Search Optimization

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Lab Exercise slides reference the hands-on lab exercise guide
- Do not distribute

Course Goals

- Identify methods for optimizing searches
- Explain search scheduler precedence
- Define the three types of accelerations available in Splunk
- Use the `tstats` and `datamodel` commands to query data

Course Outline

- Optimize Search
- Report Acceleration
- Data Model Acceleration
- Use the `tstats` Command

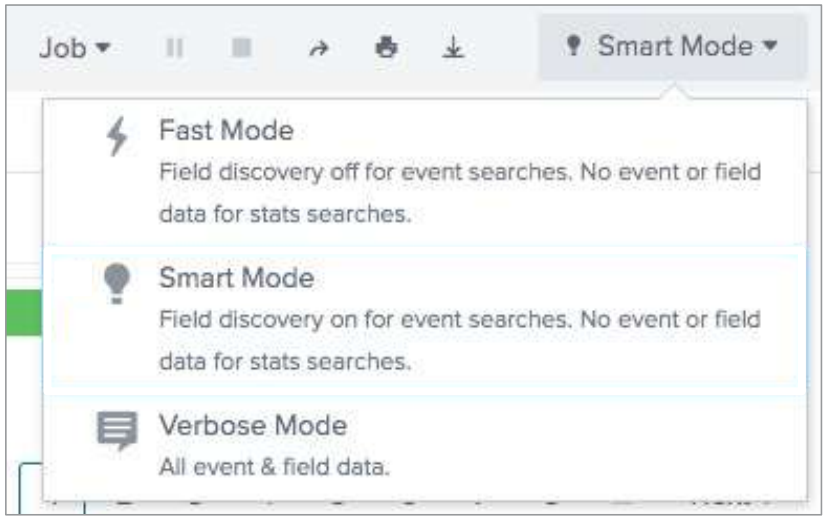
Optimize Search

Topic Objectives

- Understand how search modes affect performance
- Examine the role of the Splunk Search Scheduler
- Review general search practices

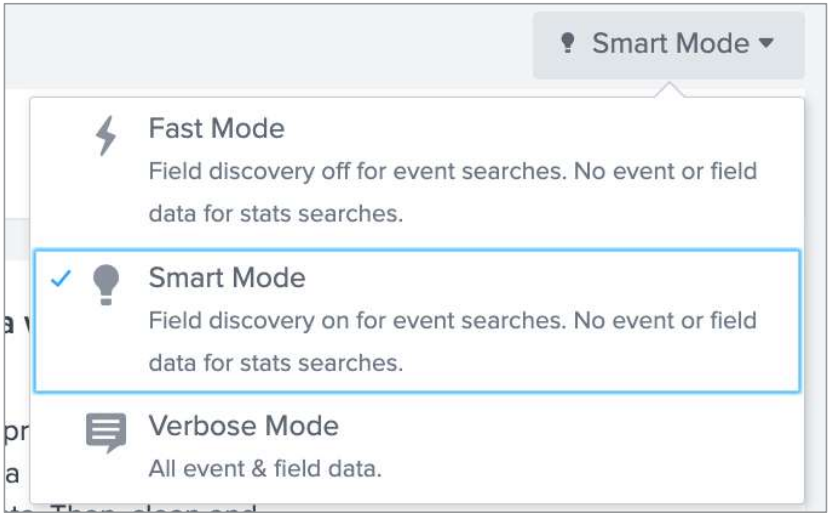
Search Modes and Search Performance

- The Splunk user interface provides 3 search modes
- Search modes determine how much field data is returned and as a result, affects how fast the search completes



Search Modes

- Smart Mode
  - Default search mode
  - Balances speed and completeness
- Fast Mode
  - Prioritizes speed over completeness
  - Disables Field Discovery
- Verbose Mode
  - Prioritizes completeness over speed
  - Returns all extracted fields



General Search Practices

- As events are stored by time, `_time` is the most efficient filter
- After time, most powerful fields to filter on: `index`, `host`, `source`, and `sourcetype`
- Create efficient searches by including specific search terms in the basic search (i.e. search criteria)

```
sourcetype=access_combined failure
```
More specific & completes faster

```
failure
```
Less specific & less efficient

- Inclusion is generally better than exclusion

```
"access denied"
```
Inclusion

```
NOT "access granted"
```
Exclusion

Splunk Search Scheduler

- Manages scheduled reports and alerts (i.e. scheduled searches)
- Prioritizes searches if too many are scheduled to run concurrently
  - Concurrent search limit is determined by system configuration
  - Searches can be skipped

Priority	Type	Examples
First	Ad hoc historical searches	Searches run manually
Second	Manually scheduled searches with real-time scheduling	User-saved scheduled reports and alerts that use real-time scheduling
Third	Manually scheduled searches with continuous scheduling	User-saved scheduled reports and alerts that use continuous scheduling
Last	Automatically scheduled searches	The searches behind report acceleration and data model acceleration; "auto-summarization" reports

Note

"Real-time scheduling" has nothing to do with whether a search "runs in real time."

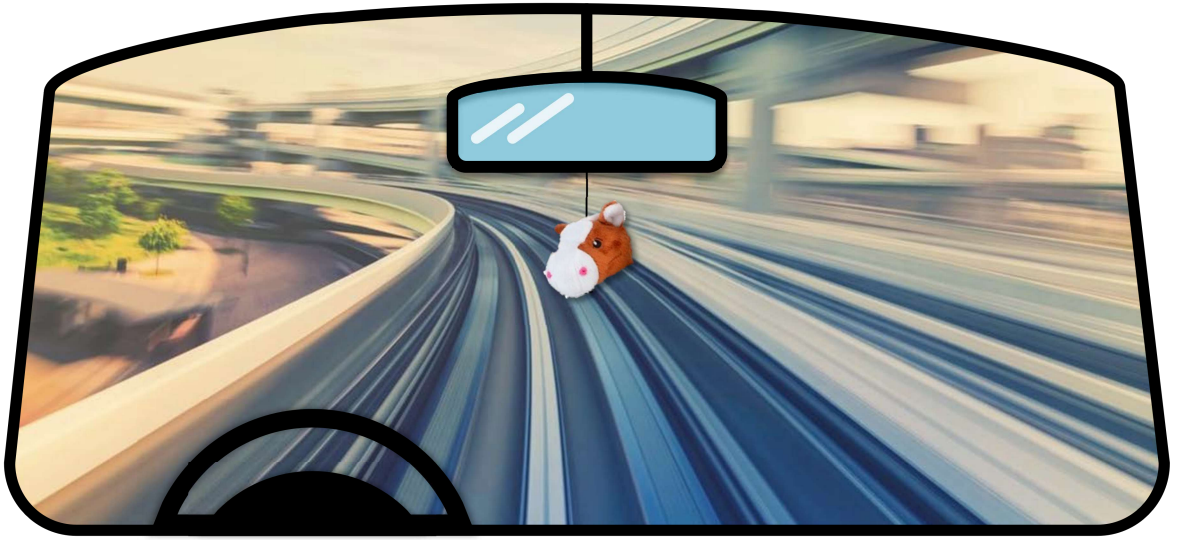
Report Acceleration

Topic Objectives

- Define acceleration and acceleration types
- Understand report acceleration and create an accelerated report
- Reveal when and how report acceleration summaries are created
- Search against acceleration summaries

What is Acceleration?

- A Splunk feature that relies on **summaries of event data to speed up search performance**
- There are 3 acceleration methods:
  - Report acceleration
  - Summary Indexing
  - Data model acceleration



Note

Data model acceleration is the easiest and most efficient acceleration option and should be your first choice.

How Summaries Make Searches Efficient

- Searches run against summaries should complete much faster because:
  - Summaries are considerably smaller than the original data set from which they are generated
  - Summaries contain only the data needed to fulfill the searches run against them
- Summaries can be automatically or manually created
  - Determined by what searches are being accelerated and which acceleration method is chosen

"Acceleration" vs "Summary"

- The terms "acceleration" and "summary" are not interchangeable
- Report acceleration, data model acceleration, and summary indexing are all **acceleration** methods that rely on **summaries**
- The differences in these methods are:
  - How they are made
  - How they are maintained
  - How they are used
- These differences are discussed in more detail in the next topics

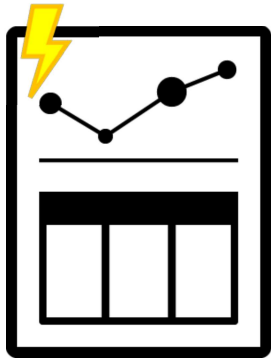
Report Acceleration

- Reports that span a large volume of data can:
  - Take a long time to complete
  - Consume a lot of system resources
- Accelerated reports run off **acceleration summaries** which:
  - Store only the data needed to fulfill the report
  - Are automatically populated in the background

Accelerated reports run faster because they are running off curated, updated data

Report Acceleration: How to Qualify

- Reports must meet certain guidelines to be eligible for report acceleration:
  - Users must have the **schedule_search** privilege and **accelerate_search** capabilities (power and admin users have this by default)
  - Search mode should be set to either 💡**Smart** or ⚡**Fast**
  - Search must include a transforming command



Report Acceleration: Commands

An accelerated report must include transforming commands and may include streaming and non-streaming commands:

Must be included

Transforming Commands

- Order results into a data table

```
...| stats
...| timechart
...| top
```

May be included, however order is important

Streaming commands

- Operate on each event as it is returned by search

```
...| eval
...| search
...| fields
...| rename
```

