

REPUBLIQUE DU CAMEROUN

PAIX-TRAVAIL-PATRIE

MINISTERE DE L'ENSEIGNEMENT

SUPERIEURE

INSTITUT UNIVERSITAIRE DES

TECHNOLOGIES



REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

MINISTRY OF HIGHER EDUCATION

INSTITUT UNIVERSITAIRE DES

TECHNOLOGIES

EXPOSE DE CONCEPTS DE BASE DE DONNEES

THEME :
BASE DE DONNEES CLE-VALEUR

REDIGE PAR :

- ❖ **NGAKO MBAKOP BORIS DARYL (Chef du groupe 5C)**
- ❖ **KOMNA MELONG JEANNE**
- ❖ **MELI TIAYO YLONA**
- ❖ **SONKWA TCHIOH LOUIS MARCEL**

SUPERVISE PAR :

M. ARISTIDE MOUNGOLE

ANNEE ACAMDEMIQUE : 2023 - 2024

TABLE DE MATIERES

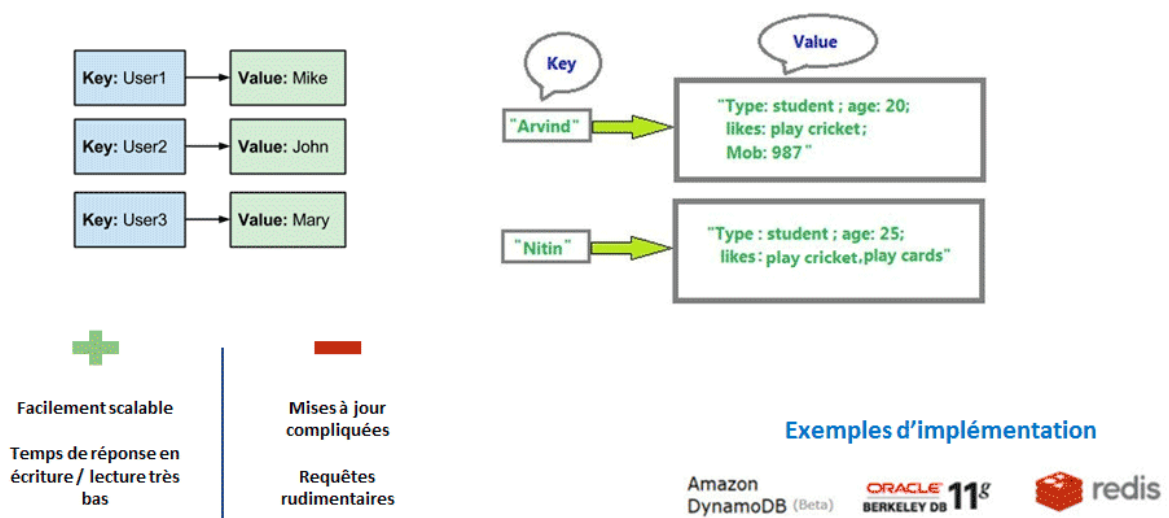
INTRODUCTION	1
I Qu'est-ce qu'une base de données clé-valeur ?	2
II Cas d'utilisation des bases de données clé-valeur	2
III Avantages des bases de données clé-valeur	4
IV Inconvénients des bases de données clé-valeur	4
V Quelques exemples de base de données clé-valeur	5
VI Comparaison avec d'autres types de base de données NoSQL	7
VII Etude d'une base de données clé-valeur : Cas de Redis	7
VII.1 C'est quoi Redis ?	7
VII.2 Types de données dans Redis.....	7
VII.3 Commandes dans Redis.....	8
VII.4 Cas pratique d'utilisation d'une base de données clé-valeur à l'instar de Redis : Gestion d'un panier d'achat	8
VII.4.1 Présentation du service uptash	8
VII.4.2 Création de l'application	9
VII.4.3 Fonctionnement de la base de données Redis.....	10
VIII CONCLUSION	11
IX RÉFÉRENCES BIBLIOGRAPHIQUES	A

INTRODUCTION

Le domaine informatique utilise des bases de données très variées. **Les modèles relationnels** sont particulièrement appréciés, car ils s'appuient sur des schémas rigides et des tables interconnectées. Mais il existe également d'autres modèles qui offrent des avantages particuliers et constituent donc une meilleure alternative dans des cas bien précis selon certaines circonstances : **base de données orientées document**, **base de données en colonnes** ou **base de données orientées graphe**. En outre, il existe par ailleurs ce que l'on appelle les **Key Value Databases** (bases de données clé-valeur) qui adoptent un modèle plus simple et flexible, où chaque donnée est stockée sous forme de pair clé-valeur. Dans ce contexte, un système tel que Redis apparaît comme une véritable référence dans le domaine témoignant ainsi de l'importance croissante de cette technologie.

