Search Optimization – Lab Solutions Guide

Overview

Welcome to the Splunk Education lab environment. These lab exercises will test your knowledge of report acceleration, data model acceleration, and querying of `tsidx` files and acceleration summaries with **`tstats`** and **`datamodel`** commands.

Scenario

You will use data from the international video game company, Buttercup Games. A list of source types is provided below.

NOTE: This is a lab environment driven by data generators with obvious limitations. This is not a production environment. Screenshots approximate what you should see, not the **exact** output.

Index	Type	Sourcetype	Interesting Fields
web	Online sales	**access_combined**	action, bytes, categoryId, clientip, itemId, JSESSIONID, price, productId, product_name, referer, referer_domain, sale_price, status, user, useragent
security	Active Directory	**winauthentication_security**	LogName, SourceName, EventCode, EventType, User
	Badge reader	**history_access**	Address_Description, Department, Device, Email, Event_Description, First_Name, last_Name, Rfid, Username
	Web server	**linux_secure**	action, app, dest, process, src_ip, src_port, user, vendor_action
sales	Business Intelligence server	**sales_entries**	AcctCode, CustomerID, TransactionID
	Retail sales	**vendor_sales**	categoryId, product_name, productId, sale_price, Vendor, VendorCity, VendorCountry, VendorID, VendorStateProvince
network	Email security data	**cisco_esa**	dcid, icid, mailfrom, mailto, mid
	Web security appliance data	**cisco_wsa_squid**	action, cs_method, cs_mime_type, cs_url, cs_username, sc_bytes, sc_http_status, sc_result_code, severity, src_ip, status, url, usage, x_mcafee_virus_name, x_wbrs_score, x_webcat_code_abbr
	Firewall data	**cisco_firewall**	bcg_ip, dept, Duration, fname, IP, lname, location, rfid, splunk_role, splunk_server, Username

splunk>

Common Commands and Functions

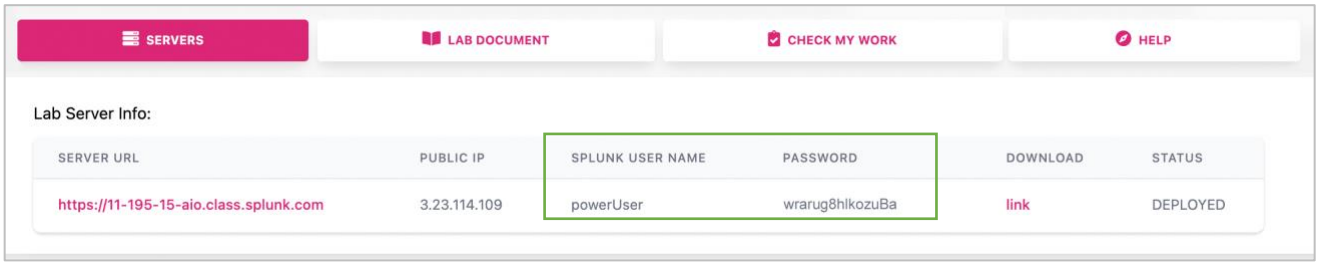
These commands and statistical functions are commonly used in searches but may not have been explicitly discussed in the module. Please use this table for quick reference. Click on the hyperlinked SPL to be taken to the Search Manual for that command or function.

SPL	Type	Description	Example
sort	command	Sorts results in descending or ascending order by a specified field. Can limit results to a specific number.	*Sort the first 100 `src_ip` values in descending order* `| sort 100 -src_ip`
where	command	Filters search results using eval-expressions.	*Return events with a `count` value greater than 30* `| where count > 30`
rename	command	Renames one or more fields.	*Rename `SESSIONID` to 'The session ID'* `| rename SESSIONID as "The session ID"`
fields	command	Keeps (+) or removes (-) fields from search results.	*Remove the `host` field from the results* `| fields - host`
stats	command	Calculates aggregate statistics over the results set.	*Calculate the total sales, i.e. the sum of `price` values* `| stats sum(price)`
eval	command	Calculates an expression and puts the resulting value into a new or existing field.	*Concatenate `first_name` and `last_name` values with a space to create a field called "`full_name`"* `| eval full_name=first_name." ".last_name`
table	command	Returns a table.	*Output `vendorCountry`, `vendor`, and `sales` values to a table* `| table vendorCountry, vendor, sales`
sum()	statistical function	Returns the sum of the values of a field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Calculate the sum of the `bytes` field* `| stats sum(bytes)`
count or count()	statistical function	Returns the number of occurrences of all events or a specific field. Can be used with **`stats`**, **`timechart`**, and **`chart`** commands.	*Count all events as "events" and count all events that contain a value for `action` as "action"* `| stats count as events, count(action) as action`

Refer to the Search Reference Manual for a full list of commands and functions.

