

Indexes

- Indexes are used to search data faster
- Indexes gets automatically created for primary key and unique key constraints
- Unique indexes do not allow to add duplicate values
- We will not be able to see the index files
- Which index to use for searching will be decided by search engines in mysql(INNODB,MYISAM)
- Indexes are always in sorted order of keys, can be arranged in ascending or descending order

2 types of indexes

1. Clustered index
2. Non clustered index

Clustered vs non clustered

Clustered	Non clustered
There is only one clustered index	There will be many non-clustered index
Clustered indexes are stored with data	These indexes are stored externally
These indexes require no extra space	These indexes require extra space

Indexes always stores key and the position of the key in the table

Types of indexes

primary	Does not allow duplicate keys and null values
unique	Does not allow duplicate keys
fulltext	Searches word, usually used in search engines, works only on text,char and varchar type data
regular	Indexes can be stored in ascending or descending order, can be based on one or more keys

Syntax

Create index idx_dname_loc

On dept(dname)

To create unique index

Create unique index idx_deptno

On dept(dname)

To delete index idx_deptno

Drop index idx_deptno

To see the list of all indexes

show indexes from <table name>;

show indexes from dept;

To find which index is used by your query, then use Explain

explain select *

-> from dept

-> where deptno between 10 and 40;

Views

- Views are virtual table based on base query
- If you fire a query on views then actually base query will get executed

Why use views

1. Give limited access to few columns or rows of existing table.
2. To hide table names, to increase security
3. To hide complexity of the query

Create view:

Create view mgr10

As

Select *

From emp

Where deptno=10;

To create a view so that DML operations will work only for deptno =10

Create view mgr10

-> As

-> Select *

-> From emp

-> Where deptno=10

-> with check option;

Views can be readonly, but readonly views are not supported by mysql, but works in oracle.

To delete views

Drop view mgr10;

Views which contains all not null columns, and it is based on single table, and it does not contain aggregate functions, or does not contains group by statement, or does not contain union

create view all_emp

-> as

-> select * from emp_us

```

-> union
-> select * from emp_india
-> union
-> select * from emp_japan;

```

To create materialized view
create materialized view all_emp

```

-> as
-> select * from emp_us
-> union
-> select * from emp_india
-> union
-> select * from emp_japan;

```

PL-SQL(procedural language structured query language)

Procedures

Functions

Triggers

Exception

Cursors

Statements

If statement

Loops

While

For

Repeat until

Loop..endloop

Procedures	These are blocks of code which gets executed on the database server
Functions	These are blocks of code which gets executed on the database server and it returns one value Functions can be called in select statement and where clause
Triggers	These are block of code which gets executed automatically, when some DML operation gets executed

Why are using PL SQL

1. We can hide table names from developers.
2. Complexity of queries can be hidden.
3. We may add multiple queries in one procedure and give the o/p to java/python program, so interactions between middleware programs and database server can be reduced.
4. Which also reduces network traffic
5. It increases security, by hiding table names from developers

Delimiter //

Create procedure <name of procedure>(parameters)

Begin

Select * from emp;

End//

delimiter //

Create procedure myproc()

Begin

Select * from emp;

End//

delimiter ;

Call myproc();

Delimiter //

Create procedure insrec(pid int,pname varchar(20),qty int,price double(9,2))

Begin

Insert into product values(pid,pname,qty,price);

End//

Dlimiter ;

Call insrec(10,'50-50',34,56)

-----to find number of employees in dept 10

Select count(*) from emp

Where deptno=10

In a procedure we can pass 3 types of parameters

in	By default the parameters are of type in, these are read only parameters and used only for reading data
out	these are write only parameters and used only for getting data as o/p
inout	These read and write parameters, can be used for sending data i/p and getting modified information as o/p

Delimiter //

Create procedure getcnt_by_dept(pdno int,out cnt int)

Begin

Select count(*) into cnt from emp where deptno= pdno;

End//

Delimiter ;

Call getcnt_by_dept(10,@c)

Select @c;

Delimiter //

Create procedure getcnt_by_dept(pdno int,out cnt int,out minsal double(9,2))

Begin

Select count(*),min(sal) into cnt,minsal from emp where deptno= pdno;

End//

Delimiter ;

Call getcnt_by_dept(10,@c,@m);

Select @c,@m;

.....

delimiter //

create procedure get_cnt_min(in dno int,out cnt int,out minsal double(9,2))

begin

select count(*) ,min(sal) into cnt,minsal

from emp

where deptno=dno;

end//

delimiter ;

----write procedure to increase count by 10

delimiter //

create procedure incCNT(inout cnt int)

begin

set cnt =cnt+10;

end//

delimiter ;

set c=5

call incCNT(@c)

```
select @c;
```

write a procedure to display all employees in given dept and sal > given sal

```
delimiter //
```

```
create procedure disp_data(dno int,s double(9,2))
```

```
begin
```

```
    select *
```

```
    from emp
```

```
    where deptno=dno and sal>s;
```

```
end//
```

```
delimiter ;
```

```
call disp_data(10,2000);
```

----- write a procedure to assign remark based on comm

If comm is null or =0 then 'need improvement'

Else if comm<300 then 'ok'

Else if comm>=300 and comm <500 then 'good'

Otherwise 'excellent'

```
Delimiter //
```

```
Create procedure get_remark(peno int, out remark varchar(20))
```

```
Begin
```

```
Declare vcomm int default 0;
```

```
Select comm into vcomm
```

```
From emp
```

```
Where empno=peno;
```

```
If vcomm is null or vcomm=0 then
```

```
    Set remark='need improvement';
```

```
Elseif vcomm<300 then
```

```
    Set remark='ok';
```

```
Elseif vcomm>=300 and vcomm<500 then
```

```
    Set remark='good';
```

```
Else
```

```
    Set remark='excellent';
```

```
End if;
Select remark
End//
Delimiter ;
Call get_remark(7902,@remark );
Select @remark;
```

-----write a procedure to find net Sal of employee by using formula sal+comm for given empno

```
If netsal <1000 then "less"
```

```
Else netsal >=1000 and <2000 'ok'
```

```
Else netsal >=2000 and <3000 'good'
```

```
Else "better"
```

```
Delimiter //
```

```
Create procedure get_sal(pempno int,out psc double(9,2),out remark varchar(20))
```

```
Begin
```

```
Select sal+ifnull(comm,0) into psc
```

```
From emp where empno=pempno;
```

```
If psc <1000 then
```

```
    Set remark='less';
```

```
Elseif psc >=1000 and psc <2000 then
```

```
    Set remark='ok';
```

```
Elseif psc >=2000 and psc <3000 then
```

```
Set remark='good';
```

```
Else
```

```
    Set remark='better';
```

```
End if;
```

```
Select psc,remark;
```

```
End//
```