

Recherche de la base de données adéquate

Principes des bases de données :

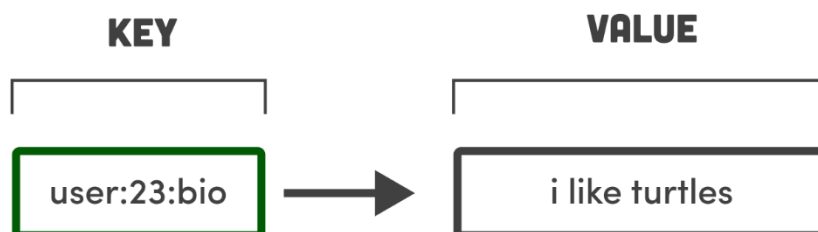
NoSQL databases :

1. Base de données Key-value :

Bases de données Key-value populaires : Redis, Memcached, Etc

Une base de données Key-value utilise une méthode simple Key-value pour stocker les données. Ces bases de données contiennent une chaîne simple (la clé) qui est toujours unique et un champ de données de taille arbitraire (la valeur). Elles sont faciles à concevoir et à mettre en œuvre.

1. KEY-VALUE



Pour redis par exemple nous pouvons lire et écrire des données avec une cmd.

```
○ ○ ○  
redis> SET user:23:bio "i like turtles"  
>> OK  
  
redis> GET user:23:bio  
>> "i like turtles"
```

Contrairement aux autres bases de données les données sont stockées dans la ram et non pas sur un disque dure ce qui rend la DB **rapide** mais **limite sa taille**.

Quels sont les cas d'utilisation des bases de données Key-value ?

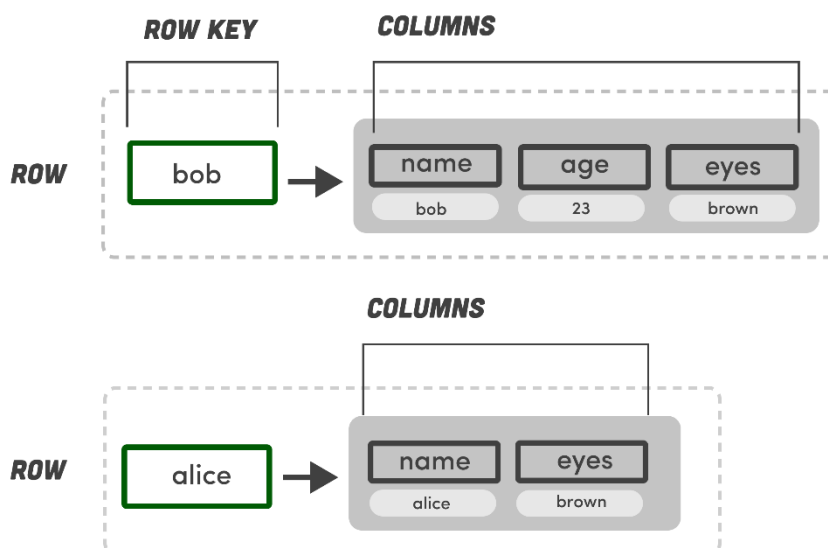
- Cache pour réduire la latence des données
- Pub/sub
- Pour mettre à jour ces données en temps réelle

2. Base de données Wide Column :

Bases de données Wide Column populaires : Cassandra, Apache HBase

Une base de données Wide Column est un type de base de données NoSQL dans laquelle les noms et le format des colonnes peuvent varier d'une ligne à l'autre, même au sein d'une même table. Les données étant stockées en colonnes, les requêtes portant sur une valeur particulière d'une colonne sont très rapides, car la colonne entière peut être chargée et recherchée rapidement. Des colonnes apparentées peuvent être modélisées comme faisant partie de la même famille de colonnes.

2. WIDE COLUMN



Quels sont les avantages d'une base de données Wide Column ?

Les avantages d'une base de données NoSQL Wide Column incluent la vitesse d'interrogation, l'évolutivité et un modèle de données flexible.

Quels sont les désavantages d'une base de données Wide Column ?

Quels sont les cas d'utilisation des bases de données Wide Column ?

