OLAP Functions in SQL iSeries per un'Analisi Dati Avanzata: Un Focus sulle Grouping Operation

22 Febbraio 2024

Introduzione

Nell'ambito dell'analisi e del reporting dei dati negli ambienti IBM iSeries, SQL offre una potente suite di funzioni OLAP (Online Analytical Processing) progettate per migliorare le capacità di aggregazione, analisi e reporting dei dati.

Tra questi, le operazioni di raggruppamento (**GROUPING SETS, ROLLUP e CUBE**) si distinguono per la loro capacità di facilitare complesse attività di analisi dei dati e di reporting con efficienza e flessibilità. Queste funzioni sono fondamentali per soddisfare le molteplici esigenze delle aziende consentendo un'aggregazione complessa dei dati su più dimensioni, supportando così sofisticati requisiti di reporting.

- **GROUPING SETS**: questa funzione rappresenta una svolta per gli analisti di dati e gli sviluppatori di report, poiché consente di specificare più livelli di aggregazione all'interno di una singola query SQL. Consentendo la creazione di report che includono sia totali parziali che totali complessivi senza la necessità di creare più query, GROUPING SETS semplifica notevolmente il processo di reporting. Questa funzionalità è particolarmente utile per generare report complessi che richiedono una suddivisione dettagliata dei dati in varie dimensioni.

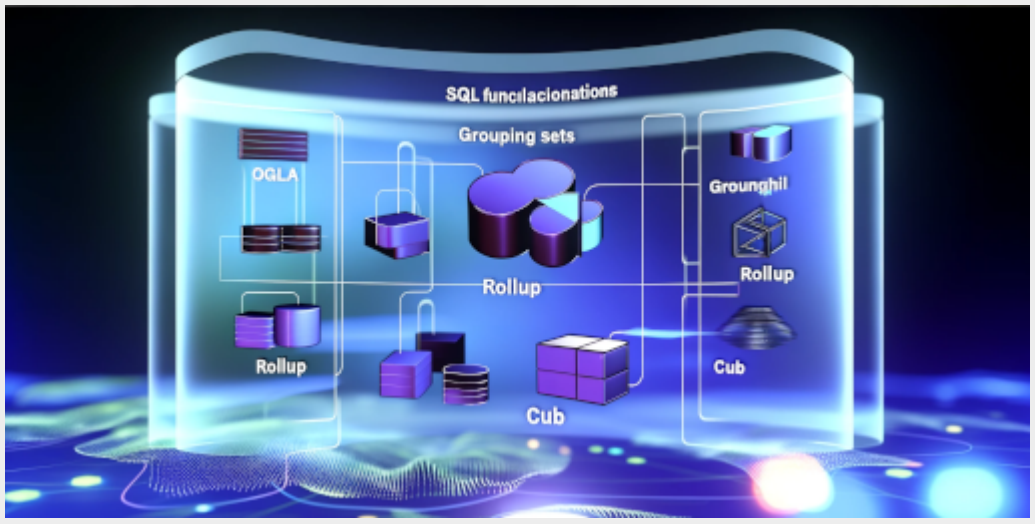
- **ROLLUP**: la funzione ROLLUP è progettata per produrre un set di risultati che non solo include l'aggregazione di base dei dati ma fornisce anche una struttura di aggregazione gerarchica. Ciò significa che gli utenti possono ottenere totali parziali a più livelli, culminando in un totale generale, il tutto all'interno di un unico set di risultati. ROLLUP è di grande aiuto per il reporting finanziario, l'analisi dell'inventario e qualsiasi scenario in cui la comprensione dell'aggregazione gerarchica dei dati è fondamentale.

- **CUBE** per coloro che cercano la massima flessibilità di aggregazione dei dati, la funzione CUBE offre la possibilità di generare tutte le possibili combinazioni di aggregazioni per un gruppo selezionato di colonne. Questa funzione è particolarmente potente per l'analisi multidimensionale, poiché consente agli utenti di esplorare i dati da ogni angolazione immaginabile. Che si tratti di dati di vendita, parametri finanziari o qualsiasi altro set di dati che richieda un'analisi completa, CUBE fornisce i mezzi per scoprire informazioni che altrimenti potrebbero rimanere nascoste.

