relational operators

find all SALESMAN

```
select * from emp where job = 'SALESMAN'
db.emp.find({ job: 'SALESMAN' })
```

find all CLERK

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

find all emps in dept 30

```
db.emp.find({ deptno: { $eq: 30 }})
db.emp.find({ deptno: 30 })
```

find all emps not in dept 30

```
db.emp.find({ deptno: { $ne: 30 }})
```

find all emps whose sal < 2500

```
db.emp.find({ sal: { $lt: 30 } })
```

find all emps whose sal > 2500

```
db.emp.find({ sal: { $gt: 30 } }).pretty()
```

find all emps who have comm field

```
db.emp.find({ comm: { $exists: true } })
```

find all emps who do not have comm field

```
db.emp.find({ comm: { $exists: false } })
```

find all emps who have comm = null

```
db.emp.find({ comm: { $eq: null } })
db.emp.find({ comm: null })
```

find all emps whose name starts with 'M'

```
db.emp.find({ ename: /^M/ })
db.emp.find({ ename: /^m/i })
```

find all emps whose name contains 'A' twice

```
db.emp.find({ ename: /A.*A/ })
```

find all emps whose name ends with S, the comparison should be case insensitive

```
db.emp.find({ ename: /s$/i })
```

find emp whose name is JAMES or MILLER

```
db.emp.find({ ename: { $in: ['JAMES', 'MILLER'] } })
```

find emp who is not SALESMAN, MANAGER or PRESIDENT

```
db.emp.find({ job: { $nin: ['SALESMAN', 'MANAGER', 'PRESIDENT'] } })
```

---

logical operators

find emps having sal more than 4000 or they are ANALYST

```
db.emp.find({ $or: [
{ sal: { $gt: 4000 } },
{ job: { $eq: 'ANALYST' } }
]})
```

find emps which are not in dept 20 and not SALESMAN

```
db.emp.find({ $and: [
{ deptno: { $ne: 20 } },
{ job: { $ne: 'SALESMAN' } }
]})
```

find all MANAGER in dept 30 or all SALESMAN in dept 30 having sal <= 1500

```
db.emp.find({ $or: [
{ $and: [
{ job: 'MANAGER' },
{ deptno: 30}
]},
{ $and: [
{ job: 'SALESMAN' },
{ deptno: 30},
{ sal: { $lte: 1500 } }
]}
]})
```

```
db.emp.find({ $and: [
{ job: 'MANAGER' },
{ deptno: 30}
]})
```

```
db.emp.find({ $and: [
{ job: 'SALESMAN' },
{ deptno: 30},
{ sal: { $lte: 1500 } }
]})
```

---

projection

display emp details _id, ename, job and sal

```
db.emp.find({}, { ename: 1, job: 1, sal: 1, _id: 1 })
db.emp.find({}, { ename: 1, job: 1, sal: 1 })
```

display emp details except mgr, sal, comm, job

```
db.emp.find({}, { mgr: 0, sal: 0, comm: 0, job: 0 })
```

display emp details ename, mgr

```
db.emp.find({}, { ename: 1, mgr: 1, _id: 0})
```

display emp details _id, ename, deptno, sal; but skip mgr, job, comm

```
db.emp.find({}, { ename: 1, sal: 1, deptno: 1})
db.emp.find({}, { mgr: 0, job: 0, comm: 0})
```

display emp details ename, deptno, sal without _id

```
db.emp.find({}, { ename: 1, deptno: 1, sal: 1, _id: 0 })
```

display emp ename where sal >= 2500

```
db.emp.find({ sal: { $gte: 2500 } }, { ename: 1, _id: 0 })
```

Aggregation pipeline

sum of sal per job

```
db.emp.aggregate([
{
// find the total salary group by job
$group: {
_id: '$job',
salary: { $sum: '$sal' }
}
}
])
```

sum of sal per job and sort them by total salary

```
db.emp.aggregate([
{
// find the total salary group by job
$group: {
_id: '$job',
totalSalary: { $sum: '$sal' }
}
},
{
// sort the total salary in ASC order
$sort: { totalSalary: 1 }
},
{
// get only the first record
$limit: 1
}
])
```

sum and avg of sal per job

```
db.emp.aggregate([
{
// find the total and avg salary group by job
$group: {
_id: '$job',
totalSalary: { $sum: '$sal' },
averageSalary: { $avg: '$sal' }
}
}
])
```

avg of sal per dept

```
db.emp.aggregate([
{
// find the total salary group by deptno
$group: {
_id: '$deptno',
avgSalary: { $avg: '$sal' }
}
}
])
```

