**Basic Single-Field Indexes**

db.collection.createIndex({ name: 1 }) // Ascending index on 'name'

db.collection.createIndex({ age: -1 }) // Descending index on 'age'

db.collection.createIndex({ email: 1 }, { unique: true }) // Unique index on 'email'

db.collection.createIndex({ createdAt: 1 }, { expireAfterSeconds: 3600 }) // TTL index on 'createdAt'

db.collection.createIndex({ isActive: 1 }, { sparse: true }) // Sparse index on 'isActive'

**Compound Indexes**

db.collection.createIndex({ firstName: 1, lastName: 1 }) // Compound index on 'firstName' and 'lastName'

db.collection.createIndex({ category: 1, price: -1 }) // Category ascending, price descending

db.collection.createIndex({ country: 1, city: 1, zipCode: 1 }) // Compound index on 'country', 'city', and 'zipCode'

db.collection.createIndex({ status: 1, priority: -1, createdAt: 1 }) // Multi-field compound index

db.collection.createIndex({ department: 1, role: 1 }, { unique: true }) // Unique compound index

**Text Indexes**

db.collection.createIndex({ description: "text" }) // Text index on 'description'

db.collection.createIndex({ title: "text", content: "text" }) // Text index on multiple fields

db.collection.createIndex({ title: "text", summary: "text" }, { default\_language: "english" }) // Text index with language option

db.collection.createIndex({ title: "text", tags: "text" }, { weights: { title: 2, tags: 1 } }) // Weighted text index

db.collection.createIndex({ "$\*\*": "text" }) // Wildcard text index for full-text search on all fields

**Geospatial Indexes**

db.collection.createIndex({ location: "2dsphere" }) // 2dsphere index for GeoJSON location data

db.collection.createIndex({ coordinates: "2d" }) // 2d index for legacy coordinate pairs

db.collection.createIndex({ area: "2dsphere", category: 1 }) // Compound geospatial index

db.collection.createIndex({ city: 1, position: "2dsphere" }) // Geospatial + regular index

db.collection.createIndex({ boundary: "2d" }) // 2d index on boundary field

**Hashed Indexes**

db.collection.createIndex({ userId: "hashed" }) // Hashed index for sharding

db.collection.createIndex({ email: "hashed" }) // Hashed index on email for distribution

db.collection.createIndex({ sessionToken: "hashed" }) // Hashed index for fast lookup

db.collection.createIndex({ transactionId: "hashed" }) // Hashed index on transactionId

db.collection.createIndex({ productCode: "hashed" }) // Hashed index on productCode

**Partial Indexes**

db.collection.createIndex({ status: 1 }, { partialFilterExpression: { status: { $eq: "active" } } }) // Index only active documents

db.collection.createIndex({ age: 1 }, { partialFilterExpression: { age: { $gte: 18 } } }) // Index only adults

db.collection.createIndex({ isDeleted: 1 }, { partialFilterExpression: { isDeleted: { $eq: false } } }) // Ignore deleted documents

db.collection.createIndex({ premiumUser: 1 }, { partialFilterExpression: { premiumUser: true } }) // Index premium users only

db.collection.createIndex({ lastLogin: 1 }, { partialFilterExpression: { lastLogin: { $exists: true } } }) // Ignore documents without lastLogin

**Sparse Indexes**

db.collection.createIndex({ phoneNumber: 1 }, { sparse: true }) // Index only documents with phoneNumber

db.collection.createIndex({ address: 1 }, { sparse: true }) // Index only documents with an address field

db.collection.createIndex({ nickname: 1 }, { unique: true, sparse: true }) // Unique sparse index on nickname

db.collection.createIndex({ profilePicture: 1 }, { sparse: true }) // Ignore documents without profilePicture

db.collection.createIndex({ secondaryEmail: 1 }, { sparse: true }) // Sparse index on secondaryEmail

**Wildcard Indexes**

db.collection.createIndex({ "$\*\*": 1 }) // Wildcard index on all fields

db.collection.createIndex({ "attributes.$\*\*": 1 }) // Wildcard index for all attributes subfields

db.collection.createIndex({ "metadata.$\*\*": "text" }) // Wildcard text index on metadata fields

db.collection.createIndex({ "details.$\*\*": 1, createdAt: -1 }) // Wildcard index with createdAt sorting

db.collection.createIndex({ "dynamicFields.$\*\*": "hashed" }) // Wildcard hashed index