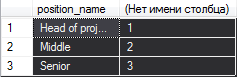


/\*1.\*/

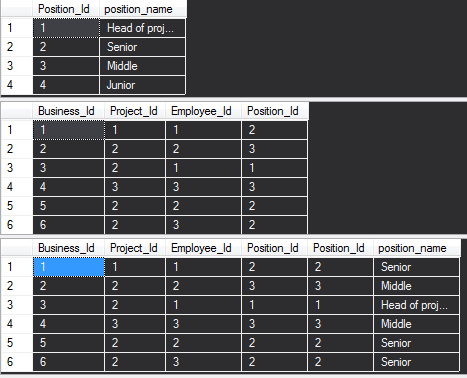
select position\_name, count(position\_name) from Project\_employment as pe left join Position p

on pe.Position\_Id = p.Position\_Id

group by position\_name;



/\*2.\*/

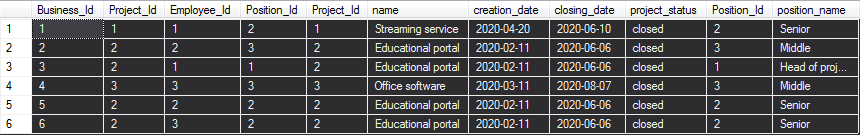


select position\_name from Position p

where p.Position\_Id not in (select Position\_Id from Project\_employment);



/\*3\*/



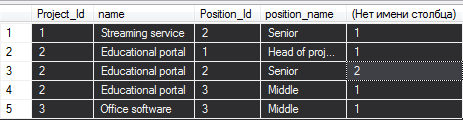
select \* from Project\_employment;

select pe.Project\_Id, p.name, pe.Position\_Id, po.position\_name, count(\*) from Project\_employment pe

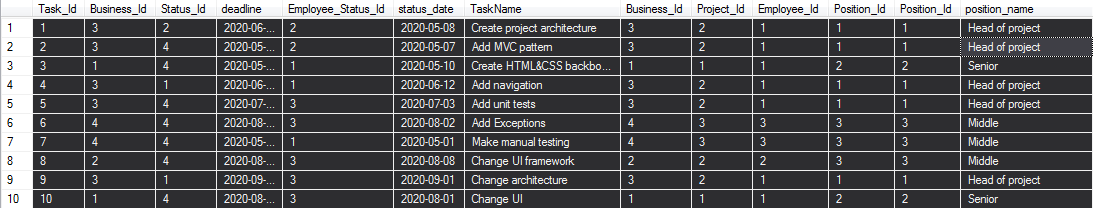
left join Project p on p.Project\_Id = pe.Project\_Id

left join Position po on po.Position\_Id = pe.Position\_Id

group by pe.Project\_Id, p.name, pe.Position\_Id, po.position\_name;



/\* 4\*/



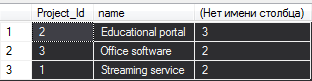
select f.Project\_Id, f.name, avg(f.c) from (select pe.Project\_Id, p.name, pe.Employee\_Id, count(\*) c from Task t

join Project\_employment pe on t.Business\_Id = pe.Business\_Id

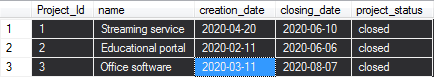
join Project p on pe.Project\_Id = p.Project\_Id

group by pe.Project\_Id, p.name, pe.Employee\_Id) as f

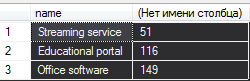
group by f.Project\_Id, f.name;



/\* 5 \*/



select p.name, DateDiff(DAY, p.creation\_date, p.closing\_date) from Project p;



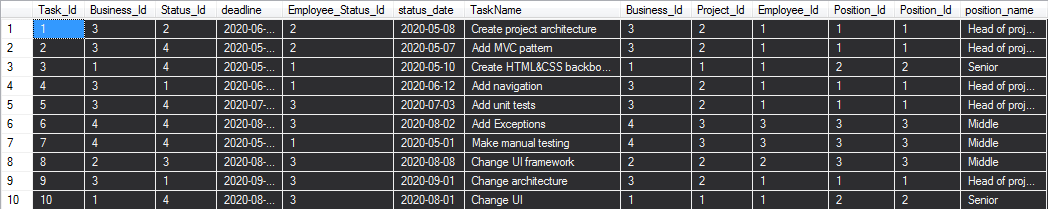
/\* 6\*/

Update Task

Set deadline = '2020-08-31',

Status\_Id = 3

where Task\_Id = 8;



