CHIT 1 : DDL

```
--create table--
create table employee_master(
emp_id int not null primary key,
first_name varchar(255),middle varchar(255),last_name varchar(255),department
varchar(255),managerid int
);

create table branch_master(
branch_id int not null primary key, branch_name varchar(255)
);

--create a sequence to generate branch id
create sequence seq start with 1 increment by 1;
--create a sequence to generate employee id
create sequence branch_seq start with 1 increment by 1;
--insert
INSERT INTO branch_master
VALUES (seq.nextval,'Mumbai');
INSERT INTO branch_master
VALUES (seq.nextval,'Pune');
--insert
INSERT INTO employee_master
VALUES (branch_seq.nextval,'Tony','Howard','Stark','Computer',3);

--create view
create view emp1 as select emp_id,first_name,middle,last_name,department,managerid
from employee_master;

--create index
CREATE INDEX emp_name ON employee_master(first_name);


select * from employee_master;
select * from branch_master;
```

CHIT 2 : DML

```
 create table student(studid integer primary key,deptname varchar(20),sem
varchar(20),name varchar(20),year integer,credits integer);
 create table teachers(teacherid integer primary key,teachername varchar(20),salary
integer,deptname varchar(20));


 insert into student values(1,'comp','second','sahil',1,90);
 insert into student values(2,'comp','fourth','dev',2,80);
```

```
insert into student values(3,'comp','third','jatin',3,70);

select * from student;

insert into teachers values(1,'MD',10000,'comp');
insert into teachers values(2,'poi',1000,'It');
insert into teachers values(3,'vina',20000,'civil');

select * from teachers;

update student set deptname='IT' where deptname='comp';

select * from student;

insert into teachers values(4,'asmita',30000,'comp');

select * from teachers;

select deptname,max(salary),avg(salary) from teachers group by deptname ;
select deptname from teachers where salary=30000 ;
select deptname from teachers where salary=30000 or salary=1000 ;

delete from teachers where salary<2000;

select * from teachers;

select deptname,sum(salary) from teachers group by deptname;
```

CHIT 3 : P&F KEY

```
create table Dept (
   deptId int not null,
   deptName varchar(10),
   primary key (deptId)
);

create table Emp (
   empId int not null,
   empName varchar(10) not null,
   empSal int,
   empDeptId int,
   PRIMARY KEY (empId),
   FOREIGN KEY (empDeptId) REFERENCES Dept(deptId)
);

insert into Emp values(2, 'abc', 20000, 1);
insert into Emp values(3, 'pqr', 201001, 2);
```

```
insert into Emp values(1, 'xyz', 15000, 1);

select * from Emp;

insert into Dept values(1, 'cs');
insert into Dept values(2, 'mech');

select * from Dept;

alter table Dept
add deptLoc varchar(10) unique;

insert into Dept values(3, 'entc', 'akurdi');
insert into Dept values(4, 'aiml', 'pune');

select * from Dept;
```

CHIT 4 : ASENDING ORDER

```
create table Dept (
   deptId int not null,
   deptName varchar(10),
   primary key (deptId)
);

create table Emp (
   empId int not null,
   empName varchar(10) not null,
   empSal int,
   empDeptId int,
   PRIMARY KEY (empId),
   FOREIGN KEY (empDeptId) REFERENCES Dept(deptId)
);

insert into Emp values(2, 'abc', 20000, 10);
insert into Emp values(3, 'pqr', 20101, 20);
insert into Emp values(1, 'xyz', 15000, 10);

select * from Emp;

insert into Dept values(10, 'cs');
insert into Dept values(20, 'mech');

select * from Dept;

select * from Emp
where empDeptId in (10, 30, 40);
```

```
select * from Emp
where empSal between 10000 and 30000;

select count(*) from Emp;

select empDeptId, avg(empSal) from Emp
group by empDeptId;

select * from Emp
order by empSal;
```

CHIT 5 : JOINS

```
CREATE TABLE Customer(
  customer_id int,
  first_name varchar(255)
  );

 CREATE TABLE orders(
  order_id int,
  amount int,
  customer_id int
  );

INSERT INTO Customer (customer_id,first_name)
VALUES (101,'John');
INSERT INTO Customer (customer_id,first_name)
VALUES (102,'James');
INSERT INTO Customer (customer_id,first_name)
VALUES (103,'Sanjay');
INSERT INTO Customer (customer_id,first_name)
VALUES (104,'Aditya');

INSERT INTO Orders (order_id,amount,customer_id)
VALUES (1001,5000,101);
INSERT INTO Orders (order_id,amount,customer_id)
VALUES (1002,4000,102);
INSERT INTO Orders (order_id,amount,customer_id)
VALUES (1003,6000,103);

--Inner Join

SELECT Customer.customer_id, Customer.first_name, Orders.amount
FROM Customer
INNER JOIN Orders
ON Customer.customer_id = Orders.customer_id;
```

