DDL Commands

QUERY:

```
Create table student( Roll_no int ,Name varchar(20),Branch varchar(20),Minor varchar(20),Address varchar(20));

 Insert into student values(20,’Anurudh’,’CSE’,’ECE’,’Koorkenchery’),
                              (54,’Nobel’,’ECE’,’ME’,’Amala Nagar’),
                              (19,’Anurag’,’ME’,’EEE’,’Ariyannur’),
                              (62,’Joyal’,’CE’,’ECE’,’Thriprayar’),
                              (44,’Akshay’,’EEE’,’ME’,’Amballur’);

Select * from student;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
20	Anurudh	CSE	ECE	Koorkencherry
54	Nobel	ECE	ME	Amala Nagar
19	Anurag	ME	EEE	Ariyannur
62	Joyal	CE	ECE	Thriprayar
44	akshay	EEE	ME	Amballur

QUERY:

```
Alter table student add column Age int;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address	Age
20	Anurudh	CSE	ECE	Koorkencherry	NULL
54	Nobel	ECE	ME	Amala Nagar	NULL
62	Joyal	CE	ECE	Thriprayar	NULL

44	akshay	EEE	ME	Amballur	NULL

QUERY:

Drop table student;

OUTPUT:

Table dropped

DML Commands

QUERY:

```
Create table student( Roll_no int ,Name varchar(20),Branch varchar(20),Minor varchar(20),Address varchar(20));

 Insert into student values(20,’Anurudh’,’CSE’,’ECE’,’koorkenchery’),
                         (54,’Nobel’,’ECE’,’ME’,’Amala Nagar’),
                         (19,’Anurag’,’ME’,’EEE’,’Ariyannur’),
                         (62,’Joyal’,’CE’,’ECE’,’Thriprayar’),
                         (44,’Akshay’,’EEE’,’ME’,’Amballur’);

Select * from student;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
20	Anurudh	CSE	ECE	Koorkencherry
54	Nobel	ECE	ME	Amala Nagar
19	Anurag	ME	EEE	Ariyannur
62	Joyal	CE	ECE	Thriprayar
44	akshay	EEE	ME	Amballur

QUERY:

```
   Update student set Address =’choondal’ where Name=’Anurag’;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
20	Anurudh	CSE	ECE	Koorkencherry
54	Nobel	ECE	ME	Amala Nagar
19	Anurag	ME	EEE	Choondal
62	Joyal	CE	ECE	Thriprayar

44	akshay	EEE	ME	Amballur

QUERY:

```
Delete from student where Roll_no=’19’;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
20	Anurudh	CSE	ECE	Koorkencherry
54	Nobel	ECE	ME	Amala Nagar
62	Joyal	CE	ECE	Thriprayar
44	akshay	EEE	ME	Amballur

ORDER BY, GROUP BY, HAVING

QUERY:

```
Create table student( Roll_no int ,Name varchar(20),Branch varchar(20),Minor varchar(20),Address varchar(20));

 Insert into student values(20,’Anurudh’,’CSE’,’ECE’,’koorkenchery’),
                              (54,’Nobel’,’ECE’,’ME’,’Amala Nagar’),
                              (19,’Anurag’,’ME’,’EEE’,’Ariyannur’),
                              (62,’Joyal’,’CE’,’ECE’,’Thriprayar’),
                              (44,’Akshay’,’EEE’,’ME’,’Amballur’);

Select * from student;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
20	Anurudh	CSE	ECE	Koorkencherry
54	Nobel	ECE	ME	Amala Nagar
19	Anurag	ME	EEE	Ariyannur
62	Joyal	CE	ECE	Thriprayar
44	Akshay	EEE	ME	Amballur

QUERY:

```
Select * from student order by name;
```

OUTPUT:

Roll_no	Name	Branch	Minor	Address
44	Akshay	EEE	ME	Amballur
19	Anurag	ME	EEE	Ariyannur
20	Anurudh	CSE	ECE	Koorkencherry

54	Nobel	ECE	ME	Amala Nagar
62	Joyal	CE	ECE	Thriprayar

QUERY:

Select * from student order by name DESC;

OUTPUT:

Roll_no	Name	Branch	Minor	Address
62	Joyal	CE	ECE	Thriprayar
54	Nobel	ECE	ME	Amala Nagar
20	Anurudh	CSE	ECE	Koorkencherry
19	Anurag	ME	EEE	Ariyannur
44	Akshay	EEE	ME	Amballur

