Mimer 🕷

Optional SQL Feature Summary

The following table lists all optional features included in the SQL standard, from SQL-2003 to SQL-2016.

It also indicates which features that are currently supported by Mimer SQL.

Feature	Feature Name	Mimer SQL	SQL–2003	SQL–2008	SQL–2011	SQL–2016
B011	Embedded Ada NOTE-1)		√	√	√	√
B012	Embedded C NOTE-1)	√	√	√	√	√
B013	Embedded COBOL NOTE-1)	√ NOTE-2)	√	√	√	√
B014	Embedded Fortran NOTE-1)	√ NOTE-2)	√	√	√	√
B015	Embedded MUMPS NOTE-1)		√	√	√	√
B016	Embedded Pascal NOTE-1)		√	√	√	√
B017	Embedded PL/I NOTE-1)		√	√	√	√
B021	Direct SQL	√	√	√	√	√
B031	Basic dynamic SQL	√	√	√	√	√
B032	Extended dynamic SQL	√	√	√	√	√
B032–01	<describe input statement>	√	√	√	√	√
B033	Untyped SQL-invoked function arguments	√	√	√	√	√
B034	Dynamic specification of cursor attributes			√	√	√
B035	Non-extended descriptor names	√		√		
B041	Extensions to embedded SQL exception declarations		√	√	√	√
B051	Enhanced execution rights		√	√	√	√
B111	Module language Ada NOTE-1)		√	√	√	√
B112	Module language C NOTE-1)		√	√	√	√
B113	Module language COBOL NOTE-1)		√	√	√	√
B114	Module language Fortran NOTE-1)		√	√	√	√
B115	Module language MUMPS NOTE-1)		√	√	√	√
B116	Module language Pascal NOTE-1)		√	√	√	√
B117	Module language PL/I NOTE-1)		√	√	√	√
B121	Routine language Ada		√	√	√	√
B122	Routine language C		√	√	√	√
B123	Routine language COBOL		√	√	√	√
B124	Routine language Fortran		√	√	√	√
B125	Routine language MUMPS		√	√	√	√
B126	Routine language Pascal		√	√	√	√
B127	Routine language PL/I		√	√	√	√
B128	Routine language SQL	√			√	√
B211	Module language Ada: VARCHAR and NUMERIC support				√	√
B221	Routine language Ada: VARCHAR and NUMERIC support				√	√
F032	CASCADE drop behavior	√	√	√	√	√
F033	ALTER TABLE statement: DROP COLUMN clause	√	√	√	√	√
F034	Extended REVOKE statement	√	√	√	√	√
F034–01	REVOKE statement performed by other than the owner of a schema object	√	√	√	√	√
F034–02	REVOKE statement: GRANT OPTION FOR clause	√	√	√	√	√
F034–03	REVOKE statement to revoke a privilege that the grantee has WITH GRANT OPTION	√	√	√	√	√
F052	Intervals and datetime arithmetic	√	√	√	√	√
F053	OVERLAPS predicate	√	√	√	√	√