Lab Connection Info

Access labs using the server URL, user name, and password shown in your lab environment.



Lab Exercise 1 – Report Acceleration

Description

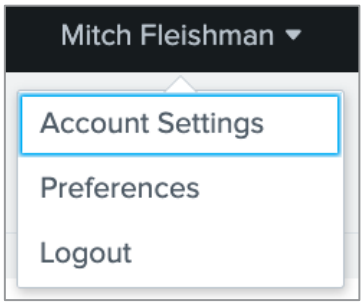
Configure the lab environment user account. Then, verify and accelerate a report and test its performance.

Steps

Task 1: Log into Splunk and change the account name and time zone.

Set up your lab environment to fit your time zone. This also allows the instructor to track your progress and assist you if necessary.

1. Log into your Splunk lab environment using the username and password provided to you.
2. You may see a pop-up window welcoming you to the lab environment. You can click **Continue to Tour** but this is not required. Click **Skip** to dismiss the window.
3. Click on the username you logged in with (at the top of the screen) and then choose **Account Settings** from the drop-down menu.
4. In the **Full name** box, enter your first and last name.
5. Click **Save**.
6. Reload your browser to reflect the recent changes to the interface. (This area of the web interface will be referred to as ***user name***.)



After you complete step 6, you will see your name in the web interface.

NOTE: Sometimes there can be delays in executing an action like saving in the UI or returning results of a search. If you are experiencing a delay, please allow the UI a few minutes to execute your action.

7. Navigate to ***user name*** **> Preferences.**
8. Choose your local time zone from the **Time zone** drop-down menu.
9. Click **Apply**.
10. (Optional) Navigate to ***user name*** **> Preferences > SPL Editor > Search auto-format** and click on the toggle to activate auto-formatting. Then click **Apply**. When the pipe character is used in search, the SPL Editor will automatically begin the pipe on a new line.

Search auto-format disabled (default)

Search auto-format enabled

Scenario: Sales wants a rolling 30 day report on all successful online purchases. Given the large volume of data, IT wants to make sure it completes as quickly as possible.

Task 2: Save a search as a report and accelerate it. Then, verify that the search is accelerated and is operating with increased speed.

11. Verify this search will qualify for report acceleration. Edit the search if necessary.

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

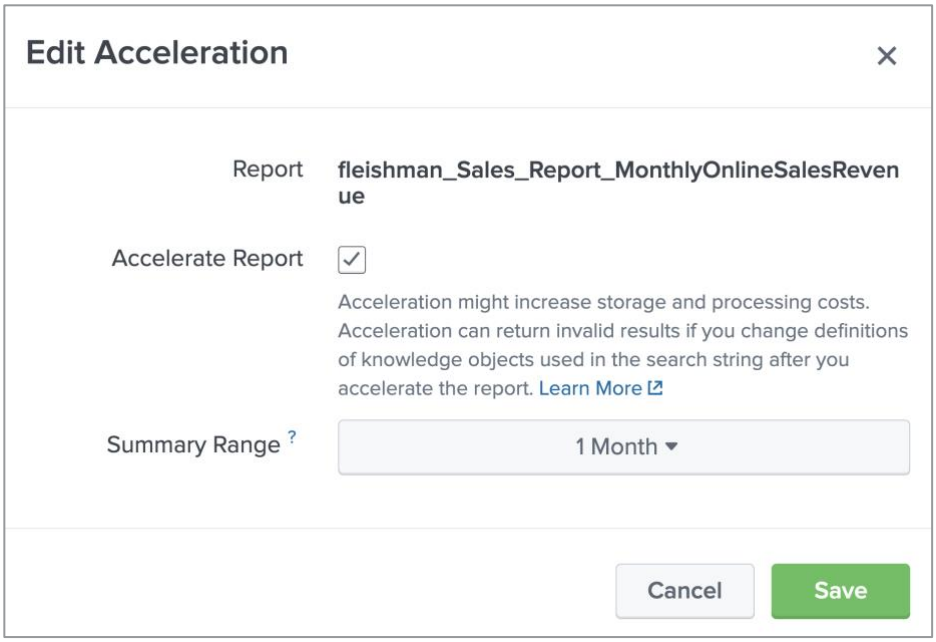
    This search qualifies for report acceleration because it has a transforming command, **`stats`**, and the transforming command is not preceded by any centralized streaming or non-streaming commands. The **`fields`** command that precedes **`stats`** is a distributable streaming command which is allowed for report acceleration.

