

practical 1:implementation of DDL commands of SQL with suitable example.

1.create table

2.alter table

3.drop table

#Create a table with name Department:

SQL: create table Dept(

Deptno int,

Dname varchar(10),

loc varchar(4),

primary key(Deptno)

);

OUTPUT: Table created.

#Rename the table Dept with Department:

SQL: Alter table Dept Rename to Department;

OUTPUT:

Table altered.

#To insert the values in the table:

SQL: insert into Department(Deptno, Dname,loc)values(1,'Development','nash');

insert into Department(Deptno, Dname,loc)values(2,'Testing','pune');

insert into Department(Deptno, Dname,loc)values(3,'production','pimp');

insert into Department(Deptno, Dname,loc)values(4,'management','pune');

insert into Department(Deptno, Dname,loc)values(5,'marketing','goa');

insert into Department(Deptno, Dname,loc)values(6,'HR','USA');

insert into Department(Deptno, Dname,loc)values(7,'Sales','UK');

insert into Department(Deptno, Dname,loc)values(8,'Sr.maneger','mumb');

OUTPUT: 1 row(s) inserted.

#Display the Table:

SQL: select *from Department;

OUTPUT:

DEPTNO	DNAME	LOC
1	Development	nash
2	Testing	pune
3	production	pimp
4	management	pune
5	marketing	goa
6	HR	USA
7	Sales	UK
8	Sr.maneger	mumb

#Add the new column pincode with not null constraint to the existing thable Department:

SQL: Alter table Department Add(pincode number(6));

OUTPUT: desc Department;

Table	Column Nullable	Data Type Default	Length Comment	Precision	Scale	Primary Key
DEPARTMENT	-	DEPTNO	Number	-	-	0 1 -
	-	DNAME	Varchar2 10	-	-	- nullable -
	-	LOC	Varchar2 4	-	-	- nullable -

PINCODE	Number	-	6	0	-	nullable	-
-							

1 - 4

#Drop table location:

SQL: Alter table Department drop column loc cascade constraints;

OUTPUT:

Table dropped.

#Rename Dname to DeptName

SQL:

Alter table Department Rename column Dname to deptName;

OUTPUT:

Table altered.

SQL:

desc Department;

OUTPUT:

Table	Column	Data Type	Length	Precision	Scale	Primary Key		
	Nullable	Default	Comment					
DEPARTMENT		DEPTNO	Number	-	-	0	1	-
	-	-						
		DEPTNAME	Varchar2	10	-	-	nullable	-
	-							
		PINCODE	Number	-	6	0	-	nullable
	-							

1 - 3

#Delete The table

SQL: drop table Department;

OUTPUT:

Table dropped.

practical no:2 implementation of DML commands of SQL with suitable examples.

1.insert

2.update

3.delete

1. Create a table EMPLOYEE with following schema:

(Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary)

SQL:

create table EMPLOYEE1(

EMP_NO INT PRIMARY KEY NOT NULL,

E_NAME VARCHAR(20),

```

E_ADDRESS VARCHAR(20),

E_PH NUMBER(10),

DEPT_NO VARCHAR(10),

DEPT_NAME VARCHAR(15),

JOB_ID CHAR(10),

SALARY NUMBER(20)

);

```

2.Insert values into the table

SQL:

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(1,'RAM','Nashik',2545256,1,'development','Jr',25000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(2,'Raghu','Pune',2548541,2,'testing','Sr',55000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(3,'Raghunat','Mumbai',5147896,3,'Networking','Sr',40000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(4,'vishnu','Navi Mumbai',4225864,4,'Technical','Jr',60000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(5,'Swami','solapur',2458751,5,'management','Sr',70000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(6,'Sujal','sinnar',9855123,6,'web design','Jr',80000);

```

