

ActiveCampaign API Data Model

for use with Invantive SQL

23.0

Copyright

(C) Copyright 2004-2023 Invantive Software B.V., the Netherlands. All rights reserved.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Despite all the care taken in the compilation of this text, neither the author nor the publisher can accept liability for any damage, which might result from any error, which might appear in this publication.

This manual is a reference guide intended to clarify usage. If data in the sample images match data in your system, the similarity is coincidental.

Important Safety and Usage Information

Intended Use and Limitations: This software, developed by Invantive, is designed to support a variety of business and information technology data processing functions, such as accounting, financial reporting and sales reporting. It is important to note that this software is not designed, tested, or approved for use in environments where malfunction or failure could lead to life-threatening situations or severe physical or environmental damage. This includes, but is not limited to:

- **Nuclear facilities:** The software should not be used for operations or functions related to the control, maintenance, or operation of nuclear facilities.
- **Defense and Military Applications:** This software is not suitable for use in defense-related applications, including but not limited to weaponry control, military strategy planning, or any other aspects of national defense.
- **Aviation:** The software is not intended for use in the operation, navigation, or communication systems of any aircraft or air traffic control environments.
- **Healthcare and Medicine Production:** This software should not be utilized for medical device operation, patient data analysis for critical health decisions, pharmaceutical production, or medical research where its failure or malfunction could impact patient health.
- **Chemical and Hazardous Material Handling:** This software is not intended for the management, control, or operational aspects of chemical plants or hazardous material handling facilities. Any malfunction in software used in these settings could result in dangerous chemical spills, explosions, or environmental disasters.
- **Transportation and Traffic Control Systems:** The software should not be used for the control, operation, or management of transportation systems, including railway signal controls, subway systems, or traffic light management. Malfunctions in such critical systems could lead to severe accidents and endanger public safety.
- **Energy Grid and Utility Control Systems:** This software is not designed for the control or operation of energy grid systems, including electrical substations, renewable energy control systems, or water utility control systems. The failure of software in these areas could lead to significant power outages, water supply disruptions, or other public utility failures, potentially endangering communities and causing extensive damage.
- **Other High-Risk Environments:** Any other critical infrastructure and environments where a failure of the software could result in significant harm to individuals or the environment.

User Responsibility: Users must ensure that they understand the intended use of the software and refrain from deploying it in any setting that falls outside of its designed purpose. It is the responsibility of the user to assess the suitability of the software for their intended application, especially in any scenarios that might pose a risk to life, health, or the environment.

Disclaimer of Liability: Invantive disclaims any responsibility for damage, injury, or legal consequences resulting from the use or misuse of this software in prohibited or unintended applications.

Contents

1	SQL Driver for ActiveCampaign API	1
2	SQL Driver Attributes for ActiveCampaign API	2
3	Schema: Native	14
3.1	Tables	14
3.1.1	NATIVEPLATFORMSCALARREQUESTS: ActiveCampaign Native Platform Scalar Requests	14
4	Schema: V3	16
4.1	Tables	16
4.1.1	AccountById: ActiveCampaign Account by ID	16
4.1.2	AccountContactsByContactId: ActiveCampaign Account Contacts by Contact ID	17
4.1.3	AccountCustomFields: ActiveCampaign Account Custom Fields	18
4.1.4	AccountCustomFieldValues: ActiveCampaign Account Custom Field Values	18
4.1.5	Accounts: ActiveCampaign Accounts	19
4.1.6	AccountWithCustomFieldValues: ActiveCampaign Account Custom Fields	19
4.1.7	Addresses: ActiveCampaign Addresses	20
4.1.8	Associations: ActiveCampaign Account Contacts	21
4.1.9	AutomationEntryCountsByContactId: ActiveCampaign Automation Entry Counts by Contact ID	21
4.1.10	Automations: ActiveCampaign Automations	22
4.1.11	BounceLogsByContactId: ActiveCampaign Bounce Logs by Contact ID	23
4.1.12	Brandings: ActiveCampaign Brandings	24
4.1.13	Calendars: ActiveCampaign Calendars	25
4.1.14	Campaigns: ActiveCampaign Campaigns	25
4.1.15	Connections: ActiveCampaign Connections	28
4.1.16	ContactAutomations: ActiveCampaign Contact Automations	29
4.1.17	ContactAutomationsByContactId: ActiveCampaign Contact Automations by Contact ID	29
4.1.18	ContactDataByContactId: ActiveCampaign Contact Data by Contact ID	31
4.1.19	ContactDealsByContactId: ActiveCampaign Deals by Contact ID	31
4.1.20	ContactGoalsByContactId: ActiveCampaign Goals by Contact ID	32
4.1.21	ContactListsByContactId: ActiveCampaign Lists by Contact ID	33
4.1.22	ContactLogsByContactId: ActiveCampaign Logs by Contact ID	35
4.1.23	Contacts: ActiveCampaign Contacts	36
4.1.24	ContactTagsByContactId: ActiveCampaign Tags by Contact ID	38
4.1.25	ContactWithBounceLogs: ActiveCampaign Contact Bounce Logs	39
4.1.26	ContactWithCustomFieldValues: ActiveCampaign Contact Custom Fields	39
4.1.27	ContactWithTags: ActiveCampaign Contact Tags	40
4.1.28	CustomFields: ActiveCampaign Custom Fields	40
4.1.29	CustomFieldValues: ActiveCampaign Custom Field Values	41
4.1.30	DealCustomFields: ActiveCampaign Deal Custom Fields	42
4.1.31	DealCustomFieldValues: ActiveCampaign Deal Custom Field Values	42
4.1.32	Deals: ActiveCampaign Deals	43
4.1.33	DealsByContactId: ActiveCampaign Deals by Contact ID	44
4.1.34	DealWithActivities: ActiveCampaign Deal Activities	45
4.1.35	DealWithContacts: ActiveCampaign Deal Contacts	46
4.1.36	DealWithNotes: ActiveCampaign Deal Notes	47
4.1.37	DealWithTasks: ActiveCampaign Deal Tasks	48
4.1.38	eComCustomers: ActiveCampaign eCommerce Customers	49
4.1.39	eComOrderProducts: ActiveCampaign eCommerce Order Products	50
4.1.40	eComOrders: ActiveCampaign eCommerce Orders	50
4.1.41	Events: ActiveCampaign Events	51

4.1.42	FieldValuesByContactId: ActiveCampaign Field Values by Contact ID	52
4.1.43	Forms: ActiveCampaign Forms	53
4.1.44	GeolpsByContactId: ActiveCampaign Geo IP-addresses by Contact ID	54
4.1.45	Groups: ActiveCampaign Groups	55
4.1.46	Lists: ActiveCampaign Lists	57
4.1.47	Messages: ActiveCampaign Messages	59
4.1.48	NotesByContactId: ActiveCampaign Notes by Contact ID	59
4.1.49	Pipelines: ActiveCampaign Pipelines	61
4.1.50	SavedResponses: ActiveCampaign Saved Responses	61
4.1.51	Scores: ActiveCampaign Scores	62
4.1.52	ScoreValuesByContactId: ActiveCampaign Score Values by Contact ID	62
4.1.53	SecondaryContacts: ActiveCampaign Secondary Contacts	63
4.1.54	Segments: ActiveCampaign Segments	64
4.1.55	SiteTracking: ActiveCampaign Site Tracking	64
4.1.56	SiteTrackingWhitelistedDomains: ActiveCampaign Site Tracking Domains	65
4.1.57	Stages: ActiveCampaign Stages	65
4.1.58	Tags: ActiveCampaign Tags	66
4.1.59	Tasks: ActiveCampaign Tasks	66
4.1.60	TaskTypes: ActiveCampaign Task Types	67
4.1.61	TrackEvents: ActiveCampaign Track Event	67
4.1.62	TrackingLogsByContactId: ActiveCampaign Tracking Logs by Contact ID	68
4.1.63	Users: ActiveCampaign Users	69
4.1.64	WebhookEvents: ActiveCampaign Webhook Events	70
4.1.65	Webhooks: ActiveCampaign Webhooks	70

Index

71

1 SQL Driver for ActiveCampaign API

Invantive SQL is the fastest, easiest and most reliable way to exchange data with the ActiveCampaign API.

Use the "Search" option in the left menu to search for a specific term such as the table or column description. When you already know the term, please use the "Index" option. When you can't find the information needed, please click on the Chat button at the bottom or place your question in the [user community](#). Invantive Support or other users will try to help you.

ActiveCampaign is cloud software to automate the marketing, sales and support process of prospects, leads and customers. ActiveCampaign includes email campaigns, sales management, website integration, chat and advanced automatic workflows.

The ActiveCampaign driver covers 66 tables and 930 columns.

ActiveCampaign API Clients

Invantive SQL is available on many user interfaces ("clients" in traditional server-client paradigm). All Invantive SQL statements can be exchanged with a close to 100% compatibility across all clients and operating systems (Windows, MacOS, Linux, iOS, Android).

The clients include Microsoft Excel, Microsoft Power BI, Microsoft Power Query, Microsoft Word and Microsoft Outlook. Web-based clients include Invantive Cloud, Invantive Bridge Online as OData proxy, Invantive App Online for interactive apps, Online SQL Editor for query execution and Invantive Data Access Point as extended proxy.

The [ActiveCampaign Power BI connector](#) is based on the Invantive SQL driver for ActiveCampaign, completed by a high-performance OData connector which works straight on Power BI without any add-on. The OData protocol is always version 4, independent whether the backing platform uses OData, SOAP or another protocol.

For technical users there are command-line editions of Invantive Data Hub running on iOS, Android, Windows, MacOS and Linux. Invantive Data Hub is also often used for enterprise server applications such as ETL. High-volume replication of data taken from the ActiveCampaign API into traditional databases such as SQL Server (on-premises and Azure), MySQL, PostgreSQL and Oracle is possible using [Invantive Data Replicator](#). Invantive Data Replicator automatically creates and maintains ActiveCampaign datawarehouses, possibly in combination with data from over 75 other (cloud) platforms. Invantive Data Replicator supports data volumes up to over 1 TB and over 5.000 companies. The on-premise edition of Invantive Bridge offers an ActiveCampaign ADO.net provider.

