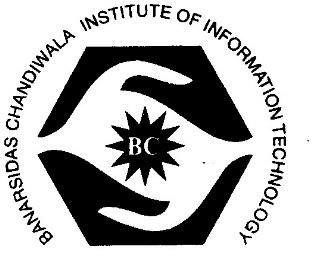
**PRACTICAL FILE**



**DATABASE MANAGEMENT (MCA-165)**

Submitted by: Submitted to:

Name: Abhishek Tyagi Name: Dr. Shushma Bahuguna Roll no : 03311104422

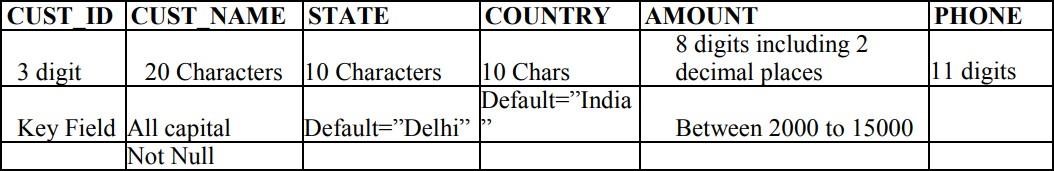
Sem: MCA 1st Designation: Asst. Professor Batch:2022-24

**BANARSIDAS CHANDIWALA INSTITUTE OF INFORMATION TECHNOLOGY**

*Affliated To*

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI**

**Q1. Create table to store customer information and solve the queries.**



Create table customer(

Cust\_id number(3) primary key,

Cust\_name varchar(20) check(cust\_name=upper(cust\_name)) not null,

State varchar(10) default 'Delhi', Country varchar(10) default 'India',

Amount number(8,2) check(amount between 2000 and 15000),

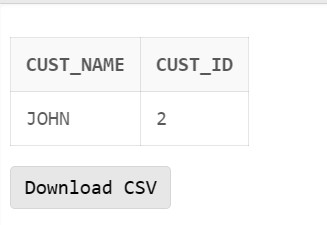
Phone number(11)

);

1. **WAQ to select customer name and id of those customers belonging to Germany**

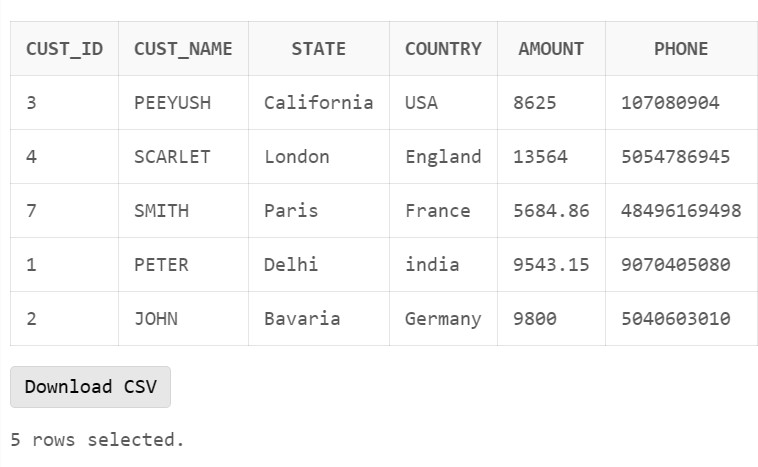
Select cust\_name,cust\_id from customer

Where country=”Germany”;



1. **WAQ to display complete information of customerwhose amount > 3000**

Select \* from customer Where amount>3000;



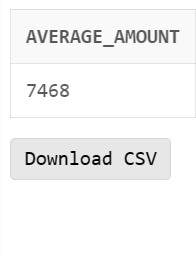
1. **WAQ to select id and country of customer whose name contain a substring as “et”.**

Select cust\_id , country from customer Where cust\_name like ‘%ET%’;



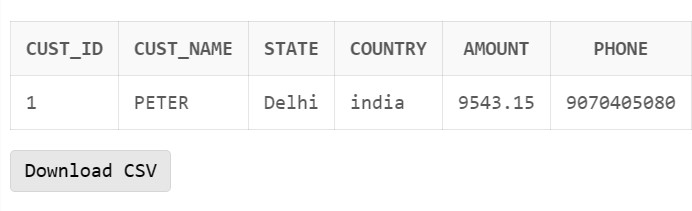
1. **WAQ to display the average of amount of all customers**

Select round(avg(amount)) as average\_amount from customer;



1. **WAQ to display the complete information of “Peter”.**

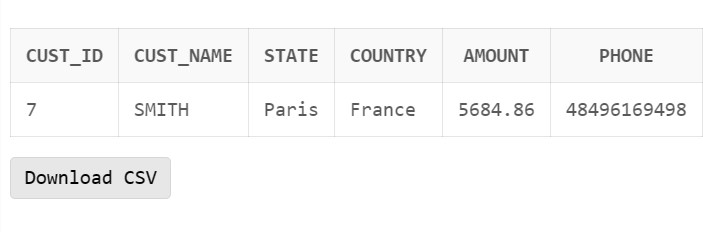
Select \* from customer Where cust\_name=”PETER”;



1. **WAQ to display the information of customer whose amount > 5000 and less than 7000.**

Select \* from customer

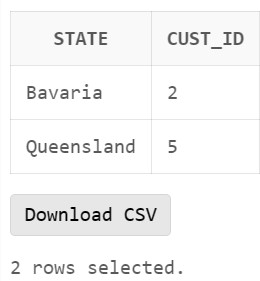
Where amount > 5000 and amount < 7000;



1. **WAQ to select state and id of customer whose name contain “h” as third character.**

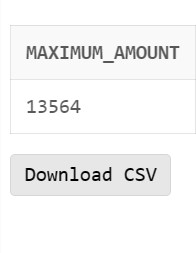
Select state ,cust\_id from customer

Where cust\_name like ‘ H%’;



