EXEMPLE PERSONNEL / MESURE RÉELLE / 2 OCTOBRE 2026

Un filtre de commandes. 280 000 lignes, même résultat.

Une table de commandes fictives a été créée sur une instance MariaDB locale. Un index composé a été posé puis retiré pour comparer le plan, l'exécution et le retour arrière.

Étape	Accès	Lignes lues	ANALYZE, 1 exécution	Médiane de 7 lectures
Avant index	ALL	280,000	99.377 ms	144.6 ms
Après index	range	50	0.178 ms	55.4 ms
Retour arrière	ALL	280,000	87.555 ms	144.3 ms

Les temps de la dernière colonne comprennent le lancement du client CLI. ANALYZE décrit une exécution par étape ; ces nombres ne sont pas des temps d'un back-office.

La requête contrôlée

```
SELECT id_order, reference, total_paid FROM demo_orders
WHERE id_shop=3 AND current_state=2 AND date_add>='2026-09-01'
ORDER BY date_add DESC, id_order DESC LIMIT 50;

CREATE INDEX demo_filter ON demo_orders
(id_shop, current_state, date_add, id_order);
```

Les 50 lignes retournées restent identiques

Empreinte SHA-256 identique avant index, après index et après suppression. Le plan utilise l'index seulement quand il existe. Six contrôles réussis ; serveur arrêté après l'exécution.

MariaDB 11.8.6-MariaDB-5ubuntu0.1 from Ubuntu · données synthétiques · 64 Mo de buffer InnoDB · résultat enregistré à 2026-10-02T12:09:55.740795+00:00

Ce que cette preuve permet de proposer

Reproduire un filtre lent, relever le plan et les lignes lues, tester un index sur une copie puis comparer les résultats. Les requêtes, versions, volumes et verrous réels du client doivent être examinés avant toute modification.

Exemple indépendant, aucune référence client. Ni boutique PrestaShop, ni API Amazon, ni Plesk connectés. Aucun gain de performance client garanti. Rapport JSON complet et script conservés séparément.