Finally, online web apps can be build for ActiveCampaign using App Online of [Invantive Cloud](#).

Monitor API Calls

When a query or DML-statement has been executed on Invantive SQL a developer can evaluate the actual calls made to the ActiveCampaign API using a query on sessionios@DataDictionary. As an alternative, extensive request and response logging can be enabled by setting log-native-calls-to-disk to true. In the %USERPROFILE%\Invantive\NativeLog folder Invantive SQL will create log files per ActiveCampaign API request and response.

Specifications

The SQL driver for ActiveCampaign does not support partitioning. Define one data container in a database for each company in ActiveCampaign to enable parallel access for data from multiple companies.

An introduction into the concepts of Invantive SQL such as databases, data containers and partitioning can be found in the [Invantive SQL grammar](#).

The configuration can be changed using various attributes from the database definition, on log on and during use. A full list of configuration options is listed in the [driver attributes](#).

The catalog name is used to compose the full qualified name of an object like a table or view. The schema name is used to compose the full qualified name of an object like a table or view. On ActiveCampaign the comparison of two texts is case sensitive by default.

Changes and bug fixes on the ActiveCampaign SQL driver can be found in the [release notes](#). Get access to the community through the [ActiveCampaign section](#) of the Invantive forums.

Driver code for use in settings.xml: ActiveCampaign

Alias: acn

Recommended alias: acn

Driver code for use in settings.xml

Authentication can be done using using a token. An API token can be generated from the ActiveCampaign menu: 'Settings' > 'Developer' > 'Key', as described on <https://developers.activecampaign.com/reference#authentication>.

Updated 22-12-2022 20:43 using Invantive SQL version 22.1.101-BETA+3681.

2 SQL Driver Attributes for ActiveCampaign API

The SQL driver for ActiveCampaign has many attributes that can be finetuned to improve handling in scenarios with unreliable network connections to the API server of ActiveCampaign or high volumes of data. Also, many drivers have driver-specific attributes to finetune actual behaviour or handle data not matching specifications.

The ActiveCampaign driver attributes are assigned a default value which seldom requires change. However, changes can be applied when needed on four levels, which are reflected in the table below by separate checkmarks:

- Connection string: the connection string from the settings*.xml file and applied during log on.
- Set SQL statement: a set SQL-statement to be executed once connection has been established.
- Log on: value to be specified interactively by user during log on in a user interface.

The connection string for ActiveCampaign can be found in the settings*.xml file used for the database. The reference manuals contain instructions how to relocate the settings*.xml files. Settings*.xml files are typically located in the %USERPROFILE%\invantive folder in most deployment scenarios. Each data container of a database in the connection string can have a `connectionString` element specifying the name and values of attributes. Both name and value must be properly escaped according to XML-semantics. Actual application of the value is solely done during log on. A new connection must be established to change the value of a driver attribute using a connection string.

The set SQL statement can be executed after log on. The syntax is: `set NAME VALUE`, or for a distributed database: `set NAME@ALIAS VALUE`. In some scenarios you may need to enclose the driver attribute name in square brackets to escape it from parsing, for instance when a reserved SQL keyword is part of the name. The new value takes effect straight after execution of the set-statement. The set-statement can be executed as often as needed during a session.

Driver attributes that can be interactively set to a value are typically presented in the log on window. Depending on the platform and design decisions of the user interface designer, some or all of the available driver attributes can have been made available.

The ActiveCampaign driver can be configured using the following attributes:

