

iDataMasker – TECHNICAL DETAILS

Purpose and key objectives:

Data masking in Salesforce sandboxes serves several important purposes, mainly centered around safeguarding sensitive information, ensuring compliance, and facilitating secure development and testing. Here are the key purposes of implementing data masking in Salesforce sandboxes:

1. Security and Compliance

Adhering to data protection regulations (such as GDPR, HIPAA, etc.) by preventing exposure of sensitive data in non-production environments, thus avoiding potential legal and regulatory issues.

2. Data Privacy

Protecting Sensitive Information: Masking personally identifiable information (PII) and other sensitive data to ensure the privacy of individuals and prevent unauthorized access.

3. Data Breach Prevention - Risk Mitigation

Reducing the risk of data breaches in non-production environments, where security measures may not be as stringent as in production.

4. Development and Testing

Helping the realistic test scenarios by enabling developers and testers to work with realistic data without exposing sensitive details, ensuring the accuracy of testing scenarios.

5. User Training - Safe Training Environments

Providing a secure environment for user training sessions, demonstrations, and other educational activities where actual data is not revealed.

6. Third-Party Collaboration

Facilitating collaboration with external partners, vendors, or contractors by limiting their access to sensitive data in sandboxes.

7. Customer Trust and Reputation

Demonstrating a commitment to data security and privacy, which can enhance customer trust and protect the organization's reputation.

Key Objectives of iDataMasker:

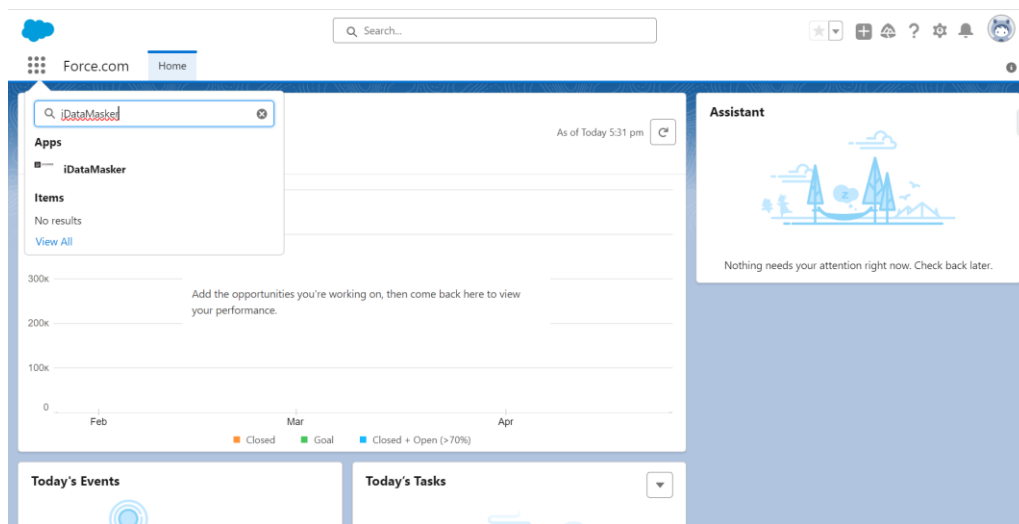
The primary objective of an iDataMasker app in Salesforce is to protect sensitive information while maintaining the functionality and integrity of the application. Data masking in Salesforce sandboxes is essential for several reasons, primarily revolving around the need to protect sensitive information, comply with regulations, and maintain a secure and trustworthy development and testing environment to ensure compliance with data protection regulations (e.g., GDPR, HIPAA) by masking sensitive data and preventing unauthorized access or exposure in all the salesforce sandboxes. Also, to safeguard sensitive information from internal and external threats, minimizing the risk of data breaches or unauthorized access in Salesforce environments.

