EXAMPLES OF VALIDATION RULES

Summary

Review examples of validation rules for various types of apps that you can use and modify for your own purposes. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record.

Review examples of validation rules for various types of apps that you can use and modify for your own purposes. *Validation rules* verify that the data a user enters in a record meets the standards you specify before the user can save the record.

Use the following samples for validation rules in Salesforce and Force.com AppExchange apps, including:

Sample Account Address Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Canadian Billing Postal Code

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is in the correct format if `Billing Country` is Canada.

Formula:	`AND(`
	`OR(BillingCountry = "CAN", BillingCountry = "CA",`
	`BillingCountry = "Canada"),`
	`NOT(REGEX(BillingPostalCode,`
	`"((?i)[ABCEGHJKLMNPRSTVXY]\\d[A-Z]?\\s?\\d[A-Z]\\d)?"))`
	`)`

Error Message:	Canadian postal code must be in A9A 9A9 format.
Error Location:	**Billing Zip/Postal Code**

Billing Zip Code Is in Billing State

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is valid by looking up the first five characters of the value in a custom object called Zip_Code__c that contains a record for every valid zip code in the US. If the zip code is not found in the Zip_Code__c object, or the `Billing State` does not match the corresponding State_Code__c in the Zip_Code__c object, an error is displayed.
Formula:	`VLOOKUP( $ObjectType.Zip_Code__c.Fields.City__c , $ObjectType.Zip_Code__c.Fields.Name , LEFT(BillingPostalCode,5)) <> BillingCity`
Error Message:	Billing Zip Code does not exist in specified Billing State.
Error Location:	**Billing Zip/Postal Code**

US Billing Zip Code

Field	Value
Description:	Validates that the account `Billing Zip/Postal Code` is in 99999 or 99999-9999 format if `Billing Country` is USA or US.
Formula:	`AND( OR(BillingCountry = "USA", BillingCountry = "US"), NOT(REGEX(BillingPostalCode, "\\d{5}(-\\d{4})?")) )` Note: This example uses the REGEX function; see Shipping Zip Code if you are not familiar with regular expressions.
Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	**Billing Zip/Postal Code**

Shipping Zip Code

Field	Value
Description:	Validates that the account `Shipping Zip/Postal Code` is in 99999 or 99999-9999 format if `Shipping Country` is USA or blank.
Formula:	

```
AND(
 OR(ShippingCountry = "USA",
ISBLANK(ShippingCountry)),
  OR(
   AND(LEN(ShippingPostalCode) <>5,
      LEN(ShippingPostalCode) <> 10),
   NOT(CONTAINS("0123456789",
      LEFT( ShippingPostalCode, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 2, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 3, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 4, 1))),
   NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 5, 1))),
   AND(
    LEN(ShippingPostalCode) = 10,
   OR(
    MID( ShippingPostalCode , 6, 1) <> "-",
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 7, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 8, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 9, 1))),
    NOT(CONTAINS("0123456789",
      MID( ShippingPostalCode , 10, 1)))
   )
  )
 )
)
```

Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Tip: You can also validate zip codes using a regular expression; for an example of a formula using a regular expression, see "REGEX" in the Salesforce Help.

Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	**Shipping Zip/Postal Code**

Valid Billing State (US)

Field	Value
Description:	Validates that the account `Billing State/Province` is a valid two-character abbreviation if `Billing Country` is US, USA, or blank.

Formula:	`AND (`
	`OR(BillingCountry = "US", BillingCountry="USA",`
	`ISBLANK(BillingCountry)),`
	`OR(`
	`LEN(BillingState) < 2,`
	`NOT(`
	`CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:"`
	`&`
	`"IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:"`
	`&`
	`"NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:"`
	`&`
	`"WA:WV:WI:WY", BillingState)`
	`)))`
	
	Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Error Message:	A valid two-letter state code is required.
Error Location:	`Billing State/Province`

Valid Billing Province (Canada)

Field	Value
Description:	Validates that the account `Billing State/Province` is a valid two-character abbreviation if `Billing Country` is CA or CAN.

Formula:	`AND (`
	`  OR(BillingCountry = "CA", BillingCountry="CAN"),`
	`  OR(`
	`  LEN(BillingState) < 2,`
	`  NOT(`
	`  CONTAINS("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT",`
	`  BillingState)`
	`)))`

Error Message:	A valid two-letter province code is required.
Error Location:	`Billing State/Province`

Valid Shipping State

Field	Value
Description:	Validates that the account `Shipping State/Province` is a valid two-character abbreviation if `Shipping Country` is US, USA, or blank.

