

## Amazon DynamoDB

Bases de données relationnelles ou non relationnelles					
	Relationnelles (SQL)				Non relationnelles
Stockage de données	Lignes et colonnes				Clé-valeur, document, graphique
Schémas	Fixes				Dynamiques
Interrogation	Utilise SQL				Axée sur un ensemble de documents
Évolutivité	Verticale				Horizontale
Exemple	ISBN	Titre	Auteur	Format	<pre>{   ISBN : 3111111223439,   Titre : « Withering Depths »,   Auteur : « Jackson, Mateo »,   Format : « Livre de poche » }</pre>
	3111111223439	Withering Depths	Jackson, Mateo	Livre de poche	
	312222223439	Wily Willy	Wang, Xiulan	E-book	

© 2012, Amazon Web Services, Inc. ou ses sociétés affiliées. Tous droits réservés.

Une **base de données relationnelle** (RDB) utilise des données **structurées, organisées par tables, enregistrements et colonnes**. Les RDB établissent une relation bien définie entre les tables de la base de données. Les RDB utilisent le langage SQL (Structured Query Language), application utilisateur standard qui offre une interface de programmation pour l'interaction avec une base de données. Les bases de données relationnelles peuvent avoir des difficultés à évoluer horizontalement ou à utiliser des données semi-structurées et peuvent également nécessiter de nombreuses jointures pour les données normalisées.

Une **base de données non relationnelle** est toute base de données qui ne suit pas le modèle relationnel fourni par les systèmes de gestion de base de données relationnelle traditionnels (RDBMS). Les bases de données non relationnelles rencontrent un succès croissant parce qu'elles ont été conçues pour faire face aux limitations des bases de données relationnelles dans le traitement des demandes de données structurées variables. Les bases de données non relationnelles effectuent une mise à l'échelle ascendante et sont compatibles avec les données non structurées et semi-structurées.

## Qu'est-ce qu'Amazon DynamoDB ?

Service de base de données NoSQL rapide et souple, adapté à toutes les échelles



**Amazon DynamoDB**

- Tables de base de données NoSQL
- Stockage virtuellement illimité
- Les éléments peuvent avoir différents attributs
- Requêtes à faible latence
- Débit de lecture/écriture pouvant être mis à l'échelle



© 2013, Amazon Web Services, Inc. ou ses sociétés affiliées. Tous droits réservés.

22

Amazon DynamoDB est un service de base de données NoSQL rapide et flexible, pour toutes les applications nécessitant une latence constante de quelques millisecondes, quelle que soit l'échelle.

Amazon gère toute l'infrastructure de données sous-jacente pour ce service et stocke les données de façon redondante dans différentes installations dans une région US native dans le cadre de son architecture tolérante aux pannes.

Avec DynamoDB, vous pouvez créer des tables et des éléments. Vous pouvez ajouter des éléments à une table. Le système partitionne automatiquement vos données et dispose d'un stockage de table pour répondre aux conditions de charge de travail. Il n'existe pratiquement aucune limite au nombre d'éléments que vous pouvez stocker sur une table. Par exemple, certains clients ont des tables de production qui comprennent des milliards d'éléments.

Un des avantages d'une base de données NoSQL est que des éléments d'une même table peuvent avoir différents attributs. Cela vous permet d'ajouter des attributs au fur et à mesure que votre application évolue. Vous pouvez stocker des éléments dont le format est récent à côté d'éléments au format ancien, dans une même table, sans avoir à réaliser de migrations de schéma.

Au fur et à mesure que la popularité de votre application grandit et que les utilisateurs interagissent avec elle, vos capacités de stockage peuvent croître avec les besoins de votre application. Toutes les données de DynamoDB sont stockées sur des disques SSD et son langage de requête simple permet des performances de requête cohérentes et à faible latence. En plus de la mise à l'échelle du stockage, DynamoDB vous permet d'allouer le débit en lecture ou en écriture dont vous avez besoin pour la table.

Au fur et à mesure que le nombre d'utilisateurs de l'application croît, les tables DynamoDB peuvent être mises à l'échelle pour gérer le nombre accru de demandes en lecture ou en écriture avec une allocation manuelle. Il est également possible d'activer la mise à l'échelle automatique afin que DynamoDB surveille la charge de la table et augmente ou diminue automatiquement le débit provisionné.

Certaines fonctions clés supplémentaires incluent des tables mondiales qui vous permettent de répliquer automatiquement sur vos choix de régions AWS, de chiffrement inactif et de durée de vie (TTL).

### Composants de base d'Amazon DynamoDB

- Les tables, les éléments et les attributs sont les composants de base de DynamoDB
- DynamoDB prend en charge deux sortes de clés primaires :  
clé de partition et clé de partition et de tri

© 2023, Amazon Web Services, Inc. ou ses sociétés affiliées. Tous droits réservés.

**Les tables, les éléments et les attributs** sont les composants de base de DynamoDB.

- Une table est un jeu de données contenant des éléments.
- Un élément est un groupe d'attributs identifiable de façon unique parmi tous les autres éléments.
- Un attribut est un élément de donnée fondamental, qui n'a pas besoin d'être décomposé plus avant.

