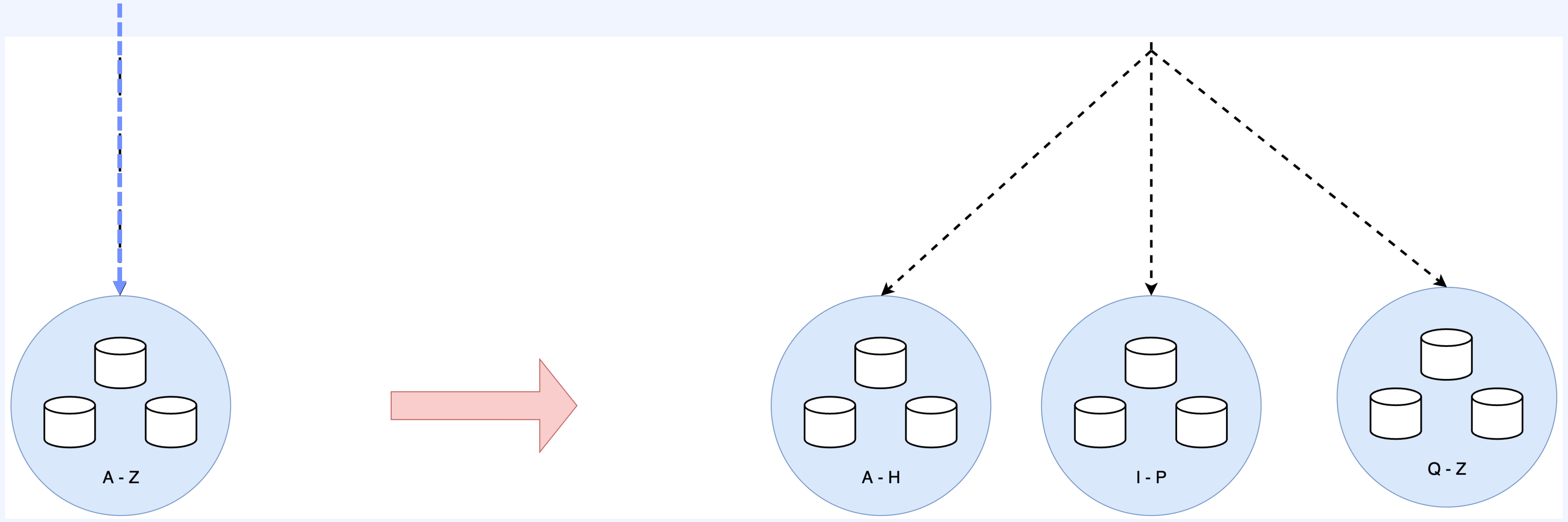
Document Query 실습

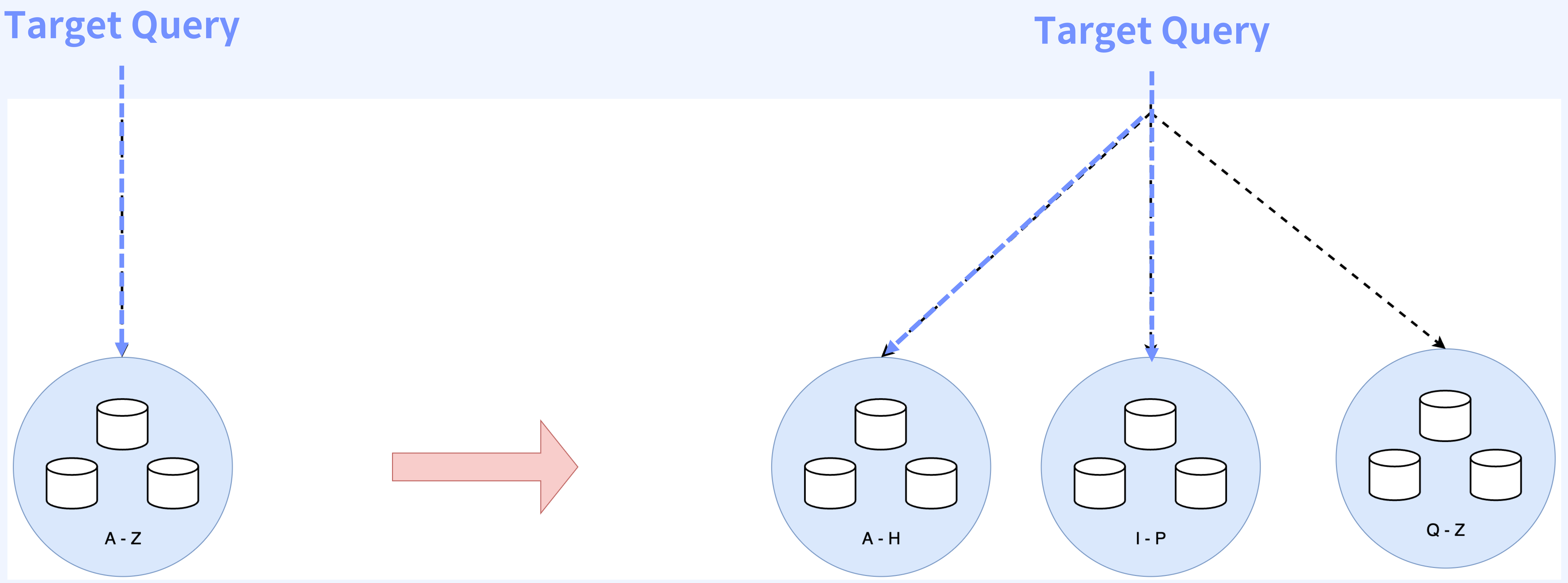
10 배포형태에 따른 CRUD 특징 (Replica Set vs Sharded Cluster)

