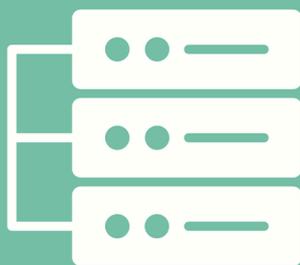


Formation
Mettre en oeuvre
un Data
Warehouse avec
Microsoft SQL
Server® 2012



contact@gate-training.com



+216 54 161 325

+216 73 362 100



12 Boulevard Abdelkader
Daghrir 4011 Hammam Sousse



gate-training.com

Description

Cette formation vous permettra d'optimiser l'intégration et les performances des applications .NET avec SQL Server 2012. Vous apprendrez à utiliser efficacement ADO.NET, Entity Framework et les procédures stockées pour améliorer la communication entre votre application et la base de données.

Grâce à des exercices pratiques, vous découvrirez les meilleures pratiques pour gérer les transactions, optimiser les requêtes et minimiser la charge sur SQL Server. Enfin, vous explorerez les outils de diagnostic et de profiling afin de détecter et corriger les problèmes de performance.

Objectifs de la formation

• Objectifs Pédagogiques

Cette formation vise à optimiser l'interaction entre les applications .NET et SQL Server 2012 en combinant bonnes pratiques de développement et techniques avancées. À l'issue de cette formation, vous serez capable de :

- ◆ Comprendre l'impact du modèle relationnel sur les performances des applications.
- ◆ Écrire des requêtes SQL optimisées et exploiter efficacement les index.



SUIVEZ NOUS

-
- ◆ Maîtriser ADO.NET, LINQ to SQL et Entity Framework pour des accès données performants.
 - ◆ Diagnostiquer et résoudre les goulots d'étranglement côté serveur et client.
 - ◆ Utiliser des stratégies asynchrones et des opérations batch pour les charges lourdes.

Public Cible

Profils concernés

Cette formation s'adresse aux :

- Développeurs .NET travaillant avec SQL Server 2012 (ou versions antérieures).
- Architectes logiciels souhaitant optimiser l'intégration données/métier.

Prérequis

- Expérience en développement .NET avec C#.
- Connaissances de base de SQL Server et du langage T-SQL.



SUIVEZ NOUS

Programme Détaillé

Module 1 : Fondamentaux de SQL Server 2012

- Modèle relationnel : Importance du schéma et des contraintes.
- Traitements ensemblistes : Pourquoi éviter les approches procédurales (curseurs).
- Impact du design de la base : Normalisation, clés étrangères, types de données.

Module 2 : Optimisation Côté Serveur

- Indexation :
 1. Choix des index cluster/non-cluster, index filtrés.
 2. Statistiques d'exécution et analyse des plans de requête.
- Analyse des performances :
 3. SQL Profiler et Extended Events pour tracer les requêtes lentes.
 4. Utilisation des DMVs (sys.dm_exec_query_stats, sys.dm_io_virtual_file_stats).
- Cas d'usage :
 5. Optimisation pour l'OLTP (transactions rapides) vs l'OLAP (requêtes analytiques).



SUIVEZ NOUS

Module 3 : ADO.NET : Performance et Bonnes Pratiques

- Choix des composants :
 1. DataReader vs DataSet : Quand utiliser l'un ou l'autre ?
 2. Pool de connexions : Configuration et gestion.
- Requêtes paramétrées :
 3. Prévention des injections SQL et réutilisation des plans d'exécution.
- Techniques avancées :
 4. MARS (Multiple Active Result Sets) : Gestion de plusieurs résultats.
 5. Commandes asynchrones : BeginExecuteReader, async/await.
 6. Table-Valued Parameters (TVP) : Passage de tableaux à des procédures stockées.

Module 4 : LINQ to SQL : Pièges et Optimisations

- Traduction des requêtes LINQ en SQL :
 1. Analyse avec SQL Server Profiler pour éviter les requêtes inefficaces.
 2. Utilisation de requêtes compilées (CompiledQuery.Compile).
- Gestion de la concurrence :
 3. Concurrence optimiste avec Timestamp.
- Limites et alternatives :
 4. Requêtes complexes : Quand basculer vers du SQL natif ou des vues.



SUIVEZ NOUS

Module 5 : Entity Framework : Performance et Scalabilité

- Choix du mode de chargement :
 1. Lazy Loading vs Eager Loading vs Explicit Loading.
- Optimisation des opérations CRUD :
 2. Utilisation de AddRange/RemoveRange pour les opérations batch.
 3. Désactivation du suivi des entités (AsNoTracking).
- Intégration de SQL natif :
 4. DbContext.Database.SqlQuery pour les requêtes complexes.
 5. Appel de procédures stockées et fonctions scalaires.
- Outils de diagnostic :
 6. Logging des requêtes générées (EF6+).
 7. Utilisation de MiniProfiler pour le débogage.

Module 6 : Intégration de Techniques Avancées

- Pagination efficace :
 1. OFFSET-FETCH en T-SQL vs Skip()/Take() en LINQ.
- Gestion des transactions :
 2. Transactions explicites avec TransactionScope.
- Cache côté client :
 3. Utilisation de MemoryCache pour réduire les appels redondants.



SUIVEZ NOUS

Travaux pratiques

TP 1 – Analyse des Plans d’Exécution

- Identifier une requête lente avec SQL Profiler et optimiser son plan.

TP 2 – Optimisation d’un Code ADO.NET

- Remplacer un DataSet par un DataReader pour un chargement 10x plus rapide.

TP 3 – Migration de LINQ to SQL vers EF Core

- Convertir une requête LINQ complexe en SQL natif pour éviter les N+1 queries.

TP 4 – Benchmarking de Performances

- Comparer les temps d’exécution d’une opération batch avec/sans AddRange.

Outils Utilisés

- Visual Studio : Débogage et profiling.
- SQL Server Management Studio (SSMS) : Analyse des index et requêtes.
- Entity Framework Profiler : Surveillance des appels ORM.
- JetBrains dotTrace : Profiling des performances .NET.



SUIVEZ NOUS

Scannez-moi!



*et accéder à l'ensemble
de nos formations!*



SUIVEZ NOUS