Large Data Volumes

Module: Large Data Volumes

Unit: Design Your Data Model

Source: https://trailhead.salesforce.com/content/learn/modules/large-data-volumes/design-your-data-model

Plan Your Data Model

Large Data Volumes (LDV) without planning can lead to

- sluggish performance
- slower queries
- slower search and list views
- slower sandbox refreshing

The Scoop on Data Skew

Data skew happens when more that 10,000 child records are associated with the same parent recordThere are three types of data skew

1. **Account Date Skew**
    - Objects like Accounts and Opportunities have special data relationships that maintain parent and child record access under private sharing models.
    - Too many child records associated with the parent record causes account data skew
    - **Record Locking**
        - Updating a large number of contacts under the same account in multiple treads
        - Each update locks the both the Account record and the Contact record
        - High rick of failing because previous update still has the lock on the Account record
    - **Sharing Issues**
        - Time out when updating owner of Account record, because each child record need to be examined to see if an update is required from them too

2. **Ownership Skew :** Large number of records of the same object are owned by a single user cause performance issues due to sharing calculations required to manage visibility

- When the skewed owner is in a role hierarchy, operations like deletes and owner updates must remove owner and all parents within the role hierarchy.
- Best Practice: Make sure user does not have a role\

3. **Lookup Skew:** large number of records associated with single record in a lookup.

- Every time a record is inserted or updates, Salesforce must lock the target records that are selected for each lookup. This ensures that when the data is committed, it's integrity is maintained.
- Custom code and LDV in an automated process may result if lock exceptions that cause insert or update failures

Using External Objects

- Data-tiring
  - Spread data across multiple objects
  - External Objects don't have sharing rules

External Object Lookups

- **Standard Lookup**
  - Child Object: Standard, Custom, External
  - Parent Object: Standard , Custom
  - ID: Salesforce 18 character ID
- **External Lookup**
  - Child: Standard, Custom, External
  - Parent: External
  - Matching Field: External ID field
- **Indirect Lookup**
  - Child: External

- Parent: Standard, Custom
- Matching Field: custom field with External ID and Unique attributes

Unit: Conduct Data Queries and Searches

Soure: https://trailhead.salesforce.com/content/learn/modules/large-data-volumes/conduct-data-queries-and-searches

Search Architecture

- Storing huge amounts of data can affect search performance
- Most text fields are automatically indexed
- Indexed Search
  1. Searching indexes for appropriate records
  2. Narrow down results by access permissions, search limits and filters
  3. Create result set to a predetermined size. Discard the rest of the results
  4. Use result set to query the database and retrieve fields that the use sees

SOQL vs SOSL Queries

- SOQL
  - query parent-to-child relationships
  - governor limit: 50,000 queries
- SOSL
  - full text search
  - governor limits: 2,000 queries

The Force.com Query Optimizer

- maintains a table of statistics about he distribution of data in each index
- uses this table to perform pre-queries to determine whether using indexing will speed up the query

- works with auto generated queries to handle reports, list views and queries that piggyback on them

Batch Apex

- Best Practice: query and process large data sets asynchronously in batches
- Batch Apex can query and process up to 50 million records

Bulk Queries

- Retrieve up to 15GB of data, divided into fifteen 1GB files
- Bulk API supports query and queryAll (include merged/deleted records, archived Tasks and Events)
- Content-Type header request must be
  - text/csv
  - application/xml
  - application/json

How Bulk Queries are Processed

- Two minute time limit to execute a bulk query – else QUERY_TIMEOUT error returned. Rewrite a simpler query are resubmit
- If the results of a successful query exceed 1GB or 10 minutes to retrieve, the results are cached and another attempt is made. After 15 attempts, the job fails with a Retried more than 15 times .
- Use PK Chunking header to split query. Successful results are stored for seven days

Using Skinny Tables

- A skinny table is a custom table that contains a subset of fields of a standard or custom object
- Skinny Tables don’t include soft deleted records (isDeleted == true)
- Useful when table contains millions of records
- Can be created on Custom Objects, Account, Contact, Opportunity , Lead and Case Objects
- Enhances performance on Reports, List Views and SOQL
- Contact Salesforce Customer Support to enable

- Side Effects of Skinny tables
  - Must contact Salesforce Customer Support to recreate Skinny Table if the required fields change
  - not synced to Sandbox
- Do not have dynamic metadata flexibility. Any changes to the source object requires Salesforce Customer Support to recreate the table

