

# Database update

Business Components. Rules, events, and checks. Review part 2

*GeneXus™*

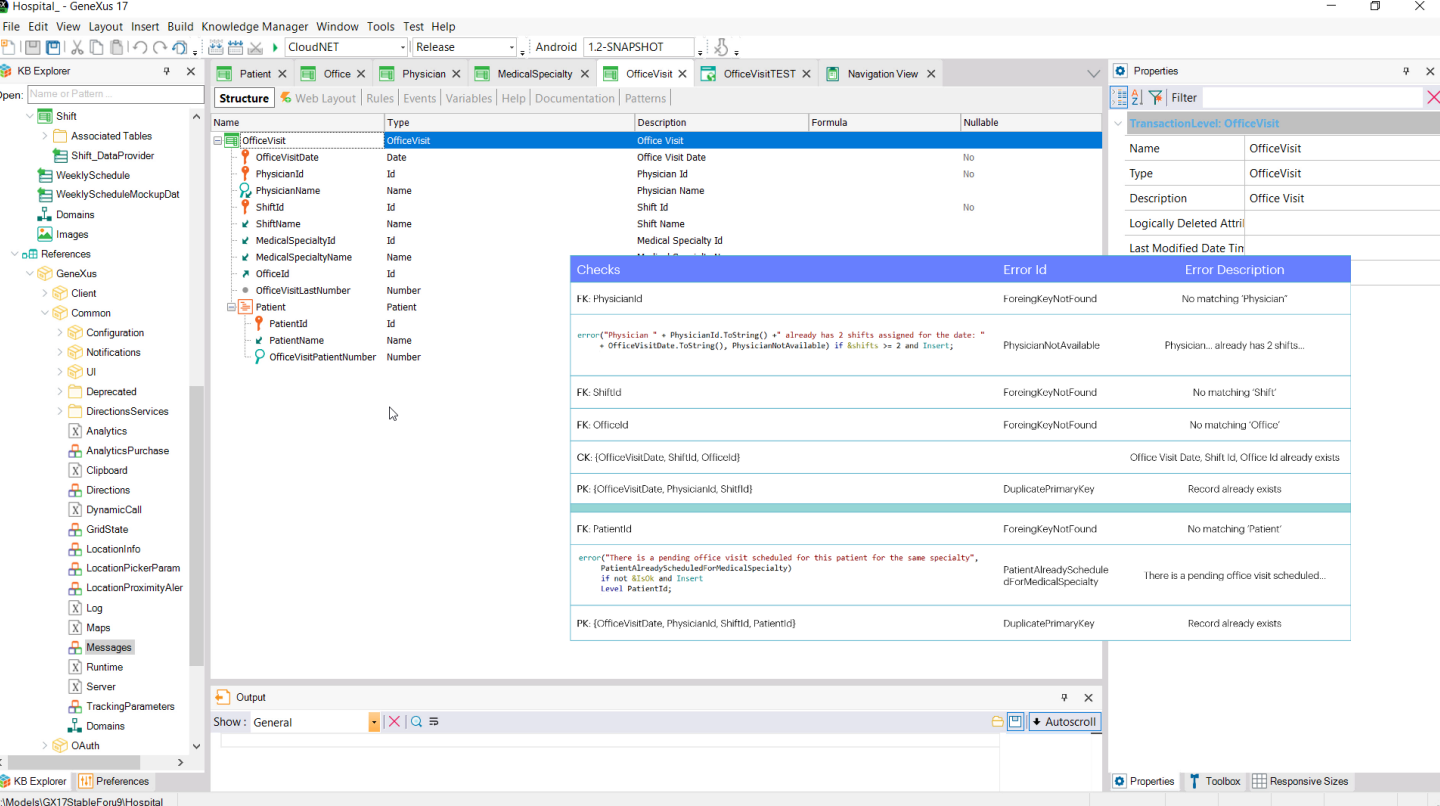


Insert, Update, Delete

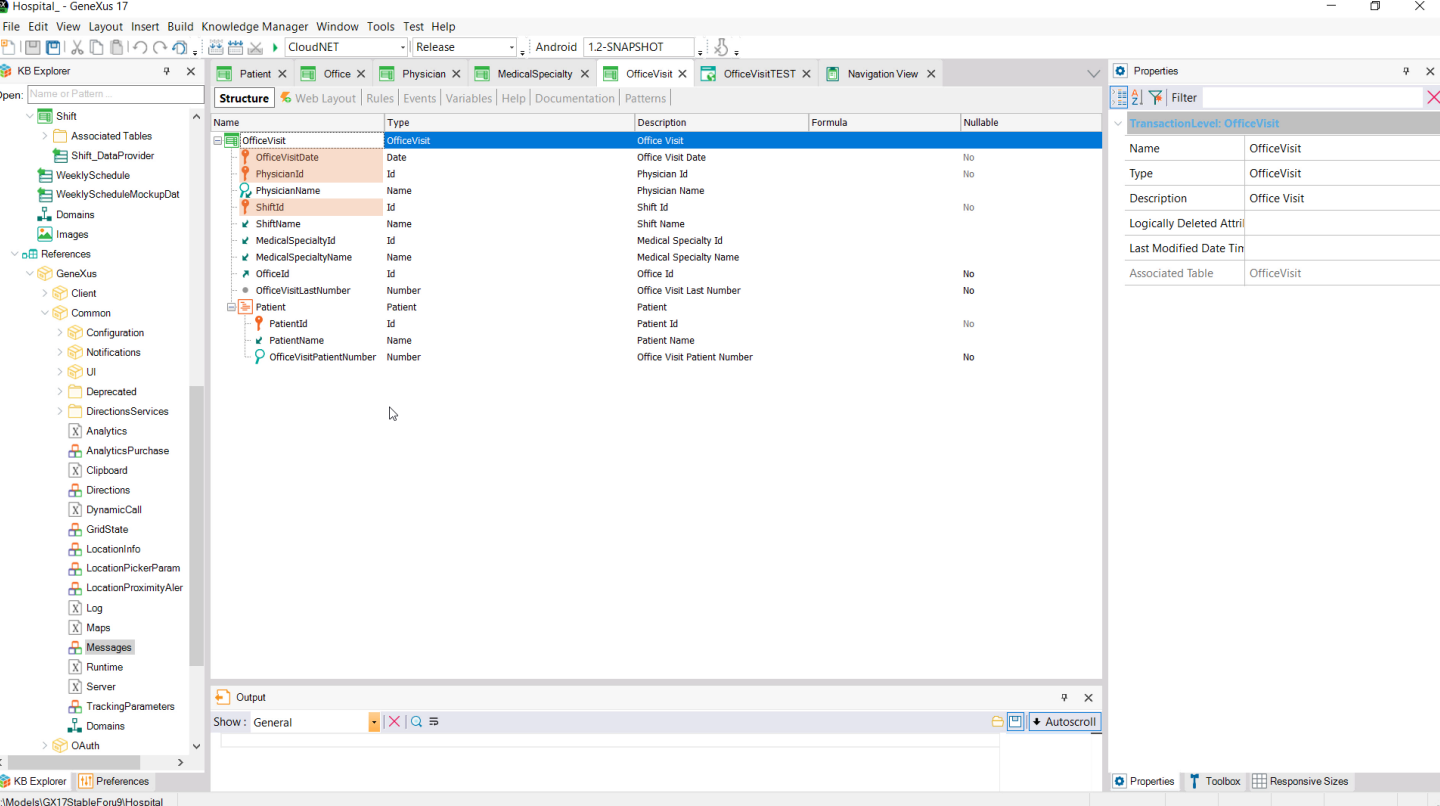
## Business Component

Hospital KB

Focused on a KB for a health care service, in the previous video we saw all the rules, events, and checks that were performed when trying to insert data...

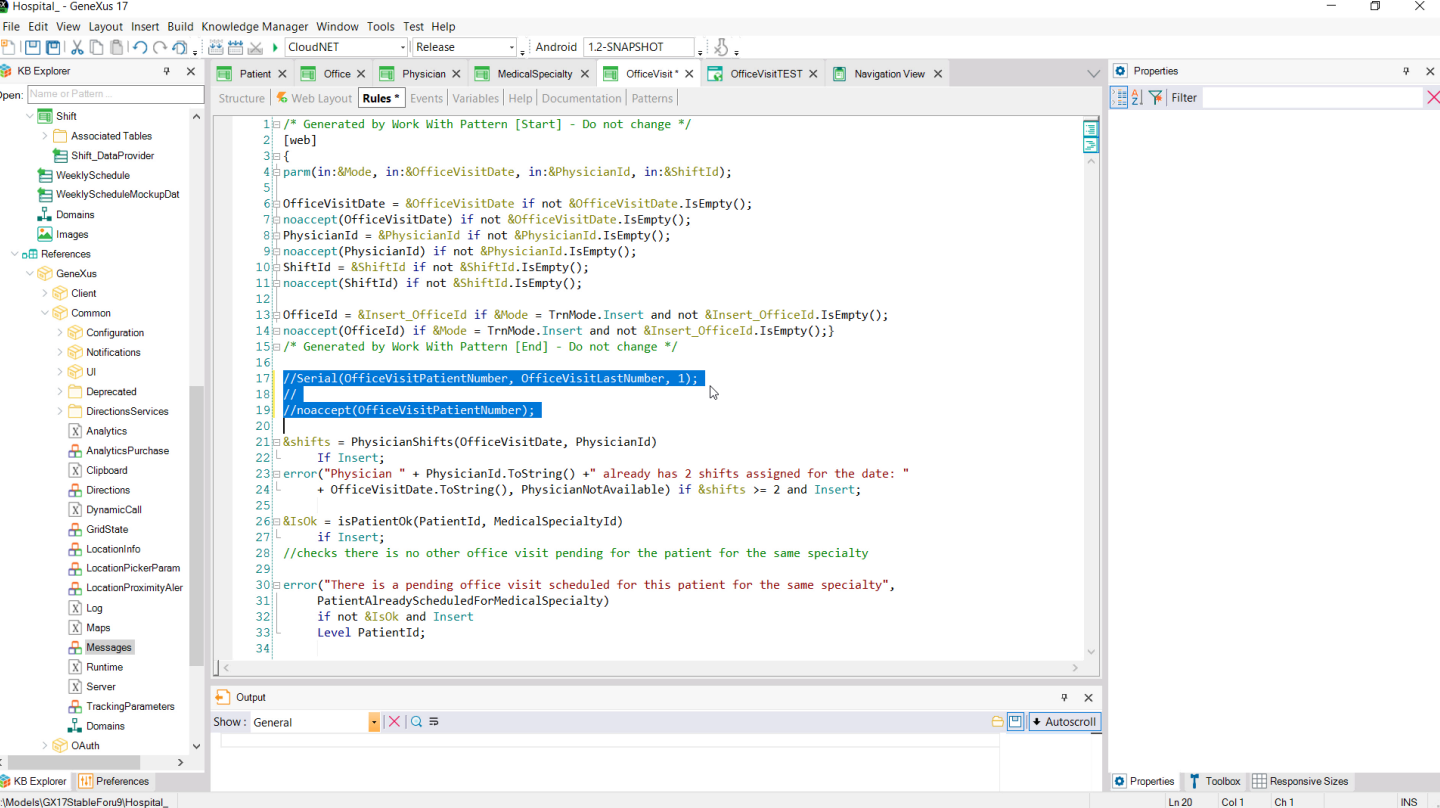


...in a two-level transaction and through the Business Component. Here we will look at update and delete.



We will want to change an office visit. Since the date, physician, and shift are part of the primary key, none of these attributes can be modified for an existing record. We can only modify the office because the last number is used for the serial rule, so it cannot be changed.

The patient entered in a line cannot be changed either because it is part of the primary key. Only the secondary attribute corresponding to the number given to the patient to receive care could be changed, except for the fact that we placed a noaccept rule for that attribute.



We can comment it to have an updatable attribute for each line. Let's comment both the serial rule and the noaccept rule so we can also modify the header attribute, the one for the last given line number.

Of course, we can also add and delete lines to the office visit.

OK, so we want to be able to change the office of a given office visit, change the number assigned to a scheduled patient, add another patient with their corresponding number, and delete one.

