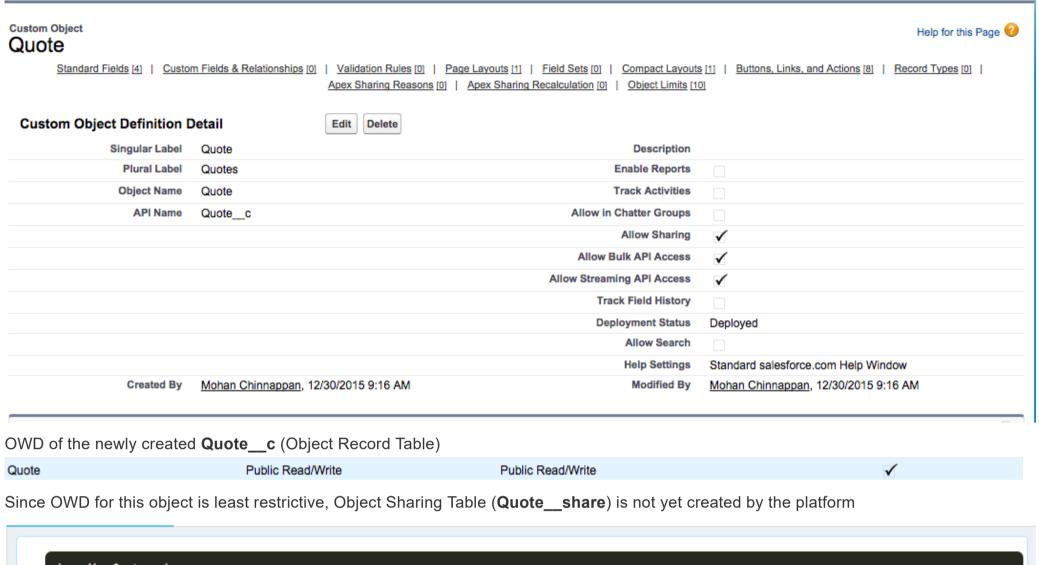
Custom Object Creation and Object Sharing Table

When we create a custom object say Quote_c, the platform will default it's OWD to Public Read/Write.

UI:

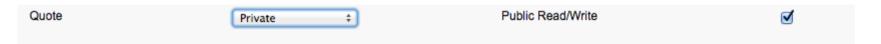
View the custom objects: https://na34.salesforce.com/p/setup/custent/CustomObjectsPage

Create a custom object: https://na34.salesforce.com/01l/e?setupid=CustomObjects





Let us make it less restrictive by changing OWD for this object to **Private**:

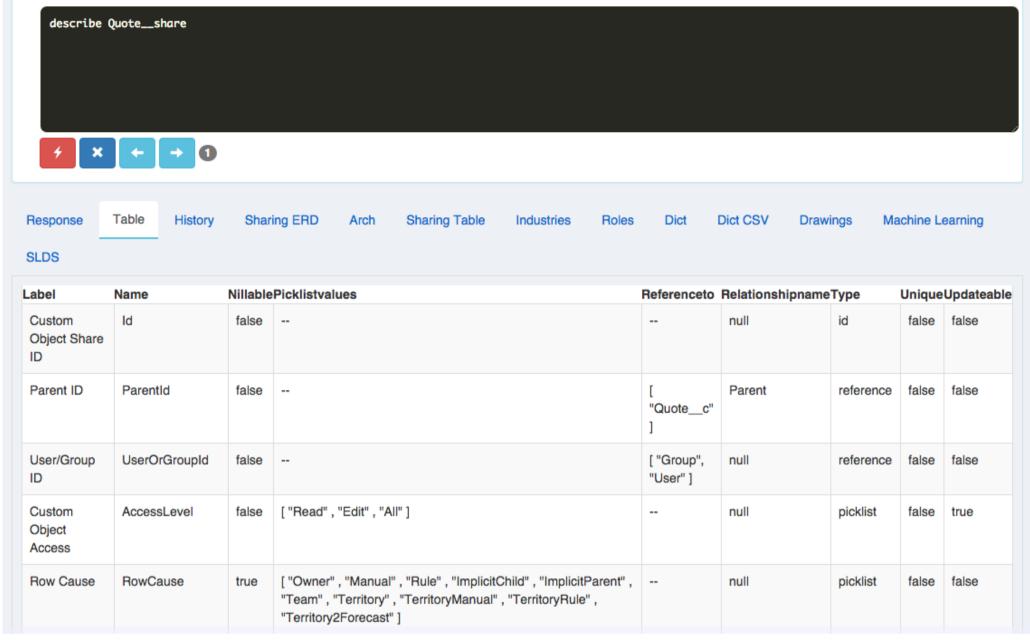


This will make the platform to setup Object Sharing Table (**Quote__share**) along with other things. This is a background task. The Admin user who made this change will be notified when this background task is completed:

This page displays your organization's sharing settings. These settings specify the level of access your users have to each others' data.

An organization-wide default update has been initiated by Mohan Chinnappan on 12/30/2015 9:25 AM. You can't submit any changes prior to the completion of the operation. Mohan Chinnappan will receive an email when the operation finishes.

Let check the availability of the share table: Quote__share

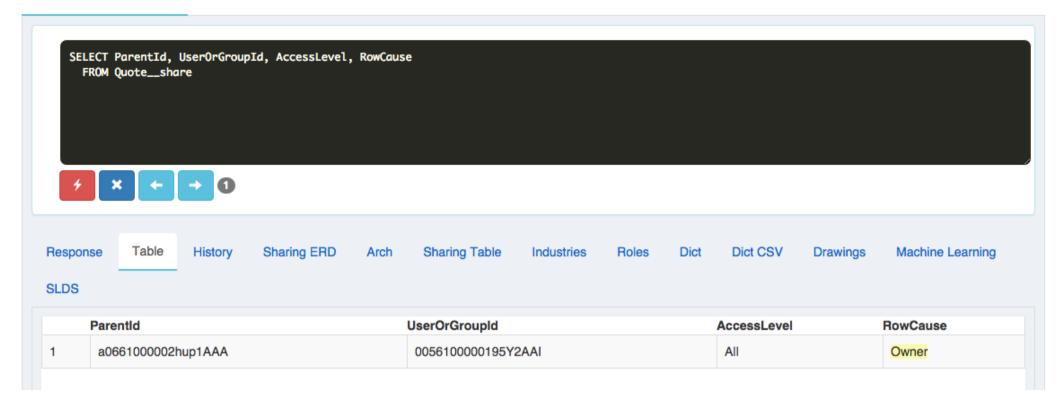


Let us query **Quote__share**: No records yet!:

```
SELECT ParentId, UserOrGroupId, AccessLevel, RowCause FROM Quote_share
                                                                                                   Dict
Response
                                  Sharing ERD
                                                                                                           Dict CSV
                                                                                                                                     Machine Learning
             Table
                       History
                                                  Arch
                                                           Sharing Table
                                                                            Industries
                                                                                         Roles
                                                                                                                        Drawings
SLDS
          "totalSize": 0,
          "done": true,
          "records":
     3
```

Let us **create** a record in Object Record Table: **Quote__c**:

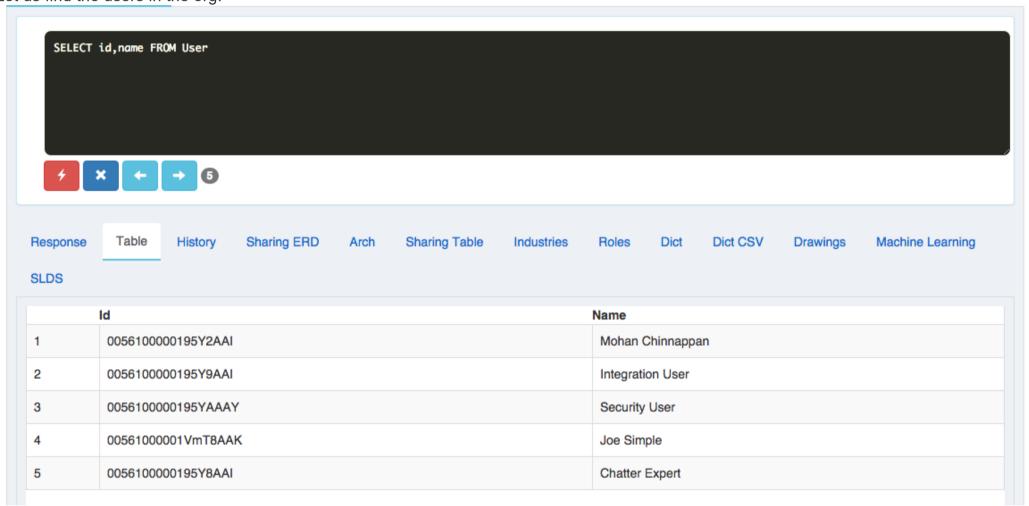
```
$ cat newQuote.json
{
    "name": "New Quote-1"
}
$ curl https://na34.salesforce.com/services/data/v33.0/sobjects/Quote__c/
    -H "Content-Type: application/json"
    -d @newQuote.json
    -H 'Authorization: Bearer 00D61000000dcQB!AQ4AQPqcU.5ect3zyKjaSlf3ebSi.WTIWW4Np4W.B8SKgADRFSm9Aq.OmRf8MikJQukNrNtzVZ9iiM93m
$ {"id":"a0661000002hup1AAA", "success":true, "errors":[]}
```



