



## **Salesforce's Big Object**

## Table of Contents

Problem Statement.....	3
Big Object Classification & Naming Convention.....	4
Implementation of Big Object.....	5
Way's to populate Data & Delete data in Big Object.....	7
Querying Big Object.....	8
Visualization Big Object data.....	9
Use case Consideration.....	10
Considerations in Big Object.....	13
Big Objects Used In Existing Technology.....	16
Frequently Asked Questions on big Object.....	17

## Problem Statement

Big data is a big deal! So much so, we can often find ourselves with too much data. Large data sets make it challenging to effectively “crunch the numbers”, or manage performance and scalability



Companies generate a huge amount of data of the customers. This data is useful for the company for various reasons such as data analysis, auditing, etc. so to store this data we either need to buy the additional storage or should need to use some external system and integrate it. It is not feasible to use the external

application for the storage of companies that generated millions of data and also to buy the storage as it is expensive. As salesforce always thinks about their customers they started working on it and innovation happen in 2018 with the big object.

These Big objects archive and manage massive data volumes within Salesforce without affecting performances, all with a processing scale of billions of records.



Big object storage does not count against Org Data Storage

# Big Object Classification & Naming Convention

## Naming Conventions for Custom Big Object

The API names of custom Big Object are identified by a suffix of two underscores immediately followed by a lowercase "b" (\_\_b).

For example, Code Review History is a Big Object, then its API name will be "Code\_Review\_History\_\_b"

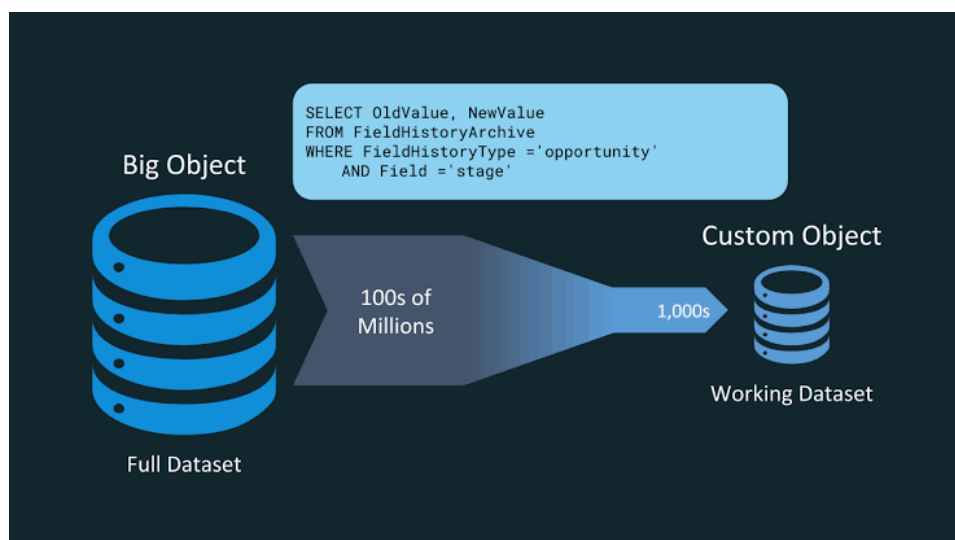
## Classification of Big Object

- Standard big object

Standard big objects are objects that are defined by Salesforce. It cannot be customized. It is not available in all org's & it required additional user permission to process it. It does not support encryption & all data type.

- Custom big object

These are new objects that you create to store unique information to your org. It Consists of Object, Permission sets & package.

