**Q2b Answer :---**

1. SELECT DISTINCT salesman id FROM orders;
2. SELECT name,city FROM salesman WHERE city='Paris';
3. SELECT \* FROM customer WHERE grade = 200;
4. SELECT ord\_no, ord\_date, purch\_amt FROM orders WHERE salesman\_id = 5001;
5. SELECT \* FROM customer WHERE cust name LIKE '%n';

6)

7)

8) SELECT SUM (purch\_amt) FROM orders;

9) SELECT customer\_id, ord\_date, MAX(purch\_amt) FROM orders GROUP BY customer id, ord date;

10) SELECT \* FROM customer WHERE city = 'New York' OR NOT grade > 100;

**Q.3 Answers :---**

create table hotel(h\_no int primary key, h\_name varchar(100), h\_address varchar(100));

insert into hotel values(01,'Grosvenor','pune');

create table room(r\_no int,primary key(r\_no),h\_no int,r\_type varchar(100), r\_price int, foreign key(h\_no) references hotel(h\_no));

insert into room values(205,1,'rk',100);

create table booking(h\_no int,g\_no int, primary key(g\_no), df date,dt date,r\_no int, foreign key(h\_no) references hotel(h\_no), foreign key(r\_no) references room(r\_no));

insert into booking values(1,2003,'2022-11-2','2022-11-5',205);

create table guest(g\_no int, primary key(g\_no),g\_name varchar(100), g\_address varchar(100));

insert into guest values(2003,'satyajeet','daund');

1. select r\_price,r\_type from room where h\_no=(select h\_no from hotel where h\_name='Grosvenor');
2. select \* from guest where g\_no in (select g\_no from booking where d\_f<=current\_date and d\_t>=current\_date and h\_no=(select h\_no from hotel where h\_name='Grosvenor'));
3. .create view roomcc(h\_no,r\_no,type,price,name)as select r.h\_no,r.r\_no,r.type,r.price,g.name from hotel h, room r,booking b,guest g where h.h\_name='Grosvenor' and (b.d\_f<=current\_date and b.d\_t>=current\_date) and h.h\_no=r.h\_no and r.h\_no=b.h\_no and r.r\_no=b.r\_no and b.g\_no=g.g\_no; create view roomca(h\_no,r\_no,type,price)as select r.h\_no,r.r\_no,r.type,r.price from hotel h,room r where h.h\_name='Grosvenor' and h.h\_no=r.h\_no; select r.r\_no,r.h\_no,r.type,r.price,p.name from roomca r left join roomcc p on r.r\_no=p.r\_no;
4. select sum(price) from booking b,room r,hotel h where (b.d\_f<=current\_date and b.d\_t>=current\_date) and r.h\_no=h.h\_no and r.h\_no=b.h\_no and r.r\_no=b.r\_no and h.h\_name='Grosvenor';
5. select \* from room r where r\_no not in (select r\_no from booking b,hotel h where (d\_f<=current\_date and d\_t>=current\_date) and b.h\_no=h.h\_no and h\_name='Grosvenor');
6. select sum(price) from room r where r\_no not in (select r\_no from booking b,hotel h where (d\_f<=current\_date and d\_t>=current\_date)and b.h\_no=h.h\_no and h\_name='Grosvenor');

**Q.4 Answers :---**

// Create two tables borrower(rollin, name, nameofbook, dateofissue, status) & fine(rollno, returndate, amount), insert values in borrower

// Syntax for procedure :-

> delimiter $

> create procedure p0(in roll int)

-> begin

-> declare fine1 int;

-> declare noofdays int;

-> declare issuedate date;

-> declare exit handler for SQLEXCEPTION select 'create table defintion';

-> select date\_of\_issue into issuedate from borrower where rollin=roll;

-> select datediff(curdate(),issuedate) into noofdays;

-> if noofdays>15 and noofdays<=30 then

-> set fine1=noofdays\*5;

-> insert into fine values(roll,curdate(),fine1);

-> elseif noofdays>30 then

-> set fine1=((noofdays-30)\*50)+15\*5;

-> insert into fine values(roll,curdate(),fine1);

-> else

-> insert into fine values(rol,curdate(),0);

-> end if;

-> update borrower set status='return' where rollin=roll;

-> end $

Query OK, 0 rows affected (0.04 sec)

// To call the procedure :-

-> call p0(3);

-> $

// Check values in borrower & fine table :-

select \* from borrower; $

select \* from fine; $

**Q.5 Answers :---**

create table area(radius float, area float);

>delimiter $

> create procedure area2(in rad float)

