

# Reglas en Transacciones

Más reglas para definir comportamientos

*GeneXus*™ 15

## Escenario

El Cliente acumula millas por cada excursión realizada.

| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

| Name        | Type |
|-------------|------|
| Trip        | Trip |
| TripId      | Id   |
| TripDate    | Date |
| CountryId   | Id   |
| CountryName | Name |
| CityId      | Id   |
| CityName    | Name |

¿Cómo funciona la regla Add?

**Rule:**

```
Add(CustomerTripMiles, CustomerTotalMiles);
```

Consideremos que un cliente realiza varias excursiones y va acumulando millas.

Para implementar la acumulación de millas al cliente, declaramos en la transacción Customer la siguiente regla:

```
Add(CustomerTripMiles, CustomerTotalMiles);
```

¿Cómo es el comportamiento de esta regla?

## Comportamiento de la regla Add

| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

- Si se inserta una nueva excursión para el Cliente...

Add(CustomerTripMiles, CustomerTotalMiles);



- Si se elimina una excursión para el Cliente...

Add(CustomerTripMiles, CustomerTotalMiles);



La regla Add varía su comportamiento, dependiendo de la operación que se está realizando:

- Si se ingresa una nueva excursión para el cliente entonces se le suma la cantidad CustomerTripMiles a CustomerTotalMiles.
- Si se elimina una excursión del cliente entonces automáticamente se resta el valor de CustomerTripMiles a CustomerTotalMiles.

## Comportamiento de la regla Add

| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

- Si se modifica un viaje para el Cliente...

```
Add(CustomerTripMiles, CustomerTotalMiles);
```



**diff**

Resta a CustomerTotalMiles el valor anterior de CustomerTripMiles y le suma el nuevo valor

Si se modifica el valor de CustomerTripMiles asociado a una excursión del cliente, entonces automáticamente se resta el valor que tenía declarado y luego se suma el nuevo valor indicado al total de millas del cliente.

## Comportamiento de la regla Subtract (Equivalente a Add pero resta en modo Insert)

- Si se registra una nueva excursión para el Cliente....

Subtract(CustomerTripMiles, CustomerTotalMiles);



- Si se elimina una excursión para el Cliente....

Subtract(CustomerTripMiles, CustomerTotalMiles);



| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

La regla Subtract tiene un comportamiento similar al visto en la regla Add, pero **sustra**e el primer valor al segundo si se inserta una excursión.

Consideremos entonces la siguiente regla Subtract:

Subtract(CustomerTripMiles, CustomerTotalMiles);

- Si se agrega una nueva excursión para el cliente, entonces se resta el valor de CustomerTripMiles a CustomerTotalMiles.
- Si se elimina una excursión del Cliente entonces automáticamente se suma el valor de CustomerTripMiles a CustomerTotalMiles.

## Comportamiento regla Subtract

(Equivalente a Add pero resta en modo Insert)

| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

- Si se modifica una excursión para el Cliente...

Subtract(CustomerTripMiles, CustomerTotalMiles);



**diff**

Suma a CustomerTotalMiles el valor anterior de CustomerTripMiles y le resta el nuevo valor

- Si se modifica el valor de CustomerTripMiles asociado a un viaje del cliente, entonces automáticamente se suma el valor que tenía declarado y luego se resta el nuevo valor indicado.

## Combinando reglas...

El Cliente puede canjear sus millas por un premio... si es que tiene la cantidad de millas disponibles.

