

# Projection in MongoDB

- Projection is selecting only required fields/data from documents instead of retrieving all fields/data.
- It is used with find() method.
- Syntax: db.collection.find({}, {field1:value, field2: value.....})
- In above syntax first parameter is used to specify condition to retrieve data.
- Second parameter of find() is used to specify the fields that are to be returned in or to be skipped from the output.
- If the value of the field is set to 1, the field will be displayed in the returned document.
- If the value of the field is set to 0, the field will not be displayed in the returned document.
- find() always display “\_id” field. To skip this filed, set it to 0.

# Projection in MongoDB

- Following are some of the examples of projection:
1. Display only rollno and percentage fields from document:
    - db.student.find({}, {rollno:1, percentage:1})
  2. Display all fields by skipping address field:
    - db.student.find({}, {address:0})
  3. Display roll numbers of students having marks greater than 70:
    - db.student.find({percentage:{gte:70}}, {rollno:1})
  4. Display names and mobile no. of students in studying in second year:
    - db.student.find({class:"SY"}, {name:1, mob:1})
  5. Display employee IDs and department ID of employees having salary<50000 and salary>20000 :
    - db.employee.find({salary:{\$lt:50000, \$gt:20000}}, {empid:1, deptid:1})

# Methods in MongoDB

## 1. limit():

- It is used to display limited/required number of documents from collection, starting from 1<sup>st</sup> document.
- limit() is used with find() method.

e.g. Display first 10 documents from book collection:

- db.book.find().limit(10)

Display 10 documents that fulfill given condition:

- db.book.find({publisher:"McGrawHill"}).limit(10)

## 2. skip():

- It is used to skip first n documents mentioned in the query.
- skip() is used with find() method.

• db.book.find().skip(05) #skip first 5 documents from result.

• Display documents within range 11 to 30 from book collection:

- db.book.find().limit(30).skip(10)

- db.book.find().skip(10).limit(20)

# Methods in MongoDB

- **save():**
- It either updates existing document or inserts new document depending on the document parameter mentioned in the query.
- It uses either insert or update method.
- There are following 3 possibilities with save():

**Case 1:** In save query, if “\_id” field is not mentioned in parameters, save() calls insert() method to insert new document, and mongo shell will provide ObjectId to the newly created document.

**Case 2:** If save query contains “\_id” field and if the document with mentioned id already exists in the collection, then save() calls update() method to update that existing document with fields:values given in the query.

**Case 3:** If save query contains “\_id” field and if the document with mentioned id does not exist in the collection, then save() calls insert() method to insert new document with given “\_id” and fields:values.

# Projection in MongoDB

1. Save a new document without specifying an \_id field:
  - db.student.save({rollno:10,name:"Jackson",percentage:67})
  - Here \_id is not mentioned, save() will insert above document in the collection.
2. Save a document with an \_id Field:
  - db.student.save({\_id:2001,rollno:11,name:"Jacky",percentage:70})
  - save() will either inserts above document or will replace existing document depending on the existence of mentioned (\_id:2001)
3. Replace an existing document:
  - db.student.save({\_id:2001,rollno:11,name:"Jacky",percentage:72, city:"Mumbai"})
  - save() will update existing document(\_id:2001) with given field:values.