

Practical exercises

GeneXus Advanced Course

GeneXusTM 18

December 2022

© GeneXus. All rights reserved. This document may not be reproduced by any means without the express permission of GeneXusTM. The information contained herein is for personal use only.

Registered Trademark

GeneXus is a trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners

CONTENTS

CONTENTS.....2

PROBLEM.....5

NEW PROJECT, NEW KNOWLEDGE BASE.....5

DATA MODEL: TRANSACTIONS AND THEIR STARTING DESIGN6

ENTER DATA AT RUNTIME.....8

RULES.....9

Client Side Validation Behaviour [suggested].....11

“Schedule” transaction.....11

MORE RULES TO DEFINE BEHAVIOR12

Optional.....16

RULE TRIGGERING EVENTS IN TRANSACTIONS 17

Rule Triggering Events in Transactions (continued).....17

Rules and events at line level19

Optional.....20

Evaluation tree for triggering rules and formulas21

New requirement:24

New requirement:26

optional (controls on lines in update mode).....27

optional: Mental exercise on evaluation tree and events28

RULE TRIGGERING EVENTS IN TRANSACTIONS (FINAL)29

Allowing null values in part of a compound foreign key31

Consequences of adding an attribute to a table that already has data [SUGGESTED].31

Parallel Transactions [SUGGESTED].	32
SUBTYPES.....	33
Subtypes: Multiple References and Specialization	33
More Use Cases of Subtypes	35
FORMULAS.....	35
Formulas VERSUS Rules	35
Horizontal formulas (optional)	36
Aggregate formulas	37
Dynamic Transactions	39
Events in Transactions	40
More about nested For each commands. Cases and navigation.....	40
Subroutines	42
Unique Clause	43
Data Selector.....	45
PAGING CLAUSES.....	47
Paging in For each, Pattern, and grid (OPTIONAL)	47
BUSINESS COMPONENT.....	48
Updating employees in charge.....	48
Working with Movies and their Actors.....	48
UPDATING THE DATABASE WITH A PROCEDURE.....	49
WEB SCREENS: WEB PANELS.....	49
Grid Variables and Actions in the Work With Pattern (OPTIONAL)	50
Creating interactive screens.....	50

Multiple Grids.....	51
Grid without base table.....	53
Loading Collections.....	54
Saving context data (OPTIONAL)	54
Special screen for ticket sales.....	54
MULTI-CHANNEL SCREENS: PANELS	57
Testing the application on smart devices	59
PROGRESSIVE WEB APPLICATIONS [SUGGESTED]	60
WEB SERVICES [SUGGESTED]	64

PROBLEM

A company that manages a Cinema hires you to develop a system to store, handle, and display the information it works with. Suppose that the system is made up of these modules:

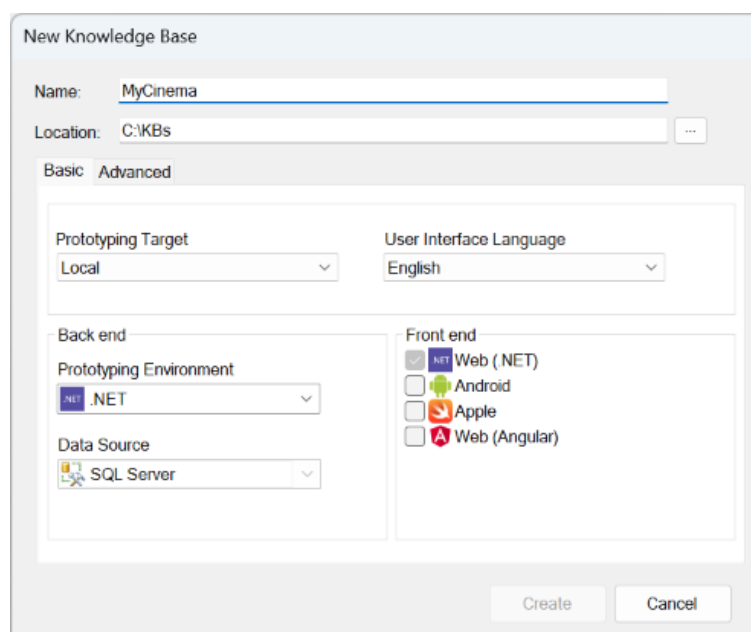
- **Back office:** part of the application that will run on a web server, so that the company's employees can handle the information from any location with an Internet connection.
- **Front end:** part of the application that will run on a web server, so that the information displayed to customers has a pleasing aesthetic.

NEW PROJECT, NEW KNOWLEDGE BASE

Open GeneXus and create a knowledge base called *MyCinema* to start developing the application.

Tips:

- Select Net Core (.Net) as the development environment. Make sure everything you need is installed (including SQL Server). If you're using GeneXus Trial, the generation environment with Net Core and SQL Server is already predefined, prototyping in the Amazon cloud.
- Do not create the knowledge base in "My Documents" folder or any other folder under "Documents and Settings" because these folders have special permissions granted by Windows.



NOTE: This hands-on document contains two types of exercises:

1. Those to be done in the GeneXus IDE.
2. Those that can be skipped in the application, but for which a space is provided in the document so that the student can write the answers. (These exercises can also be done in the IDE.)

DATA MODEL: TRANSACTIONS AND THEIR STARTING DESIGN

The cinema manages the schedules of movies shown in various theaters throughout the city. People will be able to purchase tickets for a specific movie showing in one of the theaters, according to the schedule.

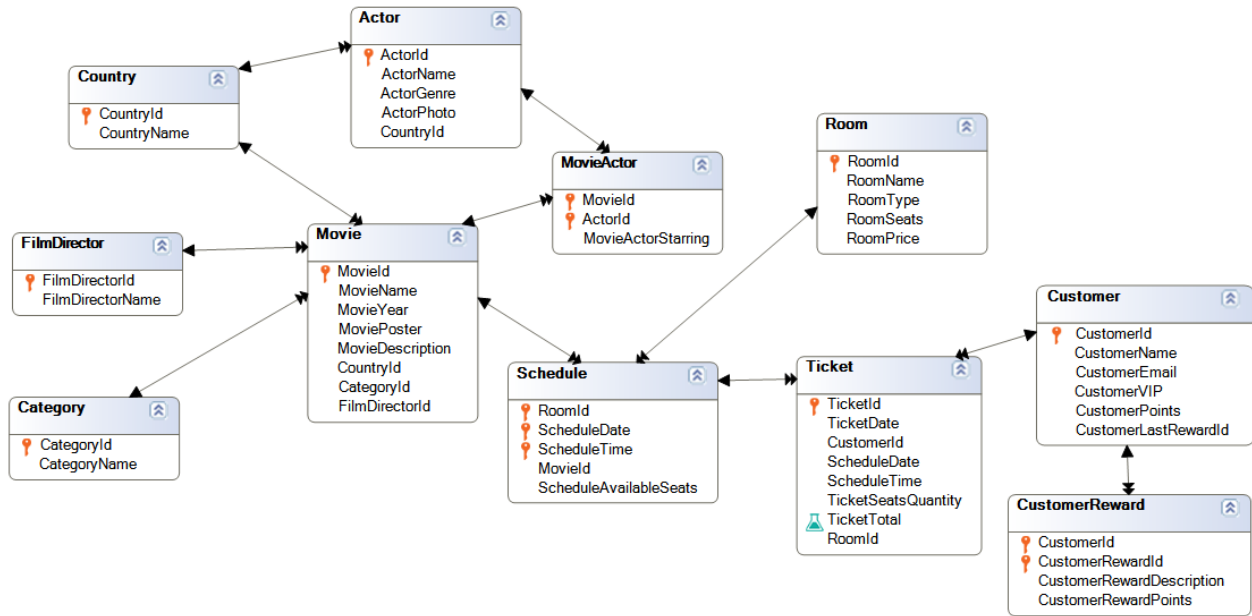
You need to manage information about the movies, their director, the main actors, the category of the movie, their country, etc.

Also, about the cinema's customers, who can obtain benefits according to how frequently they use the services.

To initialize the KB with basic transactions and start modeling this reality, you must import the attached MyCinema.xpz file into the KB. Under the Trn folder you will find, among other things, these transactions:

- Actor
- Category
- Country
- Customer
- FilmDirector
- Movie
- Room
- Schedule
- Ticket

Carefully analyze their structure and infer the relationships between the entities of the reality they model. To do so, we recommend building a transaction diagram. You can also build a table diagram (remember that with View > Tables you can get the Tables node in the KB Explorer with all the tables together, to drag them all at once to the Diagram type object). If you don't want to spend time doing it, we have already done the one with tables (Diagram1 object).



The Movie transaction records the relevant information of a movie: its name, year, poster, description or overview, the country of the movie, its category (comedy, drama, thriller, etc.), and its director, as well as the list of actors (the Boolean attribute MovieActorStarring indicates whether the actor is a leading or supporting actor).

Remember that the actors' countries may be different from those of the movie.

On the other hand, we have the rooms (Room) in which the movies will be screened. For each room, in addition to a name and a type (if it is a regular, VIP, or 3D room), the number of available seats is recorded, as well as the price of tickets for that room. Here, evidently, it is being considered that the prices of the movie screenings depend on the room.

In addition, we have the "showing" or "screening" itself, which is given by the Schedule, where for each room, day, and time of screening, the movie that will be screened and the number of seats still available (not sold) for that screening are recorded. Note the identifier composed of room, date, and time of screening (they take place at 4 fixed times, and that is why ScheduleTime is of enumerated type). According to this modeling, the same movie can be shown in two different rooms on the same day at the same time.

When a person buys tickets for a show, a Ticket is recorded; in it, the data of the show and the number of seats purchased are stored, and the total price of the ticket is calculated. If the person is registered as a customer, he/she is associated with the ticket.

In addition, the customer generates points for ticket purchases, which can later be redeemed for specific prizes, which are also recorded.

Note that unique indexes have been created on most of the tables to establish candidate keys by name.

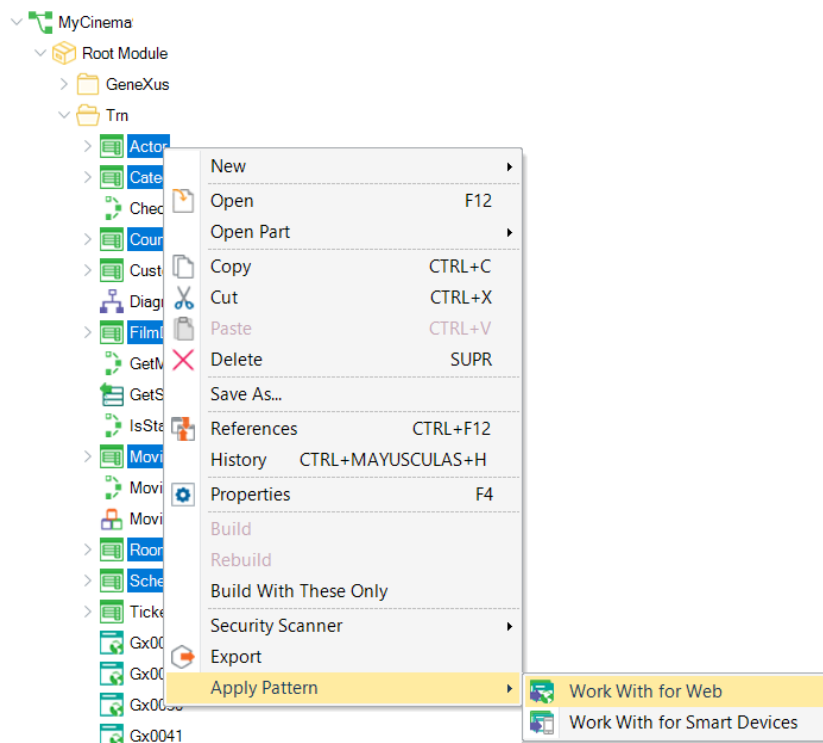
ENTER DATA AT RUNTIME

Select Run Developer Menu. Click on Create in the Impact Analysis list, and while GeneXus creates the tables, specifies and generates, note the following, taking into account that the Id domain is autoincremented:

1. What does the warning about the CustomerReward and Customer tables in the Impact Analysis list mean? Remember that only attributes that are themselves PKs of the table can be autoincremented. If they are part of a PK, or not a PK at all, they cannot be autoincremented. Their AutoNumber property is simply ignored.
2. Note that for some tables (Room, FilmDirector, CustomerReward, Customer, Actor, MovieActor, Movie, Country, and Category) it is reported that although the table “is new,” it will be initialized with a Data Provider. If you look below these transactions, you will find those Data Providers. They will be used to automatically load data into the tables in the next run. We will address this matter when we study this topic later on. For now, just note their existence.

Navigate through the loaded data until you are familiar with the transactions. The Schedule and Ticket transactions do not have any preloaded data.

Apply the Work With for Web pattern to all transactions except Ticket and Customer. Note: Remember that you can select all of them from the KB Explorer and right-click:



Run, and while you wait for GeneXus to end, continue reading.

RULES

Look at the Room transaction and its rules (the ones not generated by the pattern):

```
1 | Msg("Room name is expected")
2 | |   if RoomName.IsEmpty();
3 |
4 | | Error("Room seats must be greater than 0")
5 | |   if Insert and RoomSeats = 0;
6 |
7 | | Msg("Room price not specified")
8 | |   if RoomPrice.IsEmpty();
9 |
```

Do they adequately implement the requirements mentioned below?

- You want to warn the user if the room name is empty, if the room ticket price was left empty, and if the room type was left unspecified (note the data type of the RoomType attribute).
- It must not be possible to insert a room with zero seats.

Test at runtime:

- Leave the name of a previously loaded room empty.
You will see that it allows you to do it but a message is displayed. This is because the rule is not conditioned to a particular mode. Will you be able to save the room with an empty name?
- Try leaving “value (None)” empty in the combo box for RoomType. Note that without having programmed any rule, the message “Field Room Type is out of range” is being displayed and does not allow you to save.
Also, that to the RoomType attribute, of enumerated type, we modified the property of the Control Info group to allow the empty value. Otherwise, you wouldn't even see the “(None)” option in the combo box.

<p>▼ Type Definition</p>	
Supertype	
Based on	RoomType:Domain
Data Type	Numeric
Length	1
Decimals	0
Signed	False
Initial value	
<p>> Validation</p>	
<p>▼ Control Info</p>	
Control Type	Combo Box
Values	Normal:1,VIP:2,3D:3,(None):0
Empty Item	True
Empty Item Text	GX_EmptyItemText
Notify Context Char	False

However, that value is not a valid value of the domain. We put it this way at the attribute level so that it automatically allows choosing empty “(none)” in all Web Panels where a variable based on the attribute is going to be used. But it doesn't make sense at the attribute level in the transaction, because that value is not a valid value for the attribute. If you want GeneXus to allow you to enter an empty value in the attribute—that is, that the user can leave it unspecified—modify the domain by adding it:

Name	Description	Value
Normal	Normal	1
VIP	VIP	2
in3D	3D	3
None	(None)	0

Note that the value 0 will cause the query for “lsempty()” to return True. Then it returns the False value for the Empty Item property of the RoomType attribute. Run the transaction again with this change, leaving the attribute value at (None). Now you can leave the room type unspecified. Give it a try. You should add a **Msg rule** to warn the user that the value was left unspecified, in case it was not what they wanted.

- Add a new room with zero seats. It will not allow you to do so. But, what if you change the value of an existing one to 0? We should not allow it. Modify the rule so that it is **never** possible to leave that value at zero when the transaction is used.

Note: Is it possible to repeat the name of a room? Note that we have created a unique index by RoomName and that's why the duplicate control is being performed.

CLIENT SIDE VALIDATION BEHAVIOR [SUGGESTED]

Before we continue testing, all transactions should prevent the user from continuing to the next field when an error rule is triggered.

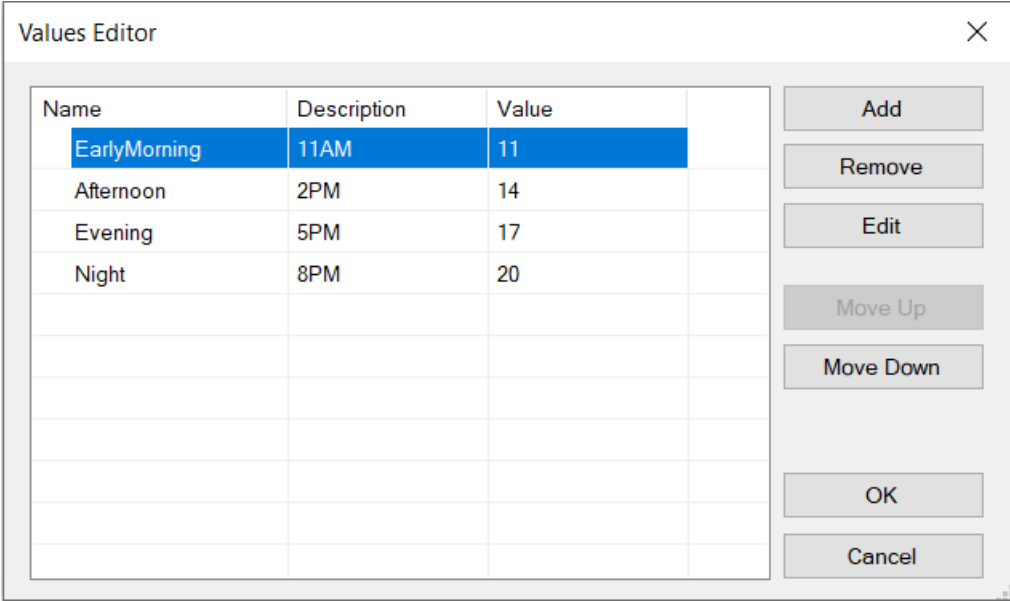
It is also desirable that all rule messages appear at the bottom of the field.