Installation Guide

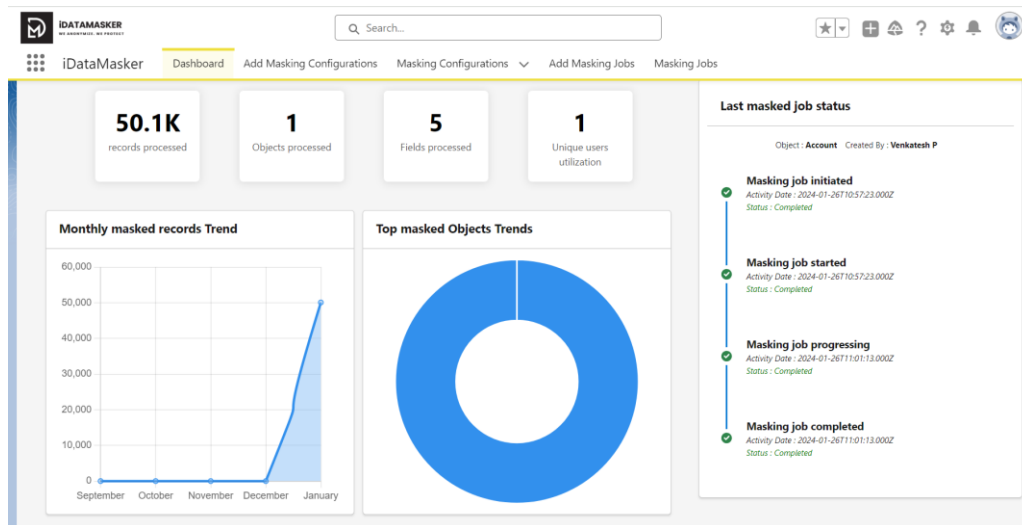
1. The 'iDataMasker' app is conveniently accessible on the Salesforce AppExchange page, where users can install the application by navigating to the installation screen and clicking the 'Install' button.
2. After clicking the 'Install' button, select the sandbox you want to install. During this installation process, it is crucial to carefully select the desired sandbox for installation, emphasizing that the app should not be installed in any production org.
3. For a streamlined installation experience, users can directly navigate to the following link: **[https://<mydomain>.sandbox.lightning.force.com/packaging/installPackage.apexp?p0=<packageId \[TBC post security review\]>](https://<mydomain>.sandbox.lightning.force.com/packaging/installPackage.apexp?p0=<packageId [TBC post security review]>)**. Once on the page, simply click the 'Install for Admins' button, ensuring a quick and efficient installation process for the 'iDataMasker' app in the designated Salesforce sandbox.
4. Following the successful installation of the package, users are required to assign a permission set named '**iDataMasker Administrator.**' This permission set has all the required object, fields and Lightning page permissions related to 'iDataMasker' app.

How to use the app

1. Once the installation and permission set assignment are done, Search for the 'iDataMasker' in the app console and click it to open the iDataMasker app.



- Once the iDataMasker app opened, you can see the default page 'Dashboard' and that shows some statistics and graph with the current statistics.



Dashboard page, you can see how many records that are masked so far in the sandbox. This may be set to 0 once you installed the package and will be shown some numbers once you started masking the records.

Also, the dashboard shows how many objects and fields you have masked in the sandbox and how many users used the iDataMasker app to mask the records.

The Right-side panel of the dashboard shows the statistics about the last masking job executed in the org.

- Add Masking Configurations

Before start masking the records, it is required to add the masking configurations to define what object and fields in the objects to be masked. You can be able to mask the objects only after you add them in the masking configurations.

The screenshot shows the 'Add Masking Configurations' page in iDataMasker. It has a search bar and navigation tabs: iDataMasker, Dashboard, Add Masking Configurations (selected), Masking Configurations, Add Masking Jobs, and Masking Jobs. The main section is titled 'Select the Object and Fields to be Masked' with the instruction 'Configure settings for applying masking techniques.' It contains two dropdown menus: 'Object Type' (set to 'All') and 'Object' (set to 'Account'), followed by a 'Submit' button. Below this is a table titled 'Account Object fields available for masking' with 15 rows and 3 columns: Label, API Name, and Data Type. The visible rows are: D-U-N-S Number (Text(9)), Fax (Fax), Data.com Key (Text(20)), and NAICS Code (Text(8)). A 'Submit' button is at the bottom. On the right, the 'Account object details' panel shows: Select object name (Account), Total fields (66), Custom fields (8), Total standard fields (58), and Fields already masked (6).

