Article
Irène Mykhailova · Avr 18, 2022 7m de lecture

Accès aux bases de données NoSQL avec l'API native Python

Vous avez probablement entendu parler des bases de données NoSQL. Il existe plusieurs définitions, mais pour simplifier, ce terme est couramment utilisé pour désigner les bases de données qui n'utilisent littéralement pas le langage SQL, c'est-à-dire les bases de données autres que les bases de données relationnelles (RDB).

InterSystems IRIS Data Platform vous permet de définir des tableaux et d'accéder aux données en SQL. Par conséquent, InterSystems IRIS Data Platform n'est pas strictement une base de données NoSQL. Cependant, les " Globales " à l'origine des hautes performances d'InterSystems IRIS sont une technologie de base d'InterSystems depuis 40 ans, fournissant ce que nous appellerions aujourd'hui une base de données NoSQL. Cet article montre comment créer des structures graphiques dans les "Globales" InterSystems IRIS et y accéder en Python.

NoSQL

Il y a des bases de données classées comme NoSQL qui traitent une variété de modèles de données. Des exemples typiques sont énumérés ci-dessous.

- **Key-Value**: Maintient la correspondance entre les clés et les valeurs. Une base de données typique : Redis
- **Document**: JSON ou XML sont utilisés comme modèle de données. Une base de données typique : MongoDB
- **Graph**: Un modèle de structure de graphe comprenant des sections et des nœuds. Une base de données typique : Neo4j

Il existe une multitude d'autres produits et modèles de données.

Historique de NoSQL

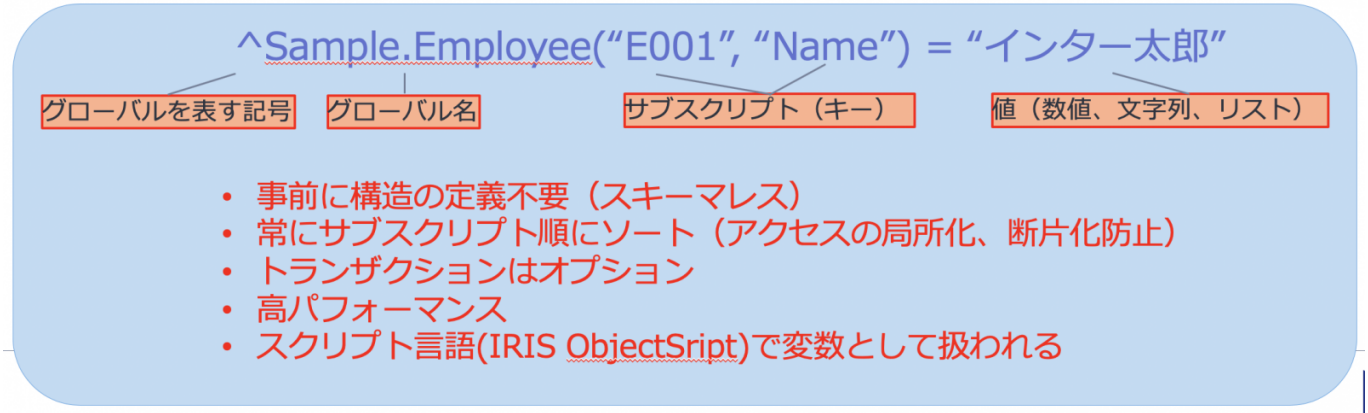
Pourquoi NoSQL est-il apparu et pourquoi est-il devenu si populaire ? Dans cet article, nous souhaiterions citer les raisons suivantes.

- **Les structures de données qui ne peuvent pas être représentées naturellement au moyen de tableaux** : Théoriquement, toute structure de données peut être représentée dans un tableau et accessible par SQL, mais il peut y avoir des problèmes tels que la complexité SQL et des performances médiocres. Dans de tels cas, il est préférable d'utiliser un modèle de données adapté à la structure plutôt qu'un tableau.
- **Optimisation pour les environnements distribués** : L'un des domaines où l'utilisation de NoSQL a pris de l'ampleur est celui de l'infrastructure des services de réseaux sociaux tels que Facebook. Pour prendre en charge de grandes quantités de données, de nombreux utilisateurs et une haute disponibilité, il est essentiel de distribuer la base de données. Les RDB ne fonctionnent pas toujours de manière optimale dans un environnement distribué. L'approche transactionnelle et sous-transactionnelle de la RDB, décrite ci-dessous, est également un facteur qui rend difficile sa mise en œuvre dans un environnement distribué.
- **Demande de traitement d'une transaction** : Le traitement des transactions et l'intégrité des données sont des fonctions dans lesquelles les RDBs possèdent des points forts. Il va sans dire qu'un support strict du traitement des transactions et de l'intégrité des données est essentiel pour les systèmes qui gèrent les comptes bancaires, par exemple. Cependant, le maintien de l'intégrité des données dans le traitement des transactions est très difficile à mettre en œuvre dans un environnement distribué, notamment en ce qui concerne la compatibilité avec les performances. Par conséquent, si vous souhaitez donner la priorité à l'amélioration des performances et de la disponibilité dans un environnement distribué plutôt qu'au maintien

d'une cohérence stricte, NoSQL pourrait mieux répondre à vos besoins que RDB.

Globales

InterSystems IRIS utilise une structure de données appelée Globales au cœur de la base de données. Les Globales sont modélisées comme des variables de tableau, comme le montre la figure suivante.



Les Globales n'ont pas besoin d'être définies préalablement (sans schéma), ce qui permet une conception très souple des structures de données. En outre, les données sont toujours triées par ordre d'indice (clé), tant sur le disque que dans le cache mémoire, ce qui localise et accélère l'accès aux données et évite la fragmentation.