Formula:	`AND (`
	`OR(ShippingCountry = "US", ShippingCountry="USA",`
	`ISBLANK(ShippingCountry)),`
	`OR(`
	`LEN(ShippingState) < 2,`
	`NOT(`
	`CONTAINS("AL:AK:AZ:AR:CA:CO:CT:DE:DC:FL:GA:HI:ID:" &`
	`"IL:IN:IA:KS:KY:LA:ME:MD:MA:MI:MN:MS:MO:MT:NE:NV:NH:"`
	`&`
	`"NJ:NM:NY:NC:ND:OH:OK:OR:PA:RI:SC:SD:TN:TX:UT:VT:VA:"`
	`&`
	`"WA:WV:WI:WY", ShippingState)`
	`)))`
	
	Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."

Error Message:	A valid two-letter state abbreviation is required.
Error Location:	`Shipping State/Province`

Valid Shipping Province (Canada)

Field	Value
Description:	Validates that the account `Shipping State/Province` is a valid two-character abbreviation, if `Billing Country` is CA or CAN.

Formula:	`AND (`
	`OR(ShippingCountry = "CA", ShippingCountry="CAN"),`
	`OR(`
	`LEN(ShippingState) < 2,`
	`NOT(`
	`CONTAINS("AB:BC:MB:NB:NL:NT:NS:NU:ON:PC:QC:SK:YT",`
	`ShippingState)`
	`)))`

Error Message:	A valid two-letter province abbreviation is required.
Error Location:	`Shipping State/Province`

Valid Billing Country

Field	Value
Description:	Validates that the account `Billing Country` is a valid ISO 3166 two-letter code.

Formula:	`OR(`
	`LEN(BillingCountry) = 1,`
	`NOT(`
	`CONTAINS(`
	`"AF:AX:AL:DZ:AS:AD:AO:AI:AQ:AG:AR:AM:" &`
	`"AW:AU:AZ:BS:BH:BD:BB:BY:BE:BZ:BJ:BM:BT:BO:" &`
	`"BA:BW:BV:BR:IO:BN:BG:BF:BI:KH:CM:CA:CV:KY:" &`
	`"CF:TD:CL:CN:CX:CC:CO:KM:CG:CD:CK:CR:CI:HR:" &`
	`"CU:CY:CZ:DK:DJ:DM:DO:EC:EG:SV:GQ:ER:EE:ET:FK:" &`
	`"FO:FJ:FI:FR:GF:PF:TF:GA:GM:GE:DE:GH:GI:GR:GL:" &`
	`"GD:GP:GU:GT:GG:GN:GW:GY:HT:HM:VA:HN:HK:HU:" &`
	`"IS:IN:ID:IR:IQ:IE:IM:IL:IT:JM:JP:JE:JO:KZ:KE:KI:" &`
	`"KP:KR:KW:KG:LA:LV:LB:LS:LR:LY:LI:LT:LU:MO:MK:" &`
	`"MG:MW:MY:MV:ML:MT:MH:MQ:MR:MU:YT:MX:FM:MD:MC:" &`
	`"MC:MN:ME:MS:MA:MZ:MM:MA:NR:NP:NL:AN:NC:NZ:NI:" &`
	`"NE:NG:NU:NF:MP:NO:OM:PK:PW:PS:PA:PG:PY:PE:PH:" &`
	`"PN:PL:PT:PR:QA:RE:RO:RU:RW:SH:KN:LC:PM:VC:WS:" &`
	`"SM:ST:SA:SN:RS:SC:SL:SG:SK:SI:SB:SO:ZA:GS:ES:" &`
	`"LK:SD:SR:SJ:SZ:SE:CH:SY:TW:TJ:TZ:TH:TL:TG:TK:" &`
	`"TO:TT:TN:TR:TM:TC:TV:UG:UA:AE:GB:US:UM:UY:UZ:" &`
	`"VU:VE:VN:VG:VI:WF:EH:YE:ZM:ZW",`
	`BillingCountry)))`

Error Message:	A valid two-letter country code is required.
Error Location:	`Billing Country`

Sample Account Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Account Number Is Numeric

Field	Value
Description:	Validates that the `Account Number` is numeric if not blank.

Formula:	`AND(`
	`  ISBLANK(AccountNumber),`
	`  NOT(ISNUMBER(AccountNumber))`
	`)`

Error Message:	Account Number is not numeric.
Error Location:	`Account Number`

Account Number Length

Field	Value
Description:	Validates that the `Account Number` is exactly seven digits (if it is not blank). The number seven is simply illustrative. You can change this to any number you like.

Formula:	`AND(`
	`  ISBLANK(AccountNumber),`
	`  LEN(AccountNumber) <> 7`
	`)`

Error Message:	Account Number must be seven digits.
Error Location:	`Account Number`

Annual Revenue Range

Field	Value
Description:	Validates that the account `Annual Revenue` is not negative and does not exceed $100 billion. This limit is designed to catch typos.