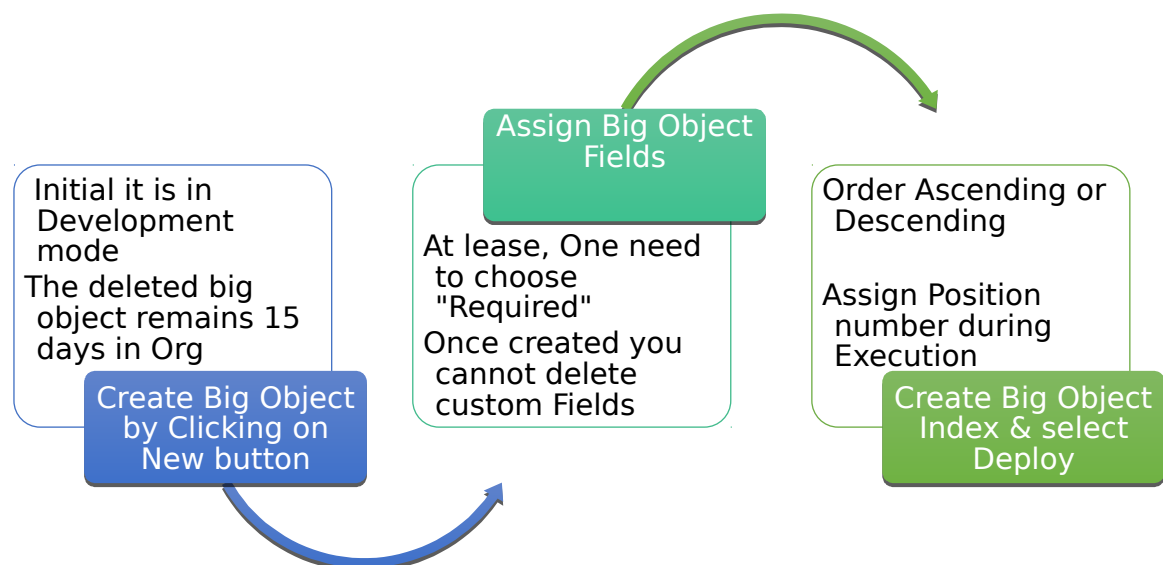


# Implementation of Big Object

## Steps to Create and Deploy Big Object

Similarly, If you aren't comfortable using the Metadata API in Latest Spring '19 Release, Salesforce has added the most awaited feature and offers an option to create custom big objects from the set-up.

- From salesforce Org in the quick find box, search for the big object.
- Create a big object by clicking on the New button appear on UI.
- While creating big object make sure you have selected in development mode in beginning, add rest details, click on save.
- Now also add custom fields & index fields that you require in big objects.
- Click on the Save button.
- Now, after all, Click Edit on the big object, and change the deployment status from In development to Deployed.



The following Image represents a Big Object view in Org.

Big Object  
Big Data Object1

[Standard Fields \[0\]](#) | [Custom Fields & Relationships \[0\]](#) | [Index \[1\]](#)

**Big Object Definition Detail** [Edit](#) [Delete](#)

Singular Label	Big Data Object1	Description	
Plural Label	Big Data Object1s	Deployment Status	Deployed
Object Name	Big_Data_Object1		
API Name	Big_Data_Object1__b		
Created By	anilgk HT, 1/13/2020, 12:55 AM	Modified By	anilgk HT, 1/13/2020, 1:10 AM

