

## Workshop 4

### Big Data

#### Department Data:

#### Department 10

```
>_MONGOSH ⓘ  
  
> db.dept.insertOne({  
  _id : 10,  
  deptno:10,  
  dname: "Accounting",  
  loc: "NewYork"  
})  
< {  
  acknowledged: true,  
  insertedId: 10  
}  
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

#### Department 20:

```
> db.dept.insertOne({  
  _id:20,  
  deptno:20,  
  dname: "RESEARCH",  
  loc: "DALLAS"  
})  
< {  
  acknowledged: true,  
  insertedId: 20  
}  
Atlas atlas-ezozlv-shard-0 [primary] hck-db> |
```

## Workshop 4

### Big Data

#### Department 40

```
> db.dept.insertOne({
  _id: 40,
  deptno: 40,
  dname: "OPERATIONS",
  loc: "BOSTON"
})
< {
  acknowledged: true,
  insertedId: 40
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

## Workshop 4

### Big Data

#### Object ID:

#### Duplicate ID ERROR:

adding a new document with an existing deptno, but different object ID, it will quite happily accept it:

```
> db.dept.insertOne({
  _id: 10,
  deptno: 50,
  dname: "TEST",
  loc: "Wolverhampton"
})
✖ ► MongoServerError: E11000 duplicate key error collection: hck-db.dept index: _id_ dup key: {
Atlas atlas-ezozlv-shard-0 [primary] hck-db> |
```

```
> db.dept.insertOne(
  {
    _id: 50,
    deptno: 40,
    dname: "OPERATIONS V2",
    loc: "BOSTON"
  })
< {
  acknowledged: true,
  insertedId: 50
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

## Workshop 4

### Big Data

#### Exercise 2.1

- Compare how you added the above data and how it differs from INSERT records in a relational database
- Add Department 30, which has the following key/values: `_id: 30`, `deptno: 30`, `dname: SALES` and `loc: CHICAGO`

```
> db.dept.insertOne({
  _id:30,
  deptno:30,
  dname: "SALES",
  loc:"CHICAGO"

})
< {
  acknowledged: true,
  insertedId: 30
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db> |
```

Workshop 4  
Big Data

Employee data (Using Insert Many)

```
>_ mongosh: sujan2024.xf...  dept  +
>_MONGOSH

  empNo: 7900,
  eName: "JAMES",
  job: "SALESMAN",
  mgr: 7689,
  hireDate: ISODate("1995-02-20"),
  sal: 1600,
  comm: 300,
  deptno: 30
},
{
  empNo: 7654,
  eName: "MARTIN",
  job: "SALESMAN",
  mgr: 7689,
  hireDate: ISODate("1995-02-20"),
  sal: 1600,
  comm: 300,
  deptno: 30
}
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67df7200688f9e7b0614d8c4'),
    '1': ObjectId('67df7200688f9e7b0614d8c5'),
    '2': ObjectId('67df7200688f9e7b0614d8c6')
  }
}
```

## Workshop 4

### Big Data

#### Exercise: 2.2

```
>_ mongosh: sujan2024.xf... hck-db dept emp +
>_MONGOSH
  mgr: 7698,
  hireDate: ISODate("1991-09-08"),
  sal: 1500,
  comm: 0,
  deptno: 30
},
{
  empNo: 7521,
  eName: "WARD",
  job: "SALESMAN",
  mgr: 7698,
  hireDate: ISODate("1994-02-22"),
  sal: 1250,
  comm: 500,
  deptno: 30
}
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67df7590688f9e7b0614d8c7'),
    '1': ObjectId('67df7590688f9e7b0614d8c8'),
    '2': ObjectId('67df7590688f9e7b0614d8c9'),
    '3': ObjectId('67df7590688f9e7b0614d8ca'),
    '4': ObjectId('67df7590688f9e7b0614d8cb'),
    '5': ObjectId('67df7590688f9e7b0614d8cc')
  }
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

