


```
String name='Shawn123';
Boolean isRecordPresent = False;
Date todayDate = System.today();
Date newDate = Date.newInstance( 2020, 10, 30 );

DateTime DT =
Datetime.newInstance(Date.newInstance(2020, 05,
25), Time.newInstance(8, 0, 0, 0));
DateTime CurrentDateTIme = System.now();

Account Acc = new Account();
Acc.Name = 'test 1';
Insert Acc;
ID accountID = Acc.Id;

Account Acc = [Select Id,Name from Account LIMIT
1];
```

B. Collection Variables

These variables store a group of similar data types. It is efficient to store data into a collection of variables instead of creating a variable for each record. The examples of Collection variables are:

1.Set: A Set is a collection type that contains multiple unordered unique records. Duplicate records storage is not allowed in Sets. Here are a few examples of Sets

```
Set<string>StudentNames =
newSet<string>{'Maria', 'Krishna'};
StudentNames.add('John');
StudentNames.remove('Maria');
System.debug(StudentNames.contains('Krishna'));
```

2.List: List is an ordered collection type. Its contents can be accessed by index.

```
List<string>Countries = new List<string>( );
Countries.add('India');
Countries.add('USA');
Countries.get(1);
Countries.set(1,'Ukraine');
Countries.clear();
System.debug('The list size is '+Countries.size( ));
```

3.Map: It is a collection type that has a key, and value pair like the dictionary in python. The keys should be unique, and the values can be accessed by using the appropriate keys. Here are a few basic methods of Map.

```
Map<Integer,String> mapRollToStudentName = new
Map<Integer,String>( );
mapRollToStudentName.put(11, 'Sakshi');
mapRollToStudentName.get(11);
System.debug(mapRollToStudentName.containsKey(14));
mapRollToStudentName.keySet( ); //returns keys
mapRollToStudentName.values( ); //returns values
```

Sample Apex Code and its parts

```
Integer count=0; // Initialize primitive data type
List<Account> updateStudentList=new List<Account>( );
List<Account> studentList = [Select ID, Name,
Class__c, Phone,Remarks__c from Account LIMIT 10];
//SOQL query

for(Account acc: studentList){ //Looping statement
    if(acc.Class__c == 10){ //Flow condition
        acc.Remarks__c = 'Class 10 student';
```

```
        updateStudentList.add(acc);
    }
    System.debug('Record count is '+count);
    count = count + 1;
}
if( !updateStudentList.isEmpty( ) ){
    try{
        update updateStudentList;
    }
    catch(DMLException e){ //error handling
        System.debug('Error '+e.getMessage( ) );
    }
}
```

C. Apex Triggers

Apex provides several events while DML statements are being executed. We can utilize these events to perform any complex business logic implementation, validation check, email services, etc. These events are leveraged by writing triggers. Apex Trigger is a piece of code that will be executed whenever there is a database operation performed like insertion, updating, deletion, etc. Apex trigger has primarily two operational modes 'before' and 'after'. Triggers are object-specific pieces of code. If a trigger is written on Object Account. It will be executed when there is a database operation happening for the object's records. Trigger.New is a standard in-built list that contains the list of all the records that are undergoing database operation. Trigger.NewMap and Trigger.OldMap can be used to check if a value has been changed in the record. Below mentioned is the list of events available in Salesforce.

- 1) before insert
- 2) before update
- 3) before delete
- 4) after insert
- 5) after update
- 6) after delete
- 7) after undelete

Example of Apex Trigger:

```
trigger AccountRemarksTrigger on Account(before
insert, before update) {
for(Account acc : Trigger.New) {
    if(acc.Class__c == 10 && acc.StudentNumber__c>100){
        account.Remarks__c = 'Registration Required';
    }
}
}
```

The above code runs before the record is inserted or updated. If the current record in the loop passes conditional IF statement, then the Remarks field is set to 'Registration Required'.

XII. GOVERNOR LIMITS AND THE CONCEPT OF BULKIFICATION

Salesforce imposes a restriction on resource usage while executing the Apex code. This restriction enables shared resource management thereby avoiding monopolization of resources in Salesforce. These limits apply to every individual transaction taking place in Apex. Hence the optimization of the Apex code becomes crucial so that the apex code transaction doesn't hit the Governor limits. This gives rise to the concept of Bulkification of code. Before understanding the methods to bulkify the code, let's check the Governor limits for Salesforce.