I Qu'est-ce qu'une base de données clé-valeur ?

Une base de données orientées clé-valeur (BDOKV) est un type de système de gestion de base de données **NoSQL** qui stocke des données sous forme de **paires clé-valeur**. Dans ce modèle, chaque entrée dans la base de données est constituée d'une **clé unique** et d'une **valeur associée**. Les clés sont utilisées pour identifier les valeurs, permettant ainsi un accès rapide et efficace aux données. Et une **valeur** peut supporter différents formats, **des valeurs simples** comme des **chaînes** ou **des entiers**, mais aussi des **objets complexes** qui peuvent apparaître comme valeurs dans la base de données. **Des tuples**, à savoir des **groupes de valeurs**, peuvent aussi être appliqués.



II Cas d'utilisation des bases de données clé-valeur

On peut utiliser des systèmes de base de données clé-valeur comme base de données principale pour notre application ou pour répondre à des besoins spécifiques. Voici quelques exemples de cas d'utilisation de bases de données orientées clé-valeur :

Gestion du panier d'achat

Les clients qui emplissent leur panier et changent ensuite d'avis sur les items sélectionnés sont plus nombreux que ceux qui procèdent immédiatement au paiement. Pourquoi remplir une base de données relationnelle avec toutes ces données, alors qu'il existe une solution plus efficace et plus fiable ? **Un dépôt de paires clé-valeur** sera rapide pour enregistrer et obtenir des données simultanément. L'extensibilité de cette technologie est utile pendant les périodes des achats de Noël ou pendant les soldes et les promotions spéciales, car elles entraînent généralement une forte augmentation des ventes et un accroissement encore plus important du trafic sur le site web.

Session utilisateur

La plupart des sites web ou applications ouvre une session lorsqu'un utilisateur se connecte, puis ferme la session lorsque l'utilisateur se déconnecte ou lorsque la session expire. Pendant cette période, l'application stocke tous les attributs de la session utilisateur dans la mémoire principale ou dans une base de données. Les données de session peuvent inclure des informations sur le **profil d'utilisateur**, des **messages**, des **thèmes personnalisés**, des **recommandations**, des **promotions ciblées** et des **remises**.

Compter le nombre de vues d'un post

Les base de données clé-valeur sont particulièrement adaptées pour **compter le nombre de vues d'un post** de manière efficace.

III Avantages des bases de données clé-valeur

Les bases de données clé-valeur permettent des schémas de base de données **flexibles** et des **performances améliorées** à grande échelle pour certains cas d'utilisation. Les avantages du stockage clé-valeur sont :

- ❖ **Plus facile à déplacer** : L'**absence de langage de requêtes** signifie que la base de données peut être plus facilement migrée entre différents systèmes sans devoir modifier l'architecture.
- ❖ **Simplicité** : Les bases de données orientées clé valeur sont souvent plus simple à mettre en place et à utiliser que les bases de données relationnelles. Elle ne nécessite pas de définir de **schéma complexes** ou de **relations** entre les tables.
- ❖ **Flexibilité** : Comme la base de données clé-valeur n'exige ni n'impose aucun **schéma homogène**, il est possible de procéder à des modifications de la base de données lors de l'exploitation. Un nouveau champ peut être inséré tout en procédant simultanément à des actions sur d'autres entrées.
- ❖ **Performance** : Contrairement aux bases de données relationnelles, les bases de données clé-valeur n'ont pas à effectuer des **jointures de tables** gourmandes en ressources, ce qui les rend beaucoup plus rapides.

IV Inconvénients des bases de données clé-valeur

Les bases de données clé-valeur présentent quelques inconvénients comme pour tous les choix technologiques.