Target Query vs Broadcast Query

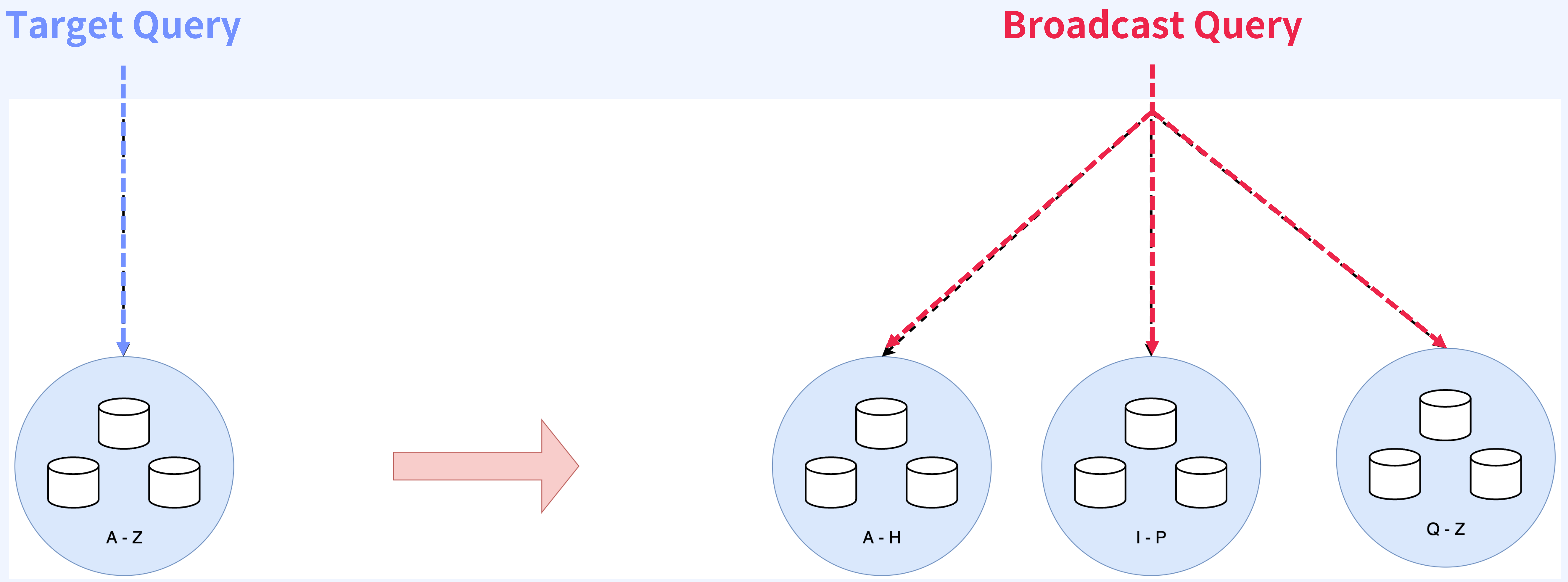
Target Query



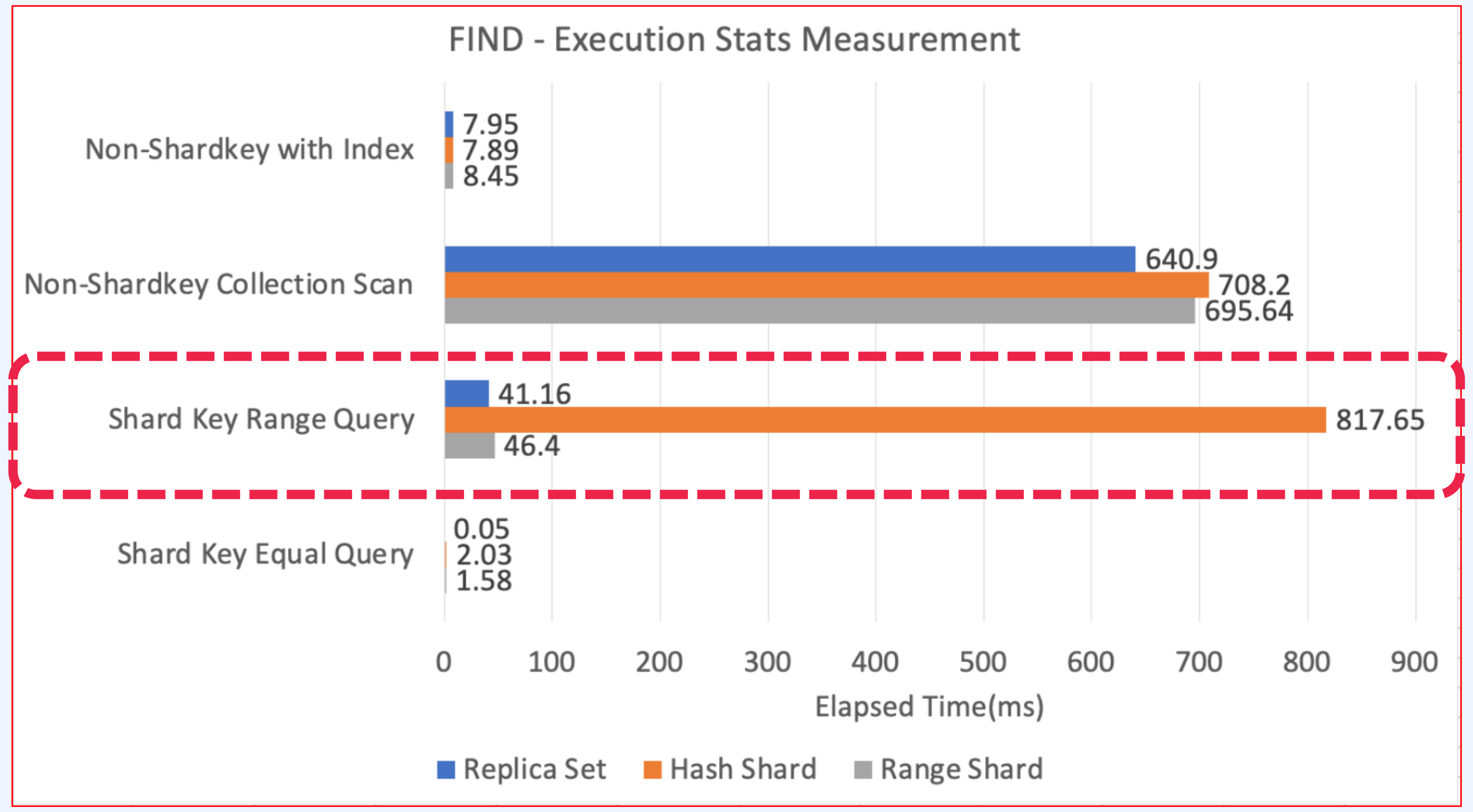
Target Query vs Broadcast Query

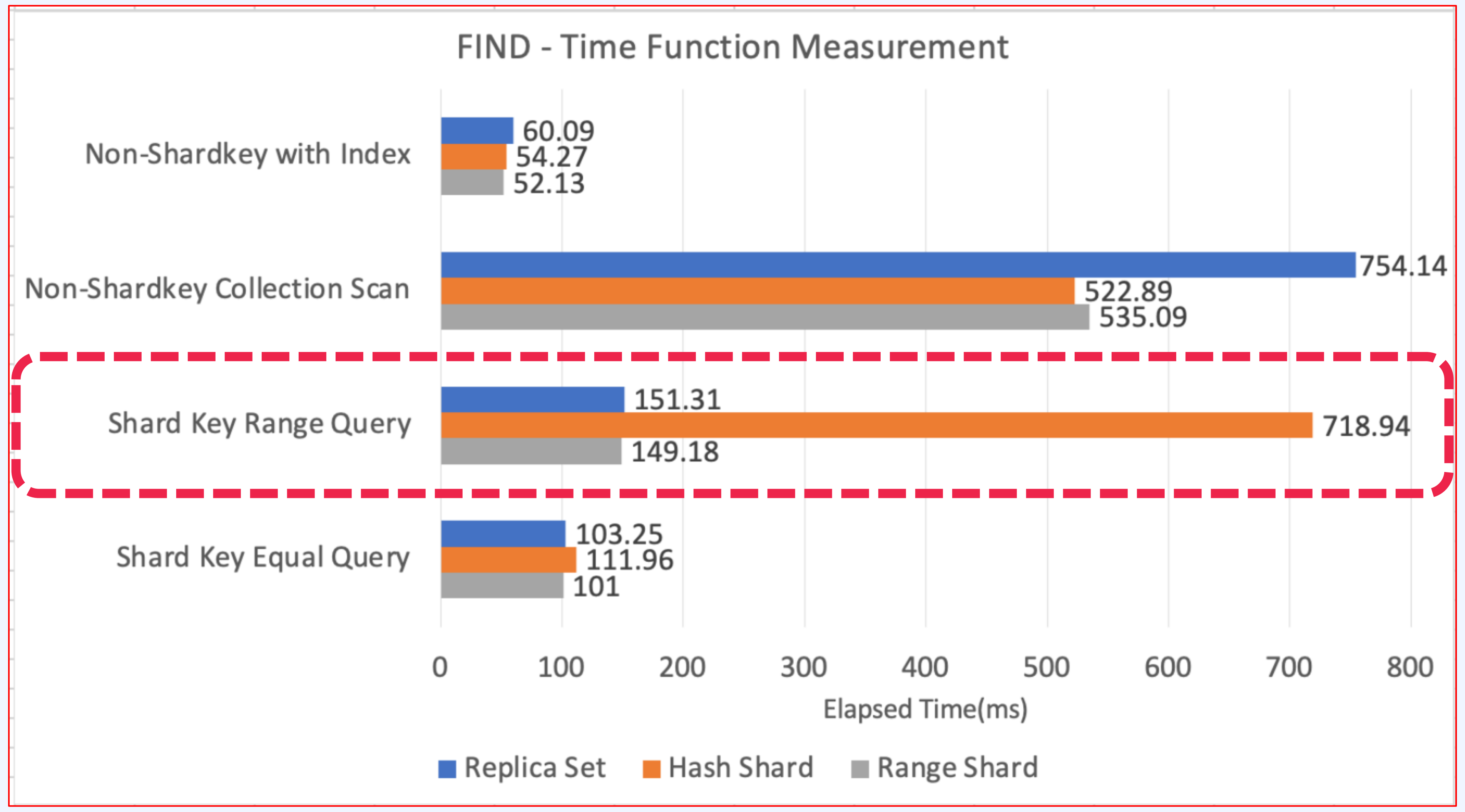


Target Query vs Broadcast Query



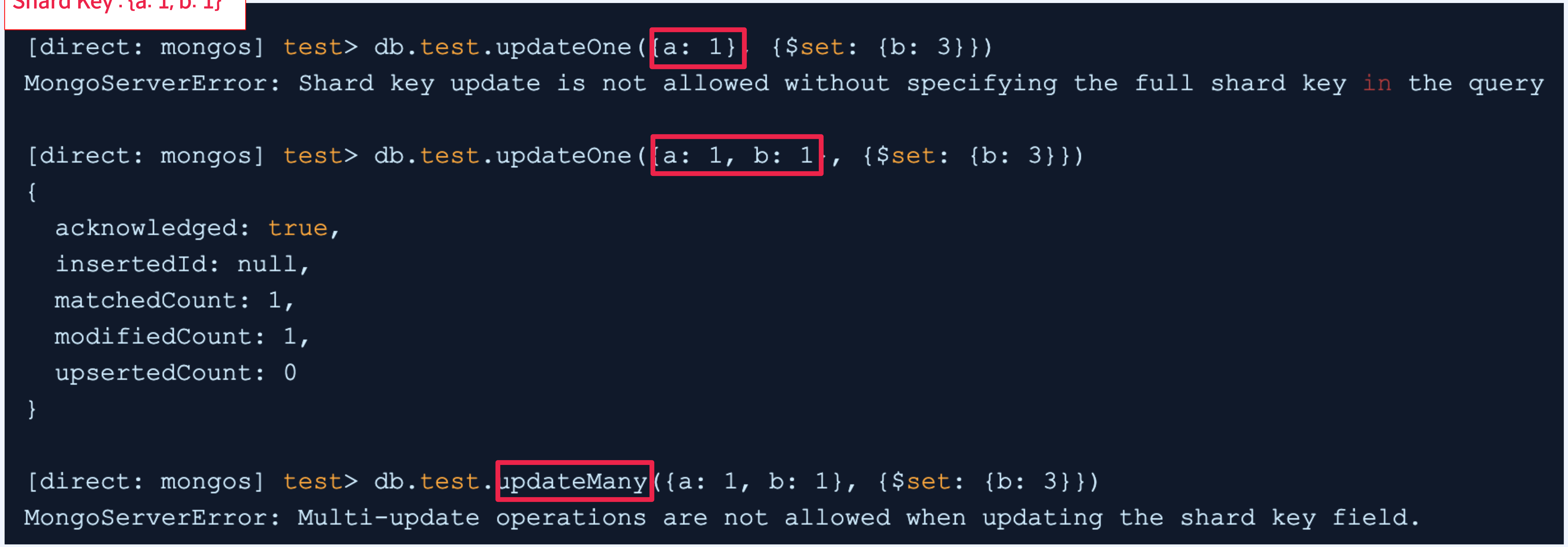
Target Query vs Broadcast Query





Updating Shard Keys

