7.AGGREGATE FUNCTION IN SQL

1.

```
SQL> select avg(physics) from student;

AVG(PHYSICS)
-----------------------
           16.2
```

2.

```
SQL> select max(maths) as highest_marks_maths from student;

HIGHEST_MARKS_MATHS
-----------------------------------------
                      48
```

3.

```
SQL> select min(chemistry) as lowest_marks_chemistry from student;

LOWEST_MARKS_CHEMISTRY
------------------------------------------------
                      7
```

4.

```
SQL> select count(physics) from student where physics>=12;

COUNT(PHYSICS)
----------------------------
           8
```

5.

```
SQL> select * from student where physics>=12 and chemistry>=12 and maths>=25;

ROLL_NO        NAME   PHYSICS        CHEMISTRY      MATHS
—-----------   —------ —-----------   —---------------  —---------
1              adam   20             20             33
8              mary   24             14             31
```

6.

```
SQL> alter table student add totalmark int add result varchar(10);
SQL> update student set totalmark=physics+chemistry+maths;
SQL> update student set result='p' where physics>=12 and chemistry>=12 and maths>=25;
SQL>  update student set result='f'  where physics<=12 and chemistry<=12 and maths<=25;
SQL> update student set result='f'  where physics<=12 or chemistry<=12 or maths<=25;
SQL> select * from student order by totalmark desc;

ROLL_NO      NAME         PHYSICS      CHEMISTRY    MATHS        TOTALMARK    RESULT
—-----------  —------      —-----------  —---------------  —-----------  —-----------------  —----------
      1      adam         20           20           33           73           p
      8      mary         24           14           31           69           p
      2      bob          18           9            41           68           f
      10      zack        8            20           36           64           f
      3      bright       22           7            31           60           f
      6      fletcher     2            10           48           60           f
      5      elvin        14           22           23           59           f
      9       tom         19           15           24           58           f
      7      georgina     22           12           22           56           f
      4      duke         13           21           20           54           f
```

7.

```
SQL> select count(maths)*10 as pass_persentage_maths from student where maths>=25

PASS_PERCENTAGE_MATHS
-------------------------------------------
      60
```

8.

```
SQL> select count(result)*10 as pass_percentage from student where result='p';

PASS_PERCENTAGE
-------------------------------
      20
```

9.

```
SQL> select sum(totalmark)/10 as class_avg from student;

 CLASS_AVG
--------------------
      62.1
```

10..

```
SQL> select count(result) from student where result='p';

COUNT(RESULT)
------------------------
      2
```

8.IMPLEMENTATION OF ORDER BY, GROUP BY&HAVING CLAUSE

1.

```
SQL>  select * from customers order by state;

CUSTID   CUSTNAME  ADDRESS        STATE
—--------    —------------- —-------------       —-------
112          patrick        street 1            chennai
111          elvin          jai street          delhi
113          soman          kottarakara         kerala
```

2.

```
SQL> select *from item where price>(select avg(price) from item)order by category;

ITEMID ITEMNAME                  CATEGORY       PRICE  INSTOCK
—-------- —--------------            —--------------     —-------- —------------
5          sony z5 premium           electronics        5000     1
```

9.IMPLEMENTATION OF SET OPERATORS NESTED QUERIES ,AND JOIN QUERIES

1.

```
SELECT * FROM customers WHERE custidIN(select custid from orders)
```

```
CUSTID CUSTNAME     ADDRESS        STATE
—-------  —---------------   —------------      —------
111       elvin             202 jai street    delhi
113       soman             puthumanao     kerala
115       mickey            juhu                mumbai
114       jaise               kottarakara      kerala
```

2.

```
SELECT* FROM customers WHEREcustid IN (select custid from delivery)
```

```
CUSTID CUSTNAME     ADDRESS        STATE
—-------  —---------------    —-------------     —------
115       mickey            juhu                mumbai
111       elvin             202 jai street    delhi
113       soman             puthumanao     kerala
```