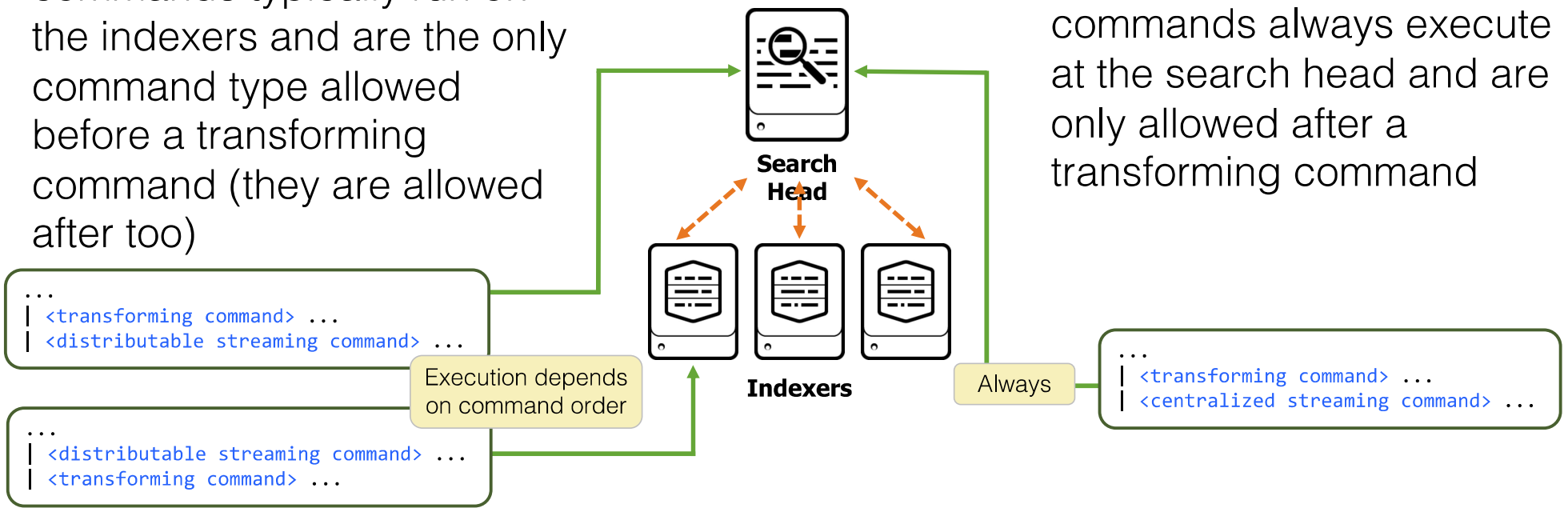
Non-streaming commands

- Execute after all events are returned

```
...| table
...| sort
...| fillnull
```

Report Acceleration: Streaming Commands

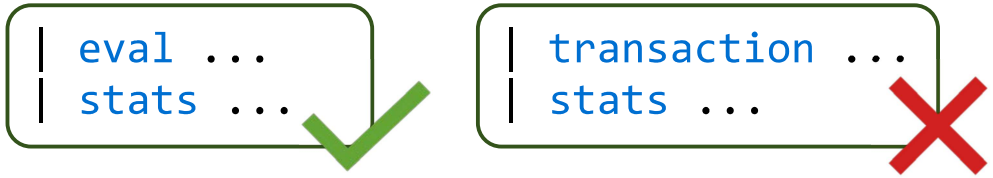
Distributable streaming commands typically run on the indexers and are the only command type allowed before a transforming command (they are allowed after too)



Centralized streaming commands always execute at the search head and are only allowed after a transforming command

Report Acceleration: To Summarize

If there are any commands that come **before** the transforming command, they *must* be distributable streaming commands



If there are any commands that come **after** the transforming command, they can be streaming (distributable or centralized) or non-streaming commands

What Reports Qualify for Acceleration?

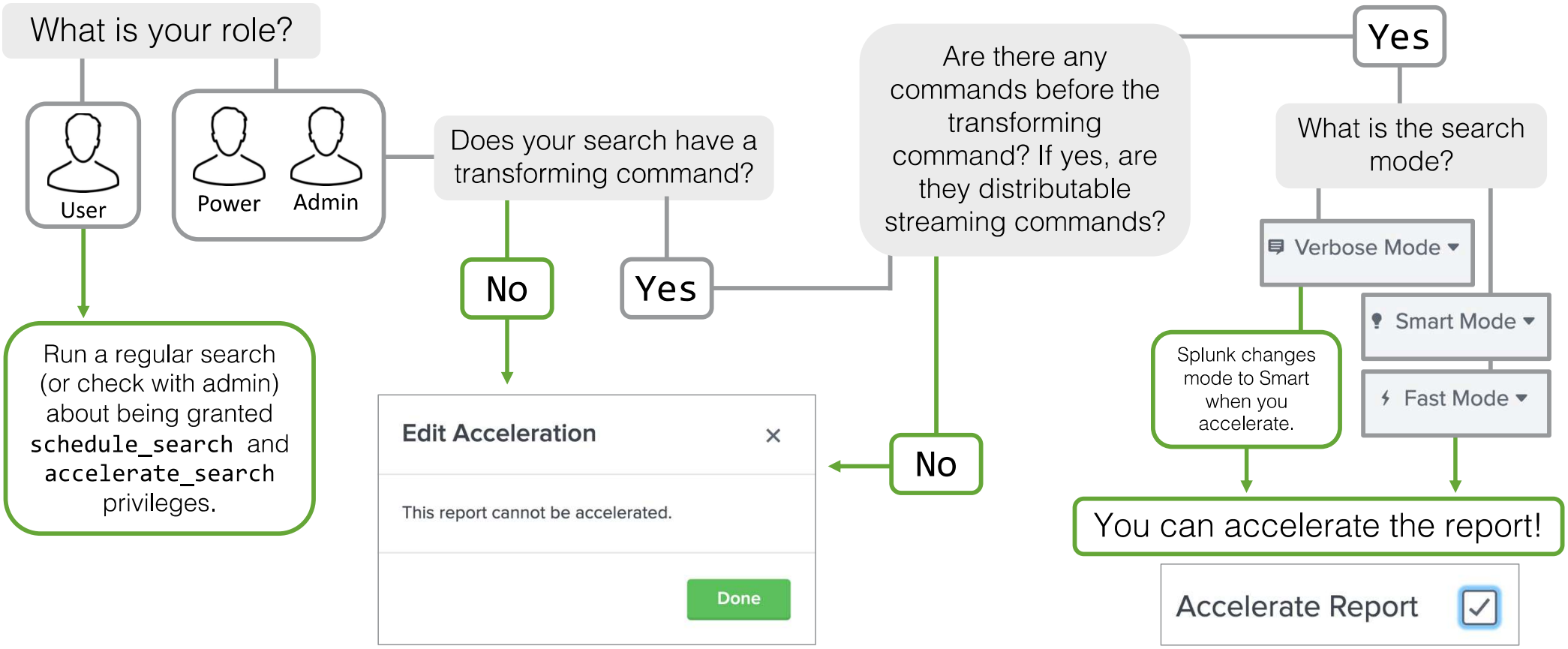
```
index=web sourcetype=access_combined action=purchase status=200
| stats sum(price) as revenue by productId
| eval revenue = "$".revenue
```

```
index=web sourcetype=access_combined action=purchase status=404
```

```
index=web sourcetype=access*
| fields price action host
| chart sum(price) over action by host
```

```
index=web sourcetype=access_combined
| transaction clientip startswith="view" endswith="purchase"
| stats avg(duration) as avgDuration
```

Report Acceleration Flowchart

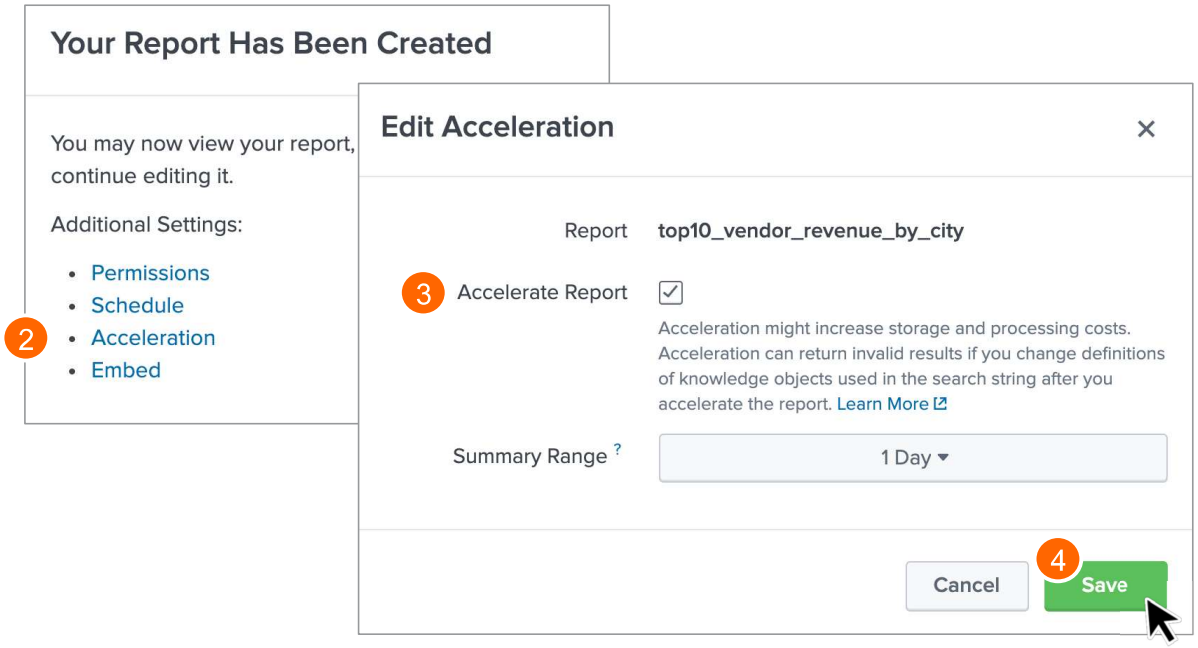


Accelerating a Report

1. Create a qualifying search and save as a report
2. After report saved, click **Acceleration**
3. Check the box next to **Accelerate Report** and choose the **Summary Range**
4. Click **Save**

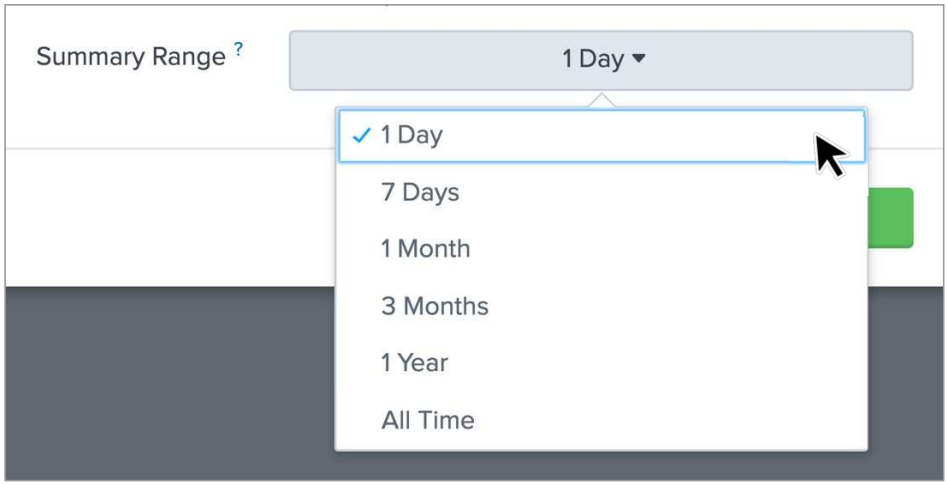
1

```
index=sales sourcetype=vendor_sales
| stats values(Vendor) as Vendor sum(price) as
  revenue by VendorCity
| sort 10 -revenue
```



