

# GELLO querying a database

GELLO can query a database. This example will show some code that queries a database for a postcode based on a given locality. This page will then show it in use in an archetype.

Here is some code for this:

```
imports DB_Postcodes

Let retrievedSuburbName: String = 'Bray Park' -- this will be the entered Suburb or locality
Let retrievedStateName: String = 'NSW' -- this will be the entered State

Let postcodeTable: Sequence(TBL_csv)= csv
Let selectedRowOrRows: Sequence(TBL_csv) = postcodeTable->select((Suburb_locality = retrievedSuburbName)
    and (State = retrievedStateName))

Let postcodeAsInteger: Integer = selectedRowOrRows.Postcode ->flatten()->last()

Let result: Real = postcodeAsInteger.oclAsType(Real)

result.toChar()
```

DB\_Postcodes in this case is a csv file. The MO Gello Authoring tool will also load SQL Lite file and Firebird files. The package presents as a sequence of rows. Here is a screen shot of the simple structure of this example:

	A	B	C	D
1	Postcode	Suburb_locality	State	
2	200	Australian National University	ACT	
3	221	Barton	ACT	
4	2540	Jervis Bay	ACT	
5	2540	Hmas Creswell	ACT	
6	2540	Wreck Bay	ACT	
7	2600	Deakin	ACT	
8	2600	Canberra	ACT	
9	2600	Parkes	ACT	
10	2600	Parliament House	ACT	
11	2600	Harman	ACT	
12	2600	Deakin West	ACT	
13	2600	Capital Hill	ACT	
14	2600	Barton	ACT	
15	2600	Hmas Harman	ACT	
16	2600	Russell	ACT	
17	2600	Duntroon	ACT	
18	2600	Yarralumla	ACT	
19	2601	Canberra	ACT	
20	2601	Acton	ACT	
21	2601	City	ACT	
22	2601	Black Mountain	ACT	
23	2602	Watson	ACT	

This is the import screen:

- Auto-Upgrade
- Gello SQL Setup**
- HTTP Client
- Logging
- PGP Configuration
- Server Group Manager
- Server Setup
- Snomed Lookup
- Specific Servers

### Gello SQL Setup

Add...

Remove

Modify...

Data Source

Queries

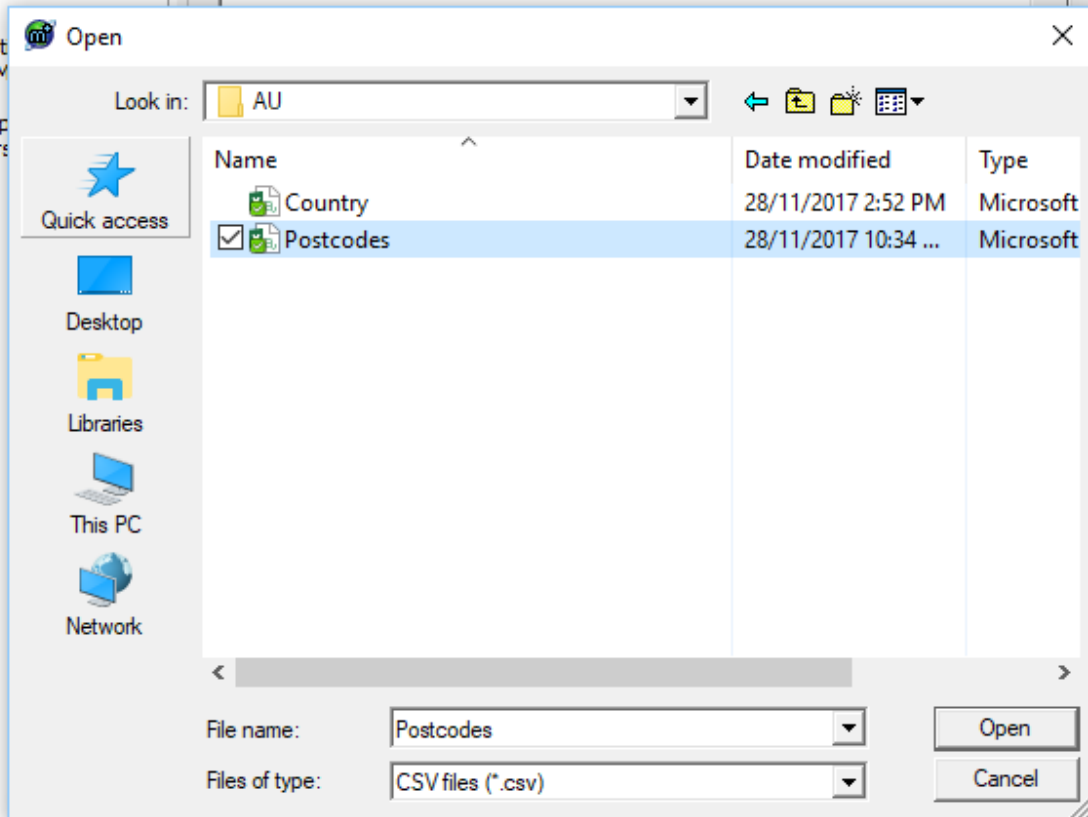
Add...

