Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

Data Modelling, Management and Automation in Salesforce

Shridhar Mashalkar
ECE, BMS College Of Engineering
Autonomous College Affiliated to VTU Belgaum
Bengaluru, India

Vineeth R
ECE, BMS College Of Engineering
Autonomous College Affiliated to VTU Belgaum
Bengaluru, India

Abstract*— A technical solution called Customer Relationship Management (CRM) was developed in the 1970s to help businesses automate the control of their internal sales staff. The configuration, installation, deployment, and security of traditional commercial systems require a large team of experts and are expensive and complex in nature. By adopting a cloud-based CRM, you can do away with all of these issues since you won't need to manage any software or hardware, since that will be handled by a seasoned vendor[1]. Customer happiness is crucial because pleased customers function as free promotion for the business [5]. It is asserted that keeping current clients is simpler than acquiring new ones. As a result, businesses are developing plans to keep customers and retraining staff to focus more on providing excellent customer service[4]. With each passing day, research on CRM is growing quickly in practically every industry. CRM is well-known in the service sector, but lately, research publications published in sectors other than the service sector are growing. More advantages and disadvantages of the CRM system are being investigated by researchers and practitioners[5]. Vendors and consultants assert that implementing Sales Force Automation will have several advantages, including quicker cash flow, shorter sales cycles that will result in quicker inventory turnover, improved customer relations, increased salesperson productivity, accurate reporting, growing market share, higher win rates, lower cost-of-sales, more closing opportunities, and improved profitability. Softer results like fewer rework, quicker information, and better management reports can be used to balance out these hard outcomes. [6]. It is a leading CRM [Customer Relationship Management] platform that enables the automation of several processes, including scheduling emails and updating records, producing reports, visualizing data, and other tasks. SAP, Oracle, Microsoft, and other CRMs are available. A few well-known businesses that have made use of CRM features are Spotify, BYJU's, D-Mart[8], Toyota, etc. Both in terms of interest as a topic of scientific research and in terms of adoption in businesses across all industries, CRM has shown an exponential development since 2010[9]. A model for enabling ubiquitous, helpful, on-demand access to a shared pool of reconfigurable computing resources, such as networks, servers, storage, applications, and services, is known as cloud computing[12]. These resources can be quickly provisioned and released with little administrative effort or service provider interaction.

Keywords*— *Salesforce; RDBMS; CRM; Data Management

I. INTRODUCTION

Data is the Oil of the 21st century. It holds important information if extracted correctly. Storage of such data for effortless retrieval is challenging when the record volume starts increasing. But this is not an issue inside salesforce which has a state-of-the-art data storage system in its databases. Salesforce is a platform that has a plethora of features embedded in it that helps in accelerating sales, customer service, and marketing. It has specialized clouds for each requirement like Sales Cloud, Community Cloud, Service Cloud, Marketing Cloud, Health Cloud, etc. Salesforce Acquisition of top-rated software like Heroku [PaaS], Tableau [data visualization], Slack [communication tool], and Quip [collaboration] shows its expansion due to the steady growth over a couple of years. It is a phenomenal tool for managing interactions like tasks, meetings, calendar scheduling, calls, emails, etc. It helps in the incorporation of different frameworks and manufacture our own applications[10]. With its integration to external systems like WhatsApp, legal documents verifications, SMS, etc. Salesforce is expanding its giant footprints in the tech industry, not to forget its **$26.49 Billion revenue in the single year of 2022**. In this paper, we are discussing data management, its automation in Salesforce systems. This paper aims to show the internal salesforce data storage procedures, Relational Database in Salesforce, The Database language, Data automation features and its configuration in Salesforce which can handle millions of records in its database. Software applications called data management programmes offer modelling services, manage enormous amounts of pertinent data that are shared by many users over a lengthy period, and manage that data[2].Let's explore the Salesforce Data Modeling.

II. DATA MODELLING

Let's discuss Data Modelling in Salesforce by parallelly drawing comparisons with RDBMS concepts. Data in salesforce is stored by utilizing/creating objects, and creating fields to be stored in the form of records. The object is an analogy to the table to store similar data, Fields is an analogy to the columns inside the table with a particular data type assigned to it. We will talk about the data types in Salesforce in a bit. Records are like the rows in an RDBMS table. To distinguish the records, there exists a unique case-sensitive 15/18 character long ID assigned to each record in Salesforce.

