What are indexes in database

On primary key and on unique key the indexes will be created automatically.

2 types of indexes

1. Clustered index
    a. There is only one clustered index
    b. It is stored along with table data. So, the data will be ordered in the table on the column
2. Non clustered index
    a. There can be more than one non clustered index, for which separate file will be created and is stored outside the table
1. To create indexes

Create index sal_idx

On emp(sal desc,deptno)

2. To delete the index

alter table category

drop index idx_cname;

3. To find all indices on a particular table

Show index from emp;

4. Create unique index on mobile in emp table
   If you create unique index on mobile column then it will add unique constraint on the mobile column, and duplicate values are not allowed in the column

Create unique index idx_mobile

On emp(mobile)

Differences between clustered and no n clustered index

Clusterd	Non clustered
There is only one clustered index	There are many non clustered index
This is stored inside table so no extra space is needed	This is stored ouside table so extra space is needed

Views

Views are logical table based on some base query

Advantages:

1. To give access to only restricted data from a table
2. To hide table names for security purpose
3. To hide complexity of the query

DML operations are allowed only on simple views , the view that is based on single table is called as simple view

The view with base query contains union, aggregate functions, or more than one table joins, nested queries then it is automatically read only

1. Create a view prod_cat to display all categories names and product names
   Create view produ_cat
   As
   Select cname,pname
   From category c , product p
   Where c.cid=p.cid;

2. To create materialized view
   If you want to store o/p of base query in RAM temporarily, then use materialized view.

Create materialized view produ_cat_1

as

select * from product;

Temporary table

Create temporary table mytab_temp

(id int primary key,

Name varchar(20));

PL-SQL --→ procedural Language SQL

1. Since this language contains if, loops and hence called as procedural language
2. In this we can write 3 types blocks
    a. Procedure --- this is block of code which does not return the value
    b. Function--- this is block of code which returns one value
    c. Trigger---- this is block of code which gets executed automatically, when some DML statement gets executed

Procedures

To a procedure we can pass 3 types of parameters

1. In
    a. These parameters are used for passing input
    b. These are read only parameters
    c. Its values cannot be changed inside the procedure
2. Out
    a. These parameters are used for get output outside the procedure
    b. These are write only parameters
    c. Its values can be changed inside the procedure

3. Inout
    a. These parameters are used for passing input and getting o/p
    b. These are read-write parameters
    c. Its values can be read and also can be changed inside the procedure

1. Write a procedure to find how many employees are there in the given department.

```
Delimiter //
Create procedure getcnt(in pdno  int, out pcnt int)
Begin
   Select count(*) into pcnt
    From emp
    Where deptno=pdno;

End//
Delimiter ;

To call the procedure
Call getcnt(10,@c);
Select @c;
```

2. Write a procedure to find avg(sal) of all employees with given job

```
Delimiter //
Create procedure getavgsal(pjob varchar(20),out pavg decimal(9,2))
Begin
  Select avg(sal) into pavg
  From emp
 Where job=pjob;
End//
Delimiter ;

Call getavgsal('CLERK',@c);
Select @c
```

3. To write a procedure to insert record in a table dept

```
Delimiter //
Create procedure insdept(pdno int,pdnm varchar(20),ploc varchar(20))
Begin
    Insert into dept values(pdno,pdnm,ploc);
End//
Delimiter ;
```

4. Write a procedure to delete all employees with sal > given salary

```
Delimiter //
Create procedure delemployee(in givensal decimal(9,2))
Begin
  Delete from emp
  Where sal > givensal;
End//
Delimiter ;

Call delemployee(2000);
```

if-----else statement

--------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;
```

Example-------

1. To check whether given number is > 10 or not

```
Delimiter //
Create procedure checkdata(v int)
Begin
   If v> 10 then
      Select “greater”
  Elseif v=10 then
      Select ‘equal’;
   Else
      Select “smaller”
   End if;
End//
```

2. To display remark of given empno based on comm
   If comm is null or 0 then ‘need improvement
   Else if comm < 500 ‘ok’
   Elseif comm>=500 and <100 then ‘good’
   Otherwise excellent

```
Delimiter //
Create procedure getremark(peno int)
Begin
    Declare vcomm decimal(9,2) ;
   Select comm into vcomm
   From emp
    Where empno=peno;
     If vcomm is null or vcomm=0 then
        Select ‘need improvement’;
     Elseif vcomm<500 then
       Select “ok”;
     Elseif vcomm <1000 then
    Select “good”;
    Else
          Select ‘excellent’;
End if;


End//
Delimiter ;
```