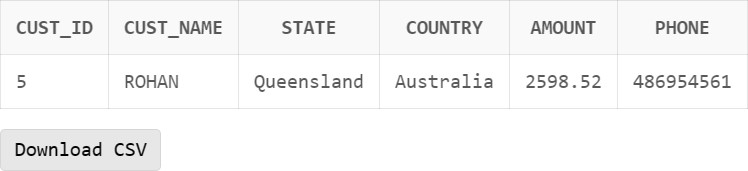
1. **WAQ to display the maximum amount**

Select max(amount) as maximum\_amount from customer;



1. **WAQ to display the complete information of customer(s) belongs to Australia**

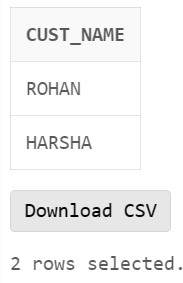
Select \* from customer Where country=”Australia”;



1. **WAQ to display name of customer whose amount >2000 and < 5000**

Select cust\_name from customer

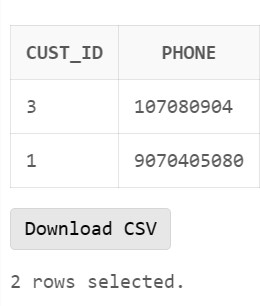
Where amount > 2000 and amount <5000;



1. **WAQ to select id and phone of customer whose name start with “pe”**

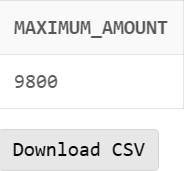
Select cust\_id , phone from customer

Where cust\_name like ‘PE%’;



1. **WAQ to display the maximum amount for country “Germany”.**

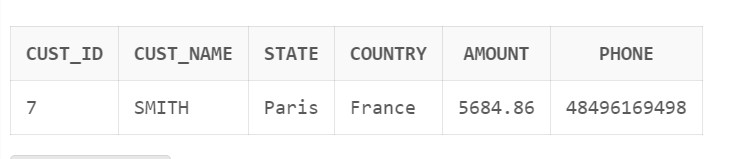
Select max(amount) as maximum\_amount from customer Where country = “germany”;



1. **WAQ to display the complete information of “Smith”.**

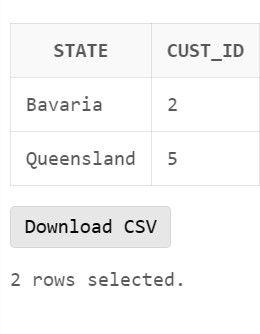
Select \* from customer Where cust\_name =

“SMITH”;



1. **WAQ to select state and id of customer whose name contain “o” as second character.**

Select state , cust\_id from customer Where cust\_name like ‘\_O%’;

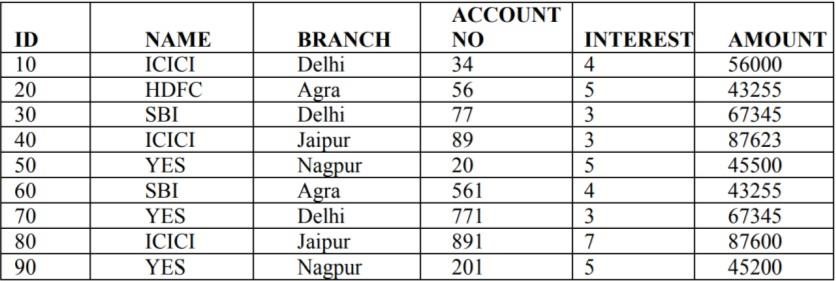


1. **WAQ to select id and country of customer whose name contain a substring as “oh”.**

Select cust\_id,country from customer Where cust\_name like ‘%OH%’;



**Q2. Create a table to store bank information and solve the queries:**



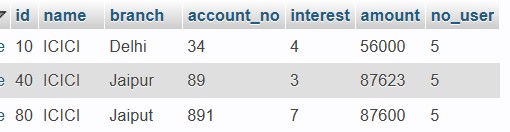
Create table bank( Id int primary key,

Name varchar(10) not null, Branch varchar(10) not null, Account\_no int not null, Interest int not null,

Amount int not null);

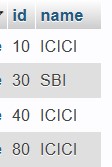
1. **WAQ to display complete information for ICICI bank.**

select \* from bank where name="icici";



1. **WAQ to select id and name of bank whose amount > 50000**.

select id , name from bank where amount>50000



1. **WAQ to select name of bank whose branch name has “pur” as a substring**

select name from bank where branch like ‘%pur%’;



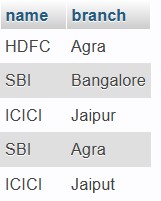
1. **WAQ to select maximum amount among all bank.**

select max(amount) as max\_amount from bank;



1. **WAQ to display name and branch of bank whose no. of account > 50.**

select name , branch from bank where account\_no>50;



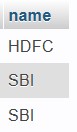
1. **WAQ to display average of amount for Delhi branch**. select avg(amount) as average\_amount from bank

where branch=`Delhi`;



1. **WAQ to select name of bank whose branch name has “g” as a substring.**

select name from bank where branch like ‘%g%’;



1. **WAQ to select minimum amount among all bank.**

select min(amount) as minimum\_amount from bank;



1. **WAQ to display id, name of bank whose interest >5 and less than 8.**

select id , name from bank where interest > 5 and interest <8;



1. **WAQ to display branch name whose amount > 20000 and < 55000.** select branch from bank where amount > 20000 and amount < 55000;



1. **WAQ to count ID of HDFC bank.**

select count(id) from bank where name = ‘HDFC’;



1. **WAQ to display the sum of amount for Delhi branch.**

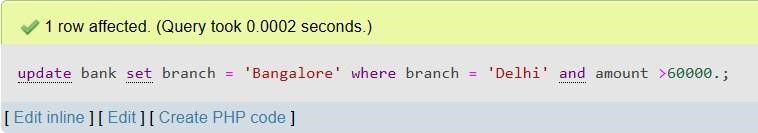
select sum(amount) from bank where branch = ‘Delhi’;



1. **WAQ to update Delhi branch by Bangalore where amount > 60000**

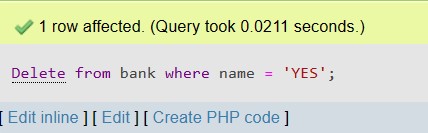
update table bank set branch = ‘Bangalore’

where branch = ‘Delhi’ and amount >60000.



