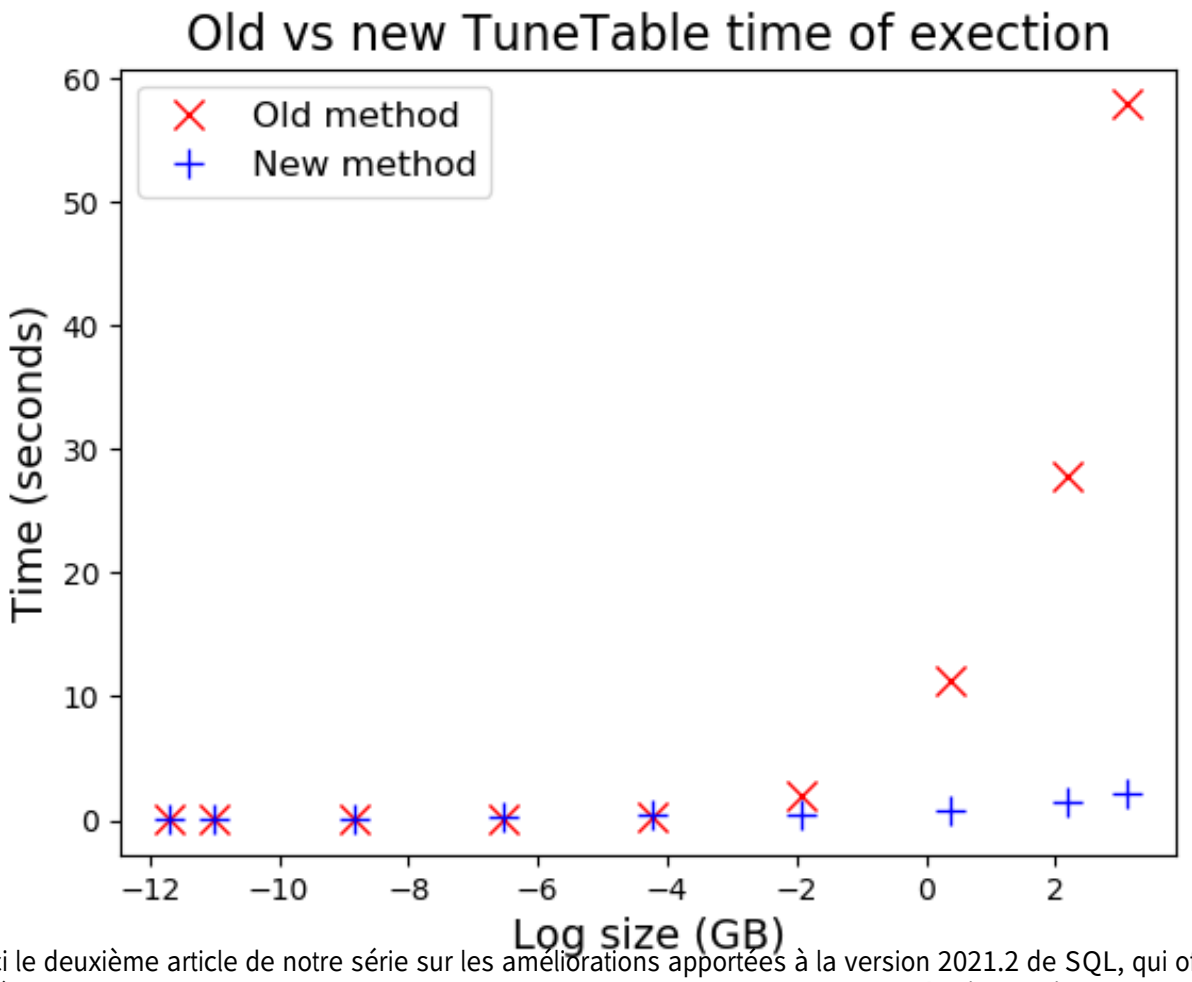
Article
Guillaume Rongier · Mai 3, 2022 5m de lecture

# 2021.2 Fonctionnalité SQL en vedette - Échantillonnage intelligent et automatisation des statistiques de table




Voici le deuxième article de notre série sur les améliorations apportées à la version 2021.2 de SQL, qui offre une expérience SQL adaptative et performante. Dans cet article, nous allons examiner **les innovations en matière de collecte Table Statistics**, qui sont bien sûr le principal élément d'entrée pour la capacité de Run Time Plan Choice que nous avons décrite dans l'article précédent.

Vous nous avez probablement entendu dire cela à plusieurs reprises : Tunez vos tables!