	Monday	Tuesday	Wednesday	Thursday	Friday
Shift 1 morning	<div style="border: 1px solid red; padding: 2px;">Physician: 1 Office: 2</div> <div style="border: 1px dashed red; padding: 2px; margin-top: 5px;">Physician: 2 Office: 3 Patient: 1</div>		<div style="border: 1px solid red; padding: 2px;">Physician: 2 Office: 1</div>	<div style="border: 1px solid red; padding: 2px;">Physician: 1 Office: 2 → 3</div> <div style="border: 1px dashed red; padding: 2px; margin-top: 5px;">Patient: 2 Patient: 3</div>	
Shift 2 intermediate	<div style="border: 1px solid red; padding: 2px;">Physician: 1 Office: 2</div>				
Shift 3 evening		<div style="border: 1px solid red; padding: 2px;">Physician: 1 Office: 2</div>			

Let's start with this: update Thursday's office visit, changing the office to 3 and adding patient 3... if we do it through the transaction, first we have to locate the record to be updated...

# Hospital Backoffice



Recents Office Visit — Office Visits

SHOW FILTERS

## Office Visits

Q Name

+ INSERT

Visit Date ▲	Physician Id	Physician Name	Shift Name	Medical Specialty ...	Office Id	Last Number		
11/14/22	1	Doctor 1	Shift morning	Family Medicine	2	0	UPDATE	DELETE
11/14/22	1	Doctor 1	Shift intermediate	Family Medicine	2	0	UPDATE	DELETE
11/14/22	2	Doctor 2	Shift morning	Family Medicine	3	1	UPDATE	DELETE
11/15/22	1	Doctor 1	Shift evening	Family Medicine	2	0	UPDATE	DELETE
11/16/22	2	Doctor 2	Shift morning	Family Medicine	1	0	UPDATE	DELETE
11/17/22	1	Doctor 1	Shift morning	Family Medicine	2	1	UPDATE	DELETE
11/21/22	1	Doctor 1	Shift morning	Family Medicine	1	1	UPDATE	DELETE

https://apps5.genexus.com/Id9df182c70350001219afeaceeb30e8e2/officevisit.aspx?Mode=U...

... by clicking here, the transaction is called by passing the primary key and the Update mode as parameters...

## Office Visit

Visit Date	11/17/22	
Physician Id	1	TrnMode.Update
Physician Name	Doctor 1	
Shift Id	1	
Shift Name	Shift morning	
Medical Specialty Id	1	
Medical Specialty Name	Family Medicine	
Office Id	<input type="text" value=""/>	
Last Number	<input type="text" value="1"/>	

In this way, the record of that primary key is searched for in the header table and its information is retrieved. The same is done for all the related records in the second level table, and their values are retrieved.



GeneXus Developer Menu x Office Visit x +

apps5.genexus.com/ld9df162c70350001219afeaceeb30e8e2/officevisit.aspx?Mode=UPD&OfficeVisitDate=20221117&PhysicianId=1&ShiftId=1

GeneXus DL Portal

Visit Date	11/17/22		
Physician Id	1		
Physician Name	Doctor 1		
Shift Id	1		
Shift Name	Shift morning		
Medical Specialty Id	1		
Medical Specialty Name	Family Medicine		
Office Id	<input type="text" value="3"/>		
Last Number	<input type="text" value="1"/>		

Checks	Error Id	Error Description
FK: PhysicianId	ForeignKeyNotFound	No matching 'Physician'
error["Physician " + PhysicianId.ToString() + " already has 2 shifts assigned for the date: " + OfficeVisitDate.ToString(), PhysicianId] if Shifts >= 2 and Insert;	PhysicianNotAvailable	Physician... already has 2 shifts...
FK: ShiftId	ForeignKeyNotFound	No matching 'Shift'
FK: OfficeId	ForeignKeyNotFound	No matching 'Office'
CK: (OfficeVisitDate, ShiftId, OfficeId)		Office Visit Date, Shift Id, Office Id already exists
PK: (OfficeVisitDate, PhysicianId, ShiftId)	DuplicatePrimaryKey	Record already exists
FK: PatientId	ForeignKeyNotFound	No matching 'Patient'
error["There is a pending office visit scheduled for this patient for the same specialty", PatientAlreadyScheduleFromMedicalSpecialty (if not Edit and Insert Level), PatientId];	PatientAlreadyScheduleFromMedicalSpecialty	There is a pending office visit scheduled...
PK: (OfficeVisitDate, PhysicianId, ShiftId, PatientId)	DuplicatePrimaryKey	Record already exists

Patient Id	Patient Name	Patient Number
<input type="text" value="2"/>	<input type="text" value="Patient 2"/>	<input type="text" value="1"/>
<input type="text" value="0"/>		<input type="text" value="0"/>

We see that the OfficeId field is the first editable field. None of the previous ones can be modified.  
 We change it to 3. Of course, if there were another office visit for that day on that shift in that office, here we would get an error stating that this (candidate) key already exists. This is not the case.

GeneXus Developer Menu    Office Visit

apps5.genexus.com/Id9df182c70350001219afeaceeb30e8e2/officevisit.aspx?Mode=UPD&OfficeVisitDate=20221117&PhysicianId=1&ShiftId=1

GeneXus    DL Portal

Medical Specialty Id	1
Medical Specialty Name	Family Medicine
Office Id	<input type="text" value=""/>
Last Number	<input type="text" value="3"/>

**Patient**

	Patient Id	Patient Name	Patient Number
×	2	Patient 2	<input type="text" value="2"/>
×	3	Patient 3	<input type="text" value="3"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>
	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="text" value="0"/>

[New row]

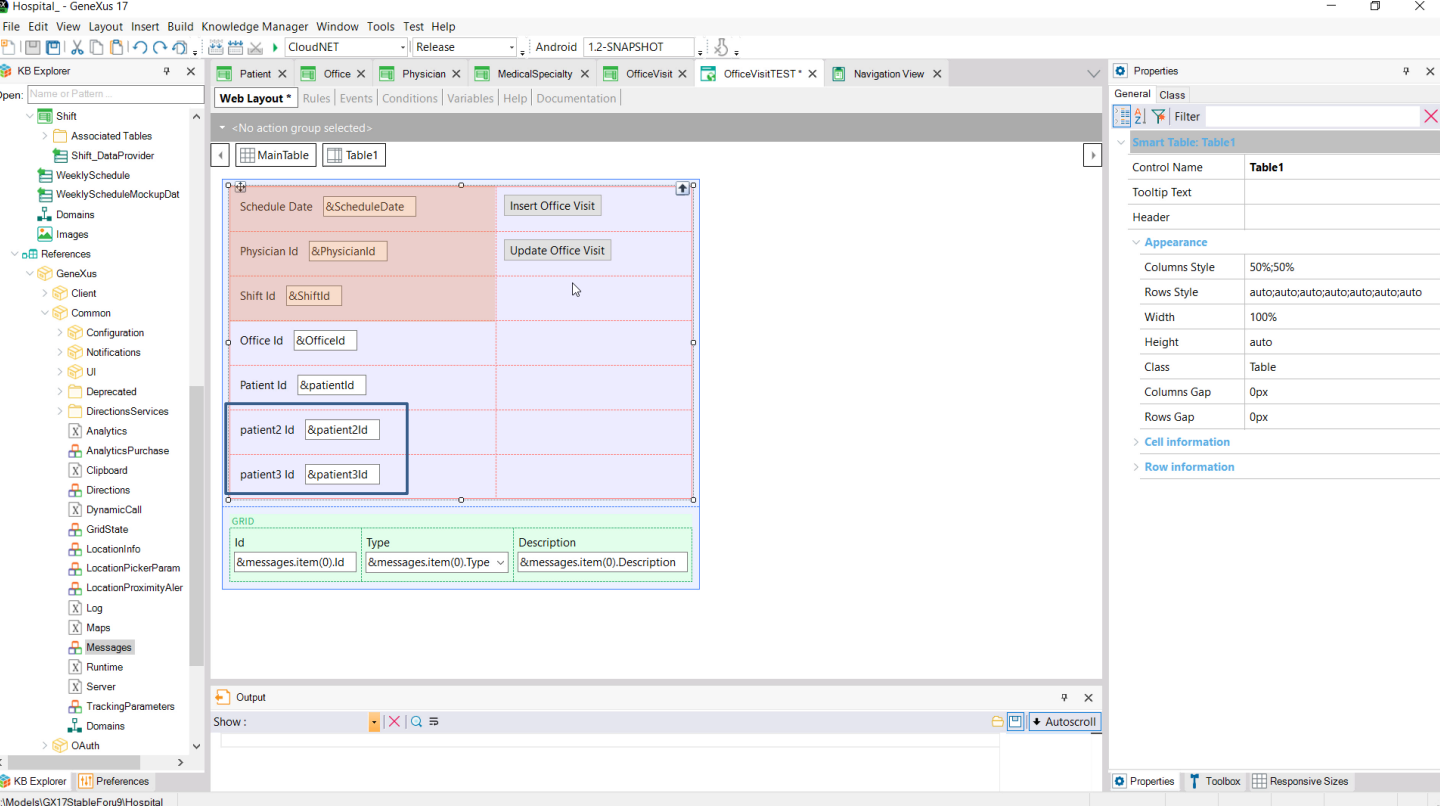
**CONFIRM**    **CANCEL**

Let's change the number given to the first patient to 2, add patient 3, with number 3, and also change the last number to 3. After confirming, the operation performed will be an update of everything, even if at the lines level this means changing one line and adding another.

	Monday	Tuesday	Wednesday	Thursday	Friday
Shift 1 morning	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Physician: 1 Office: 2</div> <div style="border: 1px dashed black; padding: 2px; margin-bottom: 5px;">Physician: 2 Office: 3 Patient: 1</div>		<div style="border: 1px solid black; padding: 2px;">Physician: 2 Office: 1</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Physician: 1 Office: 3 → 2</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Patient: 2</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Patient: 3 → 4</div> <div style="border: 1px dashed black; padding: 2px;">Patient: 4</div>	
Shift 2 intermediate	<div style="border: 1px solid black; padding: 2px;">Physician: 1 Office: 2</div>				
Shift 3 evening		<div style="border: 1px solid black; padding: 2px;">Physician: 1 Office: 2</div>			

Now we will want to work in this way, but through the Business Component.

Let's suppose that, in addition to changing the office, we want to change patient 3 for patient 4. Since we cannot do it directly in the record because it is part of the primary key and it cannot be updated, we will have to delete the record and insert a new one with the correct patient and the same number that patient 3 had.



We have added 2 more variables for the patient we want to replace with the other one for the office visit with that date, physician, and shift. We will add an event to update the office visit of this primary key....

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Release Android 1.2-SNAPSHOT

KB Explorer

Open: Name or Pattern

- Shift
- Associated Tables
- Shift\_DataProvider
- WeeklySchedule
- WeeklyScheduleMockupDat
- Domains
- Images
- References
- GeneXus
- Client
- Common
- Configuration
- Notifications
- UI
- Deprecated
- DirectionsServices
- Analytics
- AnalyticsPurchase
- Clipboard
- Directions
- DynamicCall
- GridState
- LocationInfo
- LocationPickerParam
- LocationProximityAler
- Log
- Maps
- Messages
- Runtime
- Server
- TrackingParameters
- Domains
- OAuth

Web Layout \* Rules Events \* Conditions Variables Help Documentation

'Update Office Visit'

```

1 Event 'Insert Office Visit'
2   &officeVisit = new()
3   &officeVisit.OfficeVisitDate = &ScheduleDate
4   &officeVisit.PhysicianId = &PhysicianId
5   &officeVisit.ShiftId = &ShiftId
6   &officeVisit.OfficeId = &OfficeId
7   If not &patientId.IsEmpty()
8     &officeVisitPatient = new()
9     &officeVisitPatient.PatientId = &patientId
10    &officeVisit.Patient.Add(&officeVisitPatient)
11  endif
12  if &officeVisit.Insert()
13    Commit
14  endif
15  &messages = &officeVisit.GetMessages()
16 Endevent
17
18
19 Event 'Update Office Visit'
20   &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21
22 Endevent

```

&officeVisit

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	3
OfficeVisitLastNumber	3
Patient	

BC loaded! ←

PatientId	?
Patient Name	Patient 2
OfficeVisitPatientNumber	?