1. WAQ to delete the information of yes bank.

Delete from bank where name = ‘YES’;



1. **WAQ to display name of bank where branch is Delhi and whose amount> 50000.**

Select name from bank where branch = ‘Delhi’ and amount

>50000;



1. **WAQ to select maximum amount of HDFC bank**

Select max(amount) as max\_amount from bank where name =

‘HDFC’;



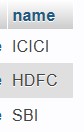
1. **WAQ to display complete information for Delhi branch.**

Select \* from bank where branch = ‘Delhi’;



1. **WAQ to find distinct bank name.**

Select distinct name from bank;



1. **WAQ to delete all data from bank table.**

Truncate bank;

1. **WAQ to select name and ID of bank where ID belongs to hdfc or yes bank.**

Select id from bank

Where name in(‘HDFC’,’YES’);



1. **WAQ to select name and branch of bank where no. of account between 50 and 90.**

Select name,branch from bank

Where account\_no BETWEEN 50 AND 90;



1. **WAQ to select complete details of all bank whose interest between 2 to 6 and belong to IDBI and HDFC bank.**

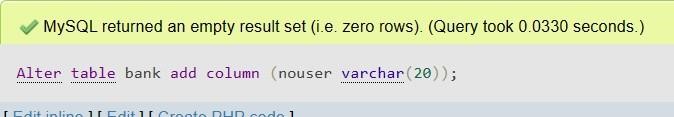
select \* from bank

where interest between 2 and 6 and name in(‘HDFC’,’IDBI’);



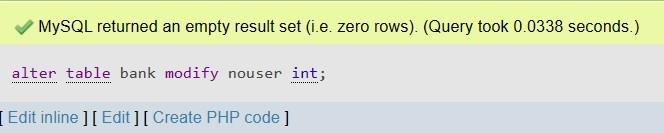
1. **WAQ to add a new column “no\_user” in bank table with char datatype.**

Alter table bank add column (no\_user varchar(20));



1. **WAQ to modify the data type of “no\_user” column from char to int.**

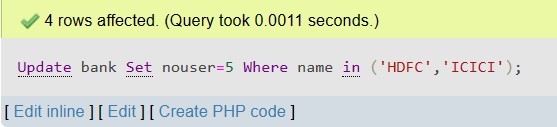
alter table bank modify no\_user int;



1. **WAQ to update the value no\_user = 5 for ICICI and HDFC bank.**

Update bank Set no\_user=5

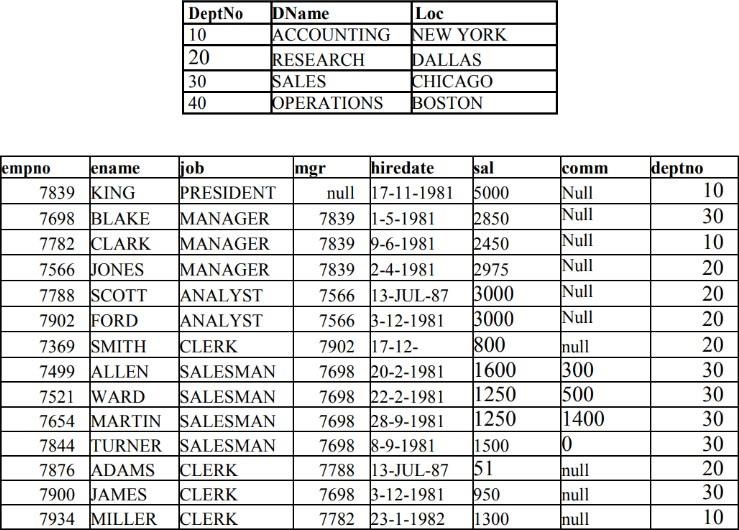
Where name in ('HDFC','ICICI');



1. **WAQ to list the details of bank whose no of user column contains null value.**

Select \* from bank Where no\_user IS null;

Q3) Create the table as shown below and perform the following query:-



Create table department( Deptno int primary key,

Dname varchar(20) not null, Loc varchar(20) not null);

Create table employee( Empno int primary key, Ename varchar(20) not null, Job varchar(20) not null, Mgr int , Hire\_date date not null, Sal int not null,

Comm int,

Deptno int references department(deptno)

);

//DUMP DATA FOR DEPARTMENT TABLE

INSERT into department values(10,'ACCOUNTING','NEW YORK');

INSERT into department values(20,'RESEARCH','DALLAS'); INSERT into department values(30,'SALES','CHICAGO'); INSERT into department values(40,'OPERATIONS','BOSTON');

//DUMP DATA FOR EMPLOYEE TABLE

INSERT INTO employee VALUES (7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, NULL, '10');

* **Display the names of all the employees who are working as clerks and drawing a salary more than 3000.**

Select ename from employee where job=’CLERK’ and sal>3000;

* **Display the names of employees who are working as clerks,salesman or analyst and drawing a salary more than 3000.**

Select ename from employee where job

in(‘CLERK’,’SALESMAN’,’ANALYST’) and sal>3000;

* **Display the list of employees who have joined the company before 30-JUN-90 or after 31-DEC-90.**

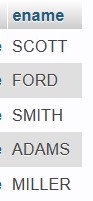
Select ename from employee where hire\_date between ‘1990- 06-30’ and ‘1990-12-31’;

