- The `reIndex` rebuilds the index.
- You add or drop an index.
- The `mongod` process restarts.

Changed in version 2.6: `explain()` operations no longer read from or write to the query planner cache.

Cached Query Plan Interface

New in version 2.6.

MongoDB provides `http://docs.mongodb.org/manual/reference/method/js-plan-cache` to view and modify the cached query plans.

Index Filters

New in version 2.6.

Index filters determine which indexes the optimizer evaluates for a *query shape*. A query shape consists of a combination of query, sort, and projection specifications. If an index filter exists for a given query shape, the optimizer only considers those indexes specified in the filter.

When an index filter exists for the query shape, MongoDB ignores the `hint()`. To see whether MongoDB applied an index filter for a query shape, check the `indexFilterSet` field of either the `db.collection.explain()` or the `cursor.explain()` method.

Index filters only affects which indexes the optimizer evaluates; the optimizer may still select the collection scan as the winning plan for a given query shape.

Index filters exist for the duration of the server process and do not persist after shutdown. MongoDB also provides a command to manually remove filters.

Because index filters overrides the expected behavior of the optimizer as well as the `hint()` method, use index filters sparingly.

See `planCacheListFilters`, `planCacheClearFilters`, and `planCacheSetFilter`.

Distributed Queries

Read Operations to Sharded Clusters

Sharded clusters allow you to partition a data set among a cluster of `mongod` instances in a way that is nearly transparent to the application. For an overview of sharded clusters, see the *Sharding* (page 661) section of this manual.

For a sharded cluster, applications issue operations to one of the `mongos` instances associated with the cluster.

Read operations on sharded clusters are most efficient when directed to a specific shard. Queries to sharded collections should include the collection's *shard key* (page 674). When a query includes a shard key, the `mongos` can use cluster metadata from the *config database* (page 670) to route the queries to shards.

If a query does not include the shard key, the `mongos` must direct the query to *all* shards in the cluster. These *scatter gather* queries can be inefficient. On larger clusters, scatter gather queries are unfeasible for routine operations.

For more information on read operations in sharded clusters, see the *Sharded Cluster Query Routing* (page 678) and *Shard Keys* (page 674) sections.