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

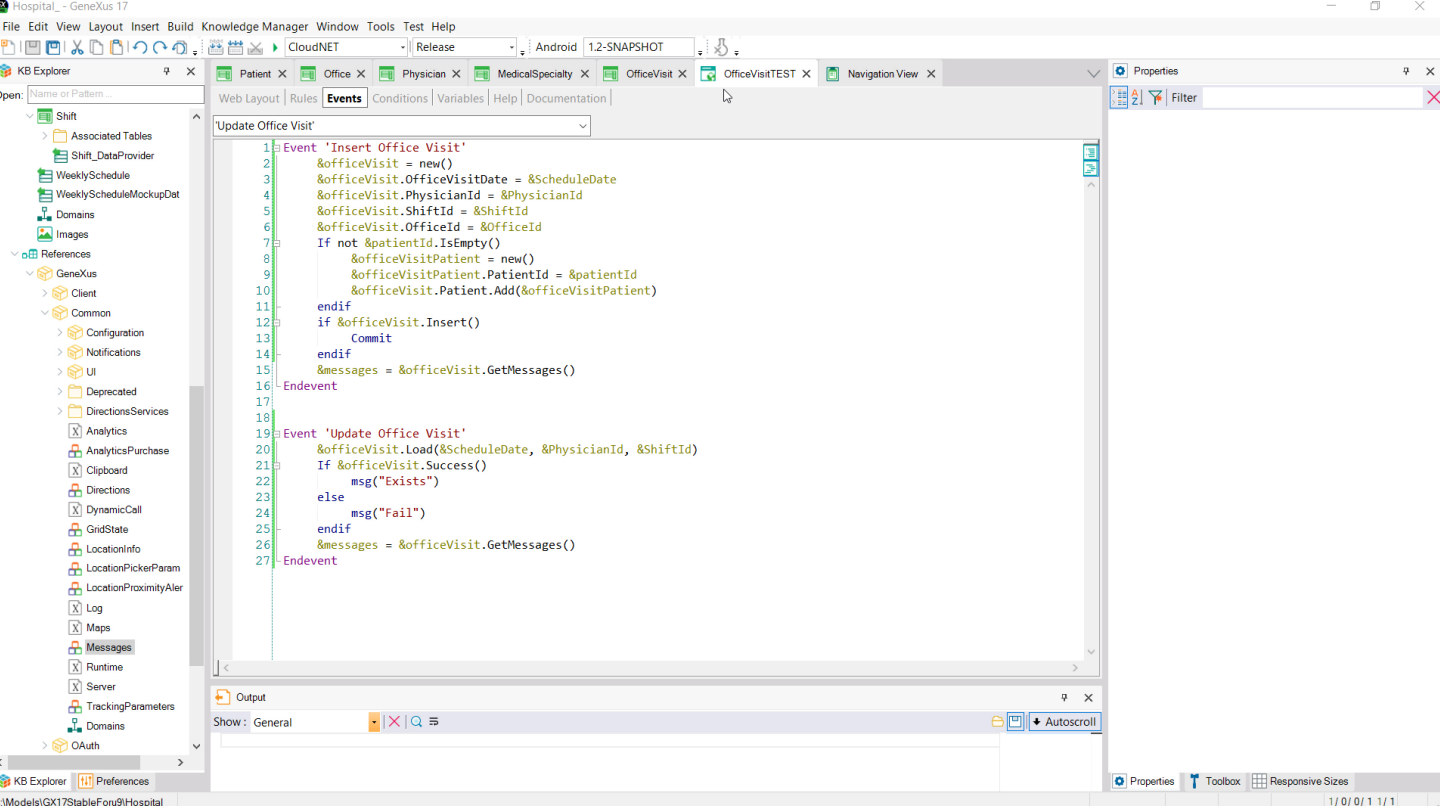
Output

Show: General

Properties

Ln 21 Col 6 Ch 2 INS

First, we have to load the data corresponding to the primary key in the BC variable. The operation will only be successful if there is a record in OfficeVisit with that key.



We can check it with the Success method (or its opposite, Fail). We will display messages on the screen indicating whether the record we tried to load exists. We will also retrieve the messages in the message collection. Let's try it quickly.

# Hospital Backoffice

Recents Office Visit TEST — Office Visits

Office Visits

Q Name

+ INSERT

X HIDE FILTERS

MEDICAL SPECIALTY NAME

OFFICE VISIT DATE

SHIFT NAME

Visit Date	Physician Id	Physician Name	Shift Name	Medical Specialt...	Office Id	Last Number		
11/14/22	1	Doctor 1	Shift morning	Family Medicine	2	0	UPDATE	DELETE
11/14/22	1	Doctor 1	Shift intermediate	Family Medicine	2	0	UPDATE	DELETE
11/14/22	2	Doctor 2	Shift morning	Family Medicine	3	1	UPDATE	DELETE
11/15/22	1	Doctor 1	Shift evening	Family Medicine	2	0	UPDATE	DELETE
11/16/22	2	Doctor 2	Shift morning	Family Medicine	1	0	UPDATE	DELETE
11/17/22	1	Doctor 1	Shift morning	Family Medicine	3	3	UPDATE	DELETE
11/21/22	1	Doctor 1	Shift morning	Family Medicine	1	1	UPDATE	DELETE

We know that this record (the one from Thursday) exists...

# Hospital Backoffice



Recents Office Visit TEST

Exists

Schedule Date

Physician Id

Shift Id

Office Id

Patient Id

patient2 Id

patient3 Id

Id	Type	Description
----	------	-------------

... so we try its key... it was found!



# Hospital Backoffice



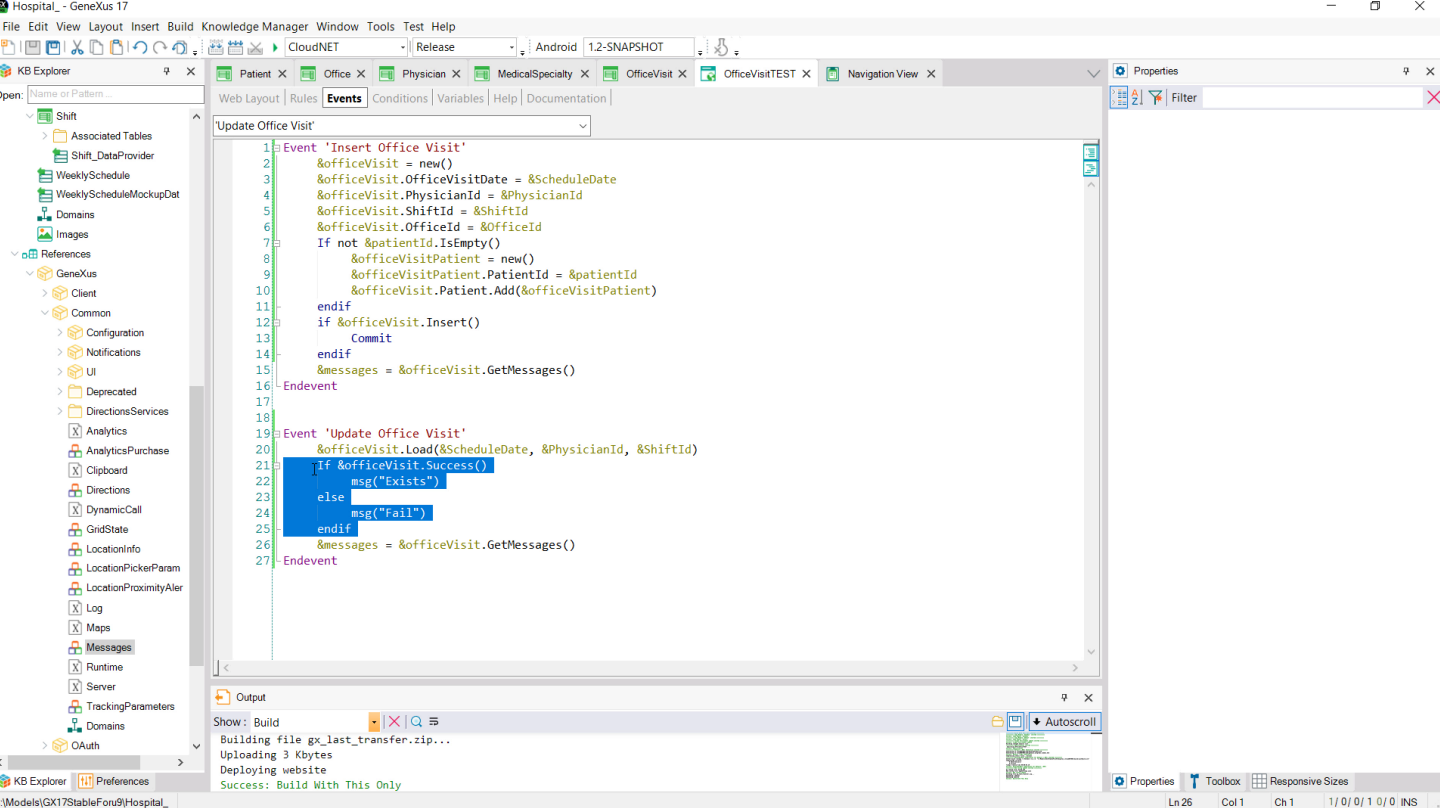
Recents Office Visit TEST

Schedule Date	11/27/22	29	Insert Office Visit
Physician Id	1		Update Office Visit
Shift Id	1		
Office Id	0		
Patient Id	0		
patient2 Id	0		
patient3 Id	0		

Id	Type	Description
PrimaryKeyNotFound	Error	Data with the specified key could not be found.

And now we enter a date for which there are no office visits... We see this other one.

The Success method was False and the message collection shows an error indicating that the primary key was not found.



Let's remove this check because it will be done anyway when we do the update.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET | Release | Android | 1.2-SNAPSHOT

KB Explorer | Patient x | Office x | Physician x | MedicalSpecialty x | OfficeVisit x | OfficeVisitTEST x | Navigation View x | Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 | If not &patientId.IsEmpty()
8 |     &officeVisitPatient = new()
9 |     &officeVisitPatient.PatientId = &patientId
10 |     &officeVisit.Patient.Add(&officeVisitPatient)
11 | endif
12 | if &officeVisit.Insert()
13 |     Commit
14 | endif
15 | &messages = &officeVisit.GetMessages()
16 | -Endevent
17 |
18 |
19 | Event 'Update Office Visit'
20 |     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21 |     &officeVisit.OfficeId = &OfficeId
22 |     &officeVisit.OfficeVisitLastNumber +=1
23 |
24 |     //Update line for &patientId
25 |
26 |     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
27 |     //Delete line for &patient2Id
28 |
29 |     &officeVisitPatient = new()
30 |     &officeVisitPatient.PatientId = &patient3Id
31 |     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
32 |     //Insert line for &patient3Id
33 |
34 |     If &officeVisit.Update()
35 |         commit
36 |     endif
37 |     &messages = &officeVisit.GetMessages()
38 | -Endevent

```

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	2

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

Output

Show: Build | X | Q | S | Autoscroll

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

KB Explorer | Preferences | Properties | Toolbox | Responsive Sizes | Ln 22 | Col 44 | Ch 40 | 1 / 0 / 1 / 1 / 1 INS

Next, we modify the office visit according to the screen variable; we add 1 to the last number given...

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Release Android 1.2-SNAPSHOT

KB Explorer

Patent x Office x Physician x MedicalSpecialty x OfficeVisit x OfficeVisitTEST x Navigation View x Properties

Web Layout | Rules **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 If not &patientId.IsEmpty()
8     &officeVisitPatient = new()
9     &officeVisitPatient.PatientId = &patientId
10    &officeVisit.Patient.Add(&officeVisitPatient)
11 endif
12 if &officeVisit.Insert()
13     Commit
14 endif
15 &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20     &officeVisit.Load(&scheduleDate, &physicianId, &shiftId)
21     &officeVisit.OfficeId = &officeId
22     &officeVisit.OfficeVisitLastNumber +=1
23
24     //Update line for &patientId
25
26     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
27     //Delete line for &patient2Id
28
29
30     &officeVisitPatient = new()
31     &officeVisitPatient.PatientId = &patient3Id
32     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
33     //Insert line for &patient3Id
34
35     If &officeVisit.Update()
36         commit
37     endif
38     &messages = &officeVisit.GetMessages()
39 -Endevent

```

OfficeVisitDate	#2022-11-17
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	2

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

Output

Show: Build

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 24 Col 32 Ch 28 1/0/0 1/1/1 INS

... because it is the one we will use to change the number of the first patient, the one in this variable.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET | Release | Android | 1.2-SNAPSHOT