You can see a tuple (record) with RowCase: **Owner** and AccessLevel: **All** This is the work of the platform's managed sharing: providing full access to the record owner.

Now we can create a Apex class to provide Read access to non-owner of this record.

Let us find the users in the org:



Here we will provide Read access to the user Joe Simple (id:00561000001VmT8AAK aka UserOrGroupId in Record Sharing table) for this record (id:a0661000002hup1AAA aka ParentId in Record Sharing table)

Create Apex class: QuoteSharing with method:

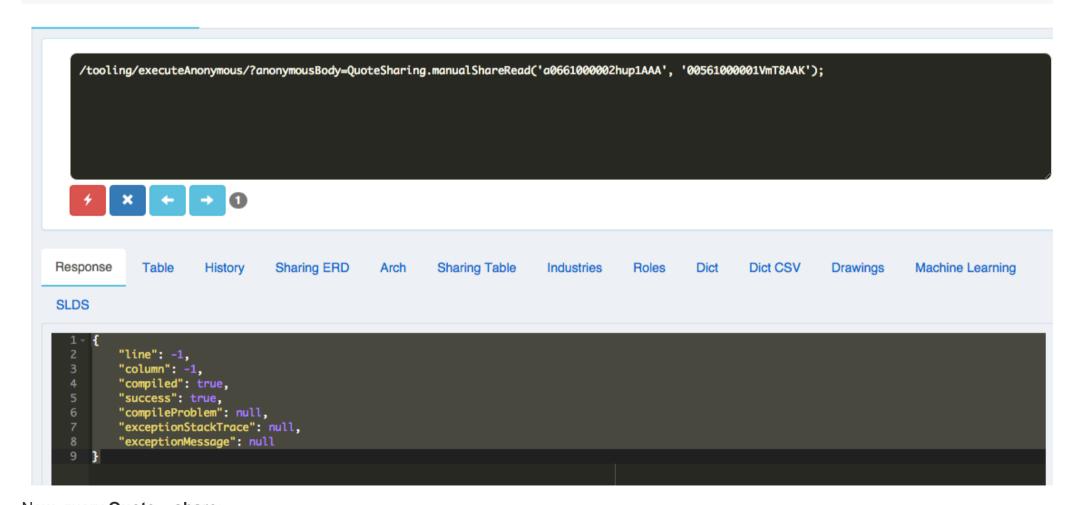
manualShareRead(ParentId, userOrGroupId)

