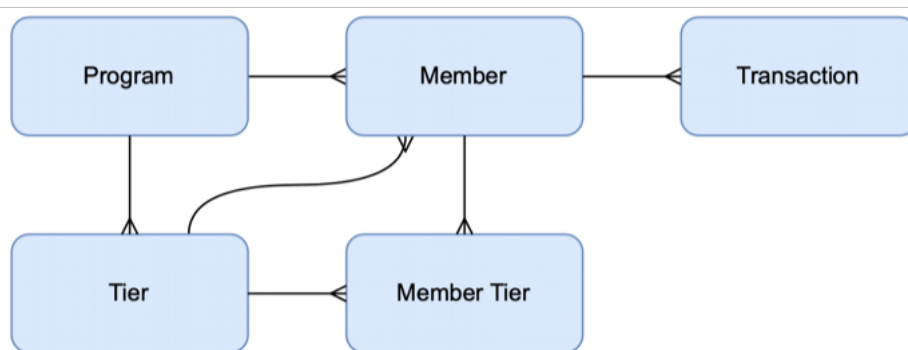


# Loyalty

## Glossary

| Object      | Description                                                                                                                             |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Program     | The placeholder that is used to setup the logic of the loyalty program. It contains Tiers and Members who joined the Loyalty program.   |
| Tier        | Represent premium service levels available to members who have achieved defined point levels.                                           |
| Member      | The customer, who joined the loyalty program.                                                                                           |
| Transaction | The record that logs the loyalty activity details (i.e. purchase), that might change the Tier of the member or attributes (points etc.) |
| Member Tier | Represents the fact of the member's tier ownership in a specific period of time. Each member can have just one active Member Tier .     |

## ERD



## The accrual logic

Once the customer has made the purchase, the Transaction record is created in Loyalty

Transaction

T0000000001

[Edit Layout](#) | [Printable View](#) | [Help for this Page](#)

Transaction Detail

EditDeleteCloneSharing

Transaction Name

T0000000001

Owner

Serhii Volynets [\[Change\]](#)

Member

Serhii V

Status

Points 1

100

Points 2

10

Points 3

Created By

Serhii Volynets, 20/08/2022, 11:24

Last Modified By

Serhii Volynets, 20/08/2022, 11:24

EditDeleteCloneSharing

This transaction has the reference to Member and contains different point type values, that must be accumulated in Member record, when the Transaction is submitted.

The field set Transaction.RollupFields defines the set of Transaction fields that will be rolled up into Member record.

So same fields with same API Names (i.e. Points\_1\_\_c, Points\_2\_\_c, Points\_3\_\_c) should exist in Member object (twin fields). When the Transaction.Status is changed to Submitted, these fields are rolled up into the Member record. This also triggers the calculation of the tier for the member.

Each Program Tier has numeric fields Min Points 1, Max Points 1, Min Points 2 ,... etc, that are used to define the boundaries of each tier applicability.

If the boundary value is not specified in the Tier record, then the boundary is not checked. For example, in this screen we defined, that Tier 1 is applicable, when Member Points 1 are between 0 and 200.

The Level field defines the priority, in which the particular Tier will be assessed for the Member. The higher is the Tier, the more restrictive criteria it should have, that's why Tiers with the higher level are assessed first.

If boundaries are applied for multiple points in the Tier record, by default all these boundaries must be met (the logical AND works).

**Tier 4**

« Back to List: Custom Object Definitions

Member Tiers [0] | Members [0]

**Tier Detail** [Edit] [Delete] [Clone]

|              |                                    |                  |                                    |
|--------------|------------------------------------|------------------|------------------------------------|
| Tier Name    | Tier 4                             | Program          | Loyalty Program                    |
| Min Points 1 | 400                                | Max Points 1     | 1,000                              |
| Min Points 2 | 40                                 | Max Points 2     | 100                                |
| Min Points 3 |                                    | Max Points 3     |                                    |
| Logic        | AND                                |                  |                                    |
| Level        | 4                                  |                  |                                    |
| Created By   | Serhii Volynets, 20/08/2022, 11:47 | Last Modified By | Serhii Volynets, 20/08/2022, 11:47 |

For example, in this screen we defined, that the Tier 4 is applicable, when Points 1 are between 400 and 1000, and Points 2 are between 40 and 100.

However, it is possible to define the custom logic for the Tier assessment using the Logic field.

