

Project Title	Project Manager	Budget	Employee	Department	Hourly Pay rate
HR	E1	9800000.00	P	IT	600.00
			Q	Database	570.00
Payroll	E2	8700000.00	W	Database	570.00
			X	IT	500.00
			Y	Business Process	730.00

1. 3 NF Normalization
2. Create a VIEW
3. Create Index (Clustered)
4. Create Procedure With Try Catch
5. Create Function (Both Scaler and In-line Tabled)
6. Create Trigger for Restricted data inserting in a Table.

```
Create Database lhmullah
GO
```

```
Use lhmullah
GO
```

```
Create table departments
(
deptid int not null primary key,
deptname varchar(30) not null
)
GO
Create table projects
(
projid int not null primary key,
projecttitle varchar (40) not null,
projectmanager varchar (30) not null,
budget Money not null
)
GO
```

```
Create Table employees
(
empid INT not null primary key,
empname varchar (30) not null,
hourlypayrate money not null,
deptid int not null references departments (deptid),
projid INT not null references projects (projid)
)
GO
```

```
Insert Into departments
values
(1, 'IT'),
(2, 'Database'),
(3, 'Business Process')
GO
```

```
Insert Into projects
values
(1, 'HR', 'E1', 9800000.00),
(2, 'Payroll', 'E2', 8700000.00)
GO
```

```

Insert Into employees
values
(1, 'P', 600.00, 1, 1),
(2, 'Q', 570.00, 2, 1),
(3, 'W', 570.00, 2, 2),
(4, 'X', 500.00, 1, 2),
(5, 'Y', 730.00, 3, 2)
GO

Create index ixProject ON employees (empid)
GO

--Justify--
EXEC sp_helpindex employees
GO

Create view vEmployee
as
select d.deptname, e.empname, p.projecttitle, p.budget, p.projectmanager
from
departments d
inner join employees e
on d.deptid = e.deptid
inner join projects p
on p.projid = e.projid
GO
-- Justify--
Select * from vEmployee
GO

Create proc spProjectInsert

                                @pid int,
                                @pt varchar (40),
                                @pm varchar (30),
                                @b Money

AS
Begin TRY
    insert Into projects(projid, projecttitle, projectmanager, budget)
    Values(@pid, @pt, @pm, @b)
    Return 0
END TRY
BEGIN Catch
    Raiserror ('Invalid Data', 10, 1)
    Return error_number()
END Catch
GO

-- Justify--
EXEC spProjectInsert 3, 'Accounts', 2, 500000.00
GO

-- Justify--
SELECT * FROM projects
GO

Create Function fnFindPayrate (@empname varchar (30))
Returns money
AS
Begin
Declare @hourlypayrate money
    Select @hourlypayrate = hourlypayrate from employees
    where @empname = empname
    Return @hourlypayrate
END

```

```

GO
-- Justify--
SELECT dbo.fnFindPayrate('P')
GO

Create Function fnFindTable (@projecttitle varchar (40))
Returns TABLE
AS
RETURN
select d.deptname, e.empname, p.projecttitle, p.budget, p.projectmanager
from
departments d
inner join employees e
on d.deptid = e.deptid
inner join projects p
on p.projid = e.projid
WHERE p.projecttitle = @projecttitle
GO

-- Justify--
SELECT * FROM fnFindTable ('HR')
GO

Create Trigger trRestrict
ON Employees
Instead of insert
As
BEGIN
        Declare @hourlypayrate money
        Select @hourlypayrate = hourlypayrate from inserted
        If @hourlypayrate <=0
        Begin
                Raiserror ('Payrate less than 0', 10, 1)
                ROLLBACK
        END
        Else
        Begin
                Insert Into employees
                Select * from inserted
        END
END
GO

-- Justify--
Insert Into employees
Values
(6, 'Z', 0.00, 1, 1)
GO

SELECT * FROM employees
GO

Create Trigger trRestrictDelete
ON Employees
For Delete
As
        If @@ROWCOUNT > 0
        Begin
                Raiserror ('Your not permitted to delete', 11, 1)
                Rollback Transaction
        END
GO

```