```

insert into EMPLOYEE1( EMP_NO,E_NAME,E_ADDRESS, E_PH, DEPT_NO,DEPT_NAME,JOB_ID,
SALARY)VALUES(7,'tejas','Germany',7542587,7,'Management','Sr',90000);

```

3.Display Table:

EMP_NO	E_NAME	E_ADDRESS	E_PH	DEPT_NO	DEPT_NAME	JOB_ID	SALARY
--------	--------	-----------	------	---------	-----------	--------	--------

1	RAM	Nashik	2545256	1	development	Jr	25000
2	Raghu	Pune	2548541	2	testing	Sr	55000
3	Raghunat	Mumbai	5147896	3	Networking	Sr	40000
4	vishnu	Navi Mumbai	4225864	4	Technical	Jr	60000
5	Swami	solapur	2458751	5	management	Sr	70000
6	Sujal	sinnar	9855123	6	web design	Jr	80000

4.UPDATE A ROW IN A TBAL

SQL

UPDATE EMPLOYEE1

SET E_NAME = 'Tejas', DEPT_NAME= 'Jr'

WHERE EMP_NO = 5;

OUTPUT

EMP_NO	E_NAME	E_ADDRESS	E_PH	DEPT_NO	DEPT_NAME	JOB_ID	SALARY
1	RAM	Nashik	2545256			1	development
	Jr	25000					
2	Raghu	Pune	2548541	2	testing	Sr	55000
3	Raghunat	Mumbai	5147896	3	Networking	Sr	40000
4	vishnu	Navi Mumbai	4225864	4	Technical	Jr	60000
5	Tejas	solapur	2458751	5	Jr	Sr	70000
6	Sujal	sinnar	9855123	6	web design	Jr	80000

5.Select query:

SQL:

select E_NAME , SALARY from EMPLOYEE1

OUTPUT:

E_NAME	SALARY
RAM	25000
Raghu	55000
Raghunat	40000

vishnu	60000
Tejas	70000
Sujal	80000

#Delete The table

SQL: drop table EMPLOYEE1;

OUTPUT:

Table dropped.

Practical 3rd: implementation of different types of function with suitable example.

1.number function

2.aggregate function

3.character function

4.conversion function

5.date function

1st

List the E_no, E_name, Salary of all employees working for MANAGER

select eno,ename,salary

from emp1

where designation='Manager';

ENO	ENAME	SALARY
-----	-------	--------

1	Hrishikesh	15000
---	------------	-------

2	Soham	25000
---	-------	-------

2nd

Display all the details of the employee whose salary is more than the Sal of any IT PROFF..

```
select * from emp1
where salary>(select salary from emp1 where dname='IT')
```

3rd

List the employees in the ascending order of Designations of those joined after 1981.

```
select * from emp1
where doj>'01-jan-1981'
order by designation
```

ENAME

Ramesh

Rohan

mahesh

sumit

tanmay

Soham

Hrishikesh

4th

List the employees along with their Experience and Daily Salary.

```
select ename,salary/30 as dailysal,SYSDATE - doj as expiriance from emp1
```

ENAME DAILYSAL EXPIRIANCE

Hrishikesh 500 4210.48289

Soham 833.333333 4210.48289

Mohit	333.333333	24664.4829
Rohan	833.333333	1228.48289
Ramesh	700	863.482894
sumit	700	15412.4829
tanmay	700	15196.4829
mahesh	700	15182.4829
mahesh	700	15880.4829

5th

List the employees who are either 'CLERK' or 'ANALYST' .

select * from emp1

where designation='Clerk'

or designation='Analyst'

ENO	ENAME	ADDRESS	PHONE	DEPTNO
4	Rohan	Mumbai	113455662	4
IT		102 Clerk	25000	01-MAR-20
5	Ramesh	Pune	113452662	5
IT		103 Analyst	21000	01-MAR-21
6	sumit	Pune	113456662	5
Management		104 Clerk	21000	01-MAY-81

ENO ENAME ADDRESS PHONE DEPTNO

DNAME JOBID DESIGNATION SALARY DOJ

7 tanmay Pune 113456602 3

Sales 104 Clerk 21000 03-DEC-81

8 mahesh Pune 113456601 4

Purchase 104 Clerk 21000 17-DEC-81

6th

List the employees who joined on 1-MAY-81, 3-DEC-81, 17-DEC-81,19-JAN-80 .

select * from emp1

where doj in ('1-MAY-81', '3-DEC-81', '17-DEC-81','19-JAN-80')

order by doj

ENO ENAME ADDRESS PHONE DEPTNO

DNAME JOBID DESIGNATION SALARY DOJ

9 mahesh Pune 113456601 4

Purchase 104 Accountant 21000 19-JAN-80

6 sumit Pune 113456662 5

Management 104 Clerk 21000 01-MAY-81

7 tanmay Pune 113456602 3

Sales 104 Clerk 21000 03-DEC-81

8 mahesh Pune 113456601 4

Purchase 104 Clerk 21000 17-DEC-81

7th

List the employees who are working for the Deptno 4 or 5.

select ename from emp1 where deptno=4 or deptno=5

ENAME

Rohan

Ramesh

sumit

maresh

maresh

8th

List the Enames those are starting with 'S' .

select ename from emp1

where ename like 'S%

ENAME

Soham'

9th

Display the name as well as the first five characters of name(s) starting with 'H'