--Left Join

```
SELECT Customer.customer_id, Customer.first_name, Orders.amount
FROM Customer
LEFT JOIN Orders
ON Customer.customer_id = Orders.customer_id;
```

--Right Join

```
SELECT Customer.customer_id, Customer.first_name, Orders.amount
FROM Customer
RIGHT JOIN Orders
ON Customer.customer_id = Orders.customer_id;
```

--Full Outer Join

```
SELECT Customer.customer_id, Customer.first_name, Orders.amount
FROM Customer
FULL OUTER JOIN Orders
ON Customer.customer_id = Orders.customer_id;
```

CHIT 6 : Borrower (OneCompiler)

```
create table borrower (
rollin int,
name varchar(20),
dateofIssue date,
bname varchar(20),
status char(1)
);

create table fine (
rollno int,
fdate date,
amt int
);

insert into borrower values(1, 'a',DATE '2018-07-01', 'java', 'I');
insert into borrower values(2, 'b',DATE'2018-05-01', 'cpp', 'I');
insert into borrower values(3, 'c',DATE'2018-07-12', 'clrs', 'I');
insert into borrower values(4, 'd',DATE '2018-06-02', 'dsa', 'I');
insert into borrower values(5, 'e',DATE '2018-08-04', 'oops', 'I');

select * from borrower;

delimiter $$
```

```
create procedure fine_calculation(IN rno int(3), bname char(20))
begin
declare i_date date;
declare diff int;
declare fine_amt int;
declare exit handler for sqlexception select 'Table not Found';
select dateofIssue into i_date from borrower where rollin = rno and bname = bname;
select datediff(curdate(), i_date) into diff;
if (diff > 15 and diff <= 30) then
set fine_amt = diff * 5;
insert into fine values(rno, curdate(), fine_amt);
elseif (diff > 30) then
set fine_amt = 15*5 + (diff - 30) * 50;
insert into fine values(rno, curdate(), fine_amt);
end if;
update borrower set status = 'R' where rollin = rno and bname = bname;
end $$

call fine_calculation(3, 'clrs');

select * from fine;
select * from borrower;
```

CHIT 7 : PLSQL GRADES

```
create table stud_marks(roll_no number(3), name varchar2(20), marks
number(5));
create table result(roll_no number(3), name varchar2(20), class
varchar2(20));
create procedure Proc_Grade1(roll_no number, name varchar2, marks number)
as
class varchar2(20);
begin
if(marks<=1500 and marks>=990) then
class:='Distinction';
elsif(marks<=989 and marks>=900) then
class:='First Class';
elsif(marks<=899 and marks>=825) then
class:='Higher Second Class';
else
class:='Pass';
end if;
insert into stud_marks values(roll_no, name, marks);
insert into result values(roll_no,name, class);
end;
/
exec Proc_Grade1(101, 'Malan', 1400);
```

```
exec Proc_Grade1(102, 'Sameer', 980);
select * from stud_marks;
select * from result;
```

CHIT 8 : CURSOR (OneCompiler)

```
 create table o_rollcall(roll_no int,name varchar(20),address varchar(20));
create table n_rollcall(roll_no int,name varchar(20),address varchar(20));

insert into o_rollcall values('1','Hitesh','Nandura');
insert into o_rollcall values('2','Piyush','MP');
insert into o_rollcall values('3','Ashley','Nsk');
insert into o_rollcall values('4','Kalpesh','Dhule');
insert into o_rollcall values('5','Abhi','Satara');

delimiter //

create procedure p3(in r1 int)
begin
declare r2 int;
declare exit_loop boolean;
declare c1 cursor for select roll_no from o_rollcall where roll_no>r1;
declare continue handler for not found set exit_loop=true;
open c1;
e_loop: loop
fetch c1 into r2;
if not exists(select * from n_rollcall where roll_no=r2) then
insert into n_rollcall select * from o_rollcall where roll_no=r2;
end if;
if exit_loop then
close c1;
leave e_loop;
end if;
end loop e_loop;
end;
//

delimiter ;

call p3(3);
select * from n_rollcall;
call p3(0);
select * from n_rollcall;
insert into o_rollcall values('6','Patil','Kolhapur');
call p3(4);
select * from n_rollcall;
```