Sfruttare queste funzioni OLAP all'interno di iSeries SQL non solo migliora le capacità analitiche delle organizzazioni, ma consente anche ai professionisti dei dati di fornire report più approfonditi, accurati e tempestivi. Sfruttando la potenza di **GROUPING SETS, ROLLUP e CUBE**, le aziende possono affrontare le complessità dei propri ambienti di dati con maggiore facilità e sicurezza, guidando un processo decisionale informato e una pianificazione strategica.

Note:

- Questi esempi utilizzano la clausola **WITH** per definire una CTE denominata **SalesData** a scopo dimostrativo. La clausola **VALUES** all'interno del CTE fornisce un modo semplice per creare dati di test direttamente all'interno della query.
- La clausola **ORDER BY** in ogni esempio garantisce che i risultati siano presentati in un ordine logico, facilitando l'interpretazione dei dati aggregati.
- Quando si eseguono questi esempi su un sistema iSeries, assicurarsi che l'ambiente SQL (ad esempio, IBM i Navigator per Windows, ACS Esegui script SQL o strumenti di terze parti) sia configurato correttamente per eseguire istruzioni SQL.
- L'output e l'utilità effettivi di queste query possono variare in base ai requisiti specifici delle attività di reporting e analisi. Questi esempi sono progettati per illustrare la sintassi e i potenziali casi d'uso per **GROUPING SETS, ROLLUP e CUBE** nel contesto di iSeries SQL.



Esempio 1: OLAP Function GROUPING SETS

Introduzione allo scopo e alle tecniche della query

Questa query SQL è progettata per aggregare i dati di vendita attraverso diverse gerarchie geografiche, fornendo un'analisi versatile delle prestazioni di vendita per macroarea, regione e totali complessivi. Utilizzando la clausola **WITH** per creare una Common Table Expression (CTE) denominata **SalesData**, la query organizza in modo efficiente i dati grezzi sulle vendite in un formato strutturato.

La clausola **VALUES** all'interno del CTE popola questa tabella temporanea con dati di vendita predefiniti, che abbracciano varie macroaree (ad esempio America, Asia, Europa), regioni all'interno di queste aree e importi di vendita corrispondenti per diverse nazioni.

Il potere analitico principale della query deriva dall'uso di **GROUPING SETS**.

Questa funzionalità SQL avanzata consente più livelli di aggregazione all'interno di una singola query, consentendo il calcolo delle vendite totali non solo per ciascuna macroarea e regione, ma anche per l'intero set di dati.

Specificando insiemi di raggruppamento solo per macroaree, macroaree con regioni e un insieme vuoto (per calcolare il totale complessivo), la query fornisce una panoramica completa della distribuzione e delle prestazioni delle vendite su diversi livelli geografici. La clausola **ORDER BY** garantisce che i risultati siano presentati in modo logico e organizzato, prima per macroarea e poi per regione, facilitando una facile interpretazione dei dati di vendita aggregati.

