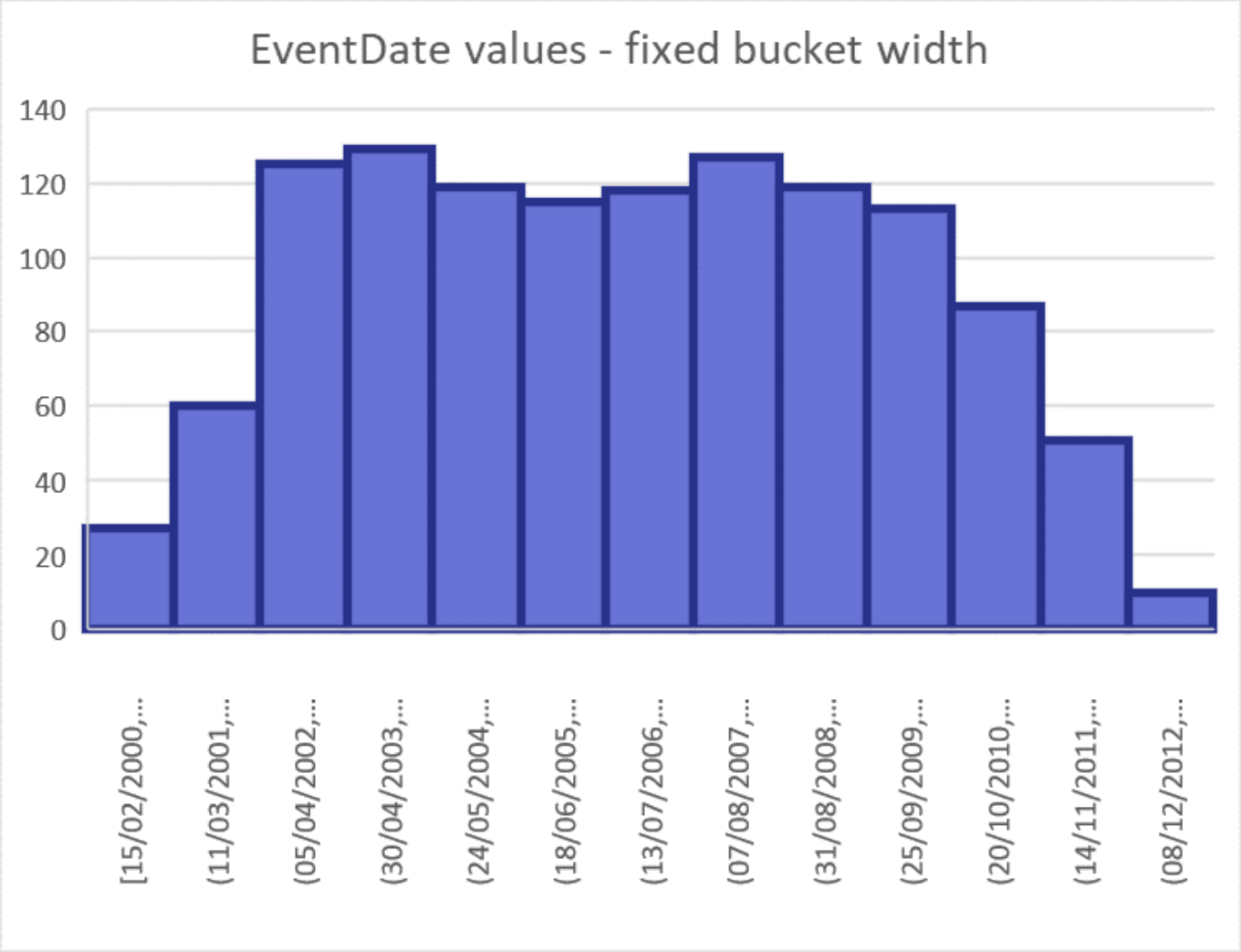
Article
Guillaume Rongier · Mai 6, 2022 4m de lecture

2021.2 Fonctionnalité SQL en vedette - Statistiques avancées de tables

Voici le troisième article de notre courte série sur les innovations d'IRIS SQL qui offrent une expérience plus adaptative et plus performante aux analystes et aux applications requérant des données relationnelles sur IRIS. Il s'agit peut-être du dernier article de cette série pour 2021.2, mais nous prévoyons plusieurs autres améliorations dans ce domaine. Dans cet article, nous allons approfondir un peu plus les statistiques de tableaux supplémentaires que nous commençons à rassembler dans cette version : **Histogrammes**



Que signifie le mot "histogramme" ?