Report Acceleration: Summary Range

- Determines how much time the acceleration summary spans relative to now
- Searches within the time range only use summary data
- Splunk automatically removes older summary data that ages out of range



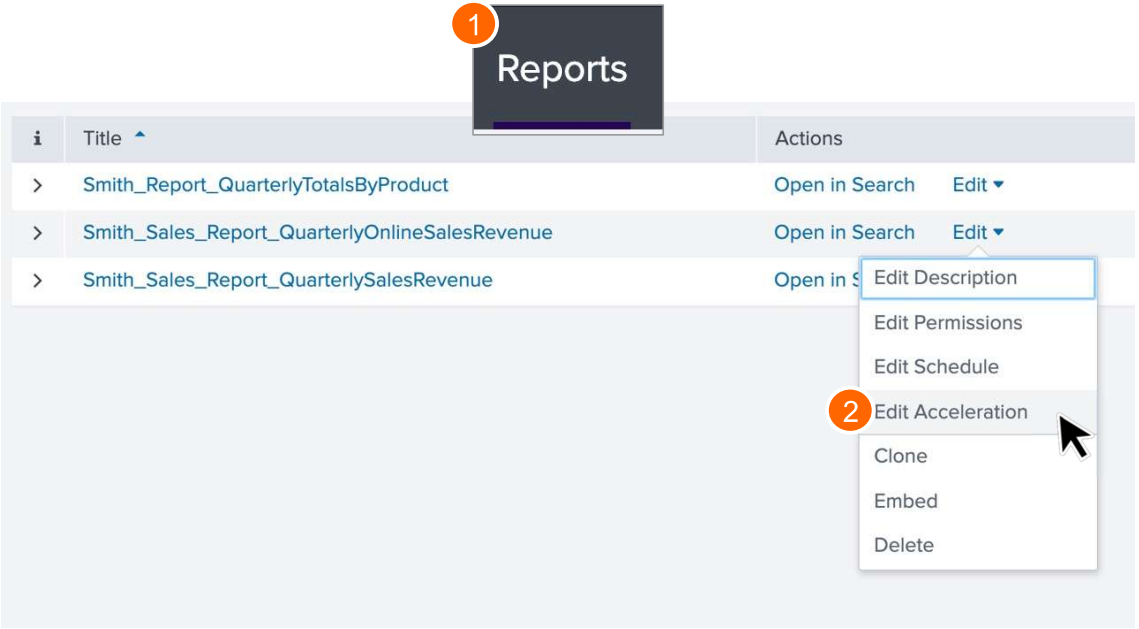
Note

Report acceleration features automatic backfill. If for some reason you have a data interruption, Splunk software can detect this and automatically update or rebuild your summaries as appropriate.

Accelerating a Previously Saved Report

A previously saved report can be accelerated too

1. Click on **Reports** in the app navigation bar and select a saved report
2. **Edit** > **Edit Acceleration** and enable the qualifying report

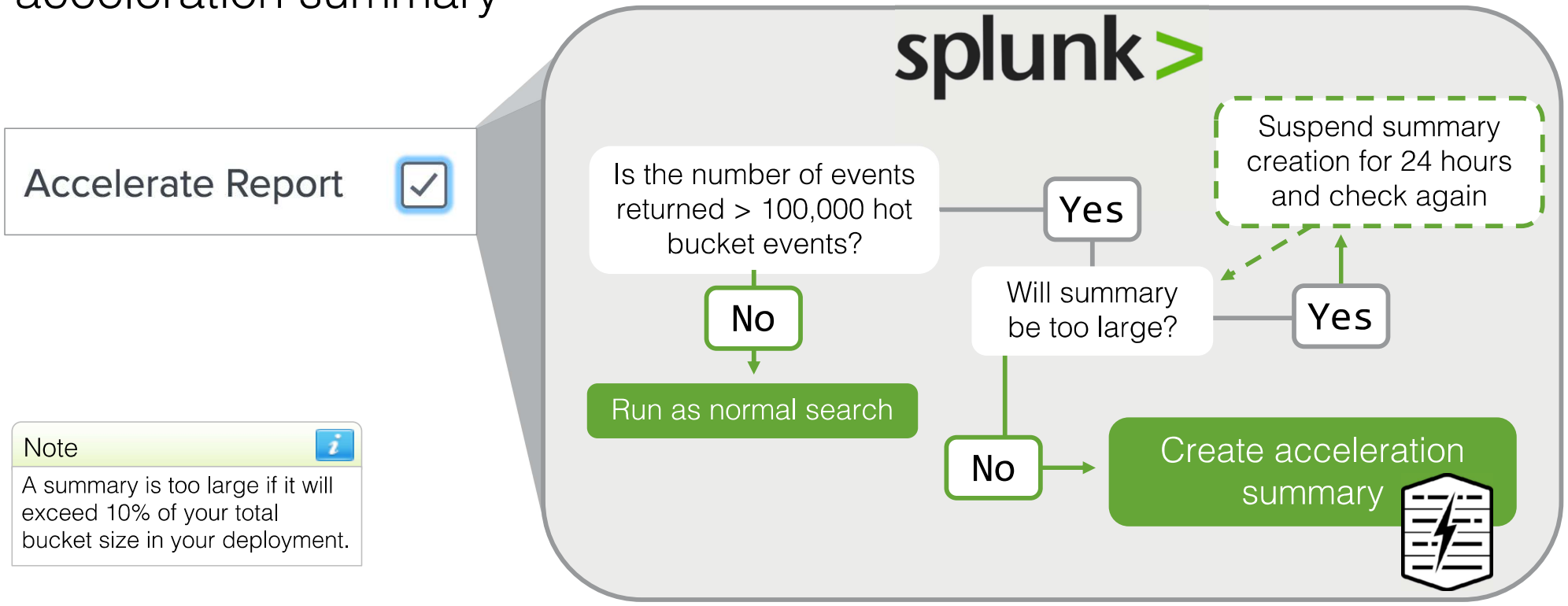


Viewing Accelerated Reports

Once accelerated, a lightning bolt appears next to the saved report in **Settings** > **Searches, Reports, and Alerts**

Acceleration Summary Not Created

Even if report acceleration is enabled, Splunk may not create an acceleration summary



Note

A summary is too large if it will exceed 10% of your total bucket size in your deployment.

Acceleration Summary Not Created (cont.)

- Some searches run faster **without** a summary if:
  - There are fewer than 100K events in the summary range
  - Summary size is projected to be too big
- If acceleration summary was defined and not created for the above reasons, Splunk:
  - Continues to check periodically
  - Creates a summary if/when the report meets the requirements

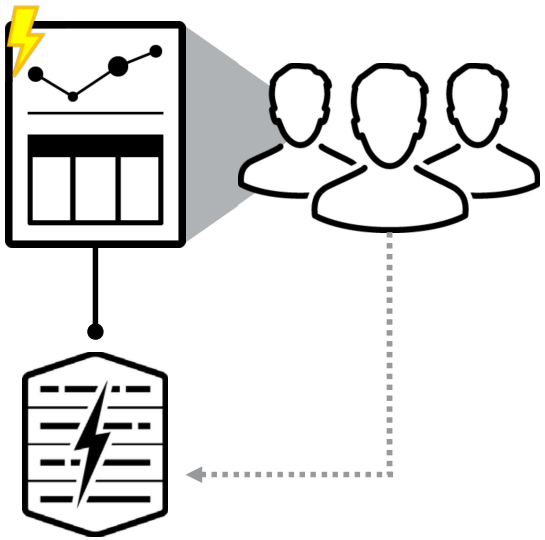
Acceleration Summary Created

- Acceleration is a good option for reports that call on 100k or more events for the summary range selected
- Splunk automatically populates acceleration summaries every 10 minutes
- Report acceleration summaries are stored by time alongside buckets in your indexes
  - Buckets are filesystem directories that store events within indexes

Acceleration Summary Created (cont.)

Splunk automatically shares summaries with users who have access to the accelerated report

Users of an accelerated shared report benefit from having access to the acceleration summary for that report



Any searches run by these users pull data from the acceleration summary when possible

Searching an Acceleration Summary

- In addition to saved accelerated reports, ad hoc searches can use the summary when:
  - Search criteria matches the base saved search
  - The user executing the ad hoc query has permission to the acceleration summary
- You can also append the search string with additional commands, for example:

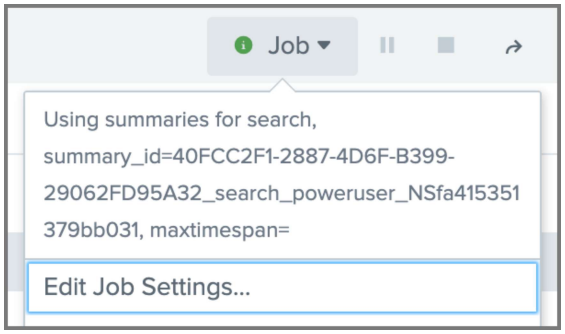
```
index=web sourcetype=access_combined
| stats count by price
```

Populating Search

```
index=web sourcetype=access_combined
| stats count by price
| eval discount = price/2
```

Ad Hoc Search

Using Summaries



- The Job Inspector shows when summaries are being used
- Deleting all reports that use an acceleration summary automatically deletes the acceleration summary

Note

The Job Inspector will show summaries being used for a search even if the search time range is longer than the summary range. This is because the portion of the search within the summary range will still benefit from acceleration. The rest of the search will run over raw data.

Viewing Report Acceleration Summaries

Settings > Report Acceleration Summaries

- **Summary ID** and **Normalized Summary ID**: unique hashes assigned to the summary (clicking these hashes loads the summary details page)
- **Reports Using Summary**: saved reports associated with the summary

Summary ID	Normalized Summary ID	Reports Using Summary	Summarization Load	Access Count	Summary Status
aa440bfb9ba071f3	NSa6199687eeb12697	License Usage Data Cube	0.0034	2 Last Access: 2m ago	Pending Updated: 45m ago
8292bd941d45d409	NScb96a73992ea0756	Buttercup Games Sales last 30 days (Product ID)	0.0392	0 Last Access: Never	Complete Updated: 9m ago
648c4d1f7a6c45b0	NS6dd2b0b61d29f02b	buttercup games sales, last 30 days	0.0156	2 Last Access: 13m ago	Pending Updated: 28m ago
e58d079826c5104c	NS57063dfa2a7932b1	Buttercup Games sales last 30 days (price and discount) Buttercup Games count by price, last 30 days	0.0000	0 Last Access: Never	Building summary - 82% Updated: < 1 min ago

Viewing Report Acceleration Summaries (cont.)