12. Run the search over the **Last 30 days**.

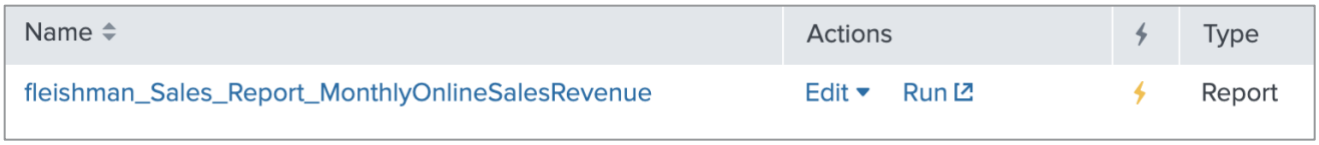
productId	revenue
BS-AG-G09	$7,896.84
CU-PG-G06	$6,336.83
DB-SG-G01	$13,369.65
DC-SG-G02	$21,634.59
FI-AG-G08	$14,356.41
FS-SG-G03	$10,395.84

13. Click **Job** > **Inspect Job** and note how long the search took to complete. (When this screenshot was taken, the search took 1.819 seconds to complete.)
14. Now, you will save the report and accelerate it. Click **Save As** > **Report**.
15. **Title** the report using your last name: *lastName*_Sales_Report_MonthlyOnlineSalesRevenue.
16. For **Time Range Picker** choose **No**.

17. Click **Save**.
18. On the **Your Report Has Been Created** screen, choose **Acceleration**.
19. On the **Edit Acceleration** screen, click the **Accelerate Report** checkbox.
20. Set the **Summary Range** to **1 Month**.



21. Click **Save**.
22. Navigate to **Settings** > **Searches, Reports, and Alerts**. Verify your report has been accelerated. There should be a yellow lightning bolt present.