3.

```
SELECT orderdate FROM orders WHEREcustidin(select custid from customers
where custname LIKE 'j%')
```

```
ORDERDATE
—----------------
22-DEC-14
```

4.

```
SELECT itemname,price FROM item where itemid IN(select itemid from orders
where custid in(select custid from customers where custname='mickey'))
```

```
ITEMNAME               PRICE
—---------------------       —------
sony z5 premium         5000
```

5.

```
SELECT * FROM customers WHEREcustidIN(select custid from orders where
orderdate>'31- jan-2013' AND orderid NOT IN(select orderid from delivery))
```

```
CUSTID   CUSTNAME  ADDRESS        STATE
—--------    —-------------- —------------      —--------
114          jaise          kottarakara      kerala
```

6.

```
SELECT itemid FROM orders UNION SELECT itemid FROM orders where custidIN(select
custid from delivery)
```

```
ITEMID
—-------
1
3
4
5
```

7.

```
SELECT custname FROM customers WHERE custidIN(select custid from orders)

INTERSECT SELECT custname FROM customers WHERE custid IN(select custid
from delivery)
```

```
CUSTNAME
—--------------
Elvin
Mickey
Soman
```

8.

```
SELECT custname FROM customers WHERE custidIN(select custid from
orders) MINUS select custname from customers where custid IN(select custid
from delivery)
```

```
CUSTNAME
—--------------
Jaise
```

9.

```
SELECT * FROM customers WHERE custidIN(select custid from orders where
quantity = (select max(quantity) from orders))
```

```
CUSTID   CUSTNAME   ADDRESS      STATE
—---------  —-------------- —------------  —-------
114       jaise        kottarakara  kerala
```

10.

```
SELECT * FROM customers WHERE custidIN(select custid from orders where itemid
IN(select itemid from item where price>5000))
```

```
CUSTID   CUSTNAME   ADDRESS      STATE
—---------  —-------------- —------------  —-------
115      mickey      juhu         mumbai
```

11.

```
SELECT custname,address FROM customers WHERE custid NOT IN(select custid
from orders where itemid IN(select itemid from item where itemname='galaxy s4'))
```

```
CUSTNAME     ADDRESS
—---------------  —-------------
elvin         202 jai street
patrick       street 1 harinagar
jaise         kottarakara
mickey        juhu
```

12.

```
SELECT * FROM customers LEFT JOIN orders ON customers.custid=orders.custid;
```

```
CUSTID   CUSTNAME   ADDRESS        STATE   ORDERID   CUSTID ITEMID QUANTITY ORDER DATE
—---------  —---------------   —-------------   —------- -------------   —---------  —-------  —-----------  —------------------
111     elvin        202 jai street   delhi     911        111     1      2      11-OCT-14
113     soman        puthumana        kerala    912        113     3      1      29-JAN-12
115     mickey       juhu             mumbai    913        115     5      1      16-MAY-13
114     jaise        kottarakara      kerala    914        114     4      3      22-DEC-14
112     patrick      harinagar        chennai    -          -      -      -          -
```

13.

```
SELECT * FROM customers ORDER BY state
```

```
CUSTID   CUSTNAME   ADDRESS      STATE
—---------   —--------------  —------------   —-------
112      patrick     harinagar    chennai
111      elvin       jai street   delhi
113      soman       puthumana    kerala
114      jaise       kottarakara  kerala
115      mickey      juhu         mumbai
```

14.

```
SELECT * FROM item WHERE price>(select avg(price) from item) ORDER BY category
```

```
ITEMID  ITEMNAME             CATEGORY     PRICE  INSTOCK
—-------  —---------------------   —--------------   —-------  —-----------
5        sony z5 premium      electronics  5005   1
```

EXPERIMENT NO: 10 TCL COMMANDS

1.

```
create table student(name varchar(20),roll number,address varchar(30));
```

2.

```
insert into student values('arun',13,'aroma');
insert into student values('kiran',28,'aroma');
select * from student;

NAME ROLL ADDRESS
—----- —------ —-----------
arun   13      aroma
kiran  28      aroma

SQL> savepoint in_student;
Savepoint created.
```