**Standard Fields**

No standard fields defined

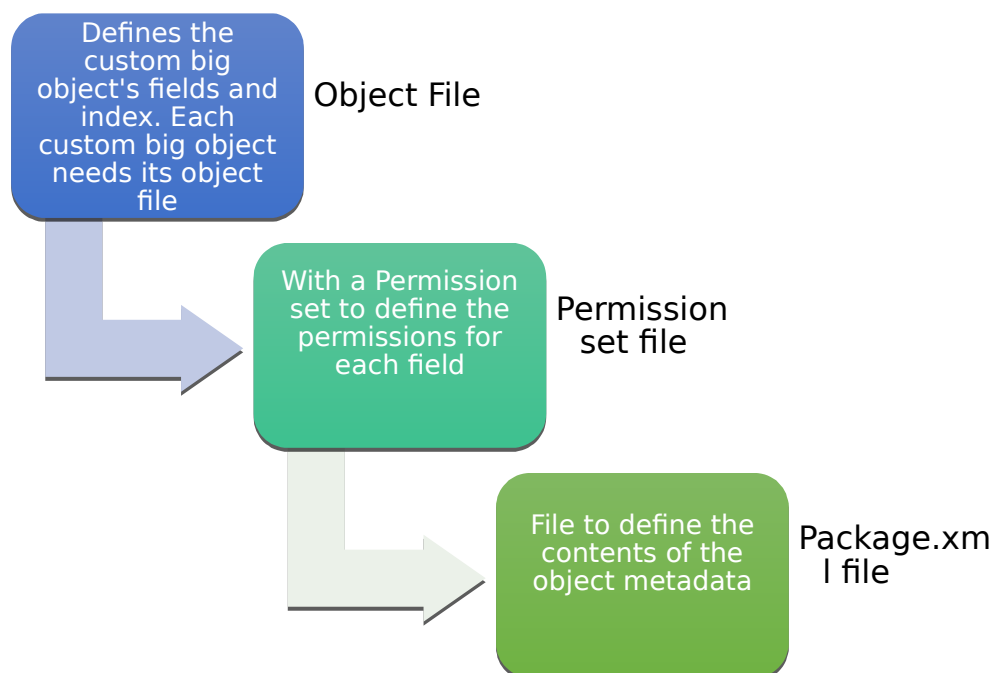
**Custom Fields & Relationships** [New](#)

Action	Field Label	API Name	Data Type	Indexed	Index Position	Index Direction	Modified By
<a href="#">Edit</a>	<a href="#">Account</a>	Account__c	Lookup(Account)				anilgk HT, 1/13/2020, 1:02 AM
<a href="#">Edit</a>	<a href="#">Description</a>	Description__c	Long Text Area(131072)				anilgk HT, 1/14/2020, 1:30 AM
<a href="#">Edit</a>	<a href="#">TaskCall</a>	TaskCall__c	Text(50)				anilgk HT, 1/13/2020, 12:57 AM
<a href="#">Edit</a>	<a href="#">Taskdatetime</a>	Taskdatetime__c	Date/Time				anilgk HT, 1/13/2020, 1:08 AM
<a href="#">Edit</a>	<a href="#">TaskStatus</a>	TaskStatus__c	Text(50)	✓	1	ASC	anilgk HT, 1/13/2020, 12:58 AM

**Index**

Action	Index Label	API Name	Type	Created By
<a href="#">View</a>	IndexField	IndexField__b	Primary	anilgk HT, 1/13/2020, 1:03 AM

Similarly, You Can Also Deployed Big object Through Workbench. Create the zip file for deployment which consists of the object file, permission set file& XML file.



# Way's to populate Data & Delete data in Big Object

## **Populate a Custom Big Object with Salesforce APIs**

You can use a .csv file and Bulk API or SOAP API to upload data into a big object via Workbench. The first row in the CSV file must contain the field labels used to map the CSV data to the fields in the custom big object during import.

## **Populate a Custom Big Object with Apex**

You can create and update custom big object records in Apex using the `insertImmediate` method. To test DML calls, use a mocking framework with the Apex stub API. To verify that all records are saved, check the `Database.SaveResult` class.

## **Delete Data in Custom Big Object**

Deleted big objects are stored for 15 days you can permanently delete it or you can restore that deleted big object within 15 days but after 15 days that temporarily deleted big object will get permanently deleted from your system. `DeleteIntermediate()` method is used to delete rows from the custom big object where we have to declare the `SObject` that is used in the index and also with the fields that map with the Custom Big Object by using the `WHERE` clause. Here the `SObject` acts like a template. All rows that match the `SObject` fields and values are deleted.

# Querying Big Object

## SOQL with Big Object



In querying big objects, SOQL is used when you have to retrieve a small amount of data and also when you want immediate results. It's challenging to work with such a huge amount of data you can say

approximately for billions of data and so The !=, LIKE, NOT IN, EXCLUDES, and INCLUDES operators are not supported in any query in the big object.



When you query big objects in Developer Console the ID is automatically generated. Remove ID while querying the big object from the developer console else it will return you empty id's.

## Asynchronous SOQL with Big Object

Asynchronous SOQL query is used to create the record and also to query. The asynchronous query can handle the used amount of data that is millions of potential record from the custom big object. Multiple queries run in the background, while the status of completion when the query is executed successfully and its process is completed.