Label	API Name	Data Type
☐ D-U-N-S Number	DunsNumber	Text(9)
☐ Fax	Fax	Fax
☐ Data.com Key	Jigsaw	Text(20)
☐ NAICS Code	NaicsCode	Text(8)

Once the object is selected, in the right-side panel, you can see the selected object's field statistics.

Once the object is selected, a table of fields will be displayed for you to select them for the masking. You can select the fields you need to mask in the selected object.

If any field is already selected for masking, you can't see those fields in the masking those will be listed in another table which have the fields that are already added into the masking configurations.

The screenshot shows the iDataMasker web application interface. The top navigation bar includes the iDataMasker logo, a search bar, and several icons. The main content area is titled 'Account Object fields available for masking' and displays a table with 15 rows. The table has columns for Label, API Name, and Data Type. The fields listed are SIC Description, Account Site, Ticker Symbol, and Year Started. A 'Submit' button is located below the table. Below the main table, there is a section titled 'Existing masking configurations added for the selected object' which displays a table with 6 rows. This table has columns for Label, API Name, Data Type, and Created On. The fields listed are Tradestyle, Account Number, Annual Revenue, Description, Account Name, and Website.

Label	API Name	Data Type
☐ SIC Description	SicDesc	Text(80)
☐ Account Site	Site	Text(80)
☐ Ticker Symbol	TickerSymbol	Content(20)
☐ Year Started	YearStarted	Text(4)

* Field types such as Formula, Lookup, Checkbox, Picklist, Record Type, Hierarchy, External ID, Unique, Roll-Up Summary and System dates cannot be masked due to the salesforce limitations

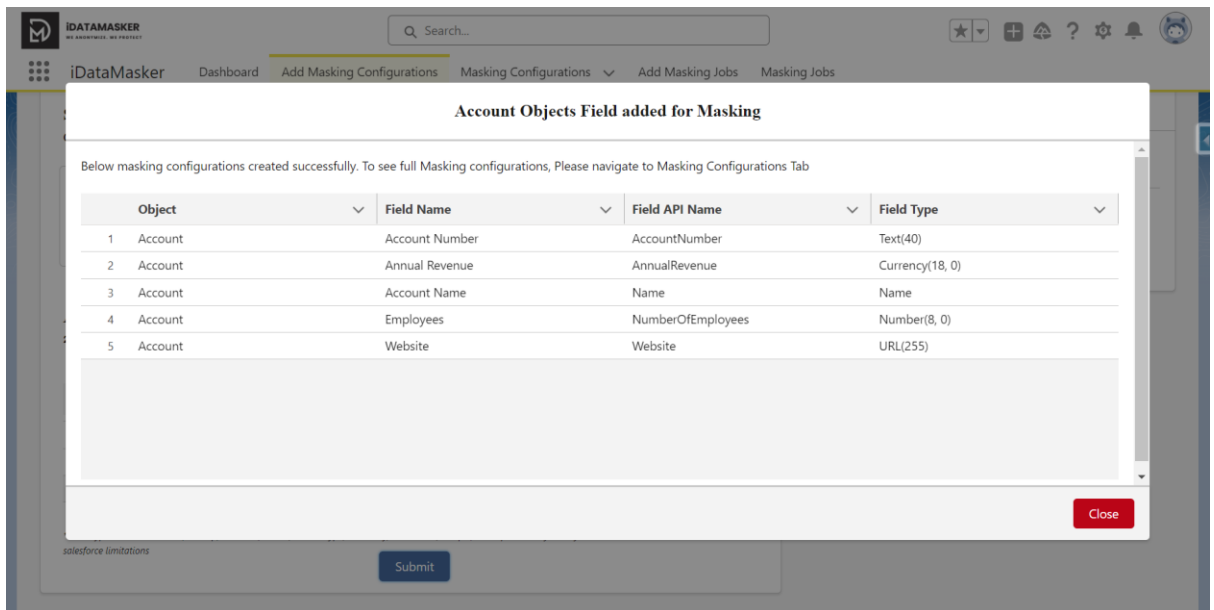
Submit

Label	API Name	Data Type	Created On
1 Tradestyle	Tradestyle	Text(255)	2024-01-25 20:39:25
2 Account Number	AccountNumber	Text(40)	2024-01-23 16:46:47
3 Annual Revenue	AnnualRevenue	Currency(18, 0)	2024-01-23 16:46:47
4 Description	Description	Long Text Area(32000)	2024-01-23 16:48:06
5 Account Name	Name	Name	2024-01-23 16:48:06
6 Website	Website	URL(255)	2024-01-23 16:48:06

