**Assignment No .2**

**Title of Assignment: SQL Queries:**

1. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.

**mysql> create database dbms;**

**mysql> use dbms;**

1. **Create table:**

**mysql> create table stud (RollNo int, Name varchar(20), Class varchar(20));**

**//Show Structure of Table//**

**mysql> desc stud;**

**//Alter table to add Branch & MobileNo//**

**mysql> alter table stud add Branch varchar(20);**

**mysql> alter table stud add MobileNo int;**

**mysql> desc stud;**

**//Insert values into table//**

**mysql> insert into stud values (1,'ABC','TE','Comp',123);**

**mysql> insert into stud values (2,'PQR','TE','Comp',456);**

**mysql> insert into stud values (3,'XYZ','TE','Comp',789);**

**//Display Table//**

**mysql> select \*from stud;**

**//Drop particular Column//**

**mysql> alter table stud drop MobileNo;**

**mysql> select\*from stud;**

**//Add particular Column//**

**mysql> alter table stud add MobileNo int;**

**mysql> select\*from stud;**

**//Update in table//**

**mysql> update stud set Class='SE' where RollNo='3';**

**mysql> select\*from stud;**

**mysql> select RollNo,Name,Class from stud;**

**//Truncate-Delete all rows//**

**mysql> truncate table stud;**

**mysql> select \*from stud;**

**Empty set (0.03 sec)**

**mysql> desc stud;**

**//Drop table//**

**mysql> drop table stud;**

**mysql> desc stud;**

**ERROR 1146 (42S02): Table 'dbms.stud' doesn't exist**

1. **View**

**mysql> use dbms;**

Database changed

**//Create Table//**

**mysql> create table stud(RollNo int, Name varchar(20), Class varchar(20));**

**mysql> insert into stud values (1,'ABC','comp');**

**mysql> insert into stud values (2,'qaz','comp');**

**mysql> insert into stud values (3,'wsx','comp');**

**mysql> select\*from stud;**

**//Create view from table//**

**mysql> create view std as select RollNo, Name from stud;**

**mysql> select\*from std;**

**//Modify View by create or replace statement//**

**mysql> create or replace view std as select Name from stud where Name='wsx';**

**mysql> select\*from std;**

**mysql> drop view std;**

**mysql> select\*from std;**

**ERROR 1146 (42S02): Table 'dbms.std' doesn't exist**

1. **INDEX**

**mysql> use dbms;**

Database changed

**//create Table for Index//**

**mysql> create table stud(RollNo int, Name char(20),Class char(20), Branch char(20));**

**mysql> insert into stud values (1,'qaz','TE','COMP');**

**mysql> insert into stud values (2,'wsx','TE','COMP');**

**mysql> insert into stud values (3,'edc','TE','COMP');**

**mysql> insert into stud values (4,'rfv','TE','COMP');**

**mysql> select\*from stud;**

**//Create INDEX//**

**mysql> create index ind\_1 on stud(Class);**

**mysql> create index ind\_2 on stud(Name);**

**//Display INDEX//**

**mysql> show indexes from stud;**

**Assignment No .2**

**Title of Assignment: SQL Queries:**

1. Design at least 10 SQL queries for suitable database application using SQL DML statements: Insert, Select, Update, Delete with operators, functions, and set operator.

**Example: Crate table for student Marksheet:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SrNo** | **SeatNo** | **Name** | **Class** | **Branch** | **DC** | **EME** | **MCA** | **DNMS** | **CNS** |
| 1 | T190273001 | Shubhangi | TE | E&TC | 41 | 42 | 48 | 40 | 47 |
| 2 | T190273002 | Shital | TE | E&TC | 40 | 40 | 48 | 45 | 48 |
| 3 | T190273003 | Shruti | TE | E&TC | 90 | 80 | 89 | 83 | 74 |
| 4 | T190273004 | Nikita | TE | E&TC | 81 | 81 | 85 | 92 | 65 |
| 5 | T190273005 | Ashwini | TE | E&TC | 47 | 40 | 44 | 50 | 60 |
| 6 | T190273006 | Manjukesh | TE | E&TC | 40 | 16 | 32 | 21 | 36 |

SQL Quarries used:

1. See available database:

* **mysql> show databases;**

1. Choose Database from available:

* **mysql> use trupti;**

1. Create table with name Student\_Marksheet according to given example:

* **mysql> create table Student\_Marksheet( SrNo int, SeatNo varchar(20), Name varchar(20), Class varchar(10), Branch varchar(10), DC int, EME int, MCA int, DBMS int, CNS int);**