Shard Key : {a: 1, b: 1}

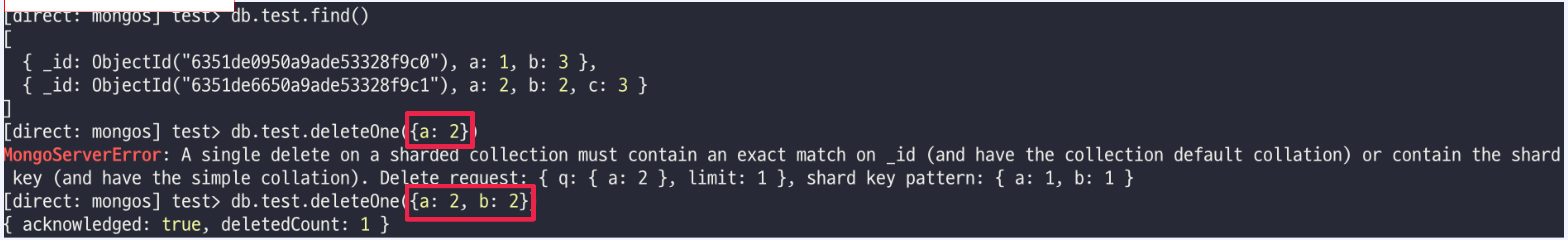


```
[direct: mongos] test> db.test.updateOne({a: 1}, {$set: {b: 3}})
MongoServerError: Shard key update is not allowed without specifying the full shard key in the query

[direct: mongos] test> db.test.updateOne({a: 1, b: 1}, {$set: {b: 3}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