SOQL queries Limit: 100
 Total records retrieved using SOQL: 50000
 SOSL queries Limit: 20
 Total records retrieved using a single SOQL: 2000
 Total records retrieved using Database.QueryLocator: 10000
 Total DML operation Limit: 150
 Maximum execution time for Apex transaction: 10 minutes

If the apex transaction hits the Governor Limits, then the System will throw an error with System.LimitException

Bulkification: The apex code needs to be optimized so that it can use resources efficiently even if the record volume increases. To avoid hitting the governor limit we need to remove SOQL queries, DML statements from inside of the Looping statements. Using Collection variables and performing a single DML operation on the collection is very efficient in resource management. We should prefer the usage of batch class if record count is high or when the apex needs to be scheduled on a frequent basis.

A. Code without Bulkification

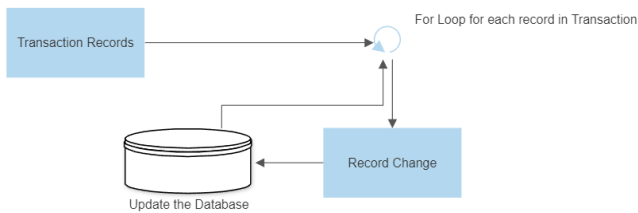


Fig 3. Approach of Non Optimized Code

```
List<Account> DealerAcc = [Select id, Income__c, Remarks__c, Type from Account where Type='Dealer'];
for(Account acc: DealerAcc){
    if(Income__c >50000){
        acc.Remarks__c = 'It is a dealer account';
        update acc; // DML statement
    }
}
```

If there are more than 150 records in DealerAcc record List that pass the if control statement, then the apex code will throw an error stating ‘DML statements limit of 150 reached’. Since the DML statement is mentioned inside the loop statement, it is executed repetitively until the Governor limits are hit and the rest of the records are left unprocessed. To avoid this situation let’s check below code snippet below where Bulkification is taken into account.

B. Code with Bulkification

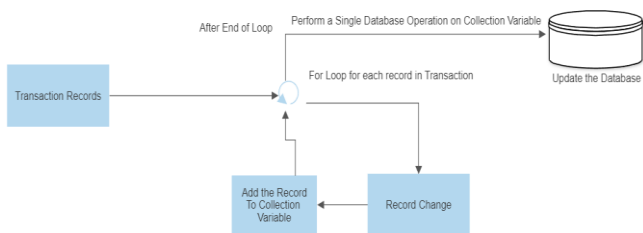


Fig 4. Approach of the Optimized Code

```
List<Account> DealerAcc = [Select id,Income__c, Remarks__c,Type from Account where Type='Dealer'];
List<Account> updateAccountList = new List<Account>();
for(Account acc: DealerAcc){
    if(Income__c >50000){
        acc.Remarks__c = 'It is a dealer account';
        updateAccountList.add(acc);
    }
}
if(!updateAccountList.isEmpty()){
    try{
        update updateAccountList; // DML
    }
    catch(DMLException e){
        System.debug('Error occurred '+e.getMessage() );
    }
}
```

Here the DML operation happens only once and it is performed on collection Type updateAccountList. So, this method of writing code prevents hitting limits and boosts the performance of the code.

This summarizes data modelling, storage procedures, Relational Database in Salesforce, The Database language, SOQL query, Data types, Import and Export. Let’s look at the Data process automation in Salesforce.

XIII. DATA AUTOMATION PROCESS IN SALESFORCE

In the current era where business logic requirements are getting more complex than ever, the requirement for a robust automation process also increases to fulfill the challenging demand. In the case of CRM software, countless operations are taking place like record creation, updation, web services, email alerts, report generation, etc. Manual handling of such a process causes a significant delay in processing. Hence the automation of these tasks can result in reduced manpower. Data automation saves time and money, increasing business efficiency. Few processes automation tools in Salesforce are Workflow Rules, Salesforce Flows and Apex.