QUERY:

Select Branch, count(Branch)

from student

group by Branch

having branch='CSE';

OUTPUT:

Branch	Count(Branch)
CSE	1

AGGREGATE FUNCTIONS

QUERY:

```
Create table employee (E_ID Int, E_NAME varchar(20),Email varchar(20), Department varchar(20),Salary Int);

Insert into employee values (111,’Anurag’,’anurag@gmail.com’,’CSE’,12000),
                            (222,’Nobel’,’nobel@gmail.com’,’ME’,20000),
                            (333,’Anurudh’,anurudh@gmail.com’,’EEE’,30000),
                            (444,’Joyal’,joyal@gmail.com’,’ECE’,40000),
                            (555,’Akshay’,akshay@gmail.com’,’CE’,50000);

 Select * from employee;
```

E_ID	E_NAME	Email	Department	Salary
111	Anurag	anurag@gmail.com	CSE	12000
222	Nobel	nobel@gmail.com	ME	50000
333	Anurudh	anurudh@gmail.com	EEE	30500
444	Joyal	joyal@gmail.com	ECE	40000
555	Akshay	akshay@gmail.com	CE	20000

```
Select count(salary) from employee;
```

Count(salary)
5

```
Select sum(salary) from employee;
```

Sum(salary)
50000

```
Select avg(salary) from employee;
```

Avg(salary)
30500

```
Select max(salary) from employee;
```

Max(salary)
50000

```
Select min(salary) from employee;
```

Min(salary)
12000

```
Select department,count(department) from employee group by dept having department=’CSE’;
```

Department	Count(Department)
CSE	1

```
Select * from employee where salary in(select max(salary) from employee);
```

E_ID	E_NAME	Email	Department	Salary
20	Nobel	nobel@gmail.com	ME	50000

SET OPERATION

QUERY:

create database Vidya;

use Vidya;

create table t1(rollno int, firstname varchar(20),lastname varchar(20),birthdate date);

create table t2(rollno int,present int,absent int,percentage varchar(10));

insert into values(1001,’Mike’,’Richard’,19960201),(1002,'Robert','Williams',19950503),

(1003,'Peter','Collin',19940104),(1004,'Alexa','Stuart',19950611),(1005,'Robert','Peterson',1997011);

insert into values(1001,200,20,’90%’), (1002,160,60,’72%’), (1003,150,70,’68%’), (1004,210,10,’95%’);

Select * from t1;

rollno	firstname	lastname	birthdate
1001	Mike	Richard	1996-02-01
1002	Robert	Williams	1995-05-03
1003	Peter	Collin	1994-01-04
1004	Alexa	Stuart	1995-06-11
1005	Robert	Peterson	1997-01-12

Select * from t2;

rollno	present	absent	percentage
1001	200	20	90%
1002	160	60	72%
1003	150	70	68%
1004	210	10	95%

UNION

Select rollno from t1 union select rollno from t2;

rollno
1001
1002
1003
1004
1005

UNIONALL

Select rollno from t1 unionall select rollno from t2;

rollno
1001
1002
1003
1004
1005
1001
1002
1003
1004

INNERJOIN

Select t1.rollno,firstname,percentage from t1 inner join t2 on t1.rollno = t2.rollno;

rollno	firstname	percentage
1001	Mike	90%
1002	Robert	72%
1003	Peter	68%
1004	Alexa	95%

LEFTJOIN

Select t1.rollno,firstname,percentage from t1 left join t2 on t1.rollno = t2.rollno;

rollno	firstname	percentage
1001	Mike	90%
1002	Robert	72%
1003	Peter	68%
1004	Alexa	95%
1005	Robert	NULL

RIGHTJOIN

Select t1.rollno,firstname,percentage from t1 right join t2 on t1.rollno = t2.rollno;

rollno	firstname	percentage
1001	Mike	90%
1002	Robert	72%
1003	Peter	68%
1004	Alexa	95%