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
VALUES ('America', 'North America'   , 'Nation_NA1', 180000)
     , ('America', 'North America'   , 'Nation_NA2',  80000)
     , ('America', 'North America'   , 'Nation_NA3',  60000)
     , ('America', 'Central America' , 'Nation_CA1',  30000)
     , ('America', 'Central America' , 'Nation_CA2',  40000)
     , ('America', 'South America'   , 'Nation_SA1',  65000)
     , ('America', 'South America'   , 'Nation_SA2',  73000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA1',  90000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA2',  67000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA3',  55000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA1', 130000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA2', 101000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA3',  69000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA1',  89000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA2',  45000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE1',  44000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE2',  58000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE3',  43000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE1',  27000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE2',  32000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE3',  63000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE4',  19000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE1',  15000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE2',  48000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE3',  56000)
     ) --  select * from SalesData;
SELECT
  MacroArea,
  Region,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  GROUPING SETS ((MacroArea), (MacroArea, Region), ())
ORDER BY
  1, 2;
```

Risultato:

MACROAREA	REGION	TOTALSALES
America	Central America	70.000
America	North America	320.000
America	South America	138.000
America		528.000
Asia	Eastern Asia	300.000
Asia	Southtern Asia	134.000
Asia	Western Asia	212.000
Asia		646.000
Europe	Eastern Europe	119.000
Europe	Northern Europe	145.000
Europe	Western Europe	141.000
Europe		405.000
		1.579.000

Descrizione dettagliata della query:

- **Inizializzazione CTE (WITH SalesData)**: la query inizia definendo un'espressione di tabella comune denominata **SalesData** che funge da tabella temporanea. Questa tabella è strutturata con colonne per macroarea, regione, nazione e importo delle vendite associato a ciascuna nazione. La clausola **VALUES** viene quindi utilizzata per compilare manualmente questa tabella con i dati di vendita, illustrando un insieme diversificato di posizioni geografiche e i corrispondenti dati di vendita.

- **Aggregazione dei dati (istruzione SELECT)**: seguendo la definizione CTE, la query procede con l'aggregazione di questi dati di vendita. Seleziona le colonne **MacroArea** e **Region** a scopo di raggruppamento e calcola le vendite totali (**SUM(Amount)**) per ciascun insieme di raggruppamento definito.

- **Grouping Sets (GROUP BY Clause)**: la clausola **GROUP BY** è il punto in cui la query specifica i propri livelli di aggregazione univoci utilizzando **GROUPING SETS**. Definisce tre insiemi:

  - Il primo insieme raggruppa per sola **MacroArea**, fornendo i dati di vendita totali per ciascuna MacroArea.
  - Il secondo set raggruppa sia per **MacroArea che per Region**, offrendo una visione più dettagliata delle vendite all'interno di ciascuna macroarea.
  - Il terzo set, indicato da un **set vuoto ()**, aggrega le vendite totali nell'intero set di dati, ottenendo un totale complessivo.

- **Ordinamento dei risultati (clausola ORDER BY)**: infine, la clausola **ORDER BY** ordina i risultati aggregati prima per **MacroArea** e poi per **Region**. Ciò garantisce che l'output sia organizzato sistematicamente, rendendo più semplice per gli utenti la navigazione tra i dati di vendita. L'ordinamento facilita una visione gerarchica dei dati, dal livello più ampio di aggregazione (macro area) fino a regioni più specifiche all'interno di tali aree, e si conclude con il totale complessivo di tutte le vendite.

Esempio altra variante:

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
. . . . as above (come sopra)
. . . .
SELECT
  MacroArea,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  GROUPING SETS ((MacroArea), ())
ORDER BY
  1, 2;
```

Risultato:

MACROAREA	TOTALSALES
America	528.000
Asia	646.000
Europe	405.000
[NULL]	1.579.000

E ancora un altro Grouping Sets:

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
. . . . as above (come sopra)
. . . .
SELECT
  Region,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  GROUPING SETS ((Region), ())
ORDER BY
  1, 2;
```

Risultato:

REGION	TOTALSALES
Central America	70.000
Eastern Asia	300.000
Eastern Europe	119.000
North America	320.000
Northern Europe	145.000
South America	138.000
Southtern Asia	134.000
Western Asia	212.000
Western Europe	141.000
[NULL]	1.579.000

Questa query, con le varianati mostrate, esemplifica come è possibile sfruttare SQL per eseguire attività complesse di aggregazione dei dati, offrendo approfondimenti sulle prestazioni di vendita in varie dimensioni geografiche. L'uso di **GROUPING SETS** in particolare mette in mostra una tecnica avanzata per l'analisi dei dati multilivello all'interno di un'unica query efficiente.

Esempio 2: OLAP Function ROLLUP

Introduzione allo scopo e alle tecniche della query

Questa query SQL è realizzata per eseguire un'aggregazione gerarchica dei dati di vendita su diversi livelli geografici, dalle macroaree fino a regioni specifiche, culminando in un riepilogo completo che include totali parziali per ciascuna macroarea e un totale generale per tutte le vendite.

La query utilizza una Common Table Expression (CTE) denominata **SalesData** per simulare un set di dati all'interno della query stessa, utilizzando la clausola **VALUES** per popolare questa tabella temporanea con dati di vendita in vari segmenti geografici.

L'essenza di questa query risiede nell'uso della funzione **ROLLUP**, una potente operazione OLAP (Online Analytical Processing) in SQL che facilita la generazione di riepiloghi aggregati a più livelli di una gerarchia in un'unica esecuzione della query.

Questa funzione non solo aggrega i dati di vendita per ciascuna regione all'interno di una macroarea, ma calcola anche automaticamente i totali parziali per ciascuna macroarea e il totale generale per tutte le macroaree e regioni.

Strutturando la query in questo modo, sfrutta le capacità di aggregazione avanzate di SQL per fornire una suddivisione multilivello dei dati di vendita, fondamentale per l'analisi approfondita e il reporting in contesti aziendali. La clausola **ORDER BY** garantisce che l'output sia organizzato in modo ordinato, prima per macroarea e poi per regione, rendendo i dati aggregati facilmente interpretabili.

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
VALUES ('America', 'North America'   , 'Nation_NA1', 180000)
     , ('America', 'North America'   , 'Nation_NA2',  80000)
     , ('America', 'North America'   , 'Nation_NA3',  60000)
     , ('America', 'Central America' , 'Nation_CA1',  30000)
     , ('America', 'Central America' , 'Nation_CA2',  40000)
     , ('America', 'South America'   , 'Nation_SA1',  65000)
     , ('America', 'South America'   , 'Nation_SA2',  73000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA1',  90000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA2',  67000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA3',  55000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA1', 130000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA2', 101000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA3',  69000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA1',  89000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA2',  45000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE1',  44000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE2',  58000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE3',  43000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE1',  27000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE2',  32000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE3',  63000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE4',  19000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE1',  15000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE2',  48000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE3',  56000)
     )  -- select * from SalesData;
SELECT MacroArea, Region,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  ROLLUP (MacroArea, Region)
ORDER BY
  1, 2;
```

Risultato:

MACROAREA	REGION	TOTALSALES
America	Central America	70.000
America	North America	320.000
America	South America	138.000
America		528.000
Asia	Eastern Asia	300.000
Asia	Southtern Asia	134.000
Asia	Western Asia	212.000
Asia		646.000
Europe	Eastern Europe	119.000
Europe	Northern Europe	145.000
Europe	Western Europe	141.000
Europe		405.000
		1.579.000

Descrizione dettagliata della query

- **Definizione CTE (WITH SalesData)**: la query inizia con la definizione di un'espressione di tabella comune denominata **SalesData**, che funge da tabella virtuale per l'ambito di questa query. Questa CTE è strutturata per includere colonne per macroarea, regione, nazione e importo delle vendite associato a ciascuna nazione. La clausola **VALUES** che segue la dichiarazione CTE delinea meticolosamente i dati di vendita, classificandoli per gerarchie geografiche e corrispondenti dati di vendita, ponendo così le basi per la successiva aggregazione.

- **Aggregazione e rollup (istruzione SELECT):** il nucleo della query è l'istruzione SELECT, che mira ad aggregare i dati di vendita. Seleziona la **MacroArea** e la **Region** ai fini del raggruppamento e calcola le vendite totali (**SUM(Amount)**) per ciascun gruppo. L'uso di **ROLLUP** nella clausola **GROUP BY** è fondamentale in questo caso, poiché specifica i livelli gerarchici (MacroArea e Region) per i quali la query deve calcolare i subtotali e il totale generale.

- **Aggregazione gerarchica con ROLLUP**: la funzione **ROLLUP** genera un set di risultati che include non solo le vendite totali per ciascuna regione all'interno di una macroarea, ma aggiunge anche i totali parziali per ciascuna macroarea e un totale generale alla fine. Questa aggregazione gerarchica è essenziale per analizzare le prestazioni di vendita su diversi livelli geografici, fornendo approfondimenti sulle tendenze di vendita sia regionali che generali.