Remove

Modify...

Auto-Upgrade  
Gello SQL Setup  
HTTP Client  
Logging  
PGP Configurati  
Server Group M  
Server Setup  
Snomed Lookup  
Specific Servers

Gello SQL Setup



Here is the code running:

File Edit Search View Run Build Tools Help



```

1 imports DB_Postcodes
2
3 Let retrievedSuburbName: String = 'Bray Park' -- this will be the entered Suburb or locality
4 Let retrievedStateName: String = 'NSW' -- this will be the entered State
5
6 Let postcodeTable: Sequence(TBL_csv) = csv
7 Let selectedRowOrRows: Sequence(TBL_csv) = postcodeTable->select((Suburb_locality = retrievedSuburbName)
8   and (State = retrievedStateName))
9
10 Let postcodeAsInteger: Integer = selectedRowOrRows.Postcode ->flatten()->last()
11
12 Let result: Real = postcodeAsInteger.oclAsType(Real)
13
14 result.toChar()
    
```

Results Explorer

Name	Class	Data
retrievedSuburbName	String	Bray Park
retrievedStateName	String	NSW
postcodeTable[16875]	Sequence(TBL_csv)	Sequence{<...
selectedRowOrRows[1]	Sequence(Select_TBL_csv)	Sequence{<...
postcodeAsInteger	Integer	2484
result	Integer	2484
_Result_	String	2484

The Suburb and the State can be captured from the UI of an archetype which runs this code in the *onCalculate* field for a Postcode Element:

File Archetypes Debug

ArchetypeID CEN.Demographic.v1  
Definition (TDefinition)  
Description (TDescription)

Preview Archetype

Title  
Given name John  
Middle name  
Family name Smith  
Sex ☐ Female ☒ Male ☐ Indeterminate / Intersex  
Date of birth 01/01/1978  
Street Address 4 Demo St  
Suburb, Town or City Tweed Heads  
State NSW  
Postcode 2485  
Country Australia  
Contact number

Here is the modified code that does this:

Medical-Objects GELLO IDE (Mowgli)

File Edit Search View Run Build Tools Help

CEN.Demographic.v1:at0009@isCalculated

```
1 Context CEN_Demographic_v1::ArchetypeRoot
2 imports DB_Postcodes
3
4 Let suburb: String = If template.Patient_Details.Suburb_Town_or_City.ocIsDefined() then
5     template.Patient_Details.Suburb_Town_or_City.value.ocIsType(CD).displayName.value else null endif
6
7 Let enteredState: String = If template.Patient_Details.State.ocIsDefined() then
8     template.Patient_Details.State.value.ocIsType(CD).displayName.value else null endif
9
10 Let postcodeTable: Sequence(TBL_csv) = csv
11 Let selectedRowOrRows: Sequence(TBL_csv) = postcodeTable->select((Suburb_locality = suburb)
12     and (State = enteredState))
13
14 Let postcodeAsInteger: Integer = selectedRowOrRows.Postcode ->flatten()->last()
15
16 Let result: Real = postcodeAsInteger.ocIsType(Real)
17
18 result.toChar()
```