KB Explorer | Patient x | Office x | Physician x | MedicalSpecialty x | OfficeVisit x | OfficeVisitTEST x | Navigation View x | Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 | If not &patientId.IsEmpty()
8 |     &officeVisitPatient = new()
9 |     &officeVisitPatient.PatientId = &patientId
10 |     &officeVisit.Patient.Add(&officeVisitPatient)
11 | endif
12 | if &officeVisit.Insert()
13 |     Commit
14 | endif
15 | &messages = &officeVisit.GetMessages()
16 | -Endevent
17 |
18 |
19 | Event 'Update Office Visit'
20 |     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21 |     &officeVisit.OfficeId = &OfficeId
22 |     &officeVisit.OfficeVisitLastNumber +=1
23 |
24 | //Update line for &patientId
25 |     &officeVisitPatient = &officeVisit.Patient.GetByKey()
26 |
27 |     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
28 | //Delete line for &patient2Id
29 |
30 |     &officeVisitPatient = new()
31 |     &officeVisitPatient.PatientId = &patient3Id
32 |     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
33 | //Insert line for &patient3Id
34 |
35 | If &officeVisit.Update()
36 |     commit
37 | endif
38 | &messages = &officeVisit.GetMessages()
39 | -Endevent

```

&officeVisit

OfficeVisitDate	#2022.11.17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	—

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	2

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

GetByKey(PrimaryKeyAtt: Variant): Patient

Output

Show: Build | X | Q | Autoscroll

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes  
 Ln 25 | Col 58 | Ch 54 | 1 / 0 / 1 / 1 | INS

To do so, remember that we have to use a variable of the BC data type of each line; and use the GetByKey method of the patient collection of the BC that we loaded here.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET | Release | Android | 1.2-SNAPSHOT

KB Explorer | Patient x | Office x | Physician x | MedicalSpecialty x | OfficeVisit x | OfficeVisitTEST x | Navigation View x | Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 | If not &patientId.IsEmpty()
8 |     &officeVisitPatient = new()
9 |     &officeVisitPatient.PatientId = &patientId
10 |     &officeVisit.Patient.Add(&officeVisitPatient)
11 | endif
12 | if &officeVisit.Insert()
13 |     Commit
14 | endif
15 | &messages = &officeVisit.GetMessages()
16 | -Endevent
17 |
18 |
19 | Event 'Update Office Visit'
20 |     &officeVisit.Load(&scheduleDate, &physicianId, &shiftId)
21 |     &officeVisit.OfficeId = &officeId
22 |     &officeVisit.OfficeVisitLastNumber +=1
23 |
24 | //Update line for &patientId
25 | &officeVisitPatient = &officeVisit.Patient.GetByKey(
26 |     &patientId
27 |     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patientId).OfficeVisitPatientNumber
28 | //Delete line for &patient2Id
29 |
30 | &officeVisitPatient = new()
31 | &officeVisitPatient.PatientId = &patient3Id
32 | &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
33 | //Insert line for &patient3Id
34 |
35 | If &officeVisit.Update()
36 |     commit
37 | endif
38 | &messages = &officeVisit.GetMessages()
39 | -Endevent

```

&officeVisit

OfficeVisitDate	#2022.11.17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

&officeVisitPatient

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	2

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

Output

Show: Build | X | Q | Autoscroll

Building file gx\_last\_transfer.zip...  
Uploading 50 Kbytes  
Deploying website  
Success: Build With This Only

Properties | Toolbox | Responsive Sizes  
Ln 25 | Col 58 | Ch 54 | 1 / 0 / 1 / 1 / 1 INS

We have to send the line ID to the method so that it can return a reference to that item in the collection. It will be a reference to the item itself, to its memory location; it will not be a copy.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Release Android 1.2-SNAPSHOT

KB Explorer

GeneXus Client Common Configuration Notifications UI Deprecated DirectionsServices Analytics AnalyticsPurchase Clipboard DynamicCall GridState LocationInfo LocationPickerParam LocationProximityAler Log Maps Messages Runtime Server TrackingParameters Domains OAuth SD Server Social Domains GXtest Customization Documentation

Patent Office Physician MedicalSpecialty OfficeVisit OfficeVisitTEST Navigation View Properties

Web Layout Rules Events Conditions Variables Help Documentation

'Update Office Visit'

```

7 If not &patientId.IsEmpty()
8     &officeVisitPatient = new()
9     &officeVisitPatient.PatientId = &patientId
10    &officeVisit.Patient.Add(&officeVisitPatient)
11 endif
12 if &officeVisit.Insert()
13     Commit
14 endif
15 &messages = &officeVisit.GetMessages()
16 -Endevent
17
18 Event 'Update Office Visit'
19     &officeVisit.Load(&scheduleDate, &physicianId, &shiftId)
20     &officeVisit.PatientId = &officeId
21     &officeVisit.OfficeVisitLastNumber += 1
22
23     //Update line for &patientId
24     &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
25     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
26
27     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
28     //Delete line for &patient2Id
29
30     &officeVisitPatient = new()
31     &officeVisitPatient.PatientId = &patient3Id
32     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
33     //Insert line for &patient3Id
34
35     If &officeVisit.Update()
36         commit
37     endif
38     &messages = &officeVisit.GetMessages()
39

```

OfficeVisitDate	#2022.11.17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

&officeVisitPatient

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

Output

Show: Build

Building file gx\_last\_transfer.zip...  
 Uploading 50 kbytes  
 Deploying website  
 Success: Build With This Only

Properties

Attribute: OfficeVisitLastNumber

Ln 26 Col 87 Ch 83 1/0/0 1/1/1 INS

Therefore, it will be enough to modify the value of this element for this variable, because it will be modifying the item of the collection directly. There is no need to do any update. This is enough.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET | Release | Android | 1.2-SNAPSHOT

KB Explorer | Patient x | Office x | Physician x | MedicalSpecialty x | OfficeVisit x | OfficeVisitTEST x | Navigation View x | Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 | If not &patientId.IsEmpty()
8 |     &officeVisitPatient = new()
9 |     &officeVisitPatient.PatientId = &patientId
10 |     &officeVisit.Patient.Add(&officeVisitPatient)
11 | endif
12 | if &officeVisit.Insert()
13 |     Commit
14 | endif
15 | &messages = &officeVisit.GetMessages()
16 | -Endevent
17 |
18 |
19 | Event 'Update Office Visit'
20 |     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21 |     &officeVisit.Patient = &officeVisit.Patient
22 |     &officeVisit.OfficeVisitLastNumber += 1
23 |
24 |     //Update line for &patientId
25 |     &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26 |     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27 |
28 |     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
29 |     //Delete line for &patient2Id
30 |
31 |     &officeVisitPatient = new()
32 |     &officeVisitPatient.PatientId = &patient3Id
33 |     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
34 |     //Insert line for &patient3Id
35 |
36 |     If &officeVisit.Update()
37 |         commit
38 |     endif
39 |     &messages = &officeVisit.GetMessages()

```

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift, morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	—

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

PatientId	4
Patient Name	Patient 4
OfficeVisitPatientNumber	3

Output

Show: Build | X | Q | S

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 27 | Col 6 | Ch 2 | 1 / 0 / 1 / 1 / 1 | INS

Now we want to delete the line corresponding to this patient, to insert another one and change the patient.



File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Android 1.2-SNAPSHOT

KB Explorer

Patent x Office x Physician x MedicalSpecialty x OfficeVisit x OfficeVisitTEST x Navigation View x Properties

Web Layout | Rules **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7   If not &patientId.IsEmpty()
8       &officeVisitPatient = new()
9       &officeVisitPatient.PatientId = &patientId
10      &officeVisit.Patient.Add(&officeVisitPatient)
11  endif
12  if &officeVisit.Insert()
13      Commit
14  endif
15  &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21     &officeVisit.OfficeId = &OfficeId
22     &officeVisit.OfficeVisitLastNumber +=1
23
24     //Update line for &patientId
25     &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
29     //Delete line for &patient2Id
30     |
31
32     &officeVisitPatient = new()
33     &officeVisitPatient.PatientId = &patient3Id
34     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35     //Insert line for &patient3Id
36
37     If &officeVisit.Update()
38         commit
39     endif

```

OfficeVisitDate	#2022.11.17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	—

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	3
Patient Name	Patient 3
OfficeVisitPatientNumber	3

Output

Show: Build | X | Q | S

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 30 Col 6 Ch 2 1/0/0 1/1/1 INS

But before deleting it, we need to retrieve the number that was assigned to it. To do so, we assign what's shown on the right to a variable where we will store that value. Here we are retrieving a reference to the BC of the lines that have this value in the identifier. And from that BC, we are asking for the Patient Number.

The screenshot shows a development environment with a code editor, a Properties window, and an Output window. The code editor displays the following code for the 'Update Office Visit' event:

```

7 If not &patientId.IsEmpty()
8     &officeVisitPatient = new()
9     &officeVisitPatient.PatientId = &patientId
10    &officeVisit.Patient.Add(&officeVisitPatient)
11 endif
12 if &officeVisit.Insert()
13     Commit
14 endif
15 &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20     &officeVisit.Load(&scheduleDate, &physicianId, &shiftId)
21     &officeVisit.PatientId = &officeId
22     &officeVisit.OfficeVisitLastNumber += 1
23
24     //Update line for &patientId
25     &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
29     //Delete line for &patient2Id
30     &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32     &officeVisitPatient = new()
33     &officeVisitPatient.PatientId = &patient3Id
34     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35     //Insert line for &patient3Id
36
37 If &officeVisit.Update()
38     commit
39 endif

```

The Properties window shows the following data for the 'OfficeVisit' object:

OfficeVisitDate	#2022.11.17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	—

The Properties window also shows the following data for the 'Patient' object:

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

The Output window shows the following build output:

```

Show: Build
Building file gx_last_transfer.zip...
Uploading 50 Kbytes
Deploying website
Success: Build With This Only

```

Now we can delete the line. To do so, on the list of patients for an office visit, we run the RemoveByKey method, to which we send the identifier of the line that we want to delete, which is the ID of this patient, that of the second variable of the web panel screen. For now, this method only deleted the item from the collection in memory. We have not done anything in the database yet.

KB Explorer - GeneXus 17

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Release Android 1.2-SNAPSHOT

KB Explorer

GeneXus

- Client
- Common
- Configuration
- Notifications
- UI
- Deprecated
- DirectionsServices
- Analytics
- AnalyticsPurchase
- Clipboard
- Directions
- DynamicCall
- GridState
- LocationInfo
- LocationPickerParam
- LocationProximityAler
- Log
- Maps
- Messages
- Runtime
- Server
- TrackingParameters
- Domains
- OAuth
- SD
- Server
- Social
- Domains
- GXtest
- Customization
- Documentation

Patent x Office x Physician x MedicalSpecialty x OfficeVisit x OfficeVisitTEST x Navigation View x Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7   If not &patientId.IsEmpty()
8       &officeVisitPatient = new()
9       &officeVisitPatient.PatientId = &patientId
10      &officeVisit.Patient.Add(&officeVisitPatient)
11  endif
12  if &officeVisit.Insert()
13      Commit
14  endif
15  &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21     &officeVisit.OfficeId = &OfficeId
22     &officeVisit.OfficeVisitLastNumber +=1
23
24     //Update line for &patientId
25     &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28     &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patientId).OfficeVisitPatientNumber
29     //Delete line for &patient2Id
30     &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32     &officeVisitPatient = new()
33     &officeVisitPatient.PatientId = &patient3Id
34     &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35     //Insert line for &patient3Id
36     |
37
38     If &officeVisit.Update()
39         commit

```

&officeVisit

OfficeVisitDate	#2022-11-1#
PhysicianId	3
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	3
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

OfficeVisitPatient

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	4
Patient Name	
OfficeVisitPatientNumber	3

Output

Show: Build

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 36 Col 6 Ch 2 1/0/0/1/1/1 INS

Lastly, we must add a new item to the collection, for the substitute patient, with the number given to the patient being replaced.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Release Android 1.2-SNAPSHOT

KB Explorer

GeneXus

- Client
- Common
- Configuration
- Notifications
- UI
- Deprecated
- DirectionsServices
- Analytics
- AnalyticsPurchase
- Clipboard
- Directions
- DynamicCall
- GridState
- LocationInfo
- LocationPickerParam
- LocationProximityAler
- Log
- Maps
- Messages
- Runtime
- Server
- TrackingParameters
- Domains
- OAuth
- SD
- Server
- Social
- Domains
- GXtest
- Customization
- Documentation

Patent x Office x Physician x MedicalSpecialty x OfficeVisit x OfficeVisitTEST x Navigation View x Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 If not &patientId.IsEmpty()
8     &officeVisitPatient = new()
9     &officeVisitPatient.PatientId = &patientId
10    &officeVisit.Patient.Add(&officeVisitPatient)
11 endif
12 if &officeVisit.Insert()
13     Commit
14 endif
15 &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20     &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21     &officeVisit.OfficeId = &OfficeId
22     &officeVisit.OfficeVisitLastNumber +=1
23
24 //Update line for &patientId
25 &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26 &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28 &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patientId).OfficeVisitPatientNumber
29 //Delete line for &patient2Id
30 &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32 &officeVisitPatient = new()
33 &officeVisitPatient.PatientId = &patient3Id
34 &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35 //Insert line for &patient3Id
36 &officeVisit.Patient.Add(&officeVisitPatient)
37
38 If &officeVisit.Update()
39     commit

