Lab assignment : 01

Sample Table – Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Rana	Hamid	100000	2014-02-20 09:00:00	HR
2	Sanjoy	Saha	80000	2014-06-11 09:00:00	Admin
3	Mahmudul	Hasan	300000	2014-02-20 09:00:00	HR
4	Asad	Zaman	500000	2014-02-20 09:00:00	Admin
5	Sajib	Mia	500000	2014-06-11 09:00:00	Admin
6	Alamgir	Kabir	200000	2014-06-11 09:00:00	Account
7	Foridul	Islam	75000	2014-01-20 09:00:00	Account
8	Keshob	Ray	90000	2014-04-11 09:00:00	Admin

1. Write an SQL query to print first three characters of FIRST_NAME from Worker table.
2. Write an SQL query to print details of the Workers who have joined from Feb 2014 to March 2014.
3. Write an SQL query to print details of the Workers who have served for at least 6 months.
4. Write an SQL query to update all worker salary whose title is manager.
5. Write an SQL query to update all worker bonus 10% whose joining_date before '2014 04-11 09:00:00' otherwise bonus update 5% and also check department name is 'Admin'.
6. Write an SQL query to delete all workers who have not taken any bonus.
7. Write an SQL query to print details for Workers with the first name "Rana" and "Sajib" from Worker table.
8. Write an SQL query to print details of workers excluding first names, "Rana" and "Sajib" from Worker table.
9. Write an SQL query to print details of the Workers whose FIRST_NAME contains 'a'.
10. Write an SQL query to print details of the Workers whose FIRST_NAME starts with 'k'.
11. Write an SQL query to print details of the Workers whose FIRST_NAME ends with 'r' and contains seven alphabets.
12. Write an SQL query to find the position of the alphabet ('n') in the FIRST_NAME column 'Sanjoy' from Worker table.
13. Find the average salary of employees for each department.
14. List all the employees who have maximum or minimum salary in each department
15. Write an SQL query to find the position of the alphabet ('r') in the FIRST_NAME column 'Rana' from Worker table.
16. Write an SQL query to print the FIRST_NAME from Worker table after removing white spaces from the right side.
17. Write an SQL query that fetches the unique values of FIRST_NAME from Worker table and prints its length.
18. Write an SQL query to print the FIRST_NAME from Worker table after replacing 'a' with 'A'.

Solve query

```
----1
select left(first_name , 3) as first_three_characters
from worker ;
---2
select * from worker where joinning_date between
'2014-02-01' and '2014-03-01';
```

```
---3
select * from worker where
datediff(month,joinning_date,getdate())>=60 ;
---4
update worker set salary=salary-20000 from
worker join title on worker.worker_id=title.worker_ref_id
where worker_title ='Manager';
---5
update worker set salary=salary+(salary*
case
when joinning_date <'2014-04-11' and department='Admin' then
0.10
when joinning_date>='2014-04-11' and department='Admin' then
0.05
else 0
end
);
---6
delete from worker where department !='Admin';
select * from worker
where department='Admin' and joinning_date >= '2014-04-11';
----or
delete from worker where salary !=.10 and salary !=.05
----or
delete from worker where not exists (select 1 from bonus
where worker.worker_id=bonus.worker_ref_id);
--7
select * from worker where first_name in('Rana','sajib')
--8
select * from worker where first_name not in('Rana','sajib')
----9
select * from worker where first_name like '%a%'
---10
select * from worker where first_name like 'k%'
---11
select * from worker where first_name like '%r' and
len(first_name)=7
----12
select first_name ,charindex('n', first_name) from worker
where first_name='sanjoy'

----13
select department , avg(salary) as average from worker
group by department
```

```sql
----14
select * from worker where salary in (select max(salary)
from worker group by department) or salary in (select
min(salary)
from worker group by department)
---15
select charindex('n',first_name) from worker
where first_name='Rana';
---16
SELECT rtrim(FIRST_NAME) FROM Worker;
---17
select distinct first_name,len(first_name) from worker
---18
select replace(first_name,'a','A') from worker
```