F054	**TIMESTAMP in DATE type precedence list**	√			√	√
F111	**Isolation levels other than SERIALIZABLE**	√	√	√	√	√
F111–01	READ UNCOMMITTED isolation level	√	√	√	√	√
F111–02	READ COMMITTED isolation level	√	√	√	√	√
F111–03	REPEATABLE READ isolation level	√	√	√	√	√
F121	**Basic diagnostics management**	√	√	√	√	√
F121–01	GET DIAGNOSTICS statement	√	√	√	√	√
F121–02	SET DIAGNOSTICS statement: DIAGNOSTICS SIZE clause	√	√	√	√	√
F122	**Enhanced diagnostics management**	√		√	√	√
F123	**All diagnostics**		√	√	√	√
F171	**Multiple schemas per user**	√	√	√	√	√
F191	**Referential delete actions**	√	√	√	√	√
F200	**TRUNCATE TABLE statement**			√	√	√
F202	**TRUNCATE TABLE: identity column restart option**			√	√	√
F222	**INSERT statement: DEFAULT VALUES clause**	√	√	√	√	√
F231	**Privilege tables**	√	√	√	√	√
F231–01	TABLE_PRIVILEGES view	√	√	√	√	√
F231–02	COLUMN_PRIVILEGES view	√	√	√	√	√
F231–03	USAGE_PRIVILEGES view	√	√	√	√	√
F251	**Domain support**	√	√	√	√	√
F262	**Extended CASE expression**		√	√	√	√
F263	**Comma-separated predicates in simple CASE expression**			√	√	√
F271	**Compound character literals**	√	√	√	√	√
F281	**LIKE enhancements**	√	√	√	√	√
F291	**UNIQUE predicate**	√	√	√	√	√
F301	**CORRESPONDING in query expressions**		√	√	√	√
F302	**INTERSECT table operator**	√	√	√	√	√
F302–01	INTERSECT DISTINCT table operator	√	√	√	√	√
F302–02	INTERSECT ALL table operator	√	√	√	√	√
F304	**EXCEPT ALL table operator**	√	√	√	√	√
F312	**MERGE statement**		√	√	√	√
F313	**Enhanced MERGE statement**			√	√	√
F321	**User authorization**		√	√	√	√
F341	**Usage tables**	√	√	√	√	√
F361	**Subprogram support**	√	√	√	√	√
F381	**Extended schema manipulation**	√	√	√	√	√
F381–01	ALTER TABLE statement: ALTER COLUMN clause	√	√	√	√	√
F381–02	ALTER TABLE statement: ADD CONSTRAINT clause	√	√	√	√	√
F381–03	ALTER TABLE statement: DROP CONSTRAINT clause	√	√	√	√	√
F382	**Alter column data type**	√		√	√	√
F383	**Set column not null clause**				√	√
F384	**Drop identity property clause**				√	√
F385	**Drop column generation expression clause**				√	√
F386	**Set identity column generation clause**				√	√
F391	**Long identifiers**	√	√	√	√	√
F392	**Unicode escapes in identifiers**	√	√	√	√	√
F393	**Unicode escapes in literals**	√	√	√	√	√
F394	**Optional normal form specification**			√	√	√
F401	**Extended joined table**	√	√	√	√	√
F401–01	NATURAL JOIN	√	√	√	√	√
F401–02	FULL OUTER JOIN	√	√	√	√	√
F401–04	CROSS JOIN	√	√	√	√	√
F402	**Named column joins for LOBs, arrays, and multisets**		√	√	√	√
F403	**Partitioned join tables**			√	√	√
F404	**Range variable for common column names**	√				√

F411	**Time zone specification**		√	√	√	√
F421	**National character**	√	√	√	√	√
F431	**Read-only scrollable cursors**	√	√	√	√	√
F431–01	FETCH with explicit NEXT	√	√	√	√	√
F431–02	FETCH FIRST	√	√	√	√	√
F431–03	FETCH LAST	√	√	√	√	√
F431–04	FETCH PRIOR	√	√	√	√	√
F431–05	FETCH ABSOLUTE	√	√	√	√	√
F431–06	FETCH RELATIVE	√	√	√	√	√
F441	**Extended set function support**	√	√	√	√	√
F442	**Mixed column references in set functions**	√	√	√	√	√
F451	**Character set definition**		√	√	√	√
F461	**Named character sets**		√	√	√	√
F491	**Constraint management**	√	√	√	√	√
F492	**Optional table constraint enforcement**			√	√	√
F502	**Enhanced documentation tables**		√	√	√	√
F502–01	SQL_SIZING_PROFILES view		√	√	√	√
F502–02	SQL_IMPLEMENTATION_INFO view		√	√	√	√
F502–03	SQL_PACKAGES view		√	√	√	√
F521	**Assertions**		√	√	√	√
F531	**Temporary tables**		√	√	√	√
F555	**Enhanced seconds precision**	√	√	√	√	√
F561	**Full value expressions**	√	√	√	√	√
F571	**Truth value tests**	√	√	√	√	√
F591	**Derived tables**	√	√	√	√	√
F611	**Indicator data types**		√	√	√	√
F641	**Row and table constructors**		√	√	√	√
F651	**Catalog name qualifiers**		√	√	√	√
F661	**Simple tables**	√	√	√	√	√
F671	**Subqueries in CHECK constraints**		√	√	√	√
F672	**Retrospective check constraints**	√	√	√	√	√
F673	**Reads sql data routine invocation in check constraints**	√				√
F690	**Collation support**	√		√	√	√
F691	**Collation and translation**		√			
F692	**Enhanced collation support**	√	√	√	√	√
F693	**SQL-session and client module collations**		√	√	√	√
F695	**Translation support**		√	√	√	√
F696	**Additional translation documentation**		√			
F701	**Referential update actions**	√	√	√	√	√
F711	**ALTER domain**		√	√	√	√
F721	**Deferrable constraints**	√	√	√	√	√
F731	**INSERT column privileges**	√	√	√	√	√
F741	**Referential MATCH types**		√	√	√	√
F751	**View CHECK enhancements**		√	√	√	√
F761	**Session management**		√	√	√	√
F762	**CURRENT_CATALOG**			√	√	√
F763	**CURRENT_SCHEMA**			√	√	√
F771	**Connection management**	√	√	√	√	√
F781	**Self-referencing operations**	√	√	√	√	√
F791	**Insensitive cursors**		√	√	√	√
F801	**Full set function**	√	√	√	√	√
F813	**Extended flagging** NOTE-3)		√	√	√	√
F821	**Local table references**		√	√	√	√
F831	**Full cursor update**		√	√	√	√
F831–01	Updateable scrollable cursors		√	√	√	√