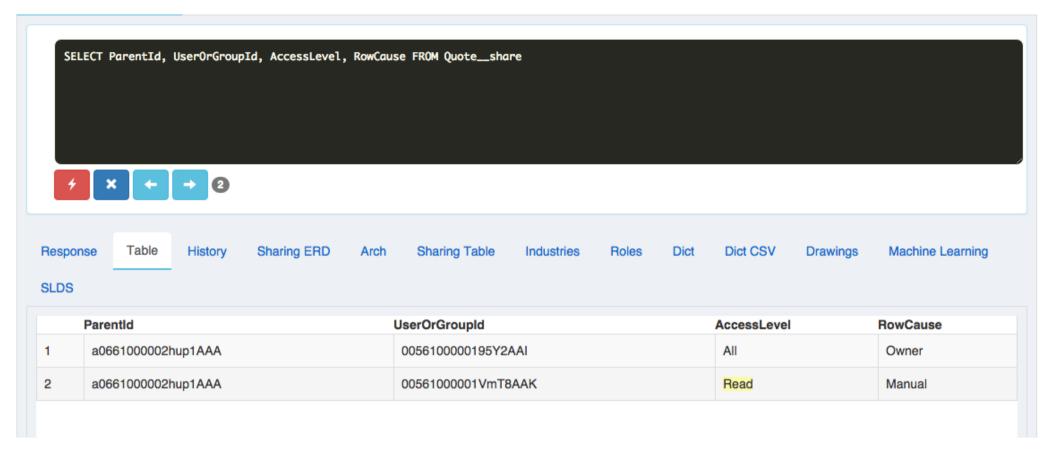
UI: https://na34.salesforce.com/01p

```
public class QuoteSharing {
   public static boolean manualShareRead(Id recordId, Id userOrGroupId){
      // Create new sharing object for the custom object Quote.
      Quote__Share quoteShr = new Quote__Share();
      // Set the ID of record being shared.
      quoteShr.ParentId = recordId;
      // Set the ID of user or group being granted access.
      quoteShr.UserOrGroupId = userOrGroupId;
      // Set the access level.
      quoteShr.AccessLevel = 'Read';
      // Set rowCause to 'manual' for manual sharing.
      // This line can be omitted as 'manual' is the default value for sharing objects.
      quoteShr.RowCause = Schema.Job__Share.RowCause.Manual;
      // Insert the sharing record and capture the save result.
      // The false parameter allows for partial processing if multiple records passed
      // into the operation.
      Database.SaveResult sr = Database.insert(quoteShr,false);
      // Process the save results.
      if(sr.isSuccess()){
         // Indicates success
         return true;
      else {
         // Get first save result error.
         Database.Error err = sr.getErrors()[0];
         // Check if the error is related to trival access level.
         // Access levels equal or more permissive than the object's default
         // access level are not allowed.
         // These sharing records are not required and thus an insert exception is acceptable.
         if(err.getStatusCode() == StatusCode.FIELD_FILTER_VALIDATION_EXCEPTION &&
                  err.getMessage().contains('AccessLevel')){
            // Indicates success.
            return true;
         else{
            // Indicates failure.
            return false;
   }
}
```

Running using tooling API:

```
QuoteSharing.manualShareRead('a0661000002hup1AAA', '00561000001VmT8AAK');
```





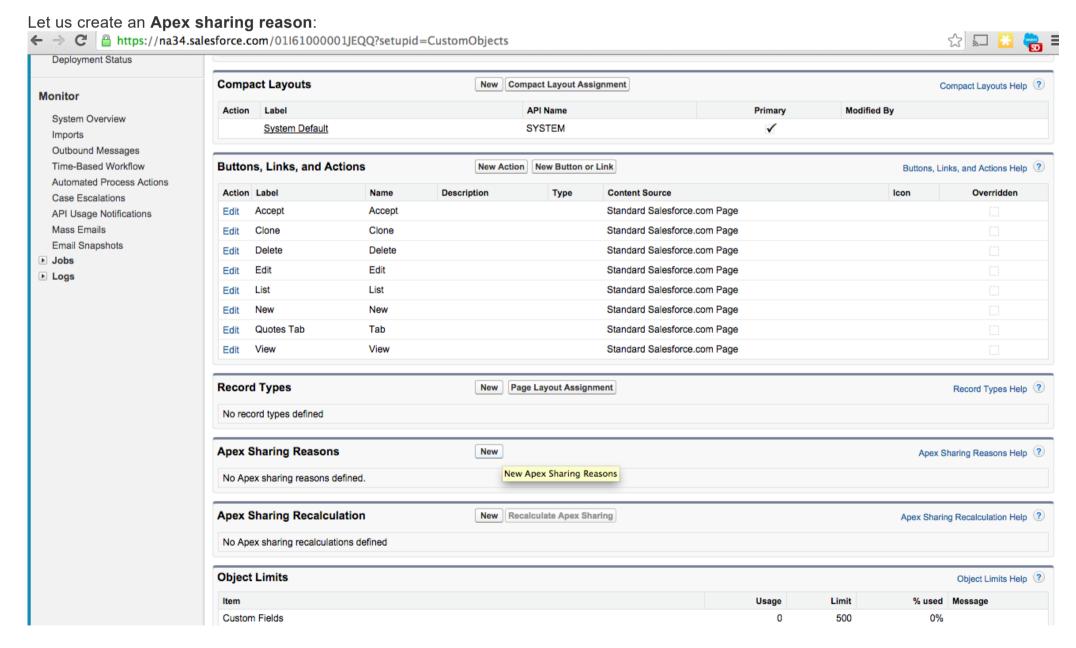
You can see new tuple (record) with RowCase: Manual and AccessLevel: Read

