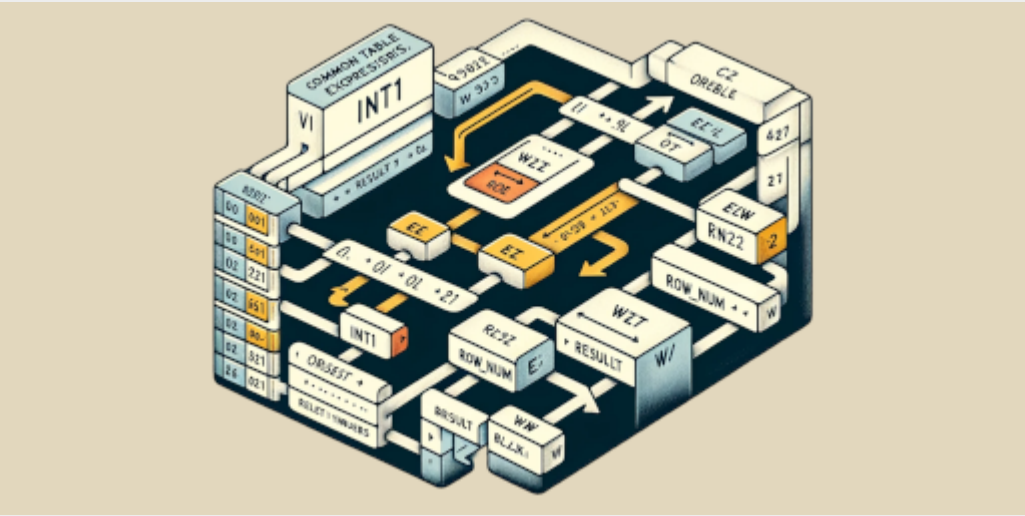
Recursive CTE come concatenare i risultati su un'unica riga

22 Gennaio 2024



Di seguito gli articoli della serie CTE (incluso questo corrente):

- iSeries Sql – CTE Common Table Expressions – definizioni e utilizzo
- CTE Ricorsive: L'Eleganza della Semplicità in SQL – 1
- CTE Ricorsive: L'Eleganza della Semplicità in SQL – 2
- CTE Ricorsive: L'Eleganza della Semplicità in SQL – 3
- Recursive CTE come concatenare i risultati su un'unica riga

Definizione del contesto

Supponiamo di avere una tabella che riporta i risultati delle elaborazioni notturne di interfacce, come da esempio seguente:

INT	RESULT
0011	
0012	
0013	
0014	W
0015	
0016	W

0017	
0018	
0019	
0020	E
0023	E
0024	W

Il risultato di una elaborazione può essere senza segnalazioni (spazio), con warning ("W") o con errori ("E").

supponiamo di voler ottenere su una sola riga la sintesi dei risultati, riportando all'inizio le elaborazioni errate, poi quelle con warning e infine quelle senza segnalazioni, come nell'esempio seguente:

ROW_NUM	INT_LIST
12	0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015 , 0017 , 0018 , 0019

tutto questo richiede un programma o può essere fatto elegantemente con un comando sql?