Do what is necessary to obtain this behavior and test again with Room.

Now focus on the following transaction.

“SCHEDULE” TRANSACTION

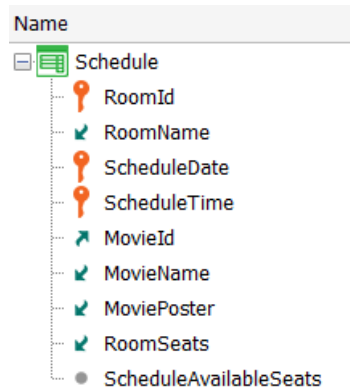
The schedule consists of the screening room (Room), the date, and the time when the movies are going to be shown (only at these times: 11:00 AM, 2:00 PM, 5:00 PM, and 8:00 PM). To record these four times, an enumerated domain, MovieTime, has been created.



Name	Description	Value
EarlyMorning	11AM	11
Afternoon	2PM	14
Evening	5PM	17
Night	8PM	20

Here we have not added a value (None), but we have added the possibility of an empty element in the combo.

Note that we have chosen to use a compound identifier and not an autonumbered one. With this design, can the same movie be shown in two different theaters on the same date and time?



The answer is yes, and this is correct as we are informed that it is part of the requirement.

The ScheduleAvailableSeats attribute will store the number of seats available (not yet sold), so it is initialized with the room capacity (RoomSeats).

Look at the rules declared in the transaction (outside those of the pattern) and try to insert a new screening at runtime, where:

- RoomId does not exist. What message is displayed? Enter an existing one.
- The screening date is before today. Enter a later one.
- Leave the time unspecified. What happens is the same as we saw in the case of RoomType. Here we need the user to enter a valid value because the time is part of the key. Enter one of the four.
- Leave the MovieId unspecified or assign a non-existent one. Enter a valid one.
- Why do RoomSeats and AvailableSeats have the same value and are readonly? When did you see the value of RoomSeats and AvailableSeats? If you missed it, confirm and start over by entering another Schedule. Pay attention to these fields when you exit RoomId.

MORE RULES TO DEFINE BEHAVIOR

Look at the Ticket transaction corresponding to the tickets sold to a person for a screening. If the person is registered in the system as a customer, then he/she will be assigned the ticket; otherwise, the field will be left empty. The date of the ticket will be the date of issue.

For the ticket, the data of the screening (according to the Schedule) and the number of seats being sold to the person are recorded, showing the total price (calculated according to the number of seats and the cost of the screening in that room).

Name	Type	Description	Formula	Nullable
Ticket	Ticket	Ticket		
TicketId	Id	Ticket Id		No
TicketDate	Date	Ticket Date		No
CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
RoomId	Id	Room Id		No
RoomName	Name	Room Name		
RoomType	RoomType	Room Type		
ScheduleDate	Date	Schedule Date		No
ScheduleTime	MovieTime	Schedule Time		No
MovieId	Id	Movie Id		
MovieName	Name	Movie Name		
RoomSeats	Numeric(4.0)	Room Seats		
RoomPrice	Price	Room Price		
TicketSeatsQuantity	Numeric(4.0)	Ticket Seats Quantity		No
TicketTotal	Price	Ticket Total	RoomPrice*TicketSeatsQuantity	

Look at the rules declared and reflect on whether they adequately implement the requirements listed below. If they don't, modify them so that they do:

Ticket

- By default, today's date should be suggested. Do not allow to modify it when inserting data, but allow it in another mode.
- The date of the movie should not be lower than the date of purchase. This only applies in case of an insertion.

Run the application and add a couple of ticket sales, testing the rules. For example:

- Enter a ticket and note at what point the rules involving TicketDate are triggered.
- Edit the newly entered ticket and see if you can now modify its date. Set it to a date later than that of the movie. Does it allow you to continue and even save? Why is the error rule not being triggered?
- Pay attention to the following: Can you modify the customer of a previously entered ticket? And the screening (Schedule)? You can do so, right? They are not readonly fields. After adding the following rules, you will see that you will no longer be able to modify them.

You need to add the following behavior:

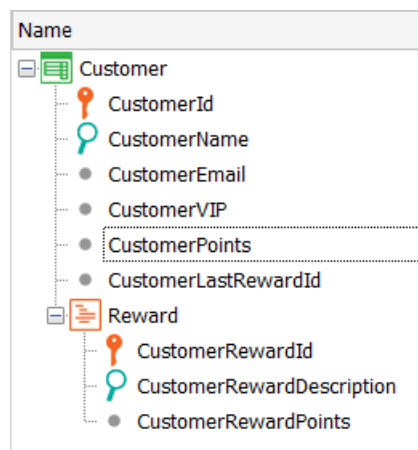
1. Verify that the number of seats being purchased for the screening (Schedule) are available. Otherwise, do not allow the sale.
2. The number of seats available for the screening (ScheduleAvailableSeats) should be reduced according to the number of seats that are being purchased.

3. If the sale is associated with a customer, the points generated by this purchase must be added to the customer's account. Each ticket purchase generates 10% of the ticket total as points.

Add the rules to the Ticket transaction that you think are necessary to achieve this behavior.

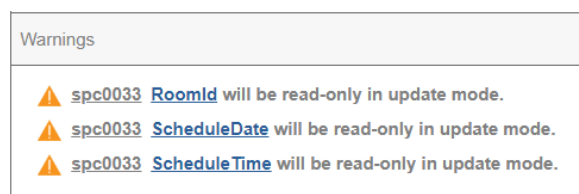
Note: remember that if you need to use an attribute in the transaction rules you must have it declared in the structure, even if it is inferred.

Note that in the Customer transaction you have the CustomerPoints attribute to store the points that the customer has generated so far.



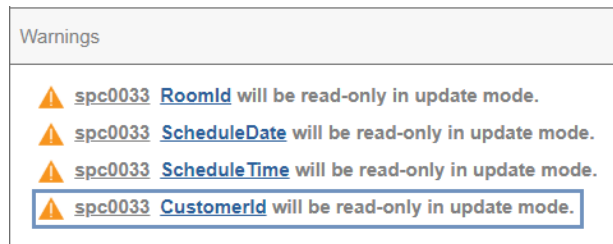
With the changes you have made, test it at runtime:

1. Insert a new Ticket without associating a customer to it and taking a number of seats that doesn't exceed the available ones. Make sure that the number of seats available in the room for the screening (Schedule) is updated.
2. Edit the ticket you have just entered. Can you change the room, date, or time of the screening? Not anymore! Go back to GeneXus and look at these warnings displayed in the navigation list (let's leave CustomerId aside for now):



They appeared when you added the Subtract rule. Try commenting it and get the navigation list again (without generating the object: by selecting its tab, right-clicking on it, and choosing View Navigation). You will see that the warnings are gone. Why is this restriction displayed? Leave the subtract rule uncommented again.

3. Insert a new Ticket and associate it with a customer, and take a number of seats that doesn't exceed the number of seats available. Make sure that the number of available seats in the room and the number of points for the customer have been updated.
4. Edit the previous ticket. Can you change the ticket's customer? In the navigation list, you will also find the warning, which is related to the Add rule. Why do you think that after entering this rule you are no longer allowed to change the customer?



5. For the ticket you are updating, modify the number of seats that were entered. Check that both the number of available seats in the room and the customer's points are updated.
6. Edit or insert a ticket and try to take more seats than are available. It should not allow you to save.
7. Delete a ticket with a customer and notice that the customer's points are restored.

Does the transaction structure look like this?

Name	Type	Description	Formula	Nullable
Ticket	Ticket	Ticket		
TicketId	Id	Ticket Id		No
TicketDate	Date	Ticket Date		No
CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
CustomerPoints	Numeric(4,0)	Customer Points		
RoomId	Id	Room Id		No
RoomName	Name	Room Name		
RoomType	RoomType	Room Type		
ScheduleDate	Date	Schedule Date		No
ScheduleTime	MovieTime	Schedule Time		No
MovieId	Id	Movie Id		
MovieName	Name	Movie Name		
RoomSeats	Numeric(4,0)	Room Seats		
RoomPrice	Price	Room Price		
ScheduleAvailableSeats	Numeric(4,0)	Schedule Available Seats		
TicketSeatsQuantity	Numeric(4,0)	Ticket Seats Quantity		No
TicketTotal	Price	Ticket Total	RoomPrice*TicketSeatsQuantity	
TicketPoints	Numeric(4,0)	Ticket Points	TicketTotal*.10	

The room type cannot be modified from the Ticket transaction because it is inferred. Can you make it possible for the user to modify it from this same screen, without going to the Room transaction? Give it a try.

Since {RoomId, ScheduleDate, ScheduleTime} together form a foreign key for the Schedule table, the integrity check on that table is not done before the user enters the last value, and the message may be confusing for the user.

If instead of this message:

Room Id	6	↑
Room Name	Room 6	
Room Type	VIP	
Schedule Date	05/28/21	29
Schedule Time	8PM	↑
Movie Id	0	

No matching 'Schedule'.

...we want to display this one:

Room Id	6	↑
Room Name	Room 6	
Room Type	VIP	
Schedule Date	05/28/21	29
Schedule Time	8PM	↑
Movie Id	0	

No movie for Room: 6, Day:05/28/21, Time: 8PM

How do you accomplish this? If you need to see the exact syntax of the rule when the foreign key is a compound one, as in this case, look it up at wiki.genexus.com.

You can easily build the string for the output as follows:

```
"No movie for Room:" + RoomId.ToString() + ", Day:" + ScheduleDate.ToFormattedString()
+ ", Time: " + ScheduleTime.EnumerationDescription()
```

Run and test the rules you have added.

OPTIONAL

On the other hand, for each customer, the rewards that he/she can redeem for points are recorded:

Name	Type	Description	Formula	Nullable
Customer	Customer	Customer		
CustomerId	Id	Customer Id		No
CustomerName	Name	Customer Name		No
CustomerEmail	Email, GeneXus	Customer Email		No
CustomerVIP	Boolean	Customer VIP		No
CustomerPoints	Numeric(4.0)	Customer Points		No
CustomerLastRewardId	Id	Customer Last Reward Id		No
Reward	Reward	Reward		
CustomerRewardId	Id	Customer Reward Id		No
CustomerRewardDescription	Numeric(4.0)	Customer Reward Description		No
CustomerRewardPoints	Numeric(4.0)	Customer Reward Points		No

These rewards belong exclusively to the customer and that is why they were modeled as a weak entity. For each reward, the number of points it is worth is recorded. Therefore, when a customer wants to redeem some of his/her generated points for a reward, the customer's points (CustomerPoints) must be updated when the reward is entered as a line of the transaction.

Of course, a reward should not be allowed if the customer does not have the points it is worth.

Implement this behavior and test it at runtime.

RULE TRIGGERING EVENTS IN TRANSACTIONS

Before starting these exercises, we recommend that you draw an outline on a piece of paper of the order of rule triggering events in transactions that you have learned so far—sometimes they are also referred to as “triggering moments.” Starting from this clear outline will be very useful for you to make the most of the exercises. We offer exercises linked to each of the three videos dealing with this topic. You can watch each video in order and do the related exercises, or you can watch all three and then do the entire practice block.

We recommend doing all the tests on the Movie transaction with the movie “Gone with the Wind,” by director Victor Fleming, from the United States, of Drama type and actors “Clark Gable,” “Olivia de Havilland,” and “Vivien Leigh.” We also recommend setting the MovieDescription attribute of the instance file of the Work With Web pattern as hidden.

RULE TRIGGERING EVENTS IN TRANSACTIONS (CONTINUED)

Let's suppose that we have a Web Service to which we can send the name of a movie and it returns its year. To make it simpler (because we haven't seen anything about Web Services yet), we will use the GetMovieYear procedure that comes with the xpz instead.

What will be the Movie transaction's behavior if we add the following rule?

```
MovieYear = GetMovieYear(MovieName) if Insert;
```

Before testing it at runtime, think about the time when it should be triggered. Clearly, it will be only when a new movie is being inserted, and not in any other mode.

1. When the transaction is opened in Insert mode and before going through the MovieName field?
2. When leaving the MovieName field and entering the MovieYear field?
3. After leaving the MovieYear field?
4. When you confirm?

Before testing, move MoviePoster so that there is another attribute between MovieName and MovieYear.

Now try inserting the movie “Gone with the Wind” (written exactly like this). Look for the poster image in the Movies folder provided with the material. After inserting it, rename it and see how the date remains unchanged.

Now change the previous rule, conditioning it in this other way:

```
MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;
```

When will it be triggered?

Try inserting the movie “Gone with the Wind” again, leaving the MovieYear field empty, and note that the rule will not be triggered until you confirm. Why?

Delete the movie and insert it again, but now leave any value other than 1939 in the MovieYear field. Is it triggered?

What will be the difference between conditioning the rule as you had it and doing so according to option 2 below?

1. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;`
2. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() and Insert on AfterValidate;`

Answer: the AfterValidate event comes right before BeforeInsert / BeforeUpdate / BeforeDelete, but no action occurs in between them, so they are almost equivalent.

Now try changing the triggering event of the rule to AfterInsert, the moment immediately after BeforeInsert, and try inserting the movie “Gone with the Wind” again, leaving the MovieYear value empty. Does it assign the value 1939 after confirming or does it remain empty?

Correct the conditional rule again: “`if MovieYear.IsEmpty() on BeforeInsert;`”

And delete the movie.

 RULES AND EVENTS AT LINE LEVEL

Note that the xpz also included an `IsStarring` procedure, which uses the `GetStarrings` Data Provider and the `MovieStarringActors` SDT. The idea is to send a movie name and actor name to the procedure and it returns true if that actor is starring in the movie. Again, we would be emulating the use of a web service that gives that information.

Therefore, you could add the following rule to the Movie transaction:

```
MovieActorStarring = IsStarring(MovieName, ActorName) if Insert;
```

Enter the movie “Gone with the Wind” again, with the actors “Clark Gable,” “Olivia de Havilland,” and “Vivien Leigh.”

The rule will be triggered in the client (in addition to being triggered later in the server as well). When is it triggered in the client? Think about the precise moment for the above-mentioned execution when you enter the three actors. How many times will it have been triggered before the user presses Confirm? And after Confirm, when will they be triggered in the server?

What if instead of directly assigning a value to the `MovieActorStarring` attribute from the procedure, we want to let the user enter the value, but check against the procedure and ask if they are sure they want to leave that value, in case it doesn't match? That is, something like this:

```
msg("Are you sure about starring?") if MovieActorStarring <> IsStarring(MovieName, ActorName);
```

Another case would be that if the user has set the field to false, then it is overwritten with the value returned by the procedure:

```
MovieActorStarring = IsStarring(MovieName, ActorName) if Insert and MovieActorStarring = False;
```

When entering “Gone with the Wind,” note that the rule is triggered only after you leave the `MovieActorStarring` field (try it with Clark Gable, leaving the field False and with Olivia de Havilland (who is not a leading actor) also set to False).

Note that the check box for Clark Gable will be selected and, if you clear it, it will be selected again; that is, the rule will be triggered again.

If you want the user to enter all the actors on the screen and select the check box as they wish, and have the value assigned according to the procedure only when they confirm for those actors that were set to False, how should you modify the rule? We suggest 4 options:

1. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on BeforeInsert;`
2. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on AfterInsert;`
3. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on AfterLevel Level ActorId;`
4. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on Aftercomplete;`

If now you look at these two rules:

1. `MovieYear = GetMovieYear(MovieName) if MovieYear.IsEmpty() on BeforeInsert;`
2. `MovieActorStarring = IsStarring(MovieName, ActorName) if MovieActorStarring = False on BeforeInsert;`

How does GeneXus know that the first one will be triggered right before inserting the header in the Movie table, and instead the second one will be triggered right before inserting the line in the MovieActor table (for each line)?

