

# Performance Efficiency Weakness Descriptions

This document presents descriptions of the 15 weaknesses contained in the CISQ Automated Quality Characteristic Measure for Performance Efficiency. These descriptions have been simplified from their description in the published OMG® specification that used formalisms from other OMG meta-models to specify the weaknesses for representation in machine-processable XMI notation. The tables below present each weakness with its unique CISQ identifier, a brief descriptive name, and a fuller description of the weakness presented as a recommendation for remediation.

## Performance Efficiency Weaknesses

a measure of the extent to which software contains weaknesses that can degrade a system's performance or cause excessive use of processor, memory, or other resources.

CISQ identifier	Descriptor	Remediation
ASCPem-PRF-1	Static block initialization	Remove instances where a variable, field, member, etc. is initialized in a static block of code
ASCPem-PRF-2	Immutable text data	Remove instances where a function, method, procedure, stored procedure, sub-routine, etc., creates immutable text data via a string concatenation (which could be avoided by using text buffer instead)
ASCPem-PRF-3	Static data outside of singleton class	Remove instances where a static field or member is declared as static but its parent class is not a singleton class; it does not account for final static fields or members
ASCPem-PRF-4	Complex read/write access	Remove instances where a very large table, that is, whose number of rows exceeds a threshold value (default is 1,000,000 rows), is accessed by a SQL statement with too many joins (default threshold value for the maximum number of joins is 5), and too many sub-queries (default threshold value for the maximum number of sub-queries is 3).
ASCPem-PRF-5	Incorrect indices	Remove instances where the syntax of the SQL SELECT statement and the index configuration of the SQL table or SQL view causes the DBMS to run sequential searches
ASCPem-PRF-6	Excessive number of indices on large tables	Remove instance where a very large table, that is, whose number of rows exceeds a threshold value (default is 1,000,000 rows), has too many indices (default threshold value for the maximum number of indices is 3)

<b>ASCPem-PRF-7</b>	Excessively large indices on large tables	Remove instances where a very large table, that is, whose number of rows exceeds a threshold value (default is 1,000,000 rows), has an index whose size is too large (default threshold value for the index range is 10)
<b>ASCPem-PRF-8</b>	Resource-consuming operation in loop	Remove instances where an operation causing consumption of platform resource (messaging, lock, file, stream, directory, etc.) is directly or indirectly called within a loop body or within a loop condition
<b>ASCPem-PRF-9</b>	Excessive data queries in non-stored procedure	Remove instances where a server-side non-stored procedure contains too many data queries (default value for the maximum number of data queries is 5)
<b>ASCPem-PRF-10</b>	Excessive data queries in client-side code	Remove instances where a client-side function, method, sub-routine, etc., contains too many data queries (default value for the maximum number of data queries is 2).
<b>ASCPem-PRF-11</b>	Data access circumventing data manager	Remove instances where a function, method, procedure, stored procedure, sub-routine, etc. accesses data without going a dedicated data manager component (as identified in the vetted data access component list), thus circumventing the authorized data access procedures
<b>ASCPem-PRF-12</b>	Excessively large data element	Remove instances where a variable, field, member, etc., is an aggregate of too many (non-primitive) data types (default value for the maximum number of aggregated non-primitive types is 5)
<b>ASCPem-PRF-13</b>	Data access not using connection pool	Remove instances where a function, method, procedure, stored procedure, sub-routine, etc. executes a data resource management action without using a connection pooling capability (the usage of a connection pooling capability is technology dependent; for example, connection pooling is disabled with the addition of 'Pooling=false' to the connection string with ADO.NET or the value of a 'com.sun.jndi.ldap.connect.pool' environment parameter in Java)
<b>ASCPem-PRF-14</b>	Unreleased memory	Remove instances where a memory resource is explicitly allocated to a variable, field, member, etc. which is used throughout the application, but is never released.
<b>ASCPem-PRF-15</b>	Unreleased data	Remove instances where a method references an object, without ever de-referencing it