## Workshop 4

### Big Data

#### Querying a collection

##### Find all documents

```
> db.emp.find()

< {
  _id: ObjectId('67df7590688f9e7b0614d8c7'),
  empNo: 7499,
  eName: 'ALLEN',
  job: 'SALESMAN',
  mgr: 7698,
  hireDate: 1995-02-20T00:00:00.000Z,
  sal: 1600,
  comm: 300,
  deptno: 30
}
```

##### Find with query criteria:

```
> db.emp.find( {job:"CLERK"} )

< {
  _id: ObjectId('67df7590688f9e7b0614d8c9'),
  empNo: 7900,
  eName: 'JAMES',
  job: 'CLERK',
  mgr: 7698,
  hireDate: 1995-12-03T00:00:00.000Z,
  sal: 950,
  comm: null,
  deptno: 30
}
```

## Workshop 4

### Big Data

**Find all employees who earn more than 2400:**

```
> db.emp.find( {sal: {$gt:2400}} )  
< {  
  _id: ObjectId('67df7590688f9e7b0614d8c8'),  
  empNo: 7698,  
  eName: 'BLAKE',  
  job: 'MANAGER',  
  mgr: 7839,  
  hireDate: 1981-05-01T00:00:00.000Z,  
  sal: 2850,  
  comm: null,  
  deptno: 30  
}
```



## Workshop 4

### Big Data

**Find all employees whose commission is less than 1000:**

```
> db.emp.find( {comm: {$lt:1000}} )
< {
  _id: ObjectId('67df7590688f9e7b0614d8c7'),
  empNo: 7499,
  eName: 'ALLEN',
  job: 'SALESMAN',
  mgr: 7698,
  hireDate: 1995-02-20T00:00:00.000Z,
  sal: 1600,
  comm: 300,
  deptno: 30
}
{
  _id: ObjectId('67df7590688f9e7b0614d8cb'),
  empNo: 7844,
  eName: 'TURNER',
  job: 'SALESMAN',
  mgr: 7698,
  hireDate: 1991-09-08T00:00:00.000Z,
  sal: 1500,
  comm: 0,
  deptno: 30
}
```

Workshop 4  
Big Data

**For example, find all employees who start after the 1st January 2000:**

```
> db.emp.find( {hiredate: {$lt: new Date("2000-01-01")}} )  
<
```

## Workshop 4

### Big Data

#### Updating a Collection

```
> db.dept.updateOne(
  {'deptno':40},
  {$set: {'loc': 'WOLVERHAMPTON'}}
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db> |
```

Check the changes have been made:

```
> db.dept.find({"deptno":40}).pretty()
< {
  _id: 40,
  deptno: 40,
  dname: 'OPERATIONS',
  loc: 'WOLVERHAMPTON',
  deptName: 'COMPUTING'
}
{
  _id: 50,
  deptno: 40,
  dname: 'OPERATIONS V2',
  loc: 'BOSTON'
}
```

## Workshop 4

### Big Data

#### Exercise 2.4

- 2.4.1 Update the name of department 40 to: COMPUTING

```
> db.dept.updateOne(
  { deptno: 40 },
  { $set: { deptName: "COMPUTING" } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

- 2.4.1 Update the salary of employee number 7788 in department 20 to 3500

```
> db.dept.updateOne(
  { empNo: 7654, deptno: 30 },
  { $set: { sal: 3500 } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-ezozlv-shard-0 [primary] hck-db>
```

## Workshop 4

### Big Data

#### Aggregation Pipeline

```
> db.deptCollection.aggregate([ {
    $project: {
        empSet: {
            $filter: {
                input: "$employees",
                as: "employee",
                cond: { $gte: [ "$$employee.sal", 2000 ] }
            }
        }
    }
}
]).pretty()
```

#### Count

```
> db.deptCollection.aggregate({
  "$project": {
    "deptno": 1,
    "Count": { "$size": { "$ifNull": [ "$employees", [] ] } }
  }
})
```

## Workshop 4

### Big Data

## Workshop 4

### Big Data

## Workshop 4

### Big Data