SALESFORCE DEVELOPER LIMITS AND ALLOCATIONS QUICK REFERENCE

Summary

Find the most critical limits for developing Lightning Platform applications.

About This Quick Reference

This quick reference provides common limits and allocations for Salesforce and does not cover all limits and allocations. It might contain limits or allocations that don't apply to your Salesforce org. Stated limits aren't a promise that the specified resource is available at its limit in all circumstances. Load, performance, and other system issues can prevent some limits from being reached.

This guide doesn't include limits or allocations for:

- User interface elements in the Salesforce application
- Field lengths of Salesforce objects
- Desktop integration clients
- Your Salesforce contract

Information for specific feature limits, such as the number of total and active rules in your org, are also in Salesforce Help; see the topics for using that feature. For allocations per edition, see Salesforce Features and Edition Allocations. Contractual limits might also apply, as per your Salesforce contract.

Apex Governor Limits

Read up on Apex limits details in Execution Governors and Limits

Because Apex runs in a multitenant environment, the Apex runtime engine strictly enforces limits so that runaway Apex code or processes don't monopolize shared resources.

Per-Transaction Apex Limits

These limits count for each Apex transaction. For Batch Apex, these limits are reset for each execution of a batch of records in the `execute` method.

This table lists limits for synchronous Apex and asynchronous Apex (Batch Apex and future methods) when they're different. Otherwise, this table lists only one limit that applies to both synchronous and asynchronous Apex.

Description	Synchronous Limit	Asynchronous Limit
Total number of SOQL queries issued[1]	100	200
Total number of records retrieved by SOQL queries	50,000	
Total number of records retrieved by `Database.getQueryLocator`	10,000	
Total number of SOSL queries issued	20	

Last updated: March 23, 2020

Description	Synchronous Limit	Asynchronous Limit
Total number of records retrieved by a single SOSL query	2,000	
Total number of DML statements issued[2]	150	
Total number of records processed as a result of DML statements, `Approval.process`, or `database.emptyRecycleBin`	10,000	
Total stack depth for any Apex invocation that recursively fires triggers due to `insert`, `update`, or `delete` statements[3]	16	
Total number of callouts (HTTP requests or web services calls) in a transaction	100	
Maximum cumulative timeout for all callouts (HTTP requests or Web services calls) in a transaction	120 seconds	
Maximum number of methods with the `future` annotation allowed per Apex invocation	50	0 in batch and future contexts; 1 in queueable context
Maximum number of Apex jobs added to the queue with `System.enqueueJob`	50	1
Total number of `sendEmail` methods allowed	10	
Total heap size[4]	6 MB	12 MB
Maximum CPU time on the Salesforce servers[5]	10,000 milliseconds	60,000 milliseconds
Maximum execution time for each Apex transaction	10 minutes	
Maximum number of push notification method calls allowed per Apex transaction	10	
Maximum number of push notifications that can be sent in each push notification method call	2,000	

[1] In a SOQL query with parent-child relationship subqueries, each parent-child relationship counts as an extra query. These types of queries have a limit of three times the number for top-level queries. The limit for subqueries corresponds to the value that `Limits.getLimitAggregateQueries()` returns. The row counts from these relationship queries contribute to the row counts of the overall code execution. This limit doesn't apply to custom metadata types. In a single Apex transaction, custom metadata records can have unlimited SOQL queries. In addition to static SOQL statements, calls to the following methods count against the number of SOQL statements issued in a request.

- `Database.countQuery`
- `Database.getQueryLocator`
- `Database.query`

[2] Calls to the following methods count against the number of DML statements issued in a request.

- `Approval.process`
- `Database.convertLead`
- `Database.emptyRecycleBin`
- `Database.rollback`
- `Database.setSavePoint`
- `delete` and `Database.delete`
- `insert` and `Database.insert`
- `merge` and `Database.merge`
- `undelete` and `Database.undelete`
- `update` and `Database.update`
- `upsert` and `Database.upsert`
- `EventBus.publish`
- `System.runAs`

[3] Recursive Apex that does not fire any triggers with `insert`, `update`, or `delete` statements, exists in a single invocation, with a single stack. Conversely, recursive Apex that fires a trigger spawns the trigger in a new Apex invocation. The new invocation is separate from the invocation of the code that caused it to fire. Spawning a new invocation of Apex is a more expensive operation than a recursive call in a single invocation. Therefore, there are tighter restrictions on the stack depth of these types of recursive calls.

