

Developer Console for Admins

It's not that scary



SOUTHEAST
DREAMIN'



Patrick Connelly

Principal Software Engineer, Red Hat



SOUTHEAST
DREAMIN'

 **CapStorm**  **formstack**

sumo
SCHEDULER

 **valence**
NATIVE SALESFORCE INTEGRATION

 **Azimuth DS**
LLC
your Salesforce navigation partner

 **metaZOO**

 **PDF
BUTLER**

 **KnowledgeNet.ai**

apsona 
Simple. Reliable. Effective.

 **FormAssembly™**

 **HIKE2**

 **Tricentis™**

 **REVENUEGRID**

 **SOUTHEAST
DREAMIN'**

What is the Developer Console?

Query Editor

- View data and fields that aren't on a layout
- Modify records
- Insert records
- Go directly to view / edit pages
- Workbench is off platform

Quick SOQL Primer

SELECT field1__c, field2__c

FROM ObjectName

WHERE <criteria>

ORDER BY <criteria>

LIMIT 100

Useful SOQL Functions

- COUNT
- MIN \ MAX
- AVG
- SUM
- THIS_YEAR
- THIS_MONTH
- LAST_N_MONTHS:n
- DAY_IN_YEAR
- CALENDAR_YEAR

Report Performance

When reports are written poorly and run poorly people tend to think that the system is “just slow.”

How to Read the Query Plan

- **Cost** - Higher the number the more time it takes
- **Leading Operation Type** - *Index* is fastest, *TableScan* is the slowest

Report Performance

Preferences ✕	
Name ▲	Value
Editor Font Size	18
Editor Theme	default
Enable Apex Autocomplete	true
Enable Query Plan	false
Enable Visualforce View State Inspector	true
Prettier: Indent with Tabs	false
Prettier: Print Width	80
Prettier: Tab Width	2
Prevent Logs on Load	false

Report Performance

Query Plan					
Cardinality	Fields	Leading Operation Type	Cost	sObject Cardinality	sObject Type
1		TableScan	1.21666...	13	Account
1	Name	Other	1.21666...	13	Account
13		TableScan	3.81666...	13	User
Notes: Not considering filter for optimization because unindexed. Table: Account Fields: ["IsDeleted"]					
Close					

Report Performance

Query Plan					
Cardinality	Fields	Leading Operation Type	Cost	sObject Cardinality	sObject Type
1		TableScan	1.21666...	13	Account
13		TableScan	3.81666...	13	User
12	OwnerId	Index	4	13	Account
Notes: Not considering filter for optimization because unindexed. Table: Account Fields: ["IsDeleted"] Not considering filter for optimization because unindexed. Table: Account Fields: ["Industry"]					
Close					

Report Performance

Query Plan					
Cardinality	Fields	Leading Operation Type	Cost	sObject Cardinality	sObject Type
8		TableScan	2.73333...	13	Account
13		TableScan	3.81666...	13	User
Notes: Not considering filter for optimization because unindexed. Table: Account Fields: ["IsDeleted"] Not considering filter for optimization because unindexed. Table: Account Fields: ["Is_a_Gas_Company__c"]					
Close					

List View Performance

Query Plan					
Cardinality	Fields	Leading Operation Type	Cost	sObject Cardinality	sObject Type
12		TableScan	3.6	13	Account
12	OwnerId	Index	4	13	Account
13		TableScan	6.63333...	13	User
Notes: Not considering filter for optimization because unindexed. Table: Account Fields: ["IsDeleted"]					
Close					

Logging

- Useful for debugging issues
- Data can be saved and sent to integrators

EmailMissionSpecialist.apxc
Log executeAnonymous @7/22/2016, 5:52:06 PM

Execution Log

Timestamp	Event	Details
17:52:06:006	HEAP_ALLOCATE	[13] Bytes:151
17:52:06:006	METHOD_EXIT	[2] 01p50000001H3vF EmailMissionSpecialist.sendMail(String, String, String
17:52:06:006	SYSTEM_MODE_...	false
17:52:06:006	FATAL_ERROR	System.EmailException: SendEmail failed. First exception on row 0; first err
17:52:06:000	FATAL_ERROR	
17:52:06:000	FATAL_ERROR	Class.EmailMissionSpecialist.sendMail: line 13, column 1
17:52:06:000	FATAL_ERROR	AnonymousBlock: line 2, column 1
17:52:06:000	FATAL_ERROR	AnonymousBlock: line 2, column 1
17:52:06:006	CUMULATIVE_LI...	
17:52:06:006	LIMIT_USAGE_F...	(default)

☐ This Frame
☐ Executable
☐ Debug Only
☐ Filter

Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems

User	Application	Operation	Time	Status	Read	Size
Your Name	Unknown	/services/da...	7/22/2016, 5...	SendEmail f...		5.53 KB
Your Name	Unknown	/services/da...	7/22/2016, 5...	Success		7.38 KB

Testing

- View test coverage percentage
- Run test suites

Logs Tests Checkpoints Query Editor View State Progress Problems									
Status	Test Run	Enqueued Time	Duration	Failures	Total		Overall Code Coverage		
✖	707R000000056dH	Tue Dec 15 2015 23:28:28 GMT...		0	2		Class	Percent	Lines
✖	AwesomeClassTest			1	4		Account_After_Insert_Trg	100%	2/2
✖	testAFailingTest		0:00				AwesomeClass	100%	3/3
✔	testMyAwesomeMethod		0:00				ComplexBatch	0%	0/9
✔	testMyAwesomeMethodIfFalse		0:00				SimpleQueueable	0%	0/8
✔	testMyAwesomeMethodIfTrue		0:00				TestUtil	0%	0/25
✔	SampleClassTest			0	2		Util	0%	0/4

More Reading

Available at <https://connelly.dev/presentations>

- [SOQL Reference](#)
- [SOQL Trailhead](#)
- [Query Plan FAQ](#)
- [Improving Report Performance](#)
- [Logs Tab](#)
- [Generate and Analyze Logs](#)
- [Tests Tab](#)
- [Checking Code Coverage](#)
- [Aggregate Functions](#)
- [Date Functions](#)
- [Date Literals](#)