- ❖ **Sécurité** : Elles sont plus vulnérables aux attaques de type injection de code car elles permettent l'insertion de code dans les clés ce qui peut entraîner

des problèmes de sécurité graves si les clés ne sont pas correctement filtrées et vérifiées.

- ❖ **Les valeurs ne peuvent pas être filtrées par défaut :** La base de données voit les valeurs comme des **blobs** et ne peut donc pas conférer beaucoup de sens à ce qu'ils contiennent. Une requête renverra toutes les valeurs attribuées à une clé plutôt qu'un élément d'information spécifique.
- ❖ **Absence de requêtes complexes :** Étant donné que les bases de données clé-valeur ne prennent pas en charge les requêtes complexes, les développeurs doivent contourner ce problème dans le code. Les opérations relatives aux données se font principalement par le biais de termes de langage de requête simples tels que **get**, **put** et **delete**.
- ❖ **Mauvaise gestion du schéma :** La conception d'un stockage clé-valeur n'impose pas de schéma aux développeurs. Les équipes de développement doivent planifier de manière systématique le modèle de données afin d'éviter les problèmes à long terme.

V Quelques exemples de base de données clé-valeur

On distingue de nombreuses bases de données clé-valeur dont certaines qui en sortent du lot à savoir :

- ❖ **Redis :** Redis est connu comme un serveur de structures de données compatibles avec plusieurs **types de valeurs**. Au lieu d'associer une clé à une chaîne de caractère, la base de données **in-memory** peut associer des clés avec des valeurs plus complexes. Le SGBD supporte les clés contenant des **listes**, des **hachages**, des **chaînes de caractères** et des **ensembles**.



- ❖ **Amazon DynamoDB** : le SGBD propriétaire appartient à **Amazon Web Services** (AWS) et s'utilise également comme base de données orientée document.



- ❖ **Berkeley DB** : Ce SGBD est développé par **Oracle** et offre des interfaces pour divers langages de programmation.



VI Comparaison avec d'autres types de base de données NoSQL

Voici un tableau qui illustre la comparaison entre les bases de données clé valeur et d'autres SGBD NoSQL :

Types de base NoSQL	Utilisation principale	Exemple de base NoSQL	Modèle
Clé-valeur	Accès rapide	Redis	Clé-valeur
Document	Données semi-structurées	Mongo DB	JSON
Colonne	Analyse de grande données	Cassandra	Tables
Graphe	Relations complexes	Neo4j	Nœuds et arrêts

VII Etude d'une base de données clé-valeur : Cas de Redis

VII.1 C'est quoi Redis ?

Redis (Remote Dictionary Server) est une base de données **NoSQL** en mémoire, souvent utilisée principalement comme **cache d'application** ou base de données à **réponse rapide**. Elle est célèbre pour sa rapidité et sa polyvalence, grâce à son **stockage en mémoire** et sa **structure de données variée**.

VII.2 Types de données dans Redis

Bien qu'il s'agisse techniquement d'un **magasin clé/valeur**, Redis est un véritable serveur de structures de données qui prend en charge plusieurs types et structures de données, notamment :

- ❖ **Les Strings** qui sont des chaînes simples
- ❖ **Les Listes** ordonnées de chaînes

- ❖ **Les Hashs** qui sont des collections de paires champ-valeur, similaires à des objets ou des dictionnaires.
- ❖ **Les Sets** qui sont des ensembles uniques et non ordonnés.