[4] Email services heap size is 36 MB.

[5] CPU time is calculated for all executions on the Salesforce application servers occurring in one Apex transaction. CPU time is calculated for the executing Apex code, and for any processes that are called from this code, such as package code and workflows. CPU time is private for a transaction and is isolated from other transactions. Operations that don't consume application server CPU time aren't counted toward CPU time. For example, the portion of execution time spent in the database for DML, SOQL, and SOSL isn't counted, nor is waiting time for Apex callouts.

Note:

- Limits apply individually to each `testMethod`.
- To determine the code execution limits for your code while it is running, use the Limits methods. For example, you can use the `getDMLStatements` method to determine the number of DML statements that have already been called by your program. Or, you can use the `getLimitDMLStatements` method to determine the total number of DML statements available to your code.

Per-Transaction Certified Managed Package Limits

Certified managed packages—managed packages that have passed the security review for AppExchange—get their own set of limits for most per-transaction limits. Salesforce ISV Partners develop certified managed packages which are installed in your org from AppExchange and have unique namespaces.

There's no limit on the number of certified namespaces that can be invoked in a single transaction. However, the number of operations that can be performed in each namespace must not exceed the per-transaction limits. There's also a limit on the cumulative number of operations that can be made across

namespaces in a transaction. This cumulative limit is 11 times the per-namespace limit. For example, if the per-namespace limit for SOQL queries is 100, a single transaction can perform up to 1,100 SOQL queries. In this case, the cumulative limit is 11 times the per-namespace limit of 100. These queries can be performed across an unlimited number of namespaces, as long as any one namespace doesn't have more than 100 queries. The cumulative limit doesn't affect limits that are shared across all namespaces, such as the limit on maximum CPU time.

Note: These cross-namespace limits apply only to namespaces in certified managed packages. Namespaces in packages that are not certified don't have their own separate governor limits. The resources they use continue to count against the same governor limits used by your org's custom code.

This table lists the cumulative cross-namespace limits.

Description	Cumulative Cross-Namespace Limit
Total number of SOQL queries issued	1,100
Total number of records retrieved by `Database.getQueryLocator`	110,000
Total number of SOSL queries issued	220
Total number of DML statements issued	1,650
Total number of callouts (HTTP requests or web services calls) in a transaction	1,100
Total number of `sendEmail` methods allowed	110

Lightning Platform Apex Limits

The limits in this table aren't specific to an Apex transaction; Lightning Platform enforces these limits.

Description	Limit
The maximum number of asynchronous Apex method executions (batch Apex, future methods, Queueable Apex, and scheduled Apex) per a 24-hour period[1]	250,000 or the number of user licenses in your org multiplied by 200, whichever is greater
Number of synchronous concurrent transactions for long-running transactions that last longer than 5 seconds for each org.[2]	10
Maximum number of Apex classes scheduled concurrently	100. In Developer Edition orgs, the limit is 5.
Maximum number of batch Apex jobs in the Apex flex queue that are in `Holding` status	100
Maximum number of batch Apex jobs queued or active concurrently[3]	5
Maximum number of batch Apex job `start` method concurrent executions[4]	1
Maximum number of batch jobs that can be submitted in a running test	5

Description	Limit
Maximum number of test classes that can be queued per 24-hour period (production orgs other than Developer Edition)[5]	The greater of 500 or 10 multiplied by the number of test classes in the org
Maximum number of test classes that can be queued per 24-hour period (sandbox and Developer Edition orgs)[5]	The greater of 500 or 20 multiplied by the number of test classes in the org
Maximum number of query cursors open concurrently per user[6]	50
Maximum number of query cursors open concurrently per user for the Batch Apex `start` method	15
Maximum number of query cursors open concurrently per user for the Batch Apex `execute` and `finish` methods	5

[1] For Batch Apex, method executions include executions of the `start`, `execute`, and `finish` methods. This limit is for your entire org and is shared with all asynchronous Apex: Batch Apex, Queueable Apex, scheduled Apex, and future methods. To check how many asynchronous Apex executions are available, make a request to REST API `limits` resource. See List Organization Limits in the *REST API Developer Guide*. The licenses that count toward this limit are full Salesforce user licenses or App Subscription user licenses. Chatter Free, Chatter customer users, Customer Portal User, and partner portal User licenses aren't included.

