Aggregate Functions in Salesforce Visualforce page

By SalesforceCodes | July 6, 2020 0 Comment

Use aggregate functions in a GROUP BY clause in SOQL queries to generate reports for analysis. Aggregate functions include AVG(), COUNT(), MIN(), MAX(), SUM(), and more.

You can also use aggregate functions without using a GROUP BY clause. For example, you could use the AVG() aggregate function to find the average Amount for all your opportunities.

Example

SELECT AVG(Amount)FROM Opportunity

GROUP BY clause is used in SOQL query to group set of records by the values specified in the field. We can perform aggregate functions using GROUP BY clause.

Aggregated functions for GROUP BY clause:

- AVG() – Returns the average value of a numeric field
- COUNT() – Returns the number of rows matching the query criteria
- MIN() – Returns the minimum value of a field
- MAX() – Returns the maximum value of a field
- SUM() – Returns the total sum of a numeric field
- COUNT (FIELD_NAME)
- COUNT_DISTINCT ()

AVG( ) :

Returns the average value of a numeric field.

For example:

SELECT CampaignId, AVG(Amount) FROM Opportunity GROUP BY CampaignId

COUNT( ) and COUNT(fieldName) :

Returns the number of rows matching the query criteria. For example

USING COUNT( ):

SELECT COUNT( ) FROM Account WHERE Name LIKE 'a%'

USING COUNT(fieldName):

SELECT COUNT(Id) FROM Account WHERE Name LIKE 'a%'

NOTE : If you are using a GROUP BY clause, use COUNT(fieldName) instead of COUNT().

COUNT_DISTINCT():

Returns the number of distinct non-null field values matching the query criteria. For example:

SELECT COUNT_DISTINCT(Company) FROM Lead

MIN( )

Returns the minimum value of a field. For example:

SELECT MIN(CreatedDate), FirstName, LastName FROM Contact GROUP BY FirstName, LastName

If you use the MIN() or MAX() functions on a picklist field, the function uses the sort order of the picklist values instead of alphabetical order.

MAX( )

Returns the maximum value of a field. For example:

SELECT Name, MAX(BudgetedCost) FROM Campaign GROUP BY Name

SUM( )

Returns the total sum of a numeric field. For example:

SELECT SUM(Amount) FROM Opportunity WHERE IsClosed = false AND Probability > 60

IMPORTANT NOTE:

You can't use a LIMIT clause in a query that uses an aggregate function, but does not use a GROUP BY clause. For example, the following query is invalid:

SELECT MAX(CreatedDate) FROM Account LIMIT 1

The aggregate functions COUNT(fieldname), COUNT_DISTINCT(), SUM(), AVG(), MIN() and MAX() in SOQL return an AggregateResult object or a List of AggregateResult objects. You can use aggregate functions result in apex by using AggregateResult object.

Example:

List<AggregateResult> result = [SELECT COUNT(Id) Total, AccountId FROM Contact WHERE AccountId != null GROUP BY AccountId];

Example Program

APEX CLASS

```
public with sharing class AggregateResultExample1 {
    public List<AggregateResult> result {get;set;}
    public List<AccWrapper> accWrapList {get;set;}
    public List<Account> accList;
    public Map<Id, Account> accMap;
    List<Id> idList;

    public void getData() {
        accWrapList = new List<AccWrapper>();
        result = new List<AggregateResult>();
        idList = new List<Id>();
        accList = new List<Account>();
```

```
    accMap = new Map<Id, Account>();

    result = [SELECT COUNT(Id) Total, AccountId FROM Contact WHERE AccountId != null GROUP BY AccountId];

    for(AggregateResult a : Result) {
        idList.add((Id)a.get('AccountId'));
    }

    accList = [SELECT Id, Name FROM Account WHERE Id IN : idList];
    for(Account a : accList) {
        accMap.put(a.Id, a);
    }

    for(AggregateResult aResult : result) {
        Account acc = accMap.get((Id)(aResult.get('AccountId')));
        accWrapList.add(new AccWrapper(aResult, acc.Name));
    }
}

public class AccWrapper {
    public Integer TotalContact {get;set;}
    public String AccountName {get;set;}

        public AccWrapper(AggregateResult a, String AccountName) {
            this.TotalContact =  (Integer)a.get('Total');
            this.AccountName = AccountName;
        }
    }
}
```