1. See Structure of table with **desc** command:

* **mysql> desc Student\_Marksheet;**

1. Enter alteast 6 Records in Student\_Marksheet table:

* **mysql> insert into Student\_marksheet values (1,'T190273001',’Shubhangi','TE','E&TC',41,42,48,40,47);**
* **mysql> insert into Student\_marksheet values (2,'T190273002','Shital','TE','E&TC',40,40,48,45,48);**
* **mysql> insert into Student\_marksheet values (3,'T190273003','Shruti','TE','E&TC',90,80,89,83,74);**
* **mysql> insert into Student\_marksheet values (4,'T190273004','Nikita','TE','E&TC',81,81,85,92,65);**
* **mysql> insert into Student\_marksheet values (5,'T190273005','Ashwini','TE','E&TC',47,40,44,50,40);**
* **mysql> insert into Student\_marksheet values (6,'T190273006','Manjukesh','TE','E&TC',40,16,32,21,36);**

1. Display all records in Student\_Marksheet by using **SELECT\*FROM** command:

* **mysql> select\*from Student\_Marksheet;**

1. Now add another two column as Total\_Marks and Average \_Marks into Student\_Marksheet by using **ALTER TABLE** command:

* **mysql> alter table Student\_Marksheet add Total\_Marks int;**
* **mysql> alter table Student\_Marksheet add Avrage\_Marks int;**

1. Display all records in Student\_Marksheet by using **SELECT\*FROM** command:

* **mysql> select\*from Student\_Marksheet;**

1. Now set value of Total\_Marks Column using **UPDATE AND SET OPERATOR** command

* **mysql> update Student\_Marksheet set Total\_marks=DC+EME+MCA+DBMS+CNS;**
* **mysql> update Student\_Marksheet set Avrage\_marks=Total\_Marks/5;**

1. Display all records in Student\_Marksheet by using **SELECT\*FROM** command:

* **mysql> select\*from Student\_Marksheet;**

1. Display who are get maximum marks in DBMS subject using **SELECT FROM AND MAX** Command:

* **mysql> select Name,DBMS from Student\_Marksheet where DBMS=(select max(DBMS) from Student\_Marksheet);**

1. Display who are get minimum marks in DBMS subject using **SELECT FROM AND MIN** Command:

* **mysql> select Name,DBMS from Student\_Marksheet where DBMS=(select min(DBMS) from Student\_Marksheet);**

1. Display who are get maximum Total\_Marks using **SELECT FROM AND MAX** Command:

* **mysql> select Name,Total\_Marks from Student\_Marksheet where Total\_Marks=(select max(Total\_marks) from Student\_Marksheet);**

1. Display Students names who are gets marks in DC subjects in between 40 and 80 using **SELECT FROM WHERE CAUSE** Command:

* **mysql> select Name,DC from Student\_Marksheet where DC between 40 and 80;**

1. Add a column of Percentage in Student\_Marksheet table to find % of each students by using **ALTER TABLE** Command:

* **mysql> alter table Student\_Marksheet add Percentage int;**

1. Calculate Percentage of each student using **UPDATE and SET OPERATOR** Command:

* **mysql> update Student\_Marksheet set Percentage=((Total\_Marks)/500)\*100;**

1. Display all records in Student\_Marksheet by using **SELECT\*FROM** command:

* **mysql> select\*from Student\_Marksheet;**

**FINAL OUTPUT :**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SrNo** | **SeatNo** | **Name** | **Class** | **Branch** | **DC** | **EME** | **MCA** | **DNMS** | **CNS** | **Percentage** |
| 1 | T190273001 | Shubhangi | TE | E&TC | 41 | 42 | 48 | 40 | 47 | 44% |
| 2 | T190273002 | Shital | TE | E&TC | 40 | 40 | 48 | 45 | 48 | 44% |
| 3 | T190273003 | Shruti | TE | E&TC | 90 | 80 | 89 | 83 | 74 | 83% |
| 4 | T190273004 | Nikita | TE | E&TC | 81 | 81 | 85 | 92 | 65 | 81% |
| 5 | T190273005 | Ashwini | TE | E&TC | 47 | 40 | 44 | 50 | 60 | 44% |
| 6 | T190273006 | Manjukesh | TE | E&TC | 40 | 16 | 32 | 21 | 36 | 29% |