[2] If more transactions are started while the 10 long-running transactions are still running, they're denied. HTTP callout processing time is not included when calculating this limit.

[3] When batch jobs are submitted, they're held in the flex queue before the system queues them for processing.

[4] Batch jobs that haven't started yet remain in the queue until they're started. If more than one job is running, this limit doesn't cause any batch job to fail.`execute` methods of batch Apex jobs still run in parallel.

[5] This limit applies to tests running asynchronously. This group of tests includes tests started through the Salesforce user interface including the Developer Console or by inserting `ApexTestQueueItem` objects using SOAP API.

[6] For example, assume that 50 cursors are open. If a client application, logged in as the same user, attempts to open a new one, the oldest of the 50 cursors is released. Cursor limits for different Lightning Platform features are tracked separately. For example, you can have all these cursors open concurrently: 50 Apex query cursors, 15 for the Batch Apex `start` method, 5 each for the Batch Apex `execute` and `finish` methods, and 5 Visualforce cursors.

Static Apex Limits

Description	Limit
Default timeout of callouts (HTTP requests or Web services calls) in a transaction	10 seconds
Maximum size of callout request or response (HTTP request or Web services call)[1]	6 MB for synchronous Apex or 12 MB for asynchronous Apex
Maximum SOQL query run time before Salesforce cancels the transaction	120 seconds
Maximum number of class and trigger code units in a deployment of Apex	5,000
Apex trigger batch size[2]	200
For loop list batch size	200
Maximum number of records returned for a Batch Apex query in `Database.QueryLocator`	50 million

[1] The HTTP request and response sizes count towards the total heap size.

[2] The Apex trigger batch size for platform events and Change Data Capture events is 2,000.

Size-Specific Apex Limits

Description	Limit
Maximum number of characters for a class	1 million
Maximum number of characters for a trigger	1 million
Maximum amount of code used by all Apex code in an org[1]	6 MB
Method size limit [2]	65,535 bytecode instructions in compiled form

[1] This limit does not apply to Apex code in first generation(1GP) or second generation(2GP) managed packages. The code in those types of packages belongs to a namespace unique from the code in your org. This limit also does not apply to any code included in a class defined with the `@isTest` annotation.

[2] Large methods that exceed the allowed limit cause an exception to be thrown during the execution of your code.

Push Notification Limits

The maximum push notifications allowed for each mobile app associated with your Salesforce org depends on the type of app.

Mobile application type	Maximum notifications per app per day
Provided by Salesforce (for example, Salesforce for iOS)	50,000
Developed by your company for internal employee use	35,000
Installed from the AppExchange	5,000

Only *deliverable* notifications count toward this limit. For example, consider the scenario where a notification is sent to 1,000 employees in your company, but 100 employees haven't installed the mobile application yet. Only the notifications sent to the 900 employees who have installed the mobile application count toward this limit.

Each test push notification that is generated through the Test Push Notification page is limited to a single recipient. Test push notifications count toward an application's daily push notification limit.

API Request Limits and Allocations

To maintain optimum performance and ensure that the Lightning Platform API is available to all our customers, Salesforce balances transaction loads by imposing two types of limits:

- Concurrent API Request Limits
- Total API Request Allocations

When a call exceeds a request limit, an error is returned.

Concurrent API Request Limits

The following table lists the limits for various types of orgs for concurrent requests (calls) with a duration of 20 seconds or longer.

Org Type	Limit
Developer Edition and Trial orgs	5
Production orgs and Sandboxes	25

Total API Request Allocations

The following table lists the limits for the total API requests (calls) per 24-hour period for an org.