Un histogramme est une représentation approximative de la distribution des données d'un champ numérique (ou de manière plus générale des données qui ont un ordre précis). Il est utile de connaître la valeur la plus petite, la plus grande et la moyenne d'un tel champ, mais cela ne vous dit pas grand-chose sur la façon dont les données sont réparties entre ces trois points. C'est là qu'intervient l'histogramme, qui divise la plage de valeurs en "godets" et compte le nombre de valeurs de champ qui apparaissent dans chaque godet. Il s'agit d'une définition assez souple et vous pouvez toujours choisir de prendre la taille des godets de telle sorte que les godets soient également "larges" en termes de valeurs de champ, ou également "larges" en termes de nombre de valeurs échantillonnées couvertes. Dans ce dernier cas, chaque godet contient le même pourcentage de valeurs et les

godets représentent donc des percentiles. Le graphique ci-dessus trace un histogramme pour le champ EventData de l'ensemble de données [Aviation Demo dataset](https://github.com/intersystems/Samples-Aviation), en utilisant la même largeur de godet exprimée en nombre de jours.

Pourquoi aurais-je besoin d'un histogramme ?

Supposons que vous cherchiez dans cet ensemble de données tous les événements antérieurs à 2004 dans l'État de Californie :

```
SELECT * FROM Aviation.Event WHERE EventDate < '2004-05-01' AND LocationCountry = 'Ca
lifornia'
```

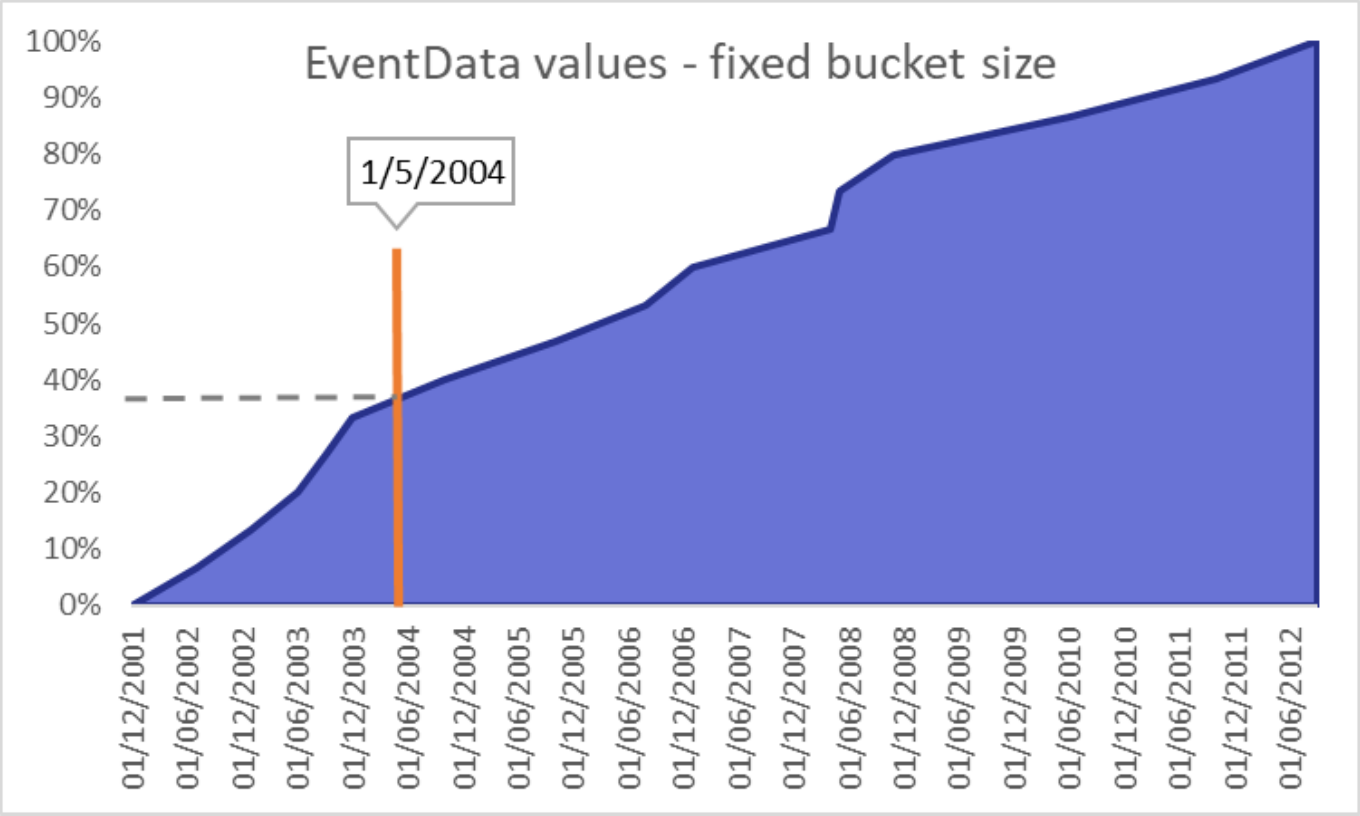
Dans notre article précédent sur [Choix du plan d'exécution](https://fr.community.intersystems.com/post/20212-fonctionnalit%C3%A9-sql...), nous avons déjà discuté comment capturer la sélectivité et les valeurs aberrantes potentielles pour un champ comme LocationCountry dans les statistiques de la table. Mais de telles statistiques pour les valeurs de champs individuels ne sont pas très pratiques pour cette condition <`sur EventDate. Pour calculer la sélectivité de cette condition, vous devez agréger la sélectivité de toutes les valeurs possibles de EventDate jusqu'au 1er mai 2004, ce qui peut être une requête assez exigeante en soi plutôt que le genre d'estimation rapide que vous pouvez vous permettre au moment de la planification de la requête. C'est là que les histogrammes sont utiles.