```
select ename from emp1
```

```
where ename like 'H%'
```

```
ENAME
```

```
-----
```

```
Hrishikesh
```

```
10
```

List all the emps except 'PRESIDENT' & 'MGR' in asc order of Salaries.(JOA,ANALY MENTION IN RECORDS)

```
select ename from emp1
```

```
where designation not in('Joa','Analyst')
```

```
order by salary
```

```
ENAME
```

```
-----
```

```
Ramesh
```

```
Rohan
```

```
mahesh
```

```
sumit
```

```
tanmay
```

```
Soham
```

```
Hrishikesh
```

practical 4:implementation of different types of operators in sql

1.arithmetic operator

2.logical operator

3.comparison operator

4.special operator

5.set operator

```
create table emp2(eid int primary key,ename varchar(20),deptid int references dept(deptid))
```

```
insert into emp2 values(1,'Mohit',1);
```

```
select * from emp2
```

1st: Display all the dept numbers available with the dept and emp tables avoiding duplicates.

```
select deptid from dept union select deptid from emp2
```

DEPTID

1

2

3

4

5

6

7

8

9

10

2nd:Display all the dept numbers available with the dept and emp tables.

```
select deptid from emp2 union all select deptid from dept
```

DEPTID

1

1
3
3
4
5
5
1
2
3
4
5
6
7
8
9
10

3rd:Display all the dept numbers available in emp and not in dept tables and vice versa.

select deptid from dept minus select deptid from emp2

DEPTID

2
6
7
8
9
10

select deptid from emp2 minus select deptid from dept

no rows selected

practical 5th:implementation of different types of joins.

1.inner join

2.outer join

3.natural join...etc

create table sailor(sid int primary key, sname varchar(20), rating int, age int)

create table boats (bid int primary key, bname varchar(20), color varchar(20))

create table reserve(sid int references sailor(sid), bid int references boats(bid), day date)

1. Find all information of sailors who have reserved boat number 101

select *

from sailor s

left join reserve r on s.sid=r.sid

where r.bid=101

SID	SNAME	RATING	AGE	SID	BID	DAY
101	Jhon	8		32	101	101 14-JAN-23
104	Wood	9		21	104	101 18-JAN-23

2. Find the name of boat reserved by Bob.

SELECT b.bname

FROM sailor s

JOIN reserve r ON s.sid = r.sid

JOIN boats b ON r.bid = b.bid

WHERE s.sname = 'Bob';

BNAME

Big boat

Small_ship

Small_ship

3.Find the names of sailors who have reserved a red boat, and list in the order of age

SELECT s.sname

FROM sailor s

JOIN reserve r ON s.sid = r.sid

JOIN boats b ON r.bid = b.bid

WHERE b.color = 'red'

order by age';

SNAME

Bob

4.Find the names of sailors who have reserved at least one boat.

SELECT distinct s.sname

FROM sailor s

LEFT JOIN reserve r ON s.sid = r.sid

SNAME

Wood

Bob

Jhon

David

5. Find the ids and names of sailors who have reserved two different boats on the same day.

```
SELECT s.sid, s.sname
```

```
FROM sailor s
```

```
JOIN reserve r1 ON s.sid = r1.sid
```

```
JOIN reserve r2 ON s.sid = r2.sid AND r1.day = r2.day AND r1.bid <> r2.bid
```

```
SID SNAME
```

```
-----
```

```
101 Jhon
```

```
102 Bob
```

```
101 Jhon
```

```
102 Bob
```

6. Find the ids of sailors who have reserved a red boat or a green boat.

```
select distinct sailor.sid,sname from sailor,boats,reserve where boats.bid=reserve.bid and  
sailor.sid=reserve.sid and color='red' or color='green'
```

```
SID SNAME
```

```
-----
```

```
102 Bob
```

```
103 David
```

```
101 Jhon
```

```
104 Wood
```

7. Find the name and the age of the youngest sailor.

