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

8 Marzo 2024

Introduzione

Nell'ambito dell'analisi e della gestione dei dati, la capacità di aggregare, interpretare e presentare i dati in modo efficiente è fondamentale.

La piattaforma IBM iSeries (AS/400), nota per la sua robustezza e affidabilità nella gestione delle esigenze di elaborazione dei dati a livello aziendale, offre una suite di funzionalità SQL progettate per supportare attività analitiche complesse.

Tra queste funzionalità, le funzioni OLAP (Online Analytical Processing) si distinguono per la loro capacità di facilitare la manipolazione avanzata dei dati e l'estrazione di insight direttamente da query SQL. Questo post si concentra su due funzioni OLAP varie critiche, ma spesso sottoutilizzate, all'interno del toolkit SQL iSeries: **LISTAGG()** e **GROUPING().**

Funzione **LISTAGG()**: essenzialmente, **LISTAGG()** funge da potente strumento per l'aggregazione dei dati, in particolare in scenari che richiedono la concatenazione di dati di stringa su più righe in un'unica riga.

Questa funzione è preziosa per generare riepiloghi completi, report o qualsiasi formato di presentazione dei dati in cui è necessario consolidare informazioni testuali da vari record.

Consentendo separatori personalizzati tra valori concatenati, **LISTAGG()** offre flessibilità nella presentazione dei dati, rendendolo uno strumento essenziale per gli analisti di dati che desiderano migliorare le proprie capacità di reporting.

Funzione GROUPING(): la funzione **GROUPING()** svolge un ruolo fondamentale nel distinguere tra righe regolari e superaggregate all'interno di un set di risultati. Questa distinzione è fondamentale quando si lavora con dati raggruppati, poiché consente agli analisti di identificare se una riga specifica nel risultato di una query aggregata fa parte del raggruppamento di dati originale o è il risultato di un'ulteriore aggregazione.

L'uso di **GROUPING()** è particolarmente utile in scenari complessi di reporting e analisi dei dati in cui è necessaria la comprensione della gerarchia e della struttura dei dati aggregati per un'interpretazione e un processo decisionale accurati.

L'inclusione nella piattaforma SQL iSeries di funzioni OLAP come **LISTAGG()** e **GROUPING()** migliora significativamente le capacità di manipolazione dei dati, aprendo la strada a presentazioni di dati innovative e approfondimenti più approfonditi.

Poiché le aziende continuano a navigare in un mondo sempre più basato sui dati, sfruttare queste funzioni può fornire un vantaggio competitivo nell'analisi dei dati e nei processi decisionali. Questo post mira a demistificare queste funzioni e incoraggiare la loro adozione nelle pratiche di analisi dei dati, assicurando di sfruttare appieno le capacità di iSeries SQL per soddisfare le vostre esigenze di elaborazione dei dati e reporting.

Esempi di OLAP Function LISTAGG()

Introduzione allo scenario

Supponiamo di avere un dataset (SalesData) che rappresenta i risultati di vendita in varie macroaree e regioni, concentrandosi sui prodotti che hanno superato gli obiettivi di vendita. Questo set di dati funge da base per due query approfondite, ciascuna progettata per aggregare e presentare i dati in modo da evidenziare diversi aspetti dei risultati di vendita.

MACROAREA	REGION	PRODUCT
America	North America	P_011
America	North America	P_035
America	Central America	P_021
America	Central America	P_027
America	Central America	P_019
America	South America	P_027
Asia	Western Asia	P_005
Asia	Western Asia	P_027

Asia	Eastern Asia	P_038
Asia	Eastern Asia	P_016
Asia	Eastern Asia	P_040
Asia	Southtern Asia	P_044
Asia	Southtern Asia	P_021
Asia	Southtern Asia	P_038
Europe	Northern Europe	P_019
Europe	Northern Europe	P_035
Europe	Northern Europe	P_021
Europe	Western Europe	P_027
Europe	Western Europe	P_021
Europe	Western Europe	P_009
Europe	Eastern Europe	P_021
Europe	Eastern Europe	P_009

Questo set di dati funge da base per due query approfondite, ciascuna progettata per aggregare e presentare i dati in modo da evidenziare diversi aspetti dei risultati di vendita.