Settings > Report Acceleration Summaries

- **Summarization Load**: calculation of effort to update summary
  SL = time to run populating report / interval of populating report
- **Access Count**: how often summary used

> **Note**
> If Summarization Load is high and Access Count is low, consider deleting the summary.

Summary ID	Normalized Summary ID	Reports Using Summary	Summarization Load	Access Count	Summary Status
aa440bfb9ba071f3	NSa6199687eeb12697	License Usage Data Cube	0.0034	2 Last Access: 2m ago	Pending Updated: 45m ago
8292bd941d45d409	NScb96a73992ea0756	Buttercup Games Sales last 30 days (Product ID)	0.0392	0 Last Access: Never	Complete Updated: 9m ago
648c4d1f7a6c45b0	NS6dd2b0b61d29f02b	buttercup games sales, last 30 days	0.0156	2 Last Access: 13m ago	Pending Updated: 28m ago
e58d079826c5104c	NS57063dfa2a7932b1	Buttercup Games sales last 30 days (price and discount) Buttercup Games count by price, last 30 days	0.0000	0 Last Access: Never	Building summary - 82% Updated: < 1 min ago

Viewing Report Acceleration Summaries (cont.)

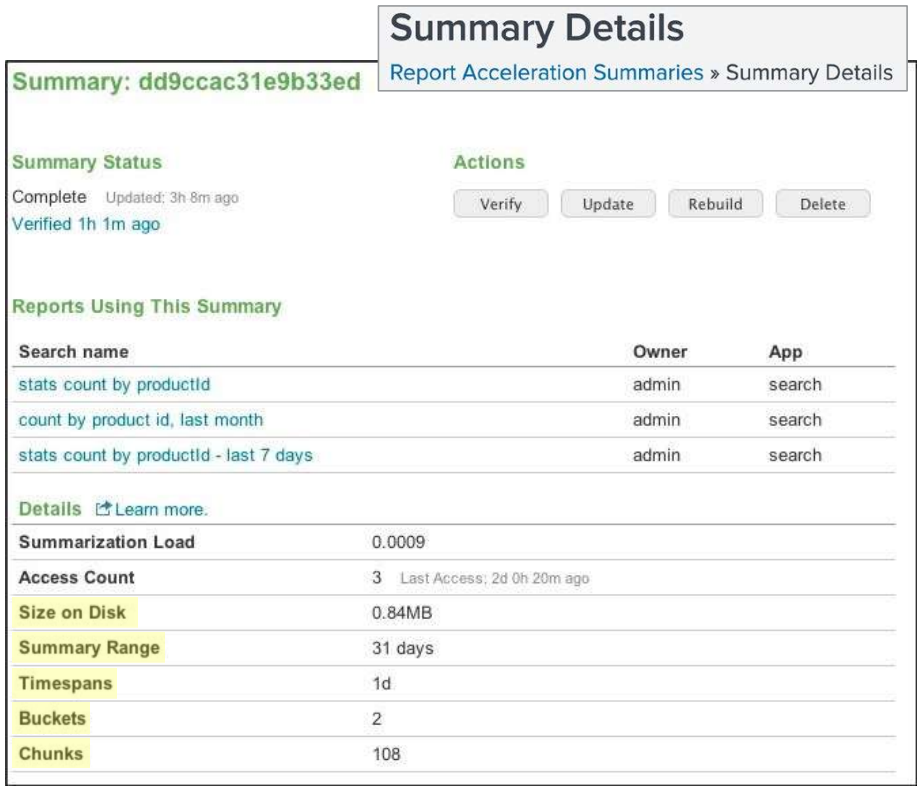
Settings > Report Acceleration Summaries

- **Summary Status**: either % of summary complete at that moment, or a status value

- **Summarization not started**
- **Pending**: the search head about to schedule new update for the summary
- **Building summary**
- **Complete**
- **Suspended**: summary size too big to be useful
- **Not enough data to summarize**: summary size too small (fewer than 100K events)

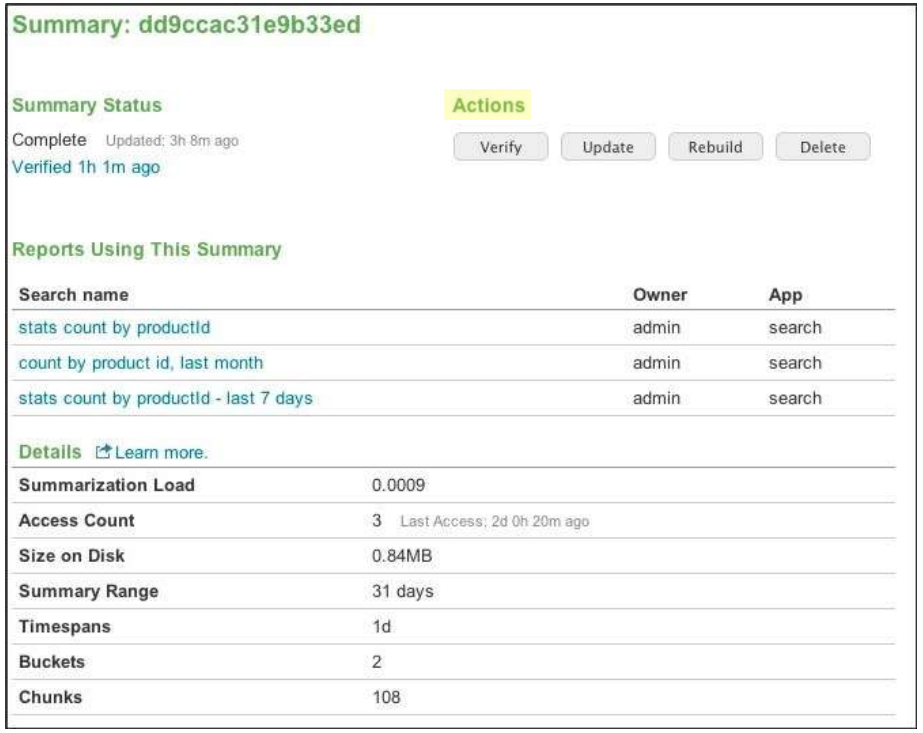
Summary ID	Normalized Summary ID	Reports Using Summary	Summarization Load	Access Count	Summary Status
aa440bfb9ba071f3	NSa6199687eeb12697	License Usage Data Cube	0.0034	2 Last Access: 2m ago	Pending Updated: 45m ago
8292bd941d45d409	NScb96a73992ea0756	Buttercup Games Sales last 30 days (Product ID)	0.0392	0 Last Access: Never	Complete Updated: 9m ago
648c4d1f7a6c45b0	NS6dd2b0b61d29f02b	buttercup games sales, last 30 days	0.0156	2 Last Access: 13m ago	Pending Updated: 28m ago
e58d079826c5104c	NS57063dfa2a7932b1	Buttercup Games sales last 30 days (price and discount) Buttercup Games count by price, last 30 days	0.0000	0 Last Access: Never	Building summary - 82% Updated: < 1 min ago

Viewing Summary Details



- Click on **Summary ID** to view **Summary Details**
  - **Size on Disk**: how much storage space the summary takes up
  - **Summary Range**: range of time spanned by the summary, relative to present moment
  - **Timespans**: size of data chunks comprising the summary
  - **Buckets**: number of index buckets the summary spans
  - **Chunks**: number of data chunks comprising the summary

Viewing Summary Details (cont.)



- Actions
  - **Verify**: examines a subset of the summary and verifies that all examined data is consistent
  - **Update**: updates the summary
  - **Rebuild**: rebuilds the summary from scratch
  - **Delete**: deletes the summary

> **Note**
>
> If accelerated report isn't returning expected results, it may be that an underlying tag, event type, or field extraction rule was changed. If that happens, use Verify to determine whether data is consistent. If verification fails, use Rebuild to recreate the summary.

Report Acceleration Lab Exercise

Time: 20 minutes

Tasks:

- Verify a search will qualify for report acceleration
- Accelerate a search
- Run an accelerated report

Data Model Acceleration

Topic Objectives

- Understand data model acceleration
- Accelerate a data model
- Use the `datamodel` command to search data models

Data Model Acceleration

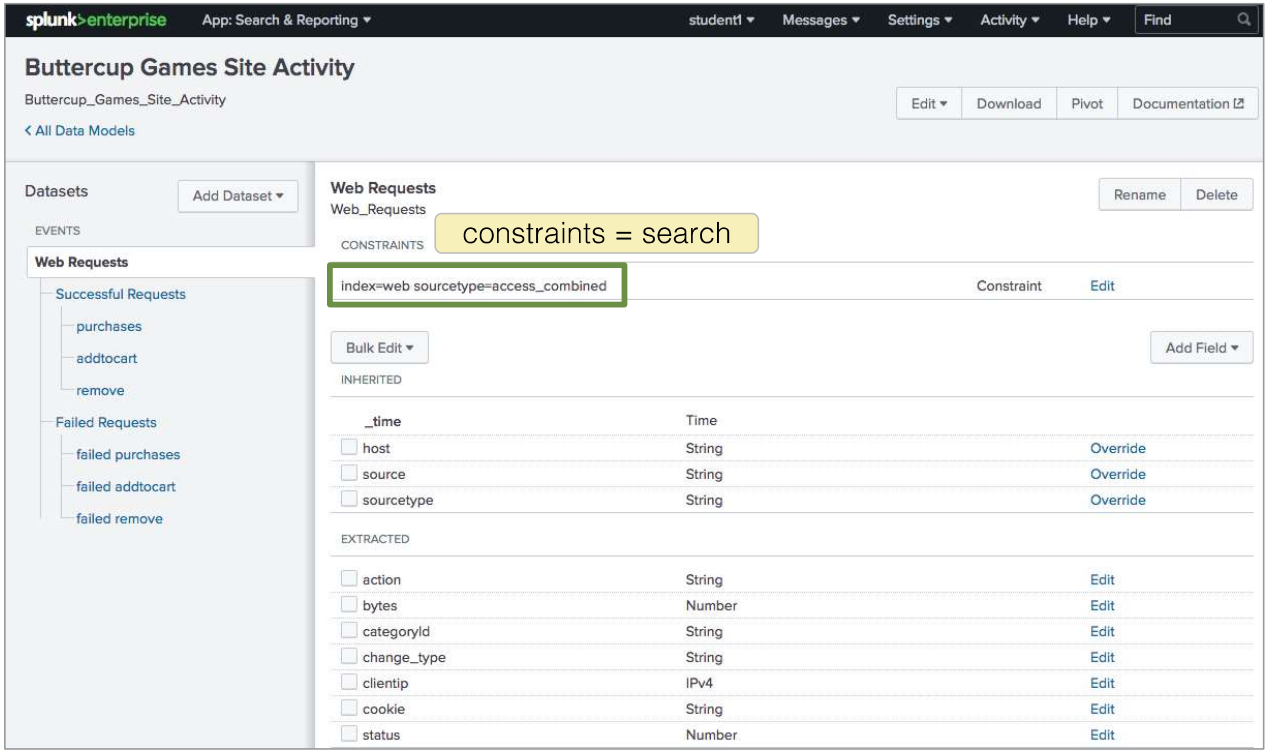
- Generates summaries to speed pivot and report completion times
- Takes the form of inverted time-series index (**`tsidx`**) files that have been optimized for speed
- Two types:
  - Ad hoc data model acceleration
  - Persistent data model acceleration

