**Dbms Lab chits**

**Chit 9,21,22) Collection “orderinfo“ which contains the documents given as below(Perform on Mongo Terminal)**

**{**

**cust\_id:123**

**cust\_name:”abc”,**

**status:”A”,**

**price:250**

**}**

->db.createCollection('orderinfo') db.orderinfo.insert([{cust\_id:123,cust\_name:'abc',status:'A',price:250},{cust\_id:1,cust\_name:'trapti',status:'A',price:50},{cust\_id:2,cust\_name:'sakshi',status:'B',price:1500},{cust\_id:3,cust\_name:'sayali',status:'B',price:500},{cust\_id:4,cust\_name:'pranjal',status:'A',price:90}])

**i. find the average price for each customers having status 'A'**

->db.orderinfo.aggregate({$match:{status:'A'}},{$group:{\_id:0, avgprice:{$avg:'$price'}}})

**ii. Display the status of the customers whose amount/price lie between 100 and 1000**

->db.orderinfo.find({price:{$gt:100,$lt:1000}},{\_id:0,status:1})

**iii. Display the customers information without “\_id” .**

-> db.orderinfo.find({},{\_id:0})

**iv. create a simple index on onderinfo collection and fire the queries.**

->db.orderinfo.find({price:{$gt:100,$lt:1000}}).explain();

db.orderinfo.createIndex({price:1});

db.orderinfo.find({price:{$gt:100,$lt:1000}}).explain()

v. **Add “Age” field to the orderinfo collection**

-> db.orderinfo.update({},{$set:{age:22}},{multi:true})

**vi. Create the complex index on orderinfo collection and fire the queries and drop the index.**

**->**db.orderinfo.createIndex({cust\_name:1,status:1})

db.orderinfo.getIndexes()

db.orderinfo.dropIndex({cust\_name:1,status:1})

**vii. Display the average price for each customer group by status**

**->** db.orderinfo.aggregate({$group:{\_id:'$status',avg\_price\_statuswise:{$avg:'$price'}}})

**viii. Change the customer’s name whose status is “B”**

**->** db.orderinfo.update({status:'B'},{$set:{cust\_name:'arya'}},{multi:true})

**ix.Display the name of the customer having the price between 250 and 450**

-> db.orderinfo.find({price:{$gte:250,$lte:450}},{\_id:0,cust\_name:1})

**x. Increment the price by 10 for cust\_id: 123 and decrement the price by 5 for cust\_id: 124**

**->** db.orderinfo.update({cust\_id:123},{$inc:{price:10}})

db.orderinfo.update({cust\_id:124},{$inc:{price:-5}})

**xi. Remove any one of the field from the orderinfo collection.**

-> db.orderinfo.update({},{$unset:{age:1}},{multi:true})

**xii. Find the name of the customer whose status is either A or price is 250 or both.**

**->** db.orderinfo.find({$or:[{status:'A'},{price:250}]},{\_id:0,cust\_name:1})

**Chit 13)Collection “movies“ which contains the documents given as below(Perform on Mongo Terminal)**

**{**

**name: “Movie1”,**

**type: “action”,**

**budget:1000000**

**producer:{**

**name: “producer1”,**

**address:”PUNE”**

**}**

**}**

**i. Find the name of the movie having budget greater than 1,00,000.**

-> db.movies.find({budget:{$gt:100000}},{name:1,\_id:0})