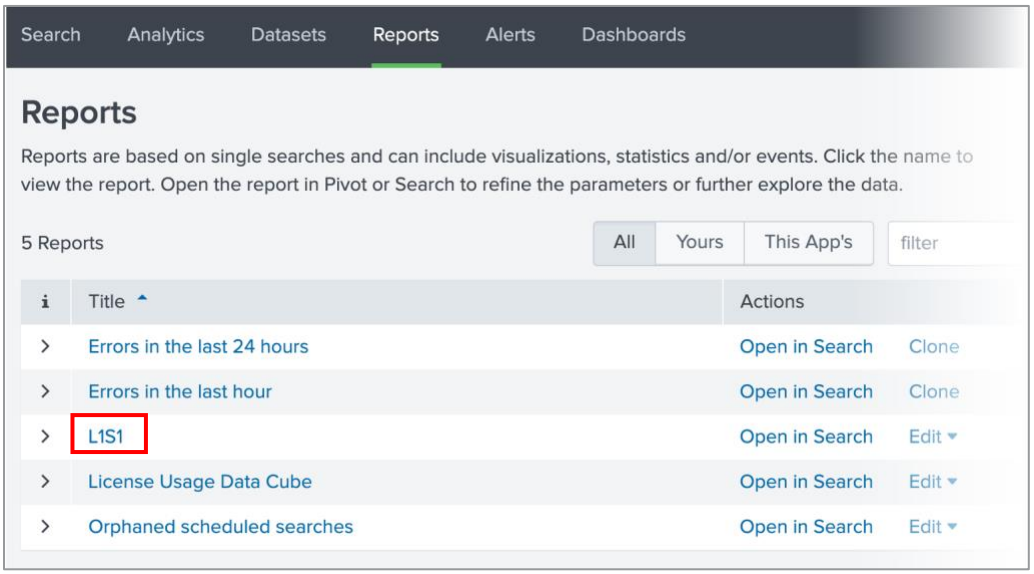


23. Navigate to **Settings** > **Report acceleration summaries**. You should see your report listed. Under **Summary Status**, you will see how much of your summary has been built. (Note: The searches that build report acceleration summaries are run every 10 minutes at :00, :10, :20, etc.)

24. Report acceleration summaries can take time to complete. An accelerated report has been created for you to use called **allStudents_Sales_Report_MonthlyOnlineSalesRevenue**. Click on the report title under **Reports Using Summary** column. This will take you back to the **Searches, Reports, and Alerts** page. Dismiss the **Edit Search** window by clicking the **X** in the upper right-hand corner.
25. Under **Actions**, click **Run**.

26. You should see the search in the search window and under **Job** you'll see a message indicating that Splunk is using summaries for your search.

27. Click **Job** > **Inspect Job** and note how long the search took to complete. (When this screenshot was taken, the accelerated report took 0.226 seconds to complete. This is about 85% faster!)
28. Save your search as a report with the name **L1S1**.
    a. Click **Save As** > **Report**
    b. For **Title**, enter L1S1.
    c. Click **Save**.
    d. You can **View** your report or exit out of the **Your Report Has Been Created** window by clicking the **X** in the upper-right corner.
    e. You can access your saved reports using the **Reports** tab in the application bar.



*Your recently saved **L1S1** report will be visible in the **Reports** tab.*

Lab Exercise 2 – Data Model Acceleration

Description

Use the **`datamodel`** command to explore unsummarized and summarized data within a specific data model.

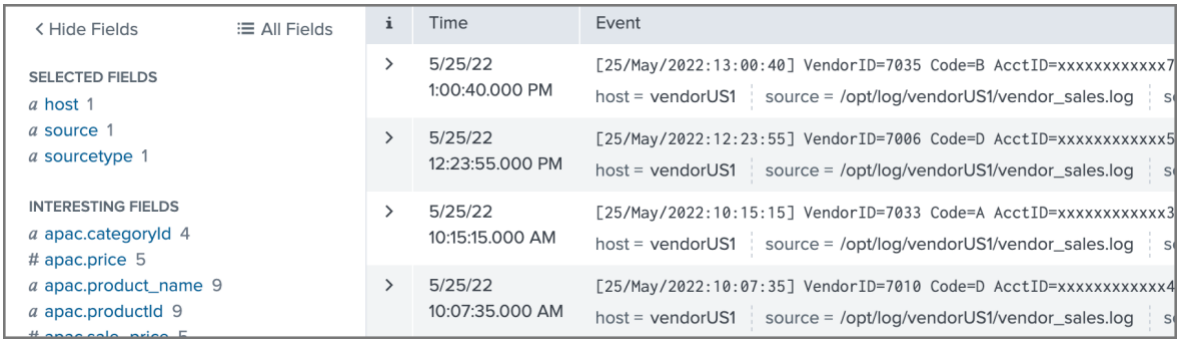
Steps

Scenario: SalesOps wants a listing of the APAC vendors with retail sales of more than $200 over the previous week.

Task 1: Search and transform summarized data in the Vendor Sales data model.

1. Use the **`datamodel`** command to view all data models you have access to.
   ```
   | datamodel
   ```
2. Use the **`datamodel`** command to browse only the Vendor Sales data model. (Hint: You must provide the modelName as an argument to the **`datamodel`** command.)
   ```
   | datamodel vsales
   ```

3. Revise your search to display the events in the APAC dataset. Set your time range to the **Previous week**. (Hint: Remember that when using the **`datamodel`** command, datasets are referred to as "objects".)
   ```
   | datamodel vsales apac search
   ```



4. Look at the **Interesting Fields** sidebar. Notice how all the fields start with **`apac`**. Revise your search so that your fields no longer start with **`apac`** and you are still able to search the events.
   ```
   | datamodel vsales apac flat
   ```