create function GetObj

(

)

returns table

as

return

(

select f.Employee\_Id, count(\*) as c2 from (select pe.Employee\_Id, count(\*) as c from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

group by pe.Employee\_Id, t.Status\_Id

having Status\_Id <> 4) as f

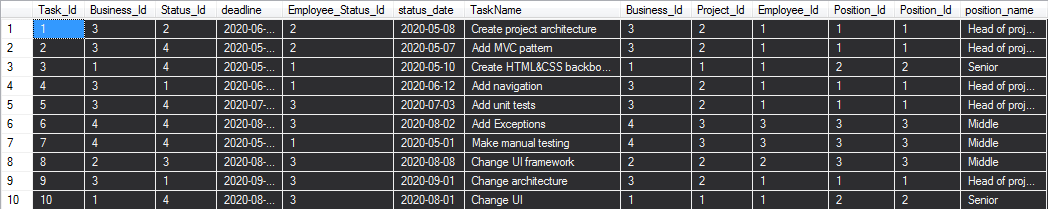
group by f.Employee\_Id);

select p.Employee\_Id, p.c2 from GetObj() p

where p.c2 = (select min(p2.c2) from GetObj() p2);



/\* 7 \*/



create function GetObjMax

(

)

returns table

as

return

(

select f.Employee\_Id, count(\*) as c2 from (select pe.Employee\_Id, count(\*) as c from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

where t.deadline < CONVERT(date, getdate())

group by pe.Employee\_Id, t.Status\_Id) as f

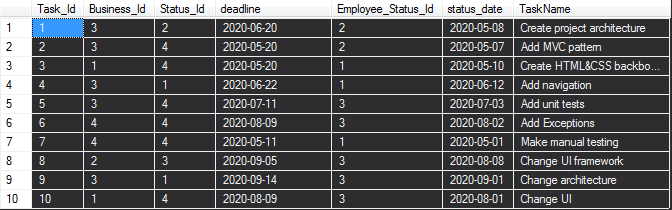
group by f.Employee\_Id);

select p.Employee\_Id, p.c2 from GetObjMax() p

where p.c2 = (select max(p2.c2) from GetObjMax() p2);



/\* 8 \*/



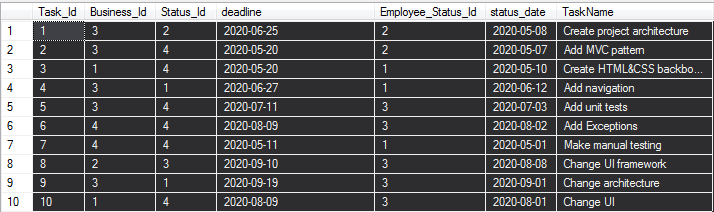
Update Task

set deadline = DATEADD(day, 5, deadline)

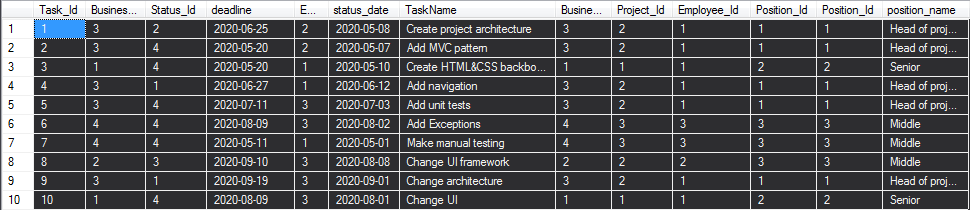
where Task\_Id in (

select t1.Task\_Id from Task t1

where t1.Status\_Id <> 4);



/\*9\*/



select p.Project\_Id, p.name, count(\*) as opened from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

group by p.Project\_Id, p.name, t.Status\_Id

having Status\_Id = 1;



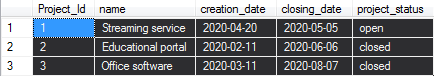
/\* 10 \*/

update Project

set project\_status = 'open',

closing\_date= '2020-05-05'

where Project\_Id = 1;



CREATE FUNCTION Task10Id

(

)