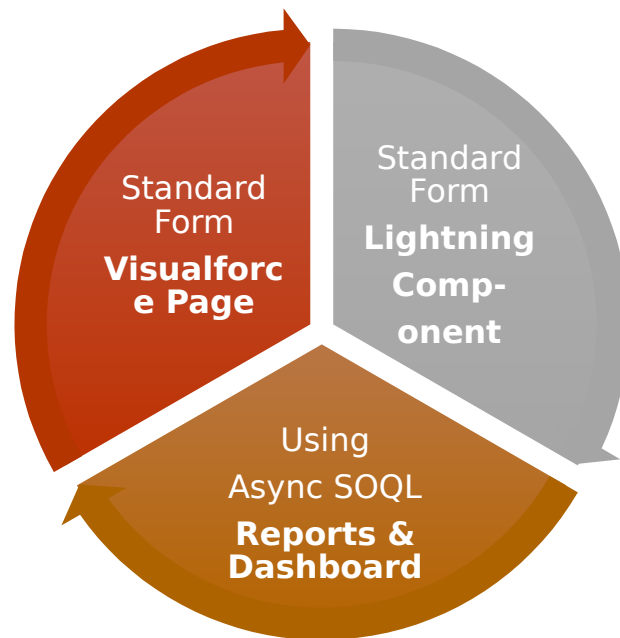
Remember that while custom big objects are included with every license, Async SOQL is included only with the licensing of additional big object capacity.



## Visualization Big Object data

Standard Form to view Big Object Data is,

- Visual force page,
- Lightning component.



The following Image shows the visualization of big object data through Lightning Component.

BIG OBJECT DATA				
Details of data store in Big Object				
Selected Contact:1				
<input checked="" type="checkbox"/>	TASKSTATUS	ACCOUNT	TASKCALL	TASKDATETIME
<input checked="" type="checkbox"/>	1 Market St		Smith	2020-01-13T00:00:00.000
<input type="checkbox"/>	Not Started	0012w000003iDkIAAU	Send Quote	2020-01-14T09:43:31.678
<input type="checkbox"/>	Waiting on someone else	0012w000003iC1rAAE	Email	2020-01-16T00:00:00.000

## Use case Consideration

The following are Some Ways for which I Can Use Custom Big Objects. Although you can use big objects to store different kinds of data, big objects were created to tackle a few specific scenarios,

- 360° view of the customer

You've got a lot of customer information you want to store. From loyalty programs to transactions, order, and billing information, use a custom big object to keep track of every detail.

- Auditing and tracking

Keep a long-term view of your users' Salesforce usage for analysis or compliance purposes.

- Historical archive

Maintain access to historical data for analysis purposes while optimizing the performance of your core CRM.

Let us Consider a Scenario where the business has a requirement to show Big Objects data as reports and dashboards.

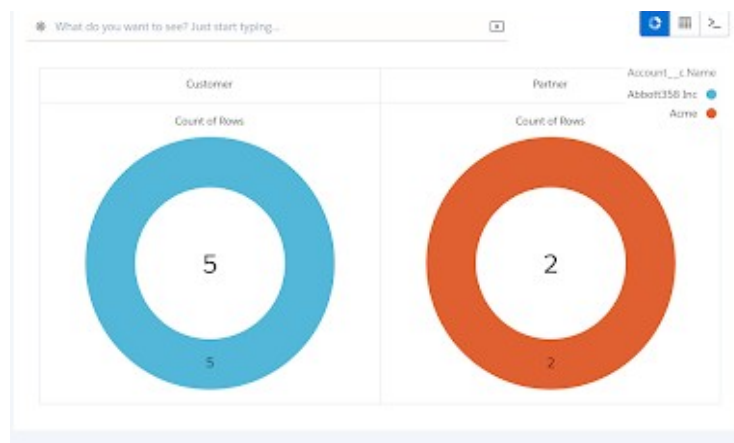


Big Objects which has the limitation of showing records in Salesforce Standard reports and dashboards

You can view Big Object data in Reports & Dashboard using the Asynchronous SOQL query. Create a custom object that holds the working data set for big object data that you want to report on. While creating custom objects just on optional features click on allow reports. Add Custom field's that match the fields that you want to report on from the big object.