```

&officeVisit

OfficeVisitDate	#2022-11-1#
PhysicianId	3
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	3
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

&officeVisitPatient

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	4
Patient Name	
OfficeVisitPatientNumber	3

Output

Show: Build

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 32 Col 22 Ch 18 1/0/0/1/1/1 INS

We still need to insert this item to the collection. We do it with the Add method.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Android 1.2-SNAPSHOT

KB Explorer

GeneXus  
 Client  
 Common  
 Configuration  
 Notifications  
 UI  
 Deprecated  
 DirectionsServices  
 Analytics  
 AnalyticsPurchase  
 Clipboard  
 Directions  
 DynamicCall  
 GridState  
 LocationInfo  
 LocationPickerParam  
 LocationProximityAler  
 Log  
 Maps  
 Messages  
 Runtime  
 Server  
 TrackingParameters  
 Domains  
 OAuth  
 SD  
 Server  
 Social  
 Domains  
 GXtest  
 Customization  
 Documentation

Patent x Office x Physician x MedicalSpecialty x OfficeVisit x OfficeVisitTEST x Navigation View x Properties

Web Layout | Rules | **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

7 If not &patientId.IsEmpty()
8   &officeVisitPatient = new()
9   &officeVisitPatient.PatientId = &patientId
10  &officeVisit.Patient.Add(&officeVisitPatient)
11 endif
12 if &officeVisit.Insert()
13   Commit
14 endif
15 &messages = &officeVisit.GetMessages()
16 -Endevent
17
18
19 Event 'Update Office Visit'
20   &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21   &officeVisit.OfficeId = &OfficeId
22   &officeVisit.OfficeVisitLastNumber +=1
23
24 //Update line for &patientId
25 &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26 &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28 &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
29 //Delete line for &patient2Id
30 &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32 &officeVisitPatient = new()
33 &officeVisitPatient.PatientId = &patient3Id
34 &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35 //Insert line for &patient3Id
36 &officeVisit.Patient.Add(&officeVisitPatient)
37
38 If &officeVisit.Update()
39   commit
  
```

&officeVisit

OfficeVisitDate	#2022-11-1#
PhysicianId	3
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	3
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

&officeVisitPatient

PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

PatientId	4
Patient Name	
OfficeVisitPatientNumber	3

Output

Show: Build | X | Q | Autoscroll

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties | Toolbox | Responsive Sizes

Ln 32 Col 22 Ch 18 1/0/0/1/1/1 INS

In this case, it was essential to ask for new memory for this BC variable of the line, because we had already used it here. If we didn't ask for new memory, then we would be inserting to the collection the same location in memory that was already there, so instead of 2 lines—one for patientId and another for patient3Id—we would have only one, with these values that will have overwritten the first ones.

The screenshot displays the GeneXus IDE interface. The main window shows the code for the 'Update Office Visit' event. The code includes logic for loading an office visit, updating its patient information, deleting an old entry, and inserting a new one. A red circle highlights the `&officeVisit.Update()` call. To the right, a data table for `&officeVisit` is shown with fields like `OfficeVisitDate`, `PhysicianId`, and `OfficeVisitLastNumber`. Below this, a database icon is connected to a table with columns `PatientId`, `Patient Name`, and `OfficeVisitPatientNumber`. The output window at the bottom shows the build process completion.

```

11  endif
12  if &officeVisit.Insert()
13      Commit
14  endif
15  &messages = &officeVisit.GetMessages()
16  -Endevent
17
18
19  Event 'Update Office Visit'
20      &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21      &officeVisit.OfficeId = &OfficeId
22      &officeVisit.OfficeVisitLastNumber +=1
23
24      //Update line for &patientId
25      &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26      &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28      &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNum
29      //Delete line for &patient2Id
30      &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32      &officeVisitPatient = new()
33      &officeVisitPatient.PatientId = &patient3Id
34      &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35      //Insert line for &patient3Id
36      &officeVisit.Patient.Add(&officeVisitPatient)
37
38      If &officeVisit.Update()
39          commit
40      endif
41      &messages = &officeVisit.GetMessages()
42  -Endevent
  
```

&officeVisit	
OfficeVisitDate	#2022-11-1#
PhysicianId	3
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	3
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	4
Patient	

Patient	
PatientId	2
Patient Name	Patient 2
OfficeVisitPatientNumber	4

Patient	
PatientId	4
Patient Name	Patient 2
OfficeVisitPatientNumber	3

Output  
 Show: Build  
 Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

At this point, we have already made all the changes we wanted on the BC variable that we loaded from the database. Now we need to ask for what we would ask in the transaction by pressing the Confirm button. That is to say, to update the information in the database, running all the Business Component checks. If the result of the Update is successful, we commit. And in any case we retrieve the messages that are generated. Let's try this.

Shift Id	1		
Shift Name	Shift morning		
Medical Specialty Id	1		
Medical Specialty Name	Family Medicine		
Office Id	2		
Last Number	3	4	
<b>Patient</b>			
	<b>Patient Id</b>	<b>Patient Name</b>	<b>Patient Number</b>
×	2	Patient 2	2 4
×	3	Patient 3	3 4
→	4		3
	0		0
	0		0
	0		0
	0		0

For this office visit, we want to change the office to 2, give patient 2 the number following the last one and update the last one; delete patient 3 and add patient 4 with their number.

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	1		<input type="button" value="Update Office Visit"/>
Shift Id	1		
Office Id	2		
Patient Id	2		
patient2 Id	3		
patient3 Id	4		

Id	Type	Description
SuccessfullyUpdated	Warning	Data has been successfully updated.

So here we choose these values, those of that primary key... office 2, patient 2; 3 that we want to change to 4. Success, as we expected.



# Hospital Backoffice

Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date: 11/17/22 [29]

Physician Id: 1

Shift Id: 1

Office Id: 3

Patient Id: 2

patient2 Id: 4

patient3 Id: 1

Insert Office Visit

Update Office Visit

Monday	Tuesday	Wednesday	Thursday	Friday
Physician: 1 Office: 2 Physician: 2 Office: 3 Patient: 1		Physician: 2 Office: 1	Physician: 1 Office: 2 Patient: 2	
Physician: 1 Office: 2				
	Physician: 1 Office: 2			

Now let's cause the update to fail to confirm that it's checking everything that we would expect it to check.

For example, let's try to change the office to 3 (no problem), give patient 2 the number following the last one (no problem), and delete patient 4 (no problem), to insert a new line for patient 1, who we know is already scheduled for an office visit involving this specialty.

# Hospital Backoffice



Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date: 11/17/22

Physician Id:

Shift Id:

Office Id:

Patient Id:

patient2 Id:

patient3 Id:

Insert Office Visit

Update Office Visit

Checks	Error Id	Error Description
FK: PhysicianId	ForeignKeyNotFound	No matching 'Physician'
<code>error("Physician " + PhysicianId.ToString() + " already has 2 shifts assigned for the date: " + OfficeVisitDate.ToString(), PhysicianNotAvailable) if &amp;shifts &gt;= 2 and Insert;</code>	PhysicianNotAvailable	Physician... already has 2 shifts...
FK: ShiftId	ForeignKeyNotFound	No matching 'Shift'
FK: OfficeId	ForeignKeyNotFound	No matching 'Office'
CK: {OfficeVisitDate, ShiftId, OfficeId}		Office Visit Date, Shift Id, Office Id already exists
PK: {OfficeVisitDate, PhysicianId, ShiftId}	DuplicatePrimaryKey	Record already exists
FK: PatientId	ForeignKeyNotFound	No matching 'Patient'
<code>error("There is a pending office visit scheduled for this patient for the same specialty", PatientAlreadyScheduledForMedicalSpecialty) if not &amp;IsOk and Insert Level PatientId;</code>	PatientAlreadyScheduledForMedicalSpecialty	There is a pending office visit scheduled...
PK: {OfficeVisitDate, PhysicianId, ShiftId, PatientId}	DuplicatePrimaryKey	Record already exists

Since the line will be inserted, the transaction error rule that prevents the database from being updated should be executed. Let's try it...

# Hospital Backoffice

Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	1		<input type="button" value="Update Office Visit"/>
Shift Id	1		
Office Id	3		
Patient Id	2		
patient2 Id	4		
patient3 Id	1		

Id	Type	Description
PatientAlreadyScheduledForMedicalSpecialty	Error	There is a pending office visit scheduled for this patient for the same specialt

It was indeed fired. And the office visit remained unchanged. No update was made to the database.

# Hospital Backoffice



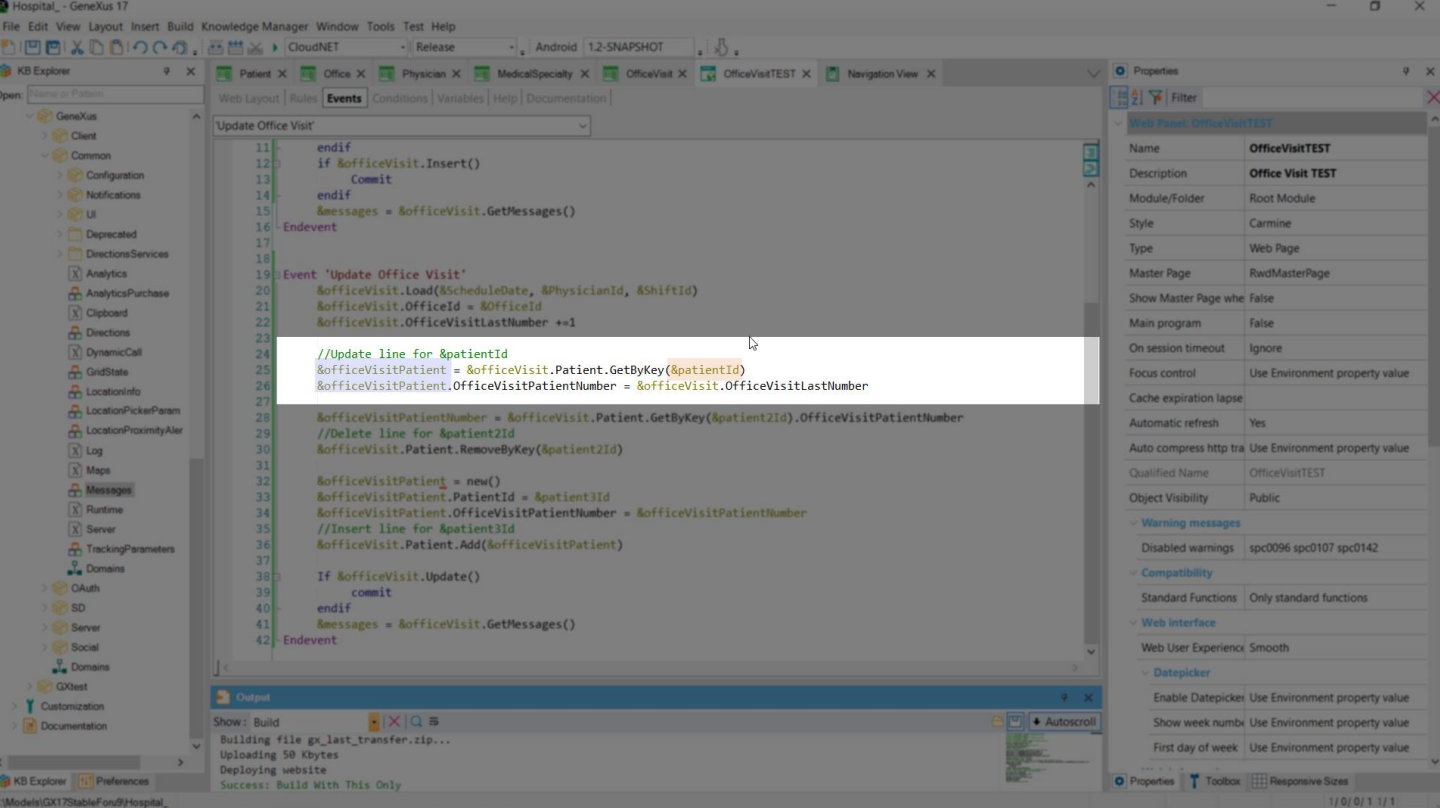
Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	<input type="text" value="1"/>		<input type="button" value="Update Office Visit"/>
Shift Id	<input type="text" value="1"/>		
Office Id	<input type="text" value="3"/>		
Patient Id	<input type="text" value="1"/>		
patient2 Id	<input type="text" value="4"/>		
patient3 Id	<input type="text" value="3"/>		

Id	Type	Description
SuccessfullyUpdated	Warning	Data has been successfully updated.

If we now choose to give the last number to a patient who is not included in the office visit (for example 1, who is not there... and we change 4 to 3, which should not cause any problems)... that is, we change the office to 3, choose patient 1 to change their number, and change 4 to 3.

Why would it let us update successfully if patient 1 does not exist in the office visit?



Let's look at the code. The problem is that GetByKey did not find an item for patient 1 in the line collection, so this variable will not be able to update the line because it will not be pointing anywhere. The other operations did not find any problems.

File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help

CloudNET Android 1.2-SNAPSHOT

KB Explorer

Open: Name or Pattern

- GeneXus
  - Client
  - Common
  - Configuration
  - Notifications
  - UI
  - Deprecated
  - DirectionsServices
  - Analytics
    - AnalyticsPurchase
    - Clipboard
    - Directions
    - DynamicCall
    - GridState
    - LocationInfo
    - LocationPickerParam
    - LocationProximityAler
    - Log
    - Maps
    - Messages
    - Runtime
    - Server
    - TrackingParameters
    - Domains
      - OAuth
      - SD
      - Server
      - Social
      - Domains
      - GXtest
      - Customization
      - Documentation

Web Layout | Rules **Events** | Conditions | Variables | Help | Documentation

'Update Office Visit'

