

## Neo4j

### Installer et utiliser Neo4j la base de données graphes

Neo4j est une base de données orientée graphes et développée en Java.

Créée en 2000 par Neo Technology, elle permet de représenter les données sous forme de nœuds reliés ensemble par un ensemble d'arcs.

Neo4j affiche ainsi de grandes performances pour les recherches sur de multiples relations entre objets.

Cette formation permettra de comprendre le fonctionnement de Neo4J, et de mettre en œuvre le stockage de données de type graphe.

#### Détails

- **Code** : DB-NEO
- **Durée** : 2 jours ( 14 heures )

#### Public

- Consultants
- Consultants informatiques
- Développeurs
- Ingénieurs

#### Pré-requis

- Posséder des connaissances des principes classiques des bases de données

#### Objectifs

- Comprendre le fonctionnement de Neo4J
- Mettre en œuvre Neo4j pour le stockage de données de type graphe

#### Programme

##### Introduction

- Présentation Neo4j, les différentes éditions, license
- Fonctionnalités, stockage des données sous forme de graphes
- CQL : Cypher Query Language
- Positionnement par rapport aux autres bases de données, apports de Neo4j
- L'analyse de données
- Cas d'usage

##### Installation et configuration

- Les différentes méthodes d'installation

##### Travaux pratiques

- Installation de Neo4J Community Edition
- Premiers pas avec l'interface web
- Création de données, requêtage
- Import de données

##### Cypher Query Language

- Syntaxe, description des relations avec CQL, les patterns
- Les clauses d'écriture : set, delete, remove, foreach
- Les clauses de lecture : match, optional match, where, count, case, ...
- Les fonctions : count, type, relationship, ...
- Principe de profondeur et de direction de relation dans une recherche
- Les listes et les projections maps
- Les algorithmes de Graphe

##### Travaux pratiques

- Création d'un graphe
- Requêtes de recherche
- Navigation dans le graphe

##### Exploitation

- Sauvegardes et restaurations
- Optimisation des transactions
- Indexation
- Client jmx
- Points de surveillance

##### Développement

- Description des APIs disponibles : .Net, Java, Javascript, Python
- Connexions, sessions et transactions
- Principe de causalité entre transactions
- La bibliothèque Apoc

##### Travaux pratiques

- Connexion et récupération de données provenant de Cassandra

##### Sécurité

- Principe et activation
- Paramétrage

##### Travaux pratiques

- Création d'un compte sécurisé

#### Modalités

- **Type d'action** :Acquisition des connaissances
- **Moyens de la formation** :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- **Modalités pédagogiques** :Exposés – Cas pratiques – Synthèse
- **Validation** :Exercices de validation – Attestation de stages