VII.3 Commandes dans Redis

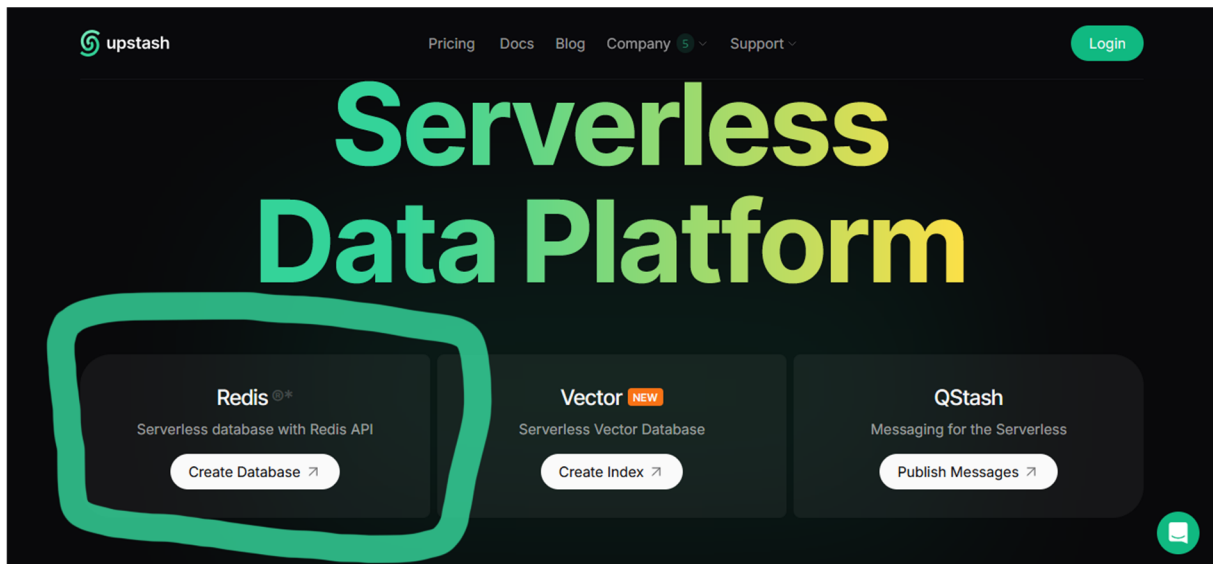
Voici quelques commandes utilisées avec Redis

Commandes	Description
SET key value	Définit une clé avec une valeur.
GET key	Récupère la valeur d'une clé.
DEL key	Supprime une ou plusieurs clés.
HSET key field value	Ajoute ou met à jour un champ dans une hash.
HGET key field	Récupère la valeur d'un champ dans une hash.

VII.4 Cas pratique d'utilisation d'une base de données clé-valeur à l'instar de Redis : Gestion d'un panier d'achat

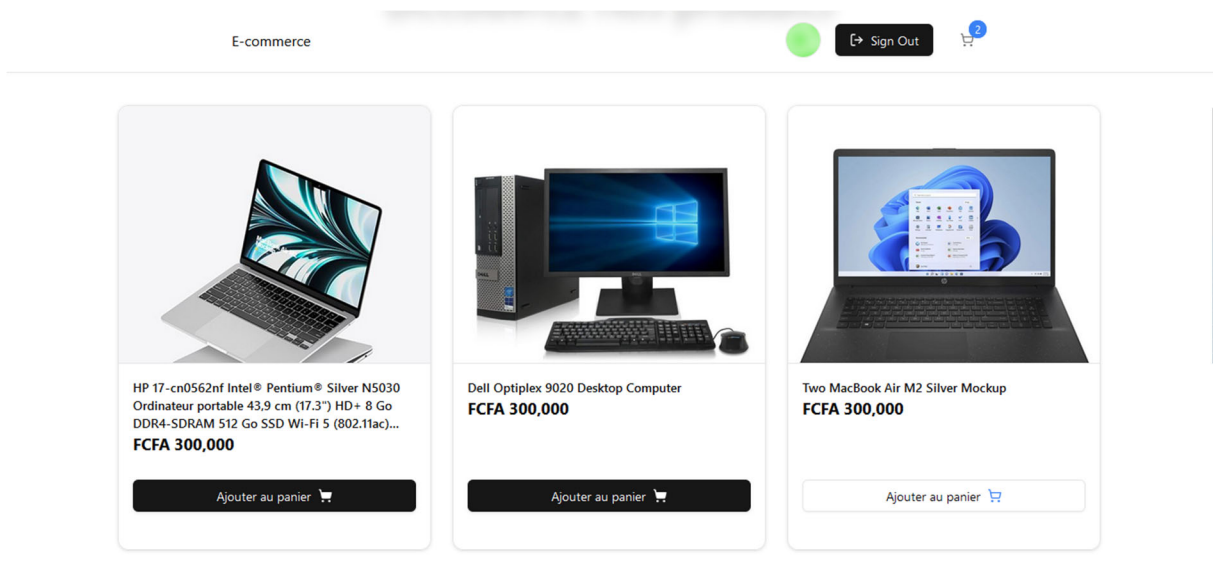
VII.4.1 Présentation du service uptash

Upstash est un fournisseur de services cloud conçu pour offrir une **base de données Redis en tant que service** (Redis-as-a-Service). Il est particulièrement adapté aux applications modernes grâce à sa tarification **serverless**, qui vous permet de ne payer que pour ce que vous utilisez, contrairement aux modèles traditionnels où vous devez maintenir une instance de serveur en permanence.



