

Optimizasyon ile İlgili Konfigürasyon Parametreleri



Konfigürasyon Parametreleri

- **org.hibernate.cfg.AvailableSettings** isimli sınıfta Hibernate'in bütün konfigürasyon parametreleri açıklamaları ile birlikte mevcuttur

Konfigürasyon Parametreleri

- **hibernate.max_fetch_depth=0-3**
 - 1:1 ve M:1 ilişkiler için geçerlidir
 - Fetch edilen object graph'ın outer join derinliğini belirler
 - 0 devre dışı bırakır, bu durumda EAGER ilişkiler ayrı SELECT ifadeleri ile yüklenirler

Konfigürasyon Parametreleri

- **hibernate.default_batch_fetch_size=4,8,16**
 - Hibernate'in lazy ilişkileri batch fetching ile getirmesini sağlar
 - Entity veya ilişki düzeyinde **@BatchSize** annotasyonu ile fine tuning de yapılabilir

Konfigürasyon Parametreleri

- **hibernate.jdbc.fetch_size=100**
 - JDBC driver'da statement'ın fetch size değerini değiştirir
 - Eğer sorgu sonucu birden fazla kayıt dönülüyor ise bu değere göre tek seferde kaç satırın fetch edileceği belirlenir
 - Uygun değer uygulamanın davranışı incelenerek tespit edilebilir
 - Örneğin sorgu sonuçlarında max 20 değer dönülüyorsa 20, 500 değer dönülüyor ise 500 set etmek uygun olacaktır

Konfigürasyon Parametreleri

- **hibernate.order_inserts=true**
- **hibernate.order_updates=true**
- **hibernate.jdbc.batch_size=(5-30)**
- **hibernate.jdbc.batch_versioned_data=true**
 - Bir grup INSERT ve UPDATE ifadesinin tek seferde JDBC API üzerinden DB'ye gönderilmesi sağlanır
 - Hepsi birlikte kullanılmalıdır
 - INSERT işlemlerinin sıralanabilmesi için ID generation stratejisinin uygun olması gerekir (SEQUENCE veya UUID)
 - JDBC driver'ın da desteklemesi gerekir (Oracle desteklemez)
 - Second level cache'in de devre dışı bırakılması gerekir

Konfigürasyon Parametreleri

- DELETE ifadelerinde ise `CascadeType.DELETE` nedeni ile JDBC batch kabiliyeti çok efektif olmamaktadır
- JDBC batch işlemi tablo düzeyinde oluşturulur
- Farklı tablolardan ifadeler aynı batch içerisine konamaz
- Cascade özelliği nedeni ile DELETE ifadelerinin kendi aralarında gruplanarak sıralanması mümkün olmamaktadır

Konfigürasyon Parametreleri

- Bunu aşmak için Cascade özelliğini göz ardı edip Hibernate ile silme işlemini her bir child entity için explicit yapıp, ardından flush yapılabilir
- Bu durumda DELETE ifadeleri uygun biçimde gruplanarak sıralı biçimde topluca çalıştırılacaktır

Konfigürasyon Parametreleri

- **javax.persistence.validation.mode=NONE**
 - INSERT, DELETE ve UPDATE'ler öncesi yapılan bean validation devre dışı bırakılabilir

Konfigürasyon Parametreleri

- **hibernate.show_sql=false**
- **hibernate.format_sql=false**
- **hibernate.use_sql_comments=false**
 - HQL ve SQL logları devre dışı bırakılabilir

Konfigürasyon Parametreleri

- **hibernate.generate_statistics=false**
 - Hibernate ile gerçekleştirilen persistence işlemleri ve sorgularla ilgili SessionFactory düzeyindeki istatistik üretimi devre dışı bırakılabilir
 - Default durumda devre dışıdır

Konfigürasyon Parametreleri

- **hibernate.bytecode.use_reflection_optimizer=true**
 - -D ile JVM sistem property'si olarak verilmelidir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