-> begin

-> declare radiusBound condition for sqlstate '45000';

-> if (rad<5 || rad>9) then

-> signal sqlstate '45000' set message\_text='radius is not between 5 & 9';

-> else

-> insert into area values(rad,3.14\*rad\*rad);

-> end if;

-> end

-> $

// Call the area procedure :-

call area(6);

-> $

// Check the values inserted int the table :-

select \* from area;

-> $

**Q.6 Answers :---**

// First create table Stud\_Marks(roll,name,totalmarks); insert values

// create table result(roll, name, class);

// Syntax Stored Procedure :-

mysql> delimiter $

mysql> create procedure proc\_Grade(in marks int, out class char(50))

-> begin

-> if marks<=100 and marks>=75 then set class='distinction';

-> end if;

-> if marks<=74 and marks>=60 then set class='first class';

-> end if;

-> end $

Query OK, 0 rows affected (0.03 sec)

//Syntax Stored Function :-

delimiter $

create function result(rollin int) returns int deterministic

beign

declare fmarks int;

declare grade char;

declare name char;

select Stud\_Marks.totalmarks, Stud\_Marks.name into fmarks, name from Stud\_Marks where Stud\_Marks.roll=rollin;

call proc\_Grade(fmarks, @grade);

insert into result values(roll, name, @grade);

return rollin;

end $

//Syntax to call function :-

select result(3);

$

// Syntax to display record :-

select \* from result;

**Q.7 Answers :---**

// Create two tables orollcall & nrollcall insert values in orollcall

// Syntax for cursor :-

create procedure p33(in r1 int)

begin

declare r2 int;

declare exit\_loop boolean;

declare c1 cursor for select rollno from orollcall where rollno>r1;

declare continue handler for not found set exit\_loop=true;

open c1;

exit\_loop : loop

fetch c1 into r2;

if not exists (select \* from nrollcall where rollno=r2)

then

insert into nrollcall select \* from orollcall where rollno=r2;

end if;

if exit\_loop

then

close c1;

leave e\_loop;

end if;

end loop e\_loop;

end //

Query OK, 0 rows affected (0.03 sec)

// Syntax to call :-

call p33(0);

//

//Check records in nrollcall table :-

select \* from nrollcall;

//

**Q.8 Answers :---**

// Create two tables borrower & libaudit. insert values in borrower

// Syntax for Trigger :-

> delimiter $

->create trigger backup after insert on borrower1 for each row

-> begin

-> insert into libaudit values(new.id,new.name,new.issuedate,new.status);

-> end $

// Values will be displayed in both tables :-

> select \* from libaudit;

->$

// Syntax to create next Trigger:-

>delimiter $

> create trigger data after delete on borrower for each row

-> begin

-> insert into libaudit values(old.id,old.name,old.issuedate,old.status);

-> end $

// Perform a delete operation on borrower table :-

>delete from borrower1 where id=1;

-> $

//Check the deleted syntax :-

>select \* from borrower1;

-> $

select \* from libaudit;

-> $

//Sytnax to create Trigger for update :-

> delimiter $

> create trigger data1 after update on borrower1 for each row

-> begin

-> insert into libaudit values(new.id,new.name,new.issuedate,new.status);

-> end $

// Perform update operation on borrower table :-

> update borrower set name='Onkar'

-> where borrower1.id=3;

-> $

// Check updated values in borrower table :-

>select \* from borrower1;

-> $

// Check values from libaudit table :-

select \* from libaudit;

-> $

**Q.9 Answer :---**

//Save python code by “.py”

// open in terminal & perform : python .\ connection.py

// Connection code :-

import mysql.connector as c

con=c.connect(host="localhost",

user="root",

passwd="mysql",

database="aniket")

cursor=con.cursor()

while True:

id=int(input("enter stud id:"))

name=input("enter stud name:")

addr=input("enter stud address:")

query="Insert into stud values({},'{}','{}')".format(id,name,addr)

cursor.execute(query)

con.commit()

print("Data inserted successfully...")

x=int(input("1->Enter More\n2->Exit\nEnter choice:"))

if x==2:

break

id1=input("Enter stud id you want to update:")

name1=input("Enter stud new name")

query="update stud set name='{}' where id={}".format(name1,id1)

cursor.execute(query)

con.commit()

print("Data updated successfully...")

print()

id2=int(input("Enter stud id you want to delete"))

query="delete from stud where name='{}'".format(id2)

cursor.execute(query)

con.commit()

print("Data deleted successfully...")

print()