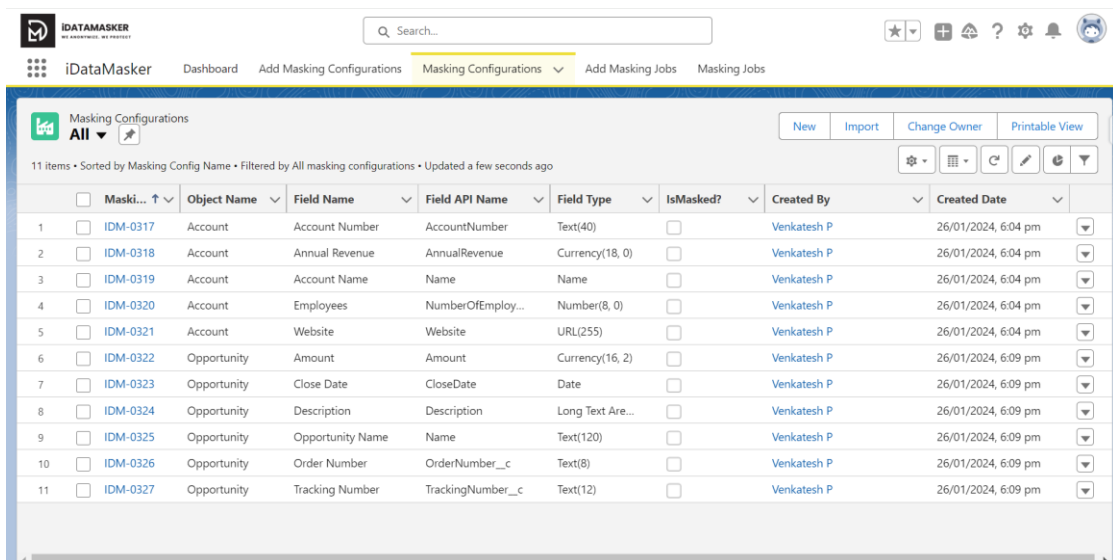
Note: The table only shows the fields that are already added in the masking configurations and used for masking when submitting the masking job. The fields already added in the masking configurations may or may not be used in the previous masking jobs for the selected object.

Also, field data types such as Formula, Lookup, Checkbox, Picklist, Record Type, Hierarchy, External ID, Unique, Roll-Up Summary and System dates cannot be masked due to the salesforce limitations.

4. Once the fields are selected, you can click the submit button and after the fields added into the masking configurations, you will get a pop-up with the fields details that are added into the masking configurations.



- After successfully added the fields for the selected object into masking configurations, there is a tab called, masking configurations which is standard salesforce list page which shows the list of fields that are added into the masking configurations across all objects. If you want to remove any field from being masked after added them into the masking configurations, you can navigate to this Masking Configurations page and can delete them from here.