**ii. Find the name of producer who lives in Pune**

**->** db.movies.find({'producer.address':'pune'},{\_id:0,'producer.name':1})

**iii. Update the type of movie “action” to “horror”**

**->** db.movies.update({type:'action'},{$set:{type:'horror'}},{multi:true})

**iv. Find all the documents produced by name “producer1” with their address**

**->** db.movies.find({'producer.name':'p1'},{\_id:0,name:1,'producer.address':1}).pretty()

**Chit 14)Consider following structure for Mongodb collection and write a query for following requirements in Mongodb**

**Teachers (Tname,dno,Experience,Salary,Data\_of\_Joining)**

**Department (Dno,Dname)**

**Students(Sname,Roll\_No,Class)**

**1. Write a query to create above collection insert some sample documents.**

->

**2. Find the information about all teachers of Dno=2 and having salary greater than or equal to 10,000/-**

**->** db.teachers.find({dno:2,salary:{$gte:10000}})

**3. Find the student information having Roll\_no=2 or Sname='xyz'**

**->** db.students.find({$or:[{rno:2},{sname:'xyz'}]})

**4. Update student name whose Roll\_No=5**

**->** db.students.update({rno:5},{$set:{sname:'shreya'}},{multi:true})

**5. Delete all student whose Class is 'FE'**

**->** db.students.remove({class:'FE'})

**6. Apply index on Students Collection**

**->** db.students.createIndex({sname:1})

**Collection creation Student and insert following data in that:**

**Rollno:1,name:'Navin',subject:'DMSA',marks:78**

**Rollno:2,name:'anusha',subject:'OSD',marks:75**

**Rollno:3,name:'ravi',subject:'TOC',marks:69**

**Rollno:4,name:'veena',subject:'TOC',marks:70**

**Rollno:5,name:‘Pravini',subject:‘OSD',marks:80**

**Rollno:6,name: ‘Reena',subject: ‘DMSA',marks:50**

**Rollno:7,name:‘Geeta',subject:‘CN',marks:90**

**Rollno:8,name:‘Akash',subject:‘CN',marks:85**

**1. Write aggregate function to find Max marks of Each Subject.**

-> db.student.aggregate({$group:{\_id:'$subject',max\_marks\_subjectwise:{$max:'$marks'}}})

**2. Write aggregate function to find Min marks of Each Subject.**

**->** db.student.aggregate({$group:{\_id:'$subject',min\_marks\_subjectwise:{$min:'$marks'}}})

**3. Write aggregate function to find Sum of marks of Each Subject.**

**->** db.student.aggregate({$group:{\_id:'$subject',total\_marks\_subjectwise:{$sum:'$marks'}}})

**4. Write aggregate function to find Avg marks of Each Subject.**

**->** db.student.aggregate({$group:{\_id:'$subject',avg\_marks\_subjectwise:{$avg:'$marks'}}})

**5. Write aggregate function to find first record Each Subject.**

**->** db.student.aggregate({$group:{\_id:'$subject',first\_record\_subjectwise:{$first:'$$ROOT'}}})

**6. Write aggregate function to find Last record of Each Subject.**

**->** db.student.aggregate({$group:{\_id:'$subject',last\_record\_subjectwise:{$last:'$$ROOT'}}})

**7. Write aggregate function to find count number of records of each subject**

**->** db.student.aggregate({$group:{\_id:'$subject',total\_records\_subjectwise:{$sum:1}}})

**Chit 20) Use MongoDB**

**Indexing**

**1. Create Collection**

->

db.createCollection('stud')

**2. Insert some Documents**

**->** db.stud.insert([{name:'s1',rno:23,class:'TE'},{name:'s2',rno:2,class:'SE'},{name:'s4',rno:34,class:'BE'},{name:'s5',rno:28,class:'TE'}])

**3. Create Single Index**

-> db.stud.createIndex({class:1})

**4. Create Compound Index**

**->** db.stud.createIndex({name:1,rno:1})

**5. Create Unique on Collection**

**->** db.stud.createIndex({rno:1},{unique:true})

**6. Show Index Information**

**->** db.stud.getIndexes()

**7. Remove Index**

-> db.stud.dropIndex({rno:1})

db.stud.dropIndex({name:1,rno:1})

db.stud.dropIndex({class:1})

**chit 18)Collection “city “ which contains the documents given as below(Perform on Mongo Terminal)**

**{**

**city:”pune”,**

**type:”urban”,**

**state:”MH”,**

**population:”5600000”**

**}**

**-using mapreduce, find statewise population**

**->** var v1 = function(){emit(this.state,this.population)}

var v2 = function(key,values){return Array.sum(values)}

db.cities.mapReduce(v1,v2,{out:'State\_wise\_population'})

db.State\_wise\_population.find()

**-using mapreduce, find citywise population**

**->** var v1 = function(){emit(this.city,this.population)}

var v2 = function(key,values){return Array.sum(values)}

db.cities.mapReduce(v1,v2,{out:'city\_wise\_population'})

db.city\_wise\_population.find()