```

11  endif
12  if &officeVisit.Insert()
13      Commit
14  endif
15  &messages = &officeVisit.GetMessages()
16  -Endevent
17
18
19  Event 'Update Office Visit'
20      &officeVisit.Load(&ScheduleDate, &PhysicianId, &ShiftId)
21      &officeVisit.OfficeId = &OfficeId
22      &officeVisit.OfficeVisitLastNumber +=1
23
24      //Update line for &patientId
25      &officeVisitPatient = &officeVisit.Patient.GetByKey(&patientId)
26      &officeVisitPatient.OfficeVisitPatientNumber = &officeVisit.OfficeVisitLastNumber
27
28      &officeVisitPatientNumber = &officeVisit.Patient.GetByKey(&patient2Id).OfficeVisitPatientNumber
29      //Delete line for &patient2Id
30      &officeVisit.Patient.RemoveByKey(&patient2Id)
31
32      &officeVisitPatient = new()
33      &officeVisitPatient.PatientId = &patient3Id
34      &officeVisitPatient.OfficeVisitPatientNumber = &officeVisitPatientNumber
35      //Insert line for &patient3Id
36      &officeVisit.Patient.Add(&officeVisitPatient)
37
38      If &officeVisit.Update()
39          commit
40      endif
41      &messages = &officeVisit.GetMessages()
42  -Endevent
  
```

Output

Show: Build | X | Q | S

Building file gx\_last\_transfer.zip...  
 Uploading 50 Kbytes  
 Deploying website  
 Success: Build With This Only

Properties

Web Panel: OfficeVisitTEST

Name	OfficeVisitTEST
Description	Office Visit TEST
Module/Folder	Root Module
Style	Carmine
Type	Web Page
Master Page	RwdMasterPage
Show Master Page whe	False
Main program	False
On session timeout	Ignore
Focus control	Use Environment property value
Cache expiration lapse	
Automatic refresh	Yes
Auto compress http tra	Use Environment property value
Qualified Name	OfficeVisitTEST
Object Visibility	Public
<b>Warning messages</b>	
Disabled warnings	spc0096 spc0107 spc0142
<b>Compatibility</b>	
Standard Functions	Only standard functions
<b>Web interface</b>	
Web User Experience	Smooth
<b>Datepicker</b>	
Enable Datepicker	Use Environment property value
Show week numb	Use Environment property value
First day of week	Use Environment property value

KB Explorer | Preferences

Models\GX17StableFon9\Hospital\_

1 / 0 / 1 / 1 / 1

If now, to update the first patient we give that of an existing line, for which we also want to delete, but we make a mistake with the new one, and choose one that does not exist...

GeneXus Developer Menu   Office Visit   Office Visit TEST

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officevisit.aspx?Mode=UPD&OfficeVisitDate=20221117&PhysicianId=1&ShiftId=1

GeneXus   DL Portal

Shift Name: Shift morning

Medical Specialty Id: 1

Medical Specialty Name: Family Medicine

Office Id:

Last Number:

**Patient**

Patient Id	Patient Name	Patient Number
× 2 <input type="button" value="↑"/>	Patient 2	<input type="text" value="4"/>
× 3 <input type="button" value="↑"/>	Patient 3	<input type="text" value="3"/>
<input type="text" value="0"/> <input type="button" value="↑"/>		<input type="text" value="0"/>
<input type="text" value="0"/> <input type="button" value="↑"/>		<input type="text" value="0"/>
<input type="text" value="0"/> <input type="button" value="↑"/>		<input type="text" value="0"/>
<input type="text" value="0"/> <input type="button" value="↑"/>		<input type="text" value="0"/>
<input type="text" value="0"/> <input type="button" value="↑"/>		<input type="text" value="0"/>

[New row]

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	3
OfficeVisitLastNumber	6
Patient	

↓

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

↓

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

↓

PatientId	25
Patient Name	
PatientOfficeVisitNumber	4

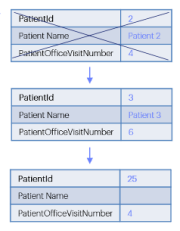
In this way, for example, we ask to give the last number to patient 3, delete patient 2 and replace it with 25...

# Hospital Backoffice

Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date	11/17/22	29	Insert Office Visit
Physician Id	1		Update Office Visit
Shift Id	1		
Office Id	2		
Patient Id	3		
patient2 Id	2		
patient3 Id	25		

OfficeVisitDate	#2022 11 17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	



Id	Type	Description
ForeignKeyNotFound	Error	No matching 'Patient'.

...changing the office to 2, for example... well, we get the referential integrity error of the patient.



GeneXus Developer Menu | Office Visit | Office Visit TEST | apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officevisittest.aspx

# Hospital Backoffice

Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date: 11/17/22 29 Insert Office Visit

Physician Id: 1 Update Office Visit

Shift Id: 1

Office Id: 2

Patient Id: 3 update

patient2 Id: 25 delete

patient3 Id: 4 insert

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

update

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

insert

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

PatientId	4
Patient Name	
PatientOfficeVisitNumber	0

Id	Type	Description
SuccessfullyUpdated	Warning	Data has been successfully updated.

What if we made a mistake the other way around, with the one we want to delete? It allowed us to update! Why? Let's see what it did.

GeneXus Developer Menu | Office Visit | Office Visit TEST | +

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officevisit.aspx?Mode=UPD&OfficeVisitDate=20221117&PhysicianId=1&ShiftId=1

GeneXus | DL Portal

Shift Name	Shift morning
Medical Specialty Id	1
Medical Specialty Name	Family Medicine
Office Id	<input type="text" value="2"/> ↑
Last Number	<input type="text" value="6"/>

**Patient**

Patient Id	Patient Name	Patient Number
× 2 ↑	Patient 2	<input type="text" value="4"/>
× 3 ↑	Patient 3	<input type="text" value="6"/>
× 4 ↑	Patient 4	<input type="text" value="0"/>
<input type="text" value="0"/> ↑		<input type="text" value="0"/>
<input type="text" value="0"/> ↑		<input type="text" value="0"/>
<input type="text" value="0"/> ↑		<input type="text" value="0"/>
<input type="text" value="0"/> ↑		<input type="text" value="0"/>
<input type="text" value="0"/> ↑		<input type="text" value="0"/>

[New row]

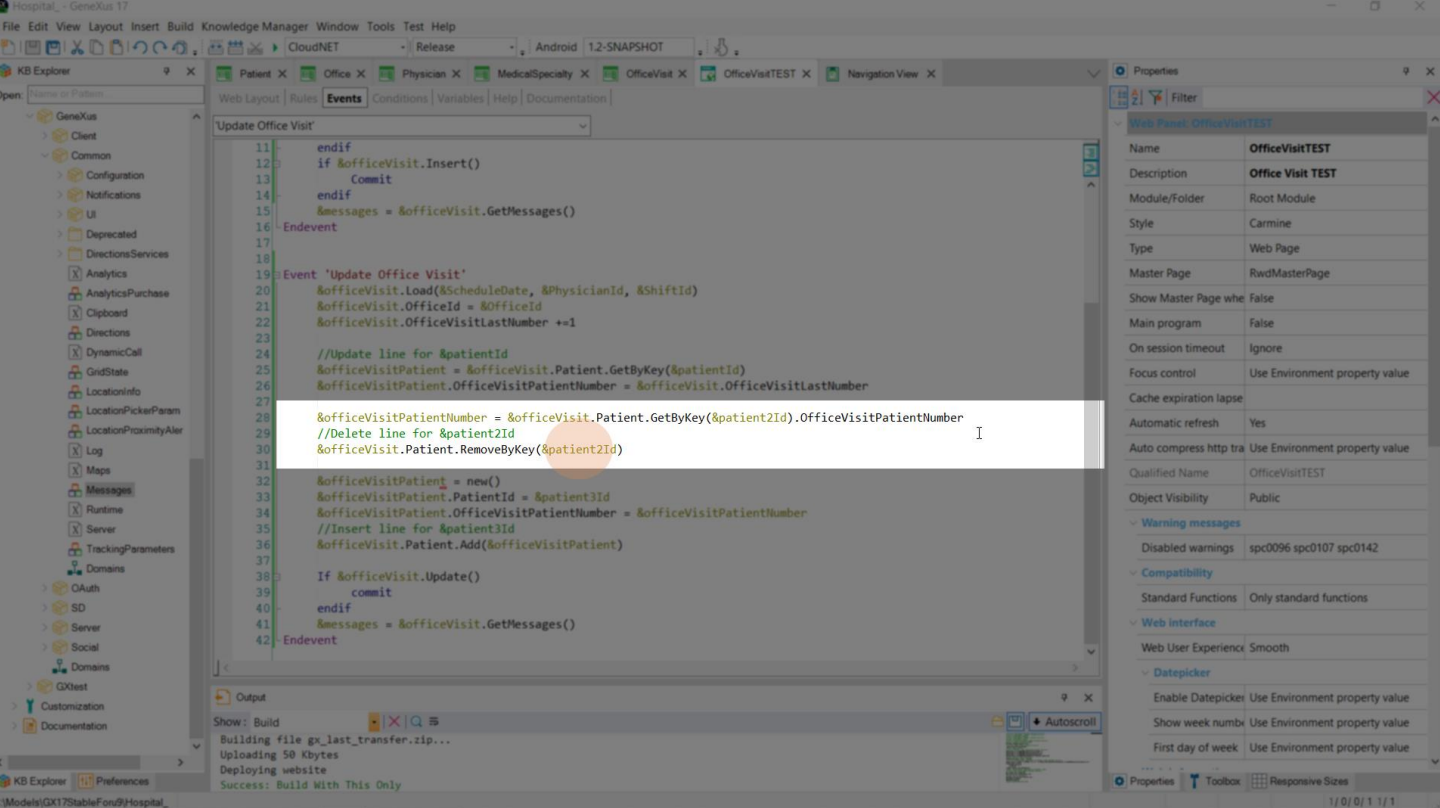
OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

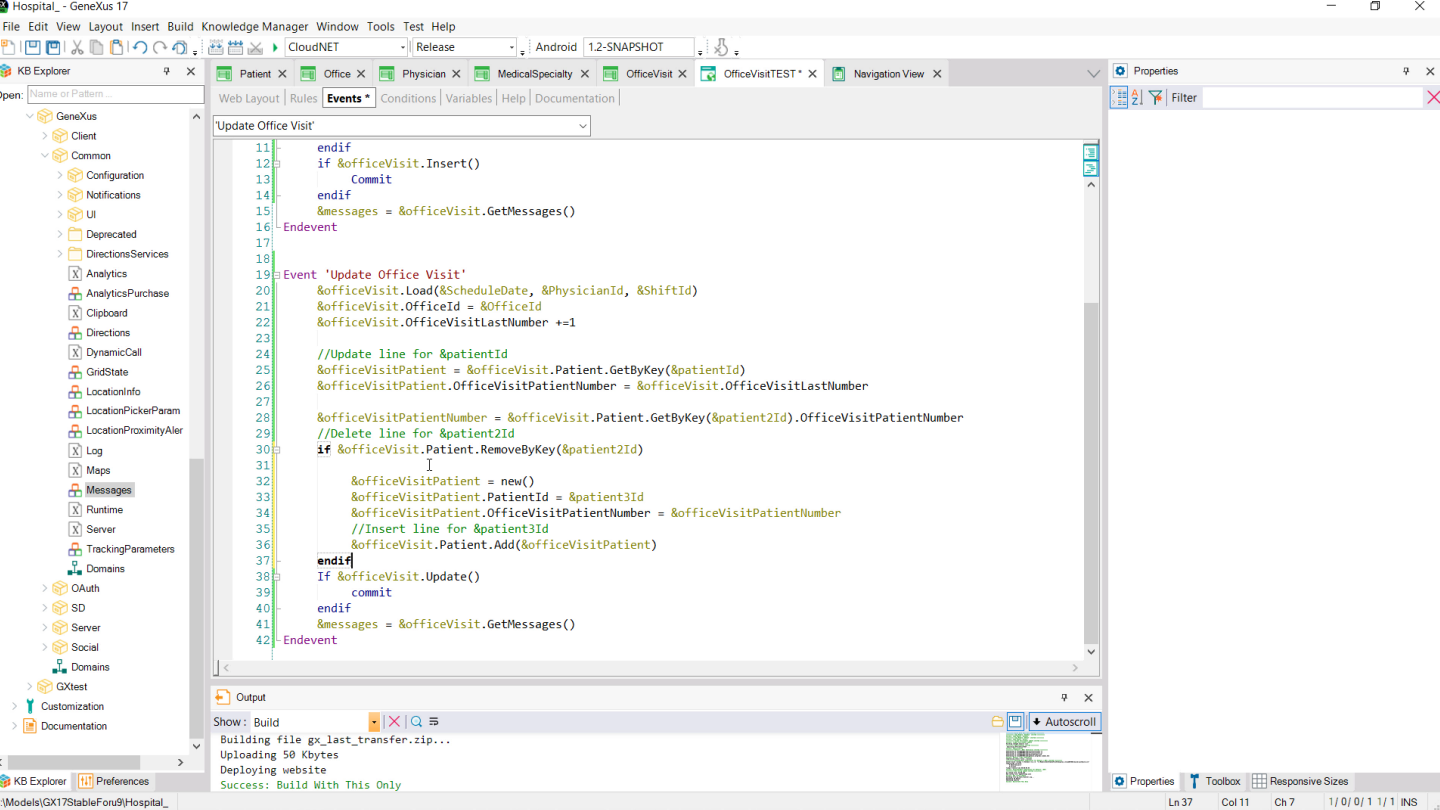
PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

PatientId	4
Patient Name	
PatientOfficeVisitNumber	0

It increased the Last Number by 1. OK. Assigned it to patient 3, fine. But left patient 2 as it was, and added 4, with 0 for the number. Let's see it in GeneXus...



This will give us 0, because it won't find the item with key 25 in the collection. And this Remove operation will not remove the line for this key, because it does not exist. So the new will be performed successfully but with value 0.



We could improve things a bit, asking for the result of the remove operation. And add the line only if it was successful.

GeneXus Developer Menu | Office Visits | Office Visit TEST

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/wwofficevisit.aspx

GeneXus DL Portal

# Hospital Backoffice

by GeneXus

Recents Office Visit — Office Visits

Office Visits

Medical Specialty Name

Office Visit Date

Shift Name

Visit Date	Physician Id	Physician Name	Shift Name	Medical Specialt...	Office Id	Last Number		
11/14/22	1	Doctor 1	Shift morning	Family Medicine	2	0	UPDATE	DELETE
11/14/22	1	Doctor 1	Shift intermediate	Family Medicine	2	0	UPDATE	DELETE
11/14/22	2	Doctor 2	Shift morning	Family Medicine	3	1	UPDATE	DELETE
11/15/22	1	Doctor 1	Shift evening	Family Medicine	2	0	UPDATE	DELETE
11/16/22	2	Doctor 2	Shift morning	Family Medicine	1	0	UPDATE	DELETE