OPTIONAL

If you want, you can try the following: make a “Save as” of the Movie transaction, MovieCopy1, removing all the rules added by the pattern and all the events (from the Events tab), and setting the Data Provider property to False. In this way, you will be able to work directly on it (later on, in the Parallel Transactions topic, you will see that it will work on the same table as Movie) by invoking it from the Developer Menu.

Also, add the following rules, and define the variable &ActorLine of Numeric type:

```
&ActorLine = &ActorLine + 1
  on AfterInsert
  Level ActorId;

Msg(&ActorLine.ToString() + " actors. Directed by: "+ FilmDirectorName)
  If Insert
  on AfterLevel
  Level ActorId;
```

The idea is to add 1 after inserting each line in the MovieActor table, and trigger the message after all the lines have been inserted. Test it at runtime.

Then change the Level clause of the first rule, so that it is now the level of FilmDirectorId and see what happens.

EVALUATION TREE FOR TRIGGERING RULES AND FORMULAS

In the Movie transaction, we had already wondered when this rule was going to be triggered:

```
MovieYear = GetMovieYear(MovieName) if Insert;
```

If in the web layout the controls are in this order:

And if we think that each rule will be triggered “as soon as possible,” depending on the information needed to do so, clearly the assignment to MovieYear will depend on the transaction's mode (Insert), and on the value that the user has given to the MovieName attribute.

- If the transaction is executed after receiving the mode by parameter (as when invoked by the Work With) then it will know whether the mode is Insert or not as soon as it is opened. However, since it still needs to know the value of MovieName, it will not be able to trigger the rule until the user leaves that field. If the rule had been written as follows instead:

```
MovieYear = GetMovieYear() if Insert;
```

That is, as soon as the transaction is opened in Insert mode, the rule will be triggered without requiring anything else but the mode; we will see the cursor positioned on the empty MovieName field with a value already loaded in MovieYear.

- If, on the other hand, the transaction is executed without a parm rule (as would be the case of the parallel transaction MovieCopy1 with which we finished the previous exercise), to know the mode we must wait for the user to exit the MovieId field. However, it will not be possible to trigger it yet because we haven't passed through the MovieName field, which is necessary because the rule depends on that value. If the rule has been written as follows:

```
MovieYear = GetMovieYear() if Insert;
```

... as soon as the user leaves MovieId, the transaction will trigger the rule.

And if in that transaction we have the following rules written in this order, when will each one be triggered?

```
MovieYear = GetMovieYear(MovieName) if Insert;  
error("The name is empty")if MovieName.IsEmpty();
```

For either of them, the user must have exited the MovieName field. Therefore, they will be evaluated and triggered in the order in which they are defined. Of course, the second one will be triggered in both Insert mode and in the others.

If we now add the Msg next rule before the other two, when will it be triggered?

```
msg("The movie belongs to the classic film era")  
  if MovieYear < 1950;  
  
MovieYear = GetMovieYear(MovieName)  
  if Insert;  
  
error("The name is empty")  
  if MovieName.IsEmpty();
```

Note that, because it depends on the MovieYear attribute, it will be not only after the second rule is triggered—the one that assigns a value to MovieYear—but also after leaving the MovieYear field, because the user may want to modify the value that was assigned in the second rule. This is where the importance of the order of the rules according to the dependency tree becomes clear, as well as the place of the controls on the screen and the flow of user interactions.

Consider the following flow:

1. Enter in Insert mode.
2. Type something in the MovieName field or leave it empty and exit. The error rule will be evaluated and the assignment rule will also be triggered, so for MovieYear the value returned by the proc will be displayed, even though you have not yet passed through that field.
3. Enter an image in MoviePoster.
4. Go to the MovieYear field. You can leave the value assigned by the rule, or change it. In any case, only when you leave the field, it will be understood that the condition of the msg rule must be evaluated.
5. You decide to return to the MovieName field to change the name of the movie. What will be triggered again when you leave the field?

Look at the navigation list of the transaction. Does it show the order in which the rules are executed? By default, it does not. To see that detailed report, you will have to go to Tools > Options and in the Build option change the value of the Detailed Navigation property to True.

Now the navigation list will be shown again. Here we show that of the MovieCopy1 transaction (only with the previous rules; all the others are commented):

```

Level Movie
READ Movie
WHERE
    Movie MovieId = MovieId
INTO MovieYear MovieName MoviePoster MoviePosterUri MovieDescription CountryId CategoryId FilmDirectorId

READ FilmDirector
WHERE
    FilmDirector FilmDirectorId = Movie FilmDirectorId
INTO FilmDirectorName

READ Category
WHERE
    Category CategoryId = Movie CategoryId
INTO CategoryName

READ Country
WHERE
    Country CountryId = Movie CountryId
INTO CountryName

READ Movie UNIQUE
WHERE
    Movie MovieName = MovieName
MovieYear = GetMovieYear ( MovieName) IF insert;
Error( "The name is empty" ) IF MovieName.isempty()
Msg( "The movie belongs to the classic film era" ) IF MovieYear < 1950

INSERT INTO Movie (MovieYear, MovieName, MoviePoster, MoviePosterUri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
UPDATE Movie (MovieYear, MovieName, MoviePoster, MoviePosterUri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
DELETE FROM Movie

Referential integrity controls on delete:
    • Schedule ( MovieId )
Level MovieActor

```

Finally, what will happen if you add a triggering event to any of the above rules? For example:

```

msg("The movie belongs to the classic film era")
    if MovieYear < 1950;

MovieYear = GetMovieYear(MovieName)
    on BeforeInsert;

error("The name is empty")
    if MovieName.IsEmpty();

```

From now on, the assignment rule is independent of the dependency tree that is triggered as the fields are run through (both interactively in the browser and in the Server, before the header is validated and inserted). The rule will be triggered only in the Server, and it will be long after each field is run through and its rules are triggered in the server. In fact, it will happen when all the fields of the header have been run through and everything related to that header has been validated (referential

integrity; that values are within the allowed ranges, etc.) and right before inserting that header in the Movie table.

Now look at the navigation list of the transaction and compare it with the previous one.

```

Level Movie
READ Movie
WHERE
    Movie.Movied = Movied
INTO MovieYear MovieName MoviePoster MoviePoster.Uri MovieDescription CountryId CategoryId FilmDirectorId

READ FilmDirector
WHERE
    FilmDirector.FilmDirectorId = Movie.FilmDirectorId
INTO FilmDirectorName

READ Category
WHERE
    Category.CategoryId = Movie.CategoryId
INTO CategoryName

READ Country
WHERE
    Country.CountryId = Movie.CountryId
INTO CountryName

READ Movie UNIQUE
WHERE
    Movie.MovieName = MovieName

Error( "The name is empty" ) IF MovieName.isempty()
Msg( "The movie belongs to the classic film era" ) IF MovieYear < 1950
Before Insert Rules
    MovieYear = GetMovieYear ( MovieName);

INSERT INTO Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
UPDATE Movie (MovieYear, MovieName, MoviePoster, MoviePoster.Uri, MovieDescription, CountryId, CategoryId, FilmDirectorId)
DELETE FROM Movie

Referential integrity controls on delete:
    • Schedule ( Movied )
Level MovieActor


```

NEW REQUIREMENT:

You are requested to control that at least two actors are registered when adding a movie; otherwise, the application should not allow saving the movie.

Movie

❗ Select at least two actors

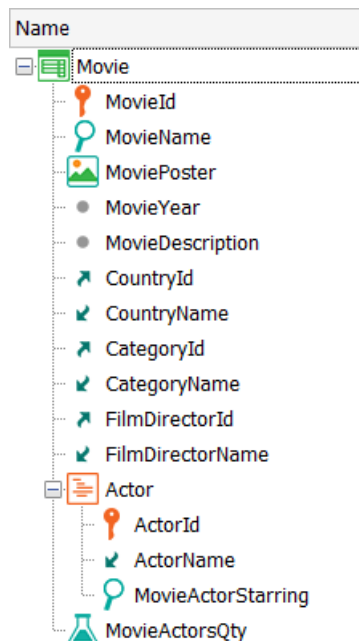
Id	1
Name	<input type="text" value="Avatar 2: The way of water"/>
Year	<input type="text" value="2022"/>
Poster	
Description	Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.

How do you condition the error rule?

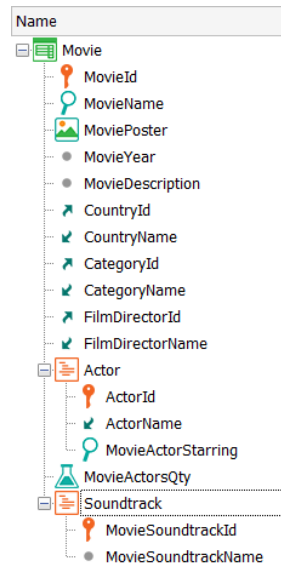
```

Error("Select at least two actors")
  If <condition>
    on <Event>;
    
```

Does the transaction structure look like this now?



Is there any difference in this case between choosing the “AfterLevel Level MovieActorStarring” and the “BeforeComplete” events? If there were another level parallel to Actor and after it—for example, Soundtracks—would there be a difference?



NEW REQUIREMENT:

Every now and then, film festivals are held and the various production companies loan copies of their films to be screened, and it is not uncommon for a production company to send the film at the last minute.

For this reason, we want to give the user the opportunity to enter the data of a movie: header and lines and finally, before Commit, check if a permission to show the movie has been received or not. If not, the movie should not be entered into the system.

In the KB we have the `MovieInStock` procedure that verifies if the permission has arrived at the Cinema. Its code doesn't matter (the one we provide in the KB gives a true or false result at random, for testing purposes only).

And we invoke it in the Movie transaction rules:

```
&IsInStock = MovieInStock(MovieId) If Insert;
```

We do this assuming the procedure will access the `Movie` table in order to retrieve the `MovieName`, for example. That is, we need that at least the record of the header is inserted.

Actually, though, keep in mind that it is preferable to have already done as much as possible, to give more time. That is, we want to delay the check until the last moment.

What triggering event should we add to it? And what else should we do to comply with the requirement that the movie should not be recorded if the permission and the copy have not arrived?

To know what the proc returned (given it was programmed as random) you can modify it to return False in one case and True in another so you can control the behavior.

OPTIONAL (CONTROLS ON LINES IN UPDATE MODE)

Now we change things a bit, and add a MovieStatus attribute of an enumerated domain with the same name (it was created in the xpz), which accepts two values: Pending and Completed. In this way, we will allow entering a movie in more than one instance, and we will only accept it when the user sets the value of that field to Completed.

We will leave the confirmation that the data of the “starring” actors is correct for when the user completes the data entry, setting the MovieStatus to Completed.

The transaction structure will then be as follows:

Name	Type
Movie	Movie
MovieId	Id
MovieName	Name
MoviePoster	Image
MovieYear	Numeric(4.0)
MovieDescription	LongVarChar(1M)
CountryId	Id
CountryName	Name
CategoryId	Id
CategoryName	Name
FilmDirectorId	Id
FilmDirectorName	Name
MovieStatus	MovieStatus
Actor	Actor
ActorId	Id
ActorName	Name
MovieActorStarring	Boolean
MovieActorsQty	Numeric(4.0)

Remove, of course, the rule that assigned a value to the MovieActorStarring attribute, and those that you don't want to be triggered to test this new requirement—for example, messages. You will have to add the new rules (defining the Boolean type for &ok):

```
Default(MovieStatus, MovieStatus.Pending);
```

```
&ok = CheckMovieActors(MovieId)
```

```

    if MovieStatus = MovieStatus.Completed
    on <event>;

    Error("Problem with starrings")
    if not &ok and MovieStatus = MovieStatus.Completed
    on <event>;

```

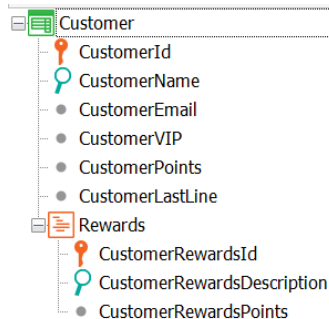
Note that the CheckMovieActors procedure comes in the xpz you imported at the beginning. Which event did you choose?

Before testing, make sure that the MovieStatus attribute has been added to the Web Layout. If it hasn't, go to the main menu, and choose Edit > Apply Default (Web Layout).

Try (you will have to reorganize) entering the movie “Gone with the Wind” again. Remember that the GetStarrings Data Provider determines the actors who play leading roles and those who don't. Experiment with that data, leaving the movie in Pending status first but with some of the starring actors' data wrong. You will see that it allows you to save. Then enter in Update mode, change the status to Completed, and you will see the error message displayed over the MovieStatus field. Change the errors in the actors' starring roles until it allows you to record successfully.

OPTIONAL: MENTAL EXERCISE ON EVALUATION TREE AND EVENTS

Considering the following structure for the Customer transaction:



Below is a list of the rules programmed in that transaction.

Put the correct order of execution in the parenthesis (you don't have to test it):

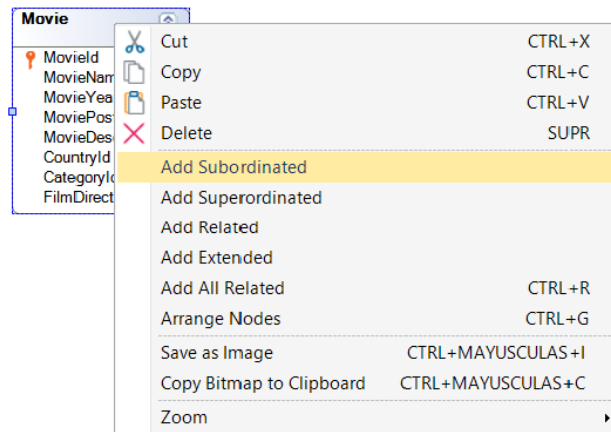
```

( )error('The name is empty') if CustomerName.IsEmpty();
( )msg('The email is empty') if CustomerEmail.IsEmpty();
( )PrintCustomerList() on afterComplete;
( )Subtract(CustomerRewardsPoints, CustomerPoints);
( )error("Not enough points") if CustomerPoints < 0;
( )PrintRewards(CustomerId, CustomerRewardsId) on AfterInsert;
( )serial(CustomerRewardsId, CustomerLastLine, 1);
( )noaccept(CustomerRewardsId);
( )Checkpoints(CustomerId) if update on AfterInsert;

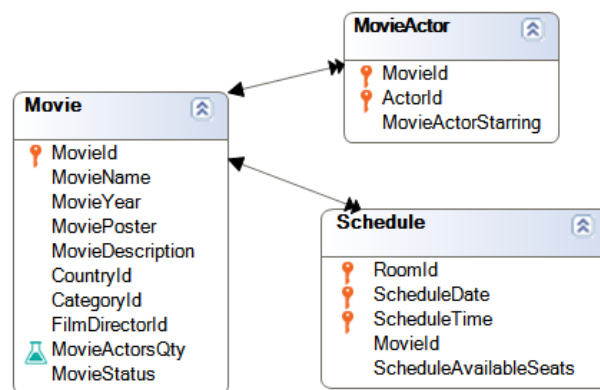
```

RULE TRIGGERING EVENTS IN TRANSACTIONS (FINAL)

Create a new diagram object, and drag the Movie table to it. From the table, right-click > Add Subordinated, to get all the tables that have foreign keys referencing Movie.



There will be only 2:



One is obvious, and corresponds to the second level of the transaction. We only have another from the Schedule transaction.

So, what will happen if from the Movie transaction you try to delete a movie for which there are related records in Schedule?

Give it a try. It will check the referential integrity and will not allow you to delete the movie. But... at what point when running the Movie transaction will this control be performed?

If we want to change this behavior so that if there are related “screenings” (Schedule records) they are deleted immediately before the referential integrity check is made in the Movie transaction, and it doesn't fail and allows us to delete the movie, how do we do it?

Note that you have the DeleteSchedules procedure already implemented, which deletes all the Schedule records of the MovieId received by parameter.

Hint: you will have to invoke it from the rules of the Movie transaction, conditioned to the mode, but... in what trigger event?

```
DeleteSchedules(MovieId)
  If Delete
    on .....;
```

Remember that it must be right before performing the referential integrity check related to the deletion of the header record from the Movie table. What is this moment? Choose it among the following:

1. BeforeValidate
2. AfterValidate
3. BeforeDelete
4. AfterDelete
5. BeforeComplete
6. AfterComplete

At runtime, try to delete a movie that has schedules associated with it (without tickets).

If there are tickets for the Schedule you are trying to delete inside the procedure, you will get a database exception. You will understand why later. What if you try to delete through the Schedule business component (you will have to turn on the Business Component property of the transaction to be able to use Schedule type variable) in the DeleteSchedules procedure?

It won't allow you to delete, but at least it won't crash the program.

```
parm(in:MovieId);

For each Schedule
  &schedule.Load(RoomId, ScheduleDate, ScheduleTime)
  &schedule.Delete()

Endfor
```

Delete the tickets for the movie and try again.