Salesforce Edition	API Calls Per License Type Per 24-Hour Period	Total Calls Per 24-Hour Period
Developer Edition	N/A	15,000

Salesforce Edition	API Calls Per License Type Per 24-Hour Period	Total Calls Per 24-Hour Period
• Enterprise Edition • Professional Edition with API access enabled	• Salesforce: 1,000 • Salesforce Platform: 1,000 • Lightning Platform - One App: 200 • Customer Community: 0 • Customer Community Login: 0 • Customer Community Plus: 200 • Customer Community Plus Login: 10 • External Identity 25,000 SKU: 70,000 • External Identity 250,000 SKU, 750,000 • External Identity 1,000,000 SKU: 4,000,000 • Partner Community: 200 • Partner Community Login: 10 • Lightning Platform Starter: 200 per member for Enterprise Edition orgs • Lightning Platform Plus: 1000 per member for Enterprise Edition orgs	100,000 + (number of licenses x calls per license type) + purchased API Call Add-Ons
• Unlimited Edition • Performance Edition	• Salesforce: 5,000 • Salesforce Platform: 5,000 • Lightning Platform - One App: 200 • Customer Community: 0 • Customer Community Login: 0 • Customer Community Plus: 200 • Customer Community Plus Login: 10 • External Identity 25,000 SKU: 70,000 • External Identity 250,000 SKU, 750,000	100,000 + (number of licenses x calls per license type) + purchased API Call Add-Ons

Salesforce Edition	API Calls Per License Type Per 24-Hour Period	Total Calls Per 24-Hour Period
	• External Identity 1,000,000 SKU: 4,000,000 • Partner Community: 200 • Partner Community Login: 10 • Lightning Platform Starter: 200 per member for Unlimited and Performance Edition orgs • Lightning Platform Plus: 5,000 per member for Unlimited and Performance Edition orgs	
Sandbox	N/A	5,000,000

Note:

- At high volume for a single org, other limiting factors such as system load can affect the total number of calls that are processed in a 24-hour period.

APIs that count toward this allocation include the Lightning Platform REST API, the Lightning Platform SOAP API, Bulk API, and Bulk API 2.0. API calls issued by certain Salesforce connected apps (for example, the Salesforce mobile app) don't count. To determine which APIs affect the allocation, see Monitoring Your API Usage.

The DebuggingHeader has an extra usage amount of 1,000 calls per 24-hour period.

Limits and allocations are enforced against the aggregate of all API calls made to the org in a 24-hour period. Limits and allocations are not on a per-user basis.

Monitoring Your API Usage

To better monitor your org's API usage and limits, you can use these resources:

- The API Usage section of the System Overview page in Setup.
- The API Requests, Last 24 Hours item in the Organization Detail section of the System Overview page in Setup.
- The API Request Limit per Month usage-based entitlement, which shows you your org's API calls aggregated over 30 days. This can be found on the Company Information page in Setup.
- Information returned in the `Sforce-Limit-Info` response header for REST APIs.
- Information returned in the response body (in `<type>API REQUESTS</type>`) for SOAP APIs.
- The `/limits` call in the Lightning Platform REST API.

You can configure your org so that email is sent to a designated user when the number of API requests has exceeded a specified percentage of the amount allotted. Perform this configuration from Setup by entering *`API Usage Notifications`* in the Quick Find box and then selecting **API Usage Notifications**.

See also the Learn About Daily Rate Limits section in the App Development Without Limits Trailhead module.

What Happens If You Reach or Exceed Your API Request Limit

If your org reaches or exceeds its daily API request limit, Salesforce still allows the operations to proceed by a certain amount, if possible. This helps avoid blocking your workflows during unexpected spikes in workloads and occasional peak periods. A hard cap is in place to safeguard platform resources and prevent API requests from exceeding the daily limit unimpeded.

Note: The ability to go over your normal daily limit is always subject to restrictions to protect the overall health of the Salesforce instance that hosts your org. (You can monitor the health of your instance on Salesforce Trust.)

This ability is designed to be used occasionally to help avoid interruptions in your workflow. Don't rely on it on an ongoing basis. To increase your allocation, contact your Salesforce account representative.

This ability only applies to paid orgs in active status. It does not apply to trial orgs, Developer Edition, or sandboxes.

API request activity is aggregated into 30 day periods, starting with your contract start date, and includes calls that exceed the org's entitled limit.

Increasing Total API Request Allocations

The calculation of the API request amounts based on user licenses is designed to allow sufficient capacity for your org based on your number of users. If you need a higher amount and you don't want to purchase extra user licenses or upgrade to Performance Edition, you can purchase extra API calls. For information, contact your account representative.

