db.customer.find()

db.customer.insert([

{cid:1200,cname:"eee"},

{cid:1201,cname:"yy",phone:[43636,675765]},

{cid:1202,cname:"yy",email\_id:["abc@gmail","ddd@gmail"]},

{cid:1203,cname:"gt"}

])

/\*

/\* address is a embedded document that is a

document within a document\*/

db.customer.insert(

{cid:1205,cname:"mmm",

address:{flatno:209,apartment:"ABC",city:"Noida",state:"up"}

})

db.customer.find()

key:[val,val,val]

address:[temp:]

hobbies:["chess","criket"]

db.customer.insert(

{

cid:1206,

phone:[686876],

hobbies:[{indoor:["chess","tv"],outdoor:["cycling","criket"]}]

})

db.customer.insert({

cid:1206,

cname:"nnn",

address:

temp:{flatno:209,apartment:"ABC",city:"pune",

state:"Maharashtra"},

perment:{flatno:20,apartment:"AB",city:"kolhapur",

state:"Maharashtra"}}

})

db.customer.insert({cid:1305,cname:"nikhil",

address:{flatno:120,appname:"abc",city:"pune"}

})

db.customer.insert({\_id:20,cid:1232,cname:"rishi",

address:

{current:{city:"pune",state:"Maha"},

perm:{city:"mumbai",state:"Maha"}}

})

db.customer.find().pretty()

db.customer.insert({cid:3000,cname:"pp",phone:[765757,68768],

address:[

{temp:{city:"pune",state:"Maha"}},

{perm:{city:"mumbai",state:"Maha"}}

]

})

db.item.insert([{itemid:11},{itemid:12},{itemid:13}])

use pgdac

/\* 1. display all documents of employees \*/

db.employees.find()

db.employees.find().pretty()

/\*2. Display the first 5 document of employees collection\*/

db.employees.find().limit(5)

/\*3. Display the first document of employees collection\*/

db.employees.find().limit(1)

db.employees.findOne()

/\*

3.display the first name,hiredate and department id

of all employees.

select firstname,hiredate,department id

from employees.

all rows but specific columns

all document for specific keys

db.employees.find({conditions},{key lists})\*/

/\* empty {} indicates all docment, no filter on document\*/

db.employees.find({},

{FIRST\_NAME:1,HIRE\_DATE:1,DEPARTMENT\_ID:1})

/\* exclude the \_id key\*/

db.employees.find({},

{FIRST\_NAME:1,HIRE\_DATE:1,DEPARTMENT\_ID:1,\_id:0})

/\* skip the n document from the beginning, and display n documents\*/

db.employees.find({},

{FIRST\_NAME:1,HIRE\_DATE:1,DEPARTMENT\_ID:1,\_id:0}).skip(3)

db.employees.find().skip(3).limit(5)

/\* sorting of documents

order by in Select query

\*/

/\*select \* from employees order by salary;\*/

db.employees.find().sort({SALARY:1})

/\*select \* from employees order by department id in descending order;\*/

db.employees.find().

sort({DEPARTMENT\_ID:-1})

/\*select first\_name,salary

from employees

order by salary;\*/

db.employees.find({},{FIRST\_NAME:1,SALARY:1,\_id:0}).

sort({SALARY:1})

/\*select first\_name,salary

from employees

order by salary desc\*/

db.employees.find({},{FIRST\_NAME:1,SALARY:1,\_id:0}).

sort({SALARY:-1})

/\*select first\_name,salary

from employees

order by department\_id asc,salary desc\*/

db.employees.find({},

{FIRST\_NAME:1,DEPARTMENT\_ID:1,SALARY:1,\_id:0}).

sort({DEPARTMENT\_ID:1,SALARY:-1})

/\* select first\_name,salary

from employees

order by department\_id asc,salary desc

limit 1\*/

db.employees.find({},

{FIRST\_NAME:1,DEPARTMENT\_ID:1,SALARY:1,\_id:0}).

sort({DEPARTMENT\_ID:1,SALARY:-1}).

limit(10)

/\* filter documents based on a condition/multi conditions

select \* from employees

where department\_id=30

db.employees.find({},{key list})

db.employees.find({condition}) all keys will be displayed\*/

db.employees.find({DEPARTMENT\_ID:30})

/\* select firstname,deptid,SALARY

from employees

where department\_id=30\*/

db.employees.find({DEPARTMENT\_ID:30},

{FIRST\_NAME:1,DEPARTMENT\_ID:1,SALARY:1,\_id:0})

