

# トランザクションにおけるルールのトリガーイベント

GeneXus™

## ルールの実行タイミング



これまでの章で、Transaction オブジェクトの [Rules] エLEMENTで、ルールの定義について説明しました。

これらのルールはどのようなタイミングで実行されたでしょうか。  
そのルールが必要となるようなタイミングで自動的に実行されました。  
これは、GeneXus がルールをトリガーするタイミングを自動的に判断していたためです。  
このため、開発者はルールを実行するタイミングを明記する必要はありません。

しかし、この GeneXus によって決定されるタイミングが、意図するタイミングではない場合があります。  
このようなケースおよび、その場合の対応方法についてこの章で説明します。

## 例：第 2 レベルの件数に基づくエラー①

名前	タイプ	デスクリプション	式
Flight	Flight	フライト	
FlightId	Id	フライト番号	
FlightDepartureAirportId	Id	出発空港番号	
～ 一部省略 ～			
AirlineDiscountPercentage	Percentage	航空会社割引率	
FlightFinalPrice	Price	フライト最終価格	$\text{FlightPrice} * (1 - \text{AirlineDiscountPerc...})$
FlightCapacity	Numeric(4.0)	フライト座席数	$\text{count}(\text{FlightSeatPosition})$
Seat	Seat	座席	
FlightSeatId	Id	座席番号	
FlightSeatChar	SeatChar	座席文字	
FlightSeatPosition	Position	座席位置	

名前	タイプ	デスクリプション	式
Flight	Flight	フライト	
FlightId	Id	フライト番号	
FlightDepartureAirportId	Id	出発空港番号	
～ 一部省略 ～			
AirlineDiscountPercentage	Percentage	航空会社割引率	
FlightFinalPrice	Price	フライト最終価格	$\text{FlightPrice} * (1 - \text{AirlineDiscountPerc...})$
FlightCapacity	Numeric(4.0)	フライト座席数	$\text{count}(\text{FlightSeatPosition})$
Seat	Seat	座席	
FlightSeatId	Id	座席番号	
FlightSeatChar	SeatChar	座席文字	
FlightSeatPosition	Position	座席位置	

フライト

フライト番号  座席数は8席よりも多く登録する必要があります。

出発空港番号  [リンク](#)

例として、第 2 レベルに入力された行数が一定以上でなければ、登録できないようにルールを定義しました。

この場合、実行したアプリケーションでは、新規登録を開始した時点で、定義したルールに基づき、エラーが発生します。

この挙動の理由は、データの新規登録を開始したため、入力行数をカウントする項目属性の式が実行され、カウントの結果、0 という値が確定したため、ルールの条件判定が行え、エラーとなりました。

もちろん、このエラーを無視し、データの登録を進めることは可能ですが、エラーのトリガーするタイミングは、適切ではありませんでした。

## 例：第 2 レベルの件数に基づくエラー②

Flight X

Structure | Web Layout | Rules | Events | Variables | Help | Documentation

```

1 Error("座席数は8席よりも多く登録する必要があります。")
2 If FlightCapacity > 0 and FlightCapacity < 8;
3

```

フライト座席数 0

座席

	座席番号	座席文字	座席位置	
-	1	A	窓側	① 座席数は8席よりも多く登録する必要があります。
-	1	B	間	① 座席数は8席よりも多く登録する必要があります。

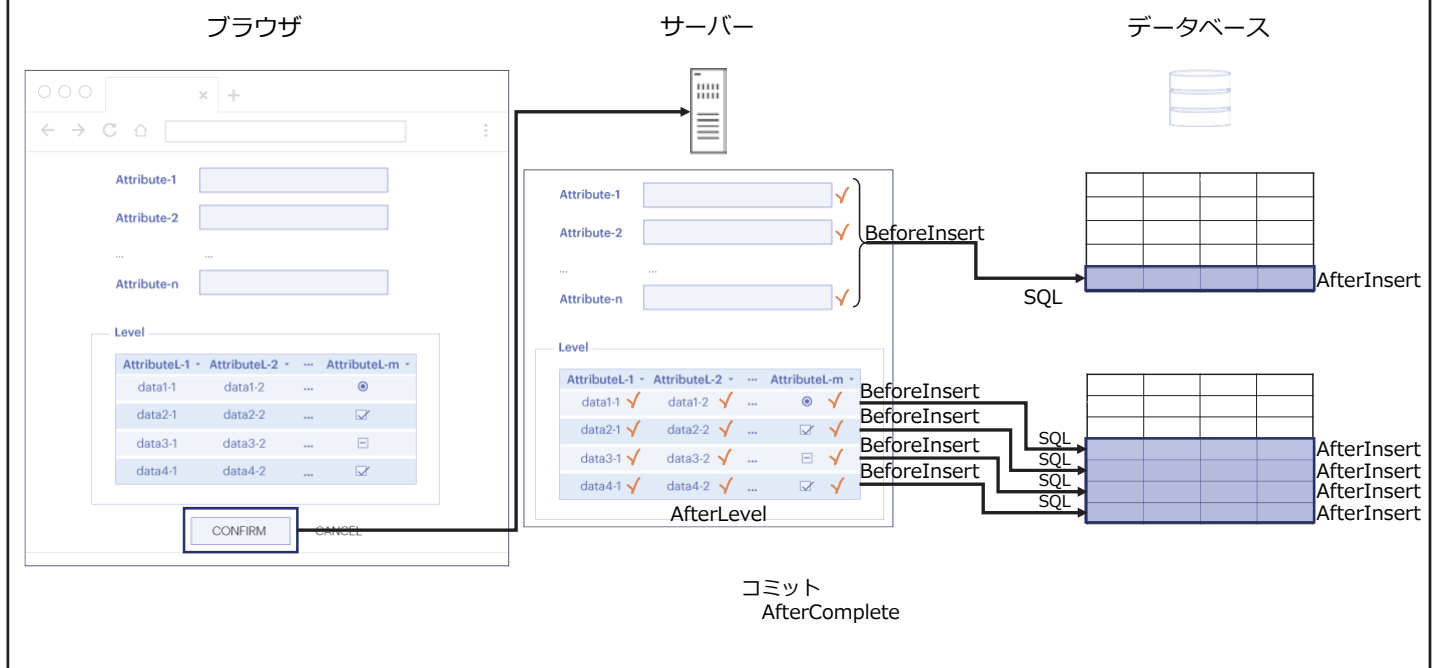
1 ページ前のルールでは、明細行の入力がない時点で、ルール実行の条件を満たしてしまったため、カウントの項目属性が「0 よりも大きい場合」という条件を付け足すことで、新規登録開始時点というタイミングでの実行を回避できます。しかし、この定義でも、適切ではありません。

