

Feature	Sub Category	Explanation (for SQL)	MongoDB	Cassandra	Couchbase	CosmosDB
Declarative language		SQL is the declarative language for databases.	Javascript based, simple SQL command based query.	CQL: SQL inspired language for Cassandra.	N1QL: SQL for JSON, based on SQL++	DocumentDB SQL along with MongoDB API, Gremlin (for graph), etc. Simple select-from-where-order-by support
INPUT and OUTPUT SELECT		INPUT: Set of rows (tuples) OUTPUT: A set of rows (tuples)	INPUT: Sets of JSON; OUTPUT: A set of JSON	INPUT: Sets of rows OUTPUT: Set of JSON	INPUT: Sets of JSON OUTPUT: Set of JSON	INPUT: Sets of JSON OUTPUT: Set of JSON
	FROM clause	Specifies the datasource tables (relations)	db.tl.find() db.tl.aggregate(). aggregate() can join with additional collections using \$lookup operator.	FROM clause with only one table allowed. Now joins, subqueries or expressions are allowed. From clause interpretation is same as SQL.	FROM clause with multiple keyspaces (subset of bucket), subqueries, expressions. Same as SQL	FROM clause supports a single collection and self joins (same as UNNEST in Couchbase).
	WHERE clause	Criteria for selecting a row	db.tl.find({x:l0}); aggregate() has the \$match clause.	Standard boolean expressions. No subqueries.	Standard boolean expressions and subqueries.	Same as SQL
	SELECT clause	Projection clause	db.tl.find({x:l0}, {a:l, b:l})	SELECT clause is same as SQL.	SELECT clause. Same as SQL	SELECT clause, same as SQL
	GROUP BY clause	expressions to group the selected rows by	Unsupported in find(). aggregate() method supports \$group	Unsupported.	GROUP BY clause; Same as SQL	Unsupported
	ORDER BY clause	Final order of the results produced by the query block	db.tl.find().sort({a:l, b:-l}); aggregate() has \$sort to specify the result order.	ORDER BY clause; Same as SQL.	ORDER BY clause; Same as SQL	ORDER BY clause; Same as SQL
	CTE - Common Table Expressions (WITH Clause)	Dynamically defined	Unsupported.	Unsupported.	WITH clause; same as SQL (in v6.5). Recursive CTE is unsupported	Unsupported
	Subquery	Subqueries in the FROM clause, WHERE clause, anywhere an expression is allowed.	Unsupported in find(). Can add \$match in the pipeline, but not exactly an equivalent of a subquery.	Unsupported.	Supports both correlated and non-correlated subqueries.	Unsupported
	HAVING clause	Filtering after the aggregation.	\$match after the grouping and aggregation.	Unsupported.	HAVING clause; Same as SQL	Unsupported
	LIMIT, OFFSET	Pagination	skip(), limit() methods with find(). \$offset, \$limit with aggregate().	LIMIT is supported. OFFSET is unsupported.	LIMIT and OFFSET clause; Same as SQL	LIMIT (TOP) and OFFSET clause; Similar to SQL
	JOINS	INNER JOIN, LEFT/RIGHT/FULL outer joins.	Limited LEFT OUTER JOIN only via \$lookup operator	Joins are unsupported. Applications will have to model the data to avoid joins or do the joins in the application layer.	Supports INNER, LEFT OUTER, NEST, UNNEST and limited RIGHT outer. Same syntax as SQL. FULL OUTER join is unsupported.	Only self JOINS. No INNER/LEFT/RIGHT/etc joins.
	Aggregation	Aggregation	\$sum, \$count, \$avg	Simple aggregation on the whole result is supported. Aggregation with GROUP BY is unsupported.	SUM, AVG, COUNT, MAX, MIN, VARIANCE: same as SQL	Simple aggregation on the whole result is supported. Aggregation with GROUP BY is unsupported.
	Window Functions	Window functions for running totals using the OVER() clause	Unsupported	Unsupported.	Supports SQL Standard window analytical functions in 6.5.	Unsupported
INSERT		Insert data into the database				