ID	Feature					
F831–02	Updateable ordered cursors		√	√	√	√
F841	LIKE_REGEX predicate			√	√	√
F842	OCCURRENCES_REGEX predicate			√	√	√
F843	POSITION_REGEX predicate			√	√	√
F844	SUBSTRING_REGEX			√	√	√
F845	TRANSLATE_REGEX			√	√	√
F846	Octet support in regular expression operators			√	√	√
F847	Nonconstant regular expressions			√	√	√
F850	Top-level <order by clause> in <query expression>	√		√	√	√
F851	<order by clause> in subqueries	√		√	√	√
F852	Top-level <order by clause> in views	√		√	√	√
F855	Nested <order by clause> in <query expression>	√		√	√	√
F856	Nested <fetch first clause> in <query expression>	√		√	√	√
F857	Top-level <fetch first clause> in <query expression>	√		√	√	√
F858	<fetch first clause> in subqueries	√		√	√	√
F859	Top-level <fetch first clause> in views	√		√	√	√
F860	Dynamic <fetch first row count> in <fetch first clause>	√			√	√
F861	Top-level <result offset clause> in <query expression>	√			√	√
F862	<result offset clause> in subqueries	√			√	√
F863	Nested <result offset clause> in <query expression>	√			√	√
F864	Top-level <result offset clause> in views	√			√	√
F865	Dynamic <offset row count> in <result offset clause>	√			√	√
F866	FETCH FIRST clause: PERCENT option				√	√
F867	FETCH FIRST clause: WITH TIES option				√	√
P001	Stored modules	√	√	√	√	√
P002	Computational completeness	√	√	√	√	√
P003	Information Schema views	√	√	√	√	√
P004	Extended CASE statement	√	√	√	√	√
P005	Qualified SQL variable references	√	√	√	√	√
P006	Multiple assignment	√	√	√	√	√
P007	Enhanced diagnostics management		√	√		
P008	Comma-separated predicates in simple CASE statement		√	√		
R010	Row pattern recognition: FROM clause					√
R020	Row pattern recognition: WINDOW clause					√
R030	Row pattern recognition: full aggregate support					√
S023	Basic structured types		√	√	√	√
S024	Enhanced structured types		√	√	√	√
S025	Final structured types		√	√	√	√
S026	Self-referencing structured types		√	√	√	√
S027	Create method by specific method name		√	√	√	√
S028	Permutable UDT options list		√	√	√	√
S041	Basic reference types		√	√	√	√
S043	Enhanced reference types		√	√	√	√
S051	Create table of type		√	√	√	√
S071	SQL paths in function and type name resolution		√	√	√	√
S081	Subtables		√	√	√	√
S091	Basic array support		√	√	√	√
S091–01	Arrays of built-in data type		√	√	√	√
S091–02	Arrays of distinct types		√	√	√	√
S091–03	Array expressions		√	√	√	√
S092	Arrays of user-defined types		√	√	√	√
S094	Arrays of reference types		√	√	√	√
S095	Array constructors by query		√	√	√	√
S096	Optional array bounds		√	√	√	√
S097	Array element assignment		√	√	√	√