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
account-id	Account ID to use for registering events.	ActiveCampaign		✓		✓	✓
account-name	Name of your company preceding 'activehosted.com' in the browser.	ActiveCampaign		✓		✓	✓
add-odata-mandatory-filters	Whether to automatically add OData filters deemed necessary by the platform.	OData	False	✓	✓	✓	
analysis-enforce-row-uniqueness	Enforce rows to be unique for software analysis.	Shared	False	✓	✓	✓	
api-url	URL of web service.	OData		✓		✓	
bulk-delete-page-size-rows	Number of rows to delete per batch when bulk deleting.	Shared	10000	✓	✓	✓	
bulk-insert-page-size-bytes	Approximate maximum size in bytes of batch when bulk inserting.	Shared	10000000	✓	✓	✓	
bulk-insert-page-size-rows	Number of rows to insert per batch when bulk inserting.	Shared	250	✓	✓	✓	
download-error-400-bad-request-max-tries	Maximum number of tries when HTTP server reports bad format during retrieval of data.		3	✓	✓	✓	
download-error-400-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		500	✓	✓	✓	
download-error-400-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		5000	✓	✓	✓	
download-error-400-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
download-error-408-request-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 408.		10	✓	✓	✓	
download-error-408-request-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 408.		10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
dow nload-error-408-request-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 408.		300000	✓	✓	✓	
dow nload-error-408-request-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 408.		2	✓	✓	✓	
dow nload-error-422-bad-request-max-tries	Maximum number of tries when HTTP server reports unprocessable entity during retrieval of data.		30	✓	✓	✓	
dow nload-error-422-bad-request-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		10000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports unprocessable entity during retrieval of data.		300000	✓	✓	✓	
dow nload-error-422-bad-request-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports unprocessable entity during retrieval of data.		2	✓	✓	✓	
dow nload-error-429-too-many-requests-max-tries	Maximum number of tries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		10000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		300000	✓	✓	✓	
dow nload-error-429-too-many-requests-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports that too many requests have been made during a timeslot of one minute or one day.		2	✓	✓	✓	
dow nload-error-502-server-unavailable-max-tries	Maximum number of tries when HTTP server reports a bad gateway during retrieval of data.		30	✓	✓	✓	
dow nload-error-502-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports a bad gateway during retrieval of data.		10000	✓	✓	✓	
dow nload-error-502-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that a bad gateway during retrieval of data.		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
dow nload-error-502-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports a bad gateway during retrieval of data.		2	✓	✓	✓	
dow nload-error-503-server-unavailable-max-tries	Maximum number of tries when HTTP server reports that the API server is unavailable during retrieval of data.		30	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-initial-ms	Initial sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		10000	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-max-ms	Maximum sleep in milliseconds between retries when HTTP server reports that the API server is unavailable during retrieval of data.		300000	✓	✓	✓	
dow nload-error-503-server-unavailable-sleep-multiplicator	Multiplication factor for sleep between retries HTTP server reports that the API server is unavailable during retrieval of data.		2	✓	✓	✓	
dow nload-error-504-gateway-timeout-max-tries	Maximum number of tries when the website reports a gateway timeout.		10	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a gateway timeout.		10000	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a gateway timeout.		300000	✓	✓	✓	
dow nload-error-504-gateway-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a gateway timeout.		2	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 590.		10	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 590.		10000	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 590.		300000	✓	✓	✓	
dow nload-error-590-netw ork-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 590.		2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
download-error-599-network-connect-timeout-max-tries	Maximum number of tries when the website reports a HTTP status 599.		10	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the website reports a HTTP status 599.		10000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the website reports a HTTP status 599.		300000	✓	✓	✓	
download-error-599-network-connect-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the website reports a HTTP status 599.		2	✓	✓	✓	
download-error-argument-exception-max-tries	Maximum number of tries when an argument exception is returned when downloading a blob.		10	✓	✓	✓	
download-error-argument-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		10000	✓	✓	✓	
download-error-argument-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an argument exception is returned when downloading a blob.		300000	✓	✓	✓	
download-error-argument-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an argument exception is returned when downloading a blob.		2	✓	✓	✓	
download-error-internet-download-max-tries	Maximum number of tries when the Internet connection seems down during retrieval of data.		10	✓	✓	✓	
download-error-internet-download-sleep-initial-ms	Initial sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		10000	✓	✓	✓	
download-error-internet-download-sleep-max-ms	Maximum sleep in milliseconds between retries when the Internet connection seems down during retrieval of data.		300000	✓	✓	✓	
download-error-internet-download-sleep-multiplicator	Multiplication factor for sleep between retries when the Internet connection seems down during retrieval of data.		2	✓	✓	✓	
download-error-io-exception-max-tries	Maximum number of tries when a network I/O connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-io-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a network I/O		10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
	connection failure occurs during retrieval of data.						
dow nload-error-io-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a network I/O connection failure occurs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-io-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a network I/O connection failure occurs during retrieval of data.		2	✓	✓	✓	
dow nload-error-json-exception-max-tries	Maximum number of tries when an invalid JSON body is returned.		3	✓	✓	✓	
dow nload-error-json-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an invalid JSON body is returned.		1000	✓	✓	✓	
dow nload-error-json-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an invalid JSON body is returned.		10000	✓	✓	✓	
dow nload-error-json-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an invalid JSON body is returned.		2	✓	✓	✓	
dow nload-error-other-exception-max-tries	Maximum number of tries when an unqualified error occurs during retrieval of data.		3	✓	✓	✓	
dow nload-error-other-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		10000	✓	✓	✓	
dow nload-error-other-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when an unqualified error occurs during retrieval of data.		300000	✓	✓	✓	
dow nload-error-other-exception-sleep-multiplicator	Multiplication factor for sleep between retries when an unqualified error occurs during retrieval of data.		2	✓	✓	✓	
dow nload-error-socket-exception-max-tries	Maximum number of tries when the network connection is forcibly dropped during retrieval of data.		10	✓	✓	✓	
dow nload-error-socket-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when the network connection is forcibly dropped during retrieval of data.		10000	✓	✓	✓	
dow nload-error-socket-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when the network connection is forcibly dropped during retrieval of data.		300000	✓	✓	✓	
dow nload-error-socket-exception-sleep-multiplicator	Multiplication factor for sleep between retries when the network connection is forcibly dropped during retrieval of data.		2	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
download-error-web-exception-max-tries	Maximum number of tries when a web connection failure occurs during retrieval of data.		10	✓	✓	✓	
download-error-web-exception-sleep-initial-ms	Initial sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		10000	✓	✓	✓	
download-error-web-exception-sleep-max-ms	Maximum sleep in milliseconds between retries when a web connection failure occurs during retrieval of data.		300000	✓	✓	✓	
download-error-web-exception-sleep-multiplicator	Multiplication factor for sleep between retries when a web connection failure occurs during retrieval of data.		2	✓	✓	✓	
download-error-web-not-implemented-max-tries	Maximum number of tries when the connection reports not implemented.		1	✓	✓	✓	
download-error-web-not-implemented-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports not implemented.		10000	✓	✓	✓	
download-error-web-not-implemented-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports not implemented.		300000	✓	✓	✓	
download-error-web-not-implemented-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports not implemented.		2	✓	✓	✓	
download-error-web-timeout-max-tries	Maximum number of tries when the connection reports a timeout.		10	✓	✓	✓	
download-error-web-timeout-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports a timeout.		1000	✓	✓	✓	
download-error-web-timeout-sleep-max-ms	Maximum sleep in milliseconds between retries when the connection reports a timeout.		30000	✓	✓	✓	
download-error-web-timeout-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports a timeout.		2	✓	✓	✓	
download-error-web-unauthorized-max-tries	Maximum number of tries when the connection reports an unauthorized error.		1	✓	✓	✓	
download-error-web-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between retries when the connection reports an unauthorized error.		10000	✓	✓	✓	
download-error-web-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between retries when the		300000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
sleep-max-ms	connection reports an unauthorized error.						
download-error-web-unauthorized-sleep-multiplicator	Multiplication factor for sleep between retries when the connection reports an unauthorized error.		2	✓	✓	✓	
event-key	Key to use for registering events.	ActiveCampaign		✓		✓	✓
force-case-sensitive-identifiers	Consider identifiers as case-sensitive independent of the platform capabilities.	Shared	False	✓	✓	✓	
forced-casing-identifiers	Forced casing of identifiers. Choose from: Unset, Lower, Upper and Mixed.	Shared		✓	✓	✓	
http-disk-cache-compression-level	Compression level for the HTTP disk cache, ranging from 1 (little) to 9 (intense). Default is 5.	Shared	5	✓	✓	✓	
http-disk-cache-directory	Directory where HTTP cache is stored.	Shared	C:\Users\gle3.WS212\Invasive\Cache\http\gle3\shared	✓	✓	✓	
http-disk-cache-ignore-write-errors	Whether to ignore write errors to disk cache.	Shared	False	✓	✓	✓	
http-disk-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP disk cache.	Shared	2592000	✓	✓	✓	
http-get-timeout-max-ms	HTTP GET maximum timeout on retry (ms).		24000	✓	✓	✓	
http-get-timeout-ms	HTTP GET timeout (ms).		56000	✓	✓	✓	
http-memory-cache-compression-level	Compression level for the HTTP memory cache, ranging from 1 (little) to 9 (intense). Default is 5.	OData	5	✓	✓	✓	
http-memory-cache-max-age-sec	Maximum acceptable age in seconds for use of data in the HTTP memory cache.	OData	14400	✓	✓	✓	
http-post-timeout-max-ms	HTTP POST maximum timeout on retry (ms).		58000	✓	✓	✓	
http-post-timeout-ms	HTTP POST timeout (ms).		57000	✓	✓	✓	
ignore-http-400-errors	Ignore HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-401-errors	Ignore HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-402-errors	Ignore HTTP 402 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
ignore-http-403-errors	Ignore HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-404-errors	Ignore HTTP 404 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-422-errors	Ignore HTTP 422 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-429-errors	Ignore HTTP 429 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-500-errors	Ignore HTTP 500 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-502-errors	Ignore HTTP 502 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
ignore-http-503-errors	Ignore HTTP 503 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
invalid-json-on-get-max-tries	Maximum number of tries when the JSON received on GET is invalid.		1	✓	✓	✓	
invalid-json-on-get-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on GET is invalid.		1000	✓	✓	✓	
invalid-json-on-get-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on GET is invalid.		10000	✓	✓	✓	
invalid-json-on-get-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on GET is invalid.		2	✓	✓	✓	
invalid-json-on-post-max-tries	Maximum number of tries when the JSON received on POST is invalid.		1	✓	✓	✓	
invalid-json-on-post-sleep-initial-ms	Initial sleep in milliseconds between retries when the JSON received on POST is invalid.		1000	✓	✓	✓	
invalid-json-on-post-sleep-max-ms	Maximum sleep in milliseconds between retries when the JSON received on POST is invalid.		10000	✓	✓	✓	
invalid-json-on-post-sleep-multiplicator	Multiplication factor for sleep between retries when the JSON received on POST is invalid.		2	✓	✓	✓	
invariant-sql-compress-sparse-arrays	Whether to compress sparse arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invariant-sql-correct-invalid-date	Whether to correct dates considered invalid since they are before 01-01-1753. When nullable, they are	SQL Engine V1	False	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
	removed. Otherwise they are replaced by 01-01-1753.						
invariant-sql-forward-filters-to-data-containers	Whether to forward filters to data containers.	SQL Engine V1	True	✓	✓	✓	
invariant-sql-share-byte-arrays	Whether to share the memory used by identical byte arrays in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invariant-sql-share-strings	Whether to share the memory used by identical strings in result sets during compression.	SQL Engine V1	True	✓	✓	✓	
invariant-sql-shuffle-fetch-results-data-containers	Whether to shuffle results fetched from data containers.	SQL Engine V1	False	✓	✓	✓	
invariant-use-cache	Whether to cache the results of a query.	SQL Engine V1	True	✓	✓	✓	
join-set-points-per-request	Maximum number of values in a request when executing a join set.	OData	60	✓	✓	✓	
limit-partition-calls-left	Minimum number of remaining API calls on a partition towards a hard limit. When below, an error is raised.	OData	500	✓	✓	✓	
log-native-calls-to-disk-max-events	Maximum number of call events to register from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-max-seconds	Maximum number of seconds to register calls from last activation.	Shared		✓	✓	✓	
log-native-calls-to-disk-on-error	Registers native calls to data container backend as disk files when the call raised an error.	Shared	False	✓	✓	✓	
log-native-calls-to-disk-on-success	Registers native calls to data container backend as disk files when the call raised no error.	Shared	False	✓	✓	✓	
log-native-calls-to-trace	Log native calls to data container backend on the trace.	Shared	False	✓	✓	✓	
maximum-length-identifiers	Non-default maximum length in characters of identifier names.	Shared		✓	✓	✓	
max-odata-filters	Maximum number of OData filter elements.	OData	100	✓	✓	✓	
max-url-length-accepted	The maximum accepted URL length before raising an error.	Shared	8000	✓	✓	✓	
max-url-length-desired	The maximum desired URL length.	Shared	8000	✓	✓	✓	
metadata-cache-max-age-sec	Maximum acceptable age in seconds for re-use of metadata.	OData		✓	✓	✓	
oauth-unauthorized-max-tries	Maximum number of tries when an OAuth exception occurs.	OData	2	✓	✓	✓	
oauth-unauthorized-sleep-initial-ms	Initial sleep in milliseconds between OAuth reauthentication tries when	OData	10000	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
	the OAuth authentication fails.						
oauth-unauthorized-sleep-max-ms	Maximum sleep in milliseconds between OAuth reauthentication tries when the OAuth authentication fails.	OData	1000	✓	✓	✓	
oauth-unauthorized-sleep-multiplicator	Multiplication factor for sleep between OAuth reauthentication tries when the OAuth authentication fails.	OData	2	✓	✓	✓	
partition-slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a partition-based rate limit.	Shared	60000	✓		✓	
partition-slot-based-rate-limit-slots	Number of slots per partition-based rate limit. Null means no slot-based rate limit.	Shared		✓		✓	
pre-request-delay-ms	Pre-request delay in milliseconds per request.	Shared	0	✓	✓	✓	
requested-page-size	Preferred number of rows to exchange per round trip; only effective on limited platforms such as AFAS Online.	Shared		✓	✓	✓	
requests-parallel-max	Maximum number of parallel data requests from individual partitions on the data container.	Shared	32	✓	✓	✓	
retry-event-upload	Whether to retry uploading an event when it fails.	ActiveCampaign		✓		✓	✓
simulate-http-400-errors	Simulate HTTP 400 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-400-errors-percentage	Percentage of simulated HTTP 400 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-401-errors	Simulate HTTP 401 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-401-errors-percentage	Percentage of simulated HTTP 401 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-403-errors	Simulate HTTP 403 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-403-errors-percentage	Percentage of simulated HTTP 403 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	
simulate-http-408-errors	Simulate HTTP 408 errors when exchanging results with the HTTP endpoint.		False	✓	✓	✓	
simulate-http-408-errors-percentage	Percentage of simulated HTTP 408 errors when exchanging results with the HTTP endpoint.		0	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
simulate-http-429-errors	Simulate HTTP 429 errors w hen exchanging results w ith the HTTP endpoint.		False	✓	✓	✓	
simulate-http-429-errors-percentage	Percentage of simulated HTTP 429 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-500-errors	Simulate HTTP 500 errors w hen exchanging results w ith the HTTP endpoint.		False	✓	✓	✓	
simulate-http-500-errors-percentage	Percentage of simulated HTTP 500 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-502-errors	Simulate HTTP 502 errors w hen exchanging results w ith the HTTP endpoint.		False	✓	✓	✓	
simulate-http-502-errors-percentage	Percentage of simulated HTTP 502 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-503-errors	Simulate HTTP 503 errors w hen exchanging results w ith the HTTP endpoint.		False	✓	✓	✓	
simulate-http-503-errors-percentage	Percentage of simulated HTTP 503 errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-protocol-errors	Simulate HTTP protocol errors w hen exchanging results w ith the HTTP endpoint.		False	✓	✓	✓	
simulate-http-protocol-errors-percentage	Percentage of simulated HTTP protocol errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
simulate-http-timeout-errors	Simulate HTTP timeout errors w hen exchanging results w ith the HTTP endpoint..		False	✓	✓	✓	
simulate-http-timeout-errors-percentage	Percentage of simulated HTTP timeout errors w hen exchanging results w ith the HTTP endpoint.		0	✓	✓	✓	
slot-based-rate-limit-length-ms	Total length in milliseconds across all slots of a slot-based rate limit.	Shared	1050	✓		✓	
slot-based-rate-limit-slots	Number of slots of a slot-based rate limit. Null means no slot-based rate limit.	Shared	5	✓		✓	
standardize-identifiers	Rew rite all identifiers to the preferred standards as configured by standardize-identifiers-casing and maximum-length-identifiers.	Shared	True	✓	✓	✓	
standardize-identifiers-casing	Rew rite all identifiers to the recommended standard platform-specific casing w hen changing a data model on a case-dependent platform.	Shared	True	✓	✓	✓	