Note

Ad hoc data model acceleration occurs automatically when accessing a data model through Pivot. Ad hoc data model acceleration and Pivot are outside the scope of this course. Persistent data model acceleration will be the focus of this course.

What are Data Models?

- Hierarchically structured datasets made up of search and fields
- Represents specific categories of data
- Can be accelerated for faster performance



Persistent Data Model Acceleration

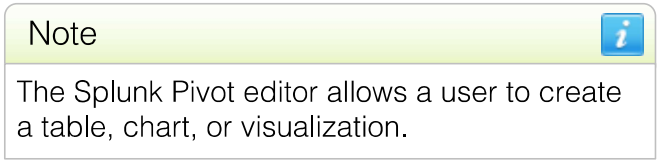
- Persistent data model acceleration builds dedicated summaries in indexes and exists as long as the data model exists
- Once accelerated, Splunk maintains the dedicated summaries
- Reports and dashboard panels generated from persistently accelerated data models complete more quickly
- Summaries can be used by Pivot, `datamodel`, and `tstats`
- Multiple users can access the summary at the same time

Note

The `datamodel` and `tstats` commands are discussed in a later topic. Pivot is outside the scope of this course.

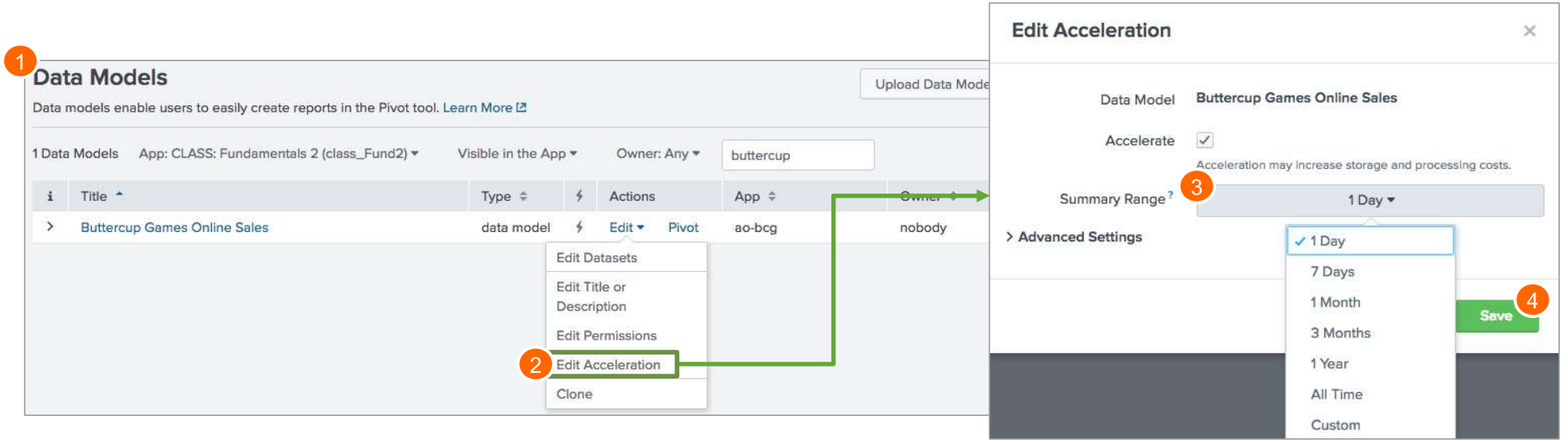
Comparing Data Model Accelerations

Ad Hoc	Persistent
The acceleration is built every time the Pivot editor is accessed	Explicitly defined before using
Exists only for the duration of user's Pivot editor session	Exists as long as data model exists
Runs over all time (i.e., can't be scoped to specific time range)	Can be scoped to specific time ranges
Reports run without any acceleration	Reports run faster and perform better overall



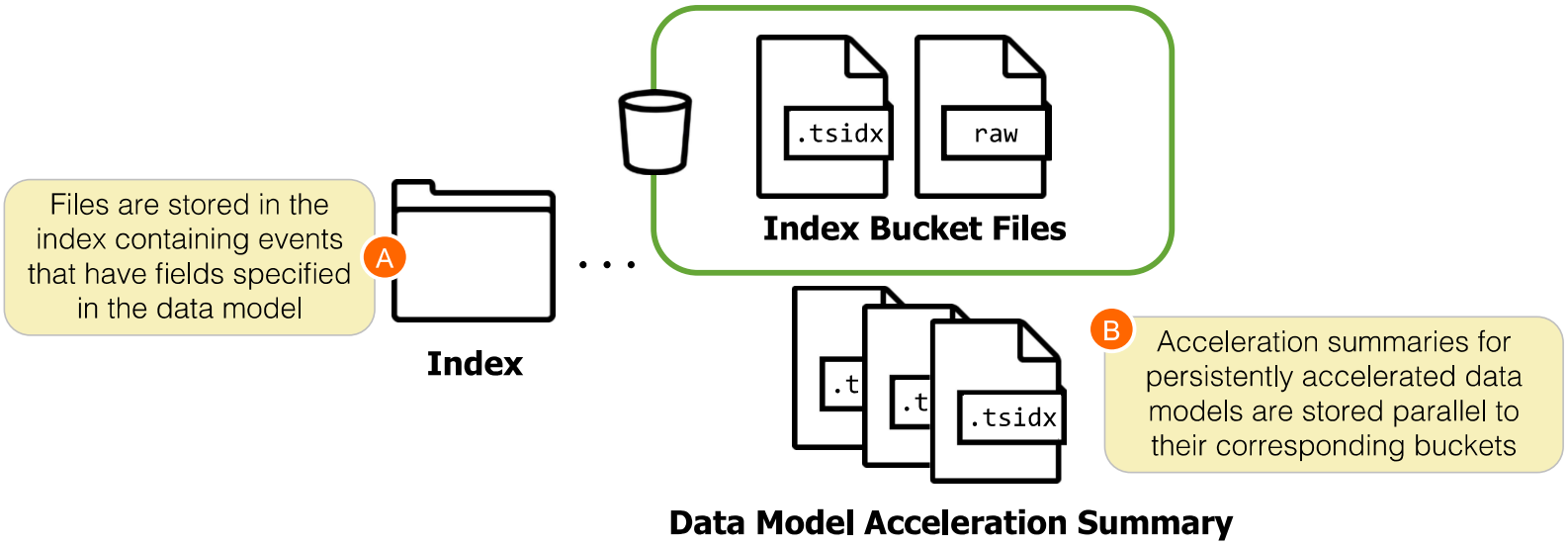
Accelerating a Data Model

1. Click **Settings > Data Models**
2. Select a data model and click **Edit > Edit Acceleration**
3. Click the **Accelerate** check box and choose a **Summary Range**
4. Click **Save**



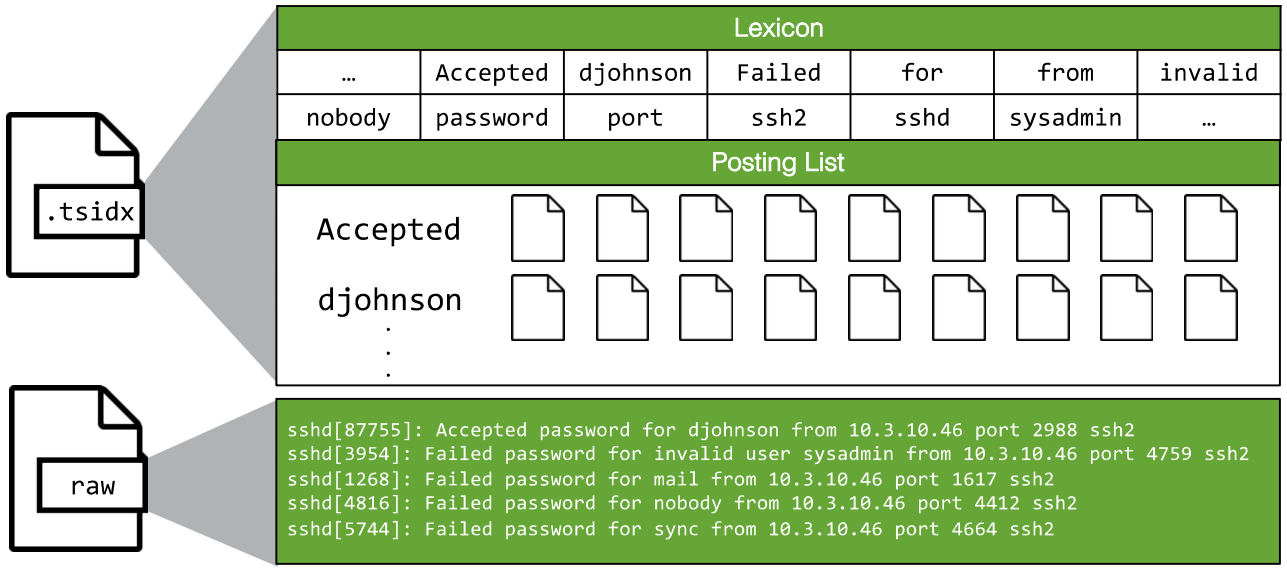
After Accelerating a Data Model

Splunk builds an acceleration summary for the specified summary range in the form of time-series index (`tsidx`) files