Query 1: elenchi di Products aggregati per Region

Questa query mira a fornire una panoramica concisa delle prestazioni dei prodotti per regione, offrendo alle aziende una prospettiva chiara su quali prodotti hanno superato le aspettative di vendita in aree geografiche specifiche. Eseguendo la query otteniamo un elenco organizzato per regione, in cui ciascuna voce presenta un elenco consolidato di prodotti che hanno ottenuto un notevole successo di vendite all'interno di quella regione. Questa aggregazione è preziosa per i responsabili delle vendite regionali e i team di marketing, poiché consente loro di identificare prodotti di successo e personalizzare le strategie di conseguenza.

Applicazioni pratiche:

- **Analisi delle prestazioni regionali**: capire quali prodotti stanno ottenendo buoni risultati in regioni specifiche per allocare le risorse di marketing in modo più efficace.
- **Ottimizzazione dell'inventario e della catena di fornitura**: adeguare i livelli di inventario e i piani di distribuzione in base alle tendenze delle prestazioni di vendita regionali.

```
WITH
  SalesData (MacroArea, Region, Product) AS (
      VALUES ('America', 'North America'   , 'P_011')
           , ('America', 'North America'   , 'P_035')
           , ('America', 'Central America' , 'P_021')
```

```
          , ('America', 'Central America' , 'P_027')
          , ('America', 'Central America' , 'P_019')
          , ('America', 'South America'   , 'P_027')
          , ('Asia'   , 'Western Asia'    , 'P_005')
          , ('Asia'   , 'Western Asia'    , 'P_027')
          , ('Asia'   , 'Eastern Asia'    , 'P_038')
          , ('Asia'   , 'Eastern Asia'    , 'P_016')
          , ('Asia'   , 'Eastern Asia'    , 'P_040')
          , ('Asia'   , 'Southtern Asia'  , 'P_044')
          , ('Asia'   , 'Southtern Asia'  , 'P_021')
          , ('Asia'   , 'Southtern Asia'  , 'P_038')
          , ('Europe' , 'Northern Europe' , 'P_019')
          , ('Europe' , 'Northern Europe' , 'P_035')
          , ('Europe' , 'Northern Europe' , 'P_021')
          , ('Europe' , 'Western Europe'  , 'P_027')
          , ('Europe' , 'Western Europe'  , 'P_021')
          , ('Europe' , 'Western Europe'  , 'P_009')
          , ('Europe' , 'Eastern Europe'  , 'P_021')
          , ('Europe' , 'Eastern Europe'  , 'P_009')
          )
, PL (Region, ProductsList) AS (
      SELECT Region,
             LISTAGG(Product, ', ')
               WITHIN GROUP (ORDER BY Product) AS ProductsList
        FROM SalesData
      GROUP BY Region
      ORDER BY Region
      )
select * from PL;
```

Risultato

REGION	PRODUCTSLIST
Central America	P_019, P_021, P_027
Eastern Asia	P_016, P_038, P_040
Eastern Europe	P_009, P_021
North America	P_011, P_035
Northern Europe	P_019, P_021, P_035
South America	P_027
Southtern Asia	P_021, P_038, P_044
Western Asia	P_005, P_027
Western Europe	P_009, P_021, P_027

Descrizione dettagliata della query

La query SQL fornita è una query ben strutturata che utilizza una Common Table Expression (CTE) per organizzare e aggregare i dati di vendita all'interno di un ambiente IBM iSeries. Questa particolare query è progettata per consolidare e visualizzare l'elenco dei prodotti che hanno superato gli obiettivi di vendita, raggruppati per regione. Di seguito una descrizione dettagliata e tecnica della query:

Common Table Expressions (CTEs)

SalesData CTE
Il CTE **SalesData** viene creato utilizzando una clausola VALUES per definire manualmente un set di righe che simulano una tabella con tre colonne: **MacroArea**, **Region** e **Product**. Ogni riga rappresenta un prodotto che ha superato gli obiettivi di vendita in una specifica regione all'interno di una macroarea.