Query

Si, con il seguente comando Sql si ottiene esattamente il risultato atteso:

```
with -- Interface result
     W1 (INT, RESULT) as (
        values ('0011', ' '), ('0012', ' '), ('0013', ' '), ('0014', 'W'),
               ('0015', ' '), ('0016', 'W'), ('0017', ' '), ('0018', ' '),
               ('0019', ' '), ('0020', 'E'), ('0023', 'E'), ('0024', 'W')
        ) -- select * from W1 order by 1;
     -- assign progressive numeration (first E errors, then W warnings,
     -- finally processing without reports
   , W2 (INTR, ROW_NUM, RESULT) as (
       select W1.INT concat result concat CAST(RPAD(' ', 399, ' ') AS CHAR(400)) INTR
            , row_number() over ( partition by 'COMMON_VALUE'
               order by case W1.result
                        when 'E' then 1
                        when 'W' then 2
                        when ' ' then 3
                        end ,
                       W1.INT) ROW_NUM
                      , W1.RESULT
          from W1 order by  1) -- select trim(INTR) INTR, ROW_NUM, RESULT from W2;
     -- recursive CTE (Common Table Expression)
   , W3 (INTR, ROW_NUM, INT_LIST)
     as (select BASE.INTR, BASE.ROW_NUM, trim(BASE.INTR)
           from W2 BASE where BASE.ROW_NUM = 1    union all
          select t1.INTR, t1.ROW_NUM, INT_LIST concat ', ' concat trim(t1.INTR) concat ' '
            from W3 t0, W2 t1 where t0.ROW_NUM + 1 = t1.ROW_NUM
```

```
        ) -- select trim(INTR) INTR, ROW_NUM, INT_LIST from W3;
select ROW_NUM, INT_LIST from W3 RQ
 where RQ.ROW_NUM = (select max(ROW_NUM) from W3);
```

Questa query, partendo da una tabella contenente il risultati di elaborazioni di interfacce (W1), produce come risultato una sola riga contenente il numero delle interfacce e l'elenco di ogni singola interfaccia riportando prima quelle con E errore, poi quelle con W warning e infine quelle senza segnalazioni.

Descrizione dettagliata Query

Vediamo in dettaglio come è composta l'istruzione Sql

- **CTE W1 (Risultato Interfacce)**:

- Questa CTE è una simulazione di una tabella che utilizza la clausola **VALUES**, che crea un insieme temporaneo di righe (**INT, RESULT**). Ogni riga rappresenta un identificatore di interfaccia (**INT**) e il relativo risultato (**RESULT**), dove il risultato potrebbe essere un errore ('E'), un avviso ('W') o nessun problema (' ').

- **CTE W2 (Numerazione Progressiva):**

- Questa CTE aggiunge un numero progressivo a ciascuna riga da **W1** utilizzando la funzione finestra **ROW_NUMBER()**. Questa numerazione è partizionata in base a un valore costante (che di fatto significa nessun partizionamento poiché tutte le righe avranno lo stesso "COMMON_VALUE") e ordinata prima in base al tipo di risultato ("E", quindi "W", quindi "") e quindi dall'identificatore di interfaccia **INT**.

- L'istruzione **SELECT** concatena il valore **INT**, il **RESULT** e una stringa riempita a destra di 399 spazi convertiti in **CHAR(400)** per garantire che tutti i valori **INTR** abbiano una lunghezza uniforme.

- Questa numerazione ordinata aiuta nella successiva concatenazione degli identificatori di interfaccia nell'ordine desiderato.

- **CTE W3 (CTE Ricorsiva per Concatenazione):**

- Questa CTE è ricorsivo e crea una singola stringa (**INT_LIST**) contenente tutti gli identificatori di interfaccia, nell'ordine specificato da **W2**.

- Si inizia con il caso base in cui **ROW_NUM** è 1, selezionando il primo identificatore di interfaccia.

- La parte ricorsiva unisce **W3** a **W2** a condizione che **W3.ROW_NUM + 1 = W2.ROW_NUM**, aggiungendo di fatto il successivo identificatore di interfaccia all'elenco (**INT_LIST**) per ogni riga successiva.

- La query ricorsiva continua finché tutte le righe non vengono elaborate, creando un elenco cumulativo di identificatori di interfaccia.

Istruzione Select finale:

- L'istruzione **SELECT** finale recupera l'ultima riga da **W3**, che contiene l'elenco cumulativo di tutte le interfacce ordinate per tipo di risultato e **INT**. **ROW_NUM** in questo caso rappresenta il conteggio totale delle interfacce.

- Utilizza una sottoquery per trovare **max(ROW_NUM)** per garantire che venga selezionato solo l'elenco finale e completo.