print total sal, avg sal, max sal, min sal per job

```
db.emp.aggregate([
{
// find the total, avg, max and min salary group by job
$group: {
_id: '$job',
totalSalary: { $sum: '$sal' },
avgSalary: { $avg: '$sal' },
maxSalary: { $max: '$sal' },
minSalary: { $min: '$sal' }
}
}
])
```

print all jobs for which total sal is more than 5700

```
db.emp.aggregate([
{
// find the total salary group by job
$group: {
_id: '$job',
totalSalary: { $sum: '$sal' }
}
},
{
// find the total salary > 5700
$match: {
totalSalary: { $gt: 5700 }
}
}
])
```

display depts total sal in desc order

```
db.emp.aggregate([
{
// find the total salary group by deptno
$group: {
_id: '$deptno',
totalSalary: { $sum: '$sal' }
}
},
{
// sort the total salary in ASC order
$sort: { totalSalary: -1 }
}
])
```

find the dept that spends max on sal

```
db.emp.aggregate([
{
// find the total salary group by deptno
$group: {
_id: '$deptno',
totalSalary: { $sum: '$sal' }
}
},
{
```

```
// sort the total salary in ASC order
$sort: { totalSalary: -1 }
},
{
// find the maximum total salary
$limit: 1
}
])
```

display ename, deptno & sal of all emps whose sal >= 2500

```
db.emp.find({ sal: { $gte: 2500 } }, { ename: 1, deptno: 1, sal: 1, _id: 0 })
```

```
db.emp.aggregate([
{
// find the emp whose salary >= 2500
$match: {
sal: { $gte: 2500 }
}
},
{
// select the details need to be displayed
$project: {
ename: 1, deptno: 1, sal: 1, _id: 0
}
}
])
```

display ename, deptno & sal of all emps whose sal >= 2500 in the DESC order of sal

```
db.emp.aggregate([
{
// find emp with sal >= 2500
$match: {
sal: { $gte: 2500 }
}
},
{
// sort the salary in DESC order
$sort: { sal: -1 }
},
{
// select ename, deptno, sal
$project: {
ename: 1, deptno: 1, sal: 1, _id: 0
}
}
])
```

analyse data per dept per job [find the count of emps per deptno]

```
[select * from emp group by deptno, job]
db.emp.aggregate([
{
// find the emp per dept per job
$group: {
_id: {
deptno: '$deptno',
job: '$job'
},
```

```
    // get the count of them
    count: { $sum: 1 }
  }
}
```

])

find the job with max AVG sal

```
db.emp.aggregate([
{
$group: {
_id: '$job',
avgSalary: { $avg: '$sal' }
}
},
{ $sort: { avgSalary: -1 } },
{ $limit: 1 }
])
```

find number of managers, analysts and clerks in company

```
db.emp.aggregate([
{
// find the jobs
$match: {
job: { $in: ['MANAGER', 'ANALYST', 'CLERK'] }
}
},
{
// get the count by job
$group: {
_id: '$job',
count: { $sum: 1 }
}
}
])
```

print ename and dept name and dept location

```
db.emp.aggregate([
{
// find the dept details based on deptno
$lookup: {
from: 'dept',
localField: 'deptno',
foreignField: '_id',
as: 'deptinfo'
}
},
{
// take the deptinfo out of the array
$unwind: '$deptinfo'
},
{
// add new fields
$addFields: {
deptName: '$deptinfo.dname',
deptLocation: '$deptinfo.loc'
}
},
{
// select the required fields
$project: {
ename: 1, deptName: 1, deptLocation: 1, _id: 0
}
}
])
```

print depts and emps in that dept

```
db.dept.aggregate([
{
// get all the emp based on deptno from emp
$lookup: {
from: 'emp',
localField: '_id',
foreignField: 'deptno',
as: 'empInfo'
}
},
{
// sort on the basis of ename
$sort: { 'empInfo.ename': 1 }
}
])
```

print emp name and his manager details [final result -> { ename: '', manager: '' }]

```
db.emp.aggregate([
{
// self join
$lookup: {
from: 'emp',
localField: 'mgr',
foreignField: '_id',
as: 'manager'
}
},
{
// take the manager object out of the array
$unwind: '$manager'
},
{
// add a temporary field for manager name
$addFields: {
managerName: '$manager.ename'
}
},
{
// select the employee name and manager name
$project: {
ename: 1, managerName: 1, _id: 0
}
}
])
```