ALLOWING NULL VALUES IN PART OF A COMPOUND FOREIGN KEY

Sometimes it is possible to sell “open” movie tickets so that the customer can watch the movie at the time that suits him/her best; that is to say, tickets can be sold for a certain movie theater today and the customer can go at any time.

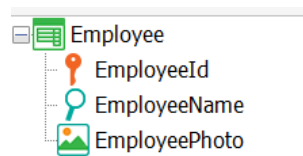
What change should be made in the Ticket transaction to disregard the referential integrity check in the Schedule table between RoomId, ScheduleDate and ScheduleTime?

Make the necessary changes and try to enter a ticket without assigning a room, date, or time to the screening. It should allow you to save without any problems.

Now undo the changes you made to leave it as it was before.

CONSEQUENCES OF ADDING AN ATTRIBUTE TO A TABLE THAT ALREADY HAS DATA [SUGGESTED].

Another requirement is to add an employee catalog. First, add the following transaction to the KB.

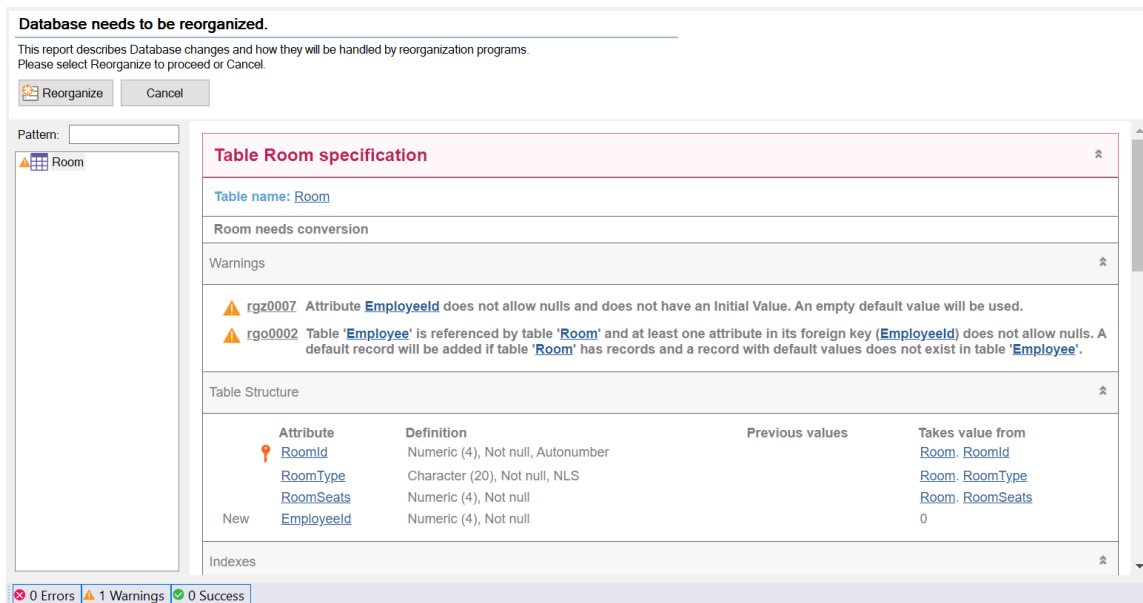


Run the application and add a couple of records for the employee.

We are asked for the following change: for each screening room there must be an employee in charge, and each employee can be in charge of one or more screening rooms. **How would you model this in GeneXus, taking into account the Employee and Room transactions?**

Replicate in GeneXus what you have just analyzed.

The impact analysis shows the following. **What will happen in the Employee table if we press Reorganize?**



If we don't want these warnings to come true, **what can we do?**

PARALLEL TRANSACTIONS [SUGGESTED].

Film companies share certain preview information with all Cinemas. They share: The name of the movie, year, a poster, director, country and its category; the description data and actors are confidential until some time later, so they are not entered into the system until the movie is shown on the Schedule.

To record the initial data of the movie and not see the other fields, a parallel transaction is required where only the available information is recorded.

Make this implementation.

Remember that we had already defined a parallel transaction before: it was MovieCopy1, parallel to Movie. We had done it to have one without the pattern (without parameters and everything that the pattern added to Movie).

SUBTYPES

SUBTYPES: MULTIPLE REFERENCES AND SPECIALIZATION

The following improvements are required in the application:

Remember that a movie has a country, and that it can have 1 or more actors associated with it. It is required to add, also for its actors, information of their country.

What should we change in our transaction structure to store the Country of the Movie and make reference to the Country of the Actor? Think of 3 different designs that meet this requirement. Write for each one their advantages and disadvantages (if any), and choose the most suitable one to implement it in GeneXus.

After the analysis, make the implementation in the application we have been developing.

Keep in mind that with the most suitable implementation you will also have to make changes in the Movie transaction DP, as well as in the MovieCopy1 and MoviePreview transactions.

DESIGN 1:

ADVANTAGES	DRAWBACKS

DESIGN 2:

ADVANTAGES	DRAWBACKS

DESIGN 3:

ADVANTAGES	DRAWBACKS

Cinema Employees can be categorized as follows:

- **Clerical:** they are in charge of selling tickets and assisting the public in general. We are interested in knowing if employees of this type speak any additional language.
- **Operators:** they are in charge of scheduling movies in the different theaters on the days and times they are screened. These employees must be over 22 years of age, so we are interested in knowing their age.
- **Technicians:** they are in charge of maintenance tasks at the Cinema, including repairs of machinery, installations, etc. For this type of employee, we are interested in knowing their technical training relevant to the position.

How can we make this distinction in GeneXus? Implement it.

MORE USE CASES OF SUBTYPES

A transaction is requested that shows the information of the Employees of the cinema and their supervisors. How should it be modeled if we take into account that the supervisor is also an employee?

The Employee transaction should show an error if we try to assign as supervisor an employee who doesn't have that rank. In addition, it shouldn't allow saving the information if the supervisor and the employee are the same person.

How should these rules be programmed?

Make the implementation in GeneXus and write the rules to test the behavior.

```
error("The employee and the Manager can't be the same")
  if _____ = _____;

error("The employee is not a Manager")
  if _____ and _____;
```

The screenshot shows a form titled "Employee" with the following fields and values:

- Id: 1
- Name: Jonh Smith
- Is Manager:
- Employee Id: Billie Robin (with a tooltip error message "The employee is not a Manager")
- Employee Name: Billie Robin
- Is Manager:

At the top of the form are navigation buttons: <<, <, >, >>, and SELECT. At the bottom are buttons: CONFIRM, CANCEL, and DELETE.

FORMULAS

FORMULAS VERSUS RULES

Go back to the control of the customer's points available to redeem for prizes.

Name	Type	Description	Formula	Nullable
Customer	Customer	Customer		
CustomerId	Id	Customer Id		No
CustomerName	Name	Customer Name		No
CustomerEmail	Email, GeneXus	Customer Email		No
CustomerVIP	Boolean	Customer VIP		No
CustomerPoints	Numeric(4.0)	Customer Points		No
CustomerLastRewardId	Id	Customer Last Reward Id		No
Reward	Reward	Reward		
CustomerRewardId	Id	Customer Reward Id		No
CustomerRewardDescription	Numeric(4.0)	Customer Reward Description		No
CustomerRewardPoints	Numeric(4.0)	Customer Reward Points		No

Is it possible to solve it using global formulas instead of the add and subtract rules?

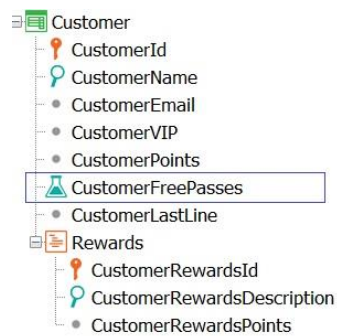
Do we get the same behavior?

HORIZONTAL FORMULAS (OPTIONAL)

Before further testing, a new requirement is made:

Customers can get free passes (CustomerFreePasses) in the current month, depending on the number of tickets recorded:

- If they have more than 2 and less than 6 tickets in a month, 1 free pass is generated.
- If they have more than 5 and less than 9 tickets in a month, 2 free passes are generated.
- If they have more than 8 tickets in a month, 3 free passes are generated.



What definition would you make for the CustomerFreePasses formula attribute?

One option could be as follows:

```

Formula Editor
1 IF count(TicketDate, TicketDate.month()=month(today()))>2 and count(TicketDate, TicketDate.month()=month(today())) < 6;
2 IF count(TicketDate, TicketDate.month()=month(today()))>5 and count(TicketDate, TicketDate.month()=month(today())) < 9;
3 IF count(TicketDate, TicketDate.month()=month(today())) > 8;
    
```

Another option is to define a FreePasses procedure that returns the corresponding value. Implement this option.

What will its Parm rule look like?

How will its invocation be so that the CustomerFreePasses attribute takes the value returned by the procedure?

AGGREGATE FORMULAS

Previously, the price of the tickets was defined at the time of purchase; however, now the price will be defined according to the screening room selected. To this end, it is required to save a price history log for the room and the date on which the price was saved (NOTE: it must not be possible to repeat the date). The value that will be used to calculate the ticket will be that of the latest date.

According to the above, which would be the correct option to model the requirement?



What aggregation formula would you use to return the price value according to the most current date?

RoomUpdatedPrice = _____

Once the Room transaction is updated, **what changes** do you need to make in Ticket? Is Ticket Price still valid?

Make the necessary changes (in the Room trn and in the Ticket trn) and test it at runtime.

In the Movie transaction, a field is to be added to indicate whether the movie is Premium or not.

If the movie is Premium, a surcharge of 7% of its value should be added at the time of purchase, and if the movie doesn't have this field specified, the regular price should be taken.

Room Type	VIP
Movie Id	1
Movie Name	Avengers End Game
Seats Quantity	<input type="text" value="1"/>
Room Updated Price	135.00
Total Price	144.45
Points	14

You need to know the number of tickets sold and available seats for each Schedule record.

UNANIMO

- Actors
- Billboards
- Categories
- Countries
- Customers
- Film Directors
- Candy Stores
- Cinema Products
- Employees
- Movie

Billboards

Room Id	Movie Date	Movie Schedule	Movie Id	Movie Name	Movie Poster	Room Seats	Seats Available	Seats Sell		
1	12/15/22	11 AM	2	Avengers: Endgame		50	50	0	UPDATE	DELETE
4	12/15/22	5 PM	4	Avatar: Way to water		30	2	28	UPDATE	DELETE
6	12/15/22	8 PM	8	Unofficial Mission		50	50	0	UPDATE	DELETE
5	12/19/22	5 PM	4	Avatar: Way to water		40	37	3	UPDATE	DELETE
1	12/19/22	11 AM	5	Forest Gump		50	50	0	UPDATE	DELETE

If all the tickets for that room have already been sold, the system should not allow saving the purchase.

DYNAMIC TRANSACTIONS


Another requirement of the Cinemas system is to merge the information and have a catalog of movies and products sold in the candy store. It will store the product identifier, its name, an image, and its price.

In order to know all the products offered by the Cinema (movies and candy), a report is required to list all the movies and candy products.








It is necessary to display its identifier, its name, a picture and the type (if it is a movie or a candy item).

Tip

Use a dynamic transaction to retrieve the information from both catalogs and list it in the previously requested report.



All Products

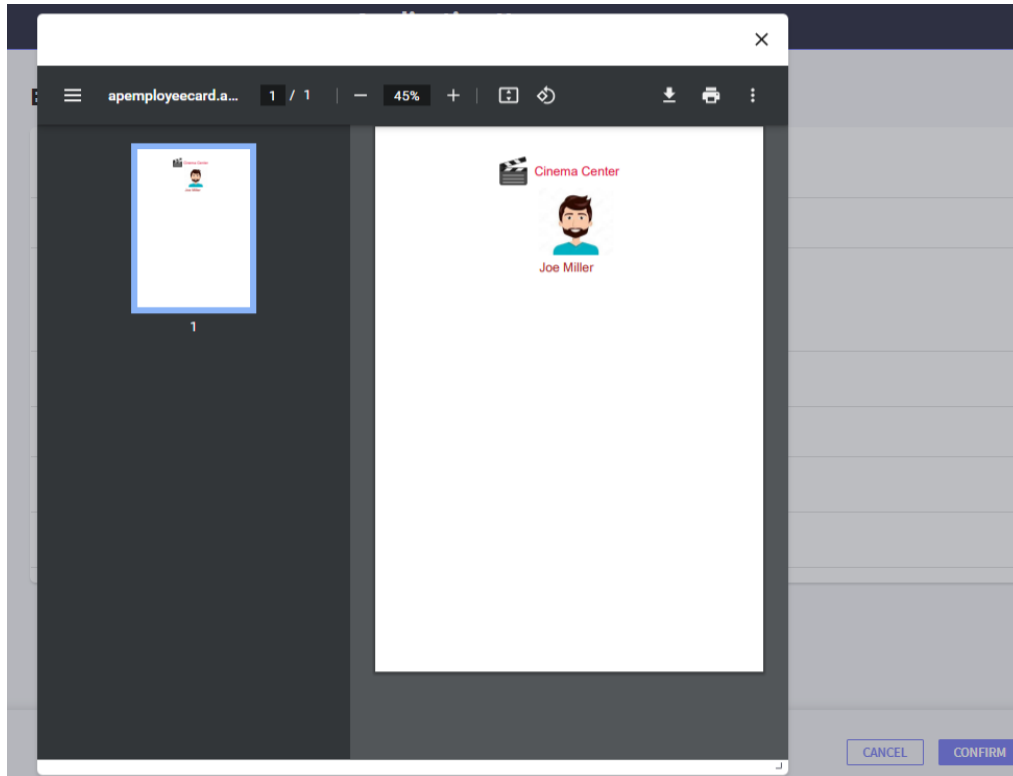
Id	Type	Product Name	Photo
1	Candy	Popcorn	
1	Movie	Avengers End Game	
2	Candy	Soda	
2	Movie	Catch me if you can	
3	Candy	Gummies	
3	Movie	Forrest Gump	
4	Candy	Hot Dog	

First of all, you must create a transaction (CandyStoreProduct) to enter, modify and delete the candies to be sold. About them, we are interested in recording their name, a photo, and their price.

What changes should you make to the proposed implementation if the end user would like to be able to enter, change, or delete movies and candies through the form of a new transaction that will be dynamic?

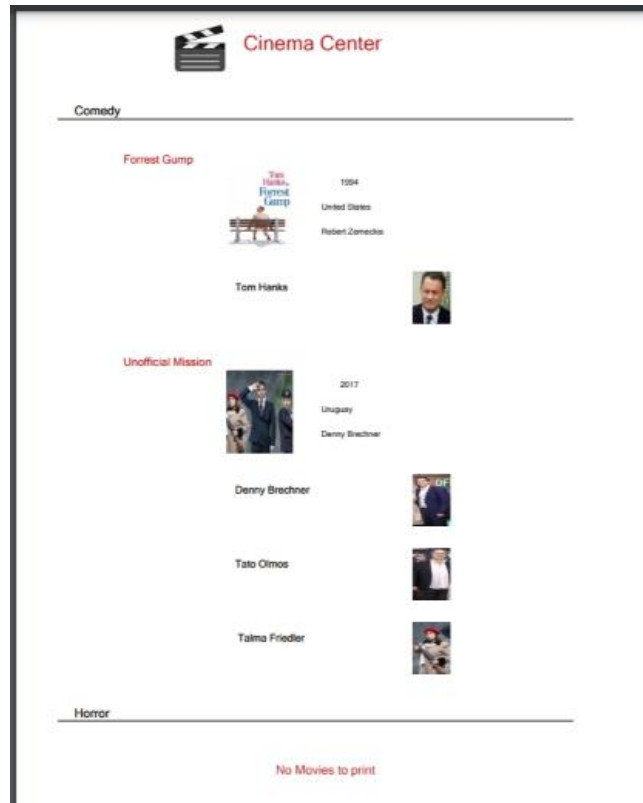
EVENTS IN TRANSACTIONS

The cinema system requires that when saving a new employee, a popup window is displayed on the screen with his/her information in the style of a badge:



MORE ABOUT NESTED FOR EACH COMMANDS. CASES AND NAVIGATION

We are asked for a report with the following format. Place the categories and their related movies, as well as the actors appearing in those movies. If there are no movies associated with the category, a message should be displayed informing that there is no information:

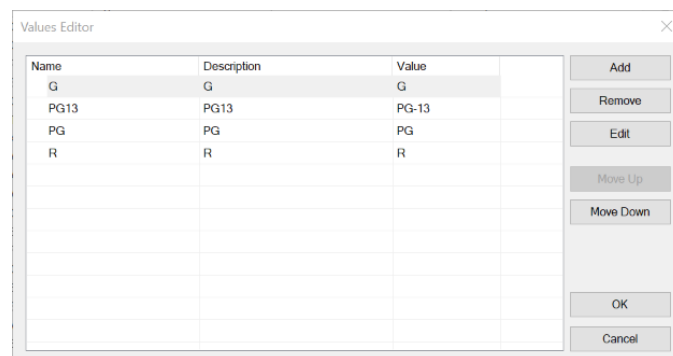


We are asked for a list with the same format of the previous one and with the restriction that only the categories with associated movies be listed. How would you achieve the above? Write the code below with the changes to be made.