1) Workflow rules: It is a tool that helps in the automation of recursive tasks like field updation, email notification, etc. It is object-specific, meaning while configuring workflow rules, we need to specify which object in Salesforce is this applicable to. Workflow rules have mainly two parts ‘criteria’ and ‘action’. Criteria are the condition that needs to be satisfied while an operation is taking place in Salesforce. Action is the subsequent process that needs to be followed once the criteria are fulfilled.

The different types of criteria in Salesforce are:

a. Evaluation criteria is the initial entry condition to the workflow rule. Evaluate the rule when a record is:

- Created
- Created, and every time it's edited
- Created, and any time it's edited to subsequently meet criteria

b. *Rule criteria* can be configured by combining multiple conditions depending upon the field values in the record or the current user settings.

For ex: An object storing the Salaries of employees. Whenever a record is created where the Salary is above 100000 and the employee division is Sales, the rule criteria must evaluate to true and perform required operations.

The Actions in Workflow rules are divided into two types:

a. *Immediate action*: These actions are executed immediately after the rule criteria are satisfied. There are primarily four types of immediate actions available in Salesforce.

- **Email Alert**: Trigger an email to be sent when the criteria evaluate to be true. This can be achieved by creating email templates and alerts within salesforce. This helps in the dynamic email generation which contains the field values specific to the record.
- **Field Updates**: If there is a requirement to update the fields in the record, this can be easily achieved using Field updates action.
- **Tasks**: A task can be created using workflow rules. The fields of task like Subject, Description, AssignedTo, DueDate, Status, Priority, Comments can be configured using workflow rules.
- **Outbound Message**: If a secure, configurable message needs to be sent to external system, an outbound message can be sent by using the endpoint URL of the external system and specifying the record fields to be shared with it. There is also an option to pass the Session ID in this outbound message.

b. *Time based action*: These actions can be scheduled at a later point in time. The scheduling can be managed using the date-related fields in the record and specifying the scheduling interval like 30 days after record creation etc.

2. *Flows*: Flows are point-and-click automation tools with a much wider range of operability compared to workflow rules. With its wide range of use cases and the enhancements in every release, Flows are one of the powerful tools in Salesforce. The popularity of flows is because of their power to perform the complex business operation and their ease to build them. Flows can be configured by clicks and no code is required. Salesforce flows have three components namely

- Elements
- Connectors
- Resources

Let us understand each of the components

a. *Elements*: The individual components in the flow are known as elements. The elements in Salesforce are of three types. Interaction, Logic, and Data Elements

- **Interaction**: It includes Action, Sub flow, and screen. Action element helps to call custom apex action, sending an email, creating a task, etc. The screen is available only on-screen flows. A screen helps to

collect data from the user or display data to the user. An example of a screen is an onboarding journey which helps the user to navigate hassle-free. A sub-flow provides the option to call another flow inside the existing flow. This helps in better structuring/designing the flow without creating messy dangling elements in a single flow.

- **Logic**: Logic elements consists of decisions, loops and assignments, Collection sort and Collection filter, and pause. Decisions are used to branch the flow into segments and apply an entry criterion to enter each branch. Loops are used to handle collection variables that contain multiple records/values. Assignment helps to assign value to the variables/ records. Collection sort is used for sorting a record collection based on a field in the record. A collection filter is used to apply filters to the record collection and extract a fraction of the records satisfying the condition. Pause is used to temporarily halt the flow and resume when the org receives a platform event message.
- **Data Elements**: These elements are used while transacting with the database. There are five available features namely Get records, Update records, Create records and Delete Records, and Rollback records [Cancel pending record changes]. Records can be selected by specifying the filters similar to a SOQL query but configurable in a point-and-click window.
- **Connectors**: As the name itself suggests, these are the links that connect the elements. The connectors are directional and influence the direction of the flow.
- **Resources**: These are the variables that can be stored and used during the execution of flow. The types of resources available in salesforce flows are Variable, Choice, Constant, Collection, Formula, etc. Variables are used to store data. It can be a number, string, record, Boolean, date, etc.