S098	**ARRAY_AGG**			√	√	√
S111	**ONLY in query expressions**		√	√	√	√
S151	**Type predicate**		√	√	√	√
S161	**Subtype treatment**		√	√	√	√
S162	**Subtype treatment for references**		√	√	√	√
S201	**SQL-invoked routines on arrays**		√	√	√	√
S201–01	Array parameters		√	√	√	√
S201–02	Array as result type of functions		√	√	√	√
S202	**SQL-invoked routines on multisets**		√	√	√	√
S211	**User-defined cast functions**		√	√	√	√
S231	**Structured type locators**		√	√	√	√
S232	**Array locators**		√	√	√	√
S233	**Multiset locators**		√	√	√	√
S241	**Transform functions**		√	√	√	√
S242	**Alter transform statement**		√	√	√	√
S251	**User-defined orderings**		√	√	√	√
S261	**Specific type method**		√	√	√	√
S271	**Basic multiset support**		√	√	√	√
S272	**Multisets of user-defined types**		√	√	√	√
S274	**Multisets of reference types**		√	√	√	√
S275	**Advanced multiset support**		√	√	√	√
S281	**Nested collection types**		√	√	√	√
S291	**Unique constraint on entire row**		√	√	√	√
S301	**Enhanced UNNEST**			√	√	√
S401	**Distinct types based on array types**				√	√
S402	**Distinct types based on distinct types**				√	√
S403	**ARRAY_MAX_CARDINALITY**				√	√
S404	**TRIM_ARRAY**				√	√
T011	**Timestamp in Information Schema**	√	√	√		
T021	**BINARY and VARBINARY data types**	√		√	√	√
T022	**Advanced support for BINARY and VARBINARY data types**			√	√	√
T023	**Compound binary literals**	√		√	√	√
T024	**Spaces in binary literals**	√		√	√	√
T031	**BOOLEAN data type**	√	√	√	√	√
T041	**Basic LOB data type support**		√	√	√	√
T041–01	BLOB data type	√	√	√	√	√
T041–02	CLOB data type	√	√	√	√	√
T041–03	POSITION, LENGTH, LOWER, TRIM, UPPER, and SUBSTRING functions for LOB data types		√	√	√	√
T041–04	Concatenation of LOB data types		√	√	√	√
T041–05	LOB locator: non-holdable		√	√	√	√
T042	**Extended LOB data type support**		√	√	√	√
T043	**Multiplier T**			√	√	√
T044	**Multiplier P**			√	√	√
T051	**Row types**		√	√	√	√
T052	**MAX and MIN for row types**				√	√
T053	**Explicit aliases for all-fields reference**		√	√	√	√
T061	**UCS support**		√	√	√	√
T071	**BIGINT data type**	√	√	√	√	√
T101	**Enhanced nullability determination**	√		√	√	√
T111	**Updatable joins, unions, and columns**		√	√	√	√
T121	**WITH (excluding RECURSIVE) in query expression**	√	√	√	√	√
T122	**WITH (excluding RECURSIVE) in subquery**	√	√	√	√	√
T131	**Recursive query**	√	√	√	√	√
T132	**Recursive query in subquery**	√	√	√	√	√

T141	**SIMILAR predicate**		√	√	√	
T151	**DISTINCT predicate**	√	√	√	√	√
T152	**DISTINCT predicate with negation**	√	√	√	√	√
T171	**LIKE clause in table definition**		√	√	√	√
T172	**AS subquery clause in table definition**		√	√	√	√
T173	**Extended LIKE clause in table definition**		√	√	√	√
T174	**Identity columns**		√	√	√	√
T175	**Generated columns**		√	√	√	√
T176	**Sequence generator support**	√	√	√	√	√
T177	**Sequence generator support: simple restart option**	√			√	√
T178	**Identity columns: simple restart option**				√	√
T180	**System-versioned tables**				√	√
T181	**Application-time period tables**				√	√
T191	**Referential action RESTRICT**	√	√	√	√	√
T201	**Comparable data types for referential constraints**		√	√	√	√
T211	**Basic trigger capability**		√	√	√	√
T211–01	Triggers activated on UPDATE, INSERT, or DELETE of one base table	√	√	√	√	√
T211–02	BEFORE triggers	√	√	√	√	√
T211–03	AFTER triggers	√	√	√	√	√
T211–04	FOR EACH ROW triggers	√	√	√	√	√
T211–05	Ability to specify a search condition that shall be True before the trigger is invoked	√	√	√	√	√
T211–06	Support for run-time rules for the interaction of triggers and constraints		√	√	√	√
T211–07	TRIGGER privilege		√	√	√	√
T211–08	Multiple triggers for the same event are executed in the order in which they were created in the catalog	√	√	√	√	√
T212	**Enhanced trigger capability**	√	√	√	√	√
T213	**INSTEAD OF triggers**	√		√	√	√
T231	**Sensitive cursors**		√	√	√	√
T241	**START TRANSACTION statement**	√	√	√	√	√
T251	**SET TRANSACTION statement: LOCAL option**		√	√	√	√
T261	**Chained transactions**		√	√	√	√
T271	**Savepoints**		√	√	√	√
T272	**Enhanced savepoint management**		√	√	√	√
T281	**SELECT privilege with column granularity**		√	√	√	√
T285	**Enhanced derived column names**	√		√	√	√
T301	**Functional dependencies**		√	√	√	√
T312	**OVERLAY function**		√	√	√	√
T323	**Explicit security for external routines**		√	√	√	√
T324	**Explicit security for SQL routines**		√	√	√	√
T325	**Qualified SQL parameter references**		√	√	√	√
T326	**Table functions**		√	√	√	√
T331	**Basic roles**		√	√	√	√
T332	**Extended roles**		√	√	√	√
T341	**Overloading of SQL-invoked functions and SQL-invoked procedures**	√			√	√
T351	**Bracketed comments**		√	√	√	√
T431	**Extended grouping capabilities (previously "CUBE and ROLLUP operations")**		√	√	√	√
T432	**Nested and concatenated GROUPING SETS**		√	√	√	√
T433	**Multiargument GROUPING function**		√	√	√	√
T434	**GROUP BY DISTINCT**		√	√	√	√
T441	**ABS and MOD functions**	√	√	√	√	√
T461	**Symmetric BETWEEN predicate**	√	√	√	√	√
T471	**Result sets return value**		√	√	√	√
T472	**DESCRIBE CURSOR**				√	√
T491	**LATERAL derived table**		√	√	√	√
T495	**Combined data change and retrieval**				√	√