Do a Save As of the first procedure and make the appropriate modifications to be able to receive a Category as a parameter from a web panel. The procedure must show the information of the selected category and, if no category is selected, all the information of the Categories with their corresponding Movies and Actors must be shown.

SUBROUTINES

In the Movie transaction, add a new attribute called MovieRating where we will store the age rating for each movie (G, PG, PG-13, R). For this, create an enumerated domain called Rating that will contain the following data:

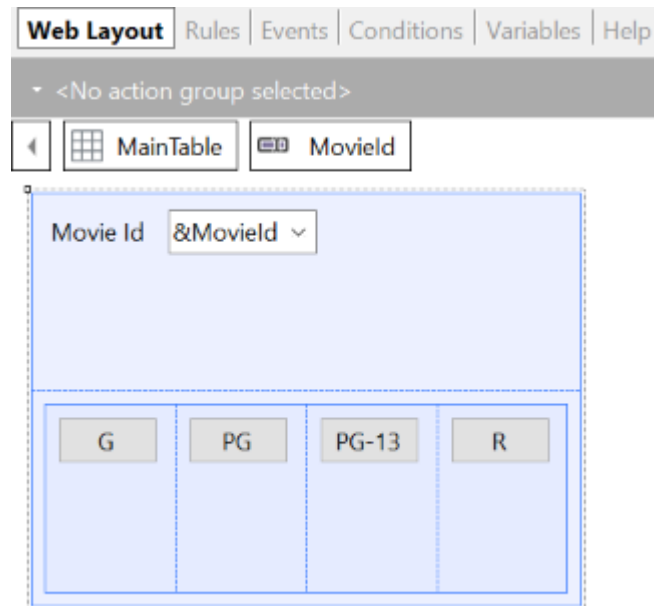


Name	Description	Value
G	G	G
PG13	PG13	PG-13
PG	PG	PG
R	R	R

About enumerated domains:

Names containing hyphens or other special characters are not allowed in the "Name" and "Description" columns. To enter the PG-13 rating classification, do so without the hyphen.

Next, create the following web panel where we will select a movie and click a button to update its rating according to the selected value.



Note that certain code will need to be repeated for each button, so we can resort to subroutines to use the same code but update as selected.

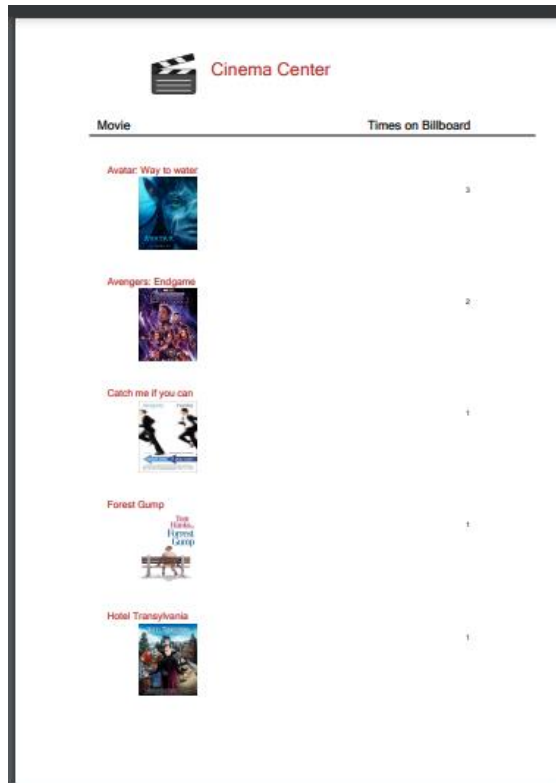
```
1 Event 'G'  
2     &Rating = Rating.G  
3     do "UpdateRating"  
4 Endevent
```

Tip

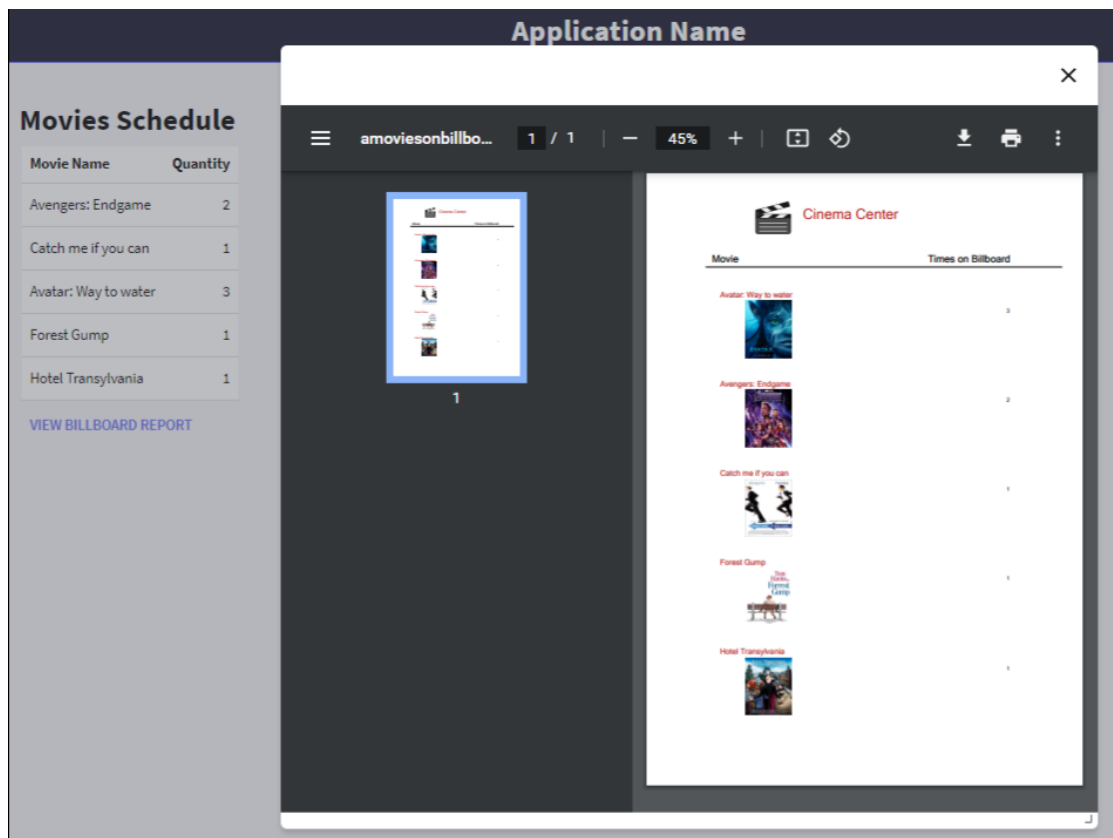
Use a responsive table to place the buttons, and change the class of the buttons to one that matches the appearance of the program, for example: BTN Cancel.

UNIQUE CLAUSE

Another requirement of the Cinema is to view a report of the movies that appear on the schedule, without repetitions and with the number of times they have appeared on the schedule.



A new requirement is to be able to view the above information in a web panel. Place a button where you can query the above list as a popup.



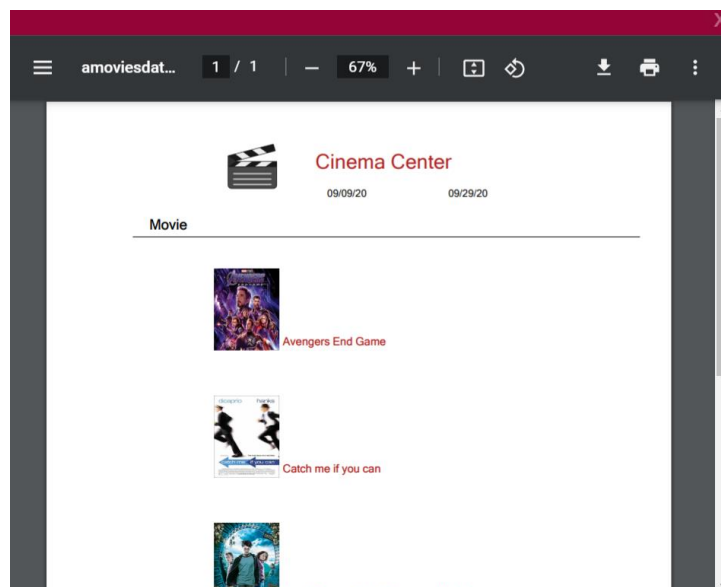
Tip

Change the Grid classes to WorkWith and the Button class to Btn Cancel to enhance the application's design.

DATA SELECTOR

A web screen is required that allows selecting a date range. Then, then by pressing a button it should show as a popup the list of all the movies that appear on the schedule, but only those movies that are in the selected date range.

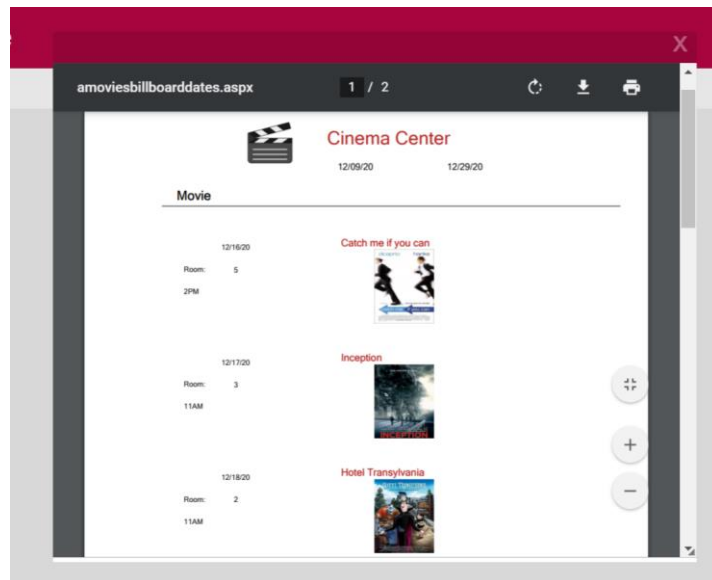
What's the resulting navigation? Explain it...



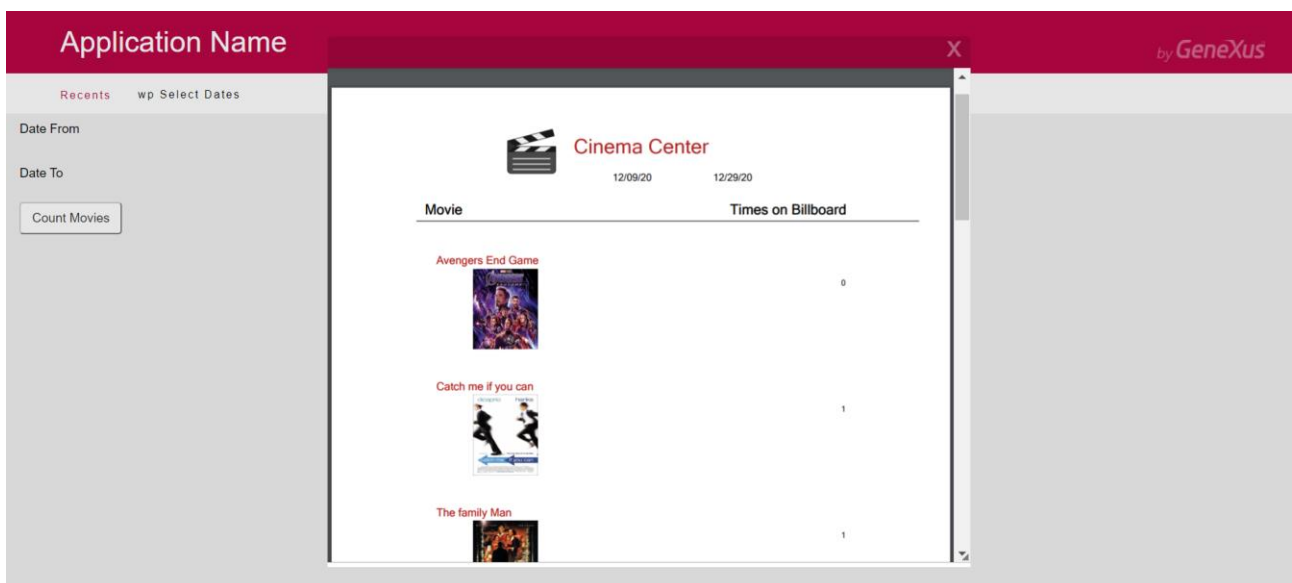
NOTE: Since this is a query that will be frequently made in the Cinema system, generate a new data selector object for later use.

In the same web screen, another popup list is to be generated to query the information of the movies that are going to be shown, as well as their time, room and date, within a certain date range.

What is the difference between this resulting navigation and the previous one? Explain it...



Additionally, it should allow viewing a popup list with the movies on the schedule, without repetitions and counting only the number of times they have appeared on the schedule in that date range.



What is the difference between this resulting navigation and the previous ones? Explain it...

PAGING CLAUSES

In the Work With screen associated with the Schedule, I am only interested in viewing 5 records at a time, and I want the output format to be an infinite scroll. Apply the requested change.

PAGING IN FOR EACH, PATTERN, AND GRID (OPTIONAL)

A screen is required where the user can call a list with the actors' information. In particular, the screen must allow selecting how many records to skip and how many to display later. Example: "I want to skip the first 3 records and be able to access the next 5 to work with that data."

Index	Actor Name	Country	Portrait
4	Daisy Ridley	United Kingdom	
5	Emma Watson	United Kingdom	
6	Robert Downey Jr	United States	
7	Chris Hemsworth	Australia	
8	Adam Sadler	United States	

BUSINESS COMPONENT

We will perform a series of operations on the database through Business Components.

UPDATING EMPLOYEES IN CHARGE

A specialized screen is required to select a room type (VIP, 3D, etc.) and select an employee. The employee will be in charge of all the rooms of that type.



Room Type (None) ▾

Employee Id Billie Robin ▾

UPDATE EMPLOYEE

WORKING WITH MOVIES AND THEIR ACTORS

A screen with a special format that allows selecting a movie to work with its actors is requested. Two buttons will be offered:

- One that allows us to modify the actor's role to lead or to remove the lead role.
- Another one that allows deleting the actor from the selected movie.

Tip

Use the &Messages variable of GeneXus to program the system's messages in case of a failed or successful update.

What could you do so that the variable associated with the actor only shows the actors who are in the selected movie?

Application Name

▲ Data has been successfully updated.

Movie Id

Actor Id Is Starring

UPDATING THE DATABASE WITH A PROCEDURE

In the same screen where you made the update of the employees in charge of the rooms according to their type, implement a button that performs the same update mentioned above but now using a procedure.

How could you delete the movies and their associated actors through a procedure?

Insert a new movie (any movie) with its corresponding actors.

Program a button on the screen you created earlier to work with movies and all their actors; this button should delete a selected movie and delete the associated actors. Delete the Movie you have just created.

What happens when you try to delete Movies that are related to other tables?

WEB SCREENS: WEB PANELS

Next, we will start generating customized interactive screens that have been requested for the Cinemas system.

GRID VARIABLES AND ACTIONS IN THE WORK WITH PATTERN (OPTIONAL)

In the screen that allows us to work with the movies, we are asked to make the following changes:

- IDs must not be displayed.
- A calculated field that shows the number of lead actors in the movie.
- A button that shows the information about the movies previously created where the Category, the movie, and the actors in it are displayed.

