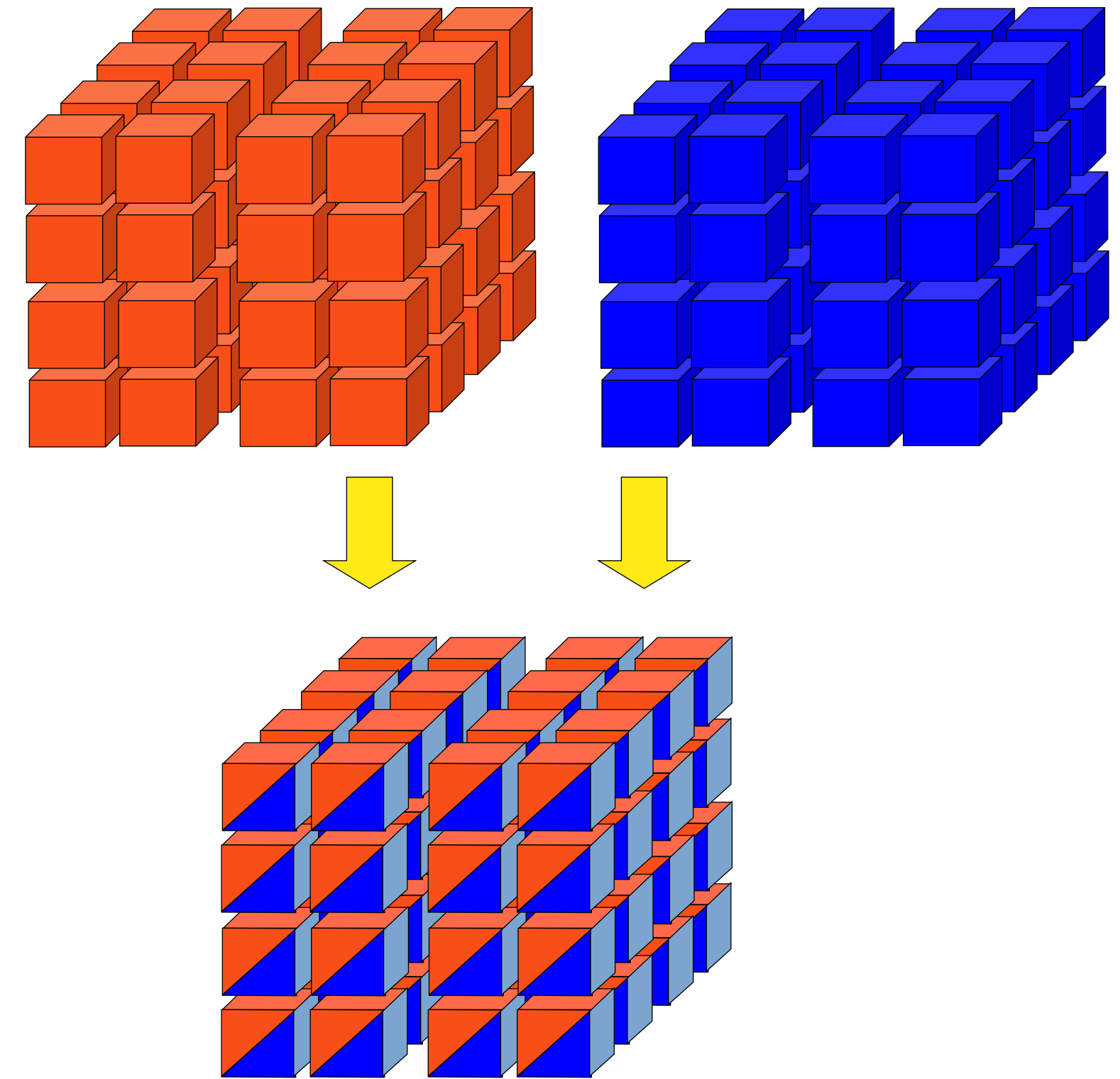
- Riorganizzazione/rotazione della struttura multidimensionale.
- L'insieme dei fatti e il livello di dettaglio a cui sono rappresentati non cambiano.
- Impatto solo sulla visualizzazione.