PL (**ListaProdotti**) CTE
Dopo **SalesData** CTE, **PL** CTE viene utilizzato per creare un elenco di prodotti aggregati dalla colonna **Region**. Ciò si ottiene utilizzando la funzione **LISTAGG**, una funzione OLAP che concatena i valori di più righe in un'unica stringa con un delimitatore specificato, in questo caso una virgola (,).

Funzione LISTAGG
La funzione **LISTAGG** in questo contesto accetta due argomenti:

- **Prodotto**: la colonna contenente i valori da concatenare.
- **', '**: la stringa letterale che definisce il delimitatore per separare i valori concatenati.

La clausola **WITHIN GROUP** specifica l'ordine in cui i valori Product devono essere concatenati, ordinati in base alla colonna Product stessa.

Clausola **GROUP BY**
La clausola **GROUP BY** viene applicata alla colonna Region, che garantisce che la funzione LISTAGG concatena i valori Product per ciascuna regione univoca in un'unica stringa, creando così un elenco distinto di prodotti per ciascuna regione.

Clausola **ORDER BY**
Dopo il raggruppamento e l'aggregazione, alla colonna Regione viene applicata la clausola ORDER BY, che impone che l'output della query venga ordinato in base alla regione in ordine crescente.

Dichiarazione SELECT finale
Infine, il select * from PL; L'istruzione esegue PL CTE, risultando in un elenco di record recuperati dalla tabella virtuale PL, che ora contiene due colonne: **Region** e **ProductsList**. Ogni riga nel set di risultati rappresenta una regione univoca insieme a una stringa che elenca tutti i prodotti per quella regione, aggregati e ordinati secondo la specifica **LISTAGG**.

Questo output è particolarmente utile per una rapida consultazione e per scopi di reporting, poiché mostra chiaramente le prestazioni dei prodotti all'interno di specifici mercati regionali.

Query 2: elenchi di Region aggregati per Product

Al contrario, la seconda query sposta l'attenzione sui singoli prodotti, aggregando le regioni in cui ciascun prodotto ha superato gli obiettivi di vendita. Questa prospettiva è particolarmente utile per i product manager e i pianificatori strategici, poiché identifica l'ampiezza geografica dell'attrattiva e del successo del mercato di un prodotto.

Applicazioni pratiche:

• **Sviluppo e innovazione del prodotto**: le informazioni su dove i prodotti hanno successo possono guidare il miglioramento del prodotto e gli sforzi di sviluppo di nuovi prodotti.
• **Strategie di espansione del mercato**: l'identificazione di prodotti con un ampio appeal geografico può informare le strategie di penetrazione ed espansione del mercato.

```
WITH
  SalesData (MacroArea, Region, Product) AS (
. . . .
. . . . (as above)
. . . .
           )
, RL (Product, RegionsList) AS (
        SELECT Product,
             LISTAGG(Region, ', ')
               WITHIN GROUP (ORDER BY Region) AS RegionsList
        FROM SalesData
      GROUP BY Product
          ORDER BY Product
      )
select * from RL;
```

Risultato

PRODUCT	REGIONSLIST
P_005	Western Asia
P_009	Eastern Europe, Western Europe
P_011	North America
P_016	Eastern Asia
P_019	Central America, Northern Europe
P_021	Central America, Eastern Europe, Northern Europe, Southtern Asia, Western Europe
P_027	Central America, South America, Western Asia, Western Europe
P_035	North America, Northern Europe
P_038	Eastern Asia, Southtern Asia
P_040	Eastern Asia
P_044	Southtern Asia

Descrizione dettagliata della query

La query SQL è strutturata per utilizzare una Common Table Expression (CTE) per organizzare ed eseguire query sui dati di vendita all'interno di un ambiente IBM iSeries. Questa query mira specificamente ad aggregare i dati per mostrare in quali regioni ciascun prodotto ha avuto successo, in base al superamento degli obiettivi di vendita. Ecco una ripartizione dettagliata e tecnica della query:

Common Table Expressions (CTEs)

SalesData CTE
Il CTE SalesData funziona come un set di risultati temporaneo definito nell'ambito di esecuzione dell'istruzione SQL più ampia. Questo CTE non è mostrato per intero nell'immagine fornita ma si dice che sia come definito sopra, probabilmente contenente un insieme predefinito di righe con colonne: **MacroArea**, **Region** e **Product**. Ogni riga all'interno di questo CTE rappresenta un record di vendita in cui un particolare prodotto ha superato il suo obiettivo di vendita all'interno di una regione specifica.