## Assignment No: 3

**Aim:** Design at least 10 SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query and View.

**Example: 1. Crate table for Student:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SeatNo** | **Name** | **Class** | **Branch** |
| T190273001 | Shubhangi | TE | E&TC |
| T190273002 | Shital | TE | E&TC |
| T190273003 | Shruti | TE | E&TC |
| T190273004 | Nikita | TE | E&TC |
| T190273005 | Ashwini | TE | E&TC |
| T190273006 | Manjukesh | TE | E&TC |

**2. Create Table for Marksheet:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SeatNo** | **DC** | **EME** | **MCA** | **DNMS** | **CNS** |
| T190273001 | 41 | 42 | 48 | 40 | 47 |
| T190273002 | 40 | 40 | 48 | 45 | 48 |
| T190273003 | 90 | 80 | 89 | 83 | 74 |
| T190273004 | 81 | 81 | 85 | 92 | 65 |
| T190273005 | 47 | 40 | 44 | 50 | 60 |
| T190273006 | 40 | 16 | 32 | 21 | 36 |

SQL Quarries used:

1. See available database:

* **mysql> show databases;**

1. Choose Database from available:

* **mysql> use trupti;**

1. Create table with name Student according to given example:

* **mysql> create table Student( SeatNo varchar(20), Name varchar(20), Class varchar(10), Branch varchar(10));**

1. See Structure of table with **desc** command:

* **mysql> desc Student;**

1. Enter alteast 6 Records in Student table:

* **mysql> insert into Student values ('T190273001',’Shubhangi','TE','E&TC');**
* **mysql> insert into Student\_marksheet values ('T190273002','Shital','TE','E&TC');**
* **mysql> insert into Student\_marksheet values ('T190273003','Shruti','TE','E&TC');**
* **mysql> insert into Student\_marksheet values ('T190273004','Nikita','TE','E&TC');**
* **mysql> insert into Student\_marksheet values ('T190273005','Ashwini','TE','E&TC');**
* **mysql> insert into Student\_marksheet values ('T190273006','Manjukesh','TE','E&TC');**

1. Create table with name Marksheet according to given example:

* **mysql> create table Marksheet( SeatNo varchar(20), DC int, EME int, MCA int, DBMS int, CNS int);**

1. See Structure of table with **desc** command:

* **mysql> desc Marksheet;**

1. Enter alteast 6 Records in Marksheet table:

* **mysql> insert into Marksheet values ('T190273001', 41,42,48,40,47);**
* **mysql> insert into Marksheet values ('T190273002',40,40,48,45,48);**
* **mysql> insert into Marksheet values ('T190273003' ,90,80,89,83,74);**
* **mysql> insert into Marksheet values ('T190273004', 81,81,85,92,65);**
* **mysql> insert into Marksheet values ('T190273005', 47,40,44,50,40);**
* **mysql> insert into Marksheet values ('T190273006', 40,16,32,21,36);**

1. Display all records in Student and Marksheet by using **SELECT\*FROM** command:

* **mysql> select\*from Student;**
* **mysql> select\*from Marksheet;**

1. As both Table having SeatNo Column same value then we can perform join function on table Student and Marksheet using **INNER JOIN with CLOUMN NAME FROM TABLE1 & TABLE2** command:

* **mysql> select student.Name, marksheet.DC, marksheet.EME, marksheet.DBMS, marksheet.MCA, marksheet.CNS from student inner join marksheet on student.SeatNo=marksheet.SeatNo;**

1. As both Table having SeatNo Column same value then we can perform join function on table Student and Marksheet using **LEFT JOIN with CLOUMN NAME FROM TABLE1 & TABLE2** command:

* **mysql> select student.SeatNo, student.Name,marksheet.DC from student left join marksheet on student.SeatNo=marksheet.SeatNo;**

1. As both Table having SeatNo Column same value then we can perform join function on table Student and Marksheet using **RIGHT JOIN with CLOUMN NAME FROM TABLE1 & TABLE2** command:

* **mysql> select student.SeatNo, student.Name,marksheet.EME,marksheet.CNS from student right join marksheet on student.SeatNo=marksheet.SeatNo;**

1. As both Table having SeatNo Column same value then we can perform join function on table Student and Marksheet using **CROSS JOIN with CLOUMN NAME FROM TABLE1 & TABLE2** command:

* **mysql> select student.SeatNo, student.Name,marksheet.DC,marksheet.EME from student cross join marksheet;**

1. **Pass SUB-QURRIES on table Student or Marksheet**

**SELECT**

**FROM**

**WHERE................. Sub-Quarries**

**HAVING..................Sub-Quarries**

* **Find Maximum marks in EME subject:** 
  + **mysql> select SeatNo, EME from marksheet where EME=(select max(EME) from marksheet);**
* **Find all Subject Average Values:** 
  + **mysql> select avg(DC),avg(EME),avg(DBMS),avg(MCA),avg(CNS) from marksheet;**
* **Find total Count of Student in Table:**
* **mysql> select count(SeatNo) from marksheet;**
* **find SUM of all subject in table:**
* **mysql> select sum(DC), sum(EME), sum(DBMS), sum(MCA), sum(CNS) from marksheet;**
* **Find MIN, MAX & Average marks for DC & CNS subject:**
* **mysql> select min(DC), max(DC), avg(DC), min(CNS), max(CNS), avg(CNS) from marksheet;**
* **Find Students who got marks greater than 60 in DC Subject:**
* **mysql> select SeatNo, DC from marksheet having DC>60;**
* **Find Students who got marks in rage of 50 to 90 in MCA Subject:**
* **mysql> select SeatNo,MCA from marksheet having MCA between 50 and 90;**

1. Create View from marksheet table with **CLOUMN LIST FROM TABLE** command:

* **Create View DC from Marksheet Table:**
* **mysql> create view DC as select SeatNo, DC from marksheet;**
* **update view by create or replace command**
* **mysql> create or replace view DC as select SeatNo, DC from marksheet where SeatNo='T190273004';**
* **display all rows in DC by SELECT \*FROM:**
* **mysql> select \* from DC;**
* **Delete view by DROP:**
* **mysql> drop view DC;**

# Assignment No: 4

**Aim:** PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9.

**create table areas ( r number(2), area number (14,2));**

**declare  
r number(5);  
area number(14,2);  
pi constant number (4,2):=3.14;  
begin  
r:=5;  
while r<=9  
loop  
area:=pi\*power(r,2);  
insert into areas values(r,area );  
r:=r+1;  
end loop;  
end;**

**/**

**select \* from areas;**

**Assignment No .5**

**Title of Assignment: Write a Stored Procedure namely proc\_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks899and 825 category is Higher Second Class.**

set autoprint on;

set serveroutput on;

set verify off;

create table marks(rollno number(10), name varchar2(20), sub1 number, sub2 number, sub3 number, sub4 number, sub5 number);

insert into marks values(1,'Karishma',200,250,260,270,150);

insert into marks values(2,'Mayuri',210,220,230,240,250);

insert into marks values(3,'Asim',250,260,270,280,290);

insert into marks values(4,'Vaishnavi',180,190,200,220,250);

select \* from marks;

create table stud\_marks(rollno number(10), name varchar2(20), total\_marks number(10));

create table Result(rollno number(10), name varchar2(20), class varchar2(30));

create procedure PROC\_GRADE

is

sum number(20);

i number(10);

n number(10);

rollnol number(10);

namel varchar2(30);

totall number(10);

classl varchar2(30);

s1 number;

s2 number;

s3 number;

s4 number;

s5 number;

begin

select count(\*) into n from marks;

i := 0;

loop

i:=i+1;

select rollno into rollnol from marks where rollno=i;

select name into namel from marks where rollno=i;

select sub1 into s1 from marks where rollno=i;

select sub2 into s2 from marks where rollno=i;

select sub3 into s3 from marks where rollno=i;

select sub4 into s4 from marks where rollno=i;

select sub5 into s5 from marks where rollno=i;

totall:=a1+s2+s3+s4+s5;

if totall<=1500 AND totall>=990 then

classl:='DISTINCTION';

end if;

if totall<=989 AND totall>=900 then

classl:='FIRST CLASS';

end if;

if totall<899 AND totall>=825 then

classl:='HIGER SECOND CLASS';

end if;

if totall<825 then

classl:='PASS CLASS';

end if;

insert into stud\_marks values(rollnol,namel,totall);

insert into result values(rollnol,namel,classl);

if i=n then

exit;

end if;

end loop;

end;

/

execute PROC\_GRADE;

select \* from stud\_marks;

select \* from result;