Drill across



- Permette di combinare il contenuto di due Data Mart.



Estensioni OLAP di SQL



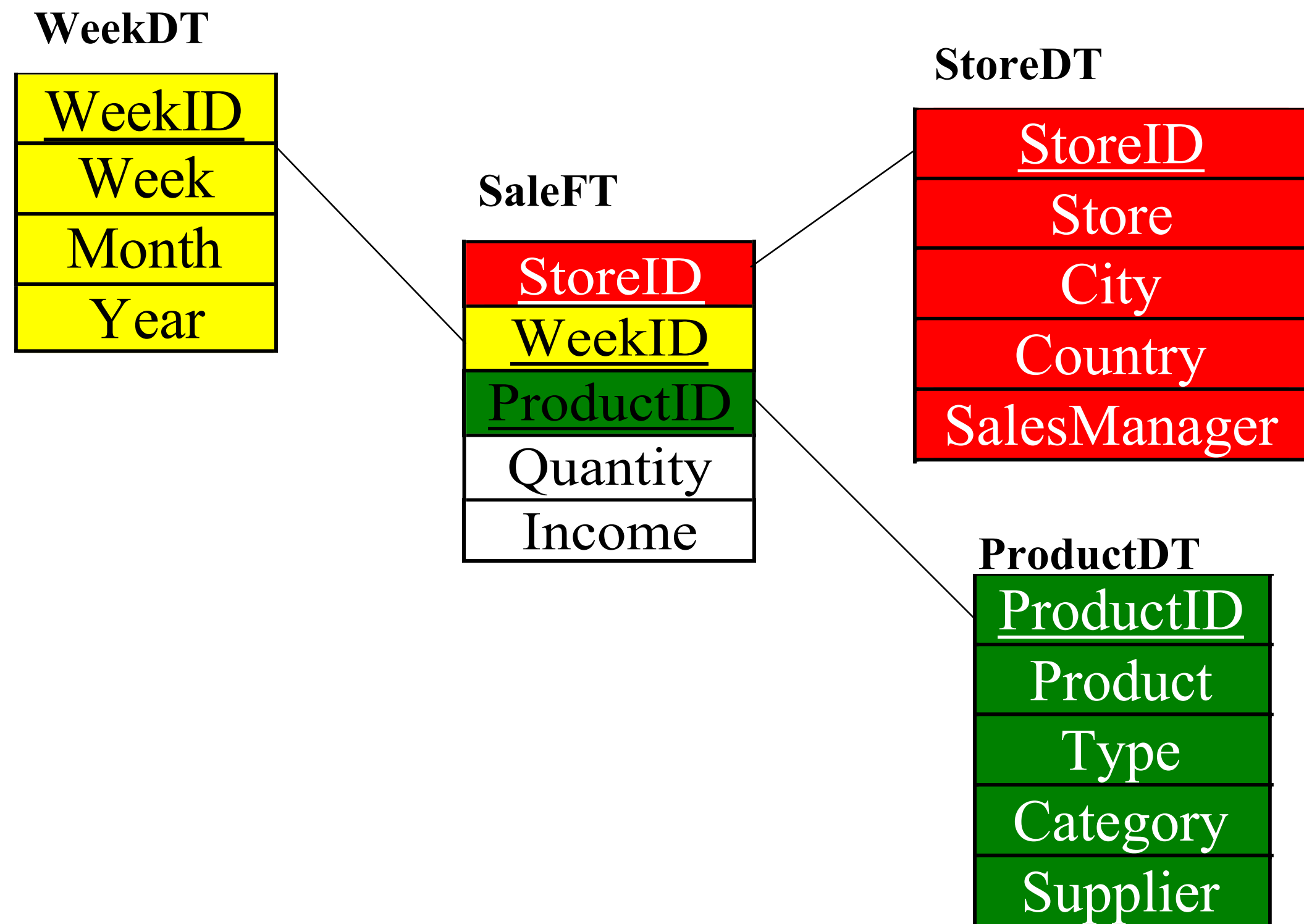
- Lo standard SQL è stato esteso con operatori OLAP:
 - **Nuovi operatori di raggruppamento**
 - **Nuovi operatori di aggregazione**