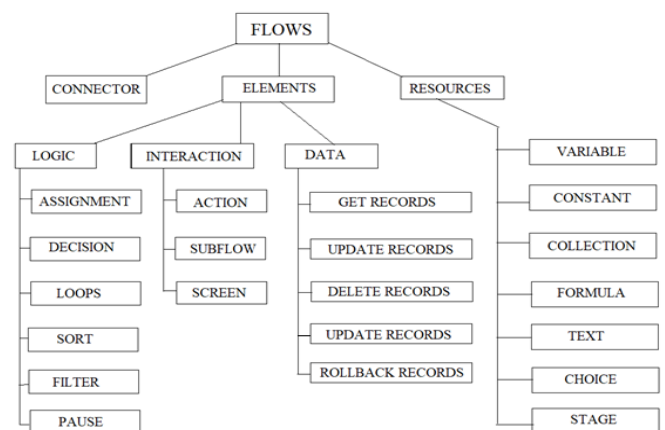


Fig 5. Flow Components

Above picture depicts the hierarchical representation of the features available inside the Salesforce Flow.

There are multiple types of Flows available in Salesforce.

- *Screen flow*: Guides users through a business process that's launched from Lightning pages, Experience Cloud sites, quick actions, and more.
- *Record Triggered Flow*: Launches when a record is created, updated, or deleted. This auto launched flow runs in the background.
- *Schedule Triggered Flow*: Launches at a specified time and frequency for each record in a batch. This auto launched flow runs in the background.
- *Platform Event Triggered Flow*: Launches when a platform event message is received. This auto launched flow runs in the background.
- *Auto launched Flow*: Launches when invoked by Apex, processes, REST API, and more. This auto launched flow runs in the background.

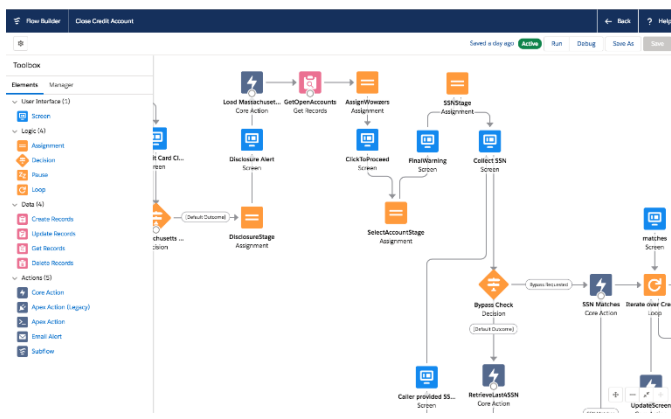


Fig 6. A Typical Salesforce Flow Representation

3) *Apex*: This is the most powerful tool available inside Salesforce. It provides a rich set of configurable options that can be leveraged to perform complex automation. Apex can configure almost everything in Salesforce. Database transactions, email services, web services, invocable methods, and many more features can be achieved using Apex. Triggers discussed earlier, are used to configure complex automation tasks while database transaction is undergoing. The apex documentation is very well designed to provide information about the in-built functions.

TABLE II. AUTOMATION TOOLS COMPARISON

| Automation Type | Complexity | Configurability Power | Requires Code |
|-----------------|------------|-----------------------|---------------|
| Workflow Rules | Least | Medium | No |
| Flows | Medium | High | No |
| Apex | Most | Highest | Yes |

The table above compares the automation tools based on various parameters.

XIV. ADVANTAGES OF USING SALESFORCE CRM

- 1) Salesforce has a high range of configuration power providing flexibility while designing.
- 2) It provides its design system called SLDS which is ready to use. This helps in building buttons, layouts, progress bar, options menu, etc. with much ease.

3) Salesforce is incorporating more features with every release. With the current range of rich available features and its improvements in the future, it is the right tool to use.

4) Companies have started to use Salesforce as a productivity tool also because of its ability to configure tasks, schedule meetings, using chatter[salesforce messaging platform] for communication.

5) Similar to App Store/Play Store, Salesforce has its store known as 'App Exchange' where both paid and free applications can be installed into the salesforce system.

6) Cloud-based platform doesn't require any software installation on the User's end.

