Travaux pratiques : Interrogation de l'entrepôt de données (cubes, cumuls, ensembles de regroupements et vues matérialisées)

Temps estimé nécessaire : 30 minutes

Objectif du laboratoire :

L'objectif de ce laboratoire est de fournir une expérience pratique des techniques avancées de requête SQL à l'aide de PostgreSQL dans un environnement Cloud IDE. Le laboratoire se concentre sur l'enseignement de la création d'ensembles de regroupement, de cumuls et de cubes pour l'agrégation et la synthèse des données, ainsi que sur la mise en œuvre et l'utilisation de tables de requêtes matérialisées (vues matérialisées) pour une interrogation efficace des données. Ces compétences sont essentielles pour gérer et analyser de grands ensembles de données, en particulier dans les contextes d'entreposage de données et de veille stratégique.

Avantages de l'apprentissage en laboratoire :

En complétant ce laboratoire, vous acquerrez des connaissances précieuses sur l'application pratique des requêtes SQL complexes et des techniques de manipulation des données. La connaissance des ensembles de regroupement, des cumuls et des cubes permettra aux apprenants de résumer et d'analyser efficacement les données, ce qui est essentiel pour prendre des décisions commerciales éclairées. La compréhension et la mise en œuvre des vues matérialisées offrent un moyen efficace de gérer des données à grande échelle en réduisant la charge de calcul lors des exécutions de requêtes fréquentes. Ces compétences sont très utiles pour les carrières dans l'analyse de données, l'administration de bases de données et la veille stratégique, améliorant votre capacité à gérer et à analyser des données dans des scénarios réels.

Objectifs

Dans ce laboratoire, vous apprendrez à créer :

- Ensembles de regroupement
- Enroulement
- Cube
- Vues matérialisées

Exercice 1 - Lancez une instance de serveur PostgreSQL sur Cloud IDE et ouvrez l'interface utilisateur graphique pgAdmin

Ce laboratoire nécessite que vous ayez terminé le laboratoire précédent Remplir un entrepôt de données.

Si vous n’avez pas encore terminé le laboratoire Remplir un entrepôt de données, veuillez le terminer avant de continuer.

GROUPING SETS, CUBE et ROLLUP nous permettent de créer facilement des sous-totaux et des totaux généraux de diverses manières. Tous ces opérateurs sont utilisés avec l'opérateur GROUP BY.

L'opérateur GROUPING SETS nous permet de regrouper des données de différentes manières dans une seule instruction SELECT.

L' opérateur **ROLLUP** permet de créer des sous-totaux et des totaux généraux pour un ensemble de colonnes. Les totaux récapitulatifs sont créés en fonction des colonnes transmises à l'opérateur ROLLUP.

L' opérateur **CUBE** génère des sous-totaux et des totaux généraux. De plus, il génère des sous-totaux et des totaux généraux pour chaque permutation des colonnes fournies à l'opérateur CUBE.

Exercice 2 – Écrire une requête en utilisant des ensembles de regroupement

Après avoir lancé une instance de serveur PostgreSQL sur Cloud IDE et ouvert l'interface utilisateur graphique pgAdmin, exécutez la requête ci-dessous.

Pour créer un ensemble de regroupement pour trois colonnes intitulées année, catégorie et somme du montant facturé, exécutez l'instruction SQL ci-dessous.

```
select year,category, sum(billedamount) as totalbilledamount
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by grouping sets(year,category);
```

La sortie partielle peut être vue dans l'image ci-dessous.

Exercice 3 – Écrire une requête en utilisant rollup

Pour créer un cumul en utilisant les trois colonnes année, catégorie et somme du montant facturé, exécutez l'instruction SQL ci-dessous.

```
select year,category, sum(billedamount) as totalbilledamount
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by rollup(year,category)
order by year, category;
```

La sortie partielle peut être vue dans l'image ci-dessous.