```
SELECT s.sname, s.age
```

```
FROM sailor s
```

WHERE s.age = (SELECT MIN(age) FROM sailor)

SNAME	AGE
-------	-----

Wood	21
------	----

8. Count the number of different sailor names.

SELECT COUNT(DISTINCT s.sname)

FROM sailor s;

COUNT(DISTINCTS.SNAME)

4

9. Find the average age of sailors for each rating level.

SELECT rating, AVG(age) AS average_age

FROM sailor

GROUP BY rating;

RATING	AVERAGE_AGE
--------	-------------

8	29
---	----

7	46
---	----

9	21
---	----

10. Find the average age of sailors for each rating level that has at least two sailors.

SELECT rating, AVG(age) AS average_age

FROM sailor

GROUP BY rating

HAVING COUNT(sid) >= 2

RATING	AVERAGE_AGE
--------	-------------

Practical 6: study and implementation of

1. group by and having clause

2. orderby clause

3. indexing

1. Display total salary spent for each job category.

select sum(salary), dname from dept, emp2 where dept.deptid=emp2.deptid group by dname

SUM(SALARY) DNAME

40000 Administration

40000 Purchasing

12000 Human Resources

38000 IT

2. Display lowest paid employee details under each manager.]

select min(salary) from dept, emp2 where dept.deptid=emp2.deptid group by manager

MIN(SALARY)

20000

12000

14000

3. Display number of employees working in each department and their department name

select count(*), dname from emp2, dept where dept.deptid=emp2.deptid group by dname

COUNT(*) DNAME

2 Administration

2 Purchasing

1 Human Resources

2 IT

4. Display the details of employees sorting the salary in increasing order

select * from emp2 order by(salary) desc

EID	ENAME	DEPTID	SALARY
-----	-------	--------	--------

7	rahul	5	24000
---	-------	---	-------

2	sumit	1	20000
---	-------	---	-------

3	rohan	3	20000
---	-------	---	-------

1	Mohit	1	20000
---	-------	---	-------

4	mohan	3	20000
---	-------	---	-------

6	soham	5	14000
---	-------	---	-------

5	khushi	4	12000
---	--------	---	-------

5. Show the record of employee earning salary greater than 16000 in each department.

select * from dept,emp2 where dept.deptid=emp2.deptid and salary>16000 group by dname group by

DEPTID	DNAME	LOC	PINCODE
--------	-------	-----	---------

MANAGER	EID	ENAME	DEPTID	SALARY
---------	-----	-------	--------	--------

1 Administration	Nashik	422007
Soham	1 Mohit	1 20000

3 Purchasing	Nashik	422007
Mohit	3 rohan	3 20000

5 IT	Mumbai	411002
Rahul	7 rahul	5 24000

practical no 7:study and implementation of

1.subquerries

2.views

create table sailor(sid int primary key, sname varchar(20), rating int, age int)

create table boats (bid int primary key, bname varchar(20), color varchar(20))

create table reserve(sid int references sailor(sid), bid int references boats(bid), day date)

1. Find all information of sailors who have reserved boat number 101

select * from sailor s,reserve r where s.sid=r.sid and r.bid=101

SID	SNAME	RATING	AGE	SID	BID	DAY
101	Jhon	8	32	101	101	14-JAN-23
104	Wood	9	21	104	101	18-JAN-23

2. Find the name of boat reserved by Bob.

select bname from sailor,boats,reserve where boats.bid=reserve.bid and sname='Bob'

BNAME

Small_boat

Big boat

Small_ship

Small_ship

Small_boat

Small_ship

Small_ship

3. Find the names of sailors who have reserved a red boat, and list in the order of age

select sname, age from sailor, boats, reserve where boats.bid=reserve.bid and sailor.sid=reserve.sid and color='red' order by age

SNAME	AGE
-------	-----

Bob	26
-----	----

4. Find the names of sailors who have reserved at least one boat.

select distinct sname from sailor, reserve where sailor.sid=reserve.sid

SNAME

Wood

Bob

Jhon

5. Find the ids and names of sailors who have reserved two different boats on the same day.

SELECT s.sid, s.sname

FROM sailor s

JOIN reserve r1 ON s.sid = r1.sid

JOIN reserve r2 ON s.sid = r2.sid AND r1.day = r2.day AND r1.bid <> r2.bid

SID SNAME

101 Jhon

102 Bob

101 Jhon

102 Bob

6. Find the ids of sailors who have reserved a red boat or a green boat.

select distinct sailor.sid,sname from sailor,boats,reserve where boats.bid=reserve.bid and
sailor.sid=reserve.sid and color='red' or color='green'

SID SNAME

102 Bob

103 David

101 Jhon

104 Wood

7. Find the name and the age of the youngest sailor.

select sname,age from sailor where age=(select min(age) from sailor)

SNAME	AGE	-----	Wood	21
-------	-----	-------	------	----

8. Count the number of different sailor names.

select distinct(sname) from sailor

SNAME

Wood

Bob

Jhon

David

9. Find the average age of sailors for each rating level.