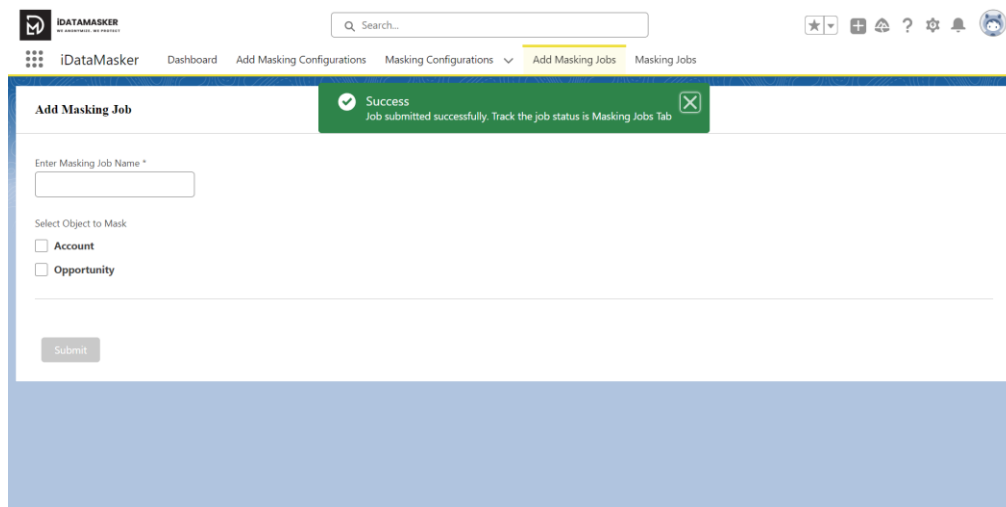


- After adding the Masking configuration, you need to create a masking job which will do the masking for the records under the object. For that, you need to navigate to 'Add Masking Job' tab.

7. You will be able to select object only if you have added at least one field of the object in the masking configurations. After you added at least one field in the masking configuration, the object will be displayed in the 'Add Masking Job' page.
8. In the Add Masking Job page, you need to enter the Masking Job name which can be any custom defined name which suits for the masking process and select the object for the masking. You can either select all the objects displayed in the screen or you can select specific object from the displayed list.
9. Only the selected objects will be submitted for the masking and masking will happen to only these selected object records.
10. When you select any object, a text box will appear in the downside of the object after you select it which allows you to write a filter query to filter records under the object. This option will give you a control for the Surgical masking of selected records.

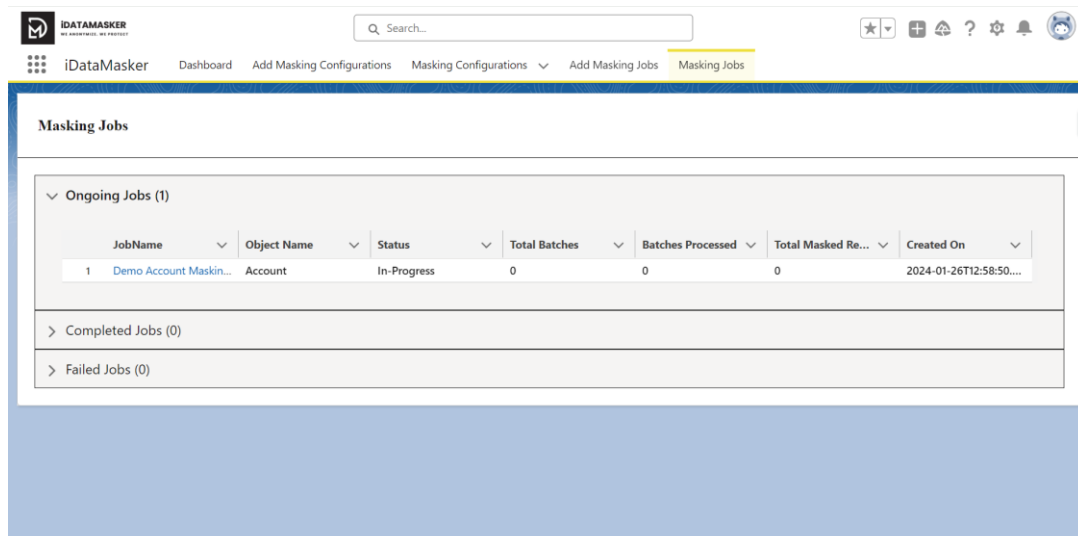
So, the masking of records will happen only to the records that comes under the scope of the filter query you have added. If the filter query structure is wrong, the job will be failed.

11. If you need to mask all the records under the selected object, you can leave the textbox as blank which means all the records under the selected object will be in a scope for the masking.



Once you submit the job, a record will be created under the custom object called 'Masking Job' which tracking all the records that are under the scope of the selected object.