Name	Year	Poster	Description	Film Director Name	Country Name	Category Name	Starring Actors	Premium Movie	Rating		
Avatar: Way to water	2012		Jelie Sully and Neytiri have formed a family and are doing everything to stay together. However they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, she must fight a difficult war against the humans.	James Cameron	United States	Fantasy	0	<input checked="" type="checkbox"/>	PG13	UPDATE	DELETE
Avengers: Endgame	2019		After Thanos, an intergalactic overlord, disintegrates half of the universe, the Avengers must reunite and assemble again to reinvigorate their trounced allies and restore balance.	Joe Russo	United States	Action	2	<input checked="" type="checkbox"/>	PG	UPDATE	DELETE
Catch me if you can	2002		Notorious con artist Frank Abagnate has duped people worth millions of dollars with his masterful art of deception. With his seams getting tighter, he is soon pursued by FBI agent Carl Hanratty.	Steven Spielberg	United States	Comedy	2	<input type="checkbox"/>	PG13	UPDATE	DELETE
Forest Gump	1994		Forest, a man with low IQ, recounts the early years of his life when he found himself in the middle of key historical events, all he wants now is to be reunited with his childhood sweetheart, Jenny.	Robert Zemeckis	United States	Drama	1	<input type="checkbox"/>	PG	UPDATE	DELETE
Hotel Transylvania	2012		Count Dracula runs a high-end resort for monsters and is overprotective of his daughter, Mavis. When a human named Johnny enters the hotel, Dracula tries to prevent Mavis from falling for him.	Genndy Tartakovsky	Australia	Children	1	<input type="checkbox"/>	G	UPDATE	DELETE


CREATING INTERACTIVE SCREENS

- A screen showing the Schedule information sorted by date of screening is requested. It should be possible to filter by:
 - Date, which must be greater than or equal to today's date. (The system must automatically suggest using today's date.)
 - Time.
 - Type of Room.
 - Name of the movie.

If the filters are empty, all the information of the schedule should be displayed.

For each record, a button is required to show the details of the selected movie when clicked on. In the detail screen, implement a button that allows returning to the previous web panel.







GeneXus™ **Application Name**

Room Movie Date 

Room Movie Schedule ▼

Room Type ▼

Movie Name

Room Type	Room Movie Date	Room Movie Schedule	Movie Name	Movie Poster	Goto
VIP	12/15/22	5 PM	Avatar: Way to water		
VIP	12/19/22	5 PM	Avatar: Way to water		
Normal	12/21/22	5 PM	Hotel Transylvania		

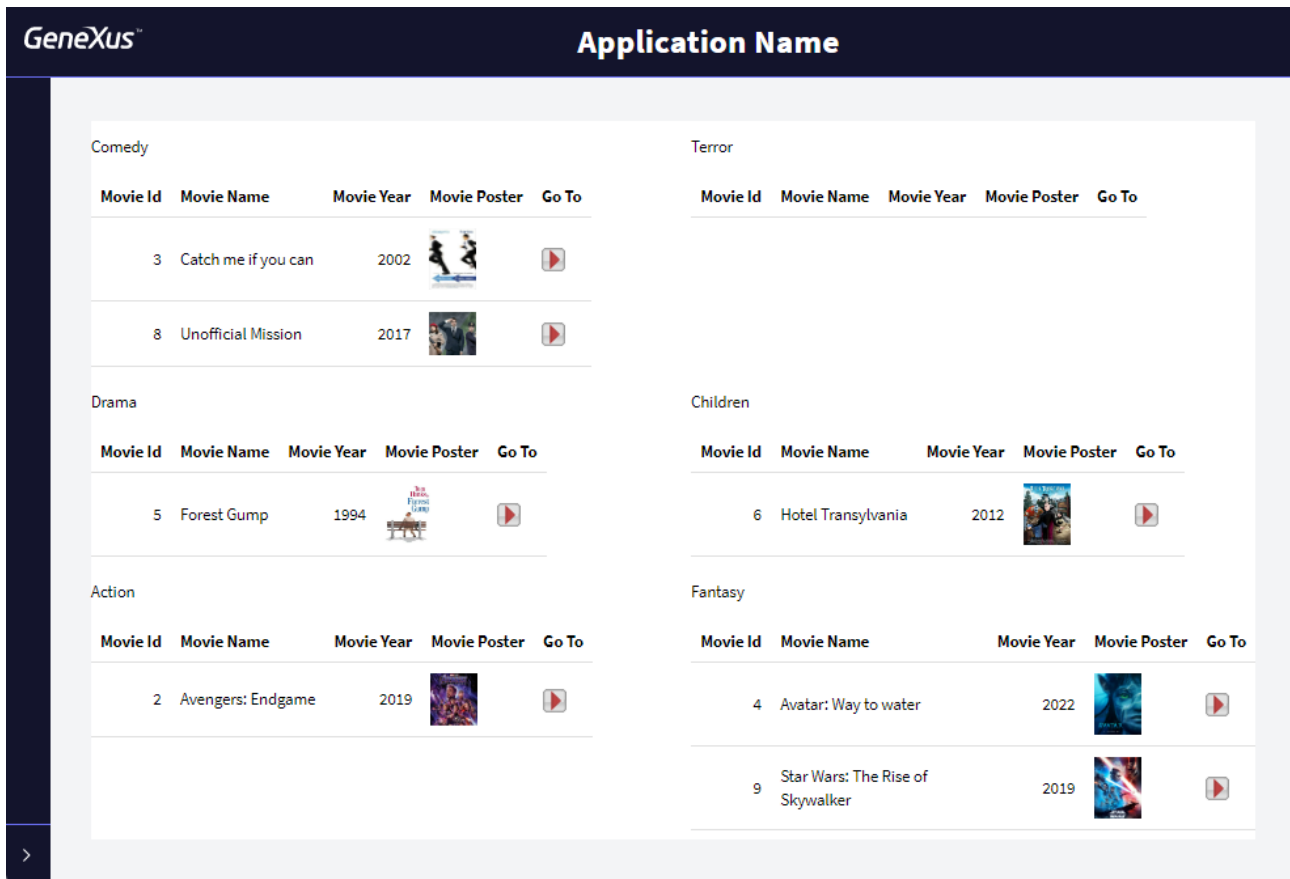
>

MULTIPLE GRIDS

A Web Panel is requested that allows viewing all Categories and, for each one of them, their associated movies.

For each movie, a button is requested that shows the details of the selected movie when clicked on. In the detail screen, implement a button that allows returning to the previous web panel.

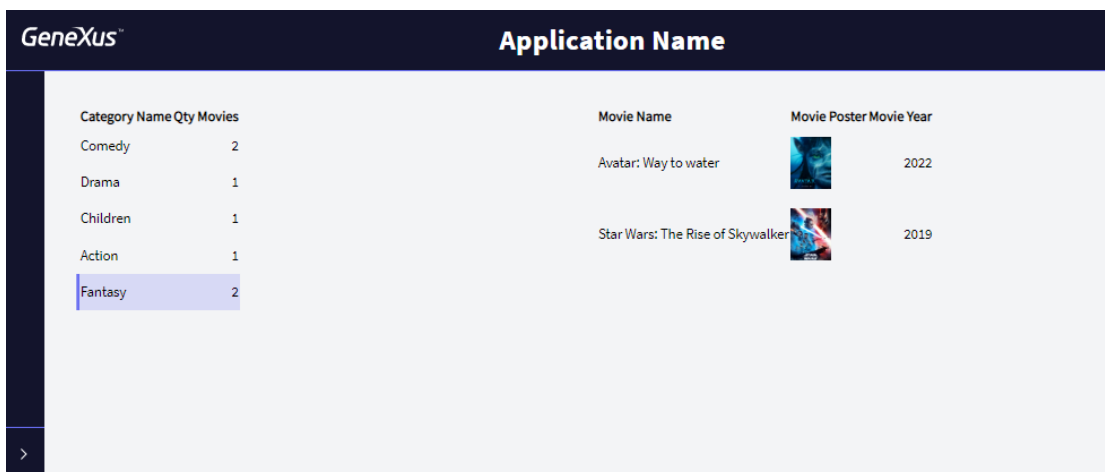
Note: The web panel of the film detail can be reused.



How could you make sure that only categories that have associated movies are displayed?

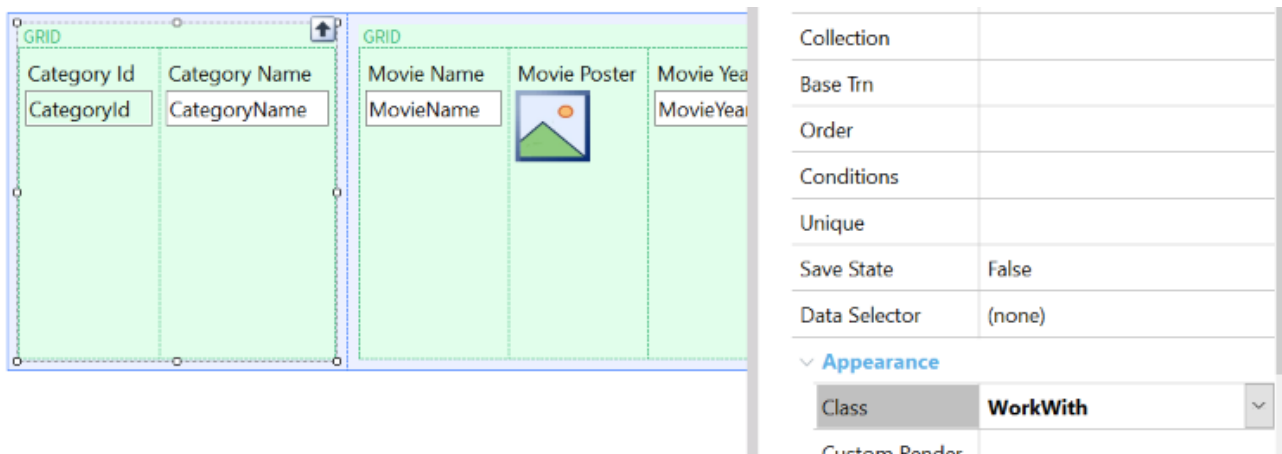
Implement this change.

A screen is required to show the list of categories that have associated movies (without repetitions) and their number of movies. In addition, it should allow selecting a category and then display the associated movies on the same screen.



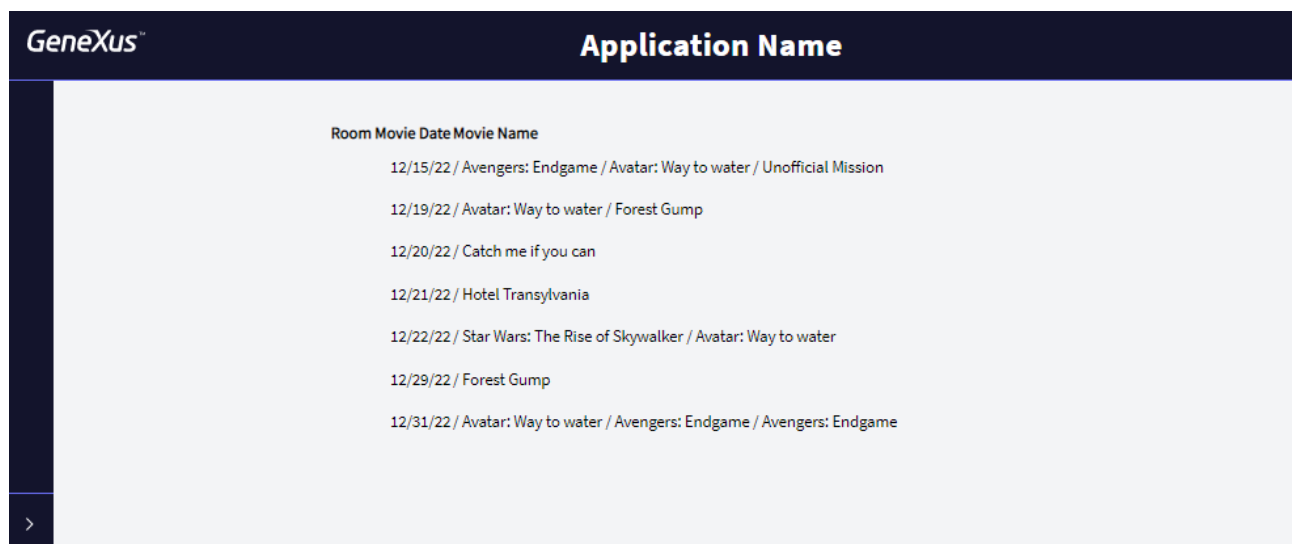
What should be the base table of the Grid that displays the Category information?

Note: It is recommended that the WorkWith class be applied to the Grid to improve its appearance.



GRID WITHOUT BASE TABLE

A special screen is required that shows the movies to be presented for each date. The format is as follows:



LOADING COLLECTIONS

A ranking of Customers is required according to the seats of each ticket they have purchased. To solve the above, use a structure.

Structure Documentation			
Name	Type	Description	Is Collection
sdRankingCustomer		sd Ranking Customer	<input checked="" type="checkbox"/>
sdRankingCustomerItem			
• CustomerName	Attribute:CustomerName	Customer Name	<input type="checkbox"/>
• CustomerTickets	Numeric(4.0)	Customer Tickets	<input type="checkbox"/>

In addition to the above, when selecting a customer, the screen must show the customer's information in PDF format with the corresponding number of tickets.

Tip

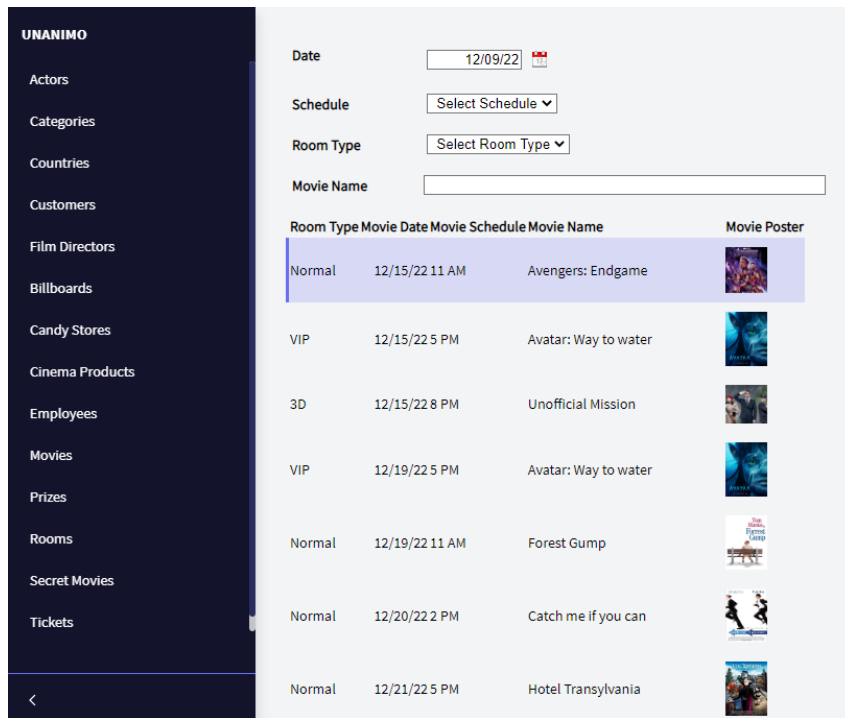
Use the AllowSelection property of the Grid and the OnLineActivate event.

SAVING CONTEXT DATA (OPTIONAL)

Open the web panel containing the entire information of the Schedule. It is required that when returning to this screen after viewing the movie's details, the filter values are saved.

SPECIAL SCREEN FOR TICKET SALES

Make a copy of the previous screen and modify the web panel classes so that it looks similar to the following:



Tip

Use the *WorkWith* class of the Grid and the *WWColumn* and *WWOptionalColumn* classes.

When selecting a line of the Schedule, a popup window should be displayed with the details of the selected movie. The Employee may or may not select a customer and the number of seats to be purchased; by clicking on the Buy button, the purchase will be made. Note: sometimes it will not be possible to make the purchase due to some control that we have programmed, so the system messages should be displayed on the screen:

GeneXus Application Name

Date: 12/15/22
Schedule: 5 PM
Room Type: Select Room Type
Movie Name:

Room Type	Movie	Date	Movie Schedule	Movie Name	Movie Poster
VIP	12/15/22 5 PM	Avatar: Way to water			
VIP	12/19/22 5 PM	Avatar: Way to water			
Normal	12/21/22 5 PM	Hotel Transylvania			
VIP	12/31/22 5 PM	Avatar: Way to water			

Buy Tickets

⚠ Not enough seats

Room	Date	Schedule	Movie Name	Seats Available
4	12/15/22	5 PM	Avatar: Way to water	2

Customer Id: Anne Parker

Ticket Seats Quantity: 1, 2, 3, 4, 5, 6

BUY

GeneXus Application Name

Date: 12/15/22
Schedule: 5 PM
Room Type: Select Room Type
Movie Name:

Room Type	Movie	Date	Movie Schedule	Movie Name	Movie Poster
VIP	12/15/22 5 PM	Avatar: Way to water			
VIP	12/19/22 5 PM	Avatar: Way to water			
Normal	12/21/22 5 PM	Hotel Transylvania			
VIP	12/31/22 5 PM	Avatar: Way to water			

Buy Tickets

⚠ Data has been successfully added.