- Log data
- Données de capteurs IoT (Internet of Things)

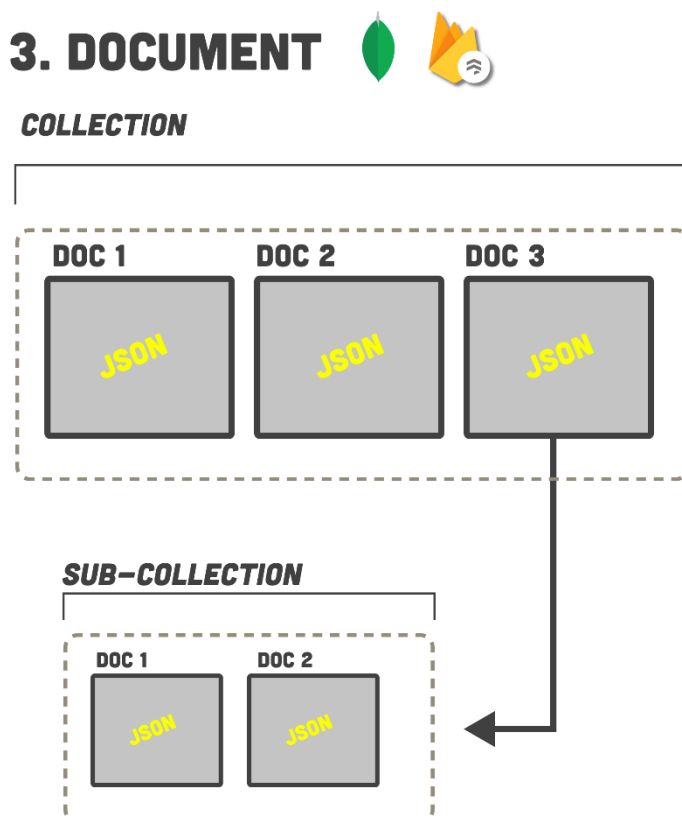
- Time-series data, telles que la surveillance de la température ou les données de transactions financières
- Attribute-based data, telles que les préférences des utilisateurs ou les caractéristiques des équipements
- Analyse en temps réel

3. Base de données Document Oriented :

Bases de données Document populaires : MongoDB, Firestore, CouchDB

Une base de données orientée documents ou NoSQL document store est un moyen moderne de stocker des données en documents JSON plutôt que de simples lignes et colonnes. Elle vous permet d'exprimer les données sous leur forme naturelle, comme elles sont censées l'être.

Ces documents peuvent être regroupés en collections pour former des systèmes de bases de données.



Chaque document est constitué d'un certain nombre de paires clé-valeur. Voici un exemple de document composé de 4 paires Key-value :

```
{  
  "ID" : "001",  
  "Book" : "Java: The Complete Reference",  
  "Genre" : "Reference work",  
  "Author" : "Herbert Schildt",  
}
```

L'utilisation de JSON permet aux développeurs de stocker et query (d'interroger) des données dans le même format de modèle de document qu'ils utilisent pour organiser le code de leur application.

Quels sont les avantages d'une base de données Document Oriented ?

- **Schema-less** : Il n'existe aucune restriction quant au format et à la structure du stockage des données. C'est une bonne chose pour conserver des données existantes à des volumes massifs et à différents états structurels, en particulier dans un système en transformation continue.
- Création et entretien plus rapides. Une maintenance minimale est requise une fois le document créé, ce qui peut être aussi simple que d'ajouter une fois votre objet complexe.
- Pas de clés étrangères. Grâce à l'absence de cette dynamique de relation, les documents peuvent être indépendants les uns des autres.
- Formats ouverts. Un processus de construction propre qui utilise XML, JSON et d'autres dérivés pour décrire les documents.
- Versioning intégré. La taille de vos documents peut augmenter, tout comme leur complexité. Le versionnage réduit les conflits.
- Très utile pour base de données que son sujet a des évolutions ou qui n'a pas un schéma défini.

Quels sont les désavantages d'une base de données Document Oriented ?