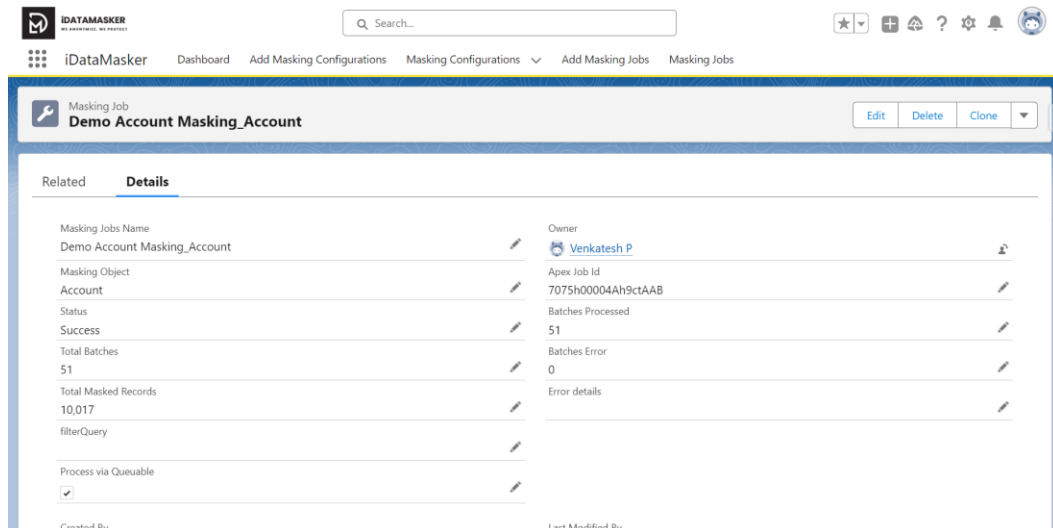
12. After the masking job successfully submitted, you can navigate to the Masking Job tab and see the list of available jobs in all the 3 categories. (On-going, Completed and Failed jobs).



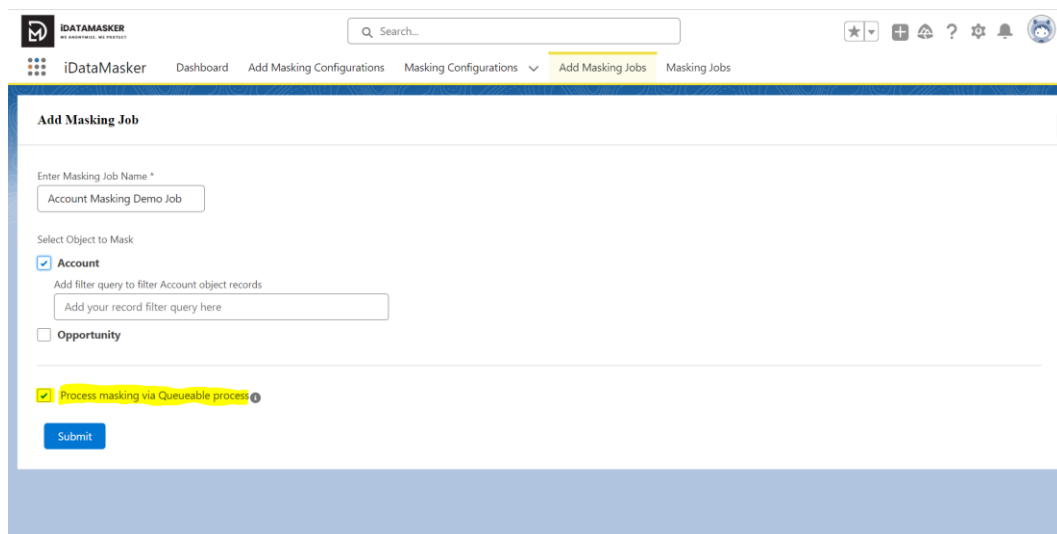
JobName	Object Name	Status	Total Batches	Batches Processed	Total Masked Re...	Created On
1 Demo Account Maskin...	Account	In-Progress	0	0	0	2024-01-26T12:58:50....