Code	Description	Origin	Default Value	Set from Connection String	Set from Set SQL-Statement	Set from Driver's File	Set from Log On
use-batch-insert	Whether to use batch insert.	OData	False	✓	✓	✓	
use-http-disk-cache-read	Whether to use HTTP responses from previous queries stored on disk to answer the current query.	Shared	False	✓	✓	✓	
use-http-disk-cache-write	Whether to memorize HTTP responses on disk.	Shared	False	✓	✓	✓	
use-http-memory-cache-read	Whether to use HTTP responses from previous queries stored in memory that can answer the current query.	OData	True	✓	✓	✓	
use-http-memory-cache-write	Whether to memorize HTTP responses from previous queries for use by future queries.	OData	True	✓	✓	✓	

3 Schema: Native

3.1 Tables

3.1.1 NATIVEPLATFORMSCALARREQUESTS: ActiveCampaign Native Platform Scalar Requests

{res:itgen_native_platform_scalar_requests_desc}

Catalog: ActiveCampaign

Schema: Native

Alias: npt

Label: Native Platform Scalar Requests

Documentation:

The NativePlatformScalarRequests table provides direct access to the native API protocol over an established connection to the ActiveCampaign API server. It will contain a new row for every row inserted with a native API request in PAYLOAD_TEXT with the results of unaltered forwarding of the payload to the ActiveCampaign API server.

Retrieve: true

Insert: true

Update: false

Delete: false

View Columns

The columns of the view NATIVEPLATFORMSCALARREQUESTS are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
BLOB_PREFERRED	boolean	BLOB Preferred	☒	Indicator whether a BLOB result is preferred over text.
BOL_RESPONSE_CACHE_MAX_AGE_SEC	int32	Response Cache Maximum Age (sec)	☐	Maximum age in seconds of Bridge Online response cache entries to be used.
CONTENT_TYPE	string(240)	Content Type	☐	
DATE_ENDED	datetime	End Date	☒	
DATE_STARTED	datetime	Start Date	☒	
DRY_RUN	boolean	Run without Actions	☒	
DURATION_MS	int32	Duration (ms)	☒	
ERROR_MESSAGE_CODE	string(30)	Error Message Code	☐	
ERROR_MESSAGE_TEXT	string(32000)	Error Message Text	☐	
FAIL_ON_ERROR	boolean	Fail on Error	☒	Whether to raise an exception when processing the native request triggered an error from the provider.
HTTP_DISK_CACHE_MAX_AGE_SEC	int32	HTTP Disk Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP disk cache entries to be used.
HTTP_DISK_CACHE_SAVE	boolean	Save HTTP Disk Cache	☐	Whether results can be stored in HTTP disk cache.
HTTP_DISK_CACHE_USE	boolean	Use HTTP Disk Cache	☐	Whether results can be fetched from HTTP disk cache.
HTTP_MEMORY_CACHE_MAX_AGE_SEC	int32	HTTP Memory Cache Maximum Age (sec)	☐	Maximum age in seconds of HTTP memory cache entries to be used.
HTTP_MEMORY_CACHE_SAVE	boolean	Save HTTP Memory Cache	☐	Whether results can be stored in HTTP memory cache.
HTTP_MEMORY_CACHE_USE	boolean	Use HTTP Memory Cache	☐	Whether results can be fetched from HTTP memory cache.
HTTP_METHOD	string(30)	HTTP Method	☐	
HTTP_STATUS_CODE	int16	HTTP Status Code	☐	
ORIG_SYSTEM_GROUP	string(4000)	Original System Group	☐	
ORIG_SYSTEM_REFERENCE	string(4000)	Original System Reference	☐	
PAYLOAD_TEXT	string	Payload	☐	
RESULT_BLOB	byte[]	Result BLOB	☐	
RESULT_DATE_TIME_UTC	datetime	Result Date Time	☐	
RESULT_NUMBER	decimal	Result Number	☐	
RESULT_TEXT	string	Result Text	☐	
SUCCESSFUL	boolean	Successful	☒	
TIMEOUT_SEC	int32	Timeout (sec)	☐	Timeout in seconds.
TRANSACTION_ID	int32	Transaction ID	☒	Incrementing ID of the transaction.
URL	string(4000)	URL	☐	

4 Schema: V3

4.1 Tables

4.1.1 AccountById: ActiveCampaign Account by ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account by ID

This is a read-only table function. The ActiveCampaign API may not support changing the data or the Invariant SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountById. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function AccountById are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
accountUrl	string	Account (URL)	☐	Website.
contactCount	int64	Number of Contacts	☐	
createdTimestamp	datetime	Created	☐	
dealCount	int64	Number of Deals	☐	
id	int64	ID	☒	Unique ID.
name	string	Name	☒	
updatedTimestamp	datetime	Last Modified	☐	

4.1.2 AccountContactsByContactId: ActiveCampaign Account Contacts by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account Contacts by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AccountContactsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function AccountContactsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
account	int64	Account	☒	
contact	int64	Contact	☒	
createdTimestamp	datetime	Created	☒	
id	int64	ID	☒	
jobTitle	string	Job Title	☐	
updatedTimestamp	datetime	Last Modified	☒	

4.1.3 AccountCustomFields: ActiveCampaign Account Custom Fields

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account Custom Fields

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table AccountCustomFields are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
createdTimestamp	datetime	Created	☒	
displayOrder	int64	Display Order	☒	Order for displaying the field on Manage Fields page and deal profiles.
fieldDefault	string	Field Default Value	☐	Default value of the field.
fieldDefaultCurrency	string	Field Default Currency	☐	The 3-letter currency code of the default currency for the field.
fieldLabel	string	Field Label	☒	Name of the field.
fieldOptions	string	Field Options	☐	
fieldType	string	Field Type	☒	Type of field. Possible values are: text, textarea, date, dropdown, multiselect, radio, checkbox, hidden, currency or number.
hideFieldFlag	int64	Hide Field	☒	
id	int64	ID	☒	ID of the custom field.
isFormVisible	int64	Form Visible	☒	Whether or not the field is visible on forms.
isRequired	int64	Required	☒	Whether or not the field is required on forms.
knownFieldId	int64	Known Field ID	☐	
personalization	string	Personalization	☐	
updatedTimestamp	datetime	Last Modified	☒	

4.1.4 AccountCustomFieldValues: ActiveCampaign Account Custom Field Values

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account Custom Field Values

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table AccountCustomFieldValues are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
accountCustomFieldMetumId	int64		☒	
accountId	int64	Account ID	☒	
createdTimestamp	datetime	Created	☒	
customFieldId	int64	Custom Field ID	☒	
fieldValue	string	Field Value	☐	
id	int64	ID	☒	
updatedTimestamp	datetime	Last Modified	☒	

4.1.5 Accounts: ActiveCampaign Accounts

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Accounts

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Accounts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
accountUrl	string	Account (URL)	☐	Website.
createdTimestamp	datetime	Created	☐	
id	int64	ID	☒	Unique ID.
name	string	Name	☒	
updatedTimestamp	datetime	Last Modified	☐	

4.1.6 AccountWithCustomFieldValues: ActiveCampaign Account Custom Fields

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account Custom Fields

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table AccountWithCustomFieldValues are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
created_timestamp	datetime	Created	☐	
custom_field_currency_type	string		☐	
custom_field_currency_value	decimal		☐	
custom_field_date_value	datetime		☐	
custom_field_id	int64		☒	
custom_field_number_value	decimal		☐	
custom_field_text_blob	string		☐	
custom_field_text_value	string		☐	
customer_account_id	int64		☒	Account ID.
id	int64	ID	☒	Unique ID.
updated_timestamp	datetime	Updated	☐	

4.1.7 Addresses: ActiveCampaign Addresses

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Addresses

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Addresses are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
address1	string	Address Line 1	☐	
address2	string	Address Line 2	☐	
allgroup	string		☒	
city	string	City	☐	
companyName	string	Company Name	☐	
country	string	Country	☒	Accepts a (2) two character string - country code (eg 'US', 'CA', 'MX').
district	string	District	☐	

Name	Data Type	Label	Required	Documentation
id	int64	ID	☒	
isDefault	string	Default	☒	Indicates default address.
state	string	State	☐	
zip	string	ZIP Code	☐	

4.1.8 Associations: ActiveCampaign Account Contacts

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Account Contacts

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Associations are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
account	int64	Account	☒	
contact	int64	Contact	☒	
createdTimestamp	datetime	Created	☒	
jobTitle	string	Job Title	☐	
updatedTimestamp	datetime	Last Modified	☒	

4.1.9 AutomationEntryCountsByContactId: ActiveCampaign Automation Entry Counts by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Automation Entry Counts by Contact ID

This is a read-only table function. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function AutomationEntryCountsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the

execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `AutomationEntryCountsByContactId` are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
contactEntryCount	int64		☒	
hidden	int64	Hidden	☒	
id	int64	ID	☒	
name	string	Name	☒	
status	int64	Status	☒	

4.1.10 Automations: ActiveCampaign Automations

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Automations

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table `Automations` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
defaultscreenshot	string	Default Screenshot	☒	
entered	string		☒	
exited	string		☒	
hidden	string	Hidden	☒	

