Les bases de données NoSQL (Not Only SQL) sont une famille de systèmes de gestion de bases de données qui ont été créées à la fin des années 2000 sous l'impulsion des grandes entreprises du web (meeting San Francisco). Les systèmes NoSQL ont été développés par ces entreprises afin de pouvoir traiter un nombre plus important de données, là où le système de bases de données relationnelles montrait ses limites. Les bases NoSQL sont donc principalement utilisées par les entreprises collectant ou traitant un grand nombre de données, ce qu'on appelle le big data. Les plus connues sont BigTable (Google), Dynamo (Amazon), Voldemort (Linkedin), Cassandra puis HBase (Facebook).

Pouvoir gérer plus efficacement un plus grand nombre de données permet notamment à ces grands sites internet de pouvoir rester fonctionnels et performants même dans le cas de millions de connexions simultanées. Pour faire court, les bases de données traditionnelles utilisent ce qu'on appelle le système ACID. Le système ACID garde un historique détaillé, dans un ordre précis de toutes les modifications apportées aux données par les utilisateurs. Donc, dans le cas de nombreuses modificatons simultanées, le processus de traitement est beaucoup plus long, car toutes les modifs doivent être remises dans l'ordre et historisées. Les bases NoSQL n'utilisent pas ce système, ce qui permet égalemet de mettre en place ce qu'on appelle un environnement distribué, qui consiste à faire des répliques de données sur d'autres serveurs ailleurs dans le monde afin d'accéder plus rapidement aux données.

Mais les bases NoSQL répondent à un autre système qu'ACID, qu'on appelle CAP (Consistency, Availability, Partition Tolerance = Cohérence, Disponibilité, Partionnement).

Cohérence : Impose que la base de données doit être cohérente après chaque opération. Ce qui signifie que toutes les applications ou « clients » accèdent à la même donnée (valeur) à un instant T, et que ceux-ci sont cohérents entre eux.

Disponibilité : Impose que le système soit toujours disponible, et que même en cas de panne le système ne connaisse aucun arrêt de service, ce qui garantit la continuité de service.

Partitionnement : Permet au système de continuer à fonctionner même en cas de problème de communication entre les serveurs (ex : problème de réseau). Les serveurs sont partitionnés en plusieurs groupes qui ne pourront pas communiquer entre eux.

On parle de théorème CAP, car une base de données NoSQL ne peut répondre au maximum qu'à deux des 3 exigences CAP, elles sont donc divisées en 3 couples répondant chacun à deux exigences.

- Le couple AC (Cohérence, Disponibilité) : Le site est basé sur un ensemble de machines unique, l'information est toujours disponible et cohérente, mais en cas de problème de communication ou de réseau, l'information n'est plus disponible. Le site est donc inaccessible car il ne prend pas en charge le partitionnement. (Oracle RAC, Cluster MSSQL)

- Le couple PC (Cohérence, Partition) : Le site est basé sur un système distribué, en cas de problème de communication ou de réseau, on peut toujours aller sur le site, les données restent cohérentes, mais certaines données (et donc certaines pages) peuvent être inacessibles. (MongoDB, HBase, Redis)

- Le couple AP (Disponibilité, Partition) : Le site est aussi basé sur un système distribué, le site et toutes les pages sont accessibles en permanence, mais en cas de problème de réseau, certaines informations seront inexactes. (CouchDB, Cassandra, DynamoDB)

Elles se différencient aussi par leur souplesse, que ce soit sur le langage utilisé, elles n'utilisent pas SQL, mais peuvent utiliser plusieurs langages, que par leur structure, qui n'est pas définie dans un schéma, mais qui est dynamique et peut être adaptée selon la situation. Ce qui nous donne plusieurs types de bases NoSQL, on en distingue 4 principaux.

- BD orientée clé-valeur : Ce système de BD NoSQL est souvent comparé à un dictionnaire. La clé utilisée est unique et la valeur contient en fait un ensemble de valeurs regroupant toutes les informations liées à cette clé, donc un peu comme toutes les définitions d'un mot. La clé utilisée peut être attribuée ou générée aléatoirement lors de l'enregistrement des valeurs. Elles sont adaptées à la collecte d'événements (jeux en ligne), de traces (mesures d'audience) ou la gestion de profils utilisateurs (e-commerce). SGBD : Voldemort, Redis

- BD orientée colonnes : Ce système ressemble au modèle relationnel, car les données sont stockées dans un tableau, la différence étant que les valeurs sont stockées par colonnes, pas par lignes. La clé représente le nom de la ligne unique du tableau, et la valeur est le contenu de la colonne. La gestion des données est donc plus dynamique car dans un tableau il y a en général plus de données dans une colonne que dans une ligne. SGBD: Cassandra, HBase, Big Table.

- BD orientée documents : Les valeurs sont organisées dans des ensembles de documents écrits en JSON ou en XML. Il n'y a donc pas de contrainte de structure, mais le contenu du document est déjà formaté, elles ont donc un format de représentation à respecter. SGBD : CouchDB, SimpleDB, MongoDB.

- BD orientée graphes : Elles sont représentées de manière graphique, sous formes de noeuds et de relations entre les propriétés qui leur sont rattachées. Elles sont adaptées pour modéliser des structures de données relationelles diverses, et sont utilisées pour traiter des données sociales ou financières. Elles sont utiles pour des structures de données particulièrement complexes. SGBD : Neo4J

En conclusion, ça ne veut pas dire que les bases relationnelles sont maintenant obsolètes. Les bases relationnelles sont plus adaptées au traitement de données complexes alors que les NoSQL sont plus adaptées à traiter une énorme quantité de données simples. Elles ont donc des rôles et des utilités différentes, aucune n'est là pour remplacer l'autre sur le long terme.