Formula:	`OR(`
	`  AnnualRevenue < 0,`
	`  AnnualRevenue > 100000000000`
	`)`

Error Message:	Annual Revenue cannot exceed 100 billion.
Error Location:	`Annual Revenue`

Sample Call Center Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Conditionally Require Description When Case Reason is "Other"

Field	Value
Description:	Validates that a custom field called `Other Reason` contains a value if a case has a `Case Reason` of "Other."

Formula:	`AND(`
	`  ISPICKVAL( Reason, "Other" ),`
	`  ISBLANK(Other_Reason__c)`
	`)`

Error Message:	Description of Other Reason is required.
Error Location:	`Other Reason`

Prevent Open Cases from Being Reset to New

Field	Value
Description:	If a case is already open, prevents the `Status` from being changed back to "New."

Formula:	`AND(`
	`  ISCHANGED( Status ),`
	`  NOT(ISPICKVAL(PRIORVALUE( Status ), "New")),`
	`  ISPICKVAL( Status, "New")`
	`)`

Error Message:	Open case Status cannot be reset to New.
Error Location:	`Status`

Restrict Status of Re-Opened Cases

Field	Value
Description:	Validates that the case `Status` is "Re-opened" when a closed case is opened again.
Formula:	`AND( ISCHANGED( Status ), OR( ISPICKVAL(PRIORVALUE( Status ), "Closed"), ISPICKVAL(PRIORVALUE( Status ), "Closed in SSP")), NOT( ISPICKVAL( Status, "Re-Opened")) )`
Error Message:	Closed case can only be changed to "Re-opened."
Error Location:	`Status`

Prevent Case Milestone Completion After Cases Are Closed

Field	Value
Description:	Validates that a milestone's `Completion Date` can't occur after the case's `Status` is Closed.
Formula:	`Case.IsClosed = true`
Error Message:	You can't complete a milestone after a case is closed.
Error Location:	`Top of Page`

Prevent Case Milestone Completion Before Case Creation Dates

Field	Value
Description:	Validates that the milestone's `Completion Date` has occurred after the case's `Date/Time Opened`.
Formula:	`CompletionDate >= Case.CreatedDate && CompletionDate <= Case.ClosedDate`
Error Message:	The milestone Completion Date must occur after the date the case was created and before the case was closed.
Error Location:	`Top of Page`

Sample Community Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Preventing Offensive Language in Questions

Field	Value
Description:	Prevents users from entering offensive language in the `Title` and `Description` fields when asking a question.
Formula:	`OR(CONTAINS(Title, 'darn'), CONTAINS(Body, 'darn'))`
Error Message:	Question title or description contains offensive language.

Preventing Offensive Language in Replies

Field	Value
Description:	Prevents users from entering offensive language when replying to a question.
Formula:	`OR(CONTAINS(Body, 'darn'), CONTAINS(Body, 'dang'))`
Error Message:	Reply contains offensive language.

Preventing Offensive Language in Ideas

Field	Value
Description:	Prevents users from entering offensive language in the `Title` and `Description` fields when posting an idea.
Formula:	`OR(CONTAINS(Title, 'darn'), CONTAINS(Body, 'darn'))`
Error Message:	Idea title or description contains offensive language.

Preventing Offensive Language in Idea Comments

Field	Value
Description:	Prevents users from entering offensive language when posting a comment.
Formula:	`OR(CONTAINS(CommentBody , 'darn'), CONTAINS(CommentBody, 'dang'))`
Error Message:	Comment contains offensive language.

Sample Contact Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Mailing Address Fields Are Required

Field	Value
Description:	Validates that the contact `Mailing Street`, `Mailing City`, and `Mailing Country` are provided.
Formula:	`OR( ISBLANK( MailingStreet ), ISBLANK( MailingCity ), ISBLANK( MailingCountry ) )`
Error Message:	Mailing Street, City, and Country are required.
Error Location:	`Top of Page`

Mailing Street Is Required

Field	Value
Description:	Validates that the contact `Mailing Street` is provided.
Formula:	`ISBLANK( MailingStreet )`
Error Message:	Mailing Street is required.
Error Location:	`Mailing Street`

Mailing Zip Code

Field	Value
Description:	Validates that the contact `Mailing Zip/Postal Code` is in 99999 or 99999-9999 format if `Mailing Country` is USA or blank.

Formula:	
	``
	AND(
	 OR(MailingCountry = "USA", ISBLANK(MailingCountry)),
	
	  OR(
	  AND(LEN(MailingPostalCode) <>5,
	  LEN(MailingPostalCode) <> 10),
	  NOT(CONTAINS("0123456789",
	  LEFT(MailingPostalCode, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 2, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 3, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 4, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 5, 1))),
	  AND(
	  LEN(MailingPostalCode) = 10,
	  OR(
	  MID(MailingPostalCode , 6, 1) <> "-",
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 7, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 8, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 9, 1))),
	  NOT(CONTAINS("0123456789",
	  MID(MailingPostalCode , 10, 1)))
	  )
	  )
	)
	)
	``
	
	Note: This example interprets a blank country as US. To use this example with other countries, remove the clause that checks the length of the country field. Also, validation rule criteria are case sensitive, so this rule is only enforced when the country is blank or "USA" in all capital letters. The rule is not enforced when the country is "usa."
	