Before you purchase more API calls, perform due diligence of your API usage. You can optimize a client application, whether it's your own enterprise application or partner application, to use fewer API calls and still accomplish the same work. If you use a partner product, consult with the vendor to verify that the product makes optimal use of the API. A product that makes inefficient use of the API incurs unnecessary costs for your company. Use REST API composite resources to improve your application's performance by minimizing the number of round-trips between the client and server.

Example API Usage Metering Calculations

The following examples illustrate API usage metering calculations for several scenarios.

- For an Enterprise Edition org with 15 Salesforce licenses, the request limit is 115,000 requests (100,000 plus 15 licenses x 1,000 calls).
- For a Developer Edition org that made 14,500 calls at 5:00 AM Wednesday, 499 calls at 11:00 PM Wednesday, only one more call can successfully be made until 5:00 AM Thursday.

Length of Stored Third-Party Refresh and Access Tokens

Salesforce stores up to 2,000 characters for each third-party access token and 1,024 for each refresh token.

Chatter REST API Limits

Limits protect shared resources. These are the limits for consumers of Chatter REST API.

Chatter REST API requests are subject to rate limiting. Chatter REST API has a different rate limit than other Salesforce APIs. Chatter REST API has a per user, per application, per hour rate limit. When you exceed the rate limit, all Chatter REST API resources return a 503 Service Unavailable error code.

For applications using a session ID from Salesforce, the rate limit is per user, per hour—there isn't a separate bucket for applications. All applications the user accesses with a session ID use this general quota. To take advantage of the per user, per application, per hour limit, use OAuth tokens.

Note: Load, performance, and other system issues can prevent some limits from being reached. Limits can change without notice. Applications should make efficient use of available requests and gracefully handle the 503 error code.

Bulk API Allocations

Bulk API Allocation	Limit Description
Batch Allocation	You can submit up to 10,000 batches per rolling 24-hour period. You can't create batches associated with a job that is more than 24 hours old. If a batch is submitted to a closed job, the batch will not be created, however it will still count against the batch allocation as a submitted batch.
Batch lifespan	Batches and jobs that are older than seven days are removed from the queue regardless of job status. The seven days are measured from the youngest batch associated with a job, or the age of the job if there are no batches. You can't create batches associated with a job that is more than 24 hours old.
Batch size	• Batches for data loads can consist of a single CSV, XML, or JSON file that is no larger than 10 MB. • A batch can contain a maximum of 10,000 records. • A batch can contain a maximum of 10,000,000 characters for all the data in a batch. • A field can contain a maximum of 32,000 characters. • A record can contain a maximum of 5,000 fields. • A record can contain a maximum of 400,000 characters for all its fields. • A batch must contain some content or an error occurs.
Batch processing time	Batches are processed in chunks. The chunk size depends on the API version. In API version 20.0 and earlier, the chunk size is 100 records. In API version 21.0 and later, the chunk size is 200 records. There's a 5-minute limit for processing each chunk. Also, if it takes longer than 10 minutes to process a whole batch, the Bulk API places the remainder of the batch back in the queue for later processing. If the Bulk API

Bulk API Allocation	Limit Description
	continues to exceed the 10-minute limit on subsequent attempts, the batch is placed back in the queue and reprocessed up to 10 times before the batch is permanently marked as failed.
Binary content	• The length of any file name can't exceed 512 bytes. • A zip file can't exceed 10 MB. • The total size of the unzipped content can't exceed 20 MB. • A maximum of 1,000 files can be contained in a zip file. Directories don't count toward this total.
Bulk query number of attempts to query	15 attempts at 10 minutes each to process the batch. There is also a 2-minute limit on the time to process the query. If more than 15 attempts are made for the query, an error message of "Tried more than fifteen times" is returned. If the query takes more than 2 minutes to process, a QUERY_TIMEOUT error is returned.
Bulk query number of retrieved files	15 files. If the query needs to return more than 15 files, add filters to the query to return less data. Bulk batch sizes are not used for bulk queries.
Bulk query results lifespan	7 days.
Bulk query retrieved file size	1 GB
Job open time	The maximum time that a job can remain open is 24 hours. The Bulk API doesn't support clients that, for example, post one batch every hour for many hours.

API Query Cursor Limits

A user can have up to 10 query cursors open at a time. If 10 `QueryLocator` cursors are open when a client application, logged in as the same user, attempts to open a new one, then the oldest of the 10 cursors is released. If the client application attempts to open the released query cursor, an error results.