Lab assignment : 02

Sample Table – Worker

WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	Rana	Hamid	100000	2014-02-20 09:00:00	HR
2	Sanjoy	Saha	80000	2014-06-11 09:00:00	Admin
3	Mahmudul	Hasan	300000	2014-02-20 09:00:00	HR
4	Asad	Zaman	500000	2014-02-20 09:00:00	Admin
5	Sajib	Mia	500000	2014-06-11 09:00:00	Admin
6	Alamgir	Kabir	200000	2014-06-11 09:00:00	Account
7	Foridul	Islam	75000	2014-01-20 09:00:00	Account
8	Keshob	Ray	90000	2014-04-11 09:00:00	Admin

Sample Table – Bonus

WORKER_REF_ID	BONUS_DATE	BONUS_AMOUNT
1	2019-02-20	5000
2	2019-06-11	3000
3	2019-02-20	4000
4	2019-02-20	4500
5	2019-06-11	3500
6	2019-06-12	NULL

Sample Table – Title

WORKER_REF_ID	WORKER_TITLE	AFFECTED_FROM
1	Manager	2019-02-20
2	Executive	2019-06-11
8	Executive	2019-06-11
5	Manager	2019-06-11
4	Asst. Manager	2019-06-11
7	Executive	2019-06-11
6	Lead	2019-06-11
3	Lead	2019-06-11

1. List all the employees except ‘Manager’ & ‘Asst. Manager’.
2. List the workers in the ascending order of Designations of those joined after April 2014.
3. Write an SQL query to fetch the number of employees working in the department ‘Admin’.
4. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.
5. Write an SQL query to fetch the no. of workers for each department in the descending order.
6. Write an SQL query to print details of the Workers who are also Managers.
7. Write an SQL query to show only odd rows from a table.
8. Write an SQL query to show only even rows from a table.
9. Write an SQL query to clone a new table from another table.
10. Write an SQL query to show the current date and time.
11. Write an SQL query to show the top n (say 10) records of a table with Name and Designation.
12. Write an SQL query to determine the nth (say n=5) highest salary from a table.
13. Write an SQL query to fetch the list of employees with the same salary.
14. Write an SQL query to show the second highest salary from a table.
15. Write an SQL query to fetch the first 50% records from a table.
16. Write an SQL query to fetch the departments that have less than five people in it.
17. Write an SQL query to show all departments along with the number of people in there.
18. Write an SQL query to show the last record from table.
19. Write an SQL query to fetch the first row of a table.
20. Write an SQL query to fetch the last five records from table.
21. Write an SQL query to print the name of employees having the highest salary in each department.
22. Write an SQL query to fetch three max salaries from table.

Solve query:

```
----1
select * from worker join title
on worker.worker_id=title.worker_ref_id
where worker_title not in ('Manager','Asst. Manager');
---2
select * from worker where joinning_date > '2014-04-01'
order by worker_id asc
----3
select count(*) from worker where department = 'Admin';
--4
select first_name,salary from worker
where salary >=50000 and salary<=100000;
---5
select department ,count(*) from worker group by department
order by count(*) desc ;
--6
select * from worker join title on
worker.worker_id=title.worker_ref_id
where worker_title='Manager'
--7
select * from worker where worker_id%2! =0 ;
---8
select * from worker where worker_id%2 =0 ;
--9
select * into newtable from worker;
--10
select getdate();
--11
select Top 5 worker.first_name ,title.worker_title from worker
join title on worker.worker_id=title.worker_ref_id
---12
select distinct top 1 salary from  (select
distinct top 5 salary from worker order by
salary desc ) as work order by salary asc
---13
select first_name,salary from worker where salary
in (select salary from worker group by salary
having count(*)>1)
--14
select max(salary) from worker where
salary<(select max(salary)from worker)
--15
select top 50 percent * from worker order by worker_id
```

```
---16
select * from worker where department in(select
department,count(*)
from worker group by department having count(*)<5)
---17
select department,count(*) from worker group by
department
---18
select top 1 worker_id from worker order by worker_id desc
---19
select top 1 * from worker order by worker_id asc
--20
select top 5 * from worker order by worker_id desc
--21
select first_name , salary,department from worker
where salary in (select distinct max(salary) from worker
group by department)
---22
select distinct top 3 salary from worker
order by salary desc
```

Lab assignment : 03

TID	FirstName	LastName	Dept	Age	Salary
1	Mizanur	Rahman	CSE	28	35000
2	Delwar	Hossain	CSE	26	33000
3	Shafiul	Islam	EEE	24	30000
4	Faisal	Imran	CSE	30	50000
5	Ahsan	Habib	English	28	28000

deptID	deptName	location
1	CSE	Talaimari
2	EEE	Talaimari
3	English	Kazla
4	BBA	Talaimari