RL (RegionsList) CTE
Viene quindi definito il **RL** CTE per trasformare SalesData aggregando le regioni associate a ciascun prodotto. La trasformazione viene eseguita dalla funzione **LISTAGG**.

Funzione LISTAGG
La funzione **LISTAGG** è una funzione OLAP utilizzata qui per concatenare i valori Region da più righe in un'unica stringa per ciascun prodotto. Richiede i seguenti parametri:

- **Region**: la colonna i cui valori devono essere concatenati.
- ', ': Il delimitatore utilizzato per separare i valori concatenati, che in questo caso è una virgola seguita da uno spazio.

La clausola **WITHIN GROUP** è inclusa per ordinare i valori **Region** prima della concatenazione, determinata dalla clausola **ORDER BY Region** al suo interno.

GROUP BY Clause
La clausola **GROUP BY Product** è fondamentale in quanto specifica che l'aggregazione eseguita da **LISTAGG** deve essere eseguita su righe con lo stesso valore Product. Di conseguenza, ogni prodotto unico avrà un elenco corrispondente di regioni in cui ha superato gli obiettivi di vendita.

ORDER BY Clause
La clausola **ORDER BY** Product garantisce che l'output finale di RL CTE sia ordinato in base alla colonna Product in ordine crescente.

SELECT finale
L'istruzione finale select * from RL; esegue la query utilizzando RL CTE e il risultato è un elenco di prodotti, ciascuno accompagnato da una stringa che consolida tutte le regioni in cui quel prodotto ha superato gli obiettivi di vendita. Le regioni nella stringa sono ordinate come specificato dalla clausola **WITHIN GROUP**.

Questo formato di output è particolarmente vantaggioso per le parti interessate che necessitano di una rappresentazione compatta della penetrazione del mercato e del successo di ciascun prodotto nelle diverse regioni. Aiuta nel processo decisionale strategico relativo al marketing, alla distribuzione dei prodotti e all'allocazione delle risorse.

Aggregazione gerarchica dei dati di vendita con funzioni OLAP in iSeries SQL

Introduzione alla Query

In questo esempio esploreremo l'uso avanzato delle funzioni OLAP in SQL, in particolare su un sistema iSeries, per analizzare e aggregare i dati di vendita su diversi livelli geografici. Il nostro obiettivo è dimostrare come riassumere in modo efficace gli importi delle vendite fornendo al contempo informazioni dettagliate sulle prestazioni di vendita a livello regionale e nazionale.

Utilizzando i **GROUPING SETS** insieme alla funzione **GROUPING**, miriamo a migliorare la presentazione dei dati, consentendo una visione gerarchica che rivela dati di vendita sia dettagliati che consolidati.