**-using mapreduce, find typewise population.**

**->** var v1 = function(){emit(this.type,this.population)}

var v2 = function(key,values){return Array.sum(values)}

db.cities.mapReduce(v1,v2,{out:'type\_wise\_population'})

db.type\_wise\_population.find()

**Chit 16) Perform aggregation and Indexing using mongodb on below database**

**1. Create a database department**

-> use department

**2. Create a collection as teacher with fields as name , department ,experience and salary**

**->** db.createCollection('teacher')

db.teacher.insert([{name:'t1',dept:'comp',exp:2,salary:10000},{name:'t2',dept:'entc',exp:5,salary:60000},{name:'t3',dept:'comp',exp:10,salary:75000},{name:'t4',dept:'entc',exp:7,salary:34000}])

**3. Display the department wise average salary.**

**->** db.teacher.aggregate({$group:{\_id:'$dept',dept\_wise\_avg\_salary:{$avg:'$salary'}}})

**4. Display the no. Of employees working in each department.**

**->** db.teacher.aggregate({$group:{\_id:'$dept',dept\_wise\_total\_employees:{$sum:1}}})

**5. Display the department wise minimum salary.**

**->** db.teacher.aggregate({$group:{\_id:'$dept',dept\_wise\_min\_salary:{$min:'$salary'}}})

**6. Apply index and drop index**

**->** db.teacher.createIndex({name:1})

db.teacher.dropIndex({name:1})

**chit 15)Use MongoDB**

**Create Institute Database and Create Student collection**

**1. RollNo 2. Student Name 3. Age 4. Branch 5. Address :{ City, State} 6. Hobbies (Array)**

**Perform following operations**

**1. Create database Institute.**

-> use institute

**2. Create collection Students.**

-> db.createCollection('students')

**3. Insert 10 documents with above mentioned structure.**

**->** db.students.insert([{rno:1,sname:'anushka',age:12,branch:'comp',address:{city:'pune',state:'mh'},hobbies:['sing','dance']},{rno:2,sname:'aman',age:18,branch:'comp',address:{city:'pune',state:'mh'},hobbies:['chess','sing']},{rno:3,sname:'pratik',age:20,branch:'entc',address:{city:'kochi',state:'kerala'},hobbies:['reading','dance']},{rno:4,sname:'trapti',age:17,branch:'mech',address:{city:'nagpur',state:'mh'},hobbies:['dancing','chess']},{rno:5,sname:'sayali',age:20,branch:'entc',address:{city:'surat',state:'gujrat'},hobbies:['v\_edit','reading']},{rno:6,sname:'pranjal',age:24,branch:'comp',address:{city:'pune',state:'mh'},hobbies:['cooking','singing']},{rno:7,sname:'ajit',age:14,branch:'mech',address:{city:'pune',state:'mh'},hobbies:['sing','dance']},{rno:8,sname:'riya',age:19,branch:'civil',address:{city:'indore',state:'mp'},hobbies:['drawing','singing']},{rno:9,sname:'sakshi',age:20,branch:'civil',address:{city:'nagar',state:'mh'},hobbies:['reading','drawing']},{rno:10,sname:'shreya',age:26,branch:'entc',address:{city:'ahmedabad',state:'gujrat'},hobbies:['v\_edit','chess']}])

**4. Display all students’ information.**

-> db.students.find()

**5. Display Student information whose age is greater than 15.**

-> db.students.find({age:{$gt:15}})

**6. Display Student information sorted on name field**

-> db.students.find().sort({sname:1})

**7. Update student branch Computer of RollNo 3.**

**->** db.students.update({rno:3},{$set:{branch:'comp'}})

**8. Remove document with RollNo 1**

-> db.students.remove({rno:1})

**9. Display Student information whose name starts with A**

-> db.students.find({sname:/^a/})

**10. Display the total numbers of documents available in collection.**

-> db.students.find({}).count()

**11. Display only first 5 documents.**

-> db.students.find().limit(5)

**12. Display all documents instead of first 3.**

-> db.students.find().skip(3)