SOAP API Call Limits

API Name	API Limit	Limit Description
`create()`	Maximum number of records created	Your client application can add up to 200 records in a single `create()` call. If a create request exceeds 200 objects, then the entire operation fails.
`describeSObjects()`	Maximum number of objects returned	The `describeSObjects()` call is limited to a maximum of 100 objects returned.

API Name	API Limit	Limit Description
`emptyRecycleBin()`	Maximum number of records in the Recycle Bin	Your org can have up to 5,000 records per license in the Recycle Bin at any one time. For example, if your org has five user licenses, 25,000 records can be stored in the Recycle Bin. If your org reaches its Recycle Bin limit, Salesforce automatically removes the oldest records, as long as they have been in the recycle bin for at least two hours.
`getDeleted()`	Limits for returned records	• If your `getDeleted()` call returns more than 600,000 records and the user is a system administrator, an exception EXCEEDED_ID_LIMIT is returned. • If your `getDeleted()` call returns more than 20,000 records and the user is not a system administrator, an exception OPERATION_TOO_LARGE is returned. Note that this error is returned when more than 20,000 records across the organization have been deleted, not just the records viewable by the user.
`login()`	Login request size limit	The login request size is limited to 10 KB.
`merge()`	Merge request limits	• Up to 200 merge requests can be made in a single SOAP call. • Up to three records can be merged in a single request, including the master record. This limit is the same as the limit enforced by the Salesforce user interface. To merge more than 3 records, do a successive merge. • External ID fields cannot be used with `merge()`. • If you selected the option to retain the most recently updated data privacy record for merging leads and contacts, but the caller does not have CRUD permission for the selected data privacy record, the merge process selects the data privacy record already associated with the master record.
`update()`	Maximum number of records updated	Your client application can change up to 200 records in a single `update()` call. If an update request exceeds 200 records, the entire operation fails.
`query()` and `queryMore()`	Batch size limits	WSC clients can set the batch size by calling `setQueryOptions()` on the connection object. C# client applications can change this setting by specifying the batch size in the call QueryOptions portion of the SOAP header before invoking the `query()` call. The maximum batch size is 2,000 records. However this setting is only a suggestion. There is no guarantee that the requested

API Name	API Limit	Limit Description
		batch size will be the actual batch size. This is done to maximize performance. The batch size will be no more than 200 if the SOQL statement selects two or more custom fields of type long text. This is to prevent large SOAP messages from being returned.

Metadata Limits

The following limits apply to the Salesforce Extensions for Visual Studio Code, the Ant Migration Tool, and the Metadata API.

Limit	Description
Retrieving and deploying metadata	You can deploy or retrieve up to 10,000 files at once. AppExchange packages use different limits. In API version 43.0 and 44.0, AppExchange packages can contain up to 12,500 files. In API version 45.0, AppExchange packages can contain up to 17,500 files. In API version 46.0, AppExchange packages can contain up to 22,000 files. In API version 47.0 and later, AppExchange packages can contain up to 30,000 files. The maximum size of the deployed or retrieved .zip file is 39 MB. If the files are uncompressed in an unzipped folder, the size limit is 400 MB. Note the following: • If using the Ant Migration Tool to deploy an unzipped folder, all files in the folder are compressed first. The maximum size of uncompressed components in an unzipped folder is 400 MB or less depending on the compression ratio. If the files have a high compression ratio, you can migrate a total of approximately 400 MB because the compressed size would be under 39 MB. However, if the components can't be compressed much, like binary static resources, you can migrate less than 400 MB. • Metadata API base-64 encodes components after they're compressed. The resulting .zip file can't exceed 50 MB, which is the limit for SOAP messages. Base-64 encoding increases the size of the payload, so your compressed payload can't exceed approximately 39 MB before encoding. • You can perform a `retrieve()` call for a big object only if its index is defined. If a big object is created in Setup and doesn't yet have an index defined, you can't retrieve it.
Change sets	Inbound and outbound change sets can have up to 10,000 files of metadata.