- **Ordinamento dei risultati (clausola ORDER BY):** per garantire che i dati aggregati siano presentati in modo ordinato e logico, la clausola **ORDER BY** ordina i risultati prima per **MacroArea** e poi per **Region**. Questo ordinamento è fondamentale per la leggibilità e la fruibilità dei dati, poiché si allinea alla struttura gerarchica dell'aggregazione, dalle macroaree fino alle regioni specifiche, fino alla sintesi complessiva.

Esempio altra variante:

Utilizzando un'altra impostazione dei valori di **ROLLUP** si ottengono risultati con nuovi raggruppamenti

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
. . . . as above (come sopra)
. . . .
SELECT MacroArea, Region, Nation,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  ROLLUP (MacroArea, Region, Nation)
ORDER BY
  1, 2, 3;
```

Risultato:

MACROAREA	REGION	NATION	TOTALSALES
America	Central America	Nation_CA1	30.000
America	Central America	Nation_CA2	40.000
America	Central America		70.000
America	North America	Nation_NA1	180.000
America	North America	Nation_NA2	80.000
America	North America	Nation_NA3	60.000
America	North America		320.000
America	South America	Nation_SA1	65.000
America	South America	Nation_SA2	73.000
America	South America		138.000
America	[NULL]		528.000
Asia	Eastern Asia	Nation_EA1	130.000
Asia	Eastern Asia	Nation_EA2	101.000
Asia	Eastern Asia	Nation_EA3	69.000
Asia	Eastern Asia		300.000

Asia	Southtern Asia	Nation_SA1	89.000
Asia	Southtern Asia	Nation_SA2	45.000
Asia	Southtern Asia		134.000
Asia	Western Asia	Nation_WA1	90.000
Asia	Western Asia	Nation_WA2	67.000
Asia	Western Asia	Nation_WA3	55.000
Asia	Western Asia		212.000
Asia	[NULL]		646.000
Europe	Eastern Europe	Nation_EE1	15.000
Europe	Eastern Europe	Nation_EE2	48.000
Europe	Eastern Europe	Nation_EE3	56.000
Europe	Eastern Europe		119.000
Europe	Northern Europe	Nation_NE1	44.000
Europe	Northern Europe	Nation_NE2	58.000
Europe	Northern Europe	Nation_NE3	43.000
Europe	Northern Europe		145.000
Europe	Western Europe	Nation_WE1	27.000
Europe	Western Europe	Nation_WE2	32.000
Europe	Western Europe	Nation_WE3	63.000
Europe	Western Europe	Nation_WE4	19.000
Europe	Western Europe		141.000
Europe			405.000
			1.579.000

In riepilogo queste query esemplificano l'uso strategico della funzione **ROLLUP** di OLAP SQL per condurre un'analisi articolata dei dati di vendita in varie dimensioni geografiche. Mostrano come aggregare e riepilogare in modo efficiente i dati a più livelli di dettaglio in un'unica query, fornendo informazioni preziose per il processo decisionale e il reporting strategico.

Esempio 3: OLAP Function CUBE

Introduzione allo scopo e alle tecniche della query

Questa query SQL è progettata per eseguire un livello avanzato di aggregazione e analisi dei dati sui dati di vendita in diversi segmenti geografici, utilizzando una Common Table Expression (CTE) e la funzione **CUBE** per l'analisi multidimensionale.

L'obiettivo principale è quello di fornire una panoramica completa delle prestazioni di vendita per macroarea e regione, offrendo allo stesso tempo la flessibilità di analizzare i dati da vari livelli di aggregazione, comprese le vendite totali per macroarea, per regione e i totali complessivi in tutte le aree.

L'utilizzo di una CTE denominata **SalesData** consente la creazione di un set di dati temporaneo all'interno della query, popolato con dati di vendita in diverse dimensioni geografiche (MacroArea, Regione, Nazione) e i corrispondenti importi di vendita. Questa configurazione facilita la manipolazione e l'analisi dei dati senza influenzare l'origine dati originale.

La funzione **CUBE** viene utilizzata nella clausola **GROUP BY** per generare totali parziali e totali generali su più dimensioni (MacroArea e Regione) in un unico passaggio. Questa funzione OLAP (Online Analytical Processing) espande la capacità della query di fornire una visione sfaccettata dei dati di vendita, consentendo alle parti interessate di ricavare informazioni da varie prospettive della gerarchia dei dati.