For
ex: https://myorg123.lightning.force.com/lightning/r/Lead/**a0F72000001GyAvEAK**/view

This acts as a primary key helping in querying unique data from the Salesforce database. Like ER[entity relationship] diagram, Salesforce has its tool called Schema builder which shows the relationship between Objects[similar to the table in RDBMS].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

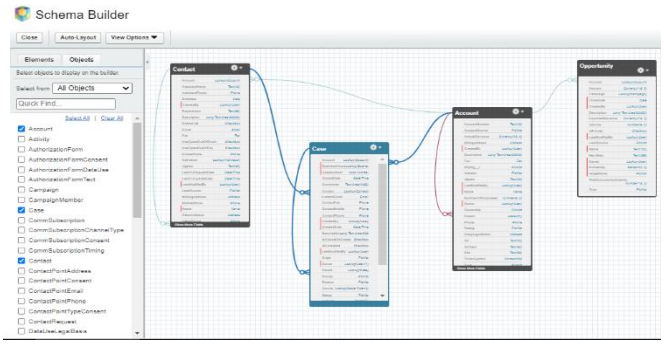

Fig 1. Schema Builder in Salesforce

Salesforce has a few in-build standard objects like Account, Lead, Opportunity, Contact, etc. which have the standard functionality for Sales process management. It also provides users, the option to create Custom objects and custom fields. Every object, field, report, code class, and UI component has two names i.e Label and API name. A label is the one which is displayed in the UI to the user and the API name is used for Integration with Database. One noticeable feature of the dual name nomenclature is that once the system is configured if there is a need to change the name in the future, no additional changes in the code systems are required because the API name remains the same. Only the Label which is displayed in the UI changes. Hence this is one appreciable feature in salesforce. Custom objects and Custom fields can be identified easily by the '__c' appending in its API name. There is an option to configure multiple kinds of records within an object, these are called record types. Record Types are used to differentiate the variations in records. To explain it better, for instance, consider the object Account which stores the information of a customer. Now we can create separate account types like Banking Account, Dealer Account, Builder Account, Seller Account, etc. and display only those fields in UI required for that record type using Layout functionality and its assignments.

III. DATATYPES OFFERED IN SALESFORCE

As discussed in the previous section, the fields are like the columns in an RDBMS table. Similar to SQL column configurations, Salesforce also provides few standard datatypes while initializing the fields in the objects. The basic field data types are Auto-Number (system-generated sequential number), Formula, Roll-Up Summary(aggregative functions), Lookup Relationship(foreign key), Master-Detail Relationship(foreign key), External Lookup(foreign key), Checkbox(Boolean), Currency, Date, Date/Time, Email, Geolocation, Number, Percent, Phone, Picklist, Picklist(Multi-Select), Text, Text Area, Text Area (Long),Text Area Rich(can add text, image, hyperlink), Text (Encrypted), Time, URL

IV. RELATIONSHIPS BETWEEN OBJECTS IN SALESFORCE

The relationship field in an object relates one object to another, a similar analogy to a foreign key used to link two tables. Let's look at a couple of examples in Salesforce.

A. Lookup Relationship

It is a loosely bound relationship between objects meaning if object A is in a lookup with B object, if object A record is deleted, the record in object B still exists. Meaning there is no cascading record deletion.
The Lookup field on the object acts like the foreign key in RDBMS. It is not mandatory to have a parent record while creating a child in this relationship.

B. Master-Detail Relationship

In this case, the relationship is tightly bound. If two objects are linked with Master[Parent]-Detail[Child], if object A is Parent and object B is the Child. If object A record is deleted then the record in Object B linked to object A will also be deleted. Meaning the record deletion exists as part of cascading effect. It is mandatory to have a parent record while creating a child record in this relationship.

V. THE CONCEPT OF BIG OBJECTS

The easiest way to describe a Big Object is to consider it as a Huge Bucket used to dump a large volume of data. The limitation of standard objects in Salesforce is that it cannot store records when the record count increases by more than a few million. So, to tackle this limitation, a special object feature has been launched in salesforce known as the big object which can store records in the range of billions[that's right in the range of billion records]. The big objects and field configurations can be created using the metadata XML file. The difference between a normal object and a big object in salesforce is the volume of data it can store. The standard functionalities like Data Validation, Flows, and processes cannot be applied to Big Objects since the main objective of big objects is to store massive data. If we add triggers and validations to such big objects, it will reduce the efficiency. While Custom Objects are appended with '__c' tag in their API name, Custom Big Objects are appended using '__b' tag in their API name.
For Example:
Custom Object- Label Teacher , API Teacher__c
Big Object – Label Student, API Student__b

