Solution 1: Query the assigned user contact

1. **Modify ABContact Summary class.**

   a) Add new **AssignedUserWorkload** property.

   b) Modify **loadSummaryData** method.

   c) Modify **buildConcatenatedSummary** method.

```
package si.ta.classes.entity

uses gw.api.database.Query
uses gw.api.database.Relop
uses gw.api.system.database.SequenceUtil

/**
 * Training lab
 */
class ABContactSummary {

  construct(i: int) {
    ExternalID = i
  }

  // Declare ExternalID property using getter and setter
  var _externalID: int  // Don’t use Integer because it will initialize as null.

  property get ExternalID(): int {
    return _externalID
  }

  property set ExternalID(externalId: int) {
    if(externalId == 0) {
      _externalID = SequenceUtil.next(1000, "externalID") as int
    } else
      _externalID = externalId
  }

  // Declare properties using shorthand syntax
  var _contactID: String as ContactID
  var _name: String as Name
  var _numCheckingAccounts: int as NumCheckingAccounts
  var _assignedUserWorkload: int as AssignedUserWorkload

  // Create functions

  /**
   * Function that initializes ABContact properties.
   */
  @Param("contact", "Input parameter of type entity.ABContact")
  function loadSummaryData(contact: ABContact): void {
    this._contactID = contact.PublicID
    this._name = contact.DisplayName
    this._numCheckingAccounts = contact.BankAccounts.countWhere(\account ->
        account.AccountType == typekey.BankAccountType.TC_CHECKING)
    // check if the contact actually has an assigned user
    if(contact.AssignedUser != null) {
      var queryObj = Query.make(ABContact)
      queryObj.compare(ABContact#AssignedUser, Relop.Equals, contact.AssignedUser)
      this._assignedUserWorkload = queryObj.select().Count
    } else {
      _assignedUserWorkload = 0
    }
  }
  /**
   * Function that builds a comma-delimited list.
   */
  @Returns("A comma-delimited list of property values")
  function buildConcatenatedSummary(): String {
    return String.format("%s,%s,%s,%s,%s",
        {_externalID, _contactID, _name, _numCheckingAccounts, _assignedUserWorkload})
  }

}
```

2. Deploy code changes.

a) Restart server.

3. Perform verification steps.

```
uses gw.api.database.Query
uses gw.api.database.Relop
uses si.ta.classes.entity.ABContactSummary

var queryObj = Query.make(ABContact)
queryObj.compare(ABContact#PublicID, Relop.Equals, "ab:5")
var targetContact = queryObj.select().AtMostOneRow

var newSummary = new ABContactSummary(0)
newSummary.loadSummaryData(targetContact)
print(newSummary.buildConcatenatedSummary())
```