**Tier 4**

« Back to List: Custom Object Definitions

Member Tiers [0] | Members [0]

**Tier Detail** [Save] [Cancel]

|              |        |              |                 |
|--------------|--------|--------------|-----------------|
| Tier Name    | Tier 4 | Program      | Loyalty Program |
| Min Points 1 | 400    | Max Points 1 | 1,000           |
| Min Points 2 | 40     | Max Points 2 | 100             |
| Min Points 3 |        | Max Points 3 |                 |
| Logic        | 1 OR 2 |              |                 |
| Level        | 4      |              |                 |

For example, if the Logic field is changed to '1 OR 2', this means, that the Tier 4 will be applicable, when Points 1 are between 400 and 1000, **OR** Points 2 are between 40 and 100.

The sequence (i.e. 1, 2) of the point type in this expression is determined by the sequence of the point type in Transaction.RollupFields fieldSet.

**RollupFields**

Save Cancel Undo Redo Field Set Properties

Transaction

Created By ID  
Last Modified By ID  
Member  
Owner ID

Quick Find Transaction Name

|                  |                    |           |                  |
|------------------|--------------------|-----------|------------------|
| Created By       | Last Modified Date | Points 2  | System Modstamp  |
| Created Date     | Member             | Points 3  | Transaction Name |
| Deleted          | Owner              | Record ID |                  |
| Last Modified By | Points 1           | Status    |                  |

Drag any of the fields above into one of the lists below.

**Available for the Field Set**

Drag and drop the fields you want administrators to have available to add to the Field Set.

**In the Field Set**

Points 1  
Points 2  
Points 3

## How to add additional point types

1. Add the field [field name] of numeric type to Transaction\_\_c object
2. Add the field [field name] of numeric type to Member\_\_c object

3. Add the field 'Min\_'[field name] of numeric type to Tier\_\_c object
4. Add the field 'Max\_'[field name] of numeric type to Tier\_\_c object
5. Add the field [field name] to Transaction.RollupFields fieldSet

## How to extend Transaction's business logic

Please use the trigger flows before insert and before update on Transaction in order to adjust the logic accordingly. The criteria to obtain the set of submitted Transactions is when the Transaction Status is Submitted, and the Transaction Status is changed (or the transaction record

## The tier change effective dates

If the member earned points enough, it can get a new Tier, which defines some benefits the member can use.

There are 3 date fields on Member Tier object, that define the member tier lifecycle:

1) **Start Date** - the effective start date, when the member starts using benefits of a new tier, it's equal to the date of new tier creation, so immediately after earning enough points, the member starts using benefits of a new tier.

2) **Qualification Date** - the date when the qualification points are deducted from member balance, and a new qualification period starts.

The Tier's field Qualification\_Date\_Determined\_By defines the algorithm of how the Qualification Date is populated. Possible field value are :

| Value                    | Description                                                                                                                                                                                                                 |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Rolling                  | The Qualification Date starts immediately                                                                                                                                                                                   |
| Member Join Anniversary  | The Qualification Date is defined by the Member's program's join date.                                                                                                                                                      |
| Next Unit of Measure Day | If the Tier's Qualification UoM field is Year, then the Qualification date is 1 January of the next year. If the Tier Qualification UoM field value is Month, then the Qualification Date is the 1st day of the next month. |

3) **End Date** - the date when the tier's effect is finished. This also finishes the qualification period , so if the customer didn't earn enough qualification points until the tier End Date, then member tier is decreased. The End Date depends on Qualification Date. The Tier's field End\_Date\_Determined\_By defines the algorithm of how the End Date is populated. Possible value are:

| Value                    | Description                                                                                                                                                                                                                    |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Duration                 | The End Date is determined by Tier's Qualification Duration and Qualification UoM fields. For example, when:<br>Qualification Duration=6,<br>Qualification UoM=Month,<br>Then the End Date is Qualification Date plus 6 Months |
| Next Unit of Measure Day | The End Date is the Next Unit of Measure date from the Effective Date. For example, if the tier's Qualification Date is 10 June 2022 and the Qualification UoM=Year, then the End Date will be 1 Jan 2023                      |

## Scheduling the qualification and expiration jobs

In order to schedule loyalty jobs, please execute in developer console:

```
System.schedule('LoyaltyBatchJobs', '0 0 1 * * ?', new LOYL.BatchJobs());
```

This will run the loyalty jobs daily at 1 AM.

## The tier qualification points deduction.

When the qualification date is reached, the points are deducted from the member balance.

The pick list field `Tier.Qualification_Points_Expire` controls, how many points will be deducted.

Possible values are:

| Value         | Description                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Minimum Bound | The amount of points to be deducted, is determined by Minimum required amount of qualification points for the tier (for example Tier.Min Points 1). |
| All           | Qualification points are fully reset to 0                                                                                                           |
| None          | Member points are not deducted at all and remain the same.                                                                                          |

Once the Tier is qualified, it can not be downgraded until the Member Tier.End Date is reached.