```
WITH SalesData (MacroArea, Region, Nation, Amount) AS (
VALUES ('America', 'North America'   , 'Nation_NA1', 180000)
     , ('America', 'North America'   , 'Nation_NA2',  80000)
     , ('America', 'North America'   , 'Nation_NA3',  60000)
     , ('America', 'Central America' , 'Nation_CA1',  30000)
     , ('America', 'Central America' , 'Nation_CA2',  40000)
     , ('America', 'South America'   , 'Nation_SA1',  65000)
     , ('America', 'South America'   , 'Nation_SA2',  73000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA1',  90000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA2',  67000)
     , ('Asia'   , 'Western Asia'    , 'Nation_WA3',  55000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA1', 130000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA2', 101000)
     , ('Asia'   , 'Eastern Asia'    , 'Nation_EA3',  69000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA1',  89000)
     , ('Asia'   , 'Southtern Asia'  , 'Nation_SA2',  45000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE1',  44000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE2',  58000)
     , ('Europe' , 'Northern Europe' , 'Nation_NE3',  43000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE1',  27000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE2',  32000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE3',  63000)
     , ('Europe' , 'Western Europe'  , 'Nation_WE4',  19000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE1',  15000)
     , ('Europe' , 'Eastern Europe'  , 'Nation_EE2',  48000)
```