[direct: mongos] test> db.test.updateMany({a: 1, b: 1}, {$set: {b: 3}})
MongoServerError: Multi-update operations are not allowed when updating the shard key field.
```

Version <= 4.0 : Shard Key 필드는 한번 생성되면 수정할 수 없다.
Version >= 4.2 : 동등 조건으로 Shard Key의 모든 필드를 Query Filter에 넣어야 수정할 수 있다. (Multi-Update는 불가능)

Deleting with Shard Keys

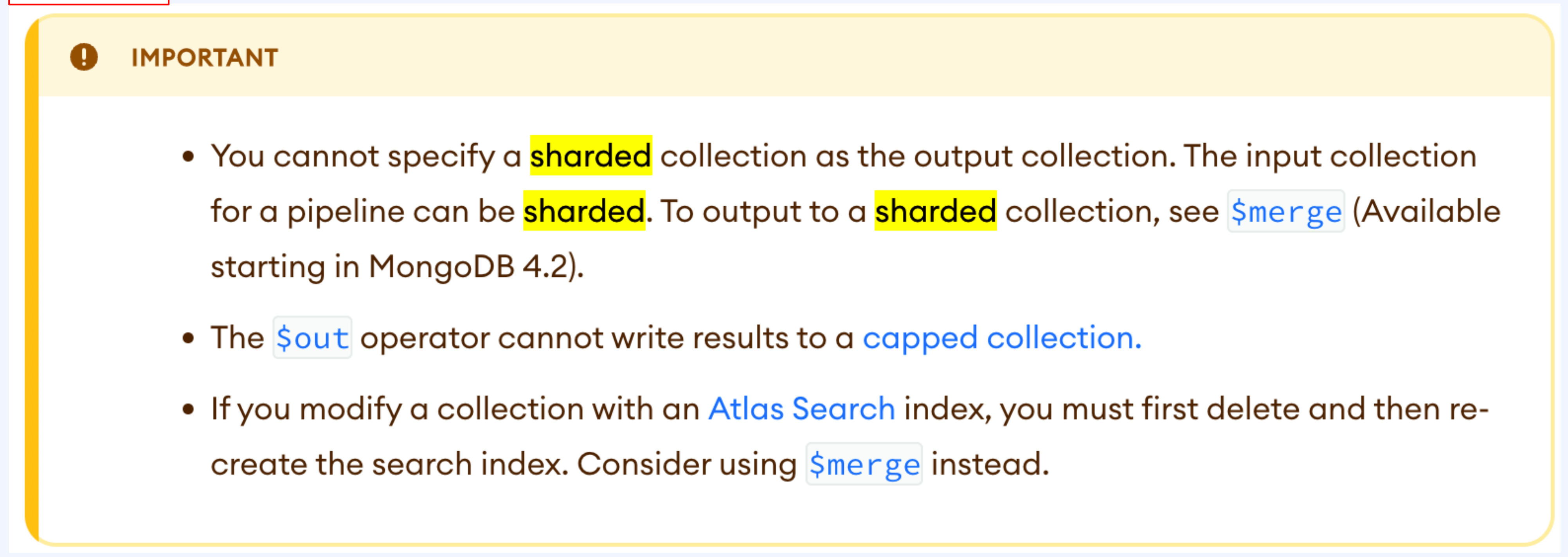
Shard Key : {a: 1, b: 1}



```
[direct: mongos] test> db.test.find()
[
  { _id: ObjectId("6351de0950a9ade53328f9c0"), a: 1, b: 3 },
  { _id: ObjectId("6351de6650a9ade53328f9c1"), a: 2, b: 2, c: 3 }
]
[direct: mongos] test> db.test.deleteOne({a: 2})
MongoServerError: A single delete on a sharded collection must contain an exact match on _id (and have the collection default collation) or contain the shard
 key (and have the simple collation). Delete request: { q: { a: 2 }, limit: 1 }, shard key pattern: { a: 1, b: 1 }
[direct: mongos] test> db.test.deleteOne({a: 2, b: 2})
{ acknowledged: true, deletedCount: 1 }
```