Exercice 4 – Écrire une requête en utilisant un cube

Pour créer un cube en utilisant les trois colonnes intitulées année, catégorie et somme du montant facturé, exécutez l'instruction SQL ci-dessous.

```
select year,category, sum(billedamount) as totalbilledamount
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by cube(year,category)
order by year, category;
```

La sortie partielle peut être vue dans l'image ci-dessous.

Exercice 5 – Créer une table de requête matérialisée (vues matérialisées)

Dans pgAdmin, nous pouvons implémenter des vues matérialisées à l'aide de tables de requêtes matérialisées.

Étape 1 : créer les vues matérialisées.

Exécutez l'instruction SQL ci-dessous pour créer une vue matérialisée nommée countrystats.

```
CREATE MATERIALIZED VIEW countrystats (country, year, totalbilledamount) AS
(select country, year, sum(billedamount)
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by country,year);
```

La commande ci-dessus crée une vue matérialisée nommée countrystats qui comporte 3 colonnes.

- Pays
- Année
- montant total facturé

Les vues matérialisées sont essentiellement le résultat de la requête ci-dessous, qui vous donne l'année, le nom du trimestre et la somme du montant facturé regroupé par année et nom du trimestre.

```
select year, quartername, sum(billedamount) as totalbilledamount
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by grouping sets(year, quartername);
```

Étape 2 : Renseignez/actualisez les données dans les vues matérialisées.

Exécutez l'instruction SQL ci-dessous pour remplir les vues matérialisées countrystats.

```
REFRESH MATERIALIZED VIEW countrystats;
```

La commande ci-dessus remplit les vues matérialisées avec les données pertinentes.

Étape 3 : interroger les vues matérialisées.

Une fois qu'une vue matérialisée est actualisée, vous pouvez l'interroger.

Exécutez l'instruction SQL ci-dessous pour interroger les vues matérialisées countrystats.

```
select * from countrystats;
```

Exercices pratiques

Problème 1 : Créer un ensemble de regroupement pour les colonnes year, quartername, sum(billedamount).

▼ Cliquez ici pour un indice

Assurez-vous que ce tableau contient l'année et le nom du trimestre.

▼ Cliquez ici pour la solution

```
select year, quartername, sum(billedamount) as totalbilledamount
from "FactBilling"
left join "DimCustomer"
on "FactBilling".customerid = "DimCustomer".customerid
left join "DimMonth"
on "FactBilling".monthid="DimMonth".monthid
group by grouping sets(year, quartername);
```

Problème 2 : Créer un cumul pour les colonnes pays, catégorie, somme(montant facturé).

▶ Cliquez ici pour un indice
▼ Cliquez ici pour la solution

```
SELECT country, category, SUM(billedamount) AS totalbilledamount
FROM "FactBilling"
LEFT JOIN "DimCustomer"
ON "FactBilling".customerid = "DimCustomer".customerid
```

```
LEFT JOIN "DimMonth"
ON "FactBilling".monthid = "DimMonth".monthid
GROUP BY ROLLUP(country, category)
ORDER BY country, category;
```

Problème 3 : Créer un cube pour les colonnes année, pays, catégorie, somme (montant facturé).

▶ Cliquez ici pour un indice
▶ Cliquez ici pour la solution

Problème 4 : Créer une vue matérialisée nommée average_billamount avec les colonnes année, trimestre, catégorie, pays, average_bill_amount.

▶ Cliquez ici pour un indice
▼ Cliquez ici pour la solution

```
CREATE MATERIALIZED VIEW average_billamount (year,quarter,category,country, average_bill_amount) AS
    (select   year,quarter,category,country, avg(billedamount) as average_bill_amount
    from "FactBilling"
    left join  "DimCustomer"
    on "FactBilling".customerid =  "DimCustomer".customerid
    left join "DimMonth"
    on "FactBilling".monthid="DimMonth".monthid
    group by year,quarter,category,country
    );

refresh MATERIALIZED VIEW average_billamount;
```

Félicitations ! Vous avez terminé avec succès le laboratoire Interrogation de l'entrepôt de données (cubes, cumuls, ensembles de regroupement et vues matérialisées).

Auteur

Amrutha Rao