1. Update the Salary of Teacher by 15% whose DeptName is 'CSE, otherwise update by 10% Salary.
2. Write a query to insert/copy the values of all attributes from one table to another using (ID in) subquery.
3. Write a query to find firstname and lastname as fullname , age whose salary is maximum.
4. Write a query to find firstname, age,dept whose age is between 23 to 27.
5. Write a query to find TID,firstname whose salary is less than average salary.
6. Write a query to update Dept by 'English' where Dept is 'EEE' using subquery.
7. Write a query to update salary by multiplying the salary by 100 where salary is greater than 5000 using subquery..
8. Write a query to find the name that starts with 'k/s' using a subquery.
9. Find the Firstname,salary for all the teachers of CSE who have a higher salary than Delwar Hossain using subquery.
10. Find out the id,names of all teachers who belong to the same department as the teacher 'Mizanur' .
11. Find TID, salary, deptID whose salary is greater than average salary
12. Find min salary from Teacher for each department where min salary is less than average salary
13. Find firstname,lastname,Dept where location name is kajla using subquery.
14. Write a query to find the TID,firsname,salary where the length of the firstname is at least 6.

Solve query:

```
---1
update teacher set salary=salary+(salary*
case
   when dept ='CSE' then 0.15
   when dept ! ='CSE' then 0.10
   else 0
   end );
----2
select * into newTable from teacher where tid in
(select tid from teacher)
```

```sql
select * from newTable
---3
select  firstName +space(2)+lastName as fullname ,
age from teacher
where salary=(select max(salary) from teacher)
--4
select firstName ,age ,dept from teacher
where age between 23 and 27
---5
select tid,firstName from teacher where
salary <(select avg(salary) from teacher)
---6
update teacher set dept ='English' where dept in
(select dept from teacher where dept='EEE')
---7
update teacher set salary=salary*100
where salary in (select salary from teacher
where salary>5000)
---8
select firstName from teacher where
firstName in (select firstName from teacher
where firstName like 's%' or firstName like 'k%');
---9
select firstName,salary from teacher where
dept='CSE' and salary>(select salary from teacher
where firstName='Delwar')
--10
select tid,firstname from teacher where dept=(select dept
from teacher
where firstname='Mizanur')
----11
select t.tid,t.salary,e.deptid from teacher t join employee e
on t.tid=e.deptid where salary >(select avg(salary)
from teacher )
---12
select dept ,min(salary) from teacher
group by dept having
min(salary)<(select avg(salary) from teacher)
---13
select teacher.firstName,teacher.lastName,
teacher.dept from teacher join employee on
teacher.tid=employee.deptid where teacher.tid in
(select employee.deptid from employee
where employee.location='kazla')
---14
select tid,firstName,salary from teacher
where len(firstName) in (select len(firstName)
from teacher where len(firstName)>6)
```

Lab assignment : 04

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen	Berlin	0.12
5007	Paul Adam	Rome	0.13

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London	300	5005
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	5003
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moscow	200	5007

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

1. Write a query to create a view for those salesmen belongs to the city New York.
2. Write a query to create a view for all salesmen with columns salesman_id, name and city.
3. Write a query to find the salesmen of the city New York who achieved the commission more than 13%.
4. Write a query to create a view to getting a count of how many customers we have at each level of a grade.
5. Write a query to create a view to keeping track the number of customers ordering, number of salesmen attached, average amount of orders and the total amount of orders in a day.
6. Write a query to create a view that shows for each order the salesman and customer by name.
7. Write a query to create a view that finds the salesman who has the customer with the highest order of a day.
8. Write a query to create a view that shows all of the customers who have the highest grade.
9. Write a query to create a view that shows the number of the salesman in each city.
10. Write a query to create a view that shows the average and total orders for each salesman after his or her name. (Assume all names are unique)

11. Write a query to create a view that shows each salesman with more than one customers.
12. Write a query to create a view that shows all matches of customers with salesman such that at least one customer in the city of customer served by a salesman in the city of the salesman.
13. Write a query to create a view that shows the number of orders in each day.
14. Write a query to create a view that finds the salesmen who issued orders on October 10th, 2012.
15. Write a query to create a view that finds the salesmen who issued orders on either August 17th, 2012 or October 10th, 2012.

