Inherited Sharing in SFDC

Name	Owner
ret re	cloud salesforce
test trs	cloud salesforce
test test	cloud salesforce
vipin Indora	cloud salesforce
as as	cloud salesforce
vipiiio odsfj	cloud salesforce
Standard 2 Gurgaon	OhanaVipin indora
Standard 1 India	OhanaVipin indora
con1	cloud salesforce
con6	cloud salesforce
con7	cloud salesforce
con8	cloud salesforce
con5	cloud salesforce
con2	cloud salesforce
con3	cloud salesforce
con4	cloud salesforce

Apex without a sharing declaration is insecure by default. Designing Apex classes that can run in either with sharing or without sharing mode at runtime is an advanced technique.

There is a distinct difference between an Apex class that is marked with inherited sharing and one with an omitted sharing declaration. If the class is used as the entry point to an Apex transaction, an omitted sharing declaration runs as without sharing. However, inherited sharing ensures that the default is to run as with sharing. A class declared as inherited sharing runs as without sharing only when explicitly called from an already established without sharingcontext.

- When parent class is with sharing then inherited also behave like with sharing and sharing rule will apply.
- if parent class is without sharing then inherited also behave like without sharing and sharing rule will not apply and can see user has no right to view the records can also view due to the insecure default behavior of omitting
  the declaration.

Lets take an example: I am showing contacts on visualforce page so by default for contact sharing setting(OWD) is **Controlled by Parent** for below example i have changed and set as **private** that means current logged in user can see there contact only.

In below apex class Main class has declared as Without sharing and in this class inner class i have used ad inherited sharing so for this sharing setting is private but current logged in user can see other user record's also on visualforce page

```
public without sharing class ParentClass{
public List<Contact> Childlst{get;set;}

public inherited sharing class InheritedSharingClass {
public List<Contact> getAllTheSecrets(){
return [SELECT name,owner.name FROM Contact LIMIT 100];
}
}
public ParentClass(){
ParentClass.InheritedSharingClass childcls = new ParentClass.InheritedSharingClass();
Childlst = childcls.getAllTheSecrets();

}
```

```
<apex:page controller="ParentClass">
<apex:form >
<apex:pageBlock >
<apex:pageBlockTable value="{!Childlst}" var="record">
<apex:column headerValue="Name" value="{!record.Name}"/>
<apex:column headerValue="Owner" value="{!record.owner.Name}"/>
</apex:pageBlockTable>
</apex:pageBlock>
</apex:form>
</apex:page>
```

For above code output will be like this

Owner Cloud Salesforce profile is System admin and owner OhonaVipin Indora profile is Standard user and currently i am logged in from standard user profile. i can see system admin record as well.

Name	Owner
ret re	cloud salesforce
test trs	cloud salesforce
test test	cloud salesforce
vipin Indora	cloud salesforce
as as	cloud salesforce
vipiiio odsfj	cloud salesforce
Standard 2 Gurgaon	OhanaVipin indora
Standard 1 India	OhanaVipin indora
con1	cloud salesforce
con6	cloud salesforce
con7	cloud salesforce
con8	cloud salesforce
con5	cloud salesforce
con2	cloud salesforce
con3	cloud salesforce
con4	cloud salesforce

Example 2 : just only replaced out from without sharing and everything in class and vfpage is same but output will be different.

In below apex class Main class has declared as With sharing and in this class inner class i have used ad inherited sharing so for this sharing setting is private but current logged in user **cannot** see other user record's on visualforce page only he/she can see own contact(s) only.

```
public with sharing class ParentClass{
public List<Contact> Childlst{get;set;}

public inherited sharing class InheritedSharingClass {
public List<Contact> getAllTheSecrets(){
return [SELECT name,owner.name FROM Contact LIMIT 100];
```

```
}
}
public ParentClass(){
ParentClass.InheritedSharingClass childcls = new ParentClass.InheritedSharingClass();
Childlst = childcls.getAllTheSecrets();

}

<apex:page controller="ParentClass">
<apex:form >
<apex:pageBlock >
<apex:pageBlockTable value="{!Childlst}" var="record">
<apex:column headerValue="Name" value="{!record.Name}"/>
<apex:column headerValue="Owner" value="{!record.owner.Name}"/>
</apex:pageBlockTable>
</apex:pageBlock>
</apex:form>
```

For above code output will be like this

Owner Cloud Salesforce profile is System admin and owner OhonaVipin Indora profile is Standard user and currently i am logged in from standard user profile. i cannot see system admin record.

Name	Owner
Standard 2 Gurgaon	OhanaVipin indora
Standard 1 India	OhanaVipin indora

Note: If we are using Inherited class only that it will behave like with sharing that means sharing rule will apply and cannot see other users contact on visualforce page.

```
public inherited sharing class InheritedSharingClass {
public List<Contact> getAllTheSecrets(){
return [SELECT name,owner.name FROM Contact LIMIT 100];
}
}
```