Each section will show you a table of jobs with few details Like Status, Total Batches, Batches Processed and Total Masked Records. You can navigate to the record page by clicking the job name.

13. Once you clicked the job name from the Masking Jobs page, you will be navigated to the selected Masking Job record page that will show you more details of the job like Status of the job, Batches processed, Batches failed, Total masked records, Filter Query (if you have added any in the Add Masking Job Page), Failed batches, Failure reason if any.



14. When you select the Object in the Add Masking Job Page, if the Selected object has more than 10,000 records, there will be a checkbox option enabled for you. This is to process the records through salesforce queueable jobs. Generally, when you process the jobs with batch class, each batch will run one by one which means one batch will be executed after another batch. This might be time consuming. When you select this option, each batch will be executed as a separate queueable and will run parallelly. So, the processing will be very quick.



NOTE: Enabling the queueable process will be counted against your org's daily asynchronous limit and not recommended to use this option frequently. Sometimes, you might get a notification from Salesforce saying your org's maximum concurrent apex limit has been reached. SO, enabling this option is your own risk. If you have a greater number of records and wanted to process them quickly, you can go with this option and otherwise it is recommended to go with the batch job.

15. When you see the job status is 'Success' which means, the masking is done for the object and ready for the sandbox users to access it.

16. It is recommended that during the Masking Job execution, not to leave the sandbox for end user access or usage. Usage of the sandbox during the masking job execution, might slowdown the process or leads to job failure.

Masking algorithm:

The masking algorithm we are using to mask your records are fully customized which includes combination of multiple algorithms like Key based Encryption, Substitution, Shuffling, and Randomization.

1. Key-based Encryption

Involves using a cryptographic key to transform the original data into an unreadable format. Requires a key to decrypt and reveal the original information. Provides a high level of security when implemented correctly.

2. Substitution

It involves replacing sensitive data with fake or anonymized values. It can be a one-to-one substitution or involve more complex mapping rules. This technic helps protect the original data while maintaining the overall structure.

3. Shuffling

Entails rearranging the order of elements in the dataset. This can be applied to rows, columns, or individual characters, depending on the context. Introduces randomness and further complicates attempts to reconstruct the original data.

4. Randomization

It adds a layer of unpredictability by introducing random values or noise. It Can include the insertion of random characters or values within the masked data. It enhances the overall security of the masked data.

NOTE: With our customized masking algorithm, once the data is masked it cannot be demasked back to original value.

List of Salesforce components and it's usage in the package

#	Component	API Name	Type	Usage
1.	iDataMasker	iDataMasker	Console app	This console app contains all the tabs used for the iDataMasker app.
2.	Dashboard	Dashboard	Lightning Component Tab	This is the default tab in the iDataMasker app which shows the current org statistics and graphs
3.	Add Masking Configurations	Add_Masking_Configurations	Lightning Component Tab	This is the 'Add Masking Configuration' tab in the iDataMasker app allows you to create masking configurations
4.	Add Masking Jobs	Add_Masking_Jobs	Lightning Component Tab	This is the 'Add Masking Job' tab in the iDataMasker app allows you to add masking jobs
5.	Masking Configurations	Masking_Configurations	Custom object Tab	This is standard custom object tab to show all the

