

When you want to combine output of more than one query, then we use union or intersection or minus

intersection and minus does not work in mysql, but it works in oracle.

union works in mysql

1. All the queries in union should have equal number of columns and corresponding columns data type should match.
2. In the output the column names of topmost query will be visible.

example

```
select * from emp_india
```

```
-> union
```

```
-> select * from emp_japan
```

```
-> union
```

```
-> select * from emp_US;
```

In the following example the first query has 4 columns, and the second query has 3 column, hence in second query we add a dummy column, by adding null

```
select * from emp_india
```

```
union
```

```
select id,name,location,null from emp_US;
```

PL_SQL—(Procedural language -Structured Query Language)

procedure	any block of code , which has business logic is called as procedure
function	any block of code , which has business logic, and returns one values, is called as functions. these can be used in select clause and where clause in SQL
triggers	any block of code , which has business logic, and gets called on some users action automatically, then it is called as trigger
exception	While executing procedures if any error occurs, we use exceptions
cursors	When you want to traverse through all rows one by one, and perform some action on each row, on by one, then use cursors

Types of parameters to the procedure

in	these are read only parameters. these are default parameters. these are used to send the input value to the procedure
out	these are write only parameters these are used to get the output from the procedure
inout	these are both read and write parameters, we can send the value to the procedure, and inside procedure we may change the value of the parametr

Why we use PL SQL

1. we can hide table names from the developer of the middleware application, which increases the security of the database.
2. For a particular task, if we need to execute many queries, then we may wrap these queries in a procedure, and call the procedure from middleware application, once, execute all the queries, complete the task and go back, this will reduce the network traffic, also improves performance efficiency of the middleware application. so it reduces the interaction between middleware program and database.
3. If any of the query is complex, then we may hide the query inside the procedure
4. Procedures will also reduce the network traffic.

delimiter //

create procedure <procedure name>(parameters...)

begin

declaration of variable;

statement1;

statement2;

end//

delimiter ;

call <procedurename>();

1. to insert record into dept table

delimiter //

create procedure insertdept(did int,dnm varchar(20),dloc varchar(20))

begin

insert into dept values(did,dnm,dloc);

end//

delimiter ;

call insertdept(10,'admin','pune')

2. write a procedure to accept eid, sal and job from user as i/p and update sal and job of the employee in emp table

delimiter //

create procedure updateemp(eid int,esal float(9,2),ejob varchar(20))

begin

update emp

set sal=esal,job=ejob

where empno=eid;

end//

delimiter ;

call updateemp(7902,6666,'QA');

3. write a procedure finddata, to get sal and comm of the employee

delimiter //

create procedure findjob(eid int,out esal float(9,2),out ecomm float(9,2))

```

begin
    select sal,comm into esal,ecomm
    from emp
    where empno=eid;
end//

```

delimiter ;

```

call findjob(7902,@s,@c)
select @s,@c

```

in above example, select ... into statement can be used only inside pl sql blocks, the select query should return single row as output. number of column names before into and number of variables after into should be same.

@s and @c are session variables. these variables will remain available till the time you logout.

4. write a procedure to find number of employees and maximum netsalary for the given department.

```

net sal =sal+comm
delimiter //
create procedure findemp(in edid int, out cnt int,out maxsal float(9,2))
begin
    select count(*),max(sal+ifnull(comm,0)) into cnt,maxsal
    from emp
    where deptno=edid;
end//
delimiter ;

```

```

call findemp(10,@c,@ms)

```

5. write a procedure which will accept a number and increment a number by 10

```

delimiter //
create procedure incrementnum(inout cnt int)
begin
    set cnt=cnt+10;
select cnt;
end//
delimiter ;

```

```

set @c=5
call incrementnum(@c)
select @c;

```

6. write a procedure to display all employees in given department and sal >1500.

```

delimiter //
create procedure getempdata(in edid int,in esal float(9,2))
begin
select *
from emp

```

```

where deptno=edid and sal>esal;
end//
delimiter ;

```

```

call getempdata(10,1500);

```

7. write a procedure to find all employees along with dname with sal>2000

```

delimiter //
create procedure findempdetails(esal float(9,2))
begin
    select empno,ename,sal,e.deptno,dname
    from emp e,dept d
    where e.deptno=d.deptno and sal >esal;
end//
delimiter ;

```

```

call findempdetails(2000);

```

8. display feedback based on comm
 if comm is null or 0 then display “poor performance”
 if comm <=300 then display ‘ok performance’
 if com >301 and <=500 then display good performance
 else display excellent performance.

<pre> if condition then statements; else statements end if; </pre>	<pre> if condition then statements; elseif condition then statements else statements end if; </pre>
--	---

```

delimiter //
create procedure getRemark(eid int,out remark varchar(50))
begin
    declare vcomm float(9,2) default 0;
    select comm into vcomm
    from emp
    where empno=eid;
    if vcomm is null or vcomm=0 then
        set remark='poor performance';
    elseif vcomm<= 300 then
        set remark='ok performance';
    elseif vcomm<= 500 then
        set remark='good performance';
    else
        set remark='excellent performance';
    end if;
end//

```

```
end//  
delimiter ;
```

9. write a procedure to find netsal of the given employee and find the remark, if
netsal <1000 “less”
if >=1000 and <2000 then ‘ok’
if netsal >=2000 and < 3000 then ‘good’
otherwise better
display remark inside the procedure

```
netsal= sal+comm  
delimiter //  
create procedure findNetsal(eid int,out remark varchar(50))  
begin  
    declare vsal,vcomm,vnetsal float(9,2);  
    select sal,comm into vsal,vcomm  
    from emp  
    where empno=eid;  
  
    set vnetsal=vsal+ifnull(vcomm,0);  
  
    if vnetsal<1000 then  
        set remark ='less';  
  
    elseif vnetsal<2000 then  
        set remark='ok';  
  
    elseif vnetsal<3000 then  
        set remark='good';  
  
    else  
        set remark='better';  
  
    end if;  
    select eid,vsal,vcomm,vnetsal,remark;  
end//  
  
delimiter ;
```

10. write a procedure getdiscount to find discount % and discounted amount from product table for the given product

if price < 50 then 3%

if price >=50 and <80 7%

if price >=80 and < 100 8%

otherwise 12%

display pid,pname,price,discount percentage and discount amount

pid	pname	qty	price
1	lays	34	50.00
2	prigles	50	150.00
3	nice	53	50.00
4	maggi	45	60.00

delimiter //

create procedure getdiscount(dpid int, out discount float(4,2))

begin

declare vpname varchar(20) default '';

declare vprice float(9,2);

select pname, price into vpname,vprice

from product

where pid=dpid;

if vprice<50 then

set discount=0.03;

elseif vprice<80 then

set discount=0.07;

elseif vprice <100 then

set discount=0.08;

else

set discount=0.12;

end if;

select dpid , vpname,vprice,vprice-(vprice*discount),discount;

end//

delimiter ;

loops in plsql

while expression then statements; end while;	It is a top tested loop, statements inside loops will get executed till the condition is true.
repeat statements until expression end repeate	It is a bottom tested loop, and statements inside this loop will get executed until the condition is false.
label1: loop if condition then leave label1 endloop;	<p>This is infinite loop, and will continue execution till leave statement gets executed, leave statement is same as break statement, it stops the loop forcefully</p> <p>In the loop you may use iterate statement, it is similar to continue statement.</p> <p>It will transfer the control at the beginning of the loop</p>