	Tip: You can also validate zip codes using a regular expression; for an example of a formula using a regular expression, see "REGEX" in the Salesforce Help.

Error Message:	Zip code must be in 99999 or 99999-9999 format.
Error Location:	`Mailing Zip/Postal Code`

Phone Number Has International Format

Field	Value
Description:	Validates that the `Phone` number begins with a plus sign (+) for country code. Note that this validation rule conflicts with the ten-digit rule.
Formula:	`LEFT(Phone, 1) <> "+"`
Error Message:	Phone number must begin with + (country code).
Error Location:	`Phone`

US Phone Number Has Ten Digits

Field	Value
Description:	Validates that the `Phone` number is in (999) 999-9999 format. This works by using the REGEX function to check that the number has ten digits in the (999) 999-9999 format.
Formula:	`NOT(REGEX(Phone, "\\D*?(\\d\\D*?){10}"))`
Error Message:	US phone numbers should be in this format: (999) 999-9999.
Error Location:	`Phone`

Sample Cross Object Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Discounts Must Be Within Range

This example consists of three validation rules on opportunity products. The examples below work together to help you manage discount amounts for products and require a custom percent field on opportunity products called `Line Discount`. The examples below also require you to use price books and customize the `Product Family` field to include the following values: *`Software`*, *`Consulting`*, and *`Training`*.

Software Discounts

Field	Value
Description:	Prevents users from saving software products with a discount over 10 percent.

Field	Value
Formula:	`AND(Line_Discount__c > 0.10, ISPICKVAL(Product2.Family, "Software"))`
Error Message:	The discount must be 10% or less for software products.
Error Location:	`Line Discount`

Consulting Discounts

Field	Value
Description:	Prevents users from saving consulting products with a discount over 15 percent.
Formula:	`AND(Line_Discount__c > 0.15, ISPICKVAL(Product2.Family, "Consulting"))`
Error Message:	The discount must be 15% or less for consulting products.
Error Location:	`Line Discount`

Training Discounts

Field	Value
Description:	Prevents users from saving training products with a discount over 20 percent.
Formula:	`AND(Line_Discount__c > 0.20, ISPICKVAL(Product2.Family, "Training"))`
Error Message:	The discount must be 20% or less for training products.
Error Location:	`Line Discount`

Prevent Changing Opportunity Products on Closed Opportunities

This example consists of two validation rules: one on opportunity products and another on opportunities.

Field	Value
Description:	Prevents users from editing opportunity products after an opportunity is closed. Create the following validation rule example on opportunity products.
Formula:	`OR(ISPICKVAL(Opportunity.StageName, "Closed Won"), ISPICKVAL(Opportunity.StageName, "Closed Lost"))`
Error Message:	Cannot change opportunity products for closed opportunities.
Error Location:	`Top of Page`

The following validation rule is on opportunities.

Field	Value
Description:	Prevents users from deleting opportunity products after an opportunity is closed. Create the following validation rule example on opportunities. It uses a custom roll-up summary field on opportunities that counts the number of opportunity products on an opportunity.
Formula:	`AND(OR(ISPICKVAL(StageName, "Closed Won"), ISPICKVAL(StageName, "Closed Lost")), Number_of_Line_Items__c < PRIORVALUE(Number_of_Line_Items__c) )`
Error Message:	Cannot delete opportunity products for closed opportunities.
Error Location:	`Top of Page`

Prevent Saving a Case When Account Does Not Have Support

Field	Value
Description:	Prevents users from saving a case for an account that does not have support. This example assumes you have a custom checkbox field on accounts called `Allowed Support` that tracks if the account has support.

Field	Value
Formula:	`Account.Allowed_Support__c = FALSE`
Error Message:	Unable to create cases for this account because it is not signed up for support.
Error Location:	`Top of Page`

Prevent Saving a Case When Contact is No Longer with the Company

Field	Value
Description:	Prevents users from saving an open case associated with a contact that is no longer with the company. This example uses a custom checkbox field on contacts called `No Longer With Company`.
Formula:	`AND(Contact.Not_Longer_With_Company__c, NOT(IsClosed))`
Error Message:	Unable to save this case because the related contact is no longer with the company. To continue, choose another contact.
Error Location:	`Contact Name`

Sample Date Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Date Must Be a Weekday

Field	Value
Description:	Validates that the value of a custom date field is a weekday (not Saturday or Sunday).