3.

```
SQL> insert into student values('ned',35,'winterfell');

SQL> select * from student;

NAME ROLL ADDRESS
—----- —------ —-----------
Arun   13      aroma
Kiran  28      aroma
Ned    35      winterfell

SQL> rollback to in_student;
Rollback complete.

SQL> select * from student;

NAME ROLL ADDRESS
—----- —------ —-----------
arun   13      aroma
kiran  28      aroma
```

EXPERIMENT NO: 11

IMPLEMENTATION OF VARIOUS CONTROL STRUCTURES USING PL/SQL

1.

```
DECLARE
        Aint;
        Bint;
BEGIN
        a:=:a;
        b:=:b;
        dbms_output.put_line('sum is'||(a+b));
        dbms_output.put_line('difference is'||(a-b));
        dbms_output.put_line('product is'||(a*b));
        dbms_output.put_line('qoutient is'||(a/b));
        END
```

```
OUTPUT:
        A:30
        B:5

sum is35
difference is25
product is150
quotient is6

        Statement processed.
```

2.

```
DECLARE
        Iint;
        j int;
        Nint;
        Numint;
        cntint
        flag int;

BEGIN
        flag:=0;
        int:=0;
        num:=2;
        While int<20
        Loop
```

```
flag:=0;
i:=num;
for j in
2..i/2 loop
If mod(i,j)=0 then
flag:=1;
end if;
end loop;
if flag=0 then
        Int:int+1;
dbms_output.put_line(i);
end if;
num:=num+1;
end loop;
END;
```

```
OUTPUT:
        2
        3
        5
        7
        11
        13
        17
        19
        23
        29
        31
        37
        41
        43
        47
        53
        59
        61
        67
        71

Statement processed.
```

3.

```
DECLARE
    Aint;
    Bint;
    Cint;
    Cntint;
    i int;
BEGIN
    i:=:I;
    int:=1;
    a:=0;
    b:=1;
    dbms_output,put_line(a);
    dbms_output,put_line(b);
    for int in 1..i-2
    Loop
    c:=a+b;
    dbms_output,put_line(c);a:=b;
    b:=c;
    end loop;
    END
```

4.

```
    CREATE TABLEstudent_grade(roll int,name varchar2(10),mark 1 int,mark2
int,mark3 int,gradevarchar(5));

    DECLARE
        Totalint;
        avrg float;
        roll int;
        mark1 int;
        mark2 int;
        mark3 int;
        grade varchar2(5);
        name varchar2(10);
    BEGIN
        roll:=:roll;
        name:=:name:
        mark1:=:mark1;mark2:=:mark2;
        mark3:=:mark3;
        total:=mark1+mark2+mark3;
        avrg:=total/3;
        ifavrg>40 then
        grade:='PASS';else
        grade:='FAIL';end if;
        insert into student_grade values(roll,name,mark1,mark2,mark3,grade);
        END;

    select* from student_grade

    ROLL        NAME        MARK1       MARK2       MARK3       GRADE
    —----       —-----      —--------   —--------   —--------   —---------
    1           anu         50          45          48          PASS
    2           manu        50          50          50          PASS
    3           manu        35          40          40          FAIL
```

5.

```
create table circle_area(radius int,area float(5));

    DECLARE
        Radiusint;
        Iint;
        area float;
    BEGIN
        radius:=0;
        for i in 1.5 loop
        radius:=radius+5;
        area:=3.14*power(radius,2);
        INSERT INTO circle_areavalues(radius,area);
        end loop;
        END;

SELECT * from circle_area;

RADIUS      AREA
—---------  —------
5           79
10          310
15          710
20          1300
25          2000
```