Solve query:

```
---1
create view salesman_newyork as
select * from salesman where
city='new york';
select * from salesman_newyork
---2
create view salesman_details as
select salesman_id,name,city from salesman;
select * from salesman_details
--3
select * from salesman where city='New york' and
commission >.13
---4
create view customer_count as
select grade ,count(*) as total_cust
from customer group by grade;
select * from customer_count
---5
create view numberofcustomer as
select ord_date ,count(customer_id)as total_customer,
count(salesman_id) as total_salesman,
avg(purch_amt) as average_amt,
sum(purch_amt) as total_amt
from orders group by ord_date;
select * from numberofcustomer
---6
create view Names as
select salesman.name,customer.cust_name from
salesman join customer on
salesman.salesman_id = customer.salesman_id
join orders on salesman.salesman_id=orders.salesman_id;
select * from Names
---7
create view highest_orders as
select salesman.name,customer.cust_name from
salesman join customer on
salesman.salesman_id = customer.salesman_id
join orders on salesman.salesman_id=orders.salesman_id
where orders.ord_date in (select top 1 ord_date from orders
```

```sql
group by ord_date order by count(*) desc);
---8
create view highest_grade as
select * from customer where
grade=(select max(grade) from customer);
----9
create VIEW SalesmanCountByCity  AS
select City,COUNT(*) AS NumberOfSalesmen
FROM Salesman group by city;
-----10
create view SalesmanOrderSummary as
select s.name ,SUM(o.purch_amt) as total ,
AVG(o.purch_amt) as average
FROM salesman s JOIN orders o ON s.salesman_id = o.salesman_id
group by s.name;
select * from SalesmanOrderSummary
----11
create view Multiple_Customers as
select s.salesman_id,s.name AS Salesman_Name,s.city,
 count(c.customer_id) AS NumberOfCustomers FROM
salesman s JOIN customer c on s.salesman_id = c.salesman_id
group by  s.salesman_id, s.name, s.city
having count(c.customer_id) > 1;
select * from Multiple_Customers
---12
create view SalesmanMatchCity as
select c.customer_id,c.cust_name,c.city AS Customer_City,
 s.salesman_id,s.name AS Salesman_Name,s.city AS Salesman_City
from customer c JOIN salesman s on c.city = s.city;
select * from SalesmanMatchCity
---13
create view NumOfOders as
select ord_date,count(*) as numOrders from orders
group by ord_date;
---14
create view salesmanordersdate as
select salesman.name,salesman.city from salesman join orders
on salesman.salesman_id=orders.salesman_id where
orders.ord_date='2012-10-10';
select * from salesmanordersdate
---15
create view salesman_orders_date as
select salesman.name,salesman.city from salesman join orders
on salesman.salesman_id=orders.salesman_id where
orders.ord_date='2012-08-17' or orders.ord_date='2012-10-10';
select * from salesman_orders_date
```

Lab assignment : 05

Consider first table as **Account_Detail** table having Account_no as a primary key, second table as **Branch** table having Br_Id as Primary key and third table as **Zone** table where Zone_Id is the primary key of that table.

Account_no	Acc_holder_name	Amount	Branch_Id	Zone_Id
1992212	Mr. Nazmuzzaman	200000	B-101	Z-803
1992213	Mr. Jibon	170000	B-102	Z-803
1882212	Bushra	180000	B-103	Z-802
1882213	%Sajib	170000	B-104	Z-801