```
select avg(age),rating from sailor group by rating
```

AVG(AGE)	RATING
----------	--------

29	8
----	---

46	7
----	---

21	9
----	---

10. Find the average age of sailors for each rating level that has at least two sailors.

```
select avg(age),rating from sailor group by rating having count(*)>1;
```

AVG(AGE)	RATING
----------	--------

29	8
----	---

practical 8: study and implementation of plsql

.....no queries and output....

practical 9:study and implementation of sql cursor

1.cursor without parameter:

```
declare
```

```
    cursor empdetail is select * from emp2;
```

```
    empd emp2%rowtype;
```

```
begin
```

```
    open empdetail;
```

```
    DBMS_output.put_line('Emp Data:');
```

```
    loop
```

```
        fetch empdetail into empd;
```



```
exit when empdetail%notfound;

        DBMS_output.put_line('Eno:' || empd.eid);

        DBMS_output.put_line('Name' || empd.ename);

        DBMS_output.put_line('Salary' || empd.salary);

    end loop;

    close empdetail;
```

```
end;
```

```
/
```

```
output:
```

```
Emp Data:
```

```
Eno:1
```

```
NameMohit
```

```
Salary20000
```

```
Eno:2
```

```
Namesumit
```

```
Salary20000
```

```
Eno:3
```

```
Namerohan
```

```
Salary20000
```

```
Eno:4
```

```
Namemohan
```

```
Salary20000
```

```
Eno:5
```

```
Namekhushi
```

```
Salary12000
```

```
Eno:6
```

```
Namesoham
```

```
Salary14000
```

```
Eno:7
```

Namerahul

Salary24000

PL/SQL procedure successfully completed.

2.cursor with parameter:

create or replace procedure e(eid in number)

is

cursor c1(eno number) is select * from emp2 where eid=eno;

empd emp2%rowtype;

begin

DBMS_output.put_line('Emp Data:');

for empd in c1(eid)

loop

DBMS_output.put_line('Eno:' || empd.eid);

DBMS_output.put_line('Name' || empd.ename);

DBMS_output.put_line('Salary' || empd.salary);

end loop;

output:

SQL> exec e(1);

Emp Data:

Eno:1

NameMohit

Salary20000

practical 10:study and implementation of sql function and procedure.

```
create procedure add4 (no1 in number,no2 in number)
```

```
is
```

```
begin
```

```
    DBMS_output.put_line("Addition:" || no1+no2);
```

```
end;
```

```
/
```

```
create or replace procedure welcome(pname in varchar2)
```

```
is
```

```
begin
```

```
    DBMS_output.put_line("Welcome:" || pname);
```

```
end;
```

```
/
```

```
create or replace function box(no1 in number,no2 in number) return n3 number
```

```
is
```

```
begin
```

```
    n3 :=no1+no2;
```

```
    return n3;
```

```
end;
```

```
/
```

Section B

Practical 2

1. Database Commands

View all databases

```
>show dbs
```

```
admin  0.000GB
```

```
cillege 0.000GB
```

```
college 0.000GB
```

```
config 0.000GB
```

```
local  0.000GB
```

Create a new or switch databases

```
>use student
```

View current Database

```
>db
```

```
college
```

Delete Database

```
>db.dropDatabase()
```

```
{ "dropped" : "cillege", "ok" : 1 }
```

Collection Commands

```
>Show Collections
```

```
student
```

Create a collection named 'student'

```
>db.createCollection('student')
```

```
{ "ok" : 1 }
```

Drop a collection named 'stud'

