

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#1 - 09/10/2026 06:50 PM - Greg Shah

Proposed Gap Marking Support Levels: Record Phrase Options and Static Query Options

Purpose

Gap marking has been added in branch 11747a revision 16739 for two Code Analytics reports which previously carried no support level columns: **Record Phrase Options** and **Static Query Options (Except Record Phrases)**. This document lists every proposed conversion and runtime support level so the team can correct them before they are relied on for gap analysis.

The levels themselves were derived by reading the FWD conversion rules and runtime source. That is good enough to distinguish "clearly implemented" from "clearly absent", but it cannot judge compatibility depth.

See [Gap Analysis](#) for the meaning of each support level constant, and #11746 for the project that motivated this work.

Please review these and report any changes that are needed.

What was changed

- rules/gaps/database.rules - new addRecordPhraseOptions() and addStaticQueryOptions() functions holding the levels below.
- rules/gaps/gap_analysis_marking.xml - two new marking rules which apply those maps.
- rules/reports/profile.rpt - supportLvlExpr enabled on both reports.
- rules/include/common-progress.rules - static_query_options() now excludes KW_QUERY. That is the node the DEFINE_QUERY statement is built around rather than a query option, so it should never have been reported as one; in an OPEN QUERY the same keyword is hidden by the parser and never reached the predicate.

The new static query marking rule is deliberately placed **after** the existing inner block options marking. The FOR, DO PRESELECT and REPEAT PRESELECT forms of BREAK, BY, COLLATE and QUERY-TUNING therefore keep the levels they already had in rules/gaps/control_flow.rules, and no existing report changes its numbers.

Record phrase options

These are the direct children of a RECORD_PHRASE node, excluding the record reference itself and the record limit literal. The set is closed by the grammar.

Option	Token	Proposed CVT	Proposed RT	Basis
FIELDS()	KW_FIELD	Full	Full	field list honored by RecordBuffer (inclusive form)
EXCEPT()	KW_EXCEPT	Full	Full	field list honored by

				RecordBuffer (exclusive form)
[LEFT] OUTER JOIN	KW_OUT_JOIN	Full	Full	converted via outer_join_location; contiguous outer joins route through PreselectQuery
OF	KW_OF	Full	Full	natural join converted to the query peer ("inverse") parameter
WHERE	KW_WHERE	Full	Full	clause contents are marked separately
USE-INDEX	KW_USE_IDX	Full	Full	drives index selection; presort queries emit the matching ORDER BY
USING	KW_USING	Full	Full	-
SHARE-LOCK	KW_SH_LOCK	Full	Full	converted to LockType.SHARE; LockManager tracks multi-context share locks
EXCLUSIVE-LOCK	KW_EXC_LOCK	Full	Full	converted to LockType.EXCLUSIVE
NO-LOCK	KW_NO_LOCK	Full	Full	converted to LockType.NONE
NO-WAIT	KW_NO_WAIT	Full	Full	folded into the lock type as LockType.*_NO_WAIT
NO-PREFETCH	KW_NO_PRE	Full	Full (R)	prefetch behavior is implicit in FWD; matches the existing attribute marking
NO-ERROR	KW_NO_ERROR	Full	Full	-
TABLE-SCAN	KW_TAB_SCAN	None	None	no conversion or runtime handling exists; grammar comment says the option is untested

Static query options

Options of OPEN QUERY and DEFINE QUERY, plus the QUERY-TUNING sub-options.

Option	Token	Proposed CVT	Proposed RT	Basis
INDEXED-REPOSITION	KW_IDX_REPO	Full	Full	annotated at conversion; honored by AbstractQuery.indexedReposition and Cursor.addResultLast
MAX-ROWS	KW_MAX_ROWS	None	None	hidden at conversion with an "Ignoring currently unsupported MAX-ROWS query option" warning and a TODO
SCROLLING	KW_SCROLLIN	Full	Full	scrolling queries are implemented by AbstractQuery
CACHE n	KW_CACHE	Full	Stubs	inferred from the Query CACHE attribute marking, whose getCache() / setCache() are stubs
RCODE-INFORMATION	KW_RCOD_INF	None	None	no conversion handling for the query option
QUERY-TUNING	KW_QRY_TUNE	None	None	Progress optimizer hint with no FWD analog; matches the existing inner block option marking

All 21 QUERY-TUNING sub-options are marked None / None on the same basis as the phrase itself:

ARRAY-MESSAGE, NO-ARRAY-MESSAGE, BIND-WHERE, NO-BIND-WHERE, CACHE-SIZE, BYTE, DEBUG, NO-DEBUG, SQL, EXTENDED, HINT, INDEX-HINT, NO-INDEX-HINT, JOIN-BY-SQLDB, NO-JOIN-BY-SQLDB, LOOKAHEAD, NO-LOOKAHEAD, ORDERED-JOIN, REVERSE-FROM, SEPARATE-CONNECTION, NO-SEPARATE-CONNECTION