Formula:	`CASE(MOD( My_Date__c - DATE(1900, 1, 7), 7),`
	`0, 0,`
	`6, 0,`
	`1) = 0`

Error Message:	Date must be a weekday.
Error Location:	`My Date`

Date Must Be a Weekend Day

Field	Value
Description:	Validates that the value of a custom date field is a Saturday or Sunday.

Formula:	`CASE( MOD( My_Date__c - DATE(1900, 1, 7), 7),`
	`0, 1,`
	`6, 1,`
	`0) = 0`

Error Message:	Date must be a weekend day.
Error Location:	`My Date`

Date Must Be in the Current Month

Field	Value
Description:	Validates that a custom date field contains a date within the current month and year.

Formula:	`OR (`
	`YEAR( My_Date__c ) <> YEAR ( TODAY() ),`
	`MONTH( My_Date__c ) <> MONTH ( TODAY() )`
	`)`

Error Message:	Date must be in the current month.
Error Location:	`My Date`

Date Must Be in the Current Year

Field	Value
Description:	Validates that a custom date field contains a date within the current year.
Formula:	`YEAR( My_Date__c ) <> YEAR ( TODAY() )`
Error Message:	Date must be in the current year.
Error Location:	`My Date`

Date Must Be the Last Day of the Month

Field	Value
Description:	Validates whether a custom field called `My Date` is the last day of the month. To do this, it determines the date of the first day of the next month and then subtracts 1 day. It includes special case logic for December.
Formula:	`DAY(My_Date__c) <> IF(Month(My_Date__c)=12, 31, DAY(DATE(YEAR(My_Date__c),MONTH(My_Date__c)+1,1) - 1))`
Error Message:	Date must be the last day of the month.
Error Location:	`My Date`

Date Must Be Within One Year of Today

Field	Value
Description:	Validates whether a custom field called `Follow-Up Date` is within one year of today's date. This example assumes a 365 day year. (It does not handle leap years.)
Formula:	`Followup_Date__c - TODAY() > 365`
Error Message:	Follow-Up Date must be within one year of today.
Error Location:	`Follow-Up Date`

Day of Month Cannot Be Greater Than 15

Field	Value
Description:	Validates that a custom field called `Begin Date` contains a date in the first 15 days of the specified month.
Formula:	`DAY( Begin_Date__c ) > 15`
Error Message:	Begin Date cannot be after the 15th day of month.
Error Location:	`Begin Date`

End Date Cannot Be Before Begin Date

Field	Value
Description:	Validates that a custom field called `End Date` does not come before another custom field called `Begin Date`.
Formula:	`Begin_Date__c > End_Date__c`
Error Message:	End Date cannot be before Begin Date.
Error Location:	`Begin Date`

Expiration Date Cannot Be Before Close Date

Field	Value
Description:	Validates that a custom field called `Expiration Date` does not come before `Close Date`.
Formula:	`Expiration_Date__c < CloseDate`
Error Message:	Expiration Date cannot be before Close Date.
Error Location:	`Expiration Date`

Sample Number Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Time Cards Must Total 40 Hours

Field	Value
Description:	Ensures that users cannot save a time card record with more than 40 hours in a work week. This example requires five custom fields on your custom object, one for each day of work.
Formula:	`Monday_Hours__c + Tuesday_Hours__c + Wednesday_Hours__c + Thursday_Hours__c + Friday_Hours__c > 40`
Error Message:	Your total hours cannot exceed 40.
Error Location:	`Top of Page`

Number Cannot Be Negative

Field	Value
Description:	Validates that a custom field called `Hours Worked` is not a negative number.
Formula:	`Hours_Worked__c < 0`
Error Message:	Hours Worked cannot be less than zero.
Error Location:	`Hours Worked`

Number Must Be Even

Field	Value
Description:	Validates that a custom field called `Ark Passengers` is a non-negative even number.
Formula:	`OR( Ark_Passengers__c < 0, MOD( Ark_Passengers__c, 2) <> 0 )`
Error Message:	Ark Passengers must be a positive even number.
Error Location:	`Ark Passengers`

Number Must Be Odd

Field	Value
Description:	Validates that a custom field called `Socks Found` is a non-negative odd number.
Formula:	`OR( Socks_Found__c < 0, MOD( Socks_Found__c, 2) = 0 )`
Error Message:	Socks Found must be an odd number.
Error Location:	`Socks Found`

Number Must Be a Multiple of Five

Field	Value
Description:	Validates that a custom field called `Multiple of 5` is a multiple of five.
Formula:	`MOD( Multiple_of_5__c, 5) <> 0`
Error Message:	Number must be a multiple of five.
Error Location:	`Multiple of 5`

Number Must Be an Integer

Field	Value
Description:	Validates that a custom field called `My Integer` is an integer.
Formula:	`FLOOR( My_Integer__c) <> My_Integer__c`
Error Message:	This field must be an integer.
Error Location:	`My Integer`