Reprenons notre histogramme pour la distribution des valeurs EventDate, en divisant cette fois les données en 16 sections de même taille, chacune contenant 6,667 % des données. De cette façon, les choses se traduisent plus facilement en percentiles et en nombres de sélectivité que nous pouvons utiliser pour les estimations du coût des requêtes. Pour lire ce tableau, regardons la quatrième ligne : 20 % des valeurs (3 godets de 6,667 % chacun) précèdent la limite inférieure de ce godet du 22 juin 2003, et il contient 6,667 % de valeurs supplémentaires, jusqu'au 19 septembre 2003.

Godet	Percentile	Valeur
	0%	21/12/2001
1	7%	02/07/2002
2	13%	19/01/2003
3	20%	22/06/2003
4	27%	19/09/2003
5	33%	30/12/2003
6	40%	01/10/2004
7	47%	01/10/2005
8	53%	20/08/2006
9	60%	14/01/2007
10	67%	02/04/2008
11	73%	14/05/2008
12	80%	29/11/2008
13	87%	01/06/2010
14	93%	30/10/2011
15	100%	26/09/2012

La date de coupure utilisée dans l'exemple de requête ci-dessus (1er mai 2004) se trouve dans le cinquième godet, et comporte entre 33 % et 40 % des valeurs précédant cette date. Au fur et à mesure que les godets deviennent plus petits, nous pouvons considérer que la distribution à l'intérieur de ceux-ci est approximativement uniforme et simplement interpoler entre les limites inférieure et supérieure, ce qui dans ce cas conduit à une sélectivité d'environ 37%, que nous pouvons utiliser dans notre estimation du coût de la requête.

Voici une autre façon de visualiser notre utilisation des histogrammes, en les traçant comme un barre de distribution cumulative. Nous pouvons voir comment la ligne tracée pour le 1er mai 2004 sur l'axe X (les valeurs), se traduit par 37% sur l'axe Y.



L'exemple ci-dessus utilise une condition d'intervalle avec juste une limite supérieure pour plus de clarté, mais l'approche fonctionne évidemment aussi bien lorsqu'on utilise une limite inférieure ou une condition d'intervalle (par exemple en utilisant le prédicat *BETWEEN*).

À partir de la version 2021.2, nous collectons des histogrammes dans le cadre des statistiques de tables pour tout champ organisé, y compris les chaînes de caractères, et nous les utilisons pour estimer la sélectivité des plages dans le cadre du RTPC. De nombreuses requêtes du monde réel impliquent une condition de plage sur les champs de date (et autres). Nous sommes donc convaincus que cette amélioration d'IRIS SQL sera bénéfique à la planification des requêtes pour bon nombre de nos clients et, comme toujours, nous sommes impatients de connaître vos expériences.

#SQL #Tables relationnelles #InterSystems IRIS

URL de la source:https://fr.community.intersystems.com/post/20212-fonctionnalit%C3%A9-sql-en-vedette-statistiques-avanc%C3%A9es-de-tables