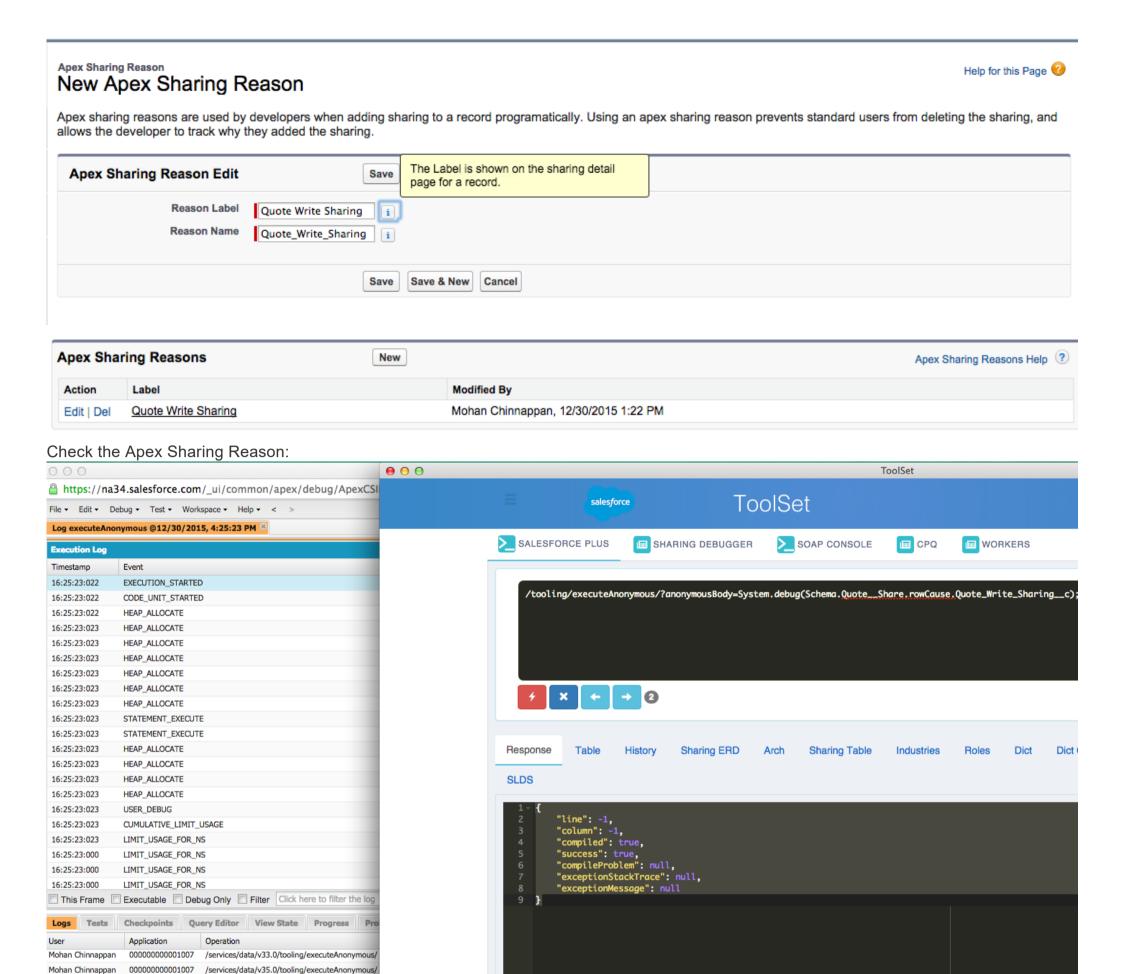
This is the result of our work: **User Managed Sharing** Using **Apex**.