* **Display the names of employees working in depart number 10 or**

**20 or 40 or employees working as CLERKS,SALESMAN or ANALYST.**

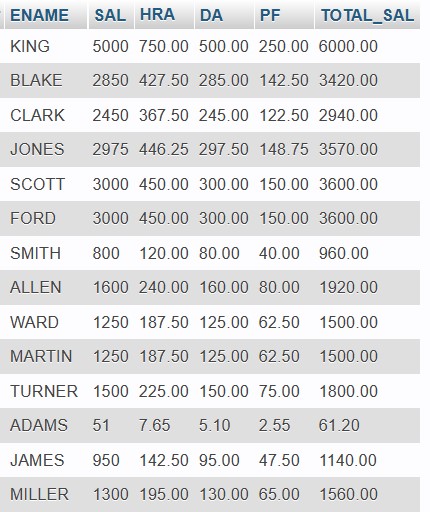
Select ename from employee where deptno in(10,20,40) and

job in(‘CLERK’,’SALESMAN’,’ANALYST’);



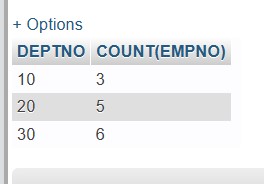
* **Display name,salary,hra,pf,da,total salary for each employee. The output should be in the order of total salary,hra 15% of salary,da 10% of salary,pf 5% salary,total salary will be(salary+hra+da)-pf**

SELECT ENAME, SAL, (0.15\*SAL) AS HRA , (0.10 \* SAL) AS DA,(0.05 \* SAL) AS PF,(SAL+0.15 \* SAL+ 0.10\*SAL - 0.05 \* SAL) AS TOTAL\_SAL FROM EMPLOYEE;



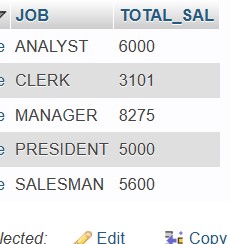
* **Display depart numbers and total number of employees working in each department.**

SELECT DEPTNO , COUNT(EMPNO) FROM EMPLOYEE GROUP BY DEPTNO;



* **Display the various jobs and total salary for each job.**

SELECT JOB , SUM(SAL) AS TOTAL\_SAL FROM EMPLOYEE GROUP BY JOB;

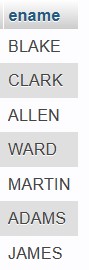


* **Display the total salary drawn by ANALYST working in depart number 40.**

Select sum(sal) as total\_sal from employee where deptno=40 and job=’ANALYST’;

* **Display the names of employees whose names have second alphabet A in their names.**

Select ename from employee where ename like ‘%A%’;



* **Display the maximum salary being paid to CLERK.**

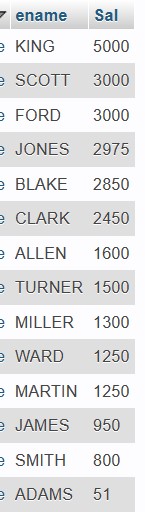
Select max(sal) as maximum\_sal from employee where job =

‘CLERK’;



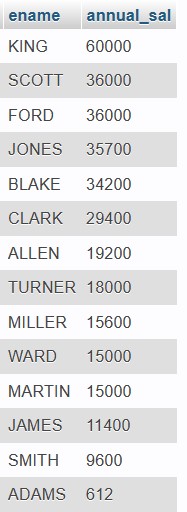
* **Display the names of the employee in descending order of salary.**

Select ename,sal from employee order by sal desc;



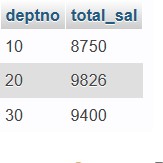
* **Display the name of the employee along with their annual salary(sal\*12).The name of the employee earning highest annual salary should appear first.**

Select ename , (sal\*12) as annual\_sal from employee order by sal\*12 desc;



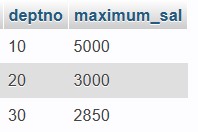
* **Display the depart numbers and total salary for each department.**

Select deptno, sum(sal) as total\_sal from employee group by deptno;



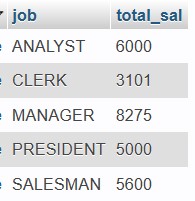
* **Display the depart numbers and max salary for each department.**

Select deptno , max(sal) as maximum\_sal from employee group by deptno;



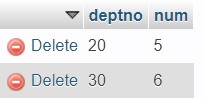
* **Display the various jobs and total salary for each job.**

Select distinct(job) as job , sum(sal) as total\_sal from employee group by job;



* **Display the depart numbers with more than three employees in each dept.**

Select deptno,count(deptno) as num from employee group by deptno having count(Empno)>3;



* **Display the employee number and name for employee working as clerk and earning highest salary among clerks.**

Select empno, ename from employee WHERE SAL=(SELECT MAX(SAL) FROM

EMPLOYEE WHERE JOB='CLERK');



* **Display the names of salesman who earns a salary more than the highest salary of any clerk.**

Select ename from employee where job=’SALESMAN’ AND sal>(select max(sal) from employee where job=’CLERK’);



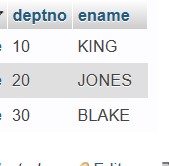
* **Display the names of clerks who earn a salary more than the lowest salary of any salesman.**

select ename from employee where job=’CLERK’ and sal>(select min(sal) from employee where job=’SALESMAN’);



* **Display the names of the employees who earn highest salary in their respective departments.**

Select deptno, ename from employee group by deptno having max(sal);

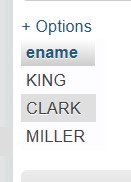


* **Display the employee names who are working in accounting department.**

Select ename from employee,DEPARTMENT where

EMPLOYEE.DEPTNO=DEPARTMENT.DEPTNO AND

DNAME=’ACCOUNTING’;



* **Display the names of employees from department number 10 with salary greater than that of any employee working in other department.**

Select distinct(ename) from employee ,(select sal from employee where deptno !=10) dsal where deptno=10 and employee.sal>dsal.sal;



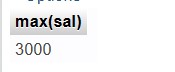
* **Display the names of the employees from department number 10 with salary greater than that of all employee working in other departments.**