CROSSJOIN

Select t1.rollno,firstname,percentage from t1 cross join t2;

rollno	firstname	percentage	present	absent
1001	Mike	Richard	200	20
1001	Mike	Richard	160	60
1001	Mike	Richard	150	70
1001	Mike	Richard	210	10
1002	Robert	Williams	200	20
1002	Robert	Williams	160	60
1002	Robert	Williams	150	70
1002	Robert	Williams	210	10
1003	Peter	Collins	200	20
1003	Peter	Collins	160	60
1003	Peter	Collins	150	70
1003	Peter	Collins	210	10
1004	Alexa	Stuart	200	20
1004	Alexa	Stuart	160	60
1004	Alexa	Stuart	150	70
1004	Alexa	Stuart	210	10
1005	Robert	Peterson	200	20
1005	Robert	Peterson	160	60
1005	Robert	Peterson	150	70
1005	Robert	Peterson	210	10

INTERSECTION

```
Select distinct(rollno) from t1 inner join t2 using (rollno);
```

rollno
1001
1002
1003
1004

VIEWS AND ASSERTION

QUERY:

```
Use Vidya;

Create table t1(rno int,fname varchar(20),lname varchar(20));

Create table t2(rno int,percentage int,mark int);

Insert into t1 values(1001,’Mike’,’Richard’),(1002,’Robert’,’William’),(1003,’Peter’,’Cotton’);

Insert into t2 values(1001,90,96),(1002,75,50),(1003,85,56);

Select * from t1;
```

rno	fname	lname
1001	Mike	Richard
1002	Robert	William
1003	Peter	Cotton

```
Select * from t2;
```

rno	percentage	mark
1001	90	96
1002	75	50
1003	85	56

Views

```
Create view t3 as

Select rno,fname

From t1;

Select * from t3;
```

rno	fname
1001	Mike
1002	Robert
1003	Peter

```
Create view t4 as

Select s.frame,a.percentage,a.mark

From t1, s,t2 a

where s.rno=a.rno;
```

```
select * from t4;
```

fname	percentage	mark
Mike	90	96
Robert	75	50
Peter	85	56

Assertion

```
Create assertion new

Check not exists

(select * from t3

Where rno is NULL);
```

PL/SQL CONTROL STRUCTURES

IF-THEN

PROGRAM:

```
DECLARE
   age integer:=20;
BEGIN
   if(age>=18) then
      dbms_output.put_line('Eligible to Vote');
   end if;
END;
```

OUTPUT:

```
Eligible to Vote
```

IF-THEN-ELSE

PROGRAM:

```
DECLARE
   age integer:=16;
BEGIN
   if(age>=18) then
      dbms_output.put_line('Eligible to Vote');
   else
      dbms_output.put_line('Not Eligible to Vote');
   end if;
END;
```

OUTPUT:

```
Not Eligible to Vote
```

IF-THEN-ELSIF

PROGRAM:

```
DECLARE
```

```
   NO1 INTEGER:= 16;
   NO2 INTEGER:= 29;
   NO3 INTEGER:= 1;
BEGIN
   if NO1>NO2 AND NO1>NO2 then
      dbms_output.put_line('NO1 IS GREATER');
   elsif NO2>NO3 then
      dbms_output.put_line('NO2 IS GREATER');
   else
      dbms_output.put_line('NO3 IS GREATER');
   end if;
END;
```

OUTPUT:

```
NO2 IS GREATER
```

CASE

PROGRAM:

```
DECLARE
   DAY INTEGER:=3;
BEGIN
   CASE DAY
   when 1 then dbms_output.put_line('MONDAY');
   when 2 then dbms_output.put_line('TUESDAY');
   when 3 then dbms_output.put_line('WEDNESDAY');
   when 4 then dbms_output.put_line('THURSDAY');
   when 5 then dbms_output.put_line('FRIDAY');
   when 6 then dbms_output.put_line('SATURDAY');
   when 7 then dbms_output.put_line('SUNDAY');
   else dbms_output.put_line('NO SUCH DAY');
   end if;
END;
```

OUTPUT:

```
WEDNESDAY
```

WHILE LOOP

PROGRAM:

```
DECLARE
   NUM INTEGER:=2;
BEGIN
   while NUM<=10 loop
      dbms_output.put_line(num);
      num := num+2;
   end loop;
END;
```

OUTPUT:

```
2
4
6
8
10
```

PROCEDURES

PROGRAM:

```
DECLARE a
number; b
number;
PROCEDURE S(a in number,b in number,c out number)
IS
BEGIN
c:=a+b;
END;
BEGIN
a:=5;
 b:=6;
S(a,b,c);
dbms_output.put_line('Sum:'||c);
END;
```