5. Append the following pipes to your search string to find the APAC vendors with retail sales over $200 from the previous week.

```
| stats sum(price) as Sales by Vendor, VendorCountry, VendorCity
| search Sales > 200
```

Vendor	VendorCountry	VendorCity	Sales
Ahimsa Games	India	Raipur	245.88
Geppetto's Toys	Australia	Melbourne	328.87
Golden Games	China (PRC)	Tianjin	277.88
Happy Fun Games	Japan	Sapporo	282.88
House of Diversions	China (PRC)	Haikou	249.87
House of Diversions	China (PRC)	Nanjing	466.81

6. Save your search as a report with the name **L2S1**.

Task 2: Confirm that events are being summarized every 5 minutes.

7. Open a second **Search** tab by right-clicking on **Search** in the application bar and choosing **Open Link in New Tab**.
   a. Then, copy and paste the search containing **`summariesonly=false`** in one search window and the search containing **`summariesonly=true`** in the other search window.

```
| datamodel AccButtercup_Games_Online_Sales search summariesonly=false
| datamodel AccButtercup_Games_Online_Sales search summariesonly=true
```

   b. Run each search over the **Last 5 minutes** by using the **Relative** tab of the **Time Range Picker**.
8. Observe your results. Do the searches have the same number of events? If not, why?
   Data model acceleration summaries are updated every 5 minutes. Therefore, the search containing **`summariesonly=true`** should have fewer events with the latest timestamp occurring before a time ending in :00, :05, :10, :15, etc. The search containing **`summariesonly=false`** should have more events because it is retrieving all events from the last 5 minutes.

   If your event count is the same, you likely ran your search right after the data model acceleration summaries were updated. Wait a minute or two and run your searches again. The search containing **`summariesonly=false`** should have additional events but the search containing **`summariesonly=true`** should now have fewer events.

Lab Exercise 3 – Using the `tstats` Command

Description

Use the **`tstats`** command to quickly search a large amount of data and to create a speedy report using **`tstats`** on the **`tsidx`** files of an accelerated data model.

Steps

Scenario: ITOps wants to determine the number of events Splunk is indexing per month to verify there will be adequate indexing volume in the future.

Task 1: Display the number of indexed events by month for the last year to date with the number and time formatted.

1. Count all events in index **`tsidx`** files over **All time**. Label the count as "events."

```
| tstats count as events
```

events
300819

2. Split the search by time with a **`span`** of one month. Sort in descending order by time. (Note: The student environments contain approximately 3 – 4 months of data.)

```
| tstats count as events by _time span=1mon
| sort - _time
```

_time	events
2022-05	125791
2022-04	84340
2022-03	77063
2022-02	13675

NOTE: The following step is optional and requires knowledge of the time and date functions of the **`eval`** command. Continue to step 4 to save your search as a report.

3. Use the **`eval`** command to create a "Month" field that contains the **`_time`** values in the format "Month YYYY". In the same pipe, format the **`events`** values to include commas. Then pipe to a **`table`** command to display **`Month`** and **`events`**.

```
| tstats count as events by _time span=1mon
| sort - _time
| eval Month = strftime(_time,"%B %Y"), events = tostring(events,"commas")
| table Month, events
```

Month	events
May 2022	125,920
April 2022	84,340
March 2022	77,063
February 2022	13,675

4. Save your search as a report with the name **L3S1**.

Scenario: Complete the scenario request from L2S1 but use the `tstats` command instead.

Task 2: Use `tstats` to create a report from the summarized data from the APAC dataset of the Vendor Sales data model that will show retail sales of more than $200 over the previous week.

5. Use the **`tstats`** command on the **`apac`** dataset of the **`vsales`** datamodel to calculate the sum of **`apac.price`** as "Sales" by **`apac.Vendor`**, **`apac.VendorCountry`**, and **`apac.VendorCity`**. Search over the **Previous week**.