Bien que les Globales aient la forme de variables de tableau, ils peuvent en fait être utilisés dans IRIS ObjectScript, qui est un langage de script pour IRIS, avec presque la même syntaxe que les variables en mémoire, ce qui améliore les perspectives de développement de programmes. Pour plus d'informations sur les Globales, voir la section Global Usage Documentation.

Accès en utilisant Python

Voici un exemple de manipulation de Globales en utiisant Python à l'aide de Python Native API.

Nous utiliserons les données "WikiSpeedia" de SNAP à l'Université de Standford comme point de départ. WikiSpeedia est un jeu dans lequel les joueurs s'affrontent pour voir à quelle vitesse ils peuvent atteindre un mot cible à partir d'un mot donné en suivant uniquement les liens Wikipédia. SNAP compte environ 110 000 liens suivis par des utilisateurs réels, qui sont stockés globalement dans InterSystems IRIS.

Structure du graphe

Cette section décrit la structure du graphe.

La figure suivante présente les principes de base de la structure d'un graphe. Un graphe est constitué de nœuds et d'arêtes. Une arête relie deux nœuds.

Par exemple, une relation d'amitié sur Facebook peut être représentée par nœud = utilisateur, arête = relation d'amitié. Une structure de données pouvant être utilisée pour la recherche d'itinéraires est aussi réalisable en définissant le nœud = station, l'arête = station voisine de la précédente station, et la caractéristique de l'arête = distance entre les stations.

Dans l'exemple de ce document, la structure du graphe est utilisée comme suit : nœud = article Wikipédia (mot-clé) et arête = relation entre l'article original et l'article lié vers lequel l'utilisateur de WikiSpeedia a cliqué.

Structure des Globales

Examinons la structure des Globales.

```
^Links("Tokyo", "18th_century")=""
^Links("Tokyo", "London")=""
^Links("Osaka", "Aquarium")=""
```

Par exemple, la ligne supérieure indique que l'utilisateur a cliqué sur un lien à partir de l'article "Tokyo" vers l'article "18thcentury".
La troisième ligne indique que l'utilisateur a cliqué à partir de "Osaka" vers "Aquarium".

Les Globales sont optimisées pour un accès de gauche à droite aux souscripteurs (clés). Ainsi, le

```
^Links("Article original", "Cliquer sur l'article")=""
```

est optimale. En outre, dans ce cas, la valeur est "", mais si vous voulez que les arêtes aient un certain attribut, vous pouvez le définir sur une valeur globale.

Code Python

Cette section décrit le code Python. D'abord, c'est import.

```
import irisnative
import networkx as nx
```

Pour utiliser l'API native Python, importez le module irisnative. Ce module est un fichier .whl placé dans le répertoire dev/python lors de l'installation d'IRIS et installé dans l'environnement python avec la commande pip. De plus, comme NetworkX est utilisé pour maintenir la structure du graphe, importez-le également.

```
connection = irisnative.createConnection("localhost", 9091, "user", "horita","horita"
)
iris_native = irisnative.createIris(connection)
```

Dans ces deux lignes, createConnection() établit la connexion à IRIS et createIris() crée l'objet d'interface pour les opérations globales.

```
def addNodes(key, g, d):
    g.add_node(key)
    if d > 3:
        return

    iter = iris_native.iterator("Links", key)
    nodelist = [k for k,v in iter.items()]

    Limité à 3 liens à suivre pour des raisons de démonstration
    random.shuffle(nodelist)
    nodelist = nodelist[0:3]

    edgelist = [(key, n) for n in nodelist]
    g.add_nodes_from(nodelist)
    g.add_edges_from(edgelist)

    for subk in nodelist:
        addNodes(subk, g, d + 1)
```

Définition de la fonction addNodes(). Cette fonction construit une structure de graphe en tant qu'objet NetworkX en suivant les liens des noeuds dans un graphe donné. Des appels récursifs sont utilisés pour traverser le troisième niveau de la hiérarchie. Le point principal ici est

```
    iter = iris_native.iterator("Links", key)
    nodelist = [k for k,v in iter.items()]
```

Dans irisnative.iterator("Links", key), l'API native Python crée un itérateur pour l'indice. Par exemple, key='Tokyo' serait un itérateur qui itère sur la chaîne dans la partie * de ^Links("Tokyo", *).

L'itérateur est ensuite répété dans Python pour créer une liste. Comme vous pouvez le constater, l'itérateur de la globale s'intègre parfaitement au modèle itératif de Python, ce qui permet d'obtenir un code simple.

```
curkey = 'Tokyo'
G = nx.Graph()
addNodes(curkey, G, 1)
```

Ensuite, addNodes() est appelé en prenant le mot "Tokyo" comme clé. Cela nous donnera l'enchaînement des liens tracés à partir du mot "Tokyo". Il est montré dans la figure suivante.

Le réseau des mots cliqués à partir du mot "Tokyo" est ainsi représenté.

Conclusion

Les globales d'InterSystems IRIS sont une force avec laquelle il faut compter dans les cas où une base de données NoSQL est la base de données de choix. En outre, l'API native Python vous permet de manipuler les globales de manière naturelle à partir de vos programmes Python.

Nous vous encourageons à l'essayer. Par ailleurs, nous serions très heureux si vous pouviez écrire vos questions dans les commentaires de cet article.

#Globals #Graphique #Multi-model #Python #Vidéo #InterSystems IRIS #InterSystems IRIS for Health

---

URL de la source:https://fr.community.intersystems.com/post/acc%C3%A8s-aux-bases-de-donn%C3%A9es-nosql-avec-lapi-native-python