Room	Date	Schedule	Movie Name	Seats Available
5	12/19/22	5 PM	Avatar: Way to water	40

Customer Id: Anne Parker

Ticket Seats Quantity: 1, 2, 3, 4, 5, 6

BUY

MULTI-CHANNEL SCREENS: PANELS

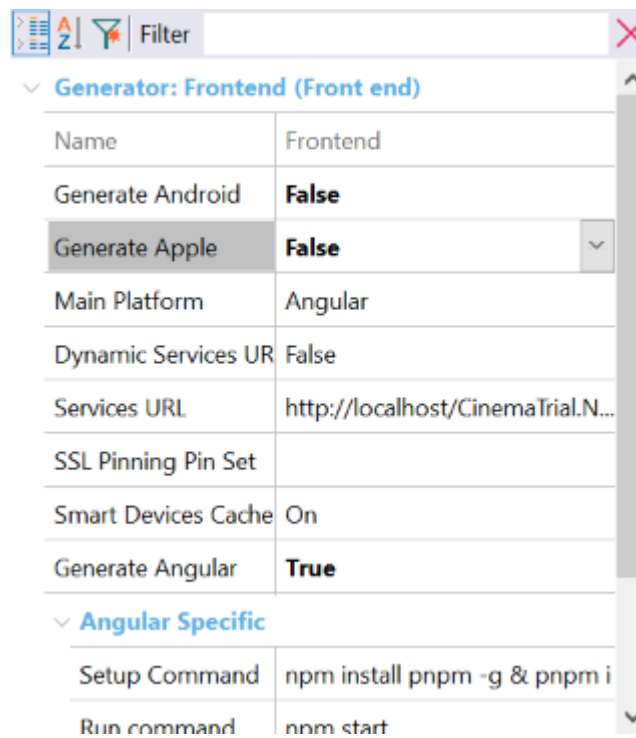
To complete the following exercise, you will need to check that you meet the [necessary requirements to generate in Angular](#).

A panel generated with Angular containing the information of the schedule is required.

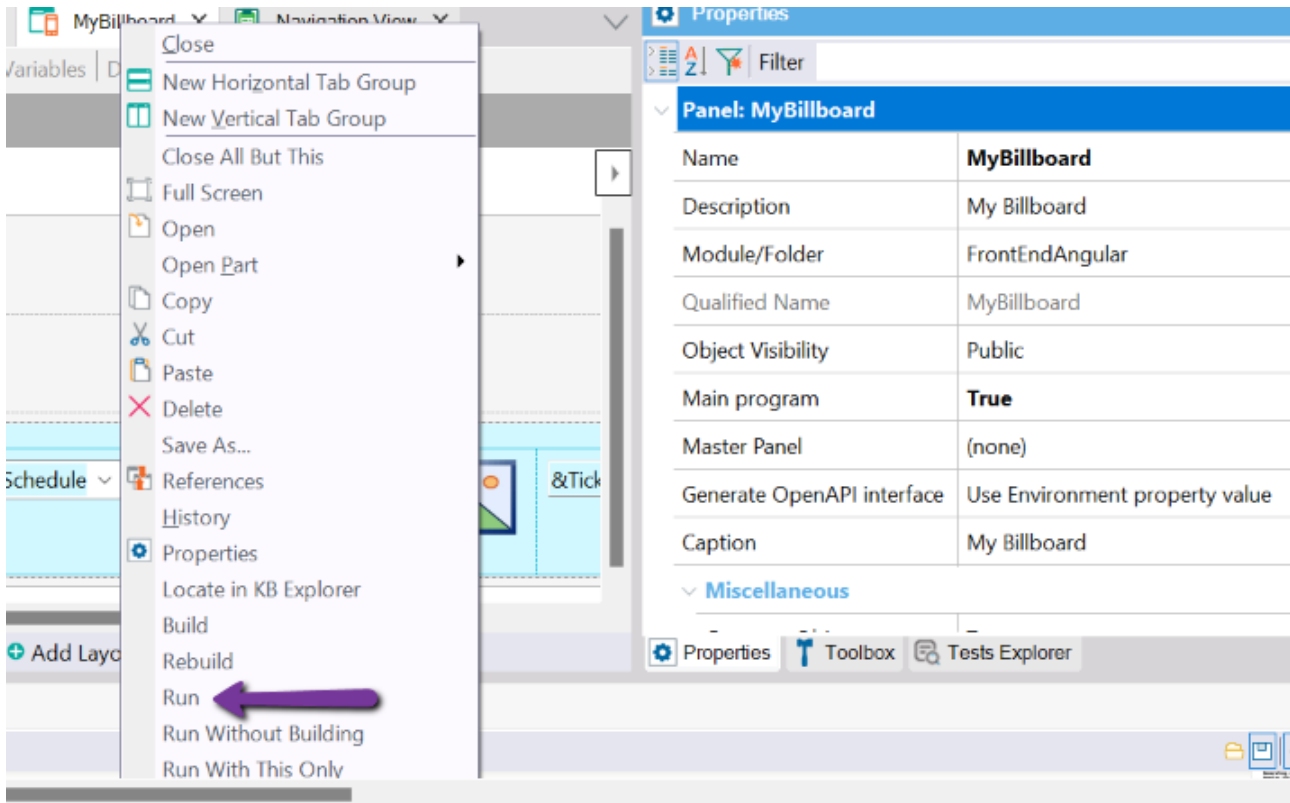
To solve the above, press Ctrl+N and select the Panel object. In the panel, drag a Grid and select the movie's date, time, room, name, and poster.

Set the Autogrow property of the Grid to True.




After saving the panel object, to start running on Angular, go to *FrontEnd* in the *KBExplorer*. Change the *Generate Angular* property to *True*; since we are not going to develop for mobile applications at the moment, change the *Generate Android* and *Generate Apple* properties to *False*.



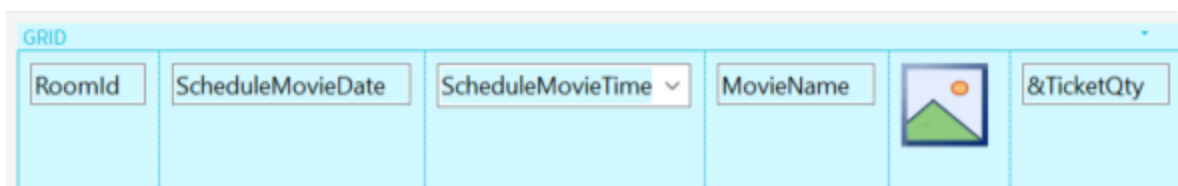
In the panel properties, change the Main Program property to True, right-click on the object name, and click on Run to run the application.



Check that the application looks as follows.


3	7/5/2021	5 PM	Hotel Transylvania	
4	6/5/2021	11 AM	Avengers End Game	
6	30/4/2021	8 PM	Inception	

Now add a variable that calculates the number of sales issued by Schedule. *Note: we have an attribute that calculates the number of seats sold; we do not want to show that information, but the number of tickets or sales.*



Program what is needed in the events section and test how the application works.

We want to see the information sorted by date and to program filters by the name of the movie; also, we want to see the screenings starting from a certain date.

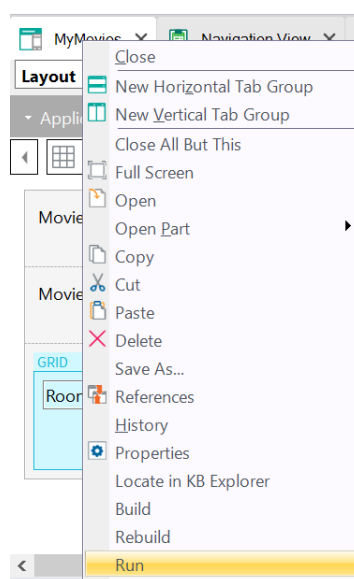
Movie Date	04/05/2021			
Movie Name	av			
6/5/2021	11 AM	4	Avengers End Game	 3

TESTING THE APPLICATION ON SMART DEVICES

Change the Generate Android property to True to start testing the application in Smart Devices and check that the Main Platform property is changed to Android.

Generators	
Generate Android	True
Generate Huawei	False
Generate Apple	False
Generate Angular	False

Now right-click on the panel's name and run the application. A screen should be displayed in Android that will be the same as the one previously generated for the web.



PROGRESSIVE WEB APPLICATIONS [SUGGESTED]

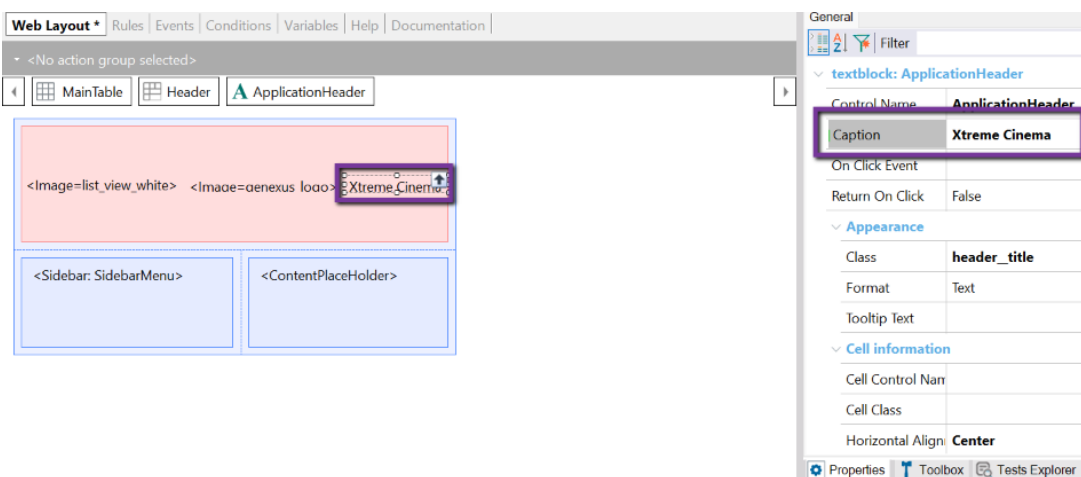
We will slightly change the application's appearance to make the PWA look better.

Open the KB Explorer and search in the filter "RWD Master Page." This is the Master Page associated with all the GeneXus objects in our KB.

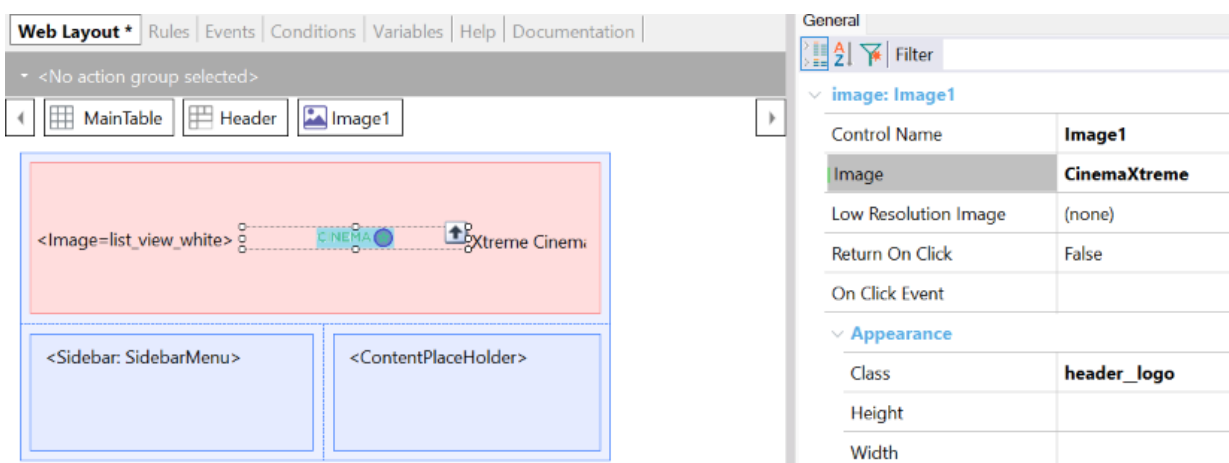
Note that it has three sections:

- A Header containing the name of the application and logo.
- A web component object that is responsible for providing the menu of recent links.
- A content placeholder where the objects of our application are displayed.

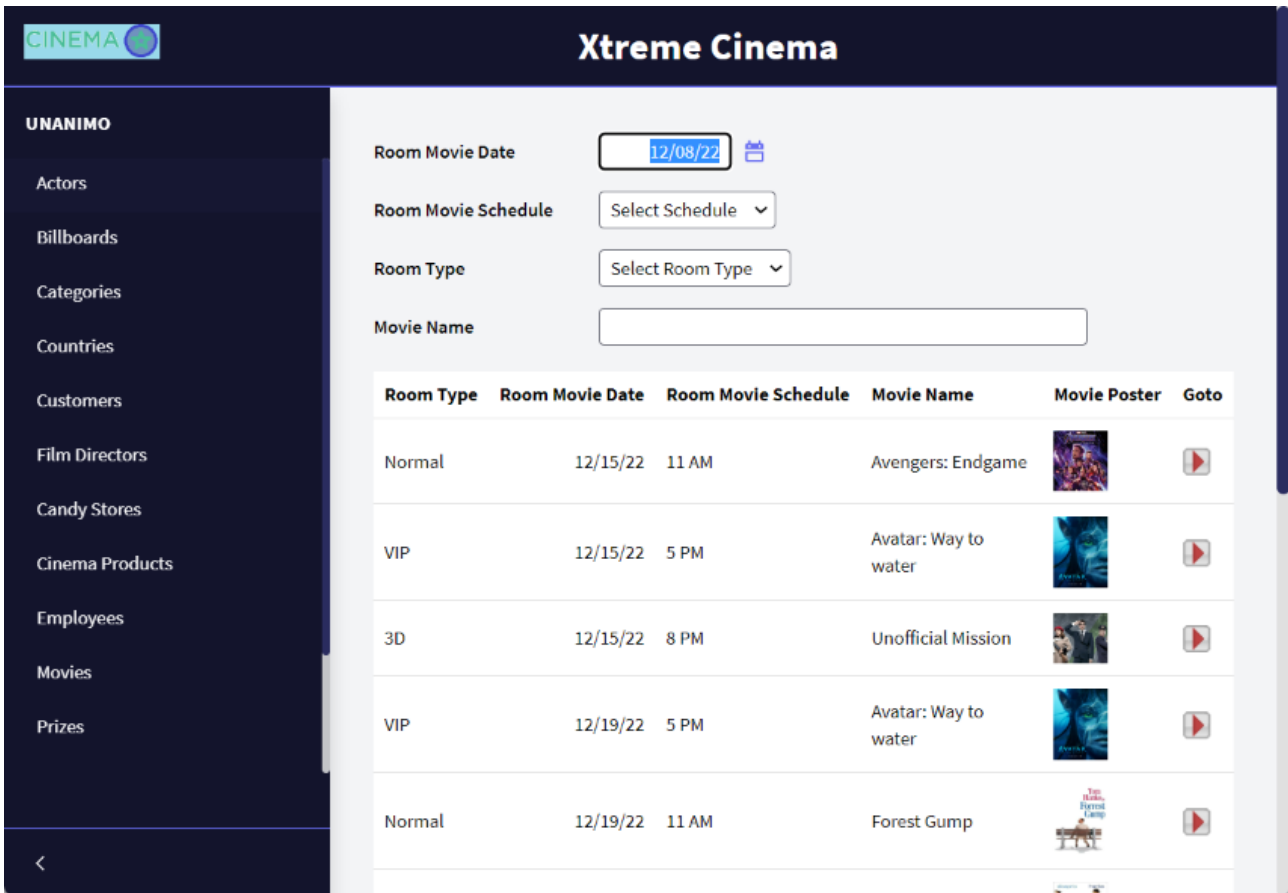
In the *Header*, select the *ApplicationHeader* control and change the caption property to the application's name: My Cinema.



Select the *Image1* control and in the Image property select an image as company logo (you can use the images provided for the practice exercises). Check that the *PoweredBy* class is applied.



Make a copy of the web panel where all the schedule information is displayed:



Inside the new web panel, select the Grid and in its class property change it to WorkWith.

Also, hide the Movie ID and delete the variable that allows us to see the movie's details and its associated event.




Select the attributes RoomId, ScheduleMovieDate, ScheduleMovieTime and MovieName; in the class property, enter *WWColumn*. For each column in the Title property, change their names to shorter ones.

Select the attributes RoomType, MovieId, and Movie Poster; in the class property, enter *WWColumn* *WWOptionalColumn*. Also, enter shorter names for each column.

