use msdb

select \* from sysjobactivity;

select \* from sysjobs;

select \* from sysjobschedules;

select \* from sysjobhistory;

select \* from syscategories;

select \* from sysschedules;

select \* from sysjobs\_view;

select \* from sys.database\_permissions;

select \* from sysusers;

select \* from sys.server\_principals ;

select \* from sysjobschedules;

SELECT CONVERT(VARCHAR(8), next\_run\_time, 108) from sysjobschedules;

----------------to count all jobs------------------------------------------------

EXEC msdb.dbo.sp\_help\_job

------------------------to count total no of jobs--------------------------------

select count(\*) from msdb.dbo.sysjobs;

------------------------to count NO. of enabled job------------------------------

                        select count(\*) NUMBER\_OF\_ENABLED\_JOB from sysjobs

                        where enabled='1';

                        select name from sysjobs

                        where enabled='1';

                        Exec sp\_get\_composite\_job\_info @enabled = 1 -- For enabled jobs

-----------------------to count NO. of Disabled job--------------------------------

                        select count(\*) NUMBER\_OF\_Disabled\_JOB from sysjobs

                        where enabled='0';

                        select name from sysjobs

                        where enabled='0';

                        Exec sp\_get\_composite\_job\_info @enabled = 0 -- For disabled jobs

--For job status

                        select distinct j.name,

                          CASE jh.run\_status

                              WHEN 0 THEN 'Error Failed'

                              WHEN 1 THEN 'Succeeded'

                              WHEN 2 THEN 'Retry'

                              WHEN 3 THEN 'Cancelled'

                              WHEN 4 THEN 'In Progress'

                              when 5 then  'Status Unknown'

                              END AS 'Last\_Run\_Status'

                       from  sysjobs j left join sysjobhistory jh

                       on j.job\_id=jh.job\_id

                       where run\_status='1'

                       --where run\_status is null

                      --where run\_status='1'or

                      --run\_status is null

                        select distinct j.name

                          --CASE jh.run\_status

                          --    WHEN 0 THEN 'Error Failed'

                          --    WHEN 1 THEN 'Succeeded'

                          --    WHEN 2 THEN 'Retry'

                          --    WHEN 3 THEN 'Cancelled'

                          --    WHEN 4 THEN 'In Progress'

                          --    when 5 then  'Status Unknown'

                          --    END AS 'Last\_Run\_Status'

                       from  sysjobs j left join sysjobhistory jh

                       on j.job\_id=jh.job\_id

                      where run\_status=0

--for job frequency(DAILY,WEEKIL)

                  select s.name,s.freq\_type,

                     CASE s.freq\_type

                        WHEN  1 THEN 'Once'

                        WHEN  4 THEN 'Daily'

                        WHEN  8 THEN 'Weekly'

                        WHEN 16 THEN 'Monthly'

                        WHEN 32 THEN 'Monthly relative'

                        WHEN 64 THEN 'When SQL Server Agent starts'

                        WHEN 128 THEN 'Start whenever the CPU(s) become idle' END as Occurs

                  from    sysschedules s

--for 1-once,2-daily,8-weekly,16-monthly,32-monthly relative,64-when sql server agent start,128-Start whenever the CPU(s) become idle

                    select s.name

                    from   sysschedules s

                    where freq\_type='8';

 --to view the next run date

                    select distinct j.name,convert(varchar(11),sj.next\_scheduled\_run\_date,110) as Next\_Run\_Date

                    from sysjobactivity sj join sysjobs j

                    on sj.job\_id = j.job\_id

                    WHERE sj.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity)

 --to view the last run date

                   select distinct j.name, convert(varchar(11),sj.run\_requested\_date,110) as Last\_Run\_Date

                   from sysjobactivity sj join sysjobs j

                   on sj.job\_id = j.job\_id

                   WHERE sj.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity)

-- to view the next & last run  Date

                  select distinct j.name, convert(varchar(11),sj.run\_requested\_date,110) as Last\_Run\_Date,convert(varchar(11),sj.next\_scheduled\_run\_date,110) as Next\_Run\_Date

                  from sysjobactivity sj join sysjobs j

                  on sj.job\_id = j.job\_id

                  WHERE sj.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity)

--to view the job is scheduled or not

                    select  distinct   j.name ,

                      CASE ss.enabled WHEN 0 THEN 'No'

                      when 1 then  'Yes' END as Scheduled

                     from sysschedules SS right  join sysjobs j

                    on ss.owner\_sid=j.owner\_sid

       ---------to find the run duration----------

       SELECT j.job\_id, j.name, h.run\_duration FROM sysjobs j

INNER JOIN sysjobhistory h ON h.job\_id=j.job\_id

INNER JOIN (

SELECT job\_id, MAX(STR(run\_date,8)+STR(run\_time,8)) as LastRunDate

FROM sysjobhistory GROUP BY job\_id

) x ON j.job\_id=x.job\_id

WHERE h.step\_id=0

AND STR(h.run\_date,8)+STR(h.run\_time,8)=x.LastRunDate

---------------------------------------all view User name & there permission------------------------------

select dp.name,su.permission\_name,su.state\_desc

from sysusers dp inner join sys.database\_permissions su

on grantee\_principal\_id=uid

where dp.uid > 4

 ---------------------or--------------------------

 SELECT prin.[name] [User], sec.state\_desc + ' ' + sec.permission\_name [Permission]