Questo approccio è particolarmente utile per l'analisi multidimensionale, poiché offre una chiara comprensione della distribuzione delle vendite attraverso diverse granularità all'interno della gerarchia organizzativa.

```
WITH SalesData (MacroArea, Region, Country, Amount) AS (
VALUES ('America', 'North America'   , 'Country_NA1', 180000)
     , ('America', 'North America'   , 'Country_NA2',  80000)
     , ('America', 'North America'   , 'Country_NA3',  60000)
     , ('America', 'Central America' , 'Country_CA1',  30000)
     , ('America', 'Central America' , 'Country_CA2',  40000)
     , ('America', 'South America'   , 'Country_SA1',  65000)
     , ('America', 'South America'   , 'Country_SA2',  73000)
     , ('Europe' , 'Northern Europe' , 'Country_NE1',  44000)
     , ('Europe' , 'Northern Europe' , 'Country_NE2',  58000)
     , ('Europe' , 'Northern Europe' , 'Country_NE3',  43000)
     , ('Europe' , 'Western Europe'  , 'Country_WE1',  27000)
     , ('Europe' , 'Western Europe'  , 'Country_WE2',  32000)
     , ('Europe' , 'Western Europe'  , 'Country_WE3',  63000)
     , ('Europe' , 'Western Europe'  , 'Country_WE4',  19000)
     , ('Europe' , 'Eastern Europe'  , 'Country_EE1',  15000)
     , ('Europe' , 'Eastern Europe'  , 'Country_EE2',  48000)
     , ('Europe' , 'Eastern Europe'  , 'Country_EE3',  56000)
     ) --  select * from SalesData;
SELECT
  Region,
  Country,
  SUM(Amount) AS TotalSales,
  GROUPING(Region) AS RegionGrouping,
  GROUPING(Country) AS CountryGrouping
FROM SalesData
GROUP BY GROUPING SETS ((Region, Country), (Region), ())
ORDER BY GROUPING(Region), GROUPING(Country), Region, Country;
```

Risultato

REGION	COUNTRY	TOTALSALES	REGIONGROUPING	COUNTRYGROUPING
Central America	Country_CA1	30.000	0	0
Central America	Country_CA2	40.000	0	0
Eastern Europe	Country_EE1	15.000	0	0
Eastern Europe	Country_EE2	48.000	0	0
Eastern Europe	Country_EE3	56.000	0	0
North America	Country_NA1	180.000	0	0
North America	Country_NA2	80.000	0	0
North America	Country_NA3	60.000	0	0
Northern Europe	Country_NE1	44.000	0	0
Northern Europe	Country_NE2	58.000	0	0
Northern Europe	Country_NE3	43.000	0	0
South America	Country_SA1	65.000	0	0

South America	Country_SA2	73.000	0	0
Western Europe	Country_WE1	27.000	0	0
Western Europe	Country_WE2	32.000	0	0
Western Europe	Country_WE3	63.000	0	0
Western Europe	Country_WE4	19.000	0	0
Central America	[NULL]	70.000	0	1
Eastern Europe	[NULL]	119.000	0	1
North America	[NULL]	320.000	0	1
Northern Europe	[NULL]	145.000	0	1
South America	[NULL]	138.000	0	1
Western Europe	[NULL]	141.000	0	1
[NULL]	[NULL]	933.000	1	1

Descrizione tecnica dettagliata della query

L'istruzione SQL fornita è una query sofisticata che utilizza il concetto di espressioni di tabella comuni (CTE) e le potenti funzioni di raggruppamento disponibili in SQL per eseguire aggregazioni e analisi avanzate dei dati.

Common Table Expression: SalesData
Il CTE **SalesData** è costruito per simulare un set di dati di vendita che include le colonne **MacroArea**, **Region**, **Country** e **Amount**. Questo set di dati è popolato con valori codificati che rappresentano i dati di vendita per vari paesi all'interno di diverse regioni e macroaree.

Dichiarazione SELECT finale
L'istruzione **SELECT** finale è progettata per estrarre e aggregare questi dati per fornire i dati sulle vendite totali, utilizzando al tempo stesso le funzionalità OLAP per distinguere tra diversi livelli di riepilogo dei dati.

- **Region, Country**: le colonne selezionate che verranno visualizzate nel set di risultati finali.
- **SUM(Amount) AS TotalSales**: calcola le vendite totali per ciascun insieme di raggruppamento definito. Riassume la colonna Importo e le dà l'alias TotalSales.
- **GROUPING(Region) AS RegionGrouping**, **GROUPING(Country) AS CountryGrouping**: queste colonne utilizzano la funzione **GROUPING** per indicare il livello di aggregazione per ogni riga. La funzione **GROUPING** restituisce 1 se la riga è un risultato aggregato o superaggregato (come un totale parziale o un totale) e 0 in caso contrario. I risultati hanno rispettivamente l'alias **RegionGrouping** e **CountryGrouping**.