```
      , ('Europe' , 'Eastern Europe'  , 'Nation_EE3',  56000)
      )  -- select * from SalesData;
SELECT
  MacroArea, Region,
  SUM(Amount) AS TotalSales
FROM
  SalesData
GROUP BY
  CUBE (MacroArea, Region)
ORDER BY
  1, 2;
```

Risultato:

MACROAREA	REGION	TOTALSALES
America	Central America	70.000
America	North America	320.000
America	South America	138.000
America		528.000
Asia	Eastern Asia	300.000
Asia	Southtern Asia	134.000
Asia	Western Asia	212.000
Asia		646.000
Europe	Eastern Europe	119.000
Europe	Northern Europe	145.000
Europe	Western Europe	141.000
Europe		405.000
[NULL]	Central America	70.000
[NULL]	Eastern Asia	300.000
[NULL]	Eastern Europe	119.000
[NULL]	North America	320.000
[NULL]	Northern Europe	145.000
[NULL]	South America	138.000
[NULL]	Southtern Asia	134.000
[NULL]	Western Asia	212.000
[NULL]	Western Europe	141.000

[NULL]		1.579.000

Descrizione dettagliata della query

- **Inizializzazione CTE (WITH SalesData)**: la query inizia definendo un'espressione di tabella comune **SalesData** che funge da tabella virtuale ai fini di questa query. Questo CTE è strutturato con colonne per MacroArea, Regione, Nazione e Importo ed è popolato con valori predefiniti che rappresentano i dati di vendita in diverse aree geografiche. Questo approccio consente la simulazione di un set di dati del mondo reale all'interno della query, facilitando la dimostrazione e il test delle capacità della funzione **CUBE**.

- **Aggregazione dei dati (istruzione SELECT)**: seguendo la definizione CTE, la query procede con l'aggregazione di questi dati di vendita. Seleziona la MacroArea e la Regione a scopo di raggruppamento e calcola le vendite totali (**SUM(Amount)**) per ogni possibile combinazione di queste dimensioni. Questo passaggio è fondamentale per analizzare le prestazioni di vendita su diversi livelli di granularità geografica.

- **Aggregazione multidimensionale con CUBE**: l'uso della funzione CUBE nella clausola **GROUP BY** è il fulcro di questa query. Consente l'aggregazione dei dati di vendita in tutte le combinazioni delle dimensioni specificate (MacroArea e Regione), includendo ciascuna dimensione in modo indipendente e tutte le dimensioni insieme. Ciò si traduce in un insieme di risultati che include le vendite totali per ciascuna regione all'interno di una macroarea, ciascuna macroarea in modo indipendente e un totale complessivo per tutte le macroaree e regioni combinate.

- **Ordinamento dei risultati (clausola ORDER BY)**: La clausola **ORDER BY** garantisce che i risultati aggregati siano presentati in modo strutturato, prima per MacroArea e poi per Regione. Questo ordinamento è essenziale per la presentazione logica dei dati, facilitando una facile interpretazione e analisi. Allinea l'output in un ordine gerarchico che rispecchia l'aggregazione multidimensionale eseguita dalla funzione **CUBE**, rendendo semplice la navigazione attraverso i vari livelli di aggregazione dei dati di vendita.

Questa query mette in mostra le potenti capacità analitiche della funzione CUBE di SQL, consentendo un'analisi approfondita dei dati di vendita su più dimensioni geografiche. Fornendo un approccio strutturato ma flessibile all'aggregazione dei dati, consente alle aziende di acquisire una comprensione completa delle proprie prestazioni di vendita, identificando tendenze e opportunità sia a livello macro che micro delle loro operazioni.

Quale, quando e perché – Proviamo a fare una comparativa

Le funzioni SQL OLAP (Online Analytical Processing), in particolare **GROUPING SETS, ROLLUP e CUBE**, sono potenti strumenti per l'analisi e il reporting dei dati, consentendo aggregazioni e riepiloghi complessi.

Sebbene queste funzioni a volte possano produrre risultati simili, ciascuna ha caratteristiche uniche e è adatta a scenari diversi.

Ecco una descrizione tecnica di ciascuna, evidenziandone le peculiarità e suggerendo contesti in cui uno potrebbe essere più vantaggiosa rispetto alle altre.

GROUPING SETS

- **Descrizione**: **GROUPING SETS** è una funzionalità che consente di specificare più raggruppamenti in un'unica query. È utile per creare report che richiedono diversi livelli di aggregazione, ad esempio totali parziali e totali generali, senza dover combinare più query.

- **Peculiarità**: fornisce un controllo esplicito sulle combinazioni di colonne che si desidera aggregare. A differenza di ROLLUP e CUBE, che generano automaticamente raggruppamenti gerarchici o combinatori, GRUPPO SETS richiede di specificare ogni insieme di raggruppamento in modo esplicito.

- **Casi d'uso**: ideali per scenari in cui sono necessarie combinazioni di aggregazione specifiche e non tutte le gerarchie o combinazioni possibili. Ad esempio, se hai bisogno di visualizzare le vendite totali per regione, per prodotto e quindi sia per regione che per prodotto, ma non sei interessato ad altre combinazioni.

ROLLUP

- **Descrizione: ROLLUP** viene utilizzata per generare un set di risultati che mostra aggregati per una gerarchia di valori, inclusi totali parziali a ciascun livello e un totale generale. Crea un raggruppamento per ogni livello di gerarchia specificato nella query.

