24. StatisticsHibernate can gather all sorts of statistics which can help you get a better insight into what Hibernate does behind the scenes. By default, the statistics are not collected because this incurs an additional processing and memory overhead. To instruct Hibernate to start collecting statistics, you need to set the hibernate.generate\_statistics configuration property to true: <property
name="hibernate.generate\_statistics"
value="true"
/> 24.1. org.hibernate.stat.Statistics methods The Hibernate statistics are made available via the Statistics interface which exposes the following methods: 24.1.1. General statistics methods isStatisticsEnabled Are statistics enabled? setStatisticsEnabled(boolean b) Enable statistics based on the provided parameter. clear Reset all statistics. logSummary Print a summary of the current statistics into the application log. getStartTime The milliseconds (JVM standard currentTimeMillis()) since the initial creation of this Statistics instance or the last time clear() was called. 24.1.2. Aggregated statistics methods getQueries Get executed query strings. The maximum number of queries tracked by the Hibernate statistics is given by the hibernate.statistics.query\_max\_size property. getEntityStatistics(String entityName) Find entity statistics for the given name. getCollectionStatistics(String role) Get collection statistics per role (collection name). getNaturalIdStatistics(String entityName) Get the Hibernate-specific natural id resolution statistics for the given entity. getQueryStatistics(String queryString) Get the statistics for the given query string (JPQL/HQL or native SQL). getDomainDataRegionStatistics(String regionName) Get the second-level cache statistics per domain data (entity, collection, natural-id) region. getQueryRegionStatistics(String regionName) Get the second-level cache statistics per query region. getCacheRegionStatistics(String regionName) Get statistics for either a domain-data or query-result region (this method checks both, preferring domain data region if one exists). 24.1.3. SessionFactory statistics methods getEntityNames Get the names of all entities configured with the current SessionFactory. getCollectionRoleNames Get the names of all collection roles configured with the current SessionFactory. 24.1.4. Session statistics methods getSessionCloseCount Global number of sessions that got closed. getSessionOpenCount Global number of sessions that got opened. getFlushCount Get the global number of flush operations executed (either manual or automatic). 24.1.5. JDBC statistics methods getPrepareStatementCount The number of JDBC prepared statements that were acquired by Hibernate. getCloseStatementCount The number of JDBC prepared statements that were released by Hibernate. getConnectCount Get the global number of connections acquired by the Hibernate sessions (the actual number of connections used may be much smaller depending whether you use a connection pool or not). 24.1.6. Transaction statistics methods getSuccessfulTransactionCount The number of transactions that completed successfully. getTransactionCount The number of transactions we know to have completed. 24.1.7. Concurrency Control statistics methods getOptimisticFailureCount The number of Hibernate StaleObjectStateExceptions or Jakarta Persistence OptimisticEntityLockExceptions that occurred. 24.1.8. Entity statistics methods getEntityDeleteCount Get the global number of entity deletes. getEntityInsertCount Get the global number of entity inserts. getEntityLoadCount Get the global number of entity loads. getEntityFetchCount Get the global number of entity fetches. getEntityUpdateCount Get the global number of entity updates. 24.1.9. Collection statistics methods getCollectionLoadCount Global number of collections that were loaded. getCollectionFetchCount Global number of collections that were fetched. getCollectionUpdateCount Global number of collections that were updated. getCollectionRemoveCount Global number of collections that were removed. getCollectionRecreateCount Global number of collections that were recreated. 24.1.10. Query statistics methods getQueryExecutionCount Get the global number of executed queries. getQueryExecutionMaxTime Get the time in milliseconds of the slowest query. getQueryExecutionMaxTimeQueryString Get the query string for the slowest query. getQueryPlanCacheHitCount Get the global number of query plans successfully retrieved from cache. getQueryPlanCacheMissCount Get the global number of query plans lookups not found in cache. 24.1.11. Natural id statistics methods getNaturalIdQueryExecutionCount Get the global number of natural id queries executed against the database. getNaturalIdQueryExecutionMaxTime Get the global maximum query time for natural id queries executed against the database. getNaturalIdQueryExecutionMaxTimeRegion Get the region for the maximum natural id query time. getNaturalIdQueryExecutionMaxTimeEntity Get the entity for the maximum natural id query time. 24.1.12. Second-level cache statistics methods getSecondLevelCacheRegionNames Get all second-level domain data cache region names. getSecondLevelCacheHitCount Global number of cacheable entities/collections successfully retrieved from the cache. getSecondLevelCacheMissCount Global number of cacheable entities/collections not found in the cache and loaded from the database. getSecondLevelCachePutCount Global number of cacheable entities/collections put in the cache. Second-level cache natural id statistics methods getNaturalIdCacheHitCount Get the global number of cached natural id lookups successfully retrieved from cache. getNaturalIdCacheMissCount Get the global number of cached natural id lookups not found in cache. getNaturalIdCachePutCount Get the global number of cacheable natural id lookups put in cache. Second-level cache query statistics methods getQueryCacheHitCount Get the global number of cached queries successfully retrieved from cache. getQueryCacheMissCount Get the global number of cached queries not found in cache. getQueryCachePutCount Get the global number of cacheable queries put in cache. Second-level cache timestamp statistics methods getUpdateTimestampsCacheHitCount Get the global number of timestamps successfully retrieved from cache. getUpdateTimestampsCacheMissCount Get the global number of timestamp requests that were not found in the cache. getUpdateTimestampsCachePutCount Get the global number of timestamps put in cache. 24.2. Query statistics max size Traditionally, Hibernate stored all executed queries when statistics were enabled. However, this was a very bad default since, if your application runs millions of different queries, you’d risk running out of memory. Therefore, to restrict the number of queries the Hibernate statistics can hold, the hibernate.statistics.query\_max\_size property was added. By default, the maximum number of queries retained is 5000, but you can increase this value via the hibernate.statistics.query\_max\_size property. So, if your application makes heavy use of the Jakarta Persistence Criteria API or if you simply have a very large number of queries, you might want to raise the maximum number of queries that are being stored by the Statistics instance. If the maximum number of queries has been reached, Hibernate uses a Least recently used (LRU) policy to make room for new query entries. 24.3. Query plan cache statistics Every entity query, be it JPQL/HQL or Criteria API, is compiled to an AST (Abstract Syntax Tree), and this process is resource-intensive. To speed up the entity query executions, Hibernate offers a query plan cache so that compiled plans can be reused. To monitor the query plan cache you have the following statistics. 24.3.1. Query plan cache global statistics The Statistics instance provides two global counters which can give you an overall picture of the query plan cache effectiveness. getQueryPlanCacheHitCount getQueryPlanCacheMissCount If the hit count is high and the miss count is low, then the query plan cache is effective, and the vast majority of entity queries are served from the query plan cache, rather than being compiled over and over again. 24.3.2. Query plan cache query-level statistics The QueryStatistics instance, which you can get via the getQueryStatistics(String queryString) method of the Statistics object, stores the following query plan cache metrics: getPlanCacheHitCount The number of query plans successfully fetched from the cache. getQueryPlanCacheMissCount The number of query plans not fetched from the cache. getPlanCompilationTotalMicroseconds The overall time spent to compile the plan for this particular query.