In sintesi

In sintesi, la query è progettata per creare un elenco ordinato di identificatori di interfaccia, dando priorità agli errori ('E'), quindi agli avvisi ('W') e infine a quelli senza problemi (' '), e lo presenta come una singola stringa insieme al conteggio totale delle interfacce.

L'utilizzo della funzione **RPAD** e il cast su **CHAR(400)** garantiscono che ciascun identificatore nell'elenco sia formattato in modo uniforme, il che è particolarmente importante per il processo di concatenazione ricorsiva.

Sguardo più approfondito

Per capire meglio il funzionamento delle CTE Ricorsive proviamo a dare una sguardo più approfondito ai singoli risultati delle CTE intermedie.

CTE W2 (Numerazione Progressiva)

Questa CTE aggiunge una numerazione progressiva in base al tipo di risultato ("E", quindi "W", quindi "") e quindi dall'identificatore di interfaccia INT.

Questa numerazione ordinata aiuta nella successiva concatenazione degli identificatori di interfaccia nell'ordine desiderato.

```
select trim(INTR) INTR, ROW_NUM, RESULT from W2;
```

INTR	ROW_NUM	RESULT
0020E	1	E
0023E	2	E
0014W	3	W
0016W	4	W
0024W	5	W
0011	6	
0012	7	
0013	8	
0015	9	
0017	10	
0018	11	

0019	12	

CTE W3 (CTE Ricorsiva per Concatenazione)

Questa CTE ricorsiva crea iterativamente (in modo ricorsivo) una stringa (INT_LIST) contenente tutti gli identificatori di interfaccia, nell'ordine specificato da W2.

La query ricorsiva continua finché tutte le righe non vengono elaborate, creando un elenco cumulativo di identificatori di interfaccia.

```
select trim(INTR) INTR, ROW_NUM, INT_LIST from W3;
```

INTR	ROW_NUM	INT_LIST
0020E		0020E
0023E		0020E, 0023E
0014W		0020E, 0023E , 0014W
0016W		0020E, 0023E , 0014W , 0016W
0024W		0020E, 0023E , 0014W , 0016W , 0024W
0011		0020E, 0023E , 0014W , 0016W , 0024W , 0011
0012		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012
0013		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013
0015		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015
0017		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015 , 0017
0018		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015 , 0017 , 0018
0019		0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015 , 0017 , 0018 , 0019

Come si può vedere il risultato della query contiene tutte le righe delle varie iterazioni e quella che a noi interessa è l'ultima, ovvero quella dell'ultimo ciclo di ricorsività, che viene estratta dalla successiva e ultima CTE.

Istruzione Select finale

Questa CTE finale recupera l'ultima riga da W3, che contiene l'elenco cumulativo di tutte le interfacce ordinate per tipo di risultato e INT. ROW_NUM in questo caso rappresenta il conteggio totale delle interfacce

```
select ROW_NUM, INT_LIST
  from W3 RQ
 where RQ.ROW_NUM = (select max(ROW_NUM) from W3)
```

ROW_NUM	INT_LIST
12	0020E, 0023E , 0014W , 0016W , 0024W , 0011 , 0012 , 0013 , 0015 , 0017 , 0018 , 0019

Conclusioni

La query SQL illustrata utilizza Common Table Expression (CTE) per organizzare e processare i dati di interfaccia, creando una lista ordinata di identificatori di interfaccia con un conteggio totale. Prioritizzando prima gli errori ('E'), poi gli avvisi ('W') e infine gli identificatori senza segnalazioni (' '), il risultato finale è una singola riga che contiene il numero totale di interfacce e un elenco concatenato di ciascuna interfaccia, facilitando la revisione e l'analisi dei risultati del processo di interfaccia.

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

#CTE Ricorsive #SQL #CTE Common Table Expression

← PRECEDENTE
CTE Ricorsive: L'Eleganza della Semplicità in SQL – 3

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

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