-

-

GROUP BY GROUPING SETS Clause
La clausola **GROUP BY GROUPING SETS** è il fulcro della capacità di aggregazione multilivello di questa query.

Specifica più livelli di raggruppamento in una query, che in questo caso sono:

- **(Region, Country)**: il livello di aggregazione più dettagliato, che fornisce dati di vendita per ciascun paese all'interno di ciascuna regione.
- (**Region**): un livello di aggregazione più elevato, che fornisce i totali parziali delle vendite per ciascuna regione.
- **()**: il livello più alto di aggregazione, che fornisce il totale complessivo delle vendite in tutte le regioni e paesi.

-

Clausola **ORDER BY**
La clausola **ORDER BY** specifica l'ordine dei file

set di risultati ed è particolarmente importante quando si utilizza il **GROUPING SETS**. Garantisce che i risultati aggregati vengano visualizzati in un ordine logico e gerarchico. Questa clausola ordina i risultati principalmente in base alle funzioni di **GROUPING** applicate a **Region** e **Country**, garantendo che gli aggregati totali appaiano per ultimi nel set di risultati. Quindi ordina per **Region** e **Country** per mantenere una sequenza alfabetica o logica all'interno dei dati raggruppati.

La funzione **GROUPING** all'interno della clausola **ORDER BY** garantisce che i risultati siano ordinati in modo da rispettare la gerarchia dell'aggregazione:

- Innanzitutto, le righe dettagliate in cui sono raggruppati sia **Region** e **Country** (indicate da **GROUPING(Region)** e **GROUPING(Country)** che restituiscono entrambi 0.
- Successivamente, le righe del totale parziale in cui è raggruppata solo la Region (indicata da **GROUPING(Region)** che restituisce 0 e **GROUPING(Country)** che restituisce 1.
- Infine, la riga del totale complessivo in cui sia Regione che Paese sono super-aggregati (indicata da **GROUPING(Region)** e **GROUPING(Country)** che restituiscono entrambi 1.

Questo ordinamento garantisce che tutti gli utenti dei risultati della query possano facilmente discernere il livello di dettaglio che stanno osservando: cifre dettagliate, totali parziali o totali generali.

La query dimostra in modo efficace come utilizzare le funzionalità SQL avanzate per analizzare e presentare i dati in un formato gerarchico comune nei report finanziari e sulle vendite. Fornisce un chiaro esempio di come riepilogare i dati a più livelli di dettaglio in un'unica query, che può essere prezioso per la business intelligence, il reporting e i processi decisionali.

Conclusioni

In sintesi, l'implementazione strategica delle funzioni OLAP come **LISTAGG**() e **GROUPING**() all'interno di un ambiente SQL iSeries offre una soluzione solida per l'analisi e la presentazione gerarchica dei dati. L'uso di queste funzioni nella nostra query facilita un'aggregazione a più livelli dei dati di vendita, consentendo una visione completa che spazia dai dettagli granulari ai riepiloghi di alto livello.

Questa metodologia non solo semplifica il processo di analisi dei dati, ma migliora anche il processo decisionale fornendo informazioni chiare sulle prestazioni di vendita nei diversi segmenti geografici. Sfruttando la potenza di queste funzioni OLAP, le organizzazioni possono trasformare i dati grezzi in informazioni fruibili, rafforzando il ruolo fondamentale della sofisticata gestione dei dati nel guidare la strategia e i risultati aziendali.

Tenere presente, come sempre, che l'implementazione e le funzionalità effettive potrebbero variare a seconda della versione del sistema operativo iSeries e del database DB2. È sempre buona norma fare riferimento alla documentazione ufficiale IBM per avere informazioni più precise e dettagliate.

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

#OLAP Online Analytical Processing #LISTAGG #GROUPING #IPower #Data Science #Business Intelligence

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

SEGUENTE →
Skillup i Power Days – 6-7 giugno 2024 Milano

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

Cookie policy Privacy policy

Powered by NDESIGN

Consenso ai Cookies