

SQL : Module 5 : CTE et récursivité

Site: [LMS UCA 2020/2021](#)

Cours: EIIN512B - ECUE Bases de donnees relationnelles

Livre: SQL : Module 5 : CTE et récursivité

Imprimé par: niget Tom

Date: samedi 28 août 2021, 15:02

Table des matières

- 1. CTE
- 2. Réciusivité

1. CTE

Les expressions de table commune (CTE, Common Table Expression) permettent de nommer des résultats intermédiaires dans l'étendue d'exécution d'une seule instruction SELECT, INSERT, UPDATE, DELETE

Exemple:

```
WITH maCTE AS (  
  SELECT *  
    FROM maTable )  
, maCTE2 AS (  
  SELECT *  
    FROM maCTE  
      INNER JOIN maTable2 ON maCTE.Clé = maTable2.Clé2 )  
  
SELECT *  
  FROM maCTE2
```

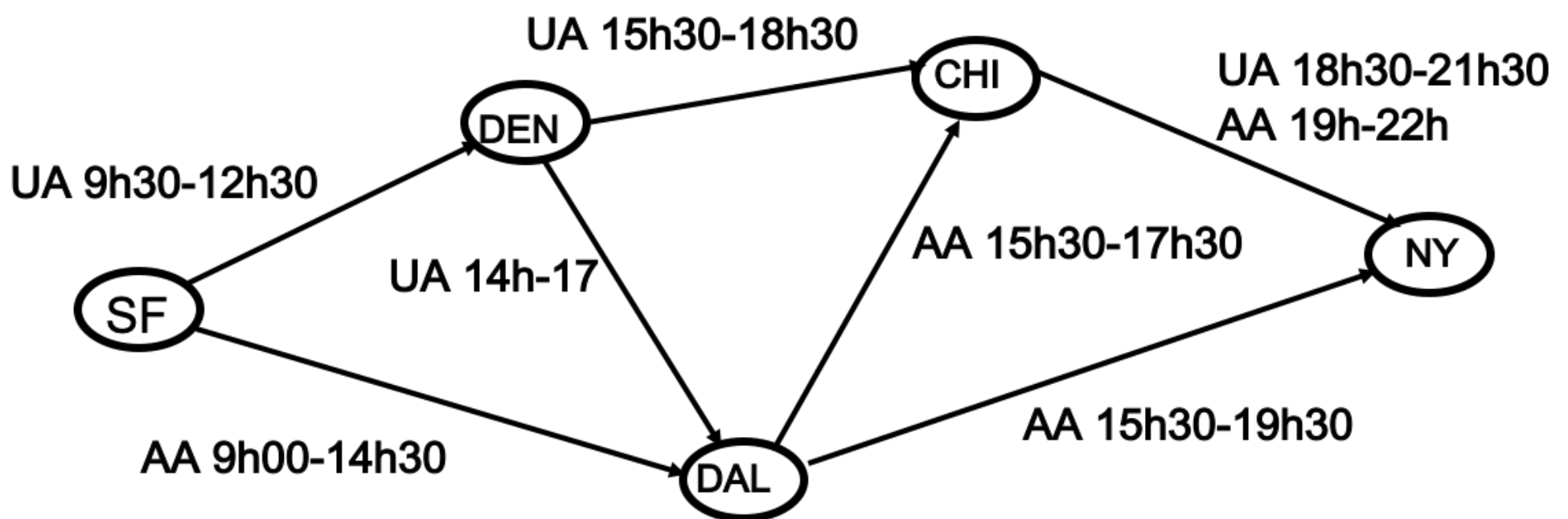
Chaque ordre auxiliaire dans une clause WITH peut être un SELECT, INSERT, UPDATE, ou DELETE;
et la clause WITH elle-même est attachée à un ordre primaire qui peut lui aussi être un SELECT, INSERT, UPDATE,
ou DELETE.

2. Réciusivité

□La récursivité en SQL s'effectue avec l'instruction WITH RECURSIVE où l'ordre primaire s'appelle lui même:

```
WITH RECURSIVE recursion(liste_champ) AS(  
  
-- Base de l'induction  
  
SELECT liste_champ FROM destables1  
  
UNION (ALL )  
  
-- Induction  
  
SELECT liste_champ FROM recursion, destables2  
  
WHERE condition )  
  
SELECT * FROM recursion;
```

Exemple



Soit la relation

vol(airline,from,to,duree,prix)

Requête: quelles sont toutes les connexions (voyages) possibles?

Les paires de villes connectées par des vols de ce graphe sont définies par la relation récursive:

Base

vol(airline,from,to,duree,prix) \Rightarrow voyage(from,to)

Induction

(vol(airline,x,z,duree,prix) \wedge voyage (z,y)) \Rightarrow voyage(x,y)

```
WITH RECURSIVE voyage (from,to) AS
```

```
(SELECT from, to FROM vol
```

```
UNION
```

```
SELECT R1.from, R2.to
```

```
FROM vol AS R1
```

```
JOIN voyage AS R2
```

```
ON R1.to =R2.from)
```

SELECT * FROM voyage ;

-- Attention aux boucles infinies si circuit dans le graphe

On souhaite optimiser son voyage, et connaitre pour chaque voyage, le nombre d’escales et le prix du voyage:

WITH RECURSIVE voyage (from,to,escales,prix) AS

(SELECT from, to, 0, prix FROM vol

UNION

SELECT R1.from, R2.to ,R2.escales+1,

R1.prix+R2.prix

FROM vol AS R1

JOIN voyage AS R2

ON R1.to =R2.from)

SELECT * FROM voyage ;