**13. Display the name of Students who live in Pune City.**

**->** db.students.find({'address.city':'pune'},{\_id:0,sname:1})

**14. Display only Name of all students.**

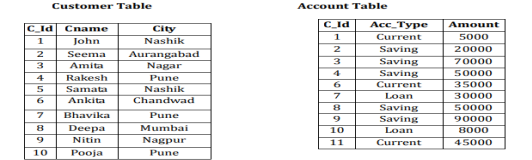
-> db.students.find({},{\_id:0,sname:1})

**15. Drop Collection**

-> db.students.drop()

**Chit 1) Use MySQL –( Joins and Subqueries)**

**Create Customer and Account table and add rows shown below**



**1. Show the cname, Acc\_Type, amount information of customer who is having an saving account.**

-> select cname, acctype, amount from cust c , acc a where c.cid=a.cid a

nd acctype='saving';

**2. Display the data using Natural, left and right join.**

-> select \* from cust c natural join acc;

select \* from cust c inner join acc a on c.cid=a.cid;

select \* from cust c left join acc a on c.cid=a.cid;

select \* from cust c right join acc a on c.cid=a.cid;

**3. Display the information of customers living in the same city as of ‘pooja’.**

**->**

select c.cid,cname,city,acctype,amount from cust c, acc a where c.cid = a.cid and city = (select city from cust where cname = 'pooja');

**4. Display the information of account, having less amount than average amount throughout the bank.**

**->** select c.cid , cname,city,acctype,amount from cust c , acc a where c.cid = a.cid and amount < (select avg(amount) from acc);

**5. Display the C\_id having maximum amount in account.**

-> select cid from acc where amount = (select max(amount) from acc);

**6. Display the amount and acc\_type of those customers whose amount is the minimum amount of that Acc\_type.**

-> select amount,acctype from acc where (acctype,amount) in (select acctype,min(amount) from acc group by acctype);

**7. Display the amount of those accounts whose amount is higher than amount of any saving account amount.**

-> select amount from acc where amount >Any (select amount from acc where acctype='saving');

**Chit 4**

**(Perform on MYSQL Terminal)**

**student(S\_ID,name,dept\_name,tot\_cred)**

**instructor(T\_ID,name,dept\_name,salary)**

**course(course\_id,title,dept\_name,credits)**

**i. Find the average salary of instructor in those departments where the average salary is more than Rs. 42000/-.**

-> select deptname , avg(salary) from instructor group by deptname havin

g avg(salary)>42000;

**ii. Increase the salary of each instructor in the computer department by 10%.**

-> update instructor set salary = salary + salary \* 0.1 where deptname='

comp';

**iii. Find the names of instructors whose names are neither ‘Amol’ nor ‘Amit’.**

-> select tname from instructor where tname not in ('amol','amit');

**iv. Find the names of student which contains ‘am’ as its substring.**

-> select sname from student where sname like '%am%';

**v. Find the name of students from computer department that “DBMS” courses they take.**

-> select sname from student s, course c where s.deptname = c.deptname a

nd title='dbms';

**Chit 8**

**(Perform on MYSQL Terminal)**

**teaches(T\_ID, course\_id, sec\_id, semester, year)**

**student(S\_ID, name, dept\_name, tot\_cred)**

**instructor(T\_ID, name, dept\_name, salary)**

**course(course\_id, title, dept\_name, credits)**

**i. Find the names of the instructor in the university who have taught the courses semester wise.**

->(not sure) select tname,semester from teaches t, instructor i where i.tid = t.tid group by semester,tname;

**ii. Create View on single table which retrieves student details.**

-> create view v1 as select \* from student;

**iii. Rename the column of table student from dept\_name to deptatrment\_name**

-> alter table student change column deptname department\_name varchar(20);

**iv. Delete student name whose department is NULL**

**-> (two ways)** update student set sname = null where department\_name is null;

Or

delete from student where department\_name is null;

**Chit 17**

**Use MySQL Create Employee table, Project table and add rows shown below**



**1. Find different locations from where employees belong to?**

-> select distinct address from employee;

**2. What is maximum and minimum salary?**

-> select max(salary) as maximum\_salary, min(salary) as minimum\_salary f

rom employee;

**3. Display the content of employee table according to the ascending order of salary amount.**

-> select \* from employee order by salary ;

(for descending) select \* from employee order by salary desc;