				available masking configurations in the org.
6.	Masking Jobs	Masking_Jobs	Custom object Tab	This is 'Masking Jobs' tab in the iDataMasker app which allows you to see the list of On-going, completed, failed app in the org.
7.	ChartJS	ChartJS	Static Resource	JavaScript library for the Graphs showing in the iDataMasker app Dashboard tab
8.	Masking Configuration	Masking_Configuration__c	Custom Object	Used to store all the masking configurations created in 'Add Masking Configuration' Tab
9.	Object Name	Object_Name__c	Custom field	Masking Configuration object field with data type Text (255) which hold the masking object name
10.	Field Name	Field_Name__c	Custom field	Masking Configuration object field with data type Text (255) which hold the masking object field label
11.	Field API Name	Field_API_Name__c	Custom field	Masking Configuration object field with data type Text (255) which hold the masking object field API name
12.	Field Type	Field_Type__c	Custom field	Masking Configuration object field with data type Text (255) which hold the masking object field data type
13.	IsMasked?	IsMasked__c	Custom field	Masking Configuration object field with data type Checkbox) which hold the details of the field whether the field is part of existing masking jobs
14.	Masking Config Layout	Masking Config Layout	Page Layout	Masking Configuration record default page layout
15.	All	All	List View	List view in Masking Configuration Tab to show all the masking configurations with required columns
16.	Masking Job	Masking_Job__c	Custom Object	Custom object used to store the submitted masking job related details
17.	Apex Job Id	ApexJobId__c	Custom field	Field in the masking job which capture the apex Job Id created when the masking job is submitted for masking
18.	Masking Object	Masking_Object__c	Custom field	Holds the object name when the masking job is created.
19.	filterQuery	filterQuery__c	Custom field	Holds the filter query when the Masking job is created for an object
20.	Batches Processed	BatchesProcessed__c	Custom field	Holds how many batches processed in the masking job
21.	Batches Error	BatchesError__c	Custom field	Holds How many batches failed in the masking job
22.	Error details	ErrorDetails__c	Custom field	Holds the error detail of the masking job if any
23.	Status	Status__c	Custom field	Holds the current status of the masking job
24.	Total Batches	TotalBatches__c	Custom field	Holds the total number of batches in the masking job
25.	Total Masked Records	TotalMaskedRecords__c	Custom field	Holds the total records masked of part of the masking job

26.	Process via Queueable	ProcessViaQueueable__c	Custom field	Holds the details whether the job can be processed via queueable.
27.	Masked record	Masked_record__c	Custom Object	This is a child object of Masking Job object and this object have lookup field to Masking job. For each masking job record, one entry will be created in the masked records.
28.	Masking Job Layout	Masking Job Layout	Page Layout	Default layout for Masking Job Object.
29.	MaskingJobTrigger	MaskingJobTrigger	Apex Trigger	Trigger on Masking Job which create the Batch job for masking
30.	MaskedRecord	MaskedRecord__c	Custom field	A formula field what holds the hyper link to the masked record
31.	Masking Job	Masking_Job__c	Custom field	Lookup field to the Masking Job
32.	Number of time masked	NumberOfTimeMasked__c	Custom field	Used for the backend purpose. Nothing has been calculated with this value.
33.	RecordId	RecordId__c	Custom field	Record Id of the masked record
34.	RecordName	RecordName__c	Custom field	Name of the Masked record used in the Masked record formula field
35.	Masked record Layout	Masked record Layout	Page Layout	Default Layout for the Masked Record object
36.	MaskingUtils	MaskingUtils	Apex Class	This apex class is used for all the Utility purpose, and it have all the @AuraEnabled methods which called from the LWC components. This class is running in the With Sharing mode and enforces the current logged-in user field and object level security.
37.	ObjectsFieldsFetcher	ObjectsFieldsFetcher	Apex Class	This apex class mostly deal with the subject and subject field retrieval related queries and logics. This class is called from the LWC component which is added in the Add Masking Configuration Tab.
38.	ObfuscationBatch	ObfuscationBatch	Apex Class	This is a batch class that will be called when a Masking job is created. This Batch has default batch size as 200. This batch called from the Masking Job trigger.
39.	ObfuscationAlgorithm	ObfuscationAlgorithm	Apex Class	This apex class is the brain of the app and holds the logic of the masking. This class is getting called from ObfuscationBatch class.
40.	LogUtility	LogUtility	Apex Class	This is another utility class created to log the errors and update the record when a specific batch or job is completed. This class is called from ObfuscationBatch.
41.	dataMaskerHome	dataMaskerHome	LWC	This component is built for the Dashboard Tab in the iDataMasker app.
42.	maskingConfig	maskingConfig	LWC	This component is built for the Add Masking Configuration tab in the app.

43.	addMaskingJob	addMaskingJob	LWC	This component is created for the Add Masking Job in the iDataMasker app.
44.	iDataMasker Administrator	iDataMasker_Administrator	Permission Set	This is the permission set that holds the access to all the above-mentioned components