Br_Id	Branch_Name
B-101	Bonani
B-102	Romna
B-103	Shaheb bazar
B-104	Ullapara

Zone_Id	Name
Z-801	Sirajgonj
Z-802	Rajshahi
Z-803	Dhaka
Z-804	Chittagong

1. Create a simple stored procedure “SPdetails” to find Acc_holder_name, Amount, Branch_Name and Zone_Name.
2. Create a simple stored procedure “SPaverage” to find Branch _name and Amount of Branch where amount will be greater than particular amount (say 17000). Here branch_name and amount will be passed by parameter
3. Create a simple stored procedure “SPbalance” to find Amount of a particular zone. Here zone name will be passed by parameter and amount will be shown by using return value ().
4. Create a simple stored procedure “SPamount” to Find all account holders name with their branch name and zone name whose name has substring ‘Mr.’ and Amount Less than Maximum Amount
5. Create a simple stored procedure “SPdetailsInfo” to find number of customer of each Zone. Here number of customers need to be printed as output parameter and zone_name will be passed as parameter
6. Create procedure like “spEmployeeSalaryDetails1” which has four parameter. three parameter match the StartAmount, EndAmount value, Branch_Name Value and another parameter return this value, in this procedure find the number of Branch_Name whose Amount between 7000 to 30000 and also check the Branch_Name substring “Ba”. where StartAmount, EndAmount value, Branch_Name value pass by parameter.
7. Create a simple stored procedure “SPdetailsInfo” to find Zone_name, number of customer of a specific Zone.
8. Creating a simple stored procedure “SPdetailsInfo1” to find Zone_name, number of Branch of a specific Zone(Branch name pass by parameter).

Solve query:

```
----1
create procedure SPdetails
as begin
select Account_detail.Acc_holder_name ,
Account_detail.Amount,branch.Branch_Name,Zone.name
from Account_detail join Branch on
Account_detail.Branch_Id=Branch.Br_id join Zone on
Account_detail.Zone_Id = Zone.Zone_id
end;
```

```
EXEC SPdetails
----2
create procedure SPaverage
@branch_names varchar(20),
@amounts decimal(10,2)
as begin
select b.Branch_Name,a.Amount from
Account_detail a join Branch b on a.Branch_Id=b.Br_Id
where a.Amount>@amounts and b.Branch_Name=@branch_names
end;
exec SPaverage @branch_names='Shaheb bazar' ,@amounts=170000;
----3
create procedure SPbalance
@Zone_name varchar(20)
as begin
DECLARE @totalamount int;
select @totalamount =sum(a.Amount) from Account_detail a join
Zone z on a.Zone_Id=z.Zone_Id where z.Name=@Zone_name
return @totalamount ;
end;
declare @amount int;
exec @amount=SPbalance @Zone_name='dhaka';
print @amount
----4
create procedure SPamount
as begin
select a.Acc_holder_name,b.Branch_Name,z.name from
Account_detail a
join Branch b on a.Branch_Id=b.Br_Id join Zone z on
z.Zone_Id=a.Zone_Id where a.Acc_holder_name like '%Mr.%' and
a.Amount < (select max(Amount) from Account_detail);
end ;
exec SPamount
-----5
create procedure SPdetailsInfo
@zone_name varchar(20),
@customer_count int output
as begin select @customer_count=count(*) from Account_detail a
join Zone z on a.Zone_id=z.Zone_Id where z.Name=@zone_name
end;
declare @count int;
exec  SPdetailsInfo 'dhaka',@customer_count=@count output
print 'no. of customer : ' +cast(@count as varchar)
```

```
---6
create procedure spEmployeeSalaryDetails
@startamounts float,
@endamount float,
@branch_name varchar(20),
@countamount decimal(10,2) output
as begin
select @countamount=count(distinct b.Branch_Name)
from Account_detail a join Branch b on a.Branch_Id=b.Br_Id
where a.Amount between @startamounts and @endamount
 AND b.Branch_Name LIKE '%' + @branch_name + '%'
and b.Branch_Name like '%Ba%'
end;
declare @findamount int;
exec spEmployeeSalaryDetails
@startamounts=7000,@endamount=30000,
@branch_name='Bamna',@countamount=@findamount output
print @findamount
----7
create procedure SPdetailsInfo1
@zonename varchar(20)
as begin
select Zone.Name,count(Account_detail.Account_no) as customer
from Account_detail join Zone on Account_detail.Zone_Id=
Zone.Zone_Id where Zone.Name=@zonename
group by Zone.Name
end;
exec SPdetailsInfo1 'dhaka'
---8
create procedure SPdetailsInfo2
@branch_name varchar(20)
as begin
select z.Name, count(b.Br_Id) from Account_detail a join Zone z
on a.Zone_id=z.Zone_Id join Branch b on b.Br_Id=a.Branch_Id
where b.Branch_Name=@branch_name
group by z.Name
end;
exec SPdetailsInfo2 'ullapara'
```

Lab assignment : 06

Tbl_Management

Mgt_id	Mgt_Name	Joining_date	Salary	Position
M2015	Keshob	2001-01-18	250000	Managing Director
M2016	Rana	2003-01-30	180000	Secretary
M2017	Jasim	2004-04-12	150000	Join secretary
M2018	Rajon	2004-06-18	140000	Join secretary

