MONGODB ATLAS SEARCHES

🔍 1. Exact Match Search

- Uses standard **find()** with exact field-value matches.

```
db.collection.find({ name: "Alice" })
```

🔎 2. Range Search

- Use operators like `$gt`, `$lt`, `$gte`, `$lte` to search numeric or date ranges.

```
db.collection.find({ age: { $gte: 18, $lte: 30 } })
```

📝 3. Pattern Matching / Regex Search

- Uses regular expressions to match string patterns.

```
db.collection.find({ name: { $regex: "^A", $options: "i" } })
```

🌐 4. Text Search (Full-Text Search)

- Requires a text index and allows searching text fields for keywords, supporting stemming and language processing.

```
db.collection.createIndex({ description: "text" })
db.collection.find({ $text: { $search: "mongodb database" } })
```

🧠 5. Contextual / Semantic Search (Vector Search with Atlas)

- Introduced with **MongoDB Atlas Vector Search**, enabling **semantic or contextual understanding**.

- Stores and searches **vector embeddings** of text (e.g., from OpenAI, BERT).

json

```
{
  "embedding": [0.123, 0.456, ...] // Dense vector
}
```

- Example (Atlas Vector Search):

js
```
db.collection.aggregate([
  {
    $search: {
      knnBeta: {
        vector: [ ... ], // Query vector
        path: "embedding",
        k: 5
      }
    }
  }
])
```

🗺️ 6. Geospatial Search

- Used to query geolocation data.

js
```
db.collection.createIndex({ location: "2dsphere" })
db.collection.find({
  location: {
    $near: {
      $geometry: {
        type: "Point",
        coordinates: [longitude, latitude]
      },
      $maxDistance: 1000
    }
  }
})
```

📊 7. Aggregation-Based Search

- Combines multiple stages (`$match`, `$group`, `$project`, `$search`, etc.) to perform complex queries.

js
```
db.collection.aggregate([
  { $match: { age: { $gt: 25 } } },
  { $group: { _id: "$city", count: { $sum: 1 } } }
])
```

🔄 8. Wildcard & Array Search

- Match documents where a field matches one of several values or contains certain array elements.

js
```
db.collection.find({ tags: { $in: ["AI", "ML"] } })
```

📌 9. Faceted Search (via Aggregation)

- Similar to search on e-commerce websites, where results can be filtered and grouped by categories.

js
```
{
  $searchMeta: {
    facet: {
      operator: { text: { query: "shoes", path: "productName" } },
      facets: {
        brandFacet: { type: "string", path: "brand" },
        priceFacet: { type: "number", path: "price", boundaries: [0,
50, 100] }
      }
    }
  }
}
```

10. Wildcard Search (on unknown field names)

- Allows search across fields without specifying exact field names using `wildcard` index (Atlas Search).

json

```
{
  $search: {
    wildcard: {
      path: "user.*",
      query: "*smith*"
    }
  }
}
```

🎯 11. Autocomplete Search

- Used in search boxes to suggest completions.

json

```
{
  $search: {
    autocomplete: {
      query: "mong",
      path: "technology",
      tokenOrder: "sequential"
    }
  }
}
```

Requires **autocomplete** index in MongoDB Atlas Search.

🔠 12. Fuzzy Search

- Allows minor typos/misspellings (useful for user input).

json

```
{
  $search: {
    text: {
      query: "mongdb", // misspelled
      path: "title",
      fuzzy: { maxEdits: 2 }
    }
  }
}
```

🌍 13. Compound Search

- Combine multiple search types in a single query (e.g., text + range).

json

```
{
  $search: {
    compound: {
      must: [
        { text: { query: "AI", path: "tags" } },
        { range: { path: "score", gte: 80 } }
      ]
    }
  }
}
```

📦 14. Embedded/Nested Document Search

- Search inside deeply nested or embedded fields.

js

```
db.collection.find({ "profile.education.degree": "MBA" })
```

📚 15. Custom Ranking Search

- Boost scores based on field values or weights in Atlas Search.

json
```
{
  $search: {
    text: {
      query: "mongodb",
      path: "content"
    },
    score: {
      boost: {
        value: { path: "popularity" }
      }
    }
  }
}
```

⏳ 16. Time-Series Search

- Used when MongoDB stores **time-series data** (e.g., sensor data, logs). You can search efficiently using time-range filters and aggregations.

js
```
db.sensorData.find({
  timestamp: {
    $gte: ISODate("2025-01-01T00:00:00Z"),
    $lte: ISODate("2025-01-31T23:59:59Z")
  }
})
```

🧩 17. Regex Search on Arrays or Documents

- Regex not only on strings but inside arrays or embedded fields.

js
```
db.users.find({ "skills": { $regex: /data/i } })
```

🧠 18. Hybrid Semantic + Keyword Search

- Combine vector-based semantic search with keyword-based filters (possible with Atlas Vector Search).

json
```
{
  $search: {
    compound: {
      must: [
        { knnBeta: { path: "embedding", vector: [ ... ], k: 5 } },
        { text: { query: "mongodb", path: "tags" } }
      ]
    }
  }
}
```

🧪 19. Custom Expression Search (via Aggregation + $expr)

- Search using complex conditions and expressions.

js
```
db.orders.find({
  $expr: { $gt: [ "$total", { $multiply: [ "$items", 100 ] } ] }
})
```

🧬 20. Search with Projections and Transformations

- Not exactly a search type but enhances output control using `$project`.

js
```
db.collection.find({}, { name: 1, age: 1, _id: 0 })
```