VI. CUSTOM METADATA TYPE IN SALESFORCE

Custom Metadata is used to configure the application in salesforce. Similar to custom objects, it has fields to store values. The main difference is that the records in custom objects are data, whereas in custom metadata the records itself is metadata. In a real-time scenario, custom field mapping between objects, API endpoints for integration, Business logic like sales incentives calculation, and any configuration inputs is stored in custom metadata. The records can be easily deployed to other orgs without worrying about manual transferring. The API name is appended with '__mdt' [short form for metadata]. In simple terms, the fields in custom objects are configuration, but in custom metadata the records themselves are configuration. Some organizations prefer to store Apex database queries [similar to SQL queries] in custom metadata so that if reconfiguration is required, it can be easily done from UI itself.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

VII. VALIDATION RULES IN SALESFORCE

Validation rules are used to enforce certain standards before saving data into Salesforce Database. It contains a mathematical equation/formula which can also contain field references. For instance, if the Stage of lead is to be set as 'Rejected' and there is a text input field called 'Rejection Reason'. The Salesforce system will throw an error when the Stage is set to 'Rejected' and the rejection reason field is blank. Such Custom Logic standards can be applied using validation rules. Salesforce provides a rich set of functions for validations like Logical Operations, Regex, Arithmetic Operations, Null Check, Prior value, etc. Few standard validations are already present in Salesforce like 'if a user enters alphabet or special character in the phone number field, it will throw an error stating The field must only contain numbers. Validations regarding the wrong data type in a field are handled by Salesforce. But the custom validation in a field is handled using Validation rules. Another way to enforce input standards is using data Triggers. Here are a few basic validations formulas and their explanation to provide a better understanding of Validation Rules.

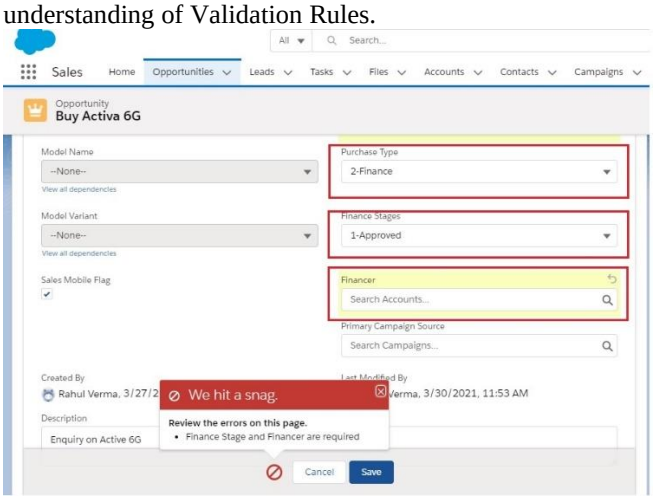

Fig 2. Example of validation rule fired while saving a record.

TABLE I. VALIDATION RULE EXAMPLES

Validation Formula	*Error Message*
`OR( ISBLANK(PhoneNumber__c), NOT(ISNUMBER(PhoneNumber__c)) )`	Please enter phone Number. It must contain only Numbers.
`AND( ISBLANK(StudentID__c), LEN(StudentID__c) <> 7 )`	Please enter the 7 digit student ID
`NOT( REGEX(PAN_Number__c, "[A-Za-z]{5}[0-9]{4}[A-z]{1}" ) )`	Please enter 10 character PAN card in format AAAAA1111B

[a] Validation Rules contain in-built formulas provided by Salesforce

VIII. DATA IMPORT AND EXPORT IN SALESFORCE

Data transactions with external system can be done with ease using the native Salesforce tools like Data Import Wizard, Data Export. We can also leverage third party software like Dataloader.io and Salesforce Workbench to import, export, delete, insert, update records.

To import Data into Salesforce, the procedure mentioned below is followed:

1) Create a Comma Separated Values file[CSV] which contains all the data to be inserted into Salesforce. Make sure to align the data type of the record's columns to the field in Salesforce.
2) Login to the Salesforce and Navigate to Setup
3) Go to Data Import Wizard
4) Choose the object whose data is to be inserted.
5) Upload the CSV file from the local computer to Salesforce.
6) Map the columns in CSV file to the fields in Salesforce
7) Click Begin Import to import the data into Salesforce.

To export Data, Navigate to Data Export and data can be easily exported into required file format. Salesforce also provides an option to schedule Data export on a frequency basis like a week/monthly etc. So that even if there are issues in the Salesforce Org. The data is safely stored as there is an option to schedule data export.

Dataloader.io and Salesforce Workbench are third-party software that also provides similar functionality. A SOQL query[similar to SQL query] can be used to export Data from these third-party applications. There is an option that an email notification is sent to the User once the task has been completed. To update Data using a CSV file we need to have the unique 15-digit Salesforce ID so that Salesforce can easily recognize the record to be updated. The bulk API of Salesforce can be used while dealing with bigger volumes of Data.

