Neao4J : graphes et analyse

Référence : **PYCB018B** Durée : **2 jours** Certification : **Aucune**

CONNAISSANCES PREALABLES

• Connaissance des principes classiques des bases de données.

PROFIL DES STAGIAIRES

• Chefs de projet. • Gestionnaires de bases de données.

OBJECTIFS

• Comprendre le fonctionnement de Neo4j. • Savoir le mettre en oeuvre pour le stockage de données de type graphe.

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Bigdata

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

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 Enterprise Edition en cluster
- 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 pratique : connexion et récupération de données provenant de Cassandra

Sécurité

- Principe et activation
- Paramètrage
- Travaux pratique : création d'un compte sécurisé