FROM [sys].[database\_permissions] sec

  JOIN [sys].[database\_principals] prin

    ON sec.[grantee\_principal\_id] = prin.[principal\_id]

WHERE sec.class = 0

ORDER BY [User], [Permission];

---------------------or-----------------------------------------

Create Table #Temp\_Users

(

DatabaseUserName varchar(128),

SYSLoginname varchar(128),

LoginType varchar(50),

Roles varchar(1024)

)

Create Table #Temp\_Roles

(

Name varchar(128),

Role varchar(128)

)

insert into #Temp\_Users

select a.name,

b.Loginname,

LoginType = case

when a.IsNTName = 1 then 'Windows Account'

when a.IsNTGroup = 1 then 'Windows Group'

when a.isSqlUser = 1 then 'SQL Server User'

when a.isAliased =1 then 'Aliased'

when a.isSQLRole = 1 then 'SQL Role'

when a.isAppRole = 1 then 'Application Role'

else 'Unknown'

end,

Roles = ''

from sysusers a

left outer join master..syslogins b on a.sid=b.sid

where a.SID is not null

order by a.Name

insert into #Temp\_Roles

select MemberName = u.name, DbRole = g.name

from sysusers u, sysusers g, sysmembers m

where g.uid = m.groupuid

and g.issqlrole = 1

and u.uid = m.memberuid

order by 1, 2

Declare @Name varchar(128)

Declare @Roles varchar(1024)

Declare @Role varchar(128)

DECLARE UserCursor CURSOR for

SELECT DatabaseUserName from #Temp\_Users

OPEN UserCursor

FETCH NEXT FROM UserCursor into @Name

WHILE @@FETCH\_STATUS = 0

BEGIN

set @Roles = ''

print @Name

DECLARE RoleCursor CURSOR for

SELECT Role from #Temp\_Roles where Name = @Name

OPEN RoleCursor

FETCH NEXT FROM RoleCursor into @Role

WHILE @@FETCH\_STATUS = 0

BEGIN

if (@Roles > '')

set @Roles = @Roles + ', '+@Role

else

set @Roles = @Role

FETCH NEXT FROM RoleCursor into @Role

end

Close RoleCursor

DEALLOCATE RoleCursor

Update #Temp\_Users set Roles = @Roles where DatabaseUserName = @Name

FETCH NEXT FROM UserCursor into @Name

END

CLOSE UserCursor

DEALLOCATE UserCursor

update #Temp\_Users

set Roles='public'

where Roles=''

select \* from #Temp\_Users where logintype='SQL Server User' and sysloginname is not null

drop table #Temp\_Users

drop table #Temp\_Roles

--------------------------------------to view users name, roles and databases and login information

sp\_helpuser

sp\_helpuser

sp\_helplogins

----------------------------------to view dbname & there size

EXEC sp\_databases

----------------------------to find userName and there roles-------------

SELECT users.name as UserName, groups.name as UserRole

FROM sysmembers membs

JOIN sysusers users on membs.memberuid = users.uid

JOIN sysusers groups on membs.groupuid = groups.uid

where users.uid > 4

--OR-----------

SELECT p.NAME as userRole

,m.NAME as UserNAme

FROM sys.database\_role\_members rm

JOIN sys.database\_principals p

ON rm.role\_principal\_id = p.principal\_id

JOIN sys.database\_principals m

ON rm.member\_principal\_id = m.principal\_id

where rm.member\_principal\_id > 4

----------------------to view all job details

SELECT Convert(varchar(20),SERVERPROPERTY('ServerName')) AS ServerName,

j.name AS Job\_Name,

CASE j.enabled WHEN 1 THEN 'Enabled' Else 'Disabled' END AS Job\_Status,

CASE jh.run\_status WHEN 0 THEN 'Error Failed'

                   WHEN 1 THEN 'Succeeded'

                   WHEN 2 THEN 'Retry'

                   WHEN 3 THEN 'Cancelled'

                   WHEN 4 THEN 'In Progress' ELSE

                  'Status Unknown' END AS 'Job\_Run\_Status',

                  convert(varchar(11),ja.run\_requested\_date,110) as Last\_Run\_Date,

CONVERT(VARCHAR(10),CONVERT(DATETIME,RTRIM(19000101))+(jh.run\_duration \* 9 + jh.run\_duration % 10000 \* 6 + jh.run\_duration % 100 \* 10) / 216e4,108) AS Run\_Duration,

convert(varchar(11),ja.next\_scheduled\_run\_date,110)as Next\_Schedule\_Run\_Date,