**Assignment No-6**

**Write a PL/SQL block of code using parameterized cursor that will merge the data available in newly created table CompDep with the data available in the Student. If the data in the first table already exists in the second table then that data should be skipped.**

set autoprint on;

set serveroutput on;

set verify off;

create table CompDep(Roll number(10), Name varchar(20));

create table Student(Roll number(10), Name varchar(20));

insert into Student values(1,'a');

insert into Student values(2,'b');

insert into Student values(3,'c');

insert into Student values(4,'d');

insert into CompDep values(2,'b');

insert into CompDep values(5,'e');

insert into CompDep values(6,'f');

declare

cursor cu1 is

select Roll,Name from Student;

cursor cu2 is

select Roll from CompDep;

rno int;

nm varchar(20);

rno2 int;

begin

open cu1;

open cu2;

loop fetch cu1 into rno,nm;

fetch cu2 into rno2;

exit when cu1%found = false;

if rno2 <> rno then insert into CompDep values(rno,nm);

end if;

end loop;

close cu1;

close cu2;

end;

/

Select \*from CompDep;

Assignment No-7

# Title : Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers). Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library\_Audit table. Frame the problem statement for writing Database Triggers of all types, in-line with above statement. The problem statement should clearly state the requirements.

**//Create USER for trigger:**

Login to SQL plus with

User ID: sys as sysdba

Password: svit

**//Write following Cammands to create user**

create user C##Computer identified by svit;

GRANT ALL PRIVILEGES TO C##Computer;

GRANT CONNECT TO C##Computer;

GRANT CREATE SESSION TO C##Computer;

GRANT CREATE TABLE TO C##COMPUTER;

GRANT UNLIMITED TABLESPACE TO C##Computer;

GRANT CREATE VIEW, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER to C##Computer;

GRANT ALTER ANY TABLE to C##Computer;

GRANT ALTER ANY PROCEDURE to C##Computer;

GRANT ALTER ANY TRIGGER to C##Computer;

GRANT DELETE ANY TABLE to C##Computer;

GRANT DROP ANY PROCEDURE to C##Computer;

GRANT DROP ANY TRIGGER to C##Computer;

GRANT DROP ANY VIEW to C##Computer;

**//Exit from SQL open new window of SQL and Login with following details:**

User ID: C##Computer

Password: svit;

create table library(bno number(5),bname varchar2(20),author varchar2(20), allowed\_days number(5));

insert into library values(1,'QWE','poi',10);

insert into library values(2,'ASD','lkj',15);

insert into library values(3,'zxc','mnb',10);

select \* from library;

create table library\_audit(bno number(5),old\_all\_days number(5),new\_all\_days

number(5));

create or replace trigger Sample before update or delete on library for each row

begin

insert into library\_audit values(:new.bno,:old.allowed\_days,:new.allowed\_days);

end;

/

select \* from library\_audit;

update library set allowed\_days = 15 where bno=1;

select \* from library\_audit;

Assignment No-9

**Title: Mongo DB (Installation, Basic CRUD operations, Execution)**

**Procedure to open Mongo dB Command line**

Open C drive on PC then go to Program files 🡺MonodB🡺server🡺version of software🡺bin folder 🡺Double click on mongod.exe 🡺wait for connection 🡺 Double click on mongo.exe🡺once connection successful then write following Quarries

**show dbs;**

**use local;**

**db.createCollection('Student');**

**db.Student.insert({'Rno':'1','Name':'Piyush','Class':'TE COMP'});**

**db.Student.insert({'Rno':'2','Name':'Abhi','Class':'TE COMP'});**

**db.Student.insert({'Rno':'3','Name':'Ashley','Class':'TE COMP'});**

**db.Student.insert({'Rno':'4','Name':'Hitesh','Class':'TE COMP'});**

**db.Student.insert({'Rno':'5','Name':'Pratik','Class':'TE COMP'});**

**db.Student.insert({'Rno':'6','Name':'Priti','Class':'TE COMP'});**

**db.Student.find();**

**db.Student.find().pretty();**

**db.Student.update({'Name':'Hitesh'},{$set:{'Name':'Henry'}});**

**db.Student.find().pretty();**

**db.Student.remove({'Name':'Pratik'},1);**

**db.Student.remove({'Name':'Priti'},1);**

**db.Student.find().pretty();**

**db.Student.drop();**

**db.Student.find().pretty();**

Assignment No- 10

Title: Implement aggregation and indexing with suitable example using MongoDB

**Show dbs;**

**Use local;**

**//CREATE COLLECTION WEBSITE**

**db.createCollection('website');**

//INSERT VALUES IN WEBSITE

**db.website.insert({'roll':'1','name':'harsh','amount':1000,'ur l':'www.yahoo.com'});**

**db.website.insert({'roll':'2','name':'jitesh','amount':2000,'url':'**[**www.yahoo.com**](http://www.yahoo.com/) **'});**