Pour ajuster vos tableaux à l'aide de la [commande SQL ŤUNE TABLE]( https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls...
) ou [$SYSTEM.SQL.Stats.Table ŏObjectScript API](https://docs.intersystems.com/irislatest/csp/documatic/%25CSP.Documatic....) vous devez recueillir des statistiques sur les données de votre tableau pour aider IRIS SQL à élaborer un bon plan de requête. Ces statistiques comprennent des informations importantes telles que le nombre approximatif de lignes dans le tableau, ce qui aide l'optimiseur à décider de choses telles que l'ordre des JOIN (commencer par le tableau le plus petit est généralement plus efficace). De nombreux appels au support InterSystems concernant les performances des requêtes peuvent être résolus en exécutant simplement ŤUNE TABLE ŏet en faisant un nouvel essai, car l'exécution de la commande invalidera les plans de requêtes existants afin que la prochaine invocation prenne en compte les nouvelles statistiques. D'après ces appels au support, nous voyons deux raisons récurrentes pour

lesquelles ces utilisateurs n'avaient pas encore collecté les statistiques de tableaux : ils n'en connaissaient pas l'existence, ou ils ne pouvaient pas se permettre les frais d'exécution sur le système de production. Dans la version 2021.2, nous avons résolu ces deux problèmes.

### Échantillonnage au niveau des blocs

Commençons par le second : le coût de la collecte des statistiques. Il est vrai que la collecte de statistiques sur les tables peut entraîner une quantité considérable d'entrées/sorties (I/O) et donc une surcharge *si* vous analysez l'ensemble de la table. L'API prenait déjà en charge l'échantillonnage seulement d'un sous-ensemble de lignes, mais cette opération avait tout de même la réputation d'être coûteuse. Dans la version 2021.2, nous avons modifié la situation pour ne plus sélectionner des lignes aléatoires en bouclant sur la globale de la carte principale, mais pour atteindre immédiatement le stockage physique sous-jacent et laisser le noyau prendre un échantillon aléatoire des blocs de base de données bruts pour cette globale. À partir de ces blocs échantillonnés, nous déduisons les lignes de tableau SQL qu'ils stockent et poursuivons avec notre logique habituelle de construction de statistiques par champ.

On peut comparer cela au fait de se rendre à un grand festival de la bière et, au lieu de parcourir toutes les allées et de choisir quelques stands de brasseries pour mettre une bouteille chacun dans son panier, de demander aux organisateurs de vous donner un casier avec des bouteilles prises au hasard et de vous épargner la marche (dans cette analogie de dégustation de bière, la marche serait en fait une bonne idée ![wink](https://community.intersystems.com/sites/all/libraries/ckeditor/plugins/... "wink") ). Pour se calmer, voici un simple graphique représentant l'ancienne approche basée sur les lignes (croix rouges) par rapport à l'approche basée sur les blocs (croix bleues). Les avantages sont énormes pour les tableaux de grande taille, qui se trouvent être ceux sur lesquels certains de nos clients hésitaient à utiliser `TUNE TABLE`...

Il y a un petit nombre de limitations à l'échantillonnage par bloc, le plus important étant qu'il ne peut pas être utilisé sur des tables qui se trouvent en dehors des mappages de stockage par défaut (par exemple, en projetant à partir d'une structure globale personnalisée en utilisant `%Storage.SQL`). Dans de tels cas, nous reviendrons toujours à l'échantillonnage par ligne, exactement comme cela fonctionnait dans le passé.

### Configuration automatique

Maintenant que nous avons résolu le problème de la perception de la surcharge, considérons l'autre raison pour laquelle certains de nos clients n'utilisaient pas `TUNE TABLE` : c'est qu'ils ne le savaient pas. Nous pourrions essayer de documenter notre façon de nous en sortir (et nous reconnaissons qu'il y a toujours de la place pour de meilleures documents), mais nous avons décidé que cet échantillonnage par blocs super efficace fournit en fait une opportunité de faire quelque chose que nous avons longtemps voulu : automatiser le tout. À partir de 2021.2, lorsque vous préparez une requête sur une table pour lequel aucune statistique n'est disponible, nous utiliserons d'abord le mécanisme d'échantillonnage par blocs ci-dessus pour collecter ces statistiques et les utiliser pour la planification de la requête, en sauvegardant les statistiques dans les métadonnées de la table afin qu'elles puissent être utilisées par les requêtes suivantes.

Si cela peut sembler effrayant, le graphique ci-dessus montre que ce travail de collecte de statistiques ne prend que quelques secondes pour les tables de la taille d'un Go. Si vous utilisez un plan de requête inapproprié pour interroger une telle table (en raison de l'absence de statistiques appropriées), cela risque d'être beaucoup plus coûteux que de procéder à un échantillonnage rapide dès le départ. Bien sûr, nous ne le ferons que pour les tables où nous pouvons utiliser l'échantillonnage par blocs et (malheureusement) nous procéderons sans statistiques pour ces tables spécialles qui ne supportent que l'échantillonnage par lignes.

Comme pour toute nouvelle fonctionnalité, nous sommes impatients de connaître vos premières expériences et vos commentaires. Nous avons d'autres idées concernant l'automatisation dans ce domaine, comme la mise à jour des statistiques en fonction de l'utilisation des tableaux, mais nous aimerions nous assurer que ces idées sont fondées sur les expériences acquises en dehors du laboratoire.

#SQL #Tables relationnelles #InterSystems IRIS

**URL de la source:**https://fr.community.intersystems.com/post/20212-fonctionnalit%C3%A9-sql-en-

vedette-%C3%A9chantillonnage-intelligent-et-automatisation-des