Select ename from employee where deptno=10 and sal>(select max(sal) from employee where deptno != 10);



* **Display the maximum salary being paid to depart number 20.**

Select max(sal) from employee where deptno=20;



* **Display the average salary drawn by MANAGERS.**

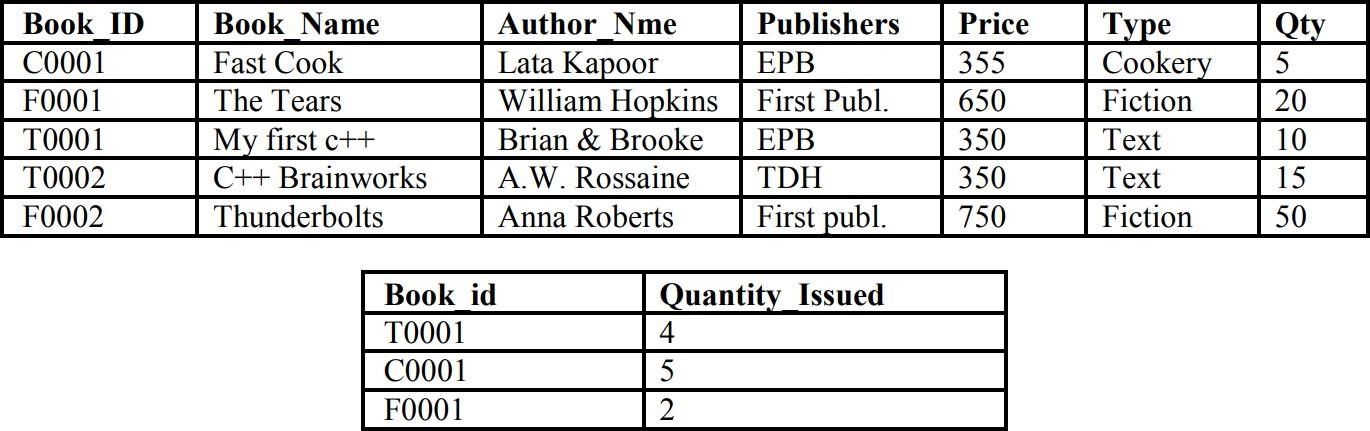
Select avg(asal)from (Select avg(sal) as asal from employee group by mgr) avg\_sal;



* **Select Avg(Sal) from emp where Joj< {01/08/81};** Select avg(sal) from employee where hire\_date<(01/08/81)



**Q4.Given the following tables for a database LIBRARY:**



Create table details(

Book\_id varchar(5) primary key, Book\_name varchar(20) not null, Author\_name varchar(20) not null, Publisher varchar(20) not null, Price int not null,

Type varchar(10) not null, Qty int not null);

Create record(

Book\_id varchar(5) references details(Book\_id), Quantity\_issued int);

**Insert queries for details table.**

insert into details values('C0001','Fast Cook','Lata

Kapoor','EPB',355,'Cookery',5);

insert into details values('F0001','The Tears','William Hopkins','First Publ',650,'Fiction',20);

insert into details values('T0001','My First c++','Brain & Brooke','EPB',350,'Text',10); insert into details values('T0002','C++ Brainworks','A.W. Rossaine','TDH',350,'Text',15);

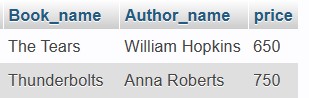
insert into details VALUES('F0002','Thunderbolts','Anna Roberts','First publ',750,'Fiction',50);

**insert queries for record table** insert into record values ('T0001',4); insert into record values('C0001',5); insert into record values('F0001',2);

**Write SQL statements for:-**

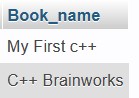
* **To show book name, author name and price of book of First Publ. publishers.**

Select Book\_name, Author\_name, price from details where publisher = ‘First Publ’;



* **To list the names from books of text type.**

Select Book\_name from details where type=’Text’;



* **To display the names and price from books in ascending order of their price.**

Select Book\_name,price from details order by price asc;



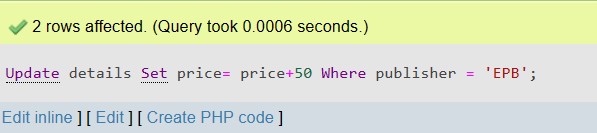
* **To increase the price of all books of EPB publishers by**

**50.**

Update details

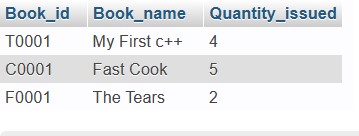
Set price= price+50

Where publisher = ‘EPB’;



* **To display the Book\_Id, Book\_Name and Quantity\_Issued for all books which have been issued.**

Select d.Book\_id,Book\_name,Quantity\_issued from details d , record r where d.Book\_id=r.Book\_id;



* **To insert a new row in the table issued during the following data: “F0003”,1**

Insert into record values(‘F0003’,1);

This query will show error as Book Id is not available in reference table.

* **Give the output for the following SQL queries:** 
  1. **select count(\*) from book.**

Select count(\*) from details;

* 1. **select max(Price) from books where quantity >= 15.** Select max(price)

from details d, record r where r.quantity\_issued >=15;



* + 1. **select book\_Name, Author\_Name from book where**

**Publishers = “EPB”.**

Select Book\_name,Author\_name from details where

Publisher=’EPB’;