11/17/22	1	Doctor 1	Shift morning	Family Medicine	2	6	UPDATE	DELETE
11/17/22	2	Doctor 2	Shift morning	Family Medicine	3	0	UPDATE	DELETE
11/21/22	1	Doctor 1	Shift morning	Family Medicine	1	1	UPDATE	DELETE

Last notes: here we have the office visit on Thursday 17, for physician 1 on shift 1 in office 2. Let's add an office visit with another physician for the same date and shift but in office 3. If now we want to change the first office visit, that of physician 1, to office 3, which is now busy....

# Hospital Backoffice



Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	1		<input type="button" value="Update Office Visit"/>
Shift Id	1		
Office Id	3I		
Patient Id	3		
patient2 Id	25		
patient3 Id	4		

Id	Type	Description
	Error	Office Visit Date,Shift Id,Office Id already exists

...the update will fail due to a duplicate candidate key.

# Hospital Backoffice



Recents Office Visits — Office Visit — Office Visit TEST

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	1		<input type="button" value="Update Office Visit"/>
Shift Id	1		
Office Id	45		
Patient Id	3		
patient2 Id	25		
patient3 Id	4		

Id	Type	Description
ForeignKeyNotFound	Error	No matching 'Office'.

And if we want to change the office to a nonexistent one, the referential integrity check will fail.

# Hospital Backoffice



Recents Office Visit TESTC...

Schedule Date	11/27/23	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	1		<input type="button" value="Update Office Visit"/>
Shift Id	1		
Office Id	3		
Patient Id	2		
patient2 Id	3		
patient3 Id	4		

Id	Type	Description
PrimaryKeyNotFound	Error	Data with the specified key could not be found.

And of course, if there is no record with this primary key, no matter what values we enter in these variables, when we try to update, we will get this error because the Load was not successful.



Office Visit

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officev... DL Portal

### Office Visit

Visit Date	11/17/22
Physician Id	1
Physician Name	Doctor 1
Shift Id	1
Shift Name	Shift morning
Medical Specialty Id	1
Medical Specialty Name	Family Medicine
Office Id	<input type="text" value=""/>
Last Number	<input type="text" value="6"/>

&officeVisit .Load(#2022-11-17#, 1, 1)

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
Officeld	2
OfficeVisitLastNumber	6
Patient	

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

PatientId	4
Patient Name	Patient 4
PatientOfficeVisitNumber	0

&officeVisit.Officeld

In summary: to update through the transaction, the value of the primary key is sent to it and there the values of the header and line attributes are brought from the database to the screen.

To do it through the BC, the Load method is applied to load the elements of the BC structure. So the attributes on the screen and the lines are like the elements of the BC and the items of the collection.

The user can change those attributes of the header that can be modified, and the developer does the same with the BC through the variable, by accessing the element: BC variable, period, element name (for example, Officeld).

In the transaction, the user can change some data of a line directly by clicking on that line and editing the desired field.

To do this with the BC, you must first access the line by code; that is, the item in the collection. To do so, there is the GetByKey method that can only be applied to the BC element that is a collection of lines. Here it is &officeVisit, period, Patient. Unlike the transaction, where we can see the line and its values, here we have to retrieve it, starting from its identifier. If there is an item with that ID in the collection, then we will perfectly retrieve the memory location of the item (and it will be convenient, surely, to assign it to a variable of the BC data type of the line, to work more comfortably). There we modify the elements of the item/line that we are interested in. It could be the PatientOfficeVisitNumber.

But what if the ID that we specify for the GetByKey does not exist? The &officeVisitPatient variable will not be pointing to the position of the line. If it had not been used before or had been assigned a new, then it will be

empty, and the change made to the PatientOfficeVisitNumber will have no effect on the original BC.

If we want to remove a line from the transaction, we identify it and mark it like this. In the BC we have the RemoveByKey method of the items collection. Again, if we pass an item ID that is not in the collection, it will not be able to do anything on that line that we wanted to delete. Everything will remain unchanged. This method has the advantage of returning True or False based on whether it finds the item.

And if we want to add a new line to the transaction, we do it directly using one of the empty lines provided by the grid.

In the BC, we will have to create the memory space for the item, with a variable of the data type of the lines. Also, by assigning it the new, complete its elements, and use the Add method of the collection of items. That's how we add this memory space to the collection.

So far, either by handling the transaction or the BC variable, we have worked in memory. For all this to impact on the database, the order must be given. In the transaction, it is done with the Confirm button (since it is in Update mode, it will try to update). In the BC, we do it with the Update method (we could also use the Save method, but we are not going to see it here). As a result, the corresponding rules, events and transaction checks will be executed. That is, in the BC all those that do not depend on the interface (and are not qualified to run only on the web, through the environment attributes that we had seen in the previous video).

In short, when working with the BC we have to be more careful because we don't see the existing lines and we can make mistakes in this programming. Then we will discuss a simpler way to insert or modify lines (but not to delete them), which is to handle them through a Data Provider.

Office Visit

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officev... | GeneXus | DL Portal

Medical Specialty Id	1
Medical Specialty Name	Family Medicine
Office Id	2
Last Number	6

**Patient**

Patient Id	Patient Name	Patient Number
X 2	Patient 2	4
X 3	Patient 3	0
X 4	Patient 4	0
0		0
0		0
0		0
0		0
0		0

(New row)

&amp;officeVisit

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

&amp;officeVisitPatient =

&amp;officeVisit.Patient .getByKey(3)

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

PatientId	4
Patient Name	Patient 4
PatientOfficeVisitNumber	0

In the transaction, the user can modify some data of a line directly by clicking on that line and editing the desired field. To do this with the BC, you must first access the line by code, that is, the item in the collection. To do so, there is the `GetByKey` method that can only be applied to the BC element that is a collection of lines. Here it is `&officeVisit`, period, `Patient`. Unlike the transaction, where we can see the line and its values, here we have to retrieve it, starting from its identifier. If there is an item with that ID in the collection, then we will perfectly retrieve the memory location of the item (and it will be convenient, surely, to assign it to a variable of the BC data type of the line, to work more comfortably). There we modify the elements of the item/line that we are interested in. It could be the `PatientOfficeVisitNumber`.