- Sécurité : Près de la moitié des applications web actuelles laissent échapper des données sensibles. Les propriétaires de bases de données NoSQL doivent donc prêter une attention particulière aux vulnérabilités des applications web.

Quels sont les cas d'utilisation des bases de données Document Oriented ?

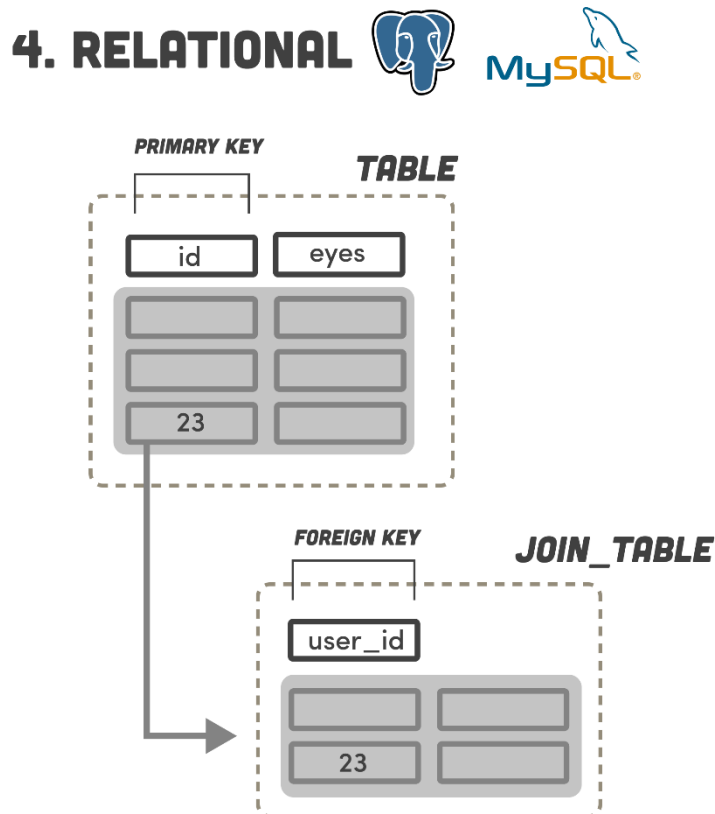
- Catalogue de produits
- Game
- Applications
- IoT
- Le choix idéal si on ne s'est pas encore comment les données seront structurées

SQL databases :

4. Base de données Relationnelles :

Bases de données Relational populaires : MySQL, MariaDB, Postgres, SQL Server, CockroachDB

Les bases de données relationnelles sont constituées d'un ensemble de tableaux. Au sein de ces tableaux, les données sont classées par catégorie. Chaque tableau comporte au moins une colonne correspondant à une catégorie. Chaque colonne comporte un certain nombre de données correspondant à cette catégorie.



Quels sont les avantages d'une base de données Relationnelles ?

- Une gestion simple des grands ensembles de données
- Un accès simple et efficace aux données enregistrées
- L'intégrité et la cohérence des données
- Le contrôle des accès pour les utilisateurs (sécurité et protection des données)
- Une disponibilité élevée

Quels sont les désavantages d'une base de données Relationnelles ?

- Un investissement de départ relativement plus coûteux (incluant les coûts supplémentaires pour le matériel)
- Plutôt moins efficace pour des volumes de données très volumineux
- Une vulnérabilité importante du fait de la centralisation des données

SQL et NoSQL : quelle technologie choisir ?

Choisir SQL :

- Besoin d'une base structurée et segmentée (l'essence des bases de données relationnelles)
- Type et validité des données très importante
- Besoin récurrent d'écriture et de modification de données sur des éléments spécifiques (SQL permet de modifier facilement des lignes spécifiques)
- Besoin de requête complexe

Choisir NoSql :

- Base de données sans schéma spécifique (Une structure non figée par exemple)
- Besoin de nombreuse requête de lecture, effectivement toutes les données nécessaires peuvent être récupéré d'un coup sans jointure particulière
- Grands ensembles de données (Big Data)
- Données distribuées (plusieurs sources)

Source :

<https://youtu.be/W2Z7fbCLSTw>