Number Must Be Between -50 and 50

Field	Value
Description:	Validates that a custom field called `Volume` is between -50 and 50.
Formula:	`ABS( Volume__c) > 50`
Error Message:	Volume must be between -50 and 50.
Error Location:	`Volume`

Number Range Validation

Field	Value
Description:	Validates that the range between two custom fields, `Salary Min` and `Salary Max`, is no greater than $20,000.
Formula:	`(Salary_Max__c - Salary_Min__c) > 20000`
Error Message:	Salary range must be within $20,000. Adjust the Salary Max or Salary Min values.
Error Location:	`Salary Max`

Percentage Must Be Between Zero and 100

Field	Value
Description:	Validates that a custom field called `Mix Pct` is between 0 and 100%. Note that percent fields are expressed divided by 100 in formulas (100% is expressed as 1; 50% is expressed as 0.5).

Formula:	`OR(`
	`  Mix_Pct__c > 1.0,`
	`  Mix_Pct__c < 0.0`
	`)`

Error Message:	Mix Pct must be between 0 and 100%.
Error Location:	`Mix Pct`

Sample Opportunity Management Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Conditionally-Required Field Based on Opportunity Stage

Field	Value
Description:	Validates that a custom field called `Delivery Date` is provided if an opportunity has advanced to the Closed Won or Negotiation/Review stage.

Formula:	`AND (`
	`  OR (`
	`  ISPICKVAL(StageName, "Closed Won"),`
	`  ISPICKVAL(StageName, "Negotiation/Review")),`
	`ISBLANK(Delivery_Date__c)`
	`)`

Error Message:	Delivery Date is required for this stage.
Error Location:	`Delivery Date`

Close Date Cannot Be Prior to Current Month

Field	Value
Description:	Validates that the `Close Date` of an opportunity is not within a month prior to the current month. Note the use of ISNEW and ISCHANGED in this formula to ensure the condition is only checked when the opportunity is being created or the `Close Date` field is modified subsequently.

Formula:	`AND(`
	`  OR (`
	`  ISNEW(),`
	`  ISCHANGED( CloseDate )),`
	`  CloseDate < DATE( YEAR(TODAY()),`
	`MONTH(TODAY()), 1) )`

Error Message:	Close Date cannot be prior to current month.
Error Location:	`Close Date`

Close Date Must Be a Future Date

Field	Value
Description:	Ensures that users do not change the `Close Date` of an opportunity to a day in the past.
Formula:	`SampleDate < TODAY()`
Error Message:	Close Date cannot be a day in the past.
Error Location:	`Close Date`

Discounts on Opportunities

Field	Value
Description:	Validates that a custom discount percent field is between 0 and 40%.
Formula:	`OR(Discount_Rate__c < 0, Discount_Rate__c > 0.40)`
Error Message:	The Discount Rate must not exceed 40%.
Error Location:	`Discount Rate`

High-Value Opportunity Must Be Approved Before Closed

Field	Value
Description:	Opportunities with amounts greater than $50,000 require that a custom checkbox field called `Approved` is checked in order to change the stage to Closed Won or Closed Lost. To automate this, set field-level security on the `Approved` checkbox so that it can only be checked via a custom approval process (Enterprise Edition, Unlimited Edition, or Performance Edition).

Formula:	`AND(`
	`  OR(`
	`  ISPICKVAL(StageName,"Closed Won"),`
	`  ISPICKVAL(StageName,"Closed Lost")),`
	`(Amount > 50000),`
	`NOT(ISPICKVAL(Approval_Status__c ,"Approved")))`

Error Message:	All high-value opportunities must be approved for closure. Click the Request Close button.
Error Location:	`Top of Page`

Opportunity Amount Cannot Exceed $10 Million

Field	Value
Description:	Validates that opportunity `Amount` is positive and no more than $10 million. This limit is designed to catch typos.

Formula:	`OR(`
	`  Amount < 0,`
	`  Amount > 10000000`
	`)`

Error Message:	Amount cannot exceed $10 million.
Error Location:	`Amount`

Opportunity Check for Products

Field	Value
Description:	Validates that an opportunity has at least one opportunity product before users can save a change to an opportunity.
Formula:	`NOT(OR(ISNEW(),HasOpportunityLineItem))`
Error Message:	You must add products to this opportunity before saving.
Error Location:	`Top of Page`

Opportunity Must Have Products if Beyond "Needs Analysis" Stage

Field	Value
Description:	Validates that an opportunity has opportunity products before the `Stage` can move beyond Needs Analysis.