OUTPUT:

```
Sum:11
```

<u>EXCEPTION HANDLING</u>

PROGRAM:

```
DECLARE
   a INTEGER:=15;
   b INTEGER:=0;
BEGIN
   a:=a/b;
   dbms_output.put_line(a);
EXCEPTION
   when ZERO.DIVIDE
then
   dbms_output.put_line('You divided by Zero');
END;
```

OUTPUT:

```
You divided by Zero
```

TRIGGER 1

PROGRAM:

```
Create Table Customer(ID int, Name Varchar(20),Salary Float);

Insert Into Customer values(7,'A',75000);(8,'B',50000);(9,'C',80000);

CREATE OR REPLACE TRIGGER display_salary_changes

BEFORE DELETE OR INSERT OR UPDATE ON Customer

FOR EACH ROW

WHEN (NEW.ID>0)

DECLARE

   sal_diff number;

BEGIN

   sal_diff:=NEW.Salary - :OLD.Salary;

   dbms_output.put_line('Old Salary:'||:OLD.Salary);

   dbms_output.put_line('New Salary:'||:NEW.Salary);

   dbms_output.put_line('Salary DIfference:'||:sal_diff);

END;

UPDATE Customer Set Salary = Salary + 500 where ID = 7;
```

OUTPUT:

```
1  row(s)  updated

Old   Salary:75000

New Salary:75500

Salary Difference:500
```

TRIGGER 2

PROGRAM:

```
CREATE TABLE T1(ID INT,AGE INT);

CREATE TABLE T2(COUNTER INT);

INSERT INTO T2 VALUES(0);

CREATE OR REPLACE PROCEDURE CI

AS

BEGIN

        UPDATE T2 set COUNTER = COUNTER + 1;

END;

CREATE OR REPLACE TRIGGER TRI

AFTER INSERT ON T1

FOR EACH ROW

BEGIN

        NEW PROC;

END;

INSERT INTO T1 VALUES(12,7);

INSERT INTO T1 VALUES(13,8);

SELECT * FROM T2;
```

OUTPUT:

COUNTER
2

CURSORS

PROGRAM:

```
create table customers(id int,name varchar(20),address varchar(30));
insert into customers values(100,'John','New York');
insert into customers values(102,'Ben','Florida');
insert into customers values(105,'Peter','Chicago');
DECLARE
  c_id customers.id%type;
  c_name customers.name%type;
  c_addr customers.address%type;
  CURSOR c_customers is
    SELECT id, name, address FROM customers;
BEGIN
  OPEN c_customers;
  LOOP
  FETCH c_customers into c_id, c_name, c_addr;
    EXIT WHEN c_customers%notfound;
    dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
  END LOOP;
  CLOSE c_customers;
END;
/
```

OUTPUT:

```
100  John  New York
102  Ben   Florida
105  Peter  Chicago
```

SQL TCL COMMANDS

PROGRAM:

```
create table players(id int, name varchar(20), age int, address varchar(20),salary int);
```

```
insert into players values(1,’Messi’,35,’Argentina’,70000),(2,’Mbappe’,23,’France’,50000),
(3,’Neymar’,30,’Brazil’,45000), (4,’Ronaldo’,37,’Portugal’,60000), (5,’Modric’,35,’Croatia’,35000);
```

```
select * from players;
```

id	name	age	address	salary
1	Messi	35	Argentina	70000
2	Mbappe	23	France	50000
3	Neymar	30	Brazil	45000
4	Ronaldo	37	Portugal	60000
5	Modric	35	Croatia	35000

```
start transaction;
```

```
savepoint sample;
```

```
delete from players where age=35;
```

```
select * from players;
```

id	name	age	address	salary
2	Mbappe	23	France	50000
3	Neymar	30	Brazil	45000
4	Ronaldo	37	Portugal	60000

```
rollback to savepoint sample;
```

```
select * from players;
```

id	name	age	address	salary
1	Messi	35	Argentina	70000
2	Mbappe	23	France	50000
3	Neymar	30	Brazil	45000
4	Ronaldo	37	Portugal	60000
5	Modric	35	Croatia	35000

```
commit;
```