Name	Data Type	Label	Required	Documentation
mdate	datetime	Last Modified	☒	
name	string	Name	☒	
screenshot	string	Screenshot	☒	
status	string	Status	☒	
userid	string	User ID	☒	

4.1.11 BounceLogsByContactId: ActiveCampaign Bounce Logs by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Bounce Logs by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function BounceLogsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function BounceLogsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
bounce	int64		☒	

Name	Data Type	Label	Required	Documentation
bounceid	int64	Bounce ID	☒	
campaign	int64		☒	
campaignid	int64		☒	
code	string		☐	
codeid	int64		☒	
contact	int64	Contact	☒	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☒	
email	string	Email Address	☐	
error	string		☐	
id	int64	ID	☒	
message	int64		☒	
messageid	int64		☒	
source	string	Source	☒	
subscriberid	int64		☒	
tstamp	datetime		☒	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☒	

4.1.12 Brandings: ActiveCampaign Brandings

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Brandings

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Brandings are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
adminTemplateCss	string		☐	
adminTemplateHtm	string		☐	
copyright	string	Copyright	☒	
favicon	string	Favicon	☐	
footerHtmlValue	string	Footer (HTML)	☐	
footerTextValue	string	Footer (text)	☐	
groupid	int64	Group ID	☒	
headerHtmlValue	string	Header (HTML)	☐	

Name	Data Type	Label	Required	Documentation
headerTextValue	string	Header (text)	☐	
help	string	Help	☒	
id	int64	ID	☒	
license	string	License	☒	
publicTemplateCss	string	Public Template (CSS)	☐	
publicTemplateHtm	string	Public Template (HTML)	☐	
siteLogo	string	Site Logo	☐	
siteLogoSmall	string	Site Logo Small	☐	
siteName	string	Site Name	☒	
version	string	Version	☒	

4.1.13 Calendars: ActiveCampaign Calendars

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Calendars

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Calendars are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
id	int64	ID	☒	
mdate	datetime	Last Modified	☒	
notification	int64	Notification	☒	
title	string	Title	☒	
token	string	Token	☒	
type	string	Type	☒	

4.1.14 Campaigns: ActiveCampaign Campaigns

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Campaigns

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Campaigns are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
activerss_interval	string		☒	
activerss_items	int64		☒	
activerss_url	string		☐	
addressid	int64	Address ID	☒	
analytics_campaign_name	string	Analytics Campaign Name	☐	
automation	int64	Automation	☐	
basemessageid	int64	Base Message ID	☒	
basetemplateid	string	Base Template ID	☒	
bounceid	int64	Bounce ID	☒	
cdate	datetime	Created	☒	
deletestamp	datetime	Deleted	☐	
embed_images	int64	Embed Images	☒	
facebook	int64	Facebook	☒	
formid	int64	Form ID	☒	
forwards	int64		☒	
hardbounces	int64		☒	
htmlunsub	string		☒	
htmlunsubdata	int64		☐	
id	int64	ID	☒	
ip4	int64	IPv4 Address	☒	
laststep	string	Last Step	☐	
ldate	datetime		☐	
linkclicks	int64	Link Clicks	☒	
mail_cleanup	int64	Mail Cleanup	☒	
mail_send	int64	Mail Send	☒	
mail_transfer	int64	Mail Transfer	☒	
mailer_log_file	int64	Mailer Log File	☒	
managetext	int64	Manage Text	☒	
mdate	datetime	Last Modified	☐	
name	string	Name	☒	
opens	int64	Number of Opens	☒	
public	int64	Public	☒	
realcid	int64	Real CID	☒	
recurring	string	Recurring	☒	
reminder_field	string		☒	
reminder_format	string		☐	

Name	Data Type	Label	Required	Documentation
reminder_last_cron_run	string		☐	
reminder_offset_sign	string		☒	
reminder_offset_type	string		☒	
reminder_offset	int64		☒	
reminder_type	string		☒	
replies	int64	Replies	☒	
replsys	int64		☒	
responder_existing	int64		☒	
responder_offset	int64		☒	
responder_type	string		☒	
schedule	int64	Schedule	☒	
scheduleddate	datetime	Scheduled Date	☐	
sdate	datetime		☐	
segmentid	int64	Segment ID	☒	
send_amt	int64	Send Amount	☒	
sendid	int64		☒	
seriesid	int64	Series ID	☒	
socialshares	int64	Social Shares	☒	
softbounces	int64	Soft Bounces	☒	
source	string	Source	☒	
split_content	int64	Split Content	☒	
split_offset_type	string	Split Offset Type	☐	
split_offset	int64	Split Offset	☒	
split_type	string	Split Type	☐	
split_w inner_awaiting	int64		☒	
split_w inner_messageid	int64		☒	
status	int64	Status	☒	
subscriberclicks	int64	Subscriber Clicks	☒	
survey	string	Survey	☐	
textunsub	string		☒	
textunsubdata	int64		☐	
threadid	int64	Thread ID	☒	
total_amt	int64	Total Amount	☒	
tracklinks	string	Track Links	☒	
tracklinksanalytics	int64	Track Link Analytics	☒	
trackreads	int64	Track Reads	☒	
trackreadsanalytics	int64	Track Reads Analytics	☒	
tweet	int64	Tweet	☒	
type	string	Type	☒	
uniqueforwards	int64	Unique Forwards	☒	
uniquelinkclicks	int64	Unique Link Clicks	☒	

Name	Data Type	Label	Required	Documentation
uniqueopens	int64	Unique Opens	☒	
uniquereplies	int64	Unique Replies	☒	
unsubreasons	int64	Unsubscription Reasons	☒	
unsubscribes	int64	Unsubscribes	☒	
updates	int64	Updates	☒	
user	int64	User	☒	
userid	int64	User ID	☒	
waitpreview	int64	Wait Preview	☒	
willrecur	int64	Will Recur	☒	

4.1.15 Connections: ActiveCampaign Connections

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Connections

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Connections are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
externalid	string	External ID	☐	
id	int64	ID	☒	
isInternal	int64	Internal	☒	
lastSync	datetime	Last Synchronization	☐	
linkUrl	string	Link (URL)	☐	
logoUrl	string	Logo (URL)	☐	
name	string	Name	☒	
service	string	Service	☒	
status	int64	Status	☒	
syncStatus	int64	Synchronization Status	☒	
update	datetime	Last Updated	☐	

4.1.16 ContactAutomations: ActiveCampaign Contact Automations

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Automations

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table ContactAutomations are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
adddate	datetime		☒	
automation	int64	Automation	☒	
batchid	string	Batch ID	☐	
completed	int64	Completed	☒	
completedElements	int64		☒	
completeValue	int64		☒	
contact	int64	Contact	☒	
id	int64	ID	☒	
lastblock	int64	Last Block	☒	
lastdate	datetime	Last Date	☒	
lastlogid	int64	Last Log ID	☒	
remdate	datetime		☐	
seriesid	int64	Series ID	☒	
startid	int64	Start ID	☒	
status	int64	Status	☒	
timespan	int64	Timespan	☐	
totalElements	int64	Total Elements	☒	

4.1.17 ContactAutomationsByContactId: ActiveCampaign Contact Automations by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Automations by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `ContactAutomationsByContactId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `ContactAutomationsByContactId` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
adddate	datetime		☒	
automation	int64	Automation	☒	
batchid	string	Batch ID	☐	
completed	int64	Completed	☒	
completedElements	int64		☒	
completeValue	int64		☒	
contact	int64	Contact	☒	
id	int64	ID	☒	
lastblock	int64	Last Block	☒	
lastdate	datetime	Last Date	☒	
lastlogid	int64	Last Log ID	☒	
remdate	datetime		☐	
seriesid	int64	Series ID	☒	
startid	int64	Start ID	☒	
status	int64	Status	☒	
timespan	int64	Timespan	☐	
totalElements	int64	Total Elements	☒	

4.1.18 ContactDataByContactId: ActiveCampaign Contact Data by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Data by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactDataByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function ContactDataByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
id	int64	ID	☒	

4.1.19 ContactDealsByContactId: ActiveCampaign Deals by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deals by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `ContactDealsByContactId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `ContactDealsByContactId` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
contact	int64	Contact	☒	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☒	
deal	int64	Deal	☒	
id	int64	ID	☒	
role	string		☐	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☒	

4.1.20 ContactGoalsByContactId: ActiveCampaign Goals by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Goals by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function `ContactGoalsByContactId`. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `ContactGoalsByContactId` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
id	int64	ID	☒	

4.1.21 ContactListsByContactId: ActiveCampaign Lists by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Lists by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactListsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function ContactListsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
automation	int64	Automation	☐	
autosyncLog	int64		☐	
campaign	int64		☐	
contact	int64	Contact	☒	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☒	
first_name	string		☐	
form	int64		☐	
id	int64	ID	☒	
ip4_last	int64		☒	
ip4Sub	int64		☒	
ip4Unsub	int64		☒	
last_name	string		☐	
list	int64		☒	
message	string		☐	
responder	int64		☒	
sdate	datetime		☒	
seriesid	int64	Series ID	☒	

Name	Data Type	Label	Required	Documentation
sourceid	int64		☒	
status	int64	Status	☒	
sync	int64		☒	
update	datetime	Last Updated	☐	
unsubreason	string		☐	
unsubscribeAutomation	int64		☐	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☒	

4.1.22 ContactLogsByContactId: ActiveCampaign Logs by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Logs by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactLogsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function ContactLogsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
id	int64	ID	☒	

4.1.23 Contacts: ActiveCampaign Contacts

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contacts

Can retrieve data and change data using insert, update and delete.

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function Contacts. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
exclude	int32	☐		Exclude from the response the contact with the given ID.
formid	int32	☐		Filter contacts associated with the given form.
in_group_lists	string	☐		Set this to 'true' in order to return only contacts that the current user has permissions to see.
listid	string	☐		Filter contacts associated with the given list.
segmentid	int32	☐		Return only contacts that match a list segment (this param initially returns segment information, when it is run a second time it will return contacts that match the segment).
seriesid	int32	☐		Filter contacts associated with the given automation.