```
db.stud.drop()
```

```
true
```

Show all Rows in a Collection

```
db.student.find()
```

```
{ "_id" : ObjectId("64ae4cb2af1ac97d704ed326"), "name" : "Aarti", "class" : "Fymca", "age" : 22 }
```

Show all Rows in a Collection (Prettified)

```
db.comments.find().pretty()
```

```
{
  "_id" : ObjectId("64ae4cb2af1ac97d704ed326"),
  "name" : "Aarti",
  "class" : "Fymca",
  "age" : 22
}
```

Find the first row matching the object

```
> db.student.find({name:"Jay"}).pretty()
```

```
{
  "_id" : ObjectId("64ae56a9bcdfcc2ecb8b471d"),
  "name" : "Jay",
  "class" : "Symca",
  "age" : 22
}
```

Insert One Row

```
> db.student.insertOne({name:"Jay",class:"Symca",age:22})
```

```
{  
  "acknowledged" : true,  
  "insertedId" : ObjectId("64ae56a9bcafcc2ecb8b471d")  
}
```

Insert many Rows

```
db.student.insertOne([ {name:"Mohit",class:"Symca",age:23}, {name:"Sumit",class:"Symca",age:26} ])
```

```
{  
  "acknowledged" : true,  
  "insertedIds" : [  
    ObjectId("64ae57e4bcafcc2ecb8b471f"),  
    ObjectId("64ae57e4bcafcc2ecb8b4720")  
  ]  
}
```

Limit the number of rows in output

```
db.comments.find().limit(2)
```

Count the number of rows in the output

```
db.comments.find().count()
```

Update a row

```
> db.student.update({"name":"Abc"},{$set:{"class":32}})
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

assignment 3

Mongodb Increment Operator

```
db.student.update({name: 'Aarti'},
```

```
{ $inc: {
```

```
  age: 1
```

```
}})
```

Less than/Greater than/ Less than or Eq/Greater than or Eq

```
db.student.find({age: {$lt: 90}})
```

```
db.student.find({age: {$lte: 90}})
```

```
db.student.find({age: {$gt: 90}})
```

```
db.student.find({age: {$gte: 90}})
```

Practical 3

Queries in mongoDB

```
db.student.insert({ "stud_name": "Maya", "class": "TYBBA_CA", "percentage":80})
```

```
db.student.insert({ "stud_name": "Pooja", "class": "TYBBA_CA", "percentage":50})
```

```
db.student.insert({ "stud_name": "Ram", "class": "SYBBA_CA", "percentage":90})
```

```
db.student.insert({ "stud_name": "Vedant", "class": "SYBBA_CA", "percentage":85})
```

```
db.student.insert({ "stud_name": "Supriya", "class": "TYBBA_CA", "percentage":56})
```

```
db.student.insert({ "stud_name": "Smita", "class": "TYBBA_CA", "percentage":90, "contact":9876543210})
```

Display the students from TYBBA_CA class.

```
>db.student.find({"class":{"$eq:"TYBBA_CA"}})
```

Display Name of students not having TYBBA_CA class.

```
> db.student.find({"class":{"$ne:"TYBBA_CA"}},{"_id":0, "stud_name":1})
```

Display the students having percentage greater than 80

```
>db.student.find({"percentage":{"$gt":80}})
```

Display the students having percentage either 80 or 90.

```
>db.student.find({"percentage":{"$in:[80,90]}})
```

Display students having percentage less than 80 and class TYBBA_CA

```
>db.student.find({$and:[{"percentage":{"$lt":80}},{"class":"TYBBA_CA"}]})
```

Display student details whose name start with “Ra” pattern.

```
>db.student.find({"stud_name": /^Ra/},{ "_id":0})
```

Display student details whose name end with “ya” pattern.

```
>db.student.find({"stud_name": /ya$/},{ "_id":0})
```

Counting the total numbers of documents

```
>db.student.count()
```

Update the percentage of the student whose name is “Maya”

```
>db.student.updateOne({stud_name: "Pooja"}, {$set:{percentage:60}})
```

delete the record of the student whose name is “Pooja”

```
> db.student.deleteOne({name:"Mohit"})
```

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

Update

```
> db.student.updateOne({"name":"Abhishek"},{$se{name:"Abhishek",class:"Fymca",age:23}},{upsert:true})
```