RETURNS TABLE AS RETURN

(

select p2.Project\_Id as id, p2.m as minim, p2.d2 as d from

(select p1.Project\_Id, min(p1.Status\_Id) as m, max(p1.d1) as d2 from

(select p.Project\_Id, t.Status\_Id, max(t.deadline) as d1, count(\*) as c from Project p

left join Project\_employment pe on pe.Project\_Id= p.Project\_Id

left join Task t on t.Business\_Id = pe.Business\_Id

group by p.Project\_Id, t.Status\_Id, t.deadline) as p1

group by p1.Project\_Id) as p2

where p2.m = 4

);

update Project

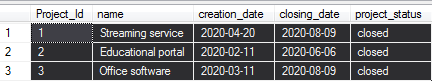
set project\_status = 'closed',

closing\_date = res.d

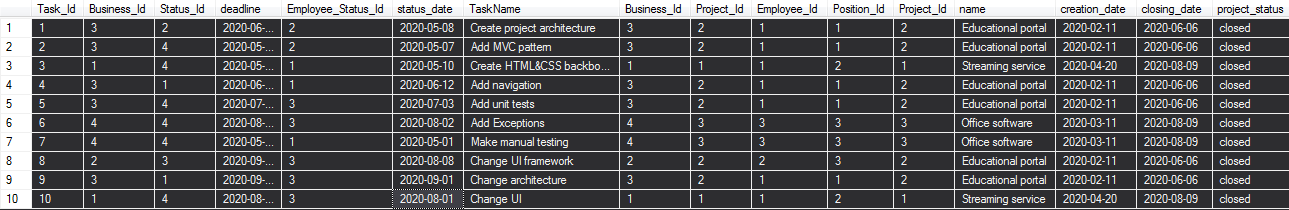
from (select \* from Task10Id()

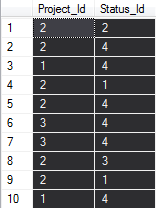
) as res

where res.id = Project.Project\_Id;



/\*11\*/





CREATE FUNCTION Task11Id

(

)

RETURNS table return

(select t1.Project\_Id, t1.Employee\_Id, min(t1.Status\_Id) as m from (select p.Project\_Id, pe.Employee\_Id, t.Status\_Id from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

group by p.Project\_Id, pe.Employee\_Id, t.Status\_Id) as t1

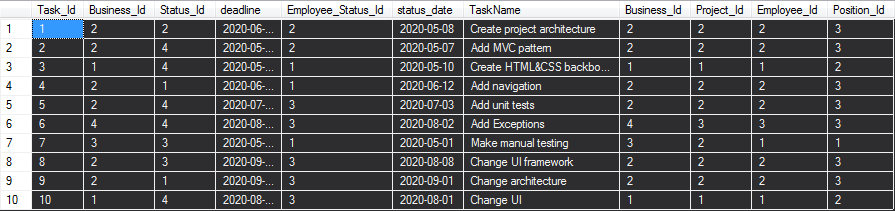
group by t1.Project\_Id, t1.Employee\_Id);

select \* from Task11Id() as t

where t.m = 4;



/\* 12 \*/



declare @index int;

declare @projectId int;

declare @businessTask int;

declare @businessEmployee int;

CREATE procedure GetTaskByName(

@name varchar(50),

@projectId int output,

@id int output,

@businessId int output

)

as

begin

set @id = (select top 1 Task\_Id from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

where TaskName = @name);

set @projectId = (select top 1 p.Project\_Id from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

where TaskName = @name);

set @businessId = (select top 1 t.Business\_Id from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

where TaskName = @name);

end;

exec GetTaskByName 'Make manual testing', @projectId output, @index output, @businessTask output;

select @index, @projectId, @businessTask;



CREATE function GetSpecificWorker

(@ProjectId integer

)

returns table

as

return

(

select pp.Project\_Id, pp.Employee\_Id, pp.b as business, pp.c as cnt from (select pe.Project\_Id, pe.Employee\_Id, pe.Business\_Id as b, count(\*) as c from Task t

left join Project\_employment pe on pe.Business\_Id = t.Business\_Id

left join Project p on pe.Project\_Id = p.Project\_Id

group by pe.Project\_Id, pe.Employee\_Id, pe.Business\_Id

) as pp

where pp.Project\_Id = @projectId

);

update Task

set Business\_Id = (select min(pii.business) from GetSpecificWorker(@projectId) as pii

where pii.cnt = (select min(p3.cnt) from GetSpecificWorker(@projectId) p3))

where Business\_Id = @businessTask;