| Name               | Type             |
|--------------------|------------------|
| Customer           | Customer         |
| CustomerId         | Id               |
| CustomerName       | Name             |
| CustomerLastName   | Name             |
| CustomerAddress    | Address, GeneXus |
| CustomerPhone      | Phone, GeneXus   |
| CustomerEmail      | Email, GeneXus   |
| CustomerAddedDate  | Date             |
| CustomerTotalMiles | Numeric(4,0)     |
| Trip               | Trip             |
| TripId             | Id               |
| TripDate           | Date             |
| CountryId          | Id               |
| CountryName        | Name             |
| CityId             | Id               |
| CityName           | Name             |
| CustomerTripMiles  | Numeric(4,0)     |

| Name               | Type          |
|--------------------|---------------|
| Prize              | Prize         |
| PrizeId            | Id            |
| PrizeDate          | Date          |
| PrizeDescription   | Character(20) |
| PrizeMiles         | Numeric(4,0)  |
| CustomerId         | Id            |
| CustomerName       | Name          |
| CustomerLastName   | Name          |
| CustomerTotalMiles | Numeric(4,0)  |

Error("The customer doesn't have enough miles") if  
CustomerTotalMiles < 0;

Subtract(PrizeMiles, CustomerTotalMiles);

Contamos con la transacción Prize, que permite definir premios a ser canjeados por millas.

Cada premio tiene una cantidad de millas requerida para poder ser otorgado a un cliente que disponga de dicha cantidad de millas (o tenga más aún).

De modo que cuando traten de asignarle 1 cliente a 1 premio, debemos validar si al cliente le alcanzan sus millas para asignarle dicho premio. En caso de alcanzarle y quedarse con el premio, hay que restarle las millas que canjeó; y si el cliente no tiene millas suficientes, hay que anunciar el error correspondiente.

Observemos las dos reglas definidas para la transacción Prize. Dado que ambas involucran al atributo CustomerTotalMiles, siendo que una de las reglas actualiza al atributo y la otra regla evalúa su valor, GeneXus determina que primero debe ejecutar la sustracción que actualiza al atributo CustomerTotalMiles y luego evaluar qué sucedió con su valor.

Dado que la sustracción se efectúa primero, si el cliente tenía menos millas que las requeridas por el premio, el atributo CustomerTotalMiles quedará con valor negativo. Por este motivo es que la regla error evalúa si CustomerTotalMiles < 0. En caso de ocurrir esto entonces, se dispara la regla error, se emite el texto que anuncia el error y la regla subtract se deshace, es decir, que se revierte su ejecución como si no se hubiera hecho y el total de millas del cliente queda sin haberse modificado.

Si en cambio después de la ejecución del subtract, no sucede que CustomerTotalMiles haya quedado con valor negativo (CustomerTotalMiles < 0) entonces el subtract quedó efectuado.

Observemos que si eliminan un registro de premio asignado a 1 cliente, la regla subtract suma en vez de restar. O sea que le suma al total de millas del cliente (CustomerTotalMiles) la cantidad de millas correspondiente al premio (PrizeMiles).

## Regla Serial

Permite numerar automáticamente el segundo nivel de una transacción.

| Name        | Type    |
|-------------|---------|
| Country     | Country |
| CountryId   | Id      |
| CountryName | Name    |
| City        | City    |
| CityId      | Id      |
| CityName    | Name    |

¿Cómo se puede numerar?

**Regla:**  
**Serial(CityId, CountryLastLine, 1);**

Donde:

**CityId** – atributo que se numera.

**CountryLastLine** – atributo que guarda el último valor asignado.

1- Incremento de la numeración

**Solución:**

| Name            | Type         |
|-----------------|--------------|
| Country         | Country      |
| CountryId       | Id           |
| CountryName     | Name         |
| CountryLastLine | Numeric(4,0) |
| City            | City         |
| CityId          | Id           |
| CityName        | Name         |

Guarda el último valor asignado.

Por último veamos cómo podemos numerar el segundo nivel de una transacción.

Dado que la propiedad Autonumber solamente es aplicable a claves primarias simples, para numerar automáticamente un 2do nivel, contamos con la regla Serial.

Esta regla requiere definir un atributo en el 1er nivel de la transacción, que guardará el último valor asignado a las líneas del segundo nivel.

# GeneXus™

Videos

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

Documentación

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

Certificaciones

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