**Exemple :**

Table : Utilisateurs

Élément 1 :

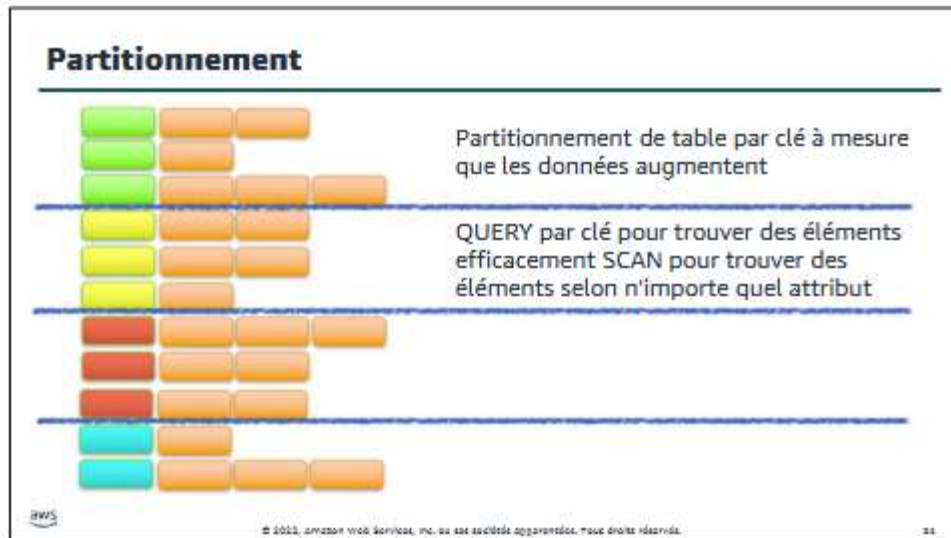
`{"UserId": "1", "Nom": "Ali", "Prenom": "Karim"}` Élément = tout le bloc

Attributs = UserId, Nom, Prenom

Valeurs = "1", "Ali", "Karim"

DynamoDB prend en charge deux sortes de clés principales :

La **clé de partition** est une simple clé primaire, composée d'un seul attribut appelé **clé sort**. La clé de partition et la clé de tri sont définies comme la **clé primaire composite**, qui est composée de deux attributs.

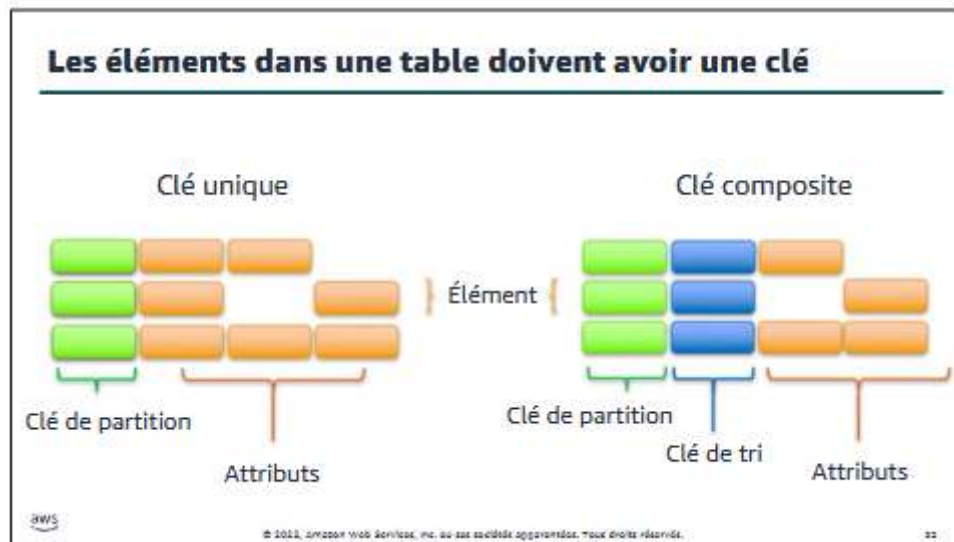


Au fur et mesure de la croissance des données, les données des tables sont partitionnées et indexées selon la clé principale.

Deux méthodes différentes permettent de récupérer des données à partir d'une table DynamoDB :

- Dans la première méthode, l'opération query tire parti du partitionnement afin de localiser efficacement des éléments en utilisant la clé principale.
- La seconde méthode consiste à utiliser une opération scan : celle-ci permet de localiser des éléments dans la table en faisant correspondre les conditions à des attributs non principaux. La seconde méthode permet de localiser des éléments au moyen d'autres attributs. Cependant, l'opération est moins efficace, car DynamoDB analyse alors tous les éléments de la table pour trouver ceux correspondants à vos critères.

À des fins d'accessibilité: le partitionnement permet à de larges tables d'être numérisées et interrogées rapidement. Partitionnement de table par clé à mesure que les données augmentent QUERY par clé pour trouver des éléments selon n'importe quel attribut.



Pour utiliser au mieux les opérations de requête et de DynamoDB, il est important de réfléchir à la clé que vous utilisez pour identifier avec précision des éléments dans la table DynamoDB. Vous pouvez définir une simple clé principale basée sur un seul attribut des valeurs de données avec une distribution uniforme, tel que le GUID (Globally Unique Identifier) ou d'autres identifiants aléatoires.

Par exemple, si vous devez modéliser une table avec des produits, vous pouvez utiliser certains attributs comme l'ID du produit.

Vous pouvez également renseigner une **clé composite**, constituée d'une **clé de partition** et d'une **clé secondaire**. Dans cet exemple, si votre table contenait des livres, vous pourriez combiner l'auteur et le titre pour identifier avec précision les éléments de la table. Cette méthode peut être utile si vous prévoyez de consulter fréquemment des livres par auteur, car vous pourriez alors utiliser une requête.

À des fins d'accessibilité: les deux différents types de clés. Une **clé unique** signifie que les données sont identifiées par **un élément** dans les données qui identifie chaque registre de façon **unique**.

Une **clé composite** est composée d'une **clé de partition** et d'une **deuxième clé** qui peut être utilisée pour trier les données.