適切ではない理由として、1 行目の入力を終えた時点で、カウントが 1 となり、これが設定した件数未満であれば、やはりエラーが発生してしまうためです。そして、このエラーは、入力行数がエラーの条件を満たさなくなるまで、各行に対してエラーが発生し続けます。これも、本来実行すべきルールではなく、ユーザーが利用する場合、前述の通り適切な実装とは言えません。

また、この条件の場合、第 2 レベルに一切の入力がない場合も登録できてしまいます。

では、どのようなタイミングでルールがトリガーされることが適切でしょうか。ユーザーがこの登録において入力を予定していた行数すべてを入力した後に、条件判定を行い、実行することが最適です。

## 「トリガーイベント」の利用



このようなケースに対応するため、GeneXus には、「トリガーイベント」という機能が用意されています。

トリガーイベントを利用することで、開発者がルールをトリガーするタイミングを具体的に指定することができます。

このトリガーイベントは、ブラウザ上に表示されたアプリケーションで、「実行」ボタンをクリックし、サーバーサイドで実施される特定の処理に関連して実行タイミングが決定しています。

本コースでは、まず、次の 4 種類を簡単に紹介します。

### BeforeInsert :

データの新規登録時、データベースへのSQL文発行前のタイミングを紐づけ

### AfterInsert :

データの新規登録時、テーブルにデータ格納後のタイミングを紐づけ

### AfterLevel :

指定されたレベルに対するデータ操作がすべて完了したタイミングを紐づけ

### AfterComplete :

画面からのデータ操作がすべて完了し、データベースに対するコミットの実行後のタイミングを紐づけ

このほかにもまだトリガーイベントがありますが、上記の 4 つに関する詳細を含め、別のコースで紹介しています。

## 例：第 2 レベルの件数に基づくエラー③



座席番号	座席文字	座席位置
1	A	窓側
1	B	間
1	C	通路側
1	D	通路側

今回の例として利用した、第 2 レベルの件数を条件にするルールの場合、最適なルールをトリガーするタイミングは、AfterLevel トリガーイベントを指定した状態です。

前述の通り、AfterLevel トリガーイベントは、サーバーサイドの処理の中で、第 2 レベルに関するグリッドコントロールで入力されたすべてのデータに対する処理が完了したタイミングでルールが評価されます。

「第 2 レベル全件のデータ登録が完了している」 = 「第 2 レベルのすべての件数がカウントできる状態」となるため、入力行数に対する正しい判定が行えます。また、0 よりも大きい件数という条件があるままでは、第 2 レベルがない状態で登録できてしまうため、この条件の一部を削除し、トリガーイベントを追記します。

トリガーイベントを追記する場合、必ずルールの条件定義よりも後に、「on」という文字列を書き、そのあとにトリガーイベントを記述します。残念ながら、トリガーイベントは、Ctrl + Space キーで表示される入力候補の中に表示されないため、手動で入力の必要があります。

また、AfterLevel の場合、どの「レベル」に対する処理が終わった場合かを指定するため、さらに Level というオプション節を記述し、対象のレベル内に含まれる項目属性名を記述する必要があります。

この結果、実行したアプリケーションでは、入力中にこのルールは実行されず、ユーザーによって「実行」ボタンをクリック後、内部の処理内で、第 2 レベルに対する処理が完了し、入力内容に基づき、ルールの条件判定が行われます。もし、エラーとなる場合、入力した内容については、テーブルに登録されないため、当初の目的通り、適切なタイミングで、要望したルールの実行ができる状態になります。

# GeneXus™

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