



**Gone are those days when you had to check each field's accessibility in Apex using Schema functions. Now with Spring19 release, we can enforce the field level security **within SOQL itself**. Sounds amazing, isn't it?**

## Use Case

For SFDCFacts Inc., security comes first. They have a quiet complex security architecture that shouldn't be compromised in any case. However while creating programmatic solutions for their business cases, it becomes very challenging to manage security in Apex. They can easily take care of record sharing by using “**with sharing**” keyword on classes, however, for field level security, it becomes quite challenging for a developer to check accessibility on each field.

## How SFDCFacts used to do it before?

For each and every field being referred in their apex class, they used to check accessibility using schema methods. For example, to check user's access on the Amount, Stage and, LeadSource fields of opportunity, this is what they used to do in Apex before:

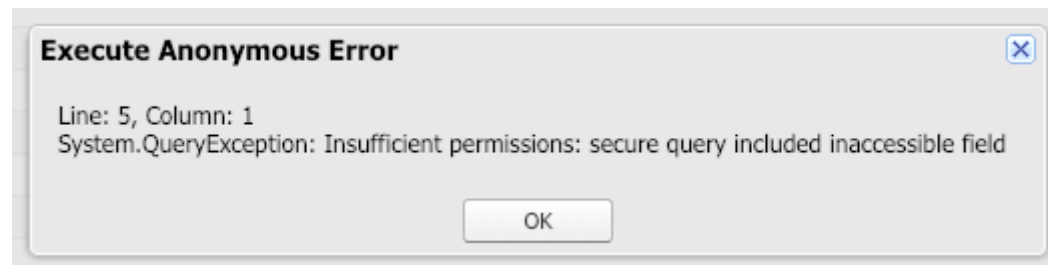
```
String query = 'Select Id, Name, ';\nif(Schema.sObjectType.Opportunity.fields.Amount.isAccessible()){ \nquery += 'Amount, ';\n} else{\n//Show exception\n}\nif(Schema.sObjectType.Opportunity.fields.StageName.isAccessible()){ \nquery += 'StageName, ';\n}\nif(Schema.sObjectType.Opportunity.fields. LeadSource.isAccessible()){ \nquery += 'LeadSource, ';\n}\nquery = query.subString(0, query.length()-2);\nquery += ' FROM Opportunity';\nList<Opportunity> opps = Database.query(query);
```

## How SFDCFacts can do it now?

Now since Salesforce has introduced this amazing feature, SFDCFacts's developer may get some relief as now they just need to write below lines of code:

```
//This query will throw an exception if current user does not have access to Amount, Stage or LeadSource\nList<Opportunity> opps = [SELECT Id, Name, StageName, Amount, LeadSource FROM Opportunity WITH SECURITY_ENFORCED];
```

The user will get below error on their screen:



*Insufficient Permissions*

**NOTE:** SOQL only checks for user's accessibility and **CANNOT** check other permissions like Create, Update or Delete similar to isCreateable(), isUpdateable() and, isDeletable() method of Schema class.

Please keep in mind that this feature is still in the pilot phase and not generally available yet. Check out more details here: [Salesforce Release Notes](#)