Formula:	`AND (`
	`  CASE( StageName,`
	`  "Value Proposition", 1,`
	`  "Id. Decision Makers", 1,`
	`  "Perception Analysis", 1,`
	`  "Proposal/Price Quote", 1,`
	`  "Negotiation/Review", 1,`
	`  "Closed Won", 1,`
	`  0) = 1,`
	`  NOT(HasOpportunityLineItem)`
	`)`

Error Message:	Opportunity products are required to advance beyond the Needs Analysis stage.
Error Location:	`Top of Page`

Opportunity Name Format

Field	Value
Description:	Validates that an opportunity contains a hyphen as a way of enforcing an "[Account] - [Amount]" opportunity naming convention.
Formula:	`FIND( " - ", Name ) = 0`
Error Message:	Opportunity Name should use "[Account] - [Amount]" format.
Error Location:	`Opportunity Name`

Prevent Sales Reps from Moving Opportunity Stage Backwards

Field	Value
Description:	Prevent sales reps from changing opportunity `Stage` "backwards" to specific values, once they have accepted the opportunity via a custom approval process. The approval process sets the custom `Accepted Flag` checkbox to True.
Formula:	`AND( Accepted_Flag__c, OR ( ISPICKVAL( StageName, "Stage 1"), ISPICKVAL( StageName, "Stage 2")) )`
Error Message:	Invalid stage for accepted opportunity.
Error Location:	`Stage`

Probability Must Be 100% for Won Opportunities

Field	Value
Description:	Validates that the probability of a won opportunity is properly set to 100%. This is useful for data cleanliness and reporting purposes.
Formula:	`AND ( ISPICKVAL( StageName, "Closed Won"), Probability <> 1)`
Error Message:	Probability must be 100% for won opportunities.
Error Location:	`Probability`

Probability Must Be Zero for Lost Opportunities

Field	Value
Description:	Validates that the probability of a lost opportunity is properly set to zero. This is useful for data cleanliness and reporting purposes.
Formula:	`AND ( ISPICKVAL( StageName, "Closed Lost"), Probability <> 0)`
Error Message:	Probability must be 0% for lost opportunities.
Error Location:	`Probability`

Project Start Date

Field	Value
Description:	Validates that a field is conditionally required based on the values of other fields. Use this validation formula to ensure that users include a `Project Start Date` for an opportunity that is closed/won.
Formula:	`AND(ISPICKVAL(StageName, "Closed Won"), ISNULL(Project_Start_Date__c))`
Error Message:	Project start date is required for won opportunities.
Error Location:	`Project Start Date`

Sample Quote Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Display Error if Quote Line Item Discount Exceeds 40%

Field	Value
Description:	Shows an error if a quote line item's discount exceeds 40%.
Formula:	`Discount > .40`
Error Message:	The discount on this quote line item cannot exceed 40%.
Error Location:	`Discount` on quote

Sample User, Role, and Profile Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Discount Percent Does Not Exceed Role-Based Limit

Field	Value
Description:	Validates that a custom field on opportunities called `Discount Percent` does not exceed a maximum value that varies depending on the user's role. The default maximum is 15%.
Formula:	`Discount_Percent__c > VLOOKUP($ObjectType.Role_Limits__c.Fields.Limit__c, $ObjectType.Role_Limits__c.Fields.Name, $UserRole.Name)`
Error Message:	Discount (%) exceeds limit allowed for your role.
Error Location:	`Discount Percent`

Expense Amount Does Not Exceed User's Max Allowed Expense

Field	Value
Description:	Validates a custom field called `Expense Amount` against a custom user field called `Max Allowed Expense`.
Formula:	`Expense_Amount__c > $User.Max_Allowed_Expense__c`
Error Message:	Amount cannot exceed your maximum allowed expense.
Error Location:	`Expense Amount`

Only Record Owner Can Change Field

Field	Value
Description:	Ensures that only the record owner can make changes to a custom field called `Personal Goal`.
Formula:	`AND( ISCHANGED( Personal_Goal__c ), Owner <> $User.Id )`
Error Message:	Only record owner can change Personal Goal.
Error Location:	`Personal Goal`

Only Record Owner or Administrator Can Change Field

Field	Value
Description:	Ensures that a user can make changes to a custom field called `Personal Goal` only if the user is the record owner or has a custom profile of "Custom: System Admin."
Formula:	`AND( ISCHANGED( Personal_Goal__c ), Owner <> $User.Id, $Profile.Name <> "Custom: System Admin" )` Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Only record owner or administrator can change Personal Goal.
Error Location:	`Personal Goal`

Opportunity Close Date Can Only Be Back-Dated by Administrator

Field	Value
Description:	Validates that the `Close Date` of an opportunity does not fall prior to the current month, except for users who have a custom profile called "Custom: System Admin."
Formula:	`AND( OR ( ISNEW(), ISCHANGED( CloseDate )), CloseDate < DATE( YEAR(TODAY()), MONTH(TODAY()), 1), $Profile.Name <> "Custom: System Admin" )` Note: $Profile merge fields are only available in Enterprise, Unlimited, Performance, and Developer Editions.
Error Message:	Close Date cannot be prior to current month.
Error Location:	`Close Date`

Miscellaneous Sample Validation Rules

For more information on any of the formula functions used in these examples, see "Formula Operators and Functions" in the Salesforce Help.