Name	Data Type	Required	Default Value	Documentation
status	string	☐		-1 = Any, 0 = Unconfirmed, 1 = Active, 2 = Unsubscribed, 3 = Bounced.
tag	int32	☐		Filter contacts associated with the given tag.
waitid	int32	☐		Filter by contacts in the wait queue of an automation block.

Columns of Table Function

The columns of the table function Contacts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
adate	datetime		☐	
bounced_date	datetime	Bounced Date	☐	
bounced_hard	int64	Bounced Hard	☒	
bounced_soft	int64	Bounced Soft	☒	
cdate	datetime	Created	☐	
deleted	int64	Deleted	☒	
edate	datetime		☐	
email_domain	string		☐	
email_local	string		☐	
email	string	Email Address	☐	
firstName	string	First Name	☐	
gravatar	int64	Gravatar	☒	
hash	string	Hash	☐	
id	int64	ID	☒	
ip	int64	IP Address	☒	
lastName	string	Last Name	☐	
organization	int64	Account	☐	Deprecated. Do not use.
orgid	int64	Organization ID	☒	
phone	string	Phone	☐	
rating_tstamp	datetime	Rating Date	☐	
segmentio_id	int64		☐	
sentcnt	int64		☒	
socialdata_lastcheck	datetime	Social Data Last Check	☐	
ua	string		☐	
update	datetime	Last Updated	☐	

4.1.24 ContactTagsByContactId: ActiveCampaign Tags by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Tags by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ContactTagsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function ContactTagsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
contact	int64	Contact	☒	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☒	
id	int64	ID	☒	
tag	int64	Tag	☒	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☒	

4.1.25 ContactWithBounceLogs: ActiveCampaign Contact Bounce Logs

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Bounce Logs

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table ContactWithBounceLogs are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
bounceid	int64	Bounce ID	☒	Bounce ID.
campaignid	int64		☒	Campaign ID.
codeid	int64		☐	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☐	
email	string	Email Address	☐	
error	string		☐	
id	int64	ID	☒	Unique ID.
messageid	int64		☒	Message ID.
source	string	Source	☐	
subscriberid	int64		☒	Contact ID.
tstamp	datetime		☐	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☐	

4.1.26 ContactWithCustomFieldValues: ActiveCampaign Contact Custom Fields

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Custom Fields

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table ContactWithCustomFieldValues are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
contact	int64	Contact	☒	Contact ID.
created_by	string	Created by	☐	
field	int64		☒	
id	int64	ID	☒	Unique ID.
update	datetime	Last Updated	☐	
updated_by	string	Updated by	☐	
value	string	Value	☐	

4.1.27 ContactWithTags: ActiveCampaign Contact Tags

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Contact Tags

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table ContactWithTags are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
contact	int64	Contact	☒	Contact ID.
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☐	
id	int64	ID	☒	Unique ID.
tag	int64	Tag	☒	Tag ID.
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☐	

4.1.28 CustomFields: ActiveCampaign Custom Fields

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Custom Fields

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table CustomFields are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
cols	int64		☒	
defval	string	Default Value	☐	
descript	string	Description	☐	
id	int64	ID	☒	
isrequired	int64	Required	☒	
ordernum	int64	Order Number	☒	
perstag	string		☒	
row s	int64	Row s	☒	
service	string	Service	☐	
show_in_list	int64	Show in List	☒	
title	string	Title	☒	
type	string	Type	☒	
update	datetime	Last Updated	☐	
visible	int64	Visible	☒	

4.1.29 CustomFieldValues: ActiveCampaign Custom Field Values

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Custom Field Values

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table CustomFieldValues are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
contact	int64	Contact	☒	
field	int64		☒	
id	int64	ID	☒	
ow ner	int64	Ow ner	☒	
update	datetime	Last Updated	☐	
value	string	Value	☐	

4.1.30 DealCustomFields: ActiveCampaign Deal Custom Fields

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Custom Fields

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table DealCustomFields are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
createdTimestamp	datetime	Created	☒	
displayOrder	int64	Display Order	☒	
fieldDefault	string	Field Default Value	☐	
fieldDefaultCurrency	string	Field Default Currency	☐	
fieldLabel	string	Field Label	☒	
fieldOptions	string	Field Options	☐	
fieldType	string	Field Type	☒	
hideFieldFlag	int64	Hide Field	☒	
id	int64	ID	☒	
isFormVisible	int64	Form Visible	☒	
isRequired	int64	Required	☒	
knownFieldId	int64	Known Field ID	☐	
personalization	string	Personalization	☐	
updatedTimestamp	datetime	Last Modified	☒	

4.1.31 DealCustomFieldValues: ActiveCampaign Deal Custom Field Values

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Custom Field Values

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table DealCustomFieldValues are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
createdTimestamp	datetime	Created	☒	

Name	Data Type	Label	Required	Documentation
customFieldId	int64	Custom Field ID	☒	
dealCustomFieldMetumId	int64		☒	
dealId	int64	Deal ID	☒	
fieldValue	string	Field Value	☐	
id	int64	ID	☒	
updatedAtTimestamp	datetime	Last Modified	☒	

4.1.32 Deals: ActiveCampaign Deals

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deals

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Deals are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
account	int64	Account	☐	Reference to the account.
activitycount	int64	Activity Count	☐	
cdate	datetime	Created	☐	Creation date.
contact	int64	Contact	☐	ID of the primary contact.
currency	string	Currency	☐	Currency code for the money value.
customerAccount	int64	Customer Account	☐	
description	string(1000)	Description	☐	Description of the deal.
edate	datetime		☐	
group	int64	Group	☐	ID of the pipeline.
id	int64	ID	☒	
isDisabled	boolean	Disabled	☒	
mdate	datetime	Last Modified	☐	Last modified date.
nextdate	datetime	Next Date	☐	
nextdealid	int64	Next Deal ID	☐	
nexttaskid	int64	Next Task ID	☐	
organization	int64	Account	☐	Deprecated. Do not use.
owner	int64	Owner	☐	ID of the user owning the deal.
percent	decimal	Percentage	☐	Percentage of the deal.
stage	int64	Stage	☐	ID of the stage.

Name	Data Type	Label	Required	Documentation
status	int64	Status	☒	0 is Open, 1 is Won and 2 is Lost.
title	string	Title	☐	Title of the deal.
value	int64	Value	☐	Values for the field. For currency fields in cents.
w inProbability	int64	Win Probability	☐	
w inProbabilityMdate	datetime	Win Probability Last Updated	☐	

4.1.33 DealsByContactId: ActiveCampaign Deals by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deals by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function DealsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function DealsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
account	int64	Account	☒	
activitycount	int64	Activity Count	☐	
cdate	datetime	Created	☒	
contact	int64	Contact	☒	
currency	string	Currency	☒	
customerAccount	int64	Customer Account	☒	
description	string	Description	☐	
edate	datetime		☐	
group	int64	Group	☐	
hash	string	Hash	☐	
id	int64	ID	☒	
isDisabled	boolean	Disabled	☒	
mdate	datetime	Last Modified	☒	
nextdate	datetime	Next Date	☐	
nextdealid	int64	Next Deal ID	☐	
nextTask	int64		☐	
nexttaskid	int64	Next Task ID	☒	
organization	int64	Account	☒	Deprecated. Do not use.
owner	int64	Owner	☐	
percent	int64	Percentage	☐	
stage	int64	Stage	☐	
status	int64	Status	☒	
title	string	Title	☐	
value	int64	Value	☐	
winProbability	int64	Win Probability	☐	
winProbabilityMdate	datetime	Win Probability Last Updated	☐	

4.1.34 DealWithActivities: ActiveCampaign Deal Activities

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Activities

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table DealWithActivities are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
automation	int64	Automation	☐	Automation ID.
cdate	datetime	Created	☐	
dataAction	string		☐	
dataId	int64		☒	Data ID.
dataOldval	string		☐	
dataType	string		☐	
deal	int64	Deal	☒	Deal ID.
deleted	int64	Deleted	☒	
id	int64	ID	☒	Unique ID.
isAddtask	int64		☒	
seriesid	int64	Series ID	☒	Series ID.
sortdate	datetime		☐	
stage	int64	Stage	☐	Deal stage ID.
user	int64	User	☐	User ID.

4.1.35 DealWithContacts: ActiveCampaign Deal Contacts

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Contacts

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table DealWithContacts are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
adate	datetime		☐	
anonymized	int64		☒	
bounced_date	datetime	Bounced Date	☐	
bounced_hard	int64	Bounced Hard	☒	
bounced_soft	int64	Bounced Soft	☒	
cdate	datetime	Created	☐	
created_by	string	Created by	☐	
created_timestamp	datetime	Created	☒	
created_utc_timestamp	datetime		☒	
deleted_at	datetime		☐	
deleted	int64	Deleted	☒	

Name	Data Type	Label	Required	Documentation
edate	datetime		☐	
email_domain	string		☐	
email_empty	boolean		☒	
email_local	string		☐	
email	string	Email Address	☐	
firstName	string	First Name	☐	
gravatar	int64	Gravatar	☒	
hash	string	Hash	☐	
id	int64	ID	☒	Unique ID.
ip	int64	IP Address	☐	
lastName	string	Last Name	☐	
orgid	int64	Organization ID	☐	
orgname	string		☐	
phone	string	Phone	☐	
rating_tstamp	datetime	Rating Date	☐	
segmentio_id	int64		☐	
sentcnt	int64		☒	
sentiment	string		☐	
socialdata_lastcheck	datetime	Social Data Last Check	☐	
ua	string		☐	
update	datetime	Last Updated	☐	
updated_by	string	Updated by	☐	
updated_timestamp	datetime	Updated	☒	
updated_utc_timestamp	datetime		☒	

4.1.36 DealWithNotes: ActiveCampaign Deal Notes

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Notes

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table DealWithNotes are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	

Name	Data Type	Label	Required	Documentation
id	int64	ID	☒	Unique ID.
is_draft	int64		☒	
mdate	datetime	Last Modified	☒	
note	string	Note	☒	
owner_id	int64	Owner ID	☒	
owner_type	string	Owner Type	☒	
relid	int64	Relationship ID	☒	
reltype	string	Relationship Type	☒	
user	int64	User	☒	
userid	int64	User ID	☒	

4.1.37 DealWithTasks: ActiveCampaign Deal Tasks

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Deal Tasks

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table DealWithTasks are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
assigned_date	datetime		☐	
assignee	int64	Assignee	☐	
automation	int64	Automation	☐	Automation ID.
cdate	datetime	Created	☐	
created_utc_timestamp	datetime		☐	
createdseriesid	int64		☐	
dealTasktype	int64	Task Type	☒	
done_by	string		☐	
done_utc_timestamp	datetime		☐	
doneAutomation	int64		☐	
donedate	datetime		☐	
duedate	datetime	Due Date	☒	
duration	int64		☒	
edate	datetime		☐	
id	int64	ID	☒	Unique ID.