Feature	Sub Category	Explanation (for SQL)	MongoDB	Cassandra	Couchbase	CosmosDB
	Single row/document		db.tl.save()	INSERT	INSERT	API insert
	Multi document		db.tl.insert()	Unsupported	INSERT with Multiple values	Unsupported
	INSERT...SELECT		Unsupported	Unsupported	INSERT... SELECT; Same as SQL	Unsupported
	INSERT... RETURNING		Unsupported	Unsupported	INSERT...RETURNING; Same as SQL	Unsupported
UPDATE			db.tl.update()	UPDATE; Same as SQL	UPDATE; Same as SQL	API update
DELETE			db.tl.delete()	DELETE; Same as SQL	DELETE; Same as SQL	API delete
MERGE			Unsupported	Unsupported	MERGE; Same as SQL	Unsupported
UPSERT			db.tl.update()	UPDATE	UPSERT; Same as SQL	API upsert
GROUP BY		Grouping of rows based on one or more expressions.	\$group operator in aggregate() pipeline	Unsupported	GROUP BY; Same as SQL	Unsupported
PREPARE		Parse, analyze and create an execution plan.	Unsupported	Unsupported	PREPARE	Unsupported
EXECUTE		Execute the ad-hoc or previously prepared statement.	Unsupported	Unsupported	EXECUTE	Unsupported
GRANT		Grant permission on data for specific operation		GRANT	GRANT ROLE	API support
REVOKE				REVOKE	REVOKE ROLE	API support
DESCRIBE			Compass tool - Graphical	DESCRIBE	INFER	Unsupported
TRUNCATE			Unsupported. Workaround via remove and createCollection	TRUNCATE	FLUSH operation	Unsupported
Value Logic						
Boolean Values			true, false, null	true, false, null	true, false, null, missing	true, false, null
Query Optimizer						
Type of Optimizer			Custom	Rule based Optimizer, mainly index selection for the one table. No joins.	Rule based Optimizer	Rule based optimizer, mainly does index selection.
Query Rewrite			None	No known rewrites.	Basic query rewrite.	No known rewrites.
Index Selection			Yes	Yes	Yes	Yes
Join type selection			Only nested loop is supported.	Not applicable since joins are unsupported.	Nested loop by default. Hash join by user hint.	Not clearly documented.
Join reordering			None	Not applicable since joins are unsupported.	Does not reorder. Executed as written in the FROM clause.	Not clearly documented.
Transactions						
ACID			Full ACID in 4.0	No	Document level ACID	Yes

Feature	Sub Category	Explanation (for SQL)	MongoDB	Cassandra	Couchbase	CosmosDB
Multi-statement tx savepoints			Yes	No	No	Yes
			No	No	No	No
Consistency			tbd	Tunable consistency on reads	consistent read of KV Tunabe for index scans	Five consistency models from eventual to strong.
Indexes						
Primary Index			Yes	Yes	Yes	Yes. Indexes everything
Secondary Index			Yes	Yes	Yes	Yes. Indexes everything
Composite key index			Yes	Yes	Yes	Yes. Indexes everything
Partial index			No	No	Yes	Yes. Indexes everything
Functional Index			No	No	Yes	Yes. Indexes everything
Array Index			Yes	Yes, for collection types	Yes	Yes. Indexes everything
Search index			No. B-tree index is used for token search	Need to use the SOLR integration to replicate the data.	Yes	No
Adaptive Index			No	NO	Yes	Yes. Indexes everything
Spatial Index			Yes	No. Need to use SOLR.	Yes (FTS Index)	Yes
Partitioned Index			Yes	No	Yes	Yes
Replica Index			Yes	No	Yes	Yes
Datatypes						
Numeric			JSON number	Numeric	JSON number	JSON number
Character			JSON string	decimal, double,int, float, varint	JSON string	JSON string
date time			ISO time	Timestamp	ISO 8601 strings	ISO 8601 strings
timezone			No	No	No. Has conversion and arithmetic functions	No.
blob			Yes	Yes	Yes	No.
spatial			GeoJSON	No	GeoJSON	GeoJSON
Array			Yes	Collection types	JSON array	JSON array
JSON			Yes	Yes.	Yes	JSON