But what if the ID that we specify for the `GetByKey` does not exist? The `&officeVisitPatient` variable will not be pointing to the position of the line. If it had not been used before or had been assigned a new, then it will be empty, and the change made to the `PatientOfficeVisitNumber` will have no effect on the original BC.

Office Visit

apps5.genexus.com/ld9df182c70350001219afeaceeb30e8e2/officev...

GeneXus DL Portal

Medical Specialty Id 1

Medical Specialty Name Family Medicine

Office Id 2

Last Number 6

**Patient**

Patient Id	Patient Name	Patient Number
×	2 Patient 2	4
×	3 Patient 3	6
+	4 Patient 4	0
I	0	0
	0	0
	0	0
	0	0
	0	0

[New row]

&amp;officeVisit

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

&amp;officeVisit.Patient.RemoveByKey(4)

<del>PatientId</del>	<del>4</del>
<del>Patient Name</del>	<del>Patient 4</del>
<del>PatientOfficeVisitNumber</del>	<del>0</del>

If we want to remove a line from the transaction, we identify it and mark it like this. In the BC we have the RemoveByKey method of the items collection. Again, if we pass an item ID that is not in the collection, it will not be able to do anything on that line that we wanted to delete. Everything will remain unchanged. This method has the advantage of returning True or False based on whether it finds the item.

Office Visit

Medical Specialty Name: Family Medicine

Office Id: 2

Last Number: 6

**Patient**

Patient Id	Patient Name	Patient Number
×	2 Patient 2	4
×	3 Patient 3	6
+	4 Patient 4	0
	0	0
	0	0
	0	0
	0	0

[New row]

CONFIRM CANCEL

&amp;officeVisit

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

&amp;officeVisit.Patient.Add(&amp;officeVisitPatient)

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

&amp;officeVisitPatient

PatientId	
Patient Name	
PatientOfficeVisitNumber	

PatientId	4
Patient Name	Patient 4
PatientOfficeVisitNumber	0

And if we want to add a new line to the transaction, we do it directly by using one of the empty lines provided by the grid. In the BC, we will have to create the memory space for the item, with a variable of the data type of the lines. Also, by assigning it the new, complete its elements, and use the Add method of the collection of items. That's how we add this memory space to the collection.

Office Visit

Medical Specialty Name: Family Medicine

Office Id: 2

Last Number: 6

**Patient**

Patient Id	Patient Name	Patient Number
2	Patient 2	4
3	Patient 3	6
4	Patient 4	0
0		0
0		0
0		0
0		0

[New row]

Confirm

CANCEL

&amp;officeVisit

OfficeVisitDate	#2022-11-17#
PhysicianId	1
PhysicianName	Doctor 1
ShiftId	1
ShiftName	Shift morning
MedicalSpecialtyId	1
MedicalSpecialtyName	Family Medicine
OfficeId	2
OfficeVisitLastNumber	6
Patient	

PatientId	2
Patient Name	Patient 2
PatientOfficeVisitNumber	4

PatientId	3
Patient Name	Patient 3
PatientOfficeVisitNumber	6

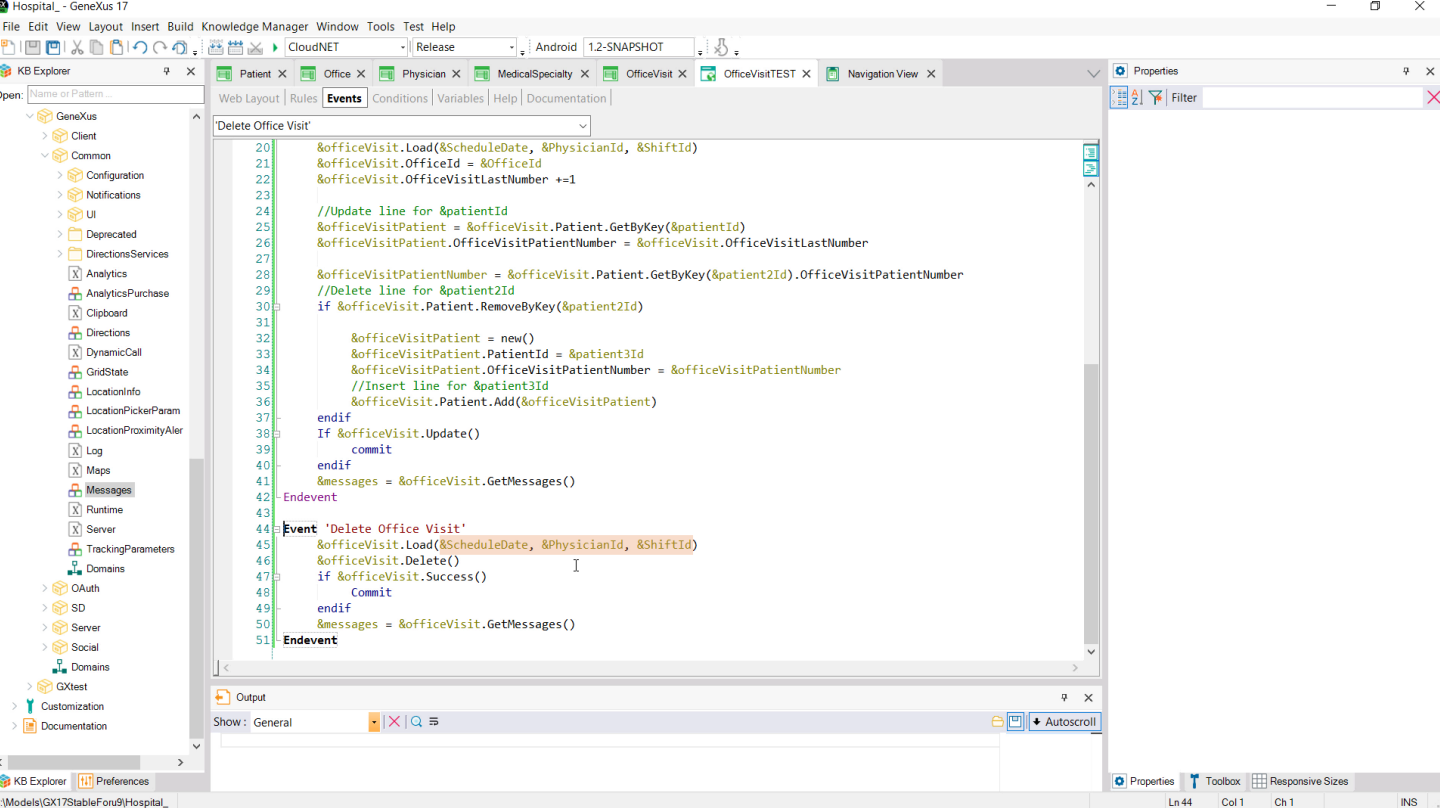
PatientId	4
Patient Name	Patient 4
PatientOfficeVisitNumber	0

&amp;officeVisit.Update()

PatientId	
Patient Name	
PatientOfficeVisitNumber	

So far, either by handling the transaction or the BC variable, we have worked in memory. For all this to impact the database, the order must be given. In the transaction, it is done with the Confirm button (since it is in Update mode, it will try to update). In the BC, we do it with the Update method (we could also use the Save method, but we are not going to see it here). As a result, the corresponding rules, events and transaction checks will be executed. That is, in the BC all those that do not depend on the interface (and are not qualified to run only on the web, through the environment attributes that we had seen in the previous video).

In short, when working with the BC we have to be more careful because we don't see the existing lines and we can make mistakes in this programming. Then we will discuss a simpler way to insert or modify lines (but not to delete them), which is to handle them through a Data Provider.



Finally, we want to delete, so we add the button and in the associated event: first we load the BC variable by primary key—the three variables are displayed here—, and then we apply the Delete method to the BC. For now, this method does not return True or False as the Insert and Update do. So to know the result of the operation we will have to execute the Success method (or its opposite, Fail).

Recents Office Visit TEST

Schedule Date	11/17/22	<input type="button" value="29"/>	<input type="button" value="Insert Office Visit"/>
Physician Id	2		<input type="button" value="Update Office Visit"/>
Shift Id	1		<input type="button" value="Delete Office Visit"/>
Office Id	0		
Patient Id	0		
patient2 Id	0		
patient3 Id	0		

Id	Type	Description
SuccessfullyDeleted	Warning	Data has been successfully deleted.

Let's first test the deletion of an office visit that we know exists. This one. We enter the key values. It deleted it without any problems.

If from another table there were a record referencing the one we wanted to delete, then the deletion would have failed due to referential integrity. But this is not the case. In our application at the moment there is no subordinate table to this one, that is to say, that references it.



# Hospital Backoffice

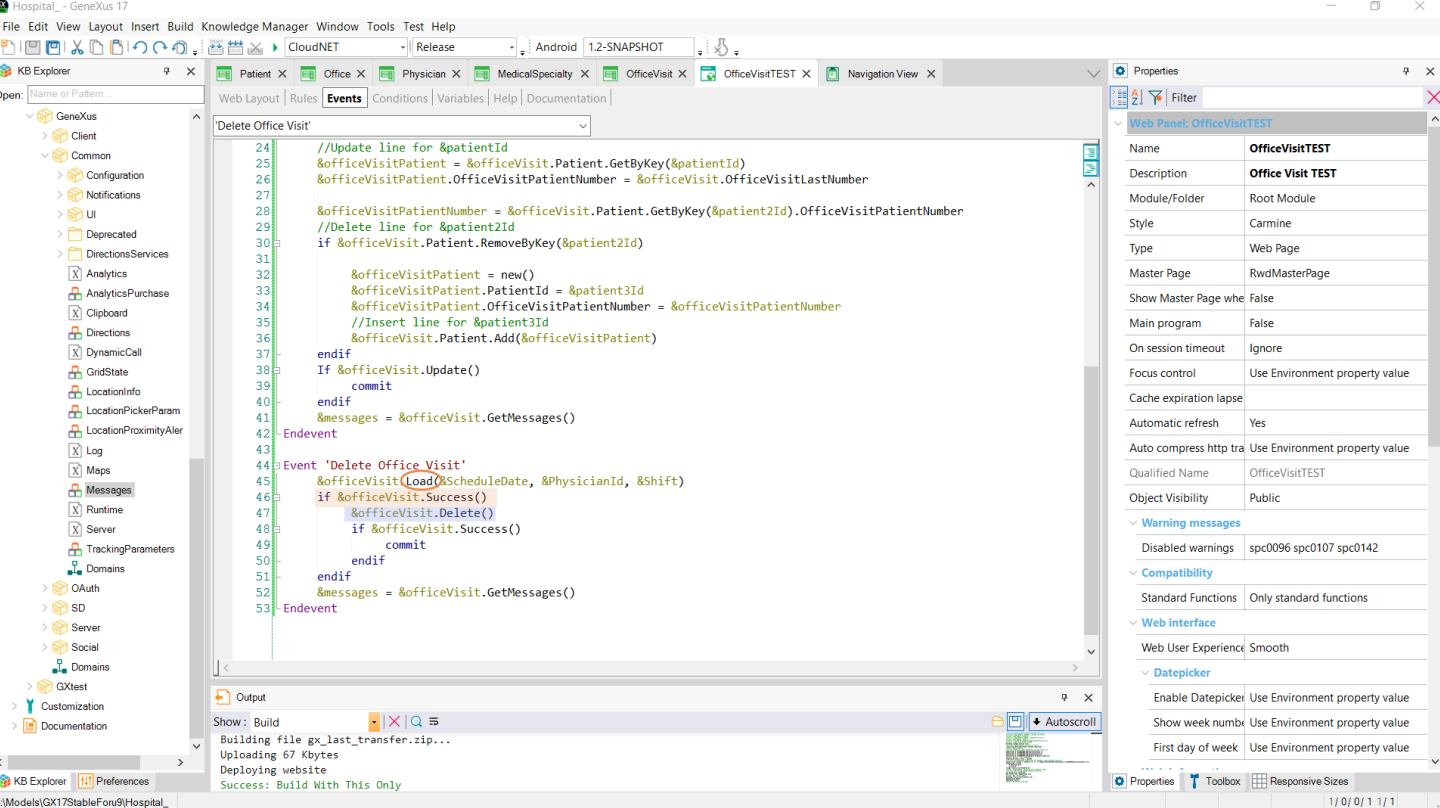


Recents Office Visit TEST

Schedule Date	12/30/22 <span>29</span>	Insert Office Visit
Physician Id	2	Update Office Visit
Shift Id	1	Delete Office Visit
Office Id	0	
Patient Id	0	
patient2 Id	0	
patient3 Id	0	

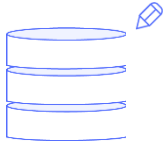
Id	Type	Description
CandidateKeyNotFound	Error	Please 'Get' data before trying to update it.

And if now we enter a nonexistent office date... it throws this error that is not very clear because it mentions an update, but actually the Get indicates that it failed because it could not perform the Load.



We could try the Delete only if the Load was successful.

Of course, it could happen that the Load is successful (because the record exists) but the Delete fails, for example, because of referential integrity, as we've just said. Or because there is an error rule conditioned to be fired on Delete and with another condition that is met.



Insert, Update, Delete

Transaction

Business Component

We have now finished reviewing the relationship between updating through the transaction and updating by code through the Business Component, with examples of the three modes.

# GeneXus™

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)  
[training.genexus.com/certifications](http://training.genexus.com/certifications)

Checks	Error Id	Error Description
FK: PhysicianId	ForeignKeyNotFound	No matching 'Physician'
<code>error("Physician " + PhysicianId.ToString() +" already has 2 shifts assigned for the date: " + OfficeVisitDate.ToString(), PhysicianNotAvailable) if &amp;shifts &gt;= 2 and Insert;</code>	PhysicianNotAvailable	Physician... already has 2 shifts...
FK: ShiftId	ForeignKeyNotFound	No matching 'Shift'
PK: {OfficeVisitDate, PhysicianId, ShiftId}	DuplicatePrimaryKey	Record already exists
FK: OfficeId	ForeignKeyNotFound	No matching 'Office'
CK: {OfficeVisitDate, ShiftId, OfficeId}		Office Visit Date, Shift Id, Office Id already exists
FK: PatientId	ForeignKeyNotFound	No matching 'Patient'
PK: {OfficeVisitDate, PhysicianId, ShiftId, PatientId}	DuplicatePrimaryKey	Record already exists
<code>error("There is a pending office visit scheduled for this patient for the same specialty", PatientAlreadyScheduledForMedicalSpecialty) if not &amp;IsOk and Insert Level PatientId;</code>	PatientAlreadyScheduledForMedicalSpecialty	There is a pending office visit scheduled...