VF PAGE

```
<apex:page controller="AggregateResultExample1" action="{!getData}">
    <apex:form >
        <apex:pageBlock >
            <apex:pageblockTable value="{!accWrapList}" var="acc">
                <apex:column headerValue="Account Name" value="{!acc.AccountName}"/>
                <apex:column headerValue="Number of Contacts" value="{!acc.TotalContact}"/>
            </apex:pageblockTable>
        </apex:pageBlock>
    </apex:form>
</apex:page>
```

OUTPUT

Account Name	Number of Contacts
Salesforce	1
Burlington Textiles Corp of America	1
University of Arizona	1
ramakrishna	1
Dickenson plc	1
GenePoint	1
Express Logistics and Transport	2
United Oil & Gas, Singapore	2

Question Related to Aggregate Queries:

▼ How many maximum number of aliased fields you can have in Aggregate Query?

100

▼ Can we use Database.QueryLocator for Aggregate Queries?

No

▼ Arrange these in appropriate Order: HAVING,WHERE,LIMIT,GROUP BY,ORDER BY

WHERE, GROUP BY, HAVING, ORDER BY, LIMIT

▼ What is “GROUP BY ROLLUP” and “GROUP BY CUBE” with Example?

GROUP BY ROLLUP: Add subtotals for aggregated data in query results, A query with a GROUP BY ROLLUP clause returns the same aggregated data as an equivalent query with a GROUP BY clause. It also returns multiple levels of subtotal rows. You can include up to three fields in a comma-separated list in a GROUP BY ROLLUP clause.

SELECT LeadSource, COUNT(Name) cnt FROM Lead GROUP BY ROLLUP(LeadSource)

SELECT Status, LeadSource, COUNT(Name) cnt FROM Lead GROUP BY ROLLUP(Status, LeadSource)

GROUP BY CUBE: To add subtotals for every possible combination of grouped field in the query results. This is particular useful if you need to compile cross-tabular reports of your data.

SELECT Type, BillingCountry,GROUPING(Type) grpType, GROUPING(BillingCountry) grpCty, COUNT(id) accts FROM Account

GROUP BY CUBE(Type, BillingCountry) ORDER BY GROUPING(Type), GROUPING(BillingCountry)

▼ Write a Aggregate Query to get total opportunity amount ‘closed lost’ ?

select sum(amount) from opportunity where stagename=’Closed Lost’

▼ Write a Query to get total amount from opportunities close date last three years?

select sum(amount) from Opportunity where closedate > Last_N_Years : 3

▼ Write a query to get total amount from opportunities with amount greater than 100000 per Account Name in 2015 ?

select sum(amount),account.name from opportunity where amount>100000 and calendar_year(account.createddate)=2015 group by account.name

▼ Write a Query to calculate total opportunity amount by stage expected to close this quarter ?

Select Sum(Amount) from Opportunity where stagename='Closed Won' and CreatedDate=this_quarter group by stageName

▼ Write a query to get total number of accounts which have opportunities

Select Count(id) from Account where id in (Select AccountId from Opportunity)

▼ What is the return type of aggregate function?

AggregateResult , AggregateResult is a read-only sObject and is only used for query results.

▼ Can we use Sub Queries in Aggregate Queries? If so, how?

No, you are not allowed to use aggregate functions in subqueries.

▼ How we can use Serialization of aggregate query result fields?

String JSONString = JSON.serialize(Database.Query('SELECT AVG(Amount)aver FROM Opportunity'));

▼ Write a soql to get the recent account records ?

List<Account>accs=[Select id,name from account Order By createddate DESC limit 50];

▼ Write a query for obtaining records which were updated this week?

SELECT Name FROM Account WHERE LastModifiedDate = THIS_WEEK

▼ Write a query for sum up the sales by fiscal year? Use Group By in query

SELECT FISCAL_YEAR(CloseDate) year,SUM(Amount) summary FROM Opportunity WHERE IsWon = true GROUP BY FISCAL_YEAR(CloseDate)

▼ Write a query for obtaining records in a particular fiscal year?

SELECT Name FROM Account WHERE THIS_FISCAL_YEAR(CreatedDate) = 2020

Category: Admin Visualforce Pages Tags: Aggregate function , Aggregate result