VII.4.2 Création de l'application

Nous avons développé une mini application **E-commerce** en NextJS fonctionnant avec **une base de données Redis** qui stocke **le panier d'achat** de chaque utilisateur.

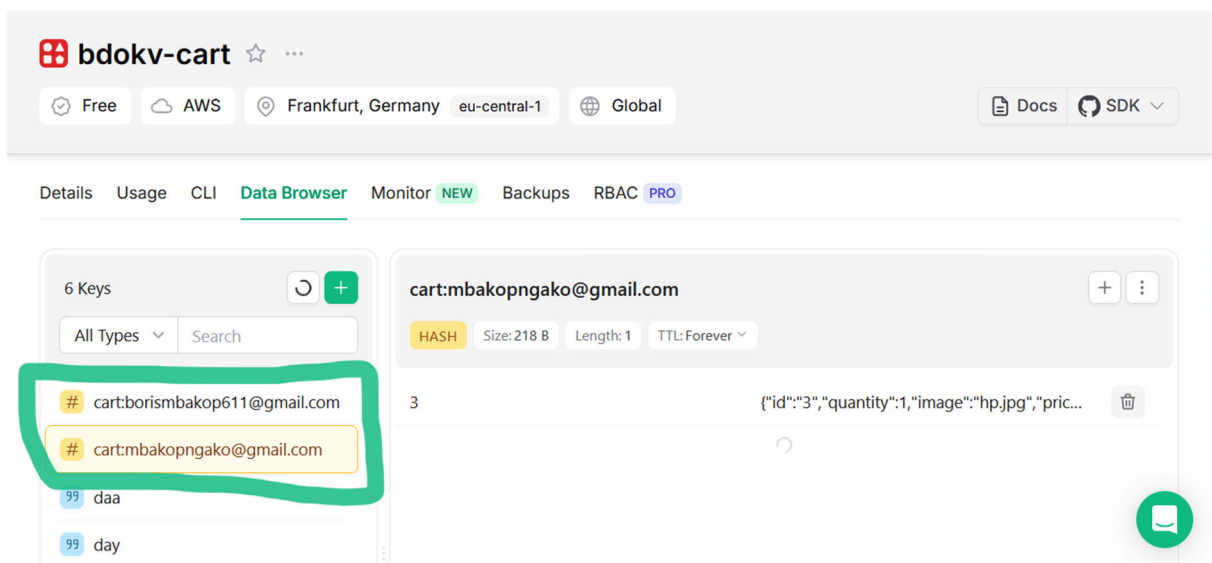


VII.4.3 Fonctionnement de la base de données Redis

On constate d'après l'image ci-dessous que la base de données Redis stocke effectivement le panier de chaque utilisateur. Chaque clé est composée :

- ❖ Du mot **cart** suivi de « : »
- ❖ Et de l'**adresse e-mail** de chaque utilisateur

Et le type de valeur de chaque clé est un **hash**.



VIII CONCLUSION

Les bases de données clé-valeur sont un type de base de données qui offrent **une grande flexibilité** et **une grande efficacité** pour les applications qui nécessitent un accès rapide et efficace aux données. Elles sont particulièrement adaptées pour les applications qui nécessitent un **grand volume de données** et **une faible latence**. Cependant elles ont également des limites qui doivent être prise en compte lors de la conception et de la mise en œuvre d'une application.

IX RÉFÉRENCES BIBLIOGRAPHIQUES

Liens bibliographiques :

<https://aws.amazon.com/fr/nosql/key-value/>

<https://www.lemagit.fr/conseil/NoSQL-lessentiel-sur-le-modele-cle-valeur>

<https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/key-value-store>