Big Object

**что это** — тип данных в СФ который позволяет хранить до 1 билионна данных  
 !!! по умолчанию 1 миллион записей  
 - доп колво записей и ассинхронный SOQL по доп лицензиям  
  
**какие типы есть** — два типа BO  
 **Standard big object** - созданный СФ и включенный в продукты СФ Пример FieldHistoryArchive - stores data as part of the Field Audit Trail product

Custom big objects — то что создает пользователь  
  
**для чего можно использовать** 360° view of the customer—Extend your Salesforce data model to include detailed information from loyalty programs, feeds, clicks, billing and provisioning information, and more.

Auditing and tracking—Track and maintain a long-term view of Salesforce or product usage for analysis or compliance purposes.

Historical archive—Maintain access to historical data for analysis or compliance purposes while optimizing the performance of your core CRM or Lightning Platform applications.

**Сравнение с обычными обьектами**

| Big Objects | sObjects |
| --- | --- |
| Horizontally scalable distributed database | Relational database |
| Non-transactional database | Transactional database |
| Hundreds of millions or even billions of records | Millions of records |

**Из предыдущего следуют ограничения**  
 - Big objects support only object and field permissions, not regular or standard sharing rules.

- Features like triggers, flows, processes, and the Salesforce app are not supported on big objects.

- When you insert an identical big object record with the same representation multiple times, only a single record is created so that writes can be idempotent. This behavior is different from an sObject, which creates a record for each request to create an object.

* To define a big object or add a field to a custom big object, use either Metadata API or Setup.
* Big objects support custom Lightning and Visualforce components rather than standard UI elements home pages, detail pages, list views, and so on.
* You can create up to 100 big objects per org. The limits for big object fields are similar to the limits on custom objects and depend on your org’s license type.
* You can’t use Salesforce Connect external objects to access big objects in another org.
* Big objects don't support encryption. If you archive encrypted data from a standard or custom object, it is stored as clear text on the big object.

Best Practice  
 - The best practice when writing to a big object is to have a retry mechanism in place. Retry the batch until you get a successful result from the API or Apex method. *(The big objects database stores billions of records and is a distributed system that favors consistency over availability. The database is designed to ensure row-level consistency. )*

- Don’t try to figure out which records succeeded and which failed. Retry the entire batch.

- Big objects don’t support transactions. If attempting to read or write to a big object using a trigger, process, or flow on a sObject, use asynchronous Apex. Asynchronous Apex has features like the Queueable interface that isolates DML operations on different sObject types to prevent the mixed DML error.

- Because your client code must retry, use asynchronous Apex to write to a big object. By writing asynchronously, you are better equipped to handle database lifecycle events.

**как создать** https://developer.salesforce.com/docs/atlas.en-us.224.0.bigobjects.meta/bigobjects/big\_object\_define.htm?\_ga=2.141257781.646137914.1712662881-944784004.1696426223 **Metadata API** object files

permissionset/profile files

package file **Setup**

Big objects support these field types

Lookup Relationship

Date/Time

Email

Number

Phone

Text

Text Area (Long)

URL

To create an index for your big object, at least one custom field must be marked as required

Considerations When Creating an Index

Plan carefully when creating an index for your custom big object. The index is used for querying and filtering the big object data and must be designed properly.

Keep these considerations in mind when creating the index.

An index must include at least one custom field and can have up to five custom fields total.

All custom fields that are part of the index must be marked as required.

You can’t include Long Text Area and URL fields in the index.

The total number of characters across all text fields in an index can’t exceed 100

(Email fields are 80 characters. Phone fields are 40 characters. Keep these lengths in mind when designing your index because they count toward the 100 character limit.)  
 After you’ve created the index, you can’t edit or delete it. To change the index, you must start over with a new big object.

Design your index so that you assign the most frequently used field in a query filter to Index Position 1. The order in which you define the fields determines the order that they’re listed in the index.

**как использовать**

Bulk API and csv to fill  
 with APEX Database.insertImmediate()  
  
 deleteImmediate()  
  
 updateImmediate()  
  
 запросы через батчи по 50000 тыс в батче  
 запись через queue (предусмотреть перезапись если падает)  
 отчеты через стандартные обьекты кот заполняются из БО  
  
**SOQL ограничения https://developer.salesforce.com/docs/atlas.en-us.bigobjects.meta/bigobjects/big\_object\_querying.htm**  
 Build an index query, starting from the first field defined in the index, without gaps between the first and last field in the query. You can use = or IN on any field in your query, although you can use IN only one time. You can use the range operations <, >, <=, or >= only on the last field of your query.

When building an index query, don’t leave gaps between the first and the last field in the query.

* The operators !=, LIKE, NOT IN, EXCLUDES, and INCLUDES aren’t valid in any query.
* Aggregate functions aren’t valid in any query.
* To retrieve a list of results, don’t use the Id field in a query. Including Id in a query returns only results that have an empty ID (000000000000000 or 000000000000000AAA).

**практика**  
добавление записи  
извлечение записи