**Q.10 Answer :---**

db.createCollection('student')

db.student.insert({Name: 'sattu', Class: 'te-b', address : 'daund', age: '19', fees : '1492'})

db.student.find().pretty()

db.student.update({address:'nandurbar'}, {$set : { address: 'nagar'}})

db.student.remove({Name : 'sattu'})

**Q. 11 Answer :---**

1) db.student.find({Class : 'te-a'}).count()

2)

3) db.student.aggregate([{$match : {age : {$gt : '15'}}}])

4) db.student.find().sort({ age : '-1'}).pretty()

5) db.student.createIndex({ Class : '1'})

db.student.getIndexes()

db.student.find({ age : '21'}).pretty()

**Q.12 Answer :--- THIS IS CONCEPT , NOT ACTUAL ANSWER**

// create collection Patient(name,age,ID) insertMany values in it

// create collection result

// Syntax of mapReduce :-

var map=function() {emit(this.Age, this.ID)};

var reduce=funtion (Age, ID) {return Array.sum(ID); };

db.Patient.mapReduce (map, reduce, {out: "resultCollection"});

// Syntax to show mapReduce :-

db.resultCollection.find()

**Q.13 Answer :---**

//Save python code by “.py”

// open in terminal & perform : python .\ connection.py

// Connection code :-

import pymongo

client=pymongo.MongoClient("mongodb://localhost:27017/")

print(client)

mydb=client["Satya"]

mycol=mydb["Students"]

data={'name':'Mobin','age':20}

mycol.insert\_one(data)

data={'name':'Ram','age':40}

mycol.insert\_one(data)

data={'name':'Lakhan','age':50}

mycol.insert\_one(data)

data={'name':'Sumit','age':20}

mycol.insert\_one(data)

data={'name':'Jeevan','age':21}

mycol.insert\_one(data)

data={'name':'Samarth','age':23}

mycol.insert\_one(data)

for data in mycol.find({}):

print(data)

print()

print("find Specific field ,using specific key and value:")

for data in mycol.find({'age':20}):

print(data)

print()

#Update /Edit Specific field ,using specific key ,method and values and update\_one()

mycol.update\_one(

{'name':'Samarth'},

{

"$set":{

'age':21

}

}

)

print("Updated the record ")

print()

#Delete / remove Specific field ,using specific key ,method and values and update\_one()

mycol.delete\_one({'name':'Jeevan'})

print("Deleted the record :Jeevan")

print()

// Code for Mysql connection using Java :---

package dbms;

import java.sql.\*;

import java.util.Scanner;

public class DBMS {

public static void main(String[] args) throws Exception

{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=

DriverManager.getConnection("jdbc:mysql://localhost:3306/bus\_reservation\_system","root","onkar@123");

System.out.println("connected to database");

int count=1;

try{

do{

System.out.println("1. for Insert record \n 2. for Delete record \n 3. for Update Record ");

Scanner sc =new Scanner (System.in);

System.out.println("Enter Choice ");

int choice =sc.nextInt();

switch(choice){

case 1:

PreparedStatement pstmt=con.prepareStatement("insert into Passenger values (?,?,?)");

System.out.println("Enter p\_id:");

pstmt.setInt(1,sc.nextInt());

System.out.println("Enter p\_name:");

pstmt.setString(2,sc.next());

System.out.println("Enter p\_Address:");

pstmt.setString(3,sc.next());

pstmt.executeUpdate();

System.out.println(" Record has been Inserted");

break;

case 2:

PreparedStatement pstmt1=con.prepareStatement("Delete from Passenger where Id=?");

System.out.println("Enter Id to delete record ");

pstmt1.setInt(1,sc.nextInt());

pstmt1.executeUpdate();

System.out.println("Record has been deleted ");

break;

case 3:

PreparedStatement pstmt2=con.prepareStatement("Update Passenger set p\_name=? where id =?");

System.out.println("Enter Name for updation ");

pstmt2.setString(1,sc.next());

System.out.println("Enter Id");

pstmt2.setInt(2,sc.nextInt());

pstmt2.executeUpdate();

System.out.println("Record has been updated");

break;

}

System.out.println("Do you want to continue Press 1 Otherwise press 0");

count=sc.nextInt();

}while (count==1);

}

catch(Exception e){

System.out.println(e);

}}}

// Code for MongoDB connection using Java :---

package pkgfinal;

import com.mongodb.\*;

public class Final {

public static void main(String[] args) {

try {

MongoClient mongo1 = new MongoClient("localhost", 27017);

DB dms = mongo1.getDB("dms");

DBCollection coll =dms.createCollection("dkjfu",null);

}

catch(Exception e) {

e.printStackTrace();

}

}

}