IX. DATA MANIPULATION LANGUAGE IN SALESFORCE

DML statements are very common in SQL databases. The DML options present in Salesforce are insert, delete, update, upsert, merge, undelete statements.

1. Insert: It inserts new records into the Database.
2. Delete: It removes the records from database.
3. Update: It updates the values in the existing record.
4. Upsert: This statement handles both insert and update operations in a single transaction.
5. Merge: It is used to merge up to three records and merge into one and removing the other two records.
6. Undelete: Once the records have been deleted in Salesforce. They are present in the recycle bin for few weeks before they are permanently deleted. During this time where records are in recycle bin, we can use the undelete option to restore the records back into the database.

Sample syntax of DML statements:

```
List<Account> recordList = new List<Account>( );
recordList.add(new Account(name='test Account 1'));
recordList.add(new Account(name='test Account 2'));

1)
try {
   insert recordList;
} catch (DmlException e) {
   System.debug('Error occurred while insertion
'+e.getMessage());
}
```

Similar to try-except method in python, Salesforce has try-catch feature.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

In this method, if a single record encounters exception while insertion, The whole list will not be committed to the Database. To overcome this disadvantage, we have Database.insert option.

2)`Database.SaveResult[] results = Database.insert(recordList, false);`

Here results list stores the operation result of the insertion of each record i.e whether it was inserted successfully, or if an exception occurred while inserting. There are two parameters in the Database.insert method. The first one, Recordlist is the list that contains data to be inserted into Salesforce and the second one is a Boolean input symbolizing AllorNone feature. If AllorNone feature is set to true, none of the records are committed into Salesforce, if even a single record throws an error while insertion. If the Boolean parameter is set to false, the error-free records are inserted, and skip the records that throw an error/exception. Transaction result of each record operation is stored in the results list.

X. SOQL

SOQL stands for Salesforce Object Query Language. SOQL has many similarities in terms of syntax with traditional SQL statements. The Structured Query Language (SQL) is the most extensively used database language. SQL is composed of a data definition language (DDL), which allows the specification of database schemas; a data manipulation language (DML), which supports operations to retrieve, store, modify and delete data[3]. SOQL queries help to shape custom queries to fetch records from the database. It has functionalities like SELECT, FROM, WHERE, ORDER BY, GROUP BY, LIKE, LIMIT, OR, AND, etc. similar to Structured Query Language [SQL]. SOQL queries can also be used to query custom metadata and big objects. It also has aggregative functions like Average, Count, Min, Max, Sum, etc. which can be used to compute aggregated results. Below are a few examples of SOQL statements for better understanding.

```
1) SELECT Id, Name, Phone, Email FROM Account

Select the fields ID, Name, Phone and Email from
the object Account

2) SELECT Name from Contact WHERE Name LIKE
'%Anand%'

Select the Name of the Contact Object whose name is
a close match to string containing 'Anand'

3) SELECT Count(Id),State__c FROM UserDetails__c
GROUP BY State__c

Returns the count of records for each state. This
is an example of Aggregative functions.

4) SELECT Id, FirstName__c, LastName__c, State__c,
UserIDNumber__c FROM UserDetails__c where (State__c
='Karnataka' OR State__c ='Punjab') AND
(UserIDNumber__c > 100) LIMIT 10

Select the fields from User Details Object for the
States Karnataka and Punjab with a User ID number
higher than 100 and restrict the number of records
retrieved from database to 10.
```

XI. APEX PROGRAMMING

The Apex programming language is an Object Oriented Programming language with Java-like Syntax used to interact with databases for web services, email services, record retrieval, data updating, trigger function, automation, executing flow, etc.

Classes can be constructed in Salesforce within which Methods are written similar to a STORED PROCEDURE in SQL. Apex provides the power to configure complex business logic by providing a variety of data types, collection types, flow control statements, SOQL queries, DML statements, Looping statements, etc. Apex Classes can be written using primarily two options. First, by using the Salesforce Org Developer Console, which is a cloud-based development platform. Secondly, using Integrated Development Environments [IDE] like VSCode, etc.

Let us understand the basic structure of Apex programming by looking into features like data types, collection types, flow control statements, SOQL queries, DML statements, and Looping statements.