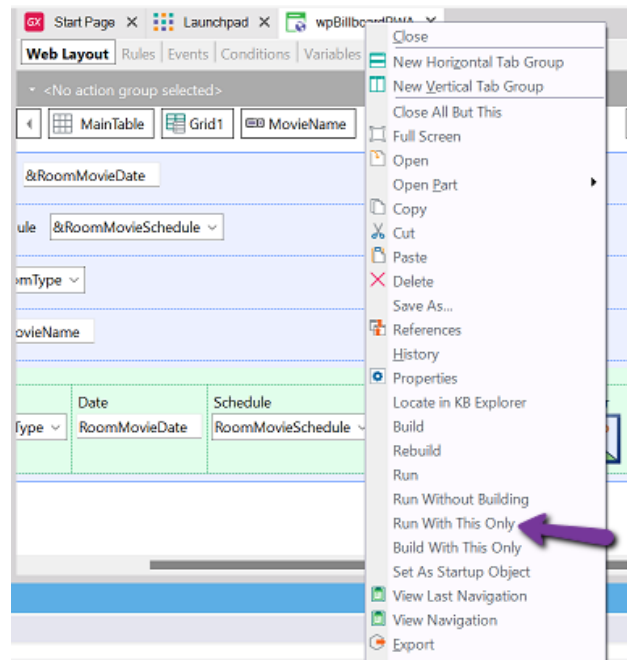
Within the properties of the web panel where the nested Grid is programmed, change *Main Program* to *True*. Note that a new group of properties called *Main object properties* have been activated. In this block, change the *Web Application* property to *Progressive*.

Filter	
Type	Web Page
Master Page	MasterUnanimoSidebar
Show Master Page when Pop-u	False
Main program	True
On session timeout	Ignore
Focus control	Use Environment property value
Cache expiration lapse	
Automatic refresh	Yes
Auto compress http traffic	Use Environment property value
Qualified Name	wpBillboardPWA
Object Visibility	Public
Main object properties	
Web Application	Progressive
Location	
Generator	Default (.NET)

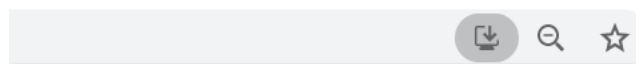
By changing the default property to Progressive, a new group of properties called Web Application has been activated and we will configure them in this way:

Filter	
Web Application	
Web Application Name	Xtreme Cinema
Web Application Short Name	XC
Web Application Description	Cinema Progressive Web App description
Primary Text Direction	Auto
Display	Fullscreen
Background Color	 #D4D8D0
Theme Color	 #0000CD
Prefer Related Applications	False
Web Application Icon	movies 
Android Alternative App	
Android Alternative App Identifier	
iOS Alternative App	
Offline Object	WWBillboard
Compatibility	

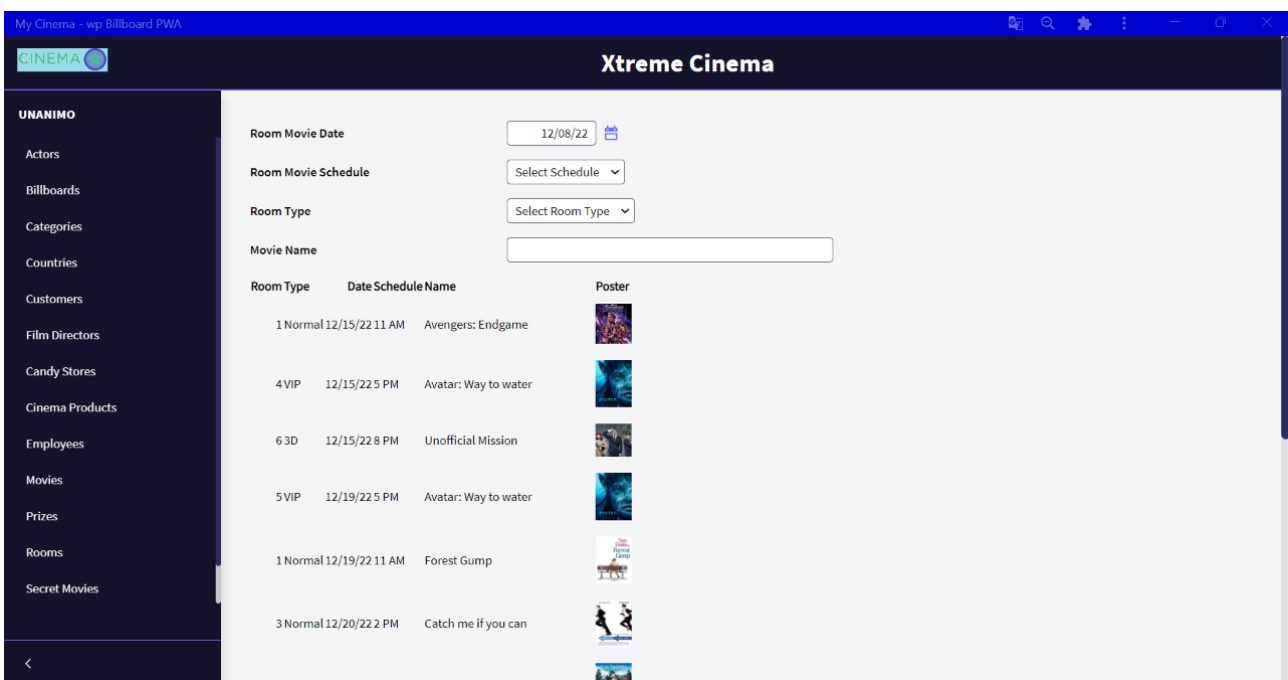
Right-click on the web panel and Team the option Run with this only.



In the browser, note the option to install the application, click on it, and go to the shortcut that has been created on the desktop.



Open the application with a double-click. Note that the colors, icon, and screen resize correspond to the settings we have made.



The screenshot shows a web application titled "Xtreme Cinema" with a navigation menu on the left. The main content area contains a form with the following fields:

- Room Movie Date:** A date picker set to 12/08/22.
- Room Movie Schedule:** A dropdown menu labeled "Select Schedule".
- Room Type:** A dropdown menu labeled "Select Room Type".
- Movie Name:** A text input field.

Below the form is a table with the following data:

Room	Date	Schedule Name
1	12/15/22	11 AM Avengers: Endgame
4	12/15/22	5 PM Avatar: Way to water
6	12/15/22	8 PM Unofficial Mission
5	12/19/22	5 PM Avatar: Way to water
1	12/19/22	11 AM Forest Gump
3	12/20/22	2 PM Catch me if you can
2	12/21/22	5 PM Hotel Transylvania
3	12/22/22	2 PM Star Wars: The Rise of Skywalker
6	12/22/22	8 PM Avatar: Way to water

WEB SERVICES [SUGGESTED]

We are requested to publish a web service that allows querying the information of a certain movie by its name.

To implement the above, we are going to create an SDT collection based on the Movie transaction. You can drag the transaction to the structure or create the SDT manually.

Next, drag the structure to a data provider and configure the following:

The screenshot shows the GeneXus IDE interface. On the left, a code editor displays the following code for a data provider named `sdtMovies`:

```

1 sdtMovies
2 where MovieName like &MovieName when not &MovieName.IsEmpty()
3 {
4     sdtMoviesItem
5     {
6         MovieId
7         MovieName
8         MovieYear
9         MoviePoster
10        MovieDescription
11        MovieCountryId
12        MovieCountryName
13        CategoryId
14        CategoryName
15        FilmDirectorId
16        FilmDirectorName
17        IsPremiumMovie
18    }
19 }
20

```

On the right, the 'Data Provider: dpViewMovies' configuration panel is visible. The 'Expose as Web Service' checkbox is checked, and the 'Web Service Protocol' is set to 'REST Protocol'. Other settings include 'Generate OpenAPI interface' (Use Environment property value), 'Use Native Soap' (Use Environment property value), 'Exposed namespace' (CinemaTrial), 'Main program' (False), 'Call protocol' (Internal), 'Module/Folder' (MyServices), 'Qualified Name' (dpViewMovies), and 'Object Visibility' (Public). The 'Output' section shows 'Infer Structure' set to 'No' and 'Output' set to 'sdtMovies'.

The condition previously programmed will allow us to retrieve the list of all movies even if the parameter is empty. We will expose the service as Rest.

Place a rule that receives the attribute we will filter by: `parm(in:&MovieName)`;

Now right-click on the DP name and select the option “Build with this only” to expose the service. To confirm that everything is correct, place the following link in the browser:

<http://server/baseURL/rest/DataProviderName>

The following should be displayed:

```

[{"MovieId":1,"MovieName":"Avengers End Game","MovieYear":2019,"MoviePoster":"PublicTempStorage/multimedia/AvengersEnd_4f30d21695f74e7fb87112f132ab0770.jpg","MovieDescription":"Avengers: Endgame is an American superhero film based on the Marvel Comics superhero team the Avengers.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":1,"FilmDirectorName":"Joe Russo","IsPremiumMovie":true}, {"MovieId":2,"MovieName":"Catch me if you can","MovieYear":2002,"MoviePoster":"PublicTempStorage/multimedia/CatchMe_c115a88cf1742529610e64b27a9f24.jpg","MovieDescription":"The film is based on the life of Frank Abagnale, who, before his 19th birthday, successfully performed cons worth millions of dollars by posing as a Pan American World Airways pilot, a Georgia doctor and a Louisiana parish prosecutor.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":3,"FilmDirectorName":"Steven Spielberg","IsPremiumMovie":false}, {"MovieId":3,"MovieName":"Forrest Gump","MovieYear":1994,"MoviePoster":"PublicTempStorage/multimedia/ForrestGump_1293eaa799bd4485abd10a7ff600d11c.jpg","MovieDescription":"The story depicts several decades in the life of Forrest Gump (Hanks), a slow-witted but kind-hearted man from Alabama who witnesses and unwittingly influences several defining historical events in the 20th century United States.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":1,"CategoryName":"Comedy","FilmDirectorId":7,"FilmDirectorName":"Robert Zemeckis","IsPremiumMovie":false}, {"MovieId":4,"MovieName":"Hotel Transylvania","MovieYear":2012,"MoviePoster":"PublicTempStorage/multimedia/HotelTransylvania_36a5b8a603c549b9db289cdb3a7d925a5.jpg","MovieDescription":"The film tells the story of Count Dracula, the owner of a hotel called Hotel Transylvania where the world's monsters can take a rest from human civilization.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":4,"CategoryName":"Cartoon","FilmDirectorId":8,"FilmDirectorName":"Genndy Tartakovsky","IsPremiumMovie":false}, {"MovieId":5,"MovieName":"The family Man","MovieYear":2000,"MoviePoster":"PublicTempStorage/multimedia/FamilyMan_ce19e3c9ddf64a45b7bb9758fb6ce33b.jpg","MovieDescription":"The film centers on a man who experiences what his life might have been if he had made a different decision earlier in his life.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":3,"CategoryName":"Drama","FilmDirectorId":4,"FilmDirectorName":"Brett Ratner","IsPremiumMovie":false}, {"MovieId":6,"MovieName":"Harry Potter and the Prisoner of Azkaban","MovieYear":2004,"MoviePoster":"PublicTempStorage/multimedia/HPPrisoner_7c5f2c468a19cc64b0898481471a385.jpg","MovieDescription":"The film stars Daniel Radcliffe as Harry Potter, with Rupert Grint as Ron Weasley, and Emma Watson as Hermione Granger. Its story follows Harry Potter's third year at Hogwarts as he is informed that a prisoner named Sirius Black has escaped from Azkaban and intends to kill him.", "MovieCountryId":4,"MovieCountryName":"United Kingdom","CategoryId":6,"CategoryName":"Fantasy","FilmDirectorId":2,"FilmDirectorName":"Alfonso Cuaron","IsPremiumMovie":true}, {"MovieId":7,"MovieName":"Unofficial Mission","MovieYear":2017,"MoviePoster":"PublicTempStorage/multimedia/misionNoOficial_372f49892b604d39b1252d62e431613f.jpg","MovieDescription":"\u000aUnofficial Mission is a documentary comedy, directed by Alfonso Guerrero, Denny Brechner and Marcos Hecht.", "MovieCountryId":1,"MovieCountryName":"Uruguay","CategoryId":1,"CategoryName":"Comedy","FilmDirectorId":5,"FilmDirectorName":"Denny Brechner","IsPremiumMovie":false}, {"MovieId":8,"MovieName":"Inception","MovieYear":2010,"MoviePoster":"PublicTempStorage/multimedia/Inception_4199f1d287e3419f80146fdb045c3e02.jpg","MovieDescription":"The film stars Leonardo DiCaprio as a professional thief who steals information by infiltrating the subconscious of his targets.", "MovieCountryId":2,"MovieCountryName":"United States","CategoryId":5,"CategoryName":"Action","FilmDirectorId":6,"FilmDirectorName":"Christopher Nolan","IsPremiumMovie":false}]

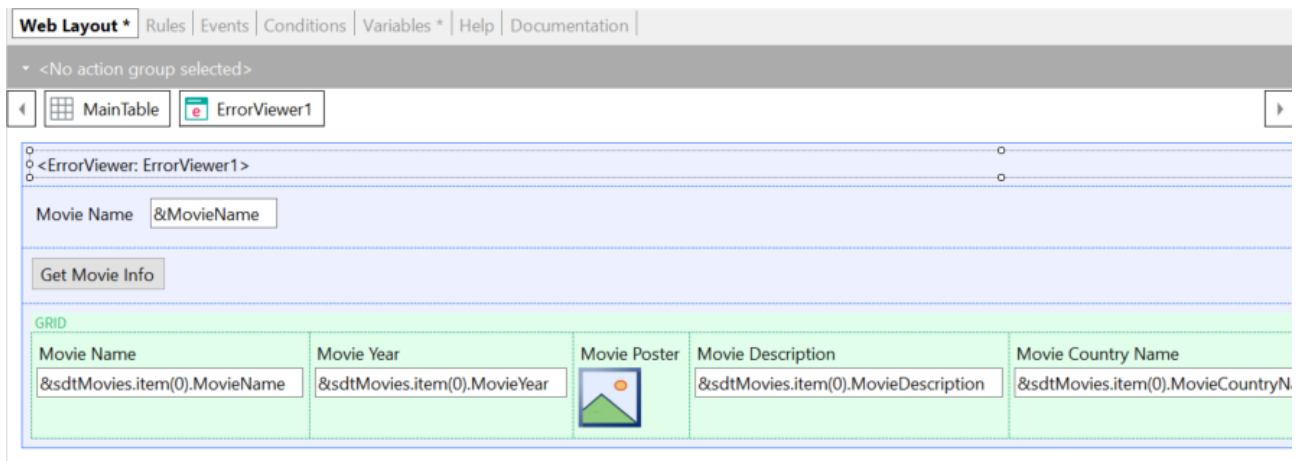
```

Now we will build a web panel that will allow us to consume the Rest service with the HttpClient data type. It can be within the same KB or in a different one. The web panel should have the following structure:

Variables:

Variable	Data type
HttpClient	HttpClient
MovieName	Character(50)
SdtMovies	Based on the SDT previously created
StringParameter	Character(200)

Web Layout



Events

```

1 Event 'Get Movie Info'
2     &httpClient.Host= "localhost"
3     &httpClient.Port = 8082
4     &httpClient.BaseUrl = "/MyCinemaNETSQLServer/rest/"
5     &httpClient.secure = 0
6     &StringParameter = '<DP_NAME>?' + 'MovieName=' + &MovieName
7     &httpClient.Execute('GET', &StringParameter)
8
9     if &httpClient.StatusCode = 200
10        msg("Everything Ok!")
11        &sdtMovies.FromJson(&httpClient.ToString())
12
13    else
14        msg("Error: " + &httpClient.StatusCode.ToString())
15    endif
16 Endevent

```

Run the application and check that the service can be consumed. Type the name of a movie or leave the field empty; in both cases, you should be able to consume the service.

NOTE: Enable the read-only property to True for each field of the SDT that is being printed on the screen.

Xtreme Cinema

Everything Ok!

Movie Name:

[GET MOVIE INFO](#)

Movie Name	Movie Year	Movie Poster	Movie Description	Movie Country Name	Category Name	Film Director Name
Avatar: Way to water	2022		Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.	United States	Fantasy	James Cameron

Xtreme Cinema

Everything Ok!

Movie Name:

[GET MOVIE INFO](#)

Movie Name	Movie Year	Movie Poster	Movie Description	Movie Country Name	Category Name	Film Director Name
Avengers: Endgame	2019		After Thanos, an intergalactic warlord, disintegrates half of the universe, the Avengers must reunite and assemble again to reinvigorate their trounced allies and restore balance.	United States	Action	Joe Russo
Catch me if you can	2002		Notorious con artist Frank Abagnale has duped people worth millions of dollars with his masterful art of deception. With his scams getting bolder, he is soon pursued by FBI agent Carl Hanratty.	United States	Comedy	Steven Spielberg
Avatar: Way to water	2022		Jake Sully and Ney'tiri have formed a family and are doing everything to stay together. However, they must leave their home and explore the regions of Pandora. When an ancient threat resurfaces, Jake must fight a difficult war against the humans.	United States	Fantasy	James Cameron
Forest Gump	1994		Forrest, a man with low IQ, recounts the early years of his life when he found himself in the middle of key historical events. All he wants now is to be reunited with his childhood sweetheart, Jenny.	United States	Drama	Robert Zemeckis