SOQL and SOSL Limits

Feature	Limit	Limit Description
SOQL statements	Maximum length of SOQL statements	By default, 100,000 characters. Note: Long, complex SOQL statements, such as statements that contain many formula fields, can sometimes result in a `QUERY_TOO_COMPLICATED` error. The error occurs because the statement is expanded internally when processed by Salesforce, even though the original SOQL statement is under the 100,000 character limit. To avoid this error, reduce the complexity of your SOQL statement. Page layouts in Lightning with more than 250 fields can also cause a `QUERY_TOO_COMPLICATED` error. Lightning uses auto-generated SOQL to retrieve fields for a record page layout, so the error can occur even if there isn't any customer-written SOQL. The character limit can also be reached by including too many currency fields. Currency fields require SOQL to use a format method, roughly doubling the field API name length for each currency field.
SOQL `WHERE` clause	Strings in SOQL `WHERE` clauses	4,000 characters.
SOSL statements	Maximum length of SOSL statements	By default, 100,000 characters. This limit is tied to the SOQL statement character limit defined for your org.
SOSL search query strings	Maximum length of `SearchQuery` string	If the `SearchQuery` string is longer than 10,000 characters, no result rows are returned. If `SearchQuery` is longer than 4,000 characters, any logical operators are removed. For example, the `AND` operator in a statement with a `SearchQuery` that's 4,001 characters will default to the `OR` operator, which could return more results than expected.
SOSL query results	Maximum rows returned	2,000 results total (API version 28.0 and later), unless you specify custom limits in the query. Previous API versions return 200 results.
Relationship queries	Relationship query limits	• No more than 55 child-to-parent relationships can be specified in a query. A custom object allows up to 25 relationships, so you can reference all the

Feature	Limit	Limit Description
		child-to-parent relationships for a custom object in one query. • No more than 20 parent-to-child relationships can be specified in a query. • In each specified relationship, no more than five levels can be specified in a child-to-parent relationship. For example, `Contact.Account.Owner.FirstName` (three levels). • In each specified relationship, only one level of parent-to-child relationship can be specified in a query. For example, if the `FROM` clause specifies Account, the `SELECT` clause can specify only the Contact or other objects at that level. It could not specify a child object of Contact.
FOR VIEW and FOR REFERENCE	Maximum RecentlyViewed records allowed	The RecentlyViewed object is updated every time the logged-in user views or references a record. It is also updated when records are retrieved using the `FOR VIEW` or `FOR REFERENCE` clause in a SOQL query. To ensure that the most recent data is available, RecentlyViewed data is periodically truncated down to 200 records per object. RecentlyViewed data is retained for 90 days, after which it is removed on a periodic basis.
OFFSET clause	Maximum number of rows skipped by OFFSET	The maximum offset is 2,000 rows. Requesting an offset greater than 2,000 results in a `NUMBER_OUTSIDE_VALID_RANGE` error.

Visualforce Limits

Limit	Value
Maximum response size for a Visualforce page	Less than 15 MB
Maximum view state size in a Visualforce page	170KB
Maximum size of a Visualforce email template	1 MB
Maximum file size for a file uploaded using a Visualforce page	10 MB
Maximum size of HTML response *before* rendering, when Visualforce page is rendered as PDF	Less than 15 MB
Maximum PDF file size for a Visualforce page rendered as a PDF	60 MB
Maximum total size of all images included in a Visualforce page rendered as a PDF	30 MB

Limit	Value
Maximum header size of a Visualforce page	8,192 bytes
Maximum request size of a JavaScript remoting call	4 MB
Maximum response size of a JavaScript remoting call	15 MB
Default timeout for a JavaScript remoting call	30,000 milliseconds (30 seconds)
Maximum timeout for a JavaScript remoting call	120,000 milliseconds (120 seconds)
Maximum rows retrieved by queries for a single Visualforce page request	50,000
Maximum rows retrieved by queries for a single Visualforce page request in read-only mode	1,000,000
Maximum collection items that can be iterated in an iteration component such as `<apex:pageBlockTable>` and `<apex:repeat>`	1,000
Maximum collection items that can be iterated in an iteration component such as `<apex:pageBlockTable>` and `<apex:repeat>` in read-only mode	10,000
Maximum field sets that can be displayed on a single Visualforce page.	50
Maximum field sets allowed per sObject.	500
Maximum fields through lookup relationships allowed per field set.	25
Maximum records that can be handled by StandardSetController	10,000