**4. Find the name of employee who lived in Nasik or Pune city.**

-> select ename from employee where address = 'nasik' or address='pune';

select ename from employee where address in ('pune','nasik');

**5. Find the name of employees who does not get commission.**

-> select ename from employee where commission is null;

**6. Change the city of Amit to Nashik.**

-> update employee set address = 'nasik' where ename='amit';

**7. Find the information of employees whose name starts with ‘A’.**

**->** select \* from employee where ename like 'a%';

**8. Find the count of staff from Mumbai.**

-> select count(\*) from employee where address='mumbai';

**9. Find the count of staff from each city**

-> select address,count(\*) as count\_of\_staff from employee group by address;

**10. Find the address from where employees are belonging as well as where projects are going on.**

-> select distinct address from employee , project where address = addr;

**11. Find city wise minimum salary.**

-> select address, max(salary) from employee group by address;

**12. Find city wise maximum salary having maximum salary greater than 26000**

-> select address, max(salary) from employee group by address having max(salary)>26000;

**13. Delete the employee who is having salary greater than 30,000.**

**->** delete from employee where salary>30000;

**Chit 19**

**Use MySQL**

**1)Create a table emp with following fields and constraints**

**Eno –(Constraint:- primary key and apply sequence starts with 101) ,Ename –(Constraint :- not null) Address ––(Constraint :-default ‘Nashik’) ,Joindate,**

-> create table emp(eno int primary key auto\_increment,ename varchar(20) not null, address varchar(20) default 'nashik', joindate date);

**2)After table creation add field - Post in the emp table.**

-> alter table emp add post varchar(20);

**3)Insert some data in emp table.Create Index on Ename field of employee table.**

-> insert into emp (eno,ename,address,joindate,post) values(101,'sakshi','nagar','2015-08-31','sde');

insert into emp (ename,address,joindate,post) values('sayali','paithan','2019-09-13','sde2'),('trapti','nagpur','2011-12-4','tester'),('pratik','thane','2014-04-19','manager');

create index i1 on emp(ename);

**4) Create View on employee table to show only Ename and Salary.**

**->** create view v2 as select ename,address from emp;

**Chit 24:**

**(Perform on MYSQL Terminal)**

**Emp(emp\_id,ename, street, city)**

**works(emp\_id,c\_id,ename, cname, sal)**

**Company(c\_id,cname, city)**

**Manager(mgr\_id, mgrname)**

**i. Modify the database so that a particular company (eg. ABC) now in Pune**

**->** update company set city='pune'where cname='abc';

**ii. Give all managers of Mbank a 10% raise. If salary is >20,000, give only 3% raise.**

**->** update works,manager set salary = salary + salary\*0.1 where ename=mname and salary <20000 and cname='abc';

update works,manager set salary = salary + salary\*0.03 where ename=mname and salary >20000 and cname='abc';

**iii. Find out the names of all the employees who works in ‘Bosch’ company in city Pune**

-> select e.ename from emp1 e , works w , company c where e.eid = w.eid and c.cid=w.cid and w.cname = 'bosch'and c.city='pune';

**iv. Delete all records in the works table for employees of a particular company (Eg, SBC Company)  whose salary>50,000.**

**->** delete from works where cname = 'bosch' and salary>50000;

**Chit 25:**

**(Perform on MYSQL Terminal)**

**Empl(e\_no, e\_name, post, pay\_rate)**

**Position(pos\_no, post)**

**Duty-alloc (pos\_no, e\_no, month,year, shift)**

**Implement the following SQL queries**

**i. Get duty allocation details for e\_no 123 for the first shift in the month of April 2003**

-> select \* from dutyalloc where eno=123 and shift=1 and month=4 and yea

r=2003;

**ii. Get the employees whose rate of pay is > or equal rate of pay of employees 'Sachin'.**

-> select \* from emp2 where payrate >= (select payrate from emp2 where ename='sachin');

**iii. Create a view for displaying minimum, maximum and average salary for all the posts.**

-> create view v3 as select post,min(payrate),max(payrate),avg(payrate) from emp2 group by post;

**iv. Get count of different employees on each shift having post ‘manager’.**

-> select shift,count(\*) from positions p , emp2 e, dutyalloc d where e.eno=d.eno and p.pno = d.pno and p.post = 'manager' group by shift;