T501	Enhanced EXISTS predicate	√	√	√	√	√
T502	Period predicates				√	√
T511	Transaction counts		√	√	√	√
T521	Named arguments in CALL statement				√	√
T522	Default values for IN parameters of SQL-invokes procedures				√	√
T551	Optional key words for default syntax	√	√	√	√	√
T561	Holdable locators		√	√	√	√
T571	Array-returning external SQL-invoked functions		√	√	√	√
T572	Multiset-returning external SQL-invoked functions		√	√	√	√
T581	Regular expression substring function		√	√	√	
T591	UNIQUE constraints of possibly null columns	√	√	√	√	√
T601	Local cursor references		√	√	√	√
T611	Elementary OLAP operations		√	√	√	√
T612	Advanced OLAP operations		√	√	√	√
T613	Sampling		√	√	√	√
T614	NTILE function				√	√
T615	LEAD and LAG functions				√	√
T616	Null treatment option for LEAD and LAG functions				√	√
T617	FIRST_VALUE and LAST_VALUE functions				√	√
T618	NTH_VALUE function				√	√
T619	Nested window functions				√	√
T620	WINDOW clause: GROUPS option				√	√
T621	Enhanced numeric functions		√	√	√	√
T641	Multiple column assignment	√	√	√	√	√
T651	SQL-schema statements in SQL routines		√	√	√	√
T652	SQL-dynamic statements in SQL routines		√	√	√	√
T653	SQL-schema statements in external routines		√	√	√	√
T654	SQL-dynamic statements in external routines		√	√	√	√
T655	Cyclically dependent routines		√	√	√	√
T811	Basic SQL/JSON constructor functions					√
T812	SQL/JSON: JSON_OBJECTAGG					√
T813	SQL/JSON: JSON_ARRAYAGG with ORDER BY					√
T814	Colon in JSON_OBJECT or JSON_OBJECTAGG					√
T821	Basic SQL/JSON query operators					√
T822	SQL/JSON: IS JSON WITH UNIQUE KEYS predicate					√
T823	SQL/JSON: PASSING clause					√
T824	JSON_TABLE: specific PLAN clause					√
T825	SQL/JSON: ON EMPTY and ON ERROR clauses					√
T826	General value expression in ON ERROR or ON EMPTY clauses					√
T827	JSON_TABLE: sibling NESTED COLUMNS clauses					√
T828	JSON_QUERY					√
T829	JSON_QUERY: array wrapper options					√
T830	Enforcing unique keys in SQL/JSON constructor functions					√
T831	SQL/JSON path language: strict mode					√
T832	SQL/JSON path language: item method					√
T833	SQL/JSON path language: multiple subscripts					√
T834	SQL/JSON path language: wildcard member accessor					√
T835	SQL/JSON path language: filter expressions					√
T836	SQL/JSON path language: starts with predicate					√
T837	SQL/JSON path language: regex_like predicate					√
T838	JSON_TABLE: PLAN DEFAULT clause					√
T839	Formatted cast of datetimes to/from character strings					√
V001	Mimer SQL extension	√	√	√	√	√

NOTE-1) A conforming SQL-implementation is required to support at least one embedded language or to

support the SQL-client module binding for at least one host language.

NOTE-2) Only on specific platforms.

NOTE-3) With "level of flagging" specified to be Core SQL Flagging and "extent of checking" specified to be Catalog Lookup.