6.

```
CREATE TABLEstud(names varchar2(10),marks int);

        DECLARE
                typestdnames is varray(10) of varchar2(10);
                typestdmarks is varray(10) of int;
                Namesstdnames;
                Marksstdmarks;
                Iint;
        BEGIN
                names:=stdnames('ARUN','AMAL','PETER','JOSE','ANNIE','MARY','JOSEPH','MARK'
                ,'MIDHUN','KEVIN');
                marks:=stdmarks(25,76,43,45,67,57,97,56,89,08);
                for i in 1..10 loop
                INSERT INTO stud values(names(i),marks(i));
                end loop;
                END;

SELECT * from stud
NAMES           MARKS
—--------       —--------
ARUN            25
AMAL            76
PETER           43
JOSE            45
ANNIE           67
MARY            57
JOSEPH          97
MARK            56
MIDHUN          89
KEVIN           8
```

7.

```
CREATE TABLE class_cse(roll int,name varchar2(10),phone varchar(20));
create sequence seql
start with 1
increment by 1
maxvalue 10
        DECLARE
                typestdnameis varray(10) of varchar2(10);
                typestdphoneis varray(10) of varchar2(20);
                Namestdname;
                phonestphone
        BEGIN
                name:=stdname('ARUN','AMAL','PETER','JOSE','ANNIE');
                phone:=stdphone('0482-239091','0484-234562','0485-11234','0489-43617','0481-23145'
                for i in 1..5 loop
                INSERT INTOclass_csevalues(seql.nextval,name(i),phone(i));
                end loop;
                END;
```

```
        SELECT * from class_cse;

        ROLL            NAME            PHONE
        —------         —-------        —---------
        1               ARUN            0482-239091
        2               AMAL            0484-234562
        3               PETER           0485-11234
        4               JOSE            0489-43617
        5               ANNIE           0481-23145
```

EXPERIMENT NO: 12
CREATION OF PROCEDURES, TRIGGERS AND FUNCTIONS

1.

```
SQL> create or replace procedure mult_table(a number,b number) is
        mul number;
        Begin
                for i in 1. .b loop
                mul:=a*i;
                dbms_output.put_line(to_char(a)||'*'||to_char(i)||'='||to_char(mul));
                end loop;
        end mult_table;

Procedure created.

SQL> set serveroutput on;
SQL> declare
        2 c number;
        3 d number;
        4 begin
        5 c:=&c;
        6 d:=&d;
        7 mult_table(c,d);
        8 end;
        9 /
        OUTPUT
        Enter value for c: 5
        old 5: c:=&c;
        new 5: c:=5;
        Enter value for d: 4
        old 6: d:=&d;
        new 6: d:=4;
        5*1=5
        5*2=10
        5*3=15
        5*4=20
        PL/SQL procedure successfully completed
        Commit complete.
```

2.

```
SQL> select * from employee;

EMPID       EMPNAME             SALARY      PID
----------  --------------------  ----------  ----------
1001        Ravi                25500       2
1002        Rajesh              30000       1
1003        Ramesh              22500       3
1004        Ratheesh            16000       5

SQL> create or replace procedure raisesal(empl number,amt number) is
        Begin
                update employee set salary=salary+amt where empid=empl;
                end;

Procedure created
SQL> set serveroutput on;
SQL> declare
        empn number;
        amt number;
        Begin
        dbms_output.put_line('Enter employee number');
        empn:=&empn;
        dbms_output.put_line('Enter salary increment');
        amt:=&amt;
        raisesal(empn,amt);
        dbms_output.put_line('Salary of '||to_char(empn)||'increased by '||to_char(amt));
        End;

OUTPUT
Enter value for empn: 1001
old 6: empn:=&empn;
new 6: empn:=1001;
Enter value for amt: 1000
old 8: amt:=&amt;
new 8: amt:=1000;
Enter employee number
Enter salary increment
Salary of 1001increased by 1000
PL/SQL procedure successfully completed.
```

```
SQL> select * from employee;

EMPID      EMPNAME              SALARY     PID
---------- -------------------- ---------- ----------
1001       Ravi                 26500      2
1002       Rajesh               30000      1
1003       Ramesh               22500      3
1004       Ratheesh             16000      5
```