Unit: Load Your Data

Source: https://trailhead.salesforce.com/content/learn/modules/large-data-volumes/load-your-data

Loading Lean

- Identifying business-critical operations before moving users to Salesforce
- Identifying the minimal data set and configurations required to implement those operations
- Defining a data and configuration strategy based on the requirements
- Loading data as quickly as possible to reduce the scope of synchronization

- **Organization-wide sharing defaults**
  - Private sharing model: System calculates sharing as records are being added
  - Public Read/Write: can defer this process until cutover
- **Complex object relationships**
  - More lookups defined, more checks have to take place when adding records.
  - Establish lookup relationships later to make loading faster
- **Sharing Rules**
  - Ownership based sharing rules and criteria based sharing rules will be calculated as records are loaded
- **Workflow Rules, Validation Rules, and Triggers**
  - Can slow down processing if they are enables during loading

But Not Too Lean

- **Parent records with master-detail children**
  - Parent records must be loaded first
- **Record Owners**
- **Role Hierarchy**
  - Deferring role hierarchy membership won’t increase performance

Bulk API vs SOAP API Data Loading

- **SOAP API**
  - Optimized for real-time client applications updating a few records at a time
- **Bulk API**
  - optimized for processing of loading and deleting large datasets

How Bulk API Works

- Records are streamed to Force.com
- temporary stored, the sliced into user-defined batches (max: 10,000) for processig
- Batches can be processing as data is still streaming
- Batches can be processed in parallel or serially depending on needs
- Time outs and fails are automatically reprocessed
- Batches are processed independently and can be monitored in the Admin UI

Increase Speed by Suspending Events

Workflow rules, validation rules and triggers will slow import of data. Good data prep can allow you to turn off these rules to speed up loading

1. **Analyzing and Preparing Data**
   - Validate the data before attempting to load
   - Identify what data can be deferred to post processing after the data has loaded
2. **Disabling Events for Loading**

- Set rules to inactive
- Trigger – use Custom Settings with checkbox to control when a trigger is fired

```
trigger setDefaultValues on Account ( before insert, before update ) {
        Load_Settings__c s = Load_Settings__c.getInstance( UserInfo.getUserID() ) ;
        if ( s.Load_Lean__c ) return; //skip trigger
        for( Account oAccount : trigger.new ) {
              //rest of trigger
        }
}
```

3. **Post Processing**

- Complete data enrichment
    - Add lookup relationships
    - Enhance record with foreign keys
    - Rest fields to turn triggers back on
    - Turn validation, workflow and assignment rules back on

Unit: Perform Data Deletes and Extracts

Source: https://trailhead.salesforce.com/content/learn/modules/large-data-volumes/perform-data-deletes-and-extracts

Use Bulk API for deleting large about of records. For more than 1 million, use hard delete bulk api option

Soft vs Hard Deletion

- **Soft Delete**
    - isDeleted flag is set to true
    - record is still in the database
    - data stays in the recycle bin for 15 days, or until the recycle bin reaches a specific side

- **Hard Delete**
  - Bulk API option
  - deleted records bypass recycle bin
  - hard delete option must be enables by administrator

Chunking Data

*

- When extracting with the Bulk API, by default, queries are split into 100,000 record chunks
- Use chunkSize header to configure smaller or larger size (max: 250,000)
- Need to experiment to find optimal chunk size
- For hundreds of millions of records, use PK Chunking

Using PK Chunking

If attribute filtering doesn't help break data into small enough sizes, us Primary Key Chunking
PK chunking splits bulk queries on very large tables into chunks based on the records ID (which is always indexed)

- Enable on tables with more than 10 million records or when bulk query consistently times out
- Enables on Custom Objects and Account. Campaign, Campaign Manager, Case, Contact, Lead, Login History, Opportunity, Task and User
- header: **Sforce-Enable-PKChunking**
- configure size: **Sforce-Enable-PKChunking:chunkSize=50000**
- Each chunk is processed as a seperate batch and counts towards your daily batch limit

When a query is successfully chunked, the original batch's status shows as **NOT_PROCESSED**. If the chunking fails, the original batch's status shows as **FAILED**, but any chunked batches that were successfully queued during the chunking attempt are processed as normal. When the original batch's status is changed to **NOT_PROCESSED**, monitor the subsequent batches. You can retrieve the results from each subsequent batch after it's completed. Then you can safely close the job.

Truncation

Truncation removes

- All records of a custom object currently in Recycle bin
- Custom Objects history
- related events, tasks, notes and attachments