CHIT 9 : TRIGGER

```
create table library(B_id number, Bname varchar2(20), B_author varchar2(20));
insert into library values(100, 'Math3', 'Dev');
insert into library values(103, 'Hindi', 'Manik');
insert into library values(102, 'Malyalam','Selvam');
insert into library values(112, 'Marathi','R Vaidya');
create table library_audit(B_id number,Bname varchar2(20),B_author varchar2(20));
create trigger trig1
before
update or delete
on library
for each row
enable
begin
insert into library_audit values(
 :old.B_id,
 :old.Bname,
 :old.B_author
 );
end;
/
select * from library;
delete from library where B_id=100;
update library set B_id=105 where Bname='Hindi'
select * from library;
select * from library_audit;
```

CHIT 10 : MYSQL CONNECTIVITY

CHIT 11 : CRUD

```
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':'Pratik','Class':'TE COMP'});
db.Student.find();
db.Student.find().pretty();
db.Student.update({'Name':'Hitesh'},{$set:
{'Name':'Henry'}});
db.Student.find().pretty();
db.Student.remove({'ADD':'MP'});
db.Student.find().pretty();
db.Student.find({$and:[{"Name":"Piyush"},{"Rno":"2"}]});
```

```
db.Student.find({$and:[{"Name":"Piyush"},{"Rno":"1"}]}).pretty();
db.Student.find({$and:[{"Name":"Piyush"},{"Rno":"2"}]}).pretty();
db.Student.find({$or:[{"Name":"Piyush"},{"Rno":"2"}]}).pretty();
db.Student.find({$nor:[{"Name":"Piyush"},{"Class":"TE COMP"}]}).pretty();
db.Student.find({$nor:[{"Name":"Piyush"},{"Rno":"2"}]}).pretty();
db.Student.find( {"Rno": { $not:{$lt:"3"}}}).pretty();
db.Student.find( {"Rno": { $eq:"5"}}).pretty();
db.Student.find( {"Rno": { $ne:"5"}}).pretty();
db.Student.find( {"Rno": { $gt:"5"}}).pretty();
db.Student.find( {"Rno": { $lte:"5"}}).pretty();
db.Student.find( {"Rno": { $lt:"5",$gt:"2"}}).pretty();
db.Student.find( {"Rno": { $lte:"5",$gte:"2"}}).pretty();
db.Student.find( {"Rno": { $lte:"5",$gt:"2"}}).pretty();
```

CHIT 12 : AGGREGATE & INDEXING

```
db.createCollection('website');
db.website.insert({'rno':'1','name':'sakshi','amount':'1000','url':'yahoo'});
db.website.insert({'rno':'2','name':'harsh','amount':'2000','url':'google'});
db.website.insert({'rno':'3','name':'manav','amount':'3000','url':'gmail'});
db.website.insert({'rno':'4','name':'ravi','amount':'2000','url':'gmail'});
db.website.insert({'rno':'5','name':'ash','amount':'4000','url':'sinhgad'});
db.website.insert({'rno':'6','name':'ash','amount':'1000','url':'sinhgad'});
db.website.aggregate({$group:{_id:'$name','total':{$sum:'$amount'}}});
db.website.aggregate({$group:{_id:'$name','total':{$sum:1}}});
db.website.aggregate({$group:{_id:'$name','total':{$max:"$amount"}}});
db.website.aggregate({$group:{_id:'$name','total':{$min:"$amount"}}});
db.website.aggregate({$group:{_id:'$name','total':{$first:"$amount"}}});
db.website.aggregate({$group:{_id:'$name','total':{$last:"$amount"}}});
db.createCollection('website2');
db.website2.insert({'rno':'1','name':'harsh'});
db.website2.insert({'rno':'1','name':'harsh'});
db.website2.find().pretty();
db.website2.createIndex({'name':1});
db.website2.createIndex({'name':-1});
db.website2.getIndices();
db.website2.dropIndex({'name':1});
db.website2.getIndices();
db.website2.dropIndex({'name':-1});
db.website2.getIndices();
```

CHIT 13 : MapReduce

```
db.createCollection('class');
 db.class.insert({'id':1,'sec':'A','marks':90});
 db.class.insert({'id':1,'sec':'B','marks':88});
 db.class.insert({'id':2,'sec':'A','marks':82});
 db.class.insert({'id':3,'sec':'A','marks':75});
 db.class.insert({'id':2,'sec':'B','marks':78});
 db.class.find().pretty();
 var map=function(){emit(this.sec,this.marks)};
 var reduce=function(key,value){return Array.sum(value);};
 db.class.mapReduce(map,reduce,{out:'Result'});
 db.Result.find().pretty();
```

CHIT 14 : MONGODB CONNECTIVITY

CHIT 15 : ER Diagram