Nuovi operatori di raggruppamento



- **GROUP BY:** permette di definire gruppi di righe che condividono gli stessi valori per una lista di colonne.
- **Operatori OLAP di raggruppamento:**
 - Permettono di definire gruppi di righe rispetto a **più di una lista di colonne**
 - Corrispondono all'esecuzione simultanea di molteplici clausole GROUP BY
 - Sono implementati efficientemente: i risultati aggregati ottenuti per una certa lista di attributi vengono riutilizzati per calcolare aggregati più generali.
- Tre operatori:
 - **GROUP BY ROLLUP** per calcolare aggregati rispetto ai valori di insiemi specifici di colonne ottenute rimuovendo una colonna alla volta da un insieme specificato
 - **GROUP BY CUBE** per calcolare aggregati rispetto a tutte le combinazioni di un insieme di colonne specificato
 - **GROUP BY GROUPING SETS** per calcolare aggregati rispetto a una lista specificata di insiemi di colonne.

GROUP BY ROLLUP



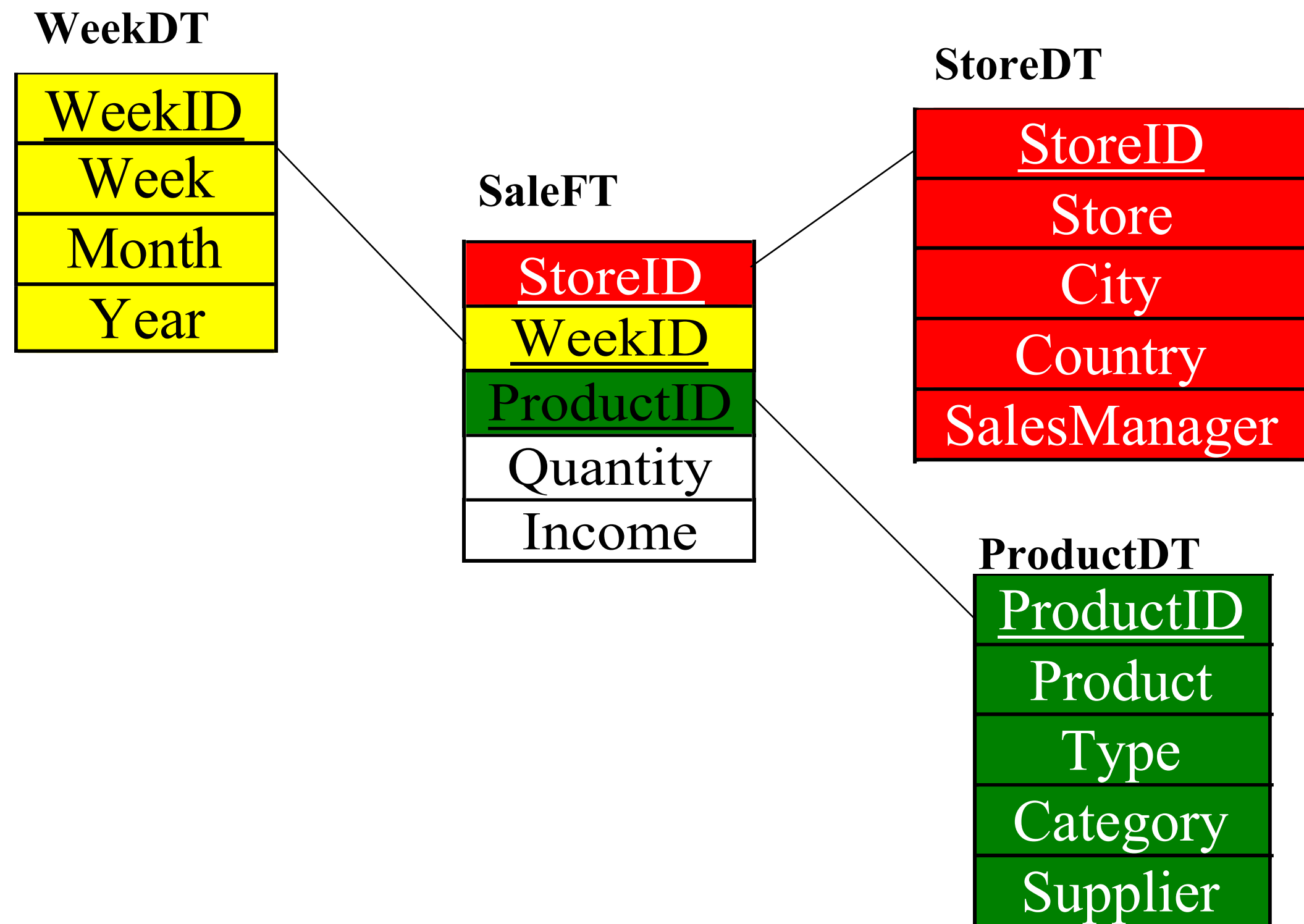
Calcolare il totale delle vendite nell'anno 2000, per la seguente combinazione di attributi:

- City, Month, ProductID
- City, Month
- City
- () [totale generale]

```
SELECT City, Month, ProductID,
       SUM(Income) AS TotIncome
FROM WeekDT NATURAL JOIN StoreDT NATURAL
     JOIN ProductDT NATURAL JOIN SaleFT
WHERE Year = 2000
GROUP BY ROLLUP (City, Month, ProductID)
```

City	Month	ProductID	TotIncome
Milano	7	145	110
Milano	7	150	10
Milano
Milano	7	NULL	8500
Milano	8
Milano	NULL	NULL	150000
Torino	150
Torino	...	NULL	2500
Torino	NULL	NULL	135000
...
NULL	NULL	NULL	25005000

GROUP BY CUBE

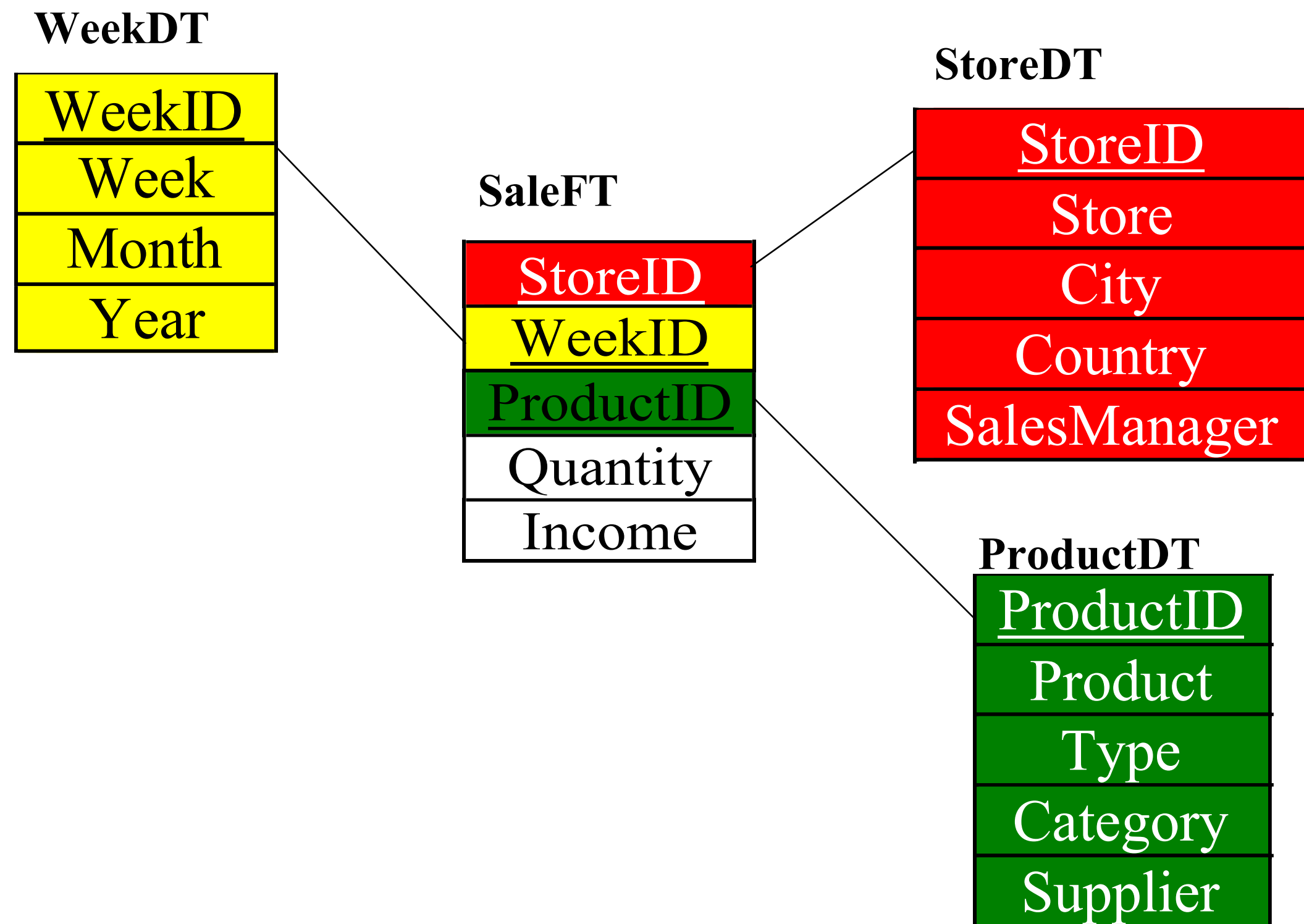


Calcolare il totale delle vendite nell'anno 2000, per tutte le combinazioni dei seguenti attributi:

- City, Month, ProductID

```
SELECT City, Month, ProductID,  
       SUM(Income) AS TotIncome  
FROM WeekDT NATURAL JOIN StoreDT NATURAL  
     JOIN ProductDT NATURAL JOIN SaleFT  
WHERE Year = 2000  
GROUP BY CUBE (City, Month, ProductID)
```


GROUP BY GROUPING SETS



Calcolare il totale delle vendite nell'anno 2000, per le seguenti combinazioni di attributi

- Month
- City, Month, ProductID

```
SELECT City, Month, ProductID,  
       SUM(Income) AS TotIncome  
FROM WeekDT NATURAL JOIN StoreDT NATURAL  
     JOIN ProductDT NATURAL JOIN SaleFT  
WHERE Year = 2000  
GROUP BY GROUPING SETS  
         (Month, (City, Month, ProductID))
```


Nuovi meccanismi di aggregazione



- Sono caratterizzati da:
 - **Finestre (WINDOW)** di calcolo: specificano, in modo flessibile, l'insieme delle righe sulle quali calcolare una certa funzione aggregata.