Time Series Index (`tsidx`) Files

- Exist inside buckets alongside raw data files
- Consist of a lexicon and a posting list and the indexed `field::value` combinations (`host`, `source`, and `sourcetype`)



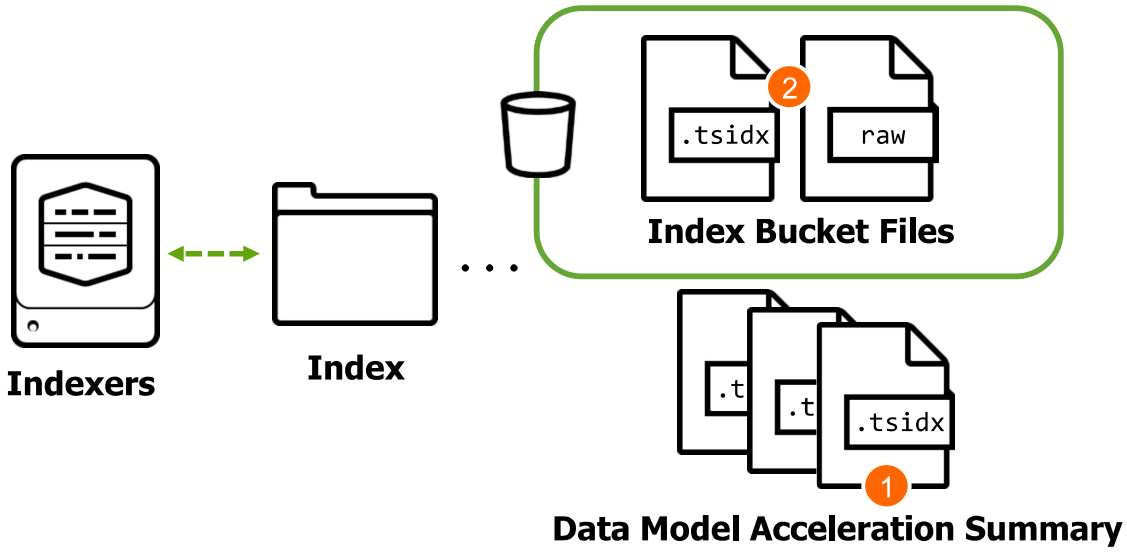
The **lexicon** is an alpha-numerically ordered list of terms found in the data at index time

The **posting list** is an array of pointers that match each term to events in the raw data files

Splunk uses the pointers to search just the events that match the terms, making the search much more efficient

Searching the Acceleration Summary

1. Indexer retrieves information about the data model that has been stored on disk in the `tsidx` files that make up the acceleration summary
2. Indexer pulls additional events from bucket files if search is outside data summary range



Accelerated Data Model Considerations

- The acceleration summary always contains a store of data that at least meets the summary range (may slightly exceed)
- Splunk updates **`tsidx`** files every 5 minutes and removes outdated summary data every 30 minutes
- Accelerated data model summaries can be accessed through:
  - Pivot editor (outside the scope of this course)
  - Searches using **`pivot`**, **`tstats`** or **`datamodel`**

Comparing Acceleration Methods

Report Acceleration	• Uses automatically created summaries to speed completion times for qualified reports • Easier to create than summary indexes and backfills automatically • Depending on the defined time span, periodically ages out data • Can correct gaps and overlaps from the UI "rebuild" feature • Cannot create a "data-cube" and report on smaller subsets
Data Model Acceleration	• Uses automatically created summaries to speed completion times for pivots • Takes the form of time-series index (`tsidx`) files

`datamodel` Command

- Used to display the structure of a data model or to search against it
- Returns a description of all or a specified data model and its objects
- A generating command
  - Must follow a leading | pipe
  - Must be the first command in a search

Note

Use the `datamodel` command by itself (without arguments) to display all the data models in your deployment that you have access to.

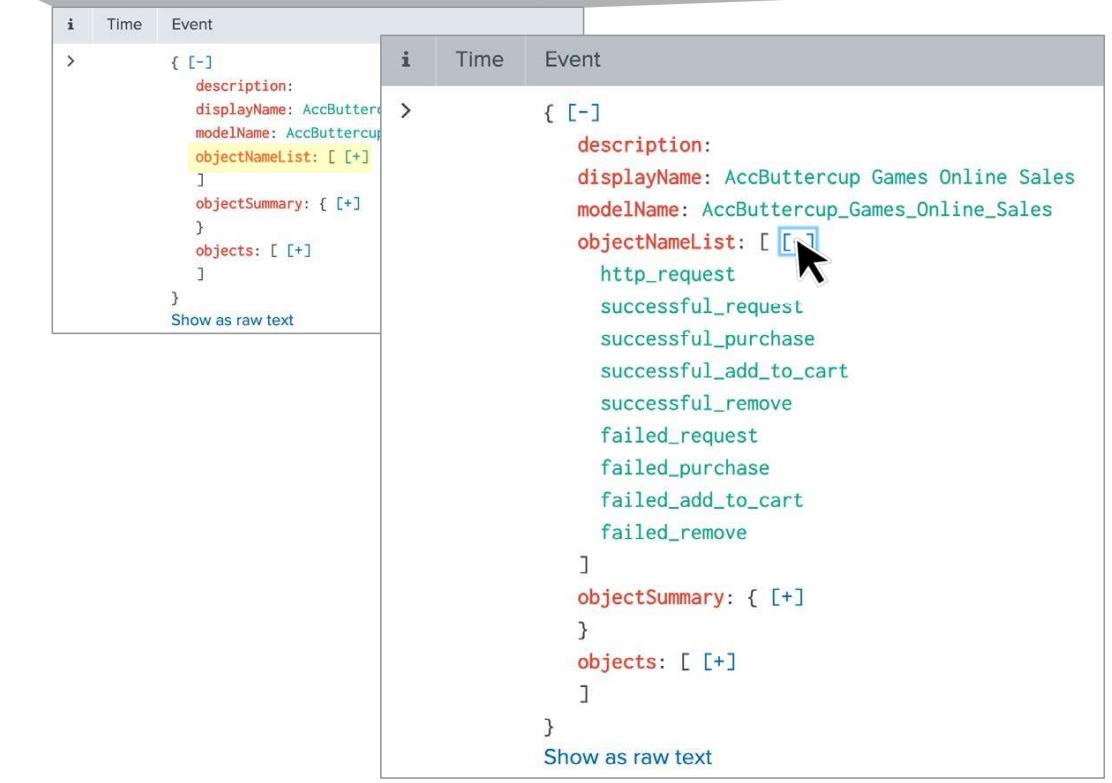
`datamodel` Command (cont.)

- If the name of the data model is included as the first argument, Splunk shows the details of that data model in JSON format

```
| datamodel [data_model_name]
```

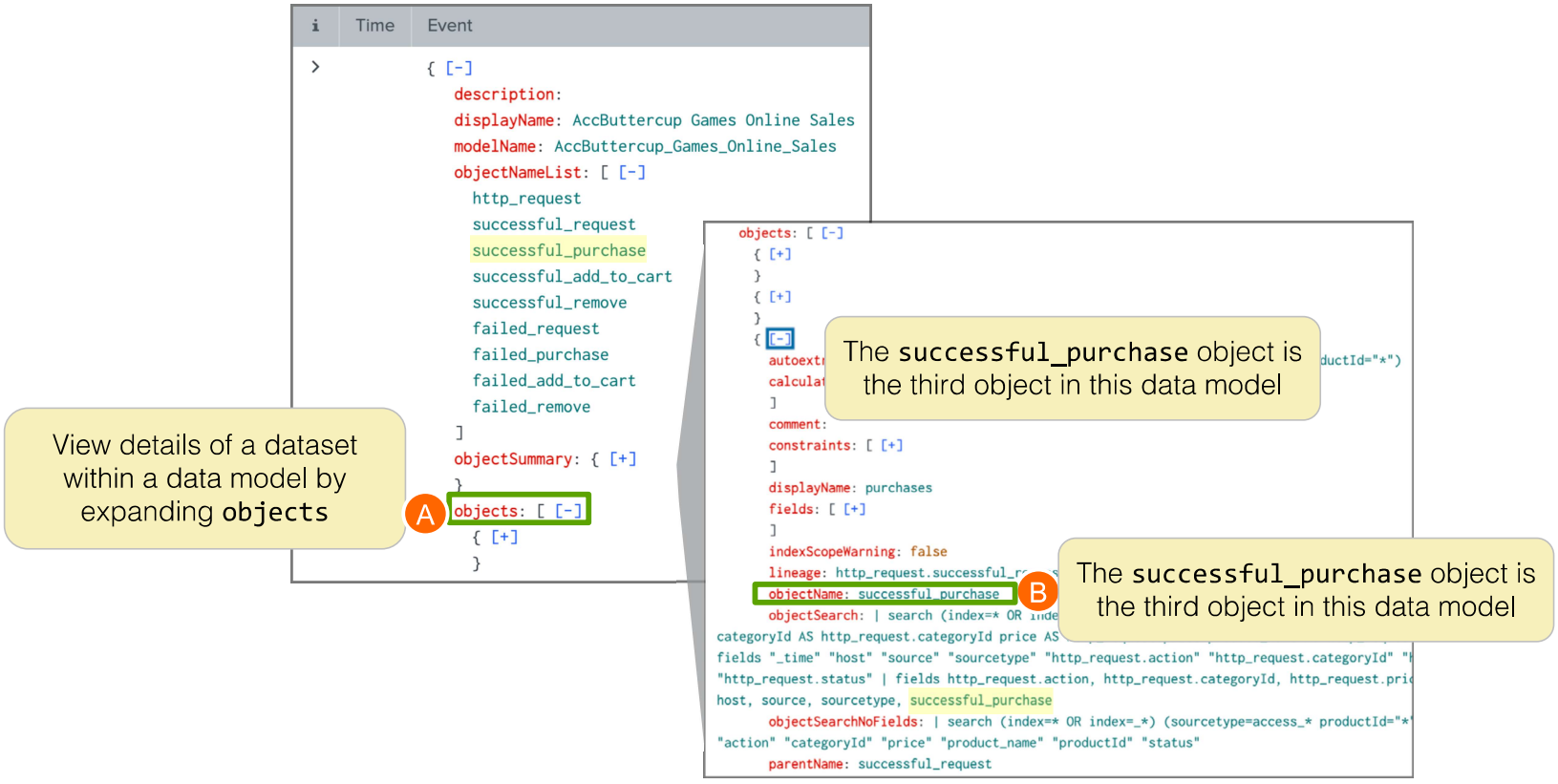
- Click the **+** next to **`objectNameList`** to show all dataset names in the selected data model

```
| datamodel AccButtercup_Games_Online_Sales
```



`datamodel` Command (cont.)

```
| datamodel AccButtercup_Games_Online_Sales
```



`datamodel` Command (cont.)

