Drug Sheet:

I. SQL storage procedure

a. Query by dosage

```
create proc  dosage
@jx nvarchar(10)
as
select * from  drug.sheet
where dosage=@jx
```

b.Insert records:

```
create proc yp_proc
@ypid int,@ypname nvarchar(50),@ypjx nvarchar(10),
@ypgg nvarchar(12),@sccj nvarchar(50),@ypprice money,
@ypbz nvarchar(10),@bzcl nvarchar (10),@byyy int
as
insert into  drug.sheet values (@ypid,@ypname,@ypjx,
@ypgg,@sccj,@ypprice,@ypbz,@bzcl,@byyy)
```

c. Query by drug name:

```
create proc name:
@ypname nvarchar(10)
as
select * from drug.sheet
where name=@ypname
```

II. Function

a. Scalar Value function

—Query amounts of drugs by dosage:

```
create function yp_num
(@yp_jx nvarchar(10))
returns int
begin
    declare @n int
    select @n=count(*) from drug.sheet
    where dosage=@yp_jx
    return @n
end
```

b. Inline Table Value Function

—Query amounts of used drugs for month by Drug ID

```
create function by_num
(@fid int)
returns table
as
```

```
return
(select drug ID,name,used.drug from  drug.sheet
where  drug ID=@fid)
```

III. Fetch
   -- Define a read-only fetch which contains drug ID, name, manufacturer and price

```
declare yp_cursor cursor
for select drug ID,name,manufacturer, price from drug.sheet
for read only
open yp_cursor
while @@fetch_status=0
begin
declare @a int,@b nvarchar(50),@c nvarchar(50),@d money
fetch next from yp_cursor into @a,@b,@c,@d
print convert(char,@a)+@b+@c+convert(char,@d)
end
close yp_cursor
deallocate yp_cursor
go
```

IV. Index
  --Create unique index by drug ID in drug.sheet:

```
create unique index ypid
on drug.sheet(drugID asc)
```

V. View
 a. View of drug:

```
create view drug
as
select *
from drug.sheet
```

 b. View of drug which dosage is injection:

```
create view injection
as
select *
from drug.sheet
where dosage='injection'
```

Doctor Sheet:

I.Storage Procedure:

a. Query by doctors' name:

```
create proc p1
@dname nvarchar(50)
as
select name,title,department,office hour
from doctors
where  name=@dname
```

b. Query by male doctors

```
create proc p2
as
select name,sex,department,office hour
from doctors
where sex='male'

exec p2
```

c. Query by title

```
create proc p3
@position nvarchar(50)
as
select name,sex,department,attendind disease, office hour
from doctors
where title=@position
```

II View:
Calculate number of people for each department

```
create view pnum
as select department ,count(department) as number
from doctors
group by department
```

III. Self-defined Function:

```
create function doctors_time
(@dtime nvarchar(50))
returns table
as
return (select name,title,attanding disease, office hour,[patients number/
day]
```

```
        from doctors
        where office like '%@time%' )
```

IV. Index:

```
Name index:
create clustered index doc_sy
on doctors(name)
```

V: Fetch:

```
declare doctors_coursor cursor
for select name,title, attending disease,[ patient number/day] from
doctors
for read only
open doctors_coursor
whilew @@fetch_status=0
begin
declare @n char(10),@m char(10),@p char(10),@g char(10)
fetch next
from doctors_coursor into @n,@m,@p,@g
print @n+@m+@p+@g
end
close doctors_coursor
```

Drug.storehouse.sheet

I. View:

a. Create information of drugs which were left over 5 last month

```
create view shangyue as
select * from dbo.drug.sheet join  dbo.Table_drug.storehouse.sheet on
dbo.drug.sheet.drug ID=dbo.Table_drug.storehouse.sheet.ID
where dbo.Table_drug.storehouse.sheet.last_month_left>5
```

b. Create information of drugs which have been used over 15 this month

```
create view yiyong as
select * from   dbo.Table_ drug.storehouse.sheet
where dbo.Table_drug.storehouse.sheet .new_drug>20
```

II. Trigger:
  --- When delete an info of a drug in drug.storehouse.sheet, it's info will be automatically deleted in drug.sheet

```
create trigger adjust
on dbo.Table_drug.storehouse.sheet
```

```
for delete
as
declare @bianhao int
select @bianhao=ID      from deleted
delete dbo.drug.sheet  where drug  ID=@bianhao

delete dbo.Table_drug.storehouse.sheet  where ID='4'
```

III. Storage Procedure:
  --- Input a drug's name and return related info of this drug and its storage into in drug storaghouse

```
create proc chakan
  @yname varchar(8)
 as
select dbo.Table_drug.storehouse.shee .*,dbo.drug.sheet.*
 from  dbo.drug.sheetjoin dbo.Table_drug.storehouse.sheet
  on dbo.drug.sheet.ID = dbo.Table_drug.storehouse.sheet.ID
where dbo.drug.sheet.name=@yname


exec chakan @yname='板蓝根颗粒'
```

IV. Index:
--- View an increasing order of number of new purchased drugs this month

```
create index order
on dbo.Table_drug.storehouse.sheet(new_drug asc)
```

V. Function:
--- The number of drugs left for all existing drugs this month:

```
create function keyong
 ( @yname varchar(8),@num int )
returns int
as
begin
select @num=dbo.Table_drug.storehouse.sheet.lat_month_left
+dbo.Table_drug.storehouse.sheet.new_drug-dbo.drug.sheet.used_drug
from dbo.Table_drug.storehouse.sheet join dbo.drug.sheet
 on dbo.Table_drug.storehouse.sheet.ID=dbo.drug.sheet.ID
where dbo.drug.sheet.name=@yname
return @num
end

select name,dbo.D(name,use_drug) from dbo.drug.sheet
```

VI. Fetch:
  --- Query information of drug storehouse

```
declare yaoku cursor
for select ID,new_drugfrom dbo.Table_drug.storehouse.sheet
for read only
open yaoku
while @@fetch_status=0
begin
declare @n char(10),@m int
fetch next
from doctors_coursor into @n,@m
print @n+@m
end
close doctors_coursor
```

Doctor Prescribing Sheet:

I. View:

```
create view v1 as
select doctor_ID, name, dosage,price,amounts,prescribe_time from
dbo.drug.sheet join  dbo.doctor.prescribing.sheet on
dbo.drug.sheet.ID=dbo.doctor.prescribing.sheet.ID
where dbo.doctor.prescribing.sheet.amounts>5
```

II. Trigger:

```
create trigger t1
on  doctor.prescribing.sheet
after insert
as
declare @bianhao nvarchar(50),@ypid int,
@num int
select @bianhao=doctor_ID,@ypid=ID,
@num=amounts from inserted
update dbo.drug.sheet  set used_drug=used_drug+@num
```