Allow Number to Be Increased but Not Decreased

Field	Value
Description:	Allows a custom field called `Commit Amount` to be increased but not decreased after initial creation. This rule uses the PRIORVALUE() function to compare the updated value of the field to its value prior to update.
Formula:	`PRIORVALUE(Commit_Amount__c) > Commit_Amount__c`
Error Message:	Commit Amount cannot be decreased.
Error Location:	`Commit Amount`

California Driver's License

Field	Value
Description:	Ensures that a custom field called `Drivers License` is in the correct A9999999 format when the `Mailing State` is "CA".

Formula:	`AND(`
	`MailingState = "CA",`
	`NOT(REGEX(Drivers_License__c, "([A-Z]\\d{7})?"))`
	`)`

Error Message:	Invalid California driver's license format.
Error Location:	`Drivers License`

Force Users to Check "I Accept Terms" to Enter Certain Values

Field	Value
Description:	Uses a checkbox labeled "I accept terms" to force the user to select a checkbox in order to enter a value called `Number of Days` that exceeds their Paid Time Off (PTO) balance available.

Formula:	`AND(`
	`NOT( I_accept_terms__c ),`
	`Number_of_Days__c > $User.PTO_Balance__c`
	`)`

Error Message:	Request will cause a negative PTO balance. You must accept Negative PTO Balance terms.
Error Location:	`I accept terms`

Prohibit Changes to a Field After It Has Been Saved

Field	Value
Description:	Prevents users from changing a custom field called `Guaranteed Rate` after it has been saved initially.
Formula:	`AND( NOT( ISNEW() ), ISCHANGED( Guaranteed_Rate__c ) )`
Error Message:	Guaranteed Rate cannot be changed.
Error Location:	`Guaranteed Rate`

Social Security Number Format

Field	Value
Description:	Validates that a custom text field called `SSN` is formatted in 999-99-9999 number format (if it is not blank). The pattern specifies: • Three single digits (0-9):\\d{3} • A dash • Two single digits (0-9):\\d{2} • A dash • Four single digits (0-9):\\d{4}
Formula:	`NOT( OR( ISBLANK(Social_Security_Number__c), REGEX( Social_Security_Number__c , "[0-9]{3}-[0-9]{2}-[0-9]{4}") ) )`
Error Message:	SSN must be in this format: 999-99-9999.
Error Location:	`SSN`

Valid Currency

Field	Value
Description:	Validates selected currency against an explicit subset of active currencies in your organization using the `Currency` picklist. Use this example if you only allow some of the active currencies in your organization to be applied to certain types of records.
Formula:	`CASE(CurrencyIsoCode, "USD", 1, "EUR", 1, "GBP", 1, "JPY", 1, 0) = 0`
Error Message:	Currency must be USD, EUR, GBP, or JPY.
Error Location:	`Currency`

Valid Credit Card Number

Field	Value
Description:	Validates that a custom text field called `Credit_Card_Number` is formatted in 9999-9999-9999-9999 or 9999999999999999 number format when it is not blank. The pattern specifies: • Four digits (0-9) followed by a dash: \\d{4}- • The aforementioned pattern is repeated three times by wrapping it in () {3} • Four digits (0-9) • The OR character (|) allows an alternative pattern of 16 digits of zero through nine with no dashes: \\d{16}
Formula:	`NOT( REGEX( Credit_Card_Number__c , "(((\\d{4}-){3}\\d{4})|\\d{16})?"))`
Error Message:	Credit Card Number must be in this format: 9999-9999-9999-9999 or 9999999999999999.
Error Location:	`Credit Card Number`

Valid IP Address

Field	Value
Description:	Ensures that a custom field called `IP Address` is in the correct format, four 3-digit numbers (0-255) separated by periods.
Formula:	`NOT( REGEX( IP_Address__c, "^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$" ))`
Error Message:	Error: IP Address must be in form 999.999.999.999 where each part is between 0 and 255.
Error Location:	`IP Address`

Website Extension

Field	Value
Description:	Validates a custom field called `Web Site` to ensure its last four characters are in an explicit set of valid website extensions.
Formula:	`AND( RIGHT( Web_Site__c, 4) <> ".COM", RIGHT( Web_Site__c, 4) <> ".com", RIGHT( Web_Site__c, 4) <> ".ORG", RIGHT( Web_Site__c, 4) <> ".org", RIGHT( Web_Site__c, 4) <> ".NET", RIGHT( Web_Site__c, 4) <> ".net", RIGHT( Web_Site__c, 6) <> ".CO.UK", RIGHT( Web_Site__c, 6) <> ".co.uk" )`
Error Message:	Web Site must have an extension of `.com`, `.org`, `.net`, or `.co.uk`.
Error Location:	`Web Site`