## Tier expiration and re-qualification.

Once the Member Tier.End Date is reached, the Active Tier is deactivated, and a new Member Tier is determined, so if there's not enough points for the actual tier, it gets downgraded.

## Customer Notifications

The Member Tier record with Active = true, contains details to encourage the Member to keep spending.

In order to notify the Member, when the actual tier to be ended,

The *Member Tier.Remaining Days* field can be used. It shows, how many days left until the tier is ended.

Member Tier

Serhii V\_Tier 1

« Back to List: Custom Object Definitions

Customize Page | Edit Layout | Printable View | Help for this Page

Members [1]

Member Tier Detail

EditDeleteClone

|                              |                                     |                              |                          |
|------------------------------|-------------------------------------|------------------------------|--------------------------|
| Member Tier Name             | Serhii V_Tier 1                     | Start Date                   | 20/08/2022, 01:00        |
| Member                       | Serhii V                            | End Date                     | 31/08/2022, 17:15        |
| Active                       | <input checked="" type="checkbox"/> | Qualification Date           | 20/08/2022, 01:00        |
| Tier                         | Tier 1                              | Next Tier                    | Tier 2                   |
| Requalify Remaining Points 1 | 70                                  | Next Tier Remaining Points 1 | 370                      |
| Requalify Remaining Points 2 | 0                                   | Next Tier Remaining Points 2 | 30                       |
| Requalify Remaining Points 3 | 0                                   | Next Tier Remaining Points 3 | 0                        |
| Remaining Days               | 6                                   | Qualification Applied Flag   | <input type="checkbox"/> |

In order to determine, how many points are needed to be earned by the customer for the actual tier re-qualification by the end of it, *Member Tier.Requalify Remaining Points 1,2,3* formula fields can be used.

## Requalify Remaining Points 1

[Back to Member Tier](#)

## Custom Field Definition Detail

Edit

Set Field-Level Security

## Field Information

|                           |                                    |
|---------------------------|------------------------------------|
| Field Label               | Requalify Remaining Points 1       |
| Field Name                | Requalify_Remaining_Points_1       |
| API Name                  | Requalify_Remaining_Points_1__c    |
| Description               |                                    |
| Help Text                 |                                    |
| Data Owner                |                                    |
| Field Usage               |                                    |
| Data Sensitivity Level    |                                    |
| Compliance Categorization |                                    |
| Created By                | Serhii Volynets, 24/08/2022, 19:00 |

## Formula Options

|                |                                                                                           |
|----------------|-------------------------------------------------------------------------------------------|
| Data Type      | Formula  |
| Decimal Places | 0                                                                                         |

IF( Tier\_\_r.Min\_Points\_1\_\_c > Member\_\_r.Points\_1\_\_c, Tier\_\_r.Min\_Points\_1\_\_c - Member\_\_r.Points\_1\_\_c ,0)

In order to extend the logic on any other custom point types, the formula should look like:

*IF( Tier\_\_r.Min\_Points\_1\_\_c > Member\_\_r.Points\_1\_\_c, Tier\_\_r.Min\_Points\_1\_\_c - Member\_\_r.Points\_1\_\_c ,0)*

In order to determine, how many points left to earn the next Tier, *Member Tier.Next Tier Remaining Points 1,2,3* can be used.

In order to extend the logic on any other custom point types, the formula should look like:

*IF( Next\_Tier\_\_r.Min\_Points\_1\_\_c > Member\_\_r.Points\_1\_\_c, Next\_Tier\_\_r.Min\_Points\_1\_\_c - Member\_\_r.Points\_1\_\_c ,0)*

## Next Tier Remaining Points 1

[Back to Member Tier](#)

## Custom Field Definition Detail

Edit

Set Field-Level Security

View

## Field Information

|                           |                                    |
|---------------------------|------------------------------------|
| Field Label               | Next Tier Remaining Points 1       |
| Field Name                | Next_Tier_Remaining_Points_1       |
| API Name                  | Next_Tier_Remaining_Points_1__c    |
| Description               |                                    |
| Help Text                 |                                    |
| Data Owner                |                                    |
| Field Usage               |                                    |
| Data Sensitivity Level    |                                    |
| Compliance Categorization |                                    |
| Created By                | Serhii Volynets, 24/08/2022, 19:08 |

## Formula Options

|                |                                                                                             |
|----------------|---------------------------------------------------------------------------------------------|
| Data Type      | Formula  |
| Decimal Places | 0                                                                                           |

IF( Next\_Tier\_\_r.Min\_Points\_1\_\_c > Member\_\_r.Points\_1\_\_c, Next\_Tier\_\_r.Min\_Points\_1\_\_c - Member\_\_r.Points\_1\_\_c ,0)