Alternatively, you can display a dataset within a data model by using the dataset name as the second argument

```
| datamodel [data_model_name] [dataset_name]
```

```
| datamodel vsales apac
```

`datamodel` Command: Options

To view the events associated with the specified dataset, use the `search` option

```
| datamodel [data_model_name] [dataset_name] search
```

```
| datamodel vsales apac search
```

i	Time	Event
>	5/24/22 6:02:53.000 PM	[24/May/2022:18:02:53] VendorID=7006 Code=A AcctID=xxxxxxxxxxx8445 host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	5/24/22 3:42:09.000 PM	[24/May/2022:15:42:09] VendorID=7035 Code=M AcctID=xxxxxxxxxxx3674 host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	5/24/22 11:21:21.000 AM	[24/May/2022:11:21:21] VendorID=7022 Code=F AcctID=xxxxxxxxxxx8798 host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	5/24/22 10:56:58.000 AM	[24/May/2022:10:56:58] VendorID=7007 Code=M AcctID=xxxxxxxxxxx5656 host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales
>	5/24/22 8:08:12.000 AM	[24/May/2022:08:08:12] VendorID=7006 Code=M AcctID=xxxxxxxxxxx1550 host = vendorUS1 source = /opt/log/vendorUS1/vendor_sales.log sourcetype = vendor_sales

`datamodel` Command: Options (cont.)

The **`flat`** option returns the same results as **`search`** but field names are "flattened" by stripping hierarchical information

```
| datamodel vsales apac search
```

INTERESTING FIELDS
- a apac.categoryId 7
- # apac.price 7
- a apac.product_name 14
- a apac.productId 14
- # apac.sale_price 6
- a apac.Vendor 22
- a apac.VendorCity 30
- a apac.VendorCountry 13
- # apac.VendorID 30
- # apac.VendorLatitude 30
- # apac.VendorLongitude 30
- a apac.VendorStateProvince 27

```
| datamodel vsales apac flat
```

INTERESTING FIELDS
- a categoryId 7
- # price 7
- a product_name 14
- a productId 14
- # sale_price 6
- a Vendor 22
- a VendorCity 30
- a VendorCountry 13
- # VendorID 30
- # VendorLatitude 30
- # VendorLongitude 30
- a VendorStateProvince 27

`datamodel` Command: Options (cont.)

- The dataset name and **`search`** argument aren't valid unless preceded by the data model name
- When using the **`datamodel`** command, the data model name and dataset name are case sensitive

`datamodel` Command: `summariesonly`

```
| datamodel [data_model_name] [dataset_name] search summariesonly=<bool>
```

- **`summariesonly=true`** returns results only from the **`tsidx`** data generated by the acceleration summary, i.e. summarized data
  - Maximizes speed of search execution
- **`summariesonly=false`** (default) returns summarized and unsummarized data

```
| datamodel vsales apac search
```

This search has completed and has returned **98** results by scanning **103** events in **0.662** seconds

```
| datamodel vsales apac search summariesonly=true
```

This search has completed and has returned **98** results by scanning **98** events in **0.227** seconds

Data Model Acceleration Lab Exercise

Time: 15 minutes

Tasks:

- Use the `datamodel` command to search a data model
- Use the `summariesonly` function of the `datamodel` command to view the event count of two data models over the last 5 minutes

Use the `tstats` Command

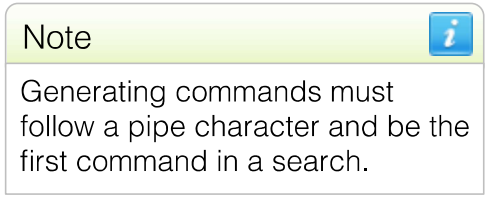
Topic Objectives

- Explore the `tstats` command
- Search acceleration summaries with `tstats`
- Search data models with `tstats`
- Compare `tstats` and `stats`

`tstats` Command

```
| tstats <stats-func> [summariesonly=<bool>]
  [from datamodel=<data_model_name>]
  [where <searchQuery>] [by <field-list>]
```

- Performs statistical queries on **`tsidx`** files
- Requires use of a statistical function
- Use a **`from`** clause to pull events from a specific data model
- Use **`where`** clause to filter results
- Group results with a **`by <field-list>`** clause
- Generating command



`tstats` Command (cont.)

- When querying indexed data, **`tstats`** searches **`tsidx`** files which means:
  - Search is limited to only indexed fields in the **`tsidx`**
  - **`tstats`** searches execute very fast because it does not open or read raw events

```
| tstats values(sourcetype) as sourcetype by index
```

index	sourcetype
alerts	generic_single_line
games	SimCubeBeta
main	netstat ps
network	cisco_esa cisco_firewall cisco_wsa_squid
sales	sales_entries vendor_sales

This search was run over **All Time** and completed in 0.17 seconds!

Note

`tstats` use acceleration summaries if they are available or if **`summariesonly=true`**.

`tstats` Command: `stats-func`

Scenario

ITOps wants a list of all source types by index.

```
| tstats values(sourcetype) as sourcetype by index
```

index	sourcetype
games	SimCubeBeta
network	cisco_esa cisco_firewall cisco_wsa_squid
os	ps
sales	sales_entries vendor_sales
security	history_access linux_secure winauthentication_security
summary	stash
systems	server_log system_info system_info_xml
web	access_combined
win_audit	win_audit

Most functions available to `stats` can be used with `tstats`

`tstats` Command: `from` Clause

Use the **`from`** clause to search through **`tsidx`** files not created at index time such as data model acceleration summary **`tsidx`** files

Scenario

TechOps wants a count of all web requests during the last 24 hours.

```
| tstats count from datamodel=AccButtercup_Games_Online_Sales
```

count
7478

`tstats` Command: Without `from` Clause

Scenario

User wants to count the events per index, for all indexes to which they have access.

If you don't use a `from` clause, a search is performed on indexed fields in the index `tsidx`

Note

Statistical queries can only be performed on indexed fields, not search time fields.

```
| tstats count by index
| sort -count
```

index	count
win_audit	40556
systems	19069
security	13075
os	11880
web	10383
sales	8315
summary	5251
network	3834
games	1772

This search has completed and has returned **9** results

by scanning **114,135** events in **0.138** seconds

`tstats` Command: `by` Clause

Group by any number of fields using `by <field-list>`

Scenario

TechOps is reconfiguring the web servers and wants a count of all web requests per web server over the last 24 hours.

```
| tstats count from datamodel=AccButtercup_Games_Online_Sales by host
```

host	count
www1	2475
www2	2664
www3	2362

`tstats` Command: `summariesonly`

```
| tstats <stats-func> [summariesonly=<bool>] [from datamodel=<data_model_name>]
```

- **`true`** or **`t`** returns results only from the **`tsidx`** data generated by the acceleration and does not include non-summarized data
- **`false`** or **`f`** (default) generates results from both summarized **and** non-summarized data

```
| tstats count from
  datamodel=AccButtercup_Games_Online_Sales
```

(All time) **348,217** results by scanning **348,223** events in **0.335** seconds

```
| tstats count from
  datamodel=AccButtercup_Games_Online_Sales
  summariesonly=t
```

(All time) **348,199** results by scanning **348,199** events in **0.107** seconds

`tstats` Command: `summariesonly` (cont.)

- When running a search with **`summariesonly`** set to **`false`**, you might notice a larger result count because:
  - Some of the index data may not have been added to the summary yet
  - The search range may be greater than the summary range
- If used with an unaccelerated data model, **`summariesonly=t`** produces no results

```
| tstats count from datamodel=AccButtercup_Games_Online_Sales summariesonly=t
```

Accelerated Data Model

count
2901

```
| tstats count from datamodel=Buttercup_Games_Online_Sales summariesonly=t
```

Unaccelerated Data Model

count
0

Data Model Field Names with `tstats`

- Use a data model field with **`tstats`** by referencing its location in the data model with dot notation (**`owner.fieldName`**)
- Use the **`datamodel`** command to return details of the data model and its objects
  - Then, note the **`owner`** for the field you want to access

```
| tstats sum(http_request.price) from
datamodel=AccButtercup_Games_Online_Sales
```

sum(http_request.price)
154782.72

```
| datamodel AccButtercup_Games_Online_Sales
```


```
displayName: Acc-Buttercup Games Online Sales
modelName: AccButtercup_Games_Online_Sales
objectNameList: [ [-]
  http_request
  successful_request
  successful_purchase
  successful_add_to_cart
  successful_remove
  failed_request
  failed_purchase
  failed_add_to_cart
  failed_remove
]
objectSummary: { [+] }
objects: [ [-]
  { [+]
    displayName: removed
    fields: [ [-]
      { [+]
      }
      { [+]
      }
      { [-]
        comment:
        displayName: price
        editable: true
        fieldName: price
        fieldSearch:
        hidden: false
        multivalue: false
        owner: http_request
        required: false
        type: number
```


`fieldName: price`

`owner: http_request`

`datamodel` Notation With `tstats`

If a data model has more than one accelerated root dataset, you must specify the dataset you want by using dot notation

`datamodel.dataset`

```
| tstats sum(http_request.price) as tsales from
  datamodel=AccButtercup_Games_Online_Sales.http_request
  where (http_request.action=purchase AND http_request.status=200) by http_request.product_name
```

Note

This example also uses `owner.fieldName` notation as shown on the previous slide. The `owner` name is the name of the dataset.

