Rapport d’évaluation de la performance de la base de données

Analyse des requêtes avant et après l’indexation

Requêtes 1

explain analyze

SELECT * FROM utilisateurs WHERE email = 'alice@email.com';

Résulat avant indexation

Seq Scan on utilisateurs (cost=0.00..11.62 rows=1 width=562) (actual time=0.024..0.025 rows=1.00 loops=1)

Résultat après indexation

Seq Scan on utilisateurs (cost=0.00..1.06 rows=1 width=562) (actual time=0.017..0.018 rows=1.00 loops=1)

Requêtes 2

explain analyze

SELECT * FROM produits WHERE prix BETWEEN 100 AND 500;

Resulat avant indexation

Seq Scan on produits (cost=0.00..13.15 rows=1 width=360) (actual time=0.033..0.034 rows=1.00 loops=1)

Résultat après indexation

Seq Scan on produits (cost=0.00..1.07 rows=1 width=360) (actual time=0.029..0.030 rows=1.00 loops=1)

Requêtes 3

explain analyze

SELECT p.nom_produit, u.nom, COUNT(c.id) as nombre_achats

FROM commandes c

JOIN produits p ON c.produit_id = p.id

JOIN utilisateurs u ON c.utilisateur_id = u.id

WHERE p.categorie = 'Electronique' AND u.ville = 'Paris'

GROUP BY p.nom_produit, u.nom;

Résultat avant indexation

GroupAggregate (cost=8.21..8.23 rows=1 width=444) (actual time=0.114..0.116 rows=2.00 loops=1)

Résultat après indexation

HashAggregate (cost=4.76..4.77 rows=1 width=444) (actual time=0.107..0.108 rows=2.00 loops=1)

Requête 4

```
explain analyse

SELECT u.nom, SUM(c.montant_total)

FROM utilisateurs u

JOIN commandes c ON u.id = c.utilisateur_id

GROUP BY u.nom

ORDER BY SUM(c.montant_total) DESC;
```

Résultat avant indexation

Sort (cost=12.09..12.34 rows=100 width=250) (actual time=0.267..0.269 rows=3.00 loops=1)

Résultat après indexation

Sort (cost=4.22..4.23 rows=5 width=250) (actual time=0.077..0.078 rows=3.00 loops=1)

Requête 5

```
explain analyse

SELECT p.categorie, SUM(c.montant_total) as CA

FROM commandes c

JOIN produits p ON c.produit_id = p.id

GROUP BY p.categorie;
```

Résultat avant indexation

HashAggregate (cost=8.18..9.43 rows=100 width=150) (actual time=0.188..0.191 rows=3.00 loops=1)

Résultat après indexation

HashAggregate (cost=4.10..4.16 rows=5 width=150) (actual time=0.073..0.074 rows=3.00 loops=1)

Requête 6

```
explain analyze

SELECT p.nom_produit

FROM produits p

LEFT JOIN commandes c ON p.id = c.produit_id

WHERE c.id IS NULL;
```

Résultat avant indexation

Hash Right Join (cost=14.72..16.99 rows=1 width=218) (actual time=0.049..0.050 rows=1.00 loops=1)

Résultat après indexation

Hash Right Join (cost=1.11..3.60 rows=1 width=218) (actual time=0.040..0.042 rows=1.00 loops=1)

Requête 7

```
explain analyze

SELECT u.ville, SUM(c.montant_total) AS chiffre_affaires

FROM utilisateurs u

JOIN commandes c ON u.id = c.utilisateur_id

GROUP BY u.ville

ORDER BY chiffre_affaires DESC;
```

Résultat avant indexation

Sort (cost=12.09..12.34 rows=100 width=150) (actual time=0.098..0.099 rows=3.00 loops=1)

Résultat après indexation

Sort (cost=4.22..4.23 rows=5 width=150) (actual time=0.144..0.145 rows=3.00 loops=1)

Requête 8

```
explain analyze

SELECT u.nom, COUNT(c.id) AS nombre_de_commandes

FROM utilisateurs u
```

```
JOIN commandes c ON u.id = c.utilisateur_id

GROUP BY u.nom

HAVING COUNT(c.id) > 3;
```

Résultat avant indexation

HashAggregate (cost=7.52..8.77 rows=33 width=226) (actual time=0.074..0.075 rows=3.00 loops=1)

Résultat après indexation

HashAggregate (cost=4.10..4.16 rows=2 width=226) (actual time=0.119..0.121 rows=3.00 loops=1)

Requête 9

```
explain analyze

SELECT c.id, p.nom_produit, c.montant_total

FROM commandes c

JOIN produits p ON c.produit_id = p.id

WHERE p.prix > (SELECT AVG(prix) FROM produits);
```

Résultat avant indexation

Nested Loop (cost=12.79..20.33 rows=33 width=228) (actual time=0.040..0.069 rows=58.00 loops=1)

Résultat après indexation

Hash Join (cost=2.16..4.65 rows=40 width=228) (actual time=0.124..0.158 rows=58.00 loops=1)

Requête 10

```
explain analyze

SELECT

  date_trunc('month', c.date_commande) AS mois,

  p.categorie,

  SUM(c.montant_total) AS chiffre_affaires,

  COUNT(c.id) AS nombre_ventes
```

```
FROM commandes c
JOIN produits p ON c.produit_id = p.id
GROUP BY mois, p.categorie
ORDER BY mois DESC, chiffre_affaires DESC
LIMIT 10;
```

Résultat avant indexation

Limit (cost=11.29..12.01 rows=10 width=166) (actual time=0.142..0.143 rows=10.00 loops=1)

Résultat après indexation

Limit (cost=7.21..7.93 rows=10 width=166) (actual time=0.122..0.123 rows=10.00 loops=1)