**db.website.insert({'roll':'3','name':'rina','amount':3000,'url':'www.google.com'});**

**db.website.insert({'roll':'4','name':'ash','amount':4000,'url':'www.gmail.com'});**

**db.website.insert({'roll':'5','name':'ash','amount':1000,'url':'**[**www.pvg.com**](http://www.pvg.com/)**'});**

**//SUM AGGREGATE**

**db.website.aggregate({$group:{\_id:"$name","total":{$sum:"$amount"}}});**

**//AVG AGGREGATE**

**db.website.aggregate({$group:{\_id:"$name","total":{$avg:"$amount"}}});**

**//MIN AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total":{$min:"$amount"}}});**

**//MAX AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total":{$max:"$amount"}}});**

**//FIRST AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total":{$first:"$amount"}}});**

**//LAST AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total":{$last:"$amount"}}});**

**//PUSH AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total":{$push:"$amount"}}});**

**//COUNT AGGREGATION**

**db.website.aggregate({$group:{\_id:"$name","total": {$sum:1}}});**

**//ADDTOSET AGGREGATE**

**db.website.aggregate({$group:{\_id:"$name","total"{$addToSet:"$amount"}}});**

**//INDEXING**

**db.createCollection('website1');**

**db.website1.insert({'r':1,'name':'harsh'});**

**db.website1.find().pretty()**

**db.website1.createIndex({'name':1})**

**//CREATE INDEXING**

**db.website1.createIndex({'name':-1})**

**db.website1.getIndexes()**

**db.website1.createIndex({'name':-1})**

**//DROP INDEX**

**db.website.dropIndex({'name':-1})**

//GET INDEXING

**db.website1.getIndexes()**

**db.website1.find().pretty()**

**db.website1.createIndex({'name':1})**

**db.website1.getIndexes()**

**db.website1.dropIndex({'name':1})**

**db.website1.getIndexes()**

**db.website1.createIndex({'name':1,'r':-1})**

**db.website1.getIndexes()**

Assignment No-11

**Show dbs;**

**Use local;**

**db.createCollection('Journal');**

**db.Journal.insert({'book\_id':1,'book\_name':'JavacdOOP','amt':500,'status':'A vailable'});**

**db.Journal.insert({'book\_id':1,'book\_name':'JavaOOP','amt':400,'status':'Not Available'});**

**db.Journal.insert({'book\_id':1,'book\_name':'Java','amt':300,'s tatus':'Not Available'});**

**db.Journal.insert({'book\_id':2,'book\_name':'Java','amt':300,'s tatus':'Available'});**

**db.Journal.insert({'book\_id':2,'book\_name':'OPP','amt':200,'st atus':'Available'});**

**db.Journal.insert({'book\_id':2,'book\_name':'C+','amt':200,'status':'Available'});**

**db.Journal.insert({'book\_id':3,'book\_name':'C+','amt':150,'status':'Available'});**

**db.Journal.insert({'book\_id':3,'book\_name':'C++','amt':200,'status':'Not Available'});**

**db.Journal.insert({'book\_id':4,'book\_name':'OPP C++','amt':300,'status':'Not Available'});**

**db.Journal.insert({'book\_id':5,'book\_name':'OPP ++','amt':400,'status':'Available'});**

**db.Journal.insert({'book\_id':5,'book\_name':'C++','amt':400,'status':'Available'});**

**db.Journal.insert({'book\_id':5,'book\_name':'C++ Java','amt':400,'status':'Not Available'});**

**var mapfunction=function(){ emit(this.book\_id,this.amt)};**

**var reducefunction=function(key,value){return Array.sum(value);};**

**db.Journal.mapReduce(mapfunction,reducefunction,{'out':'new'});**

**db.Journal.mapReduce(mapfunction,reducefunction,{'out':'new'}).find().pretty();**

**db.new.find().pretty();**