Tbl_Emp

Emp_id	Emp_Name	Joining_Date	Salary	Division
E1001	Suman	2003-04-25	92000	Software
E1002	Rasel	2004-03-13	86000	Network
E1003	Hossain	2004-06-21	82000	Software
E1004	polash	2005-05-05	9800	Network

Tbl_Project

P_id	P_Name	Mgt_id	E_id	P_Cost	Delivery_date
P3001	Office Automation	M2016	E1001	2050000	2016-05-08
P3002	Repair Hub	M2016	E1004	1200000	2017-06-14
P3003	Server Installation	M2018	E1001	1500500	2018-02-13
P3004	Network setup	M2017	E1002	2505000	2018-03-12

1. Write a sql query to show Project name, cost and Rank according to cost, assign employee name and rearrange the project according to cost ascending order.
2. Write a sql create UDF query to show Project name, cost and assign employee name and rearrange the project according to cost ascending order. Where Project name and employee name pass by parameter.
3. Write a sql query to find the rank of management Team according to their joining Date.
4. Write a sql create scalar function that has one parameter. In this function calculate the Salary of employee whose salary is maximum and that salary increase 10%. Where salary column pass by parameter
5. Write a sql UDF to show the Name of maximum Cost Project.
6. Write a sql Inline Table Valued function to show the Project name and Cost where cost in between 1200000 and 2050000. Costs are passed by parameter.
7. Create Inline Function like “fnEmployee”, in this function find the Mgt_id, Mgt_Name, Emp_Name, Joining_Date, Salary, P_Name, P_Cost, Delivery_date. Where P_id, Mgt_id, Emp_id pass by parameter.

Solve query:

```
----1
create function fndetails()
returns table
as return(
select p.p_Name,p.p_cost, dense_rank() over (order by p.p_cost
asc) as rank,
e.Emp_name from Tbl_Project p join Tbl_Emp e on
p.E_id=e.Emp_id );
select * from dbo.fndetails() order by p_cost

----2
create function UDF(
@project_name varchar (30),
@employee_name varchar (30)
)
returns @result table(
```

```
p_Name varchar(20),
p_cost int,
Emp_name varchar(20)
)
as begin
insert into @result
select p.p_Name,p.p_cost, e.Emp_name from Tbl_Project p
join Tbl_Emp e on p.E_id=e.Emp_id where p.P_Name=@project_name
and
e.Emp_Name=@employee_name order by p_cost asc ;
return
end;
select * from dbo.UDF('Network setup','Rasel')
drop function UDF

---3
create function fnrankofmanagement()
returns table as
return(select position ,DENSE_RANK() over(order by joining_date
asc)
as rank from Tbl_Management);
select * from dbo.fnrankofmanagement()

---4
create function fnHighSalary(@salarys decimal(10,2))
returns decimal(10,2)
as begin
declare @maxSalary decimal(10,0)
declare @result float
select @maxSalary= max(salary) from Tbl_management
set @result=@maxSalary*1.10
return @result
end;
select dbo.fnHighSalary('salary')

---5
create function fnMaximum()
returns varchar(20)
as begin
declare @projecName varchar(20)
select top 1 @projecName=P_Name from Tbl_Project
order by P_Cost desc
return @projecName
end;
select dbo.fnMaximum()
```

```
---6
create function fnInline(
@cost decimal(10,2),
@cost1 decimal(10,2)
)
returns table
as
return(select  P_Name,P_cost from Tbl_Project where
P_Cost between @cost and @cost1);
select * from fnInline (1200000,2050000)

---7
create function fnEmployee(@P_id varchar(10), @Mgt_id
varchar(10),
@Emp_id varchar(10))
returns table
as return
(select m.Mgt_id, m.Mgt_Name, e.Emp_Name, m.Joining_Date,
m.Salary, p.p_name, p.p_cost, p.delivery_date
from Tbl_Management m join
Tbl_Project p on m.Mgt_id=p.Mgt_id join Tbl_Emp e on
e.Emp_id=p.E_id where @P_id=p.P_id and @Mgt_id=m.Mgt_id and
@Emp_id= e.Emp_id);
select * from fnEmployee('P3001','M2016','E1001')
```