Others

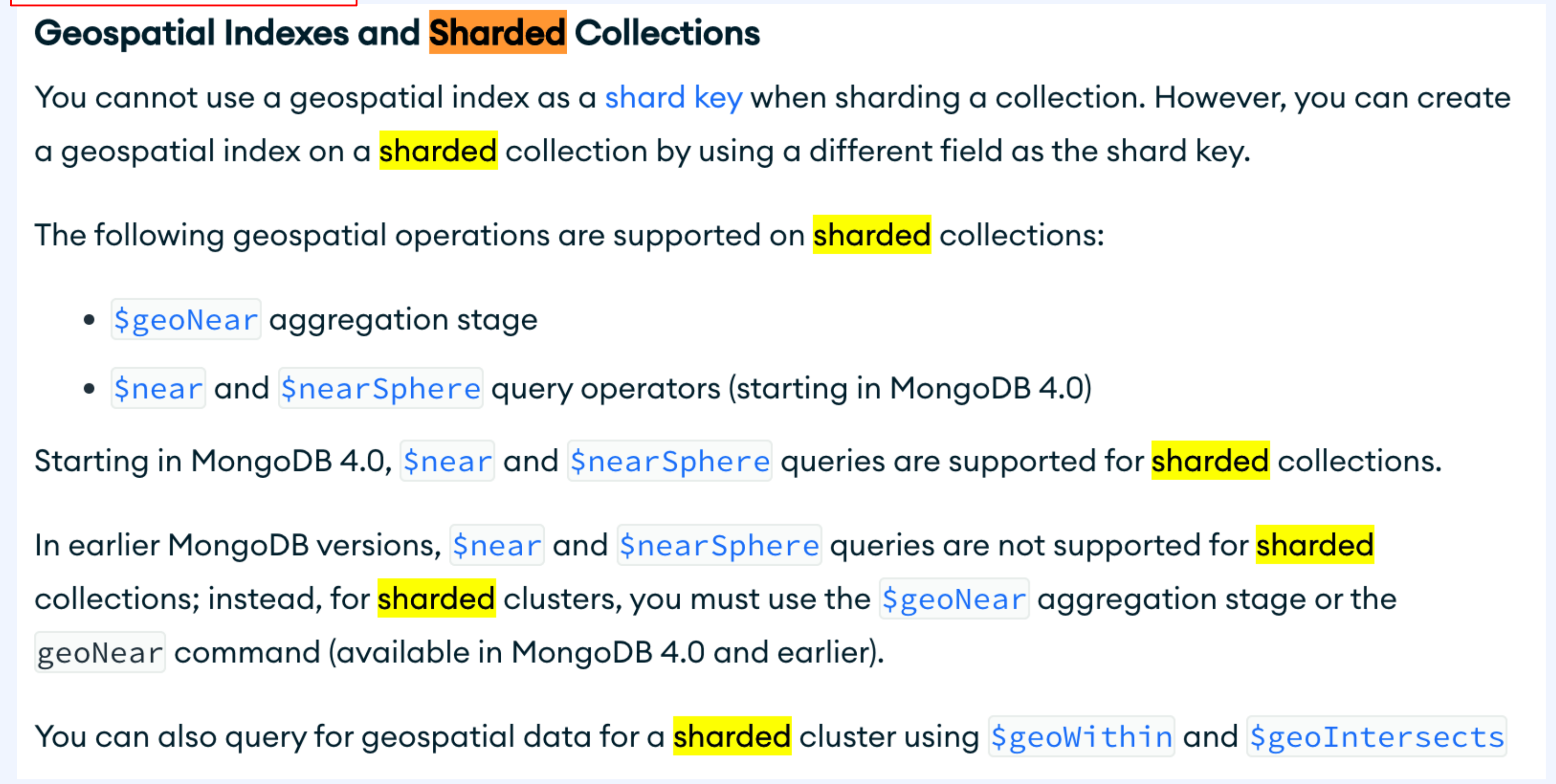
$out



IMPORTANT

- You cannot specify a sharded collection as the output collection. The input collection for a pipeline can be sharded. To output to a sharded collection, see `$merge` (Available starting in MongoDB 4.2).
- The `$out` operator cannot write results to a capped collection.
- If you modify a collection with an Atlas Search index, you must first delete and then re-create the search index. Consider using `$merge` instead.

Geospatial Index



Geospatial Indexes and Sharded Collections

You cannot use a geospatial index as a shard key when sharding a collection. However, you can create a geospatial index on a sharded collection by using a different field as the shard key.

The following geospatial operations are supported on sharded collections:

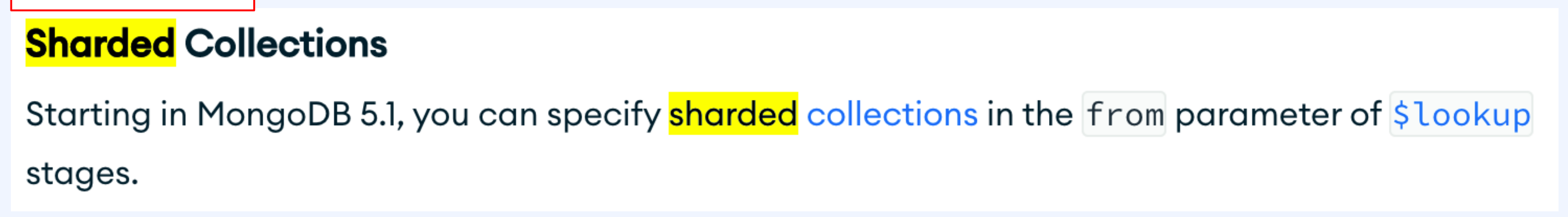
- `$geoNear` aggregation stage
- `$near` and `$nearSphere` query operators (starting in MongoDB 4.0)

Starting in MongoDB 4.0, `$near` and `$nearSphere` queries are supported for sharded collections.

In earlier MongoDB versions, `$near` and `$nearSphere` queries are not supported for sharded collections; instead, for sharded clusters, you must use the `$geoNear` aggregation stage or the `geoNear` command (available in MongoDB 4.0 and earlier).

You can also query for geospatial data for a sharded cluster using `$geoWithin` and `$geoIntersects`

$lookup



Sharded Collections

Starting in MongoDB 5.1, you can specify sharded collections in the `from` parameter of `$lookup` stages.

MongoDB Documentation
https://www.mongodb.com/docs/