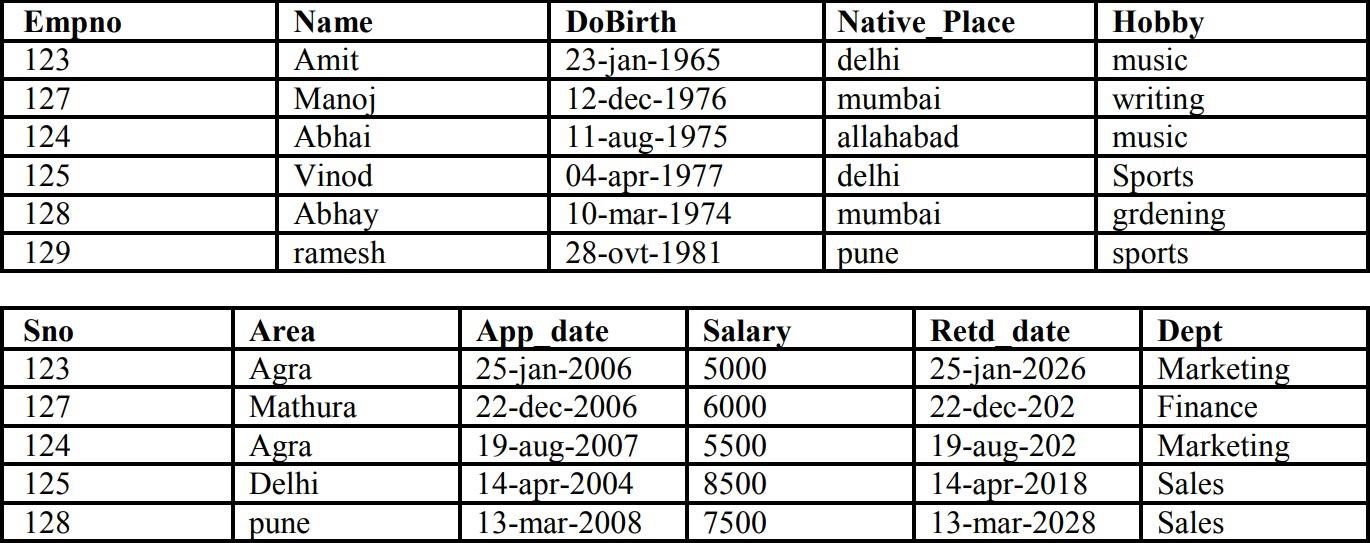
* + 1. **select count (Distinct Publishers) from books where price > = 400;**

select count(distinct(Publisher) from details where price>=40

**Q5. With references to following relations PERSONAL and JOB answer the questions that follow:**

Create following tables such that empno and sno are not null and unique, date of birth is after

’12-Jan-1960’, name is never blank, area and Native place is valid, hobby, dept is not empty, salary is between 4000 and 10000.



create table personal( Empno int primary key, Name varchar(30) not null,

DoBirth date check (DoBirth>'1960-01-12'),

Native\_Place varchar(20) check (Native\_Place in

('delhi','mumbai','pune','allahabad','agra','mathura')),

Hobby varchar(20) not null

);

create table job(

Sno int references personal(empno) ,

Area varchar(20) check (area

in('agra','mathura','delhi','pune','allahabad')),

App\_date date,

Salary int(10) check (salary between 4000 and 10000), Retd\_date date, Dept varchar(20) not null

);

Insert query fro personal table:

INSERT INTO `personal` (`Empno`, `Name`, `DoBirth`,

`Native\_Place`, `Hobby`) VALUES

(123, 'Amit', '1965-01-23', 'delhi', 'music'),

(127, 'Manoj', '1976-12-12', 'mumbai', 'writing')

(124, 'Abhai', '1975-08-11', 'allahabad', 'music'),

(125, 'Vinod', '1977-04-04', 'delhi', 'Sports'),

(128, 'Abhay', '1974-03-10', 'mumbai', 'grdening'),

(129, 'ramesh', '1981-10-28', 'pune', 'sports');

Queries for job table:

INSERT INTO `job` (`Sno`, `Area`, `App\_date`, `Salary`, `Retd\_date`, `Dept`) VALUES

(123, 'Agra', '2006-01-25', 5250, '2026-01-25', 'Marketing'),

(127, 'Mathura', '2006-12-22', 6000, '2022-12-22', 'Finance'),

(124, 'Agra', '2007-10-19', 5775, '2022-08-19', 'Marketing'),

(125, 'Delhi', '2004-04-14', 8500, '2018-04-14', 'Sales'),

(128, 'Pune', '2008-03-13', 7500, '2028-03-13', 'Sales');

* **Show empno, name and salary of those who have sports as hobby.**

select Empno,Name,salary from personal , job where empno=sno and hobby = "Sports";



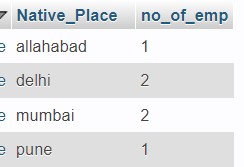
* **Show name of the eldest employee.**

Select name from personal order by DoBirth asc limit 1;



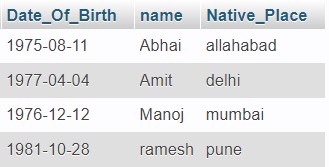
* **Show number of employee area wise.**

select Native\_Place, count(empno) as no\_of\_emp from personal group by Native\_Place;



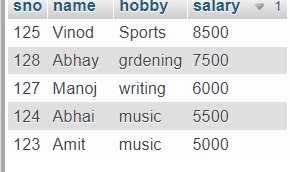
* **Show youngest employees from ache native place.**

select max(DoBirth) "Date\_Of\_Birth" , name,Native\_Place from personal group by Native\_Place;



* **Show sno, name, hobby and salary in descending order of salary.**

select sno, name, hobby ,salary from personal join job on personal.empno=job.sno order by salary desc;



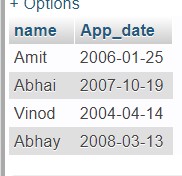
* **Show the hobbies of those whose name pronounces as ‘Abhay’.**

Select hobby from personal where name in(‘Abhay’,’Abhai’);



* **Show the appointment date and native place of those whose name starts with ‘A’ or ends in ‘d’.**

select name , App\_date from personal join job on personal.empno=job.sno where personal.name like 'A%' or personal.name like '%d'



* **Show the salary expense with suitable column heading of those who shall retire after 20-jan-2006.**

SELECT SUM(Salary) AS Salary\_Expense FROM JOB WHERE Retd\_date > '20-jan2006';