Name	Data Type	Label	Required	Documentation
note	string	Note	☐	
outcome_id	int64		☐	
outcome_info	string		☐	
relid	int64	Relationship ID	☒	
reltype	string	Relationship Type	☒	
status	int64	Status	☒	
title	string	Title	☐	
update	datetime	Last Updated	☐	
updated_by	string	Updated by	☐	
updated_utc_timestamp	datetime		☐	
user	int64	User	☒	

4.1.38 eComCustomers: ActiveCampaign eCommerce Customers

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: eCommerce Customers

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table eComCustomers are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
acceptsMarketing	int64		☒	
avgProductCategory	string		☐	
avgRevenuePerOrder	int64		☒	
connection	int64	Connection	☐	
connectionid	int64	Connection ID	☒	
email	string	Email Address	☐	
externalid	string	External ID	☐	
id	int64	ID	☒	
totalOrders	int64	Total Orders	☒	
totalProducts	int64	Total Products	☒	
totalRevenue	int64	Total Revenue	☒	
tstamp	datetime		☒	

4.1.39 eComOrderProducts: ActiveCampaign eCommerce Order Products

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: eCommerce Order Products

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table eComOrderProducts are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
category	string	Category	☒	
connectionid	int64	Connection ID	☒	
createdDate	datetime	Created	☒	
description	string	Description	☐	
ecomOrder	int64	eCommerce Order	☒	
externalid	string	External ID	☐	
id	int64	ID	☒	
imageUrl	string	Image (URL)	☐	
name	string	Name	☒	
orderid	int64	Order ID	☒	
price	int64	Price	☒	Unit price in smallest unit, typically cents.
productUrl	string	Product URL	☐	Link to the product on the store.
quantity	int64	Quantity	☒	
sku	string	SKU	☐	
tstamp	datetime		☒	
updatedAt	datetime	Last Updated	☒	

4.1.40 eComOrders: ActiveCampaign eCommerce Orders

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: eCommerce Orders

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table eComOrders are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
connection	int64	Connection	☒	
connectionid	int64	Connection ID	☒	
createdDate	datetime	Created	☒	
currency	string	Currency	☒	
customer	int64	Customer	☒	
customerid	int64	Customer ID	☒	
discountAmount	int64	Discount Amount	☐	
email	string	Email Address	☐	
externalcheckoutid	string		☐	
externalCreatedDate	datetime		☒	
externalid	string	External ID	☐	
id	int64	ID	☒	
orderDate	datetime	Order Date	☒	
orderProducts	int64	Order Products	☐	
orderUrl	string	Order URL	☐	
shippingAmount	int64	Shipping Amount	☐	
shippingMethod	int64	Shipping Method	☐	
source	int64	Source	☒	
state	int64	State	☒	
taxAmount	int64	Tax Amount	☐	
totalPrice	int64	Total Price	☒	Total sales amount including VAT in smallest unit, typically cents.
totalProducts	int64	Total Products	☒	
tstamp	datetime		☒	
updatedAt	datetime	Last Updated	☒	

4.1.41 Events: ActiveCampaign Events

Catalog: ActiveCampaign

Schema: V3

Primary Keys: name

Label: Events

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Events are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
name	string	Name	☒	

4.1.42 FieldValuesByContactId: ActiveCampaign Field Values by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Field Values by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function FieldValuesByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function FieldValuesByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
contact	int64	Contact	☒	
created_by	string	Created by	☐	
field	int64		☒	
id	int64	ID	☒	
owner	int64	Owner	☒	
update	datetime	Last Updated	☐	

Name	Data Type	Label	Required	Documentation
updated_by	string	Updated by	☐	
value	string	Value	☐	

4.1.43 Forms: ActiveCampaign Forms

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Forms

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Forms are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
action	string	Action	☐	
actiondata	string	Action Data	☒	
address	string	Address	☐	
addressid	int64	Address ID	☒	
aid	int64		☐	
body	string	Body	☐	
button	string	Button	☐	
cdate	datetime	Created	☒	
contacts	int64	Contacts	☒	
deals	int64	Deals	☒	
defaultscreenshot	string	Default Screenshot	☒	
entries	int64		☒	
id	int64	ID	☒	
layout	string	Layout	☒	
name	string	Name	☒	
options_blanks_overwrite	boolean	Overwrite Blanks	☐	
options_confaction	string		☐	
options_confform	int64		☐	
options_optin_created	boolean	Opt-in Created	☐	
options_optin_id	int64	Opt-in ID	☐	
options_sendoptin	boolean	Send Opt-in	☐	
parentformid	int64	Parent Form ID	☒	
style_ac_branding	boolean		☐	
style_background	string	Style Background	☐	

Name	Data Type	Label	Required	Documentation
style_border_color	string	Style Border Color	☐	
style_border_radius	int64	Style Border Radius	☐	
style_border_style	string	Style Border Style	☐	
style_border_width	int64	Style Border Width	☐	
style_button_background	string	Style Button Background	☐	
style_button_border_color	string		☐	
style_button_border_radius	int64		☐	
style_button_border_style	string		☐	
style_button_border_width	int64		☐	
style_button_fontcolor	string	Style Button Font Color	☐	
style_button_padding	int64	Style Button Padding	☐	
style_dark	boolean		☐	
style_fontcolor	string	Style Font Color	☐	
style_layout	string	Style Layout	☐	
style_width	int64	Style Width	☐	
submit	string	Submit	☒	
submitdata_url	string	Submit Data URL	☐	
thanks	string	Thanks	☐	
title	string	Title	☐	
update	datetime	Last Updated	☐	
url	string	URL	☐	
userid	int64	User ID	☒	

4.1.44 GeolpsByContactId: ActiveCampaign Geo IP-addresses by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Geo IP-addresses by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function GeolpsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a `select \* from table(value1, value2, value3)` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with `select \* from table(name1 => value1, name3 => value3)` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `GeolpsByContactId` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
campaignid	int64		☒	
contact	int64	Contact	☒	
geoAddress	int64		☒	
geoadddrid	int64		☒	
id	int64	ID	☒	
ip4	string	IPv4 Address	☒	
messageid	int64		☒	
tstamp	datetime		☒	

4.1.45 Groups: ActiveCampaign Groups

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Groups

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table `Groups` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
descript	string	Description	☐	
id	int64	ID	☒	
optinconfirm	int64	Opt-in Confirm	☒	

Name	Data Type	Label	Required	Documentation
p_admin	int64		✓	
pg_user_add	int64		✓	
pg_user_delete	int64		✓	
pg_user_edit	int64		✓	
pgAutomationManage	int64		✓	
pgContactActions	int64		✓	
pgContactAdd	int64		✓	
pgContactApprove	int64		✓	
pgContactDelete	int64		✓	
pgContactEdit	int64		✓	
pgContactExport	int64		✓	
pgContactFields	int64		✓	
pgContactFilters	int64		✓	
pgContactImport	int64		✓	
pgContactMerge	int64		✓	
pgContactSync	int64		✓	
pgDeal	int64		✓	
pgDealDelete	int64		✓	
pgDealGroupAdd	int64		✓	
pgDealGroupDelete	int64		✓	
pgDealGroupEdit	int64		✓	
pgDealReassign	int64		✓	
pgFormEdit	int64		✓	
pgGroupAdd	int64		✓	
pgGroupDelete	int64		✓	
pgGroupEdit	int64		✓	
pgListAdd	int64		✓	
pgListBounce	int64		✓	
pgListDelete	int64		✓	
pgListEdit	int64		✓	
pgListEmailaccount	int64		✓	
pgListHeaders	int64		✓	
pgMessageAdd	int64		✓	
pgMessageDelete	int64		✓	
pgMessageEdit	int64		✓	
pgMessageSend	int64		✓	
pgPersonalizationAdd	int64		✓	
pgPersonalizationDelete	int64		✓	
pgPersonalizationEdit	int64		✓	
pgReportsCampaign	int64		✓	
pgReportsList	int64		✓	

Name	Data Type	Label	Required	Documentation
pgReportsTrend	int64		☒	
pgReportsUser	int64		☒	
pgSavedResponsesManage	int64		☒	
pgStartupGettingsstarted	int64		☒	
pgStartupReports	int64		☒	
pgTemplateAdd	int64		☒	
pgTemplateDelete	int64		☒	
pgTemplateEdit	int64		☒	
reqApproval	int64		☒	
reqApproval1st	int64		☒	
reqApprovalNotify	string		☐	
sdate	datetime		☐	
socialdata	int64	Social Data	☒	
title	string	Title	☒	
unsubscribeLink	int64	Unsubscribe Link	☒	

4.1.46 Lists: ActiveCampaign Lists

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Lists

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Lists are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
analytics_domains	string	Analytics Domains	☐	
analytics_source	string	Analytics Source	☐	
analytics_ua	string		☐	
carboncopy	string	Carbon Copy	☐	Comma-separated list of email addresses to send a copy of all mailings to upon send.
cdate	datetime	Created	☒	
deletestamp	datetime	Deleted	☐	
facebook_session	string		☐	
fulladdress	string	Full Address	☐	
get_unsubscribe_reason	int64		☒	
id	int64	ID	☒	Unique ID.