- **Peculiarità**: Genera automaticamente una serie gerarchica di raggruppamenti in base all'ordine delle colonne specificato. La prima colonna è il livello più dettagliato e ogni colonna successiva aggiunge un livello alla gerarchia.

- **Casi d'uso**: più adatti per report in cui è necessaria l'aggregazione gerarchica, ad esempio report finanziari che richiedono totali parziali per ciascuna categoria e un totale generale alla fine. Semplifica la generazione di report con più livelli di totali su una o più dimensioni.

CUBE

- **Descrizione**: CUBE genera tutte le possibili combinazioni di aggregazioni per un insieme di colonne selezionate. È come ROLLUP ma più completo, poiché include non solo aggregazioni gerarchiche ma anche combinazioni di tabelle incrociate.

- **Peculiarità**: può produrre un set di risultati significativamente più ampio rispetto a ROLLUP perché calcola tutte le possibili combinazioni delle colonne di raggruppamento, compreso il totale complessivo.

- **Casi d'uso**: particolarmente utili per l'analisi multidimensionale, come negli scenari di data warehousing in cui potresti voler analizzare i dati su più dimensioni (ad esempio, tempo, geografia, prodotto). È ideale per creare report a campi incrociati o quando è necessario esplorare i dati senza una gerarchia predefinita.

Scegliere la funzione giusta

- **Aggregazioni specifiche e complete**: utilizzare **GROUPING SETS** quando sono necessarie combinazioni specifiche di aggregazioni. Scegli **ROLLUP** per riepiloghi gerarchici dei dati e **CUBE** per l'aggregazione più completa, comprese tutte le possibili combinazioni.

- **Considerazioni sulle prestazioni**: **CUBE** può generare un gran numero di raggruppamenti, il che potrebbe non essere efficiente in termini di prestazioni per set di dati di grandi dimensioni. **ROLLUP** e **GROUPING SETS** possono essere più efficienti in termini di prestazioni a seconda della complessità e delle dimensioni dei dati.

- **Profondità di analisi**: per analisi approfondite e multidimensionali, è preferibile **CUBE**. Per analisi più semplici e gerarchiche è adatto **ROLLUP**. **GROUPING SETS** offrono una via di mezzo, consentendo livelli di aggregazione personalizzati senza la piena esplosione combinatoria di **CUBE**.

In sintesi, anche se **GROUPING SETS**, **ROLLUP** e **CUBE** talvolta possono ottenere risultati simili, le loro differenze li rendono adatti a vari scenari. La scelta tra questi dovrebbe essere guidata dalle specifiche esigenze di reporting e analisi, considerando il livello di dettaglio desiderato, le implicazioni sulle prestazioni e la complessità delle relazioni tra i dati.

Conclusioni

In conclusione, l'applicazione strategica delle funzioni **SQL OLAP** di iSeries (**GROUPING SETS, ROLLUP e CUBE**) può migliorare significativamente le capacità di analisi dei dati e di reporting all'interno di un'organizzazione. Ciascuna funzione ha uno scopo unico, soddisfacendo diverse esigenze e complessità analitiche.

I **GROUPING SETS** offrono un controllo preciso sui livelli di aggregazione, rendendoli ideali per analisi mirate. **ROLLUP** semplifica il riepilogo gerarchico dei dati, fornendo un approccio semplificato alla generazione di report con più livelli di aggregazione. **CUBE**, d'altro canto, fornisce un'analisi multidimensionale completa, consentendo un'esplorazione esaustiva delle relazioni tra i dati.

Comprendere le sfumature e i casi d'uso appropriati di queste funzioni è fondamentale per i professionisti di database che mirano a sfruttare SQL per l'analisi e il reporting avanzati dei dati. Applicando con giudizio queste operazioni **OLAP**, le organizzazioni possono ottenere informazioni più approfondite, migliorare i processi decisionali e potenziare la business intelligence complessiva.

Poiché i dati continuano a crescere in volume e complessità, padroneggiare questi potenti strumenti SQL sarà indispensabile per navigare nel panorama della moderna analisi dei dati.

Riferimenti utili

- Data Science: opportunità e sfide da affrontare

- Funzioni OLAP in Data Science: un approccio moderno alla Business Intelligence

- Sfruttare il Potere della Data Science e SQL con OLAP Functions su IBM iSeries

- OLAP Functions in SQL iSeries per un'Analisi Dati Avanzata: Un Focus sulle Windows Functions

- OLAP Functions in SQL iSeries per un'Analisi Dati Avanzata: Un Focus sulle Ranking Functions

#Data Science #Business Intelligence #OLAP Online Analytical Processing #Grouping Sets #Rollup #Cube

← PRECEDENTE
OLAP Functions in SQL iSeries per un'Analisi Dati Avanzata: Un Focus sulle Ranking Functions

SEGUENTE →
OLAP Functions in SQL i Series per un'Analisi Dati Avanzata: Un Focus sulle Miscellaneous Functions

Articoli simili

CATEGORIE BLOG

Data Science
AI Artificial Intelligence
SkillUP
As400 Architettura
Gestire AS400 con SQL
Gestione approfondita Risorse Sistema
Spiccioli di tecnica
Windows

CERCA TESTO IN POST / TAGS

Cerca ...

SkillUp Via Marco Polo, 139 55049 Viareggio (LU)
Via Lucchese, 84 50019 Sesto Fiorentino (FI)
Tel: 055319212 - info@SkillUp.cloud

LinkedIn

Marco Del Corto
Founder/CEO Genesi srl
Visualizza profilo