* **Show names of those who earn more than all of the employees of sales department.**

SELECT Name FROM JOB JOIN PERSONAL ON JOB.Sno =

PERSONAL.Empno WHERE Salary > (SELECT MAX(Salary) FROM JOB WHERE

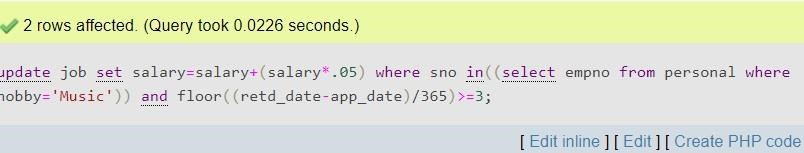
Dept = 'Sales');

* **Increase salary of the employees by 5% of their present salary with hobby as music or they have completed atleast 3 years of services.**

update job

set salary=salary+(salary\*.05)

where sno in((select empno from personal where hobby='Music')) and floor((retd\_dateapp\_date)/365)>=3;



**6.Write Pl/SQL code for**

**a) To reverse a number and print, i.e, if num is 677 then it should print 776:**

DECLARE

x number:=677; y number:=0; z number; BEGIN

z:=x;

LOOP

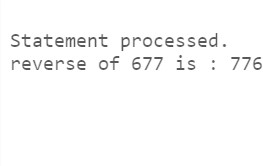
IF x=0 THEN

EXIT; END IF;

y:=(y\*10)+(mod(x,10)); x:=floor(x/10);

END LOOP;

dbms\_output.put\_line('reverse of '||z||' is : '||y); END;



**b) To print a Fibonacci series:**

DECLARE

n NUMBER := 10; a NUMBER := 0; b NUMBER := 1; temp NUMBER;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(a);

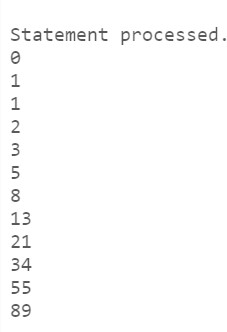
DBMS\_OUTPUT.PUT\_LINE(b);

FOR i IN 1..n LOOP

temp := a + b; a := b;

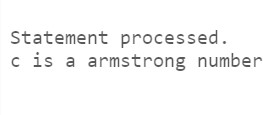
b := temp; DBMS\_OUTPUT.PUT\_LINE(temp); END LOOP;

END;



1. **To check a number is Armstrong or not:**

declare a number:=153; b number:=0; c number; begin c:=a; loop if a=0 then exit; end if; b:=b+power(mod(a,10),3); a:=floor(a/10); end loop; if b=c then dbms\_output.put\_line('c is a armstrong number'); else dbms\_output.put\_line('c is a not armstrong number'); end if; end;



1. **To print the factorial of a given number:**

DECLARE num NUMBER := 5; factorial NUMBER := 1; BEGIN

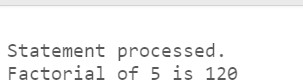
FOR i IN 1..num LOOP

factorial := factorial \* i; END

LOOP;

DBMS\_OUTPUT.PUT\_LINE('Factorial of ' || num || ' is ' || factorial);

END;



1. **To evaluate whether a given number is prime or not:**

DECLARE num NUMBER := 11; is\_prime BOOLEAN := TRUE; BEGIN

FOR i IN 2..(num-1) LOOP IF num

MOD i = 0 THEN is\_prime := FALSE; EXIT; END IF; END

LOOP;

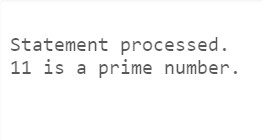
IF is\_prime THEN

DBMS\_OUTPUT.PUT\_LINE(num || ' is a prime number.'); ELSE

DBMS\_OUTPUT.PUT\_LINE(num || ' is not a prime number.');

END IF;

END;



**F) To perform the addition of two numbers**

DECLARE num1 NUMBER := 5; num2 NUMBER := 7; s NUMBER;

BEGIN

s := num1 + num2;

DBMS\_OUTPUT.PUT\_LINE('The sum of ' || num1 || ' and ' || num2 || ' is ' || s );

END;



1. **To get a number from keyboard and if it zero print “natural number”, else print “not a natural number”:**

DECLARE num NUMBER;

BEGIN

DBMS\_OUTPUT.PUT('Enter a number: '); num := &num;

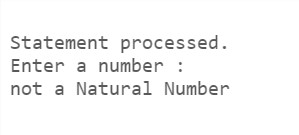
IF num = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Natural number');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Not a natural number'); END IF;

END;



1. **To find the area and perimeter of given circle:**

DECLARE

radius NUMBER := 2; pi NUMBER := 3.14; area NUMBER; perimeter NUMBER;

BEGIN

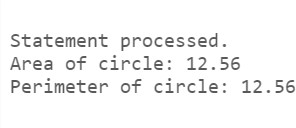
area := pi \* radius \* radius; perimeter :=

2 \* pi \* radius;

DBMS\_OUTPUT.PUT\_LINE('Area of circle: ' || area);

DBMS\_OUTPUT.PUT\_LINE('Perimeter of circle: ' || perimeter);

END;



1. **To calculate the net salary if dfa is 30% of basic, hra is 10% of basic and pf is 7%. If basic salary is less than 8000, pf is 10% if basic sal between 8000 to 160000.**

DECLARE

basic\_salary NUMBER := 10000; dfa

NUMBER;

hra NUMBER; pf

NUMBER;

net\_salary NUMBER; BEGIN

dfa := basic\_salary \* 0.3; hra := basic\_salary \* 0.1;

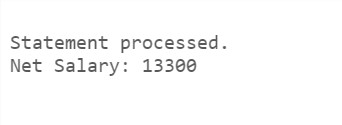
IF basic\_salary < 8000 THEN pf := basic\_salary \* 0.1;