Name	Data Type	Label	Required	Documentation
name	string	Name	☒	Name.
optinmessageid	int64	Opt-in Message ID	☒	
optinoptout	int64	Opt-in Opt-out	☒	
optoutconf	int64		☒	
p_embed_image	int64		☒	
p_use_analytics_link	int64		☒	
p_use_analytics_read	int64		☒	
p_use_captcha	int64		☒	
p_use_facebook	int64		☒	
p_use_tracking	int64		☒	
p_use_twitter	int64		☒	
private	int64	Private	☒	
require_name	int64		☒	
send_last_broadcast	int64	Send Last Broadcast	☒	
sender_addr1	string	Sender Address Line 1	☐	
sender_addr2	string	Sender Address Line 2	☐	
sender_city	string	Sender City	☐	
sender_country	string	Sender Country	☐	
sender_name	string	Sender Name	☐	
sender_phone	string	Sender Phone	☐	
sender_reminder	string	Sender Reminder	☐	A reminder for your contacts as to why they are on this list and you are messaging them.
sender_state	string	Sender State	☐	
sender_url	string	Sender URL	☐	Website URL this list is for.
sender_zip	string	Sender ZIP Code	☐	
stringid	string	String ID	☒	URL-safe list name.
subscription_notify	string	Subscription Notify	☐	Comma-separated list of email addresses to notify when a new subscriber joins this list.
to_name	string	To Name	☒	
twitter_token_secret	string	Twitter Token Secret	☐	
twitter_token	string	Twitter Token	☐	
update	datetime	Last Updated	☐	
unsubscribe_notify	string	Unsubscription Notify	☐	Comma-separated list of email addresses to notify when a subscriber unsubscribes from this list.
user	int64	User	☐	
userid	int64	User ID	☒	

4.1.47 Messages: ActiveCampaign Messages

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Messages

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Messages are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
charset	string	Character Set	☒	
ed_instanceid	int64		☒	
ed_version	int64		☒	
encoding	string	Encoding	☒	
format	string	Format	☒	
fromemail	string		☐	
fromname	string	From Name	☐	
hidden	int64	Hidden	☒	
html	string	HTML	☐	
htmlfetch	string		☒	
id	int64	ID	☒	
mdate	datetime	Last Modified	☐	
name	string	Name	☐	
preheader_text	string	Pre Header Text	☐	
preview_data	string	Preview Data	☐	
preview_mime	string	Preview MIME Type	☐	
priority	int64	Priority	☒	
reply2	string		☐	
subject	string	Subject	☐	
text	string	Text	☐	
textfetch	string		☒	
user	int64	User	☐	
userid	int64	User ID	☒	

4.1.48 NotesByContactId: ActiveCampaign Notes by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Notes by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function NotesByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function NotesByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
id	int64	ID	☒	
is_draft	int64		☒	
mdate	datetime	Last Modified	☒	
note	string	Note	☒	
owner_id	int64	Owner ID	☒	
owner_type	string	Owner Type	☒	
relid	int64	Relationship ID	☒	
reltype	string	Relationship Type	☒	
user	int64	User	☒	
userid	int64	User ID	☒	

4.1.49 Pipelines: ActiveCampaign Pipelines

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Pipelines

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Pipelines are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
allgroups	int64		☒	
allusers	int64		☒	
autoassign	int64	Auto-assign	☒	
cdate	datetime	Created	☐	
currency	string	Currency	☒	
id	int64	ID	☒	
title	string	Title	☒	
update	datetime	Last Updated	☐	

4.1.50 SavedResponses: ActiveCampaign Saved Responses

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Saved Responses

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table SavedResponses are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
body	string	Body	☐	
cdate	datetime	Created	☒	
id	int64	ID	☒	
last_sent_user_id	int64	Last Sent User ID	☐	
ldate	datetime		☐	
mdate	datetime	Last Modified	☐	
subject	string	Subject	☐	

Name	Data Type	Label	Required	Documentation
title	string	Title	☒	

4.1.51 Scores: ActiveCampaign Scores

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Scores

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Scores are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
descript	string	Description	☐	
id	int64	ID	☒	
mdate	datetime	Last Modified	☒	
name	string	Name	☒	
reltype	string	Relationship Type	☒	
status	int64	Status	☒	

4.1.52 ScoreValuesByContactId: ActiveCampaign Score Values by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Score Values by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function ScoreValuesByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function `ScoreValuesByContactId` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
contact	int64	Contact	☒	
deal	int64	Deal	☐	
id	int64	ID	☒	
mdate	datetime	Last Modified	☐	
score	int64		☒	
scoreValue	int64		☒	

4.1.53 SecondaryContacts: ActiveCampaign Secondary Contacts

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Secondary Contacts

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table `SecondaryContacts` are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
contact	int64	Contact	☒	
deal	int64	Deal	☒	
id	int64	ID	☒	

4.1.54 Segments: ActiveCampaign Segments

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Segments

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Segments are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
hidden	int64	Hidden	☒	
id	int64	ID	☒	
logic	string		☒	
name	string	Name	☐	
seriesid	int64	Series ID	☒	

4.1.55 SiteTracking: ActiveCampaign Site Tracking

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Site Tracking

Retrieve: true

Insert: false

Update: true

Delete: false

Table Columns

The columns of the table SiteTracking are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during update.

Name	Data Type	Label	Required	Documentation
enabled	boolean		☒	

4.1.56 SiteTrackingWhitelistedDomains: ActiveCampaign Site Tracking Domains

Catalog: ActiveCampaign

Schema: V3

Label: Site Tracking Domains

This is a read-only table. The ActiveCampaign API may not support changing the data or the Invantive SQL driver for ActiveCampaign does not cover it. In the latter case, please use the table NativePlatformScalarRequests to upload data to the ActiveCampaign API.

Table Columns

The columns of the table SiteTrackingWhitelistedDomains are shown below. Each column has an SQL data type.

Name	Data Type	Label	Required	Documentation
name	string	Name	☒	

4.1.57 Stages: ActiveCampaign Stages

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Stages

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Stages are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cardRegion1	string	Card Region 1	☒	
cardRegion2	string	Card Region 2	☒	
cardRegion3	string	Card Region 3	☒	
cardRegion4	string	Card Region 4	☒	
cardRegion5	string	Card Region 5	☒	
cdate	datetime	Created	☐	
color	string	Color	☐	
dealOrder	string	Deal Order	☒	
group	int64	Group	☒	
id	int64	ID	☒	
order	int64	Order	☒	
title	string	Title	☒	
update	datetime	Last Updated	☐	
width	int64	Width	☒	

4.1.58 Tags: ActiveCampaign Tags

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Tags

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Tags are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
description	string	Description	☐	
id	int64	ID	☒	
tag	string	Tag	☒	
tagType	string	Tag Type	☒	

4.1.59 Tasks: ActiveCampaign Tasks

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Tasks

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Tasks are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
assignee	int64	Assignee	☐	
automation	int64	Automation	☐	
cdate	datetime	Created	☐	
dealTasktype	int64	Task Type	☒	
doneAutomation	int64		☐	
duedate	datetime	Due Date	☒	
edate	datetime		☐	
id	int64	ID	☒	

Name	Data Type	Label	Required	Documentation
note	string	Note	☐	
owner_id	int64	Owner ID	☒	
owner_type	string	Owner Type	☒	
relid	int64	Relationship ID	☒	
reltype	string	Relationship Type	☒	
status	int64	Status	☒	
title	string	Title	☐	
update	datetime	Last Updated	☐	
user	int64	User	☒	

4.1.60 TaskTypes: ActiveCampaign Task Types

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Task Types

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table TaskTypes are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☐	
defduration	int64	Default Duration	☒	
id	int64	ID	☒	
status	int64	Status	☒	
title	string	Title	☒	
update	datetime	Last Updated	☐	

4.1.61 TrackEvents: ActiveCampaign Track Event

Catalog: ActiveCampaign

Schema: V3

Label: Track Event

Documentation:

Insert-only table to upload events to ActiveCampaign. Errors in values provided are in general not signaled due to limitations in ActiveCampaign error handling.

Retrieve: false

Insert: true

Update: false

Delete: false

Table Columns

The columns of the table TrackEvents are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
actid	int64		☒	Your unique tracking account ID (can be found on the Integrations page .../app/settings/tracking).
email	string	Email Address	☐	Contact email address to associate the event with.
event	string(128)		☒	
eventdata	string		☐	The event value (related to the event name).
key	string(40)		☒	

4.1.62 TrackingLogsByContactId: ActiveCampaign Tracking Logs by Contact ID

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Tracking Logs by Contact ID

Retrieve: true

Insert: true

Update: false

Delete: true

Parameters of Table Function

The following parameters can be used to control the behaviour of the table function TrackingLogsByContactId. A value must be provided at all times for required parameters, but optional parameters in general do not need to have a value and the execution will default to a pre-defined behaviour. Values can be specified by position and by name. In both cases, all parameters not specified will be evaluated using their default values.

Value specification by position is done by listing all values from the first to the last needed value. For example: a ``select * from table(value1, value2, value3)`` on a table with four parameters will use the default value for the fourth parameter and the specified values for the first three.

Value specification by name is done by listing all values that require a value. For example with ``select * from table(name1 => value1, name3 => value3)`` on the same table will use the default values for the second and fourth parameters and the specified values for the first and third.

Name	Data Type	Required	Default Value	Documentation
id	int64	☒		ID of the record.

Columns of Table Function

The columns of the table function TrackingLogsByContactId are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert.

Name	Data Type	Label	Required	Documentation
contact	int64	Contact	☒	
hash	string	Hash	☐	
id	int64	ID	☒	
subscriberid	int64		☒	
tstamp	datetime		☒	
type	string	Type	☒	
value	string	Value	☒	

4.1.63 Users: ActiveCampaign Users

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Users

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Users are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
email	string	Email Address	☒	
firstName	string	First Name	☒	
group	int64	Group	☐	
id	int64	ID	☒	
lastName	string	Last Name	☐	
password	string		☐	
phone	string	Phone	☐	
signature	string	