- The only type of expression that is permitted in a multi-valued key part is a `JSON` expression. The expression need not reference an existing element in a JSON document inserted into the indexed column, but must itself be syntactically valid.
- Because index records for the same clustered index record are dispersed throughout a multi-valued index, a multi-valued index does not support range scans or index-only scans.
- Multi-valued indexes are not permitted in foreign key specifications.
- Index prefixes cannot be defined for multi-valued indexes.
- Multi-valued indexes cannot be defined on data cast as `BINARY` (see the description of the `CAST()` function).
- Online creation of a multi-value index is not supported, which means the operation uses `ALGORITHM=COPY`. See Performance and Space Requirements.
- Character sets and collations other than the following two combinations of character set and collation are not supported for multi-valued indexes:
  1. The `binary` character set with the default `binary` collation
  2. The `utf8mb4` character set with the default `utf8mb4_0900_as_cs` collation.
- As with other indexes on columns of `InnoDB` tables, a multi-valued index cannot be created with `USING HASH`; attempting to do so results in a warning: `This storage engine does not support the HASH index algorithm, storage engine default was used instead`. (`USING BTREE` is supported as usual.)

Spatial Indexes

The `MyISAM`, `InnoDB`, `NDB`, and `ARCHIVE` storage engines support spatial columns such as `POINT` and `GEOMETRY`. (Section 11.4, “Spatial Data Types”, describes the spatial data types.) However, support for spatial column indexing varies among engines. Spatial and nonspatial indexes on spatial columns are available according to the following rules.

Spatial indexes on spatial columns have these characteristics:

- Available only for `InnoDB` and `MyISAM` tables. Specifying `SPATIAL INDEX` for other storage engines results in an error.
- As of MySQL 8.0.12, an index on a spatial column *must* be a `SPATIAL` index. The `SPATIAL` keyword is thus optional but implicit for creating an index on a spatial column.
- Available for single spatial columns only. A spatial index cannot be created over multiple spatial columns.
- Indexed columns must be `NOT NULL`.
- Column prefix lengths are prohibited. The full width of each column is indexed.
- Not permitted for a primary key or unique index.

Nonspatial indexes on spatial columns (created with `INDEX`, `UNIQUE`, or `PRIMARY KEY`) have these characteristics:

- Permitted for any storage engine that supports spatial columns except `ARCHIVE`.
- Columns can be `NULL` unless the index is a primary key.