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
```

apac.Vendor	apac.VendorCountry	apac.VendorCity	Sales
Ahimsa Games	India	Raipur	188.87
Bapu's Hobbies	India	Puducherry	329.85
Cinematic Games	India	Mumbai	88.96
Devilish Diversions	Australia	Hobart	288.88
EuroToys Emporium	Armenia	Yerevan	119.94
Falk's Toy Store	Sri Lanka	Colombo	204.86
Games Down Under	Australia	Sydney	105.93

6. Display only vendors with more than $200 in sales by piping results to **`search Sales > 200`**.

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
| search Sales > 200
```

apac.Vendor	apac.VendorCountry	apac.VendorCity	Sales
Geppetto's Toys	Australia	Melbourne	211.88
Golden Games	China (PRC)	Beijing	445.80
Ham's House of Fantastic Fun	South Korea	Hongseong	209.90
Happy Fun Games	Japan	Hiroshima	608.75
Happy Fun Games	Japan	Kyoto	284.87
Happy Fun Games	Japan	Sapporo	250.90
House of Diversions	China (PRC)	Haikou	210.90

7. Rename **apac.Vendor** as "Vendor", **apac.VendorCountry** as "Country", and **apac.VendorCity** as "City."

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
| search Sales > 200
| rename apac.Vendor as Vendor, apac.VendorCountry as Country, apac.VendorCity as
  City
```

This rename expression will also work:

```
| rename apac.Vendor as Vendor, apac.Vendor* as *
```

Vendor	Country	City	Sales
Geppetto's Toys	Australia	Melbourne	211.88
Golden Games	China (PRC)	Beijing	445.80
Ham's House of Fantastic Fun	South Korea	Hongseong	209.90
Happy Fun Games	Japan	Hiroshima	608.75
Happy Fun Games	Japan	Kyoto	284.87
Happy Fun Games	Japan	Sapporo	250.90

8. Sort results by **Country**, **Vendor**, **City**.

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
| search Sales > 200
| rename apac.Vendor as Vendor, apac.Vendor* as *
| sort Country, Vendor, City
```

Vendor	Country	City	Sales
Geppetto's Toys	Australia	Melbourne	211.88
Golden Games	China (PRC)	Beijing	445.80
House of Diversions	China (PRC)	Haikou	210.90
House of Diversions	China (PRC)	Hangzhou	223.88
House of Diversions	China (PRC)	Nanjing	248.87
House of Diversions	China (PRC)	Syanley	288.87

NOTE: Steps 9 and 10 are optional and require knowledge of the **eval** and **stats** commands. You can skip these steps and continue to step 11 to save your search as a report.

9. Use the **eval** command to format **Sales** so that the values start with a "$" and have commas.

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
| search Sales > 200
| rename apac.Vendor as Vendor, apac.Vendor* as *
| sort Country, Vendor, City
| eval Sales = "$".tostring(Sales,"commas")
```

Vendor	Country	City	Sales
Geppetto's Toys	Australia	Melbourne	$211.88
Golden Games	China (PRC)	Beijing	$445.80
House of Diversions	China (PRC)	Haikou	$210.90
House of Diversions	China (PRC)	Hangzhou	$223.88
House of Diversions	China (PRC)	Nanjing	$248.87
House of Diversions	China (PRC)	Syanley	$288.87

10. Improve your table by listing **City** and **Sales** values by **Vendor** and **Country**. The resulting table should have the columns **Vendor**, **Country**, **City**, **Sales**.

```
| tstats sum(apac.price) as Sales from datamodel=vsales.apac by apac.Vendor,
  apac.VendorCountry, apac.VendorCity
| search Sales > 200
| rename apac.Vendor as Vendor, apac.Vendor* as *
| sort Country, Vendor, City
| eval Sales = "$".tostring(Sales,"commas")
| stats list(City) as City, list(Sales) as Sales by Vendor, Country
```

Vendor	Country	City	Sales
Geppetto's Toys	Australia	Melbourne	$211.88
Golden Games	China (PRC)	Beijing	$445.80
Ham's House of Fantastic Fun	South Korea	Hongseong	$209.90
Happy Fun Games	Japan	Hiroshima Kyoto Sapporo	$608.75 $284.87 $250.90
House of Diversions	China (PRC)	Haikou Hangzhou Nanjing Syanley	$210.90 $223.88 $248.87 $288.87

11. Save your search as a report with the name **L3S2**.