 - Basate su tre concetti principali: **partizionamento, ordinamento, framing.**
 - **Nuove funzioni di aggregazione.**

Partizionamento



- Divide le righe di una tabella in gruppi.
- Differenza con GROUP BY: le righe di ciascun gruppo non vengono collassate in un'unica riga del risultato.
- Su ogni partizione si può calcolare una funzione di aggregazione.
- Nell'output, il risultato dell'aggregazione su una partizione viene associato a ciascuna riga della partizione.

City	Month	Income
Genova	Feb-2022	1.200
Genova	March-2022	2.100
Genova	April-2022	1.200
Torino	Feb-2022	600
Torino	March-2022	1.500
Torino	April-2022	600

SUM

TotIncome
4.500
2.700

```
SELECT SUM(Income) AS TotIncome  
FROM SaleFT  
GROUP BY City
```

City	Month	Income
Genova	Feb-2022	1.200
Genova	March-2022	2.100
Genova	April-2022	1.200
Torino	Feb-2022	600
Torino	March-2022	1.500
Torino	April-2022	600

SUM

TotIncome
4.500
4.500
4.500
2.700
2.700
2.700

```
SELECT SUM(Income) OVER (PARTITION BY City) AS TotIncome  
FROM SaleFT
```

Ordinamento



- Le righe all'interno di una partizione possono essere ordinate.
- Utile per utilizzare particolari funzioni di aggregazione OLAP che richiedono l'ordinamento delle righe in input.
- Due insiemi di funzioni:
 - Funzioni che generano **un solo valore aggregato**, uguale per tutte le righe della partizione.
 - Funzioni che generano **valori aggregati diversi** per ciascuna riga della partizione.