/\* select firstname,deptid,SALARY

from employees

where department\_id=30

order by salary\*/

db.employees.find({DEPARTMENT\_ID:30},

{FIRST\_NAME:1,DEPARTMENT\_ID:1,SALARY:1,\_id:0}).

sort({SALARY:1})

/\* 1.display lastname,job id and their joining

for the employees who works as a it programmer.\*/

db.employees.find(

{JOB\_ID:"IT\_PROG"},

{LAST\_NAME:1,JOB\_ID:1,HIRE\_DATE:1}

)

/\*2. display firstname,job id and employeeid

for the employees who works under employee 100.\*/

db.employees.find(

{MANAGER\_ID:100},

{LAST\_NAME:1,JOB\_ID:1,EMPLOYEE\_ID:1}

)

/\* 3. Display the detail of employees

whose alary is greater than 7000

comparison operators

equal to -> :

greater than -> $gt

greater than equal to-> $gte

less than -> $lt

less than equal to-> $gte

not eqyal to -> $ne

\*/

db.employees.find({SALARY:{$gte:9000}})

db.employees.find({SALARY:{$lte:9000}})

/\* display the employee name and salary

who belongs to dept 50 and thier salary is greater than 5000.

\*/

db.employees.find(

{DEPARTMENT\_ID:50,SALARY:{$gte:5000}},

{FIRST\_NAME:1,SALARY:1})

/\* $and operator

{$and:[{condition1},{condition 2}]}

\*/

db.employees.find({

$and:

[

{DEPARTMENT\_ID:50},

{SALARY:{$gte:5000}}

]},

{FIRST\_NAME:1,SALARY:1,DEPARTMENT\_ID:1,\_id:0})

/\* Display the employees who works either in dept 60 or 50

select \* from employees

where dept\_id=50 or dept\_id=60\*/

db.employees.find({

$or:

[

{DEPARTMENT\_ID:50},

{DEPARTMENT\_ID:60}

]},

{FIRST\_NAME:1,SALARY:1,DEPARTMENT\_ID:1,\_id:0})

/\* the above query can be written using the $in operator\*/

db.employees.find({DEPARTMENT\_ID:{$in:[30,50,60,400]}},

{FIRST\_NAME:1,SALARY:1,DEPARTMENT\_ID:1,\_id:0})

/\* display the list of employees who works in dept 30 or 50

and their salary is less than 5000.

select first name,deptid,salary

from employees

where deptid in(30,50)

and salary < 5000 \*/

db.employees.find({$and:

[

{DEPARTMENT\_ID:{$in:[30,50]}},

{SALARY:{$lt:5000}}

]

})

/\* Display the list of employees who do not in dept 30 or 50

and their salary is less than 5000.

select first name,deptid,salary

from employees where deptid not in(30,50)

and salary < 5000 \*/

db.employees.find({$and:

[

{DEPARTMENT\_ID:{$nin:[30,50]}},

{SALARY:{$lt:10000}}

]

})

/\* display the employee's name and

salary whose salary is in the range of 5000 to 10000.\*/

db.employees.find({$and:

[{SALARY:{$gte:5000}},{SALARY:{$lte:10000}}]

})

db.employees.find({$and:

[{SALARY:{$gte:5000,$lte:10000}}]

})

/\* Display the name of employees whose name starts sith S \*/

db.employees.find({FIRST\_NAME:/^S/})

/\* Display the name of employees whose name starts sith Sh \*/

db.employees.find({FIRST\_NAME:/^Sh/})

/\* Display the name of employees whose name ends with a \*/

db.employees.find({FIRST\_NAME:/a$/})

/\* Display the name of employees whose name an in between \*/

db.employees.find({FIRST\_NAME:/.\*an.\*/})

/\* update a document/ multiple doc/all document

update employees

set salary = 15000

where emp\_id=108 \*/

db.employees.update({EMPLOYEE\_ID:108},{$set:{SALARY:15000}})

db.employees.update({EMPLOYEE\_ID:400},{$set:{SALARY:10000}})

/\* but if no doc match is found then a new doc will be created\*/

db.employees.update(

{EMPLOYEE\_ID:103},

{$set:{expr:4}}

)

db.employees.update(

{JOB\_ID:"IT\_PROG"},

{$set:{SALARY:50000}},

{upsert:true})

db.employees.update(

{JOB\_ID:"IT\_PROG"},

{$set:{SALARY:50000}},

{multi:true})