7) Security of Salesforce is strong. It has OAuth and configurable Security aspects like Session Expiry timings, whitelisting IP address range, Login Hours, Password Policies, etc.

8) Integration with Visualization tools like Tableau has been made very easy. Tableau has an option to import data from Salesforce. Also, there are in-built visualization tools called Dashboards.

9) Python libraries like simple salesforce help in connecting python with Salesforce Org. This facilitates creating, querying records, and converting them to Pandas data frame. Thereby data cleanup and analysis outside Salesforce becomes very easy.

10) AI plays a fundamental role because AI solutions applied to CRM enable companies to better assimilate and analyze customer data making them increasingly able to anticipate, plan and take advantage of upcoming opportunities[7]. Analytics studio powered by Einstein analytics can be used to gain insights into the existing data, build a model and predict the outcomes. For example, Churn modeling of a bank customer leaving the bank can be predicted using Analytics

XV. CONCLUSION

CRM technology is booming in the current techno-space and Salesforce is at the forefront with a huge gap with its competitors. Businesses are ready to use Salesforce as the wide variety of features available in Salesforce cuts the cost of using external systems/ 3rd party software. Salesforce offers easy-to-use CRM software. Salesforce has a free, fully equipped learning platform called Trailhead which has a clean structure displaying the individual modules and trail mix that can lead one to the correct path. The Salesforce developer's community is also active to provide help via stack exchange platforms and suggesting new features that will help the smooth operation in the future. This paper showed the Data Modelling, Data Storage Procedures, Database language, and Data Automation. After reading this paper, we can draw analogies between the Salesforce data model and traditional RDBMS, Python, and SQL. With a wide community, cutting-edge technology, research, and development, easy-to-use software, trailheads, and a strong business model, Salesforce is the mammoth of the CRM industry.

REFERENCES

- [1] Omkar Sunar Verma, Ishwar Chaudhary, Mahammad Javed Khan, Akhilesh Kumar Chaudhary, Isha "Comparative Study Of Relational Database Management System And Object-Oriented Database Management System"; 2021 IJCRT | Volume 9, Issue 4 April 2021 | ISSN: 2320-2882;
- [2] Rakesh Kumar, Yougeshwary Sharma, Sonu Agarwal, Pragya, Bhanu Bhushan Parasha "Extremely effective CRM Solution Using Salesforce"; JETIR (ISSN-2349-5162); Volume 1 Issue 5
- [3] Yasin N. Silva, Isadora Almeida, Michell Queiroz "SQL: From Traditional Databases to Big Data"
- [4] Saeed Awadh Bin-Nashwan, Haslinda Hassan "Impact of customer relationship management (CRM) on customer satisfaction and loyalty": Journal of Advanced Research in Business and Management Studies 6, Issue 1 (2017) 86-107
- [5] Priyanka Meena "Customer Relationship Management Research from 2000 to 2020: An Academic Literature Review and Classification".
- [6] Francis Buttle, Lawrence Ang and Reiny Iriana "Sales force automation: review, critique, research agenda"; International Journal of Management Reviews (2006)
- [7] Cristina Ledro, Anna Nosella and Andrea Vinelli "Artificial intelligence in customer relationship management" Journal of Business & Industrial Marketing
- [8] Vidyansh Chandra "A Study On The CRM Strategies Of D-Mart" IJCRT
- [9] Vicente Guerola-Navarro, Raul Oltra-Badenes, Hermenegildo Gil-Gomez & Jose Antonio Gil-Gomez "Research model for measuring the impact of customer relationship management (CRM) on performance indicators", Taylor and Francis Economic Research Papers.
- [10] Anuradha Manchar; Ankit Chouhan "Salesforce CRM: A new way of managing customer relationship in cloud environment" IEEE 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)
- [11] Salesforce Websites
<https://www.salesforce.com/>
<https://trailhead.salesforce.com/>
<https://developer.salesforce.com/>
<https://www.lightningdesignsystem.com/>
- [12] Arockia Panimalar.S, Priyadarshan.R, Mithun Kumar.R, Visweshwaran.G "Salesforce.com – A Cloud Provider" International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 09 | Sep -2017