Ordinamento (segue)



- Funzioni che:
 - generano **un solo valore aggregato** calcolato, uguale per tutte le righe della partizione
 - dipendono dall'ordinamento delle righe nelle partizioni
- `FIRST_VALUE()`, `LAST_VALUE()`, ...

City	Month	Income
Genova	Feb-2022	1.200
Genova	March-2022	2.100
Genova	April-2022	1.200
Torino	Feb-2022	600
Torino	March-2022	1.500
Torino	April-2022	600

IncFirstMonth
1.200
1.200
1.200
600
600
600

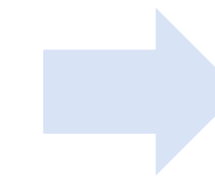
```
SELECT FIRST_VALUE(Income) OVER  
      (PARTITION BY City ORDER BY Month)  
      AS IncFirstMonth  
FROM SaleFT
```

Ordinamento (segue)



- Funzioni che:
 - generano **valori aggregati diversi** per ciascuna riga della partizione
 - dipendono dall'ordinamento delle righe nelle partizioni
- ROW_NUMBER(), RANK(), RANK_DENSE()

City	Month
Genova	Feb-2022
Torino	Feb-2022
Genova	March-2022
Torino	March-2022
Genova	April-2022
Torino	April-2022



RN	R	RD
1	1	1
2	1	1
3	3	2
4	3	2
5	5	3
6	5	3

```
SELECT ROW_NUMBER() OVER (ORDER BY Month) AS RN,  
       RANK() OVER (ORDER BY Month) AS R,  
       RANK_DENSE() OVER (ORDER BY Month) AS RD  
FROM Table
```

Framing

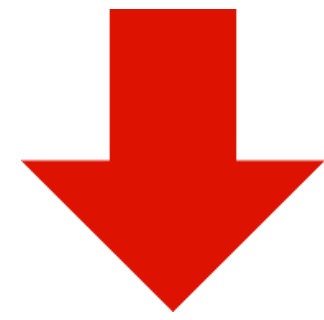


- Permette di associare a ciascuna riga di una partizione un valore aggregato calcolato su un sottoinsieme delle righe della partizione.
- **Frame** = finestra mobile (sliding window) all'interno di una partizione.
- Varie modalità per specificare i frame.
- Utile per il calcolo di aggregati mobili.

City	Month	Income		IncomeToDate
Genova	Feb-2022	1.200	}	1.200
Genova	March-2022	2.100		3.300
Genova	April-2022	1.200		4.500
Torino	Feb-2022	600		600
Torino	March-2022	1.500		2.100
Torino	April-2022	600		2.700

```
SELECT SUM(Income) OVER
(PARTITION BY City
ORDER BY Month
ROWS BETWEEN UNBOUNDED
PRECEDING AND CURRENT ROW)
AS IncomeToDate
FROM SaleFT
```


Clausola di definizione di window



- In SQL è anche possibile attribuire un nome alla definizione delle window
- Ordine di elaborazione delle clausole:
 - FROM, WHERE, GROUP BY, e HAVING
 - SELECT (WINDOW incluse)
 - ORDER BY

```
SELECT A1, ..., WFunction(expr)
        OVER ([WPartitioning ] [WOrdering [Wframe]])
FROM R1, ..., Rm
WHERE CondW
GROUP BY B1, ..., Bk
HAVING CondH
ORDER BY C1, ..., Cz
```

```
SELECT A1, ..., An, WFunction(expr) OVER WName
FROM R1, ..., Rm
WHERE CondW
GROUP BY B1, ..., Bk
HAVING CondH
WINDOW WName AS ([WPartitioning ] [WOrdering [Wframe]])
ORDER BY C1, ..., Cz
```

Riepilogo e conclusioni finali



Abbiamo visto:

- **Gli strumenti per interagire con un Data Warehouse**
- **Le principali operazioni OLAP**
- **Le estensioni SQL per operazioni OLAP**