A. Primitive Data Types

a. Integer: A 32-bit number that does not include a decimal point. Integers have a minimum value of -2,147,483,648 and a maximum value of 2,147,483,647.

b. Blob: A collection of binary data stored as a single object. You can convert this data type to String or from String using the toString and valueOf methods, respectively. Blobs can be accepted as Web service arguments, stored in a document (the body of a document is a Blob), or sent as attachments.

c. Time: A value that indicates a particular time. Always create time values with a system static method.

d. Decimal: A number that includes a decimal point. Decimal is an arbitrary precision number. Currency fields are automatically assigned the type Decimal.

e. Double: A 64-bit number that includes a decimal point. Doubles have a minimum value of -263 and a maximum value of 263-1.

f. String: Any set of characters surrounded by single quotes is called String.

g. Boolean: It stores the Boolean values i.e. True or False

h. Date: Stores the date

i. ID: It is the unique 15/18 character long case-sensitive Salesforce ID assigned to each record, report etc.

j. Object: This Data type stores the Object data. The standard and custom object data can be stored inside this data type

Examples of the syntax for the primitive datatypes.

```
Integer Count=1;

String myString = 'StringToBlob';
Blob myBlob = Blob.valueof(myString);

newInstance(hour, minutes, seconds, milliseconds)
Time expected = Time.newInstance(4, 2, 3, 4);

Double d=3.14159;
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

```
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.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

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

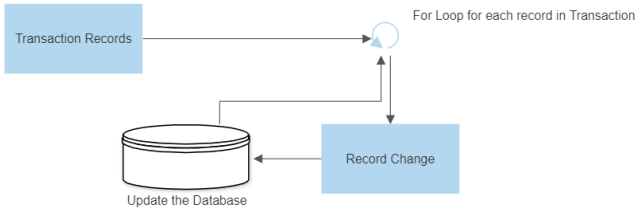

Fig 3. Approach of Non Optimized Code

```
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

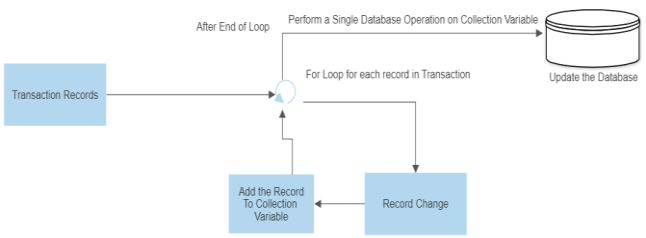

Fig 4. Approach of the Optimized Code

```
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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

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.

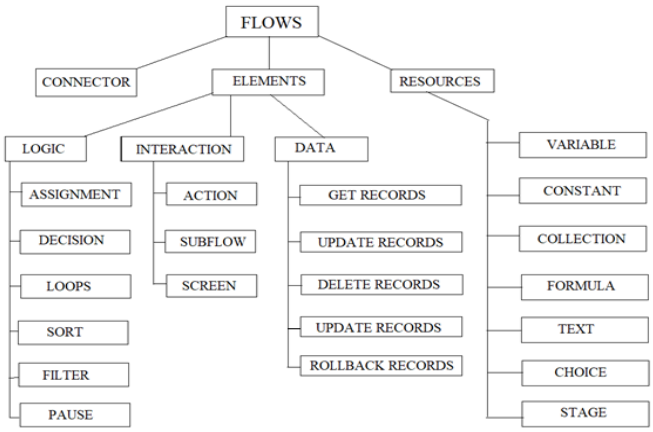

Fig 5. Flow Components

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

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.

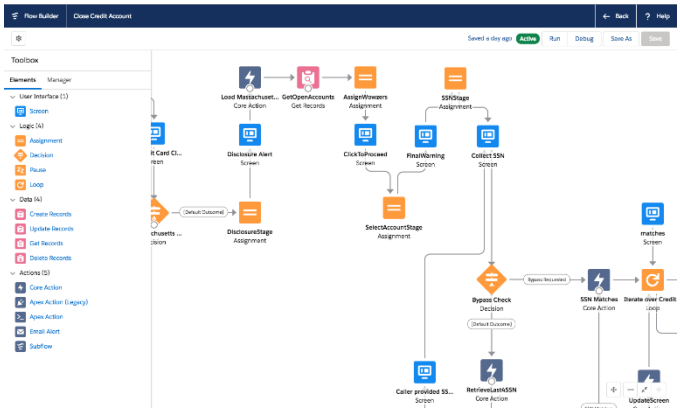

Fig 6. A Typical Salesforce Flow Representation

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.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 09, September 2022

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] SalesforceWebsites

https://www.salesforce.com/
https://trailhead.salesforce.com/
https://developer.salesforce.com/
https://www.lightningdesignsystem.com/

[12] Arockia Panimalar.S, Priyadharshan.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