

-----Mysql If else statement

```
IF expression THEN
    statements;
ELSE
    else-statements;
END IF;
```

Using If ----else-----

```
IF expression THEN

    statements;

ELSEIF elseif-expression THEN

    elseif-statements;

...

ELSE

    else-statements;

END IF;
```

-----write a program to update sal of the employee and display status for the given empno

If sal<1500 then update sal by 20%

Else if sal <2500 usdate sal by 15%

Otherwise update by 10%

Empno,newsal,status

delimiter //

create procedure updatesal(in eno int,out nsal double(10,2),out status varchar(20))

begin

declare vsal double(20,2);

select sal into vsal

from emp

where empno=eno;

if vsal<1500 then

```

set nsal=vsal*1.20;

set status='updated by 20%';

update emp

set sal=nsal

where empno=eno;

elseif vsal<2500 then

set nsal=vsal*1.15;

set status='updated by 15%';

update emp

set sal=nsal

where empno=eno;

else

set status='updated by 10%';

set nsal=vsal*1.10;

        update emp

set sal=nsal

where empno=eno;

end if;

select nsal,status;

end//

```

-----using loops

1. While loop

```

While loop-----
WHILE expression DO
statements
END WHILE

```

```

-----to write a procedure to display 1,2,3,4,5
delimiter //
create procedure whiledemo(num int)
begin
declare n int default 1;
declare str varchar(20) default "";
while n<=num do

```

```

        set str=concat(str,',',n);
        set n=n+1;
    end while;
    set str=substr(str,2);
    select str;
end//
delimiter ;

```

-----use repeat until loop to display 1,2,3,4,5

The loop will repeated at least once, and will continue repeating till the condition is false;

```

REPEAT
    statements;
UNTIL expression
END REPEAT

```

```

delimiter //
create procedure repeatdemo(num int)
begin
declare n int default 1;
declare str varchar(50) default "";
repeat
    set str=concat(str,',',n);
    set n=n+1;
    until n>num
end repeat;
set str=substr(str,2);
select str;
end//
delimiter ;

```

---using loop..... end loop

leave label1 ----similar to break statement in java, it terminates the loop with the given lable

Iterate ---→ similar to continue.

label2:loop

label1: Loop

If condition then

Leave lable2

Else

Iterate label1;

End if

End loop;

End loop;

---to print 1,2,...,num

```

delimiter //

```

```

create procedure loopydemo(num int)

begin

declare n int default 1;

declare str varchar(50) default "";

label1:loop

    set str=concat(str,',',n);

    set n=n+1;

    if n>num then

        leave label1;

    else

        iterate label1;

    end if;

end loop;

set str=substr(str,2);

select str;

end//

delimiter ;

```

-----to display pattern pass number of lines

<pre> * ** *** **** </pre>	<pre> delimiter // create procedure trianglepattern(num int) begin declare str varchar(50) default ""; declare n int default 1; while n<=num do set str=concat(str,'\n',lpad("",n,'*')); set n=n+1; </pre>
----------------------------	--

	<pre> end while; select str; end// delimiter ; </pre>
<pre> * * * * * * * * * * * * * * * * * * * * * * * * * </pre>	<pre> drop procedure diamondpattern; delimiter // create procedure diamondpattern(num int) begin declare str varchar(50) default ''; declare n int default 1; declare scnt int default 0; set scnt=floor(num/2); while n<=num do set str=concat(str,'\n',lpad('',scnt,' '),lpad('',n,'*')); set n=n+2; set scnt=scnt-1; select n,scnt,str; end while; set scnt=1; set n=num-2; while n>=1 do set str=concat(str,'\n',lpad('',scnt,' '),lpad('',n,'*')); set n=n-2; set scnt=scnt+1; select n,scnt,str; </pre>

	end while; select str; end// delimiter ;

Using cursors in mysql

If you need to write any select statement which returns multiple rows, then use cursors

Step 1

Declare all the variables, to hold the values from cursor

1. Declare cursor
 DECLARE cursor_name CURSOR FOR SELECT_statement;
 DECLARE empcurr CURSOR FOR select empno,ename,sal from emp;
2. Declare continue handler to exit the loop
 Declare continue handler for NOT FOUND set vstop=1;
3. Open the cursor
 Open empcurr;
4. Navigate through the cursor
 Fetch empcurr into vempno,venm,vsal;
5. If not found then break the loop
 If vstop=1 then
 leave label1
 End if
6. Process the data,
 Continue steps4 to 6 till the loop executes
7. Close the cursor
 Close empcurr;

-----display all employees with salary > avg(sal) of their own department
 delimiter //

create procedure displayemp()

begin

declare veno,vdeptno,vstop int default 0;

declare venm,vjob varchar(20);

```

declare vsal,vavg double(10,2);

declare empcurr cursor for select empno,ename,job,sal,deptno from emp;

declare continue handler for NOT FOUND set vstop=1;

open empcurr;

label1: loop

            fetch empcurr into veno,venm,vjob,vsal,vdeptno;

            if vstop=1 then

                leave label1;

            end if;

            select avg(sal) into vavg

            from emp

            where deptno=vdeptno;

            if vsal<vavg then

                select veno,venm,vjob,vsal,vavg,vdeptno;

            end if;

end loop;

close empcurr;

end//

delimiter ;

-----writing functions

SET GLOBAL log_bin_trust_function_creators = 1;

delimiter //

create function calcexp(hdate date) returns int

begin

    return floor(datediff(curdate(),hdate)/365);

```

```
end//
```

```
delimiter ;
```

```
---write a function to generate email by concatenating first 3 alphabets of ename, followed by .,  
followed by 3 characters from job, followed by @mycompany.com
```

```
create function generateemail(enm varchar(20),job varchar(20))
```

```
returns varchar(30)
```

```
begin
```

```
    return concat(substr(enm,1,3),".",substr(job,1,3),'@mycompany.com');
```

```
end//
```

```
--- to find functions list
```

```
show function status where db='iacsd0923';
```

```
--- to find procedures list
```

```
show procedure status where db='iacsd0923';
```

```
-----to see the definition of function or procedure
```

```
SELECT ROUTINE_DEFINITION
```

```
FROM information_schema.ROUTINES WHERE
```

```
SPECIFIC_NAME='cal_bonus';
```

```
create table emp_audit(
```

```
deptno int, olddname varchar(20),
```

```
newdname varchar(20),
```

```
oldloc varchar(20),
```

```
newloc varchar(20),
```

```
uname varchar(20),
```

```
op_date datetime
```

```
action varchar(20));
```



```
create trigger dept_monitor
before update on dept for each row
begin
    insert into dept_audit values(old.deptno,
    old.dname,
    new.dname,
    old.loc,
    new.loc,
    user(),
    now(),
    'insert');

end//
```

```
create trigger dept_monitor1
before delete on dept for each row
begin
    insert into dept_audit values(old.deptno,
    old.dname
    null,
    old.loc,
    null,
    user(),
    now(),
    'delete');
```

```
end//  
  
create trigger dept_monitor1  
  
before insert on dept for each row  
  
begin  
  
    insert into dept_audit values(new.deptno,  
  
    nul,,  
  
    new.dname,  
  
    null,  
  
    new.loc,  
  
    user(),  
  
    now(),  
  
    'insert');  
  
end//
```