`datamodel` Notation With `tstats` Example

Scenario

The Online Sales manager launched a new campaign yesterday. Provide her with the total sales for yesterday.

```
| tstats sum(http_request.price) as tsales from
  datamodel=AccButtercup_Games_Online_Sales.http_request
  where (http_request.action=purchase AND http_request.status=200)
  by http_request.product_name
| sort - tsales
| eval tsales="$".tostring(tsales,"commas")
| rename http_request.product_name as Product, tsales as "Daily Sales"
| fields Product, "Daily Sales"
```

Product	Daily Sales
Dream Crusher	$9,197.70
Orvil the Wolverine	$7,238.19
World of Cheese	$7,147.14
Manganiello Bros.	$7,118.22
Mediocre Kingdoms	$5,872.65

Searching Unaccelerated Data Models

- `tstats` can search unaccelerated data models
  - However, searches run the same as a normal search with no performance benefit
- A best practice is to use `tstats` with accelerated data models

```
| tstats count from datamodel=Buttercup_Games_Online_Sales
  by host
| sort -count
```

3 results by scanning 272,746 events in 4.885 seconds

```
| tstats count from datamodel=AccButtercup_Games_Online_Sales
  by host
| sort -count
```

3 results by scanning 272,746 events in 0.745 seconds

host	count
www3	91112
www2	89741
www1	89231

`tstats` Command: `span` Option

- If you group by `_time`, use `span` (e.g., `span=3m`) to group into time buckets
- If you don't specify a span, the value set by the time picker determines the range

Search Time Range	Default Span
5 minutes	5 seconds
15 minutes	10 seconds
60 minutes	1 minute
4 hours	5 minutes
24 hours	30 minutes
7 days	1 day

`tstats` Command: Wildcards

- `tstats` does not support wildcarded fields, however the wildcard can be used in the **`where`** clause to search on field values
- You can specify:

```
| tstats  count where host=w* by source
| sort -count
```

source	count
/var/log/secure	1523
ps	30
/opt/log/www3/access.log	16
/opt/log/cisco_router1/cisco_ironport_mail.log	9
/var/log/messages	5
/opt/log/SIMlog/simgame.log	4

- But not these:

```
| tstats count(source*)
```

Error in 'TsidxStats': Wildcards (*) are not supported in aggregate fields

```
| tstats count where host=w* by source*
```

Error in 'TsidxStats': Wildcards (*) are not supported in groupby fields

`tstats` versus `stats` for Indexed Fields

Scenario

IT is doing resource planning and wants the event load for the security index. Count the events for all time by source, sourcetype, and host. Sort descending on count and format with commas.

When working with a massive amount of data and using indexed fields, consider using `tstats`

```
| tstats count as events
  where index=security by source, sourcetype, host
| sort -events
| eval events = tostring(events,"commas")
```

11 results by scanning 971,466 events in 0.08 seconds

```
index=security
| stats count as events by source, sourcetype, host
| sort -events
| eval events = tostring(events, "commas")
```

11 results by scanning 971,016 events in 1.59 seconds

```
| tstats count as events
  where index=* by source, sourcetype, host
| sort -events
| eval events = tostring(events,"commas")
```

101 results by scanning 12,945,032 events in 0.39 seconds

```
index=*
| stats count as events by source, sourcetype, host
| sort -events
| eval events = tostring(events, "commas")
```

101 results by scanning 12,944,443 events in 58.8 seconds

`stats` to `tstats` Search Optimization

- Any **`datamodel`** search using the **`stats`** command is converted automatically to use **`tstats`**

```
| datamodel vsales us search
| stats sum(us.price) by us.product_name
```

Since `stats` to `tstats` optimization is already enabled, this search will become:

```
| tstats sum(us.sales) from
  datamodel=vsales.us by us.product_name
```

- A similar optimization is available for **`stats`** searches but, by default, this feature is not enabled

```
index=web
| stats count
```

If `stats` to `tstats` optimization is enabled, this search will become:

```
| tstats count where index=web
```

- Greatly increases speed of searches that rely solely on indexed fields or simple counts

Using the `tstats` Command Lab Exercise

Time: 15 minutes

Tasks:

- Use the `tstats` command to return a count of all events
- Use the `tstats` command to search and transform summarized data from a data model

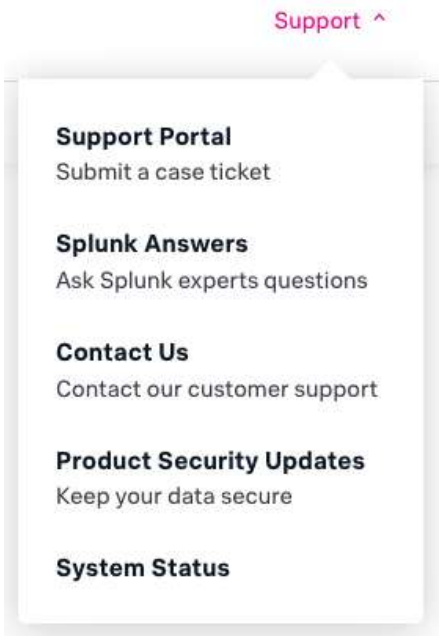
Wrap-up Slides

Community

- Splunk Community Portal
  community.splunk.com
  - Answers
  - Discussions
  - Splunk Trust
  - User Groups
  - Ideas
- Splunk Blogs
  splunk.com/blog/
- Splunk Apps
  splunkbase.com
- Splunk Dev Google Group
  groups.google.com/forum/#!forum/splunkdev
- Splunk Docs on Twitter
  twitter.com/splunkdocs
- Splunk Dev on Twitter
  twitter.com/splunkdev
- Splunk Live!
  splunklive.splunk.com
- .conf
  conf.splunk.com

Support Programs

- **Web**
  - Documentation: dev.splunk.com and docs.splunk.com
  - Wiki: wiki.splunk.com
- **Splunk Lantern**
  Guidance from Splunk experts
  - lantern.splunk.com
- **Global Support**
  Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365
  - Web: splunk.com/index.php/submit_issue
- **Enterprise, Cloud, ITSI, Security Support**
  - Web: splunk.com/en_us/about-splunk/contact-us.html#tabs/customersupport
  - Phone: (855) SPLUNK-S or (855) 775-8657



Learning Paths (cont.)

Knowledge Manager - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Introduction to Dashboards
- Dynamic Dashboards
- Using Choropleth
- Search Optimization *

Learning Paths

Search Expert - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

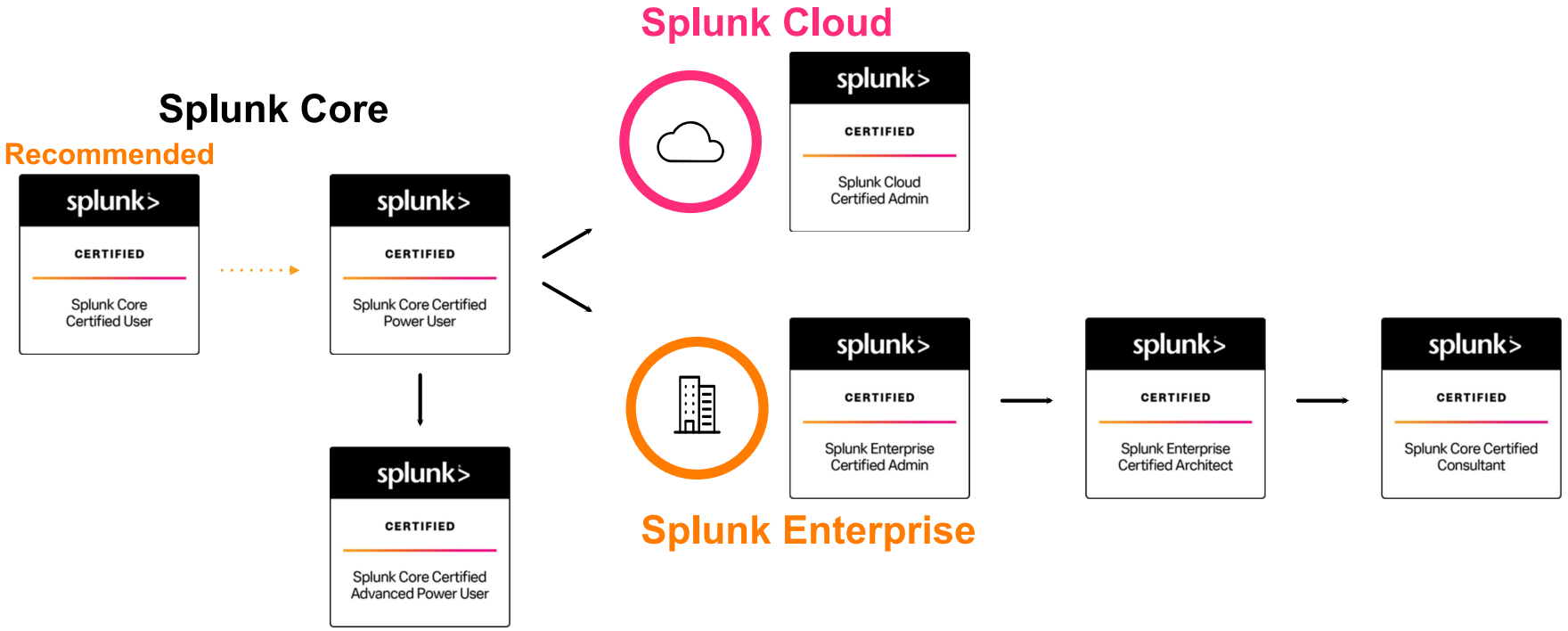
- What is Splunk *
- Introduction to Splunk *
- Using Fields *
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization *

Splunk Certification

Offerings & Requirements

Splunk Core and Beyond

Regardless of which Splunk product you use, it all starts with Splunk Core



Splunk Core Certified User

This entry-level certification demonstrates an individual's basic ability to navigate and use Splunk software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 1 **or** the following courses:

- What is Splunk?
- Intro to Splunk
- Using Fields
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Leveraging Lookups and Subsearches
- Search Optimization
- Enriching Data with Lookups
- Data Models

See here for registration assistance.

Congratulations! You are a...

Recommended Next Step

- Splunk Core Certified Power User

Splunk Core Certified Power User

This entry-level certification demonstrates an individual's foundational competence of Splunk's core software

Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 2 **or** the following courses:

- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Correlation Analysis
- Search Under the Hood
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Data Models
- Using Choropleth

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Core Certified Advanced Power User
- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

 |

Splunk Core Certified Advanced Power User

This certification demonstrates an individual's ability to generate complex searches, reports, and dashboards with Splunk's core software to get the most out of their data

Prerequisite Certification(s):

- Splunk Core Certified Power User

Prerequisite Course(s):

- None

Splunk Core Certified Advanced Power User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 3, Creating Dashboards, and Advanced Searching & Reporting **or** the following courses:

- Using Fields
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Using Choropleth
- Introduction to Dashboards
- Dynamic Dashboards

See here for registration assistance.

Congratulations! You are a...

Recommended Next Steps

- Splunk Enterprise Certified Admin
- Splunk Cloud Certified Admin

Thank You