CONVERT(VARCHAR(500),jh.message) AS Job\_Description

FROM

(msdb.dbo.sysjobactivity ja LEFT JOIN msdb.dbo.sysjobhistory jh ON ja.job\_history\_id = jh.instance\_id)

join msdb.dbo.sysjobs\_view j on ja.job\_id = j.job\_id

WHERE ja.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity) ORDER BY job\_name,job\_status

-------------to view all dbname ,size

dbcc sqlperf("logspace")

---------------------------------------------------------------new Query------------------------------------------

------------------------------------------------------------------------------------------------------------------

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SQL Query to find the Summary of Jobs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*      \*/

      Select Distinct(server) ServerName from msdb.dbo.sysjobhistory

      select (select count(\*) from msdb.dbo.sysjobs) as Total\_Jobs,

      (select count(\*) from msdb.dbo.sysjobs where enabled = 1) as Enabled\_Jobs,

      (select count(\*) from msdb.dbo.sysjobs where enabled = 0) as  Disabled\_Jobs

      SELECT CASE jh.Run\_status WHEN 0 THEN 'Failed'

      WHEN 1 THEN 'Succeeded'

      WHEN 2 THEN 'Retry'

      WHEN 3 THEN 'Cancelled'

      WHEN 4 THEN 'In Progress' ELSE

      'Status Unknown' END AS 'Last\_run\_status'

      , Count(\*) As Count

      FROM

      (msdb.dbo.sysjobactivity ja LEFT JOIN msdb.dbo.sysjobhistory jh ON ja.job\_history\_id = jh.instance\_id)

      join msdb.dbo.sysjobs\_view j on ja.job\_id = j.job\_id

      WHERE ja.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity)

      Group By Run\_status

      Order By Count(\*)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of SQL Query to find the Summary of Jobs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* Query for SQL Jobs details \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SELECT --Convert(varchar(20),SERVERPROPERTY('ServerName')) AS ServerName,

j.name AS Job\_name,CASE jh.Run\_status WHEN 0 THEN 'Failed'

                  WHEN 1 THEN 'Succeeded'

                  WHEN 2 THEN 'Retry'

                  WHEN 3 THEN 'Cancelled'

                  WHEN 4 THEN 'In Progress' ELSE

                  'Status Unknown' END AS 'Last\_run\_status',

CONVERT(VARCHAR(500),jh.message) AS Message,

ja.Next\_scheduled\_run\_date As Next\_run\_date,

ja.run\_requested\_date as StartTime,ja.stop\_execution\_date as EndTime,

CONVERT(VARCHAR(10),CONVERT(DATETIME,RTRIM(19000101))+(jh.run\_duration \* 9 + jh.run\_duration % 10000 \* 6 + jh.run\_duration % 100 \* 10) / 216e4,108) AS Run\_duration,

CASE j.enabled WHEN 1 THEN 'Enabled' Else 'Disabled' END AS Job\_status

FROM

(msdb.dbo.sysjobactivity ja LEFT JOIN msdb.dbo.sysjobhistory jh ON ja.job\_history\_id = jh.instance\_id)

join msdb.dbo.sysjobs\_view j on ja.job\_id = j.job\_id

WHERE ja.session\_id=(SELECT MAX(session\_id)  from msdb.dbo.sysjobactivity)

--And ja.run\_requested\_date = getdate()-1 /\*  Remove comment for getting previous day job reports \*/

ORDER BY job\_name,job\_status

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Query for SQL Jobs details \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* Command for LogSpace details  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

DBCC SQLPERF(logspace)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Command for LogSpace details  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

--to list all store procedure from db

exec sp\_stored\_procedures

--or

SELECT \*

FROM sys.procedures;

--or

SELECT obj.Name as SPName,

modu.definition as SPDefinition,

obj.create\_date as SPCreationDate

FROM sys.sql\_modules modu

INNER JOIN sys.objects obj

ON modu.object\_id = obj.object\_id

WHERE obj.type = 'P'

--------------------to view the dbname logicalname physicalname size

SELECT DB\_NAME(database\_id) AS DatabaseName,

Name AS Logical\_Name,

Physical\_Name, (size\*8)/1024 SizeMB

FROM sys.master\_files

-----------------

-----------to show the drive name & there free size

xp\_fixeddrives

--------------------------

--to show physical name ,size,all ad information

select name as 'File Name', physical\_name as 'Physical Name', size/128 as 'Total size in MB',

size/128.0 - cast(fileproperty(name, 'spaceUsed') as int)/128.0 as 'Available space in MB', max\_size/128.0 'Max Size in MB', \*

from sys.database\_FILES