Apex Managed Sharing:

Only users with "Modify All Data" permission can add or change Apex managed sharing on a record. Apex managed sharing is maintained across record owner changes.

Apex managed sharing uses **Apex sharing reason** (provide developers ability to track why they shared a record with a user or group of users)





Let us create an Apex class to do Apex Managed Sharing:

 Mohan Chinnappan
 00000000001007
 /services/data/v35.0/tooling/executeAnonymous/

 Mohan Chinnappan
 00000000001007
 /services/data/v35.0/tooling/executeAnonymous/

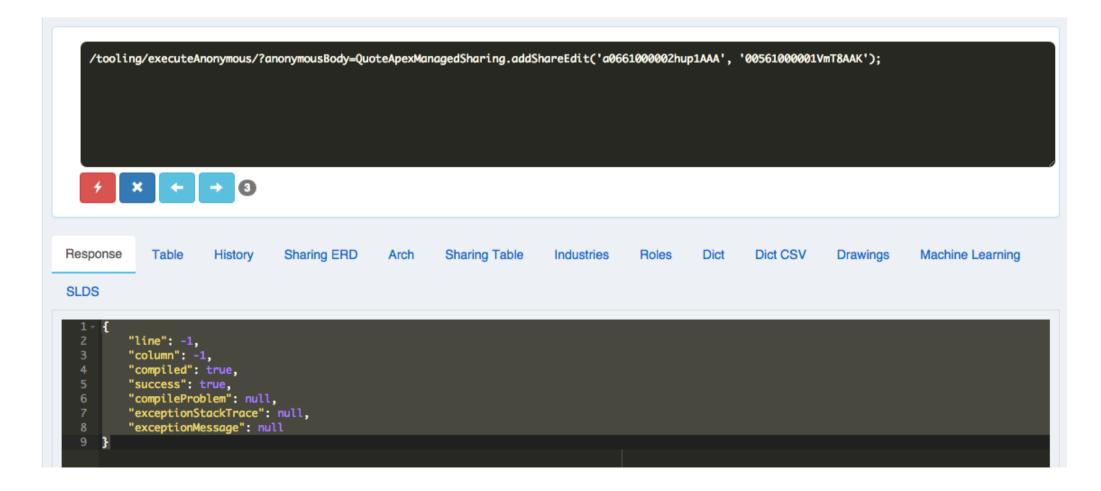
 Mohan Chinnappan
 00000000001007
 /services/data/v35.0/tooling/executeAnonymous/

 Mohan Chinnappan
 00000000001007
 /services/data/v35.0/tooling/executeAnonymous/

```
public class QuoteApexManagedSharing {
       public static boolean addShareEdit(Id recordId, Id userOrGroupId){
          // Create new sharing object for the custom object Quote.
          Quote__Share quoteShr = new Quote__Share();
          // Set the ID of record being shared.
          quoteShr.ParentId = recordId;
          // Set the ID of user or group being granted access.
          quoteShr.UserOrGroupId = userOrGroupId;
          // Set the access level.
          quoteShr.AccessLevel = 'edit';
          // Set rowCause to Schema.Quote__Share.rowCause.Quote_Write_Sharing__c
          quoteShr.RowCause = Schema.Quote__Share.rowCause.Quote_Write_Sharing__c;
          // Insert the sharing record and capture the save result.
          // The false parameter allows for partial processing if multiple records passed
          // into the operation.
          Database.SaveResult sr = Database.insert(quoteShr,false);
          // Process the save results.
          if(sr.isSuccess()){
             // Indicates success
             return true;
         else {
             // Get first save result error.
             Database.Error err = sr.getErrors()[0];
             // Check if the error is related to trival access level.
             // Access levels equal or more permissive than the object's default
             // access level are not allowed.
             // These sharing records are not required and thus an insert exception is acceptable.
             if(err.getStatusCode() == StatusCode.FIELD_FILTER_VALIDATION_EXCEPTION &&
                      err.getMessage().contains('AccessLevel')){
                // Indicates success.
                return true;
             else{
                // Indicates failure.
                return false;
```

Run using tooling API:

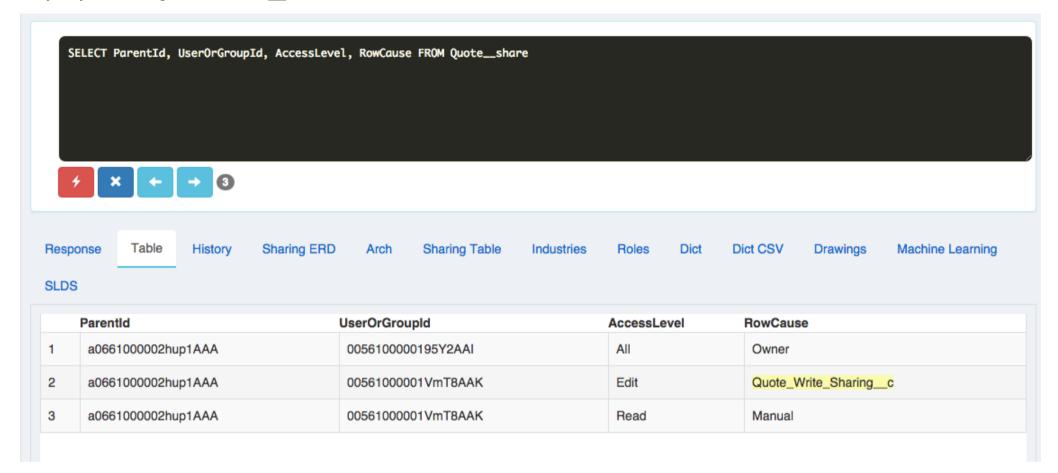
QuoteApexManagedSharing.addShareEdit('a0661000002hup1AAA', '00561000001VmT8AAK');



Execution Log		
Timestamp	Event	Details [23][ST[Datapase:Savekesuit]troe]raise
17:00:04:089	VARIABLE_ASSI	
17:00:04:089	STATEMENT_EX	[26]
17:00:04:090	STATEMENT_EX	[28]
17:00:04:090	HEAP_ALLOCATE	[28] Bytes:20
17:00:04:090	USER_DEBUG	[28] DEBUG sharing reason added
		FAAN:

View Log:

Query Object Sharing Table: **Quote__share**:



You can see new tuple (record) with RowCase: Quote_Write_Sharing__c and AccessLevel: Edit

This is the result of our work: Apex Managed Sharing .

As you here, Object-id: a0661000002hup1AAA for the UserOrGroupId: 00561000001VmT8AAK has both **Read** and **Edit**, Platform will provide more permissive access (**Edit**, in this case) to this user.

Notes

If a manual share access level is set to **Read** and you insert a new one that's set to **Write**, the original share rows are updated to **Write**, indicating the higher level of access

Testing this:

public class QuoteSharingGeneric {

```
public static boolean manualShare(Id recordId, Id userOrGroupId, String accessLevel){
      // Create new sharing object for the custom object Quote.
      Quote__Share quoteShr = new Quote__Share();
      // Set the ID of record being shared.
      quoteShr.ParentId = recordId;
      // Set the ID of user or group being granted access.
      quoteShr.UserOrGroupId = userOrGroupId;
      // Set the access level.
      quoteShr.AccessLevel = accessLevel;
      // Set rowCause to 'manual' for manual sharing.
      // This line can be omitted as 'manual' is the default value for sharing objects.
      quoteShr.RowCause = Schema.Job__Share.RowCause.Manual;
      // Insert the sharing record and capture the save result.
      // The false parameter allows for partial processing if multiple records passed
      // into the operation.
      Database.SaveResult sr = Database.insert(quoteShr,false);
      // Process the save results.
      if(sr.isSuccess()){
         // Indicates success
         return true;
      else {
         // Get first save result error.
         Database.Error err = sr.getErrors()[0];
         // Check if the error is related to trival access level.
         // Access levels equal or more permissive than the object's default
         // access level are not allowed.
         // These sharing records are not required and thus an insert exception is acceptable.
         if(err.getStatusCode() == StatusCode.FIELD_FILTER_VALIDATION_EXCEPTION &&
                  err.getMessage().contains('AccessLevel')){
            // Indicates success.
            return true;
         else{
            // Indicates failure.
            return false;
   }
}
```

Run: QuoteSharingGeneric.manualShare('a0661000002hup1AAA', '00561000001VmT8AAK', 'Edit');