Create an Asynchronous SOQL query that builds your working data-set by pulling the data from your big object into your custom object. During passing Asynchronous Query in workbench Map the source fields from the big object to the target fields in the custom object. Run the query. Build a report using the working dataset you created which extract the data from a created custom object

Similarly, you can use An Einstein Analytics app to display data of big objects in reports and dashboards. As the Einstein Analytics app is not free the starting price of Sales Analytics and Service Analytics is \$75 per user, per month. Following Image Show Representation Of Big Object data Using Einstein Analytics.



Similarly, Let Use Consider Scenario Where You want Big Object should display Existing Account's task records which are located in your Org.

Here You create Big Object to store all your task records in the big object. Create the fields whose data type is matching with Accounts' task records.



Remember while creating field big object has some limitation for a data type, if required you need to perform some workaround to access those fields.

Perform SOQL Query Using Big Object Standard Syntax to insert & retrieve the Account's Task Records. Once records are inserted in Big Object. Create a lightning component to view Big Object data. Big Object View Using Lightning Component is shown below.

1 Market St Smith 2020-01-13T00:00:00.000Z
In Progress 0012w000003iDkIAAU Send Quote 2020-01-02T00:00:00.000Z
Not Started 0012w000003nF7yAAE Call 2020-01-13T00:00:00.000Z
Waiting on someone else 0012w000003iC1rAAE Email 2020-01-16T00:00:00.000Z

When You delete this task records from Big Object then do not get delete From Org because big object storage is not counted against its org storage. So in this case, if any data deleted from the big object then the same data will not get deleted from org and Similarly vice versa.

# Considerations in Big Object

## Big Object Considerations

- Big Object has its storage (Storage Usage).
- You can define custom big objects only through the Metadata API.
- Big objects support only object and field permissions.
- Big objects ONLY support custom Salesforce Lightning and Visualforce components.
- You can create up to 100 big objects per org.
- To support the scale of data in a big object, you can't use triggers, flows, processes, and the Salesforce app.
- Custom big objects use the Custom Object metadata type.

## Field Type Considerations

Field type Supported in Big Object is,

Lookup,

Date/Time,

Email,

Number,

Phone,

Text,

Text Area (Long),

URL.

However, we can use workarounds to access field such as checkbox, Picklist, Master-detail data type in Big Objects

- Workaround for Checkbox & Picklist data type Field



- Workaround for Master-Detail data type Field



## Big object Index Considerations

- Create fields on the target Big Object that reference the source Big Object index fields.
- An index should include at least one custom field & can have a maximum of five custom fields. Custom fields that are marked as required can only be part of index fields.
- If you selected text data type then the total number of characters should not exceed after 100.
- Once the index is created you can edit it and change.
- You need to create new big objects for new index creation.

- In big object creation, planning & future, thinking is required to create the index as an index not give you a chance to edit or delete or add new once it is created.
- In the index, you can also assign position & order of that respective selected index field. Once assign you can't change it.



From API 40.0 indexes on Big Objects are compulsory

# Big Objects Used In Existing Technology

## **Big Object Utility Component by Cloud Bat**

Utility to view, filter and export big object record with ease. This component lets the user of Salesforce view, filter and export big object data with ease.

## **DataArchiva by Ceptes Software, Inc**

A Native Data Archiving Solution for Salesforce powered by Big Objects. With Auto-scheduler, Custom archiving policy & One-click restore, DataArchiva helps reduce data storage costs, enhance CRM performance & meet compliance.



# Frequently Asked Questions on big Object

- Big Object Pricing

Enterprise, Performance, Unlimited, and Developer Editions for up to 1 million records. Additional record capacity and Asynchronous SOQL query available as an add-on license. Limit of 1 Million Storage Capacity lead Additional cost of AUD 16,800 per year for per 50M records