ELSE

pf := basic\_salary \* 0.07; END IF;

net\_salary := basic\_salary + dfa + hra - pf; DBMS\_OUTPUT.PUT\_LINE('Net Salary: ' || net\_salary);

END;



1. **To select record of emp table with cursor:**

DECLARE

CURSOR emp\_cursor IS SELECT

* + FROM emp; emp\_rec emp%ROWTYPE; BEGIN

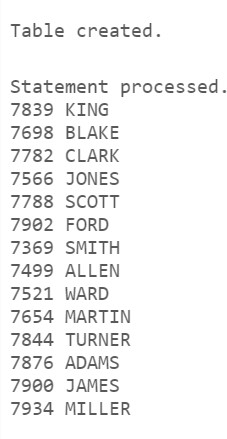
OPEN emp\_cursor; LOOP

FETCH emp\_cursor INTO emp\_rec; EXIT WHEN emp\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(emp\_rec.empno || ' ' || emp\_rec.ename);

END LOOP;

CLOSE emp\_cursor; END;



1. **To raise an error if no data found:**

DECLARE

CURSOR emp\_cursor IS SELECT

* + FROM emp; emp\_rec emp%ROWTYPE; no\_data\_found EXCEPTION;

PRAGMA EXCEPTION\_INIT(no\_data\_found, -20009); BEGIN

OPEN emp\_cursor;

FETCH emp\_cursor INTO emp\_rec; IF emp\_cursor%NOTFOUND THEN

RAISE no\_data\_found; ELSE

DBMS\_OUTPUT.PUT\_LINE(emp\_rec.empno || ' ' || emp\_rec.ename);

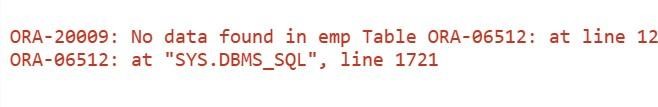
END IF;

CLOSE emp\_cursor;

EXCEPTION

WHEN no\_data\_found THEN DBMS\_OUTPUT.PUT\_LINE('No data found in the table.');

END;



**Q7 Write and explain the following Pl/SQL triggers on emp table**

**i) Before UPDATE Trigger**

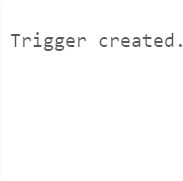
create or replace trigger

before\_update\_trigger before update on emp for each row

begin

dbms\_output.put\_line('Before executing update statement...');

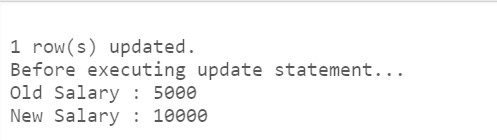
dbms\_output.put\_line('Old Salary : '||:old.sal); dbms\_output.put\_line('New Salary : '||:new.sal); end;



update emp set sal =

10000

where ename='KING';



1. **Before DELETE Trigger**

create or replace trigger before\_delete\_trigger before delete on emp for each row

begin

dbms\_output.put\_line('Old Salary : '||:old.sal); dbms\_output.put\_line('New Salary : '||:new.sal); dbms\_output.put\_line('Before executing Delete statement...'); end; delete from emp where ename=’KING’; **OUTPUT**



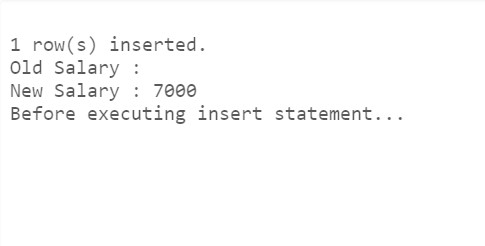
1. **Before INSERT Trigger**

create or replace trigger before\_insert\_trigger before insert on emp for each row

begin dbms\_output.put\_line('Old Salary : '||:old.sal); dbms\_output.put\_line('New Salary : '||:new.sal); dbms\_output.put\_line('Before executing insert statement...'); end;

Insert into emp(ename,sal) values('TARUN',7000);

**OUTPUT:**



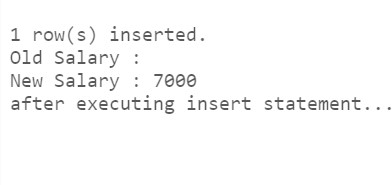
# AFTER INSERT TRIGGER

create or replace trigger after\_insert\_trigger after insert on emp for each row

begin

dbms\_output.put\_line('Old Salary : '||:old.sal); dbms\_output.put\_line('New Salary

: '||:new.sal); dbms\_output.put\_line('after executing insert statement...'); end;



# After UPDATE Trigger

create or replace trigger after\_update\_trigger after update on emp for each row

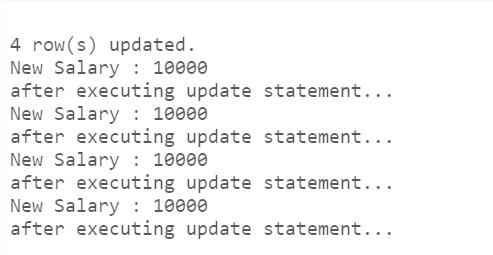
begin

dbms\_output.put\_line('New Salary : '||:new.sal);

dbms\_output.put\_line('after executing update statement...'); end;

update emp set sal=10000

where ename=’vivek’;

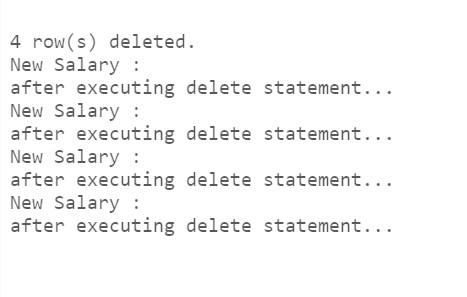


# After DELETE Trigger

create or replace trigger after\_delete\_trigger after delete on emp for each row

begin

dbms\_output.put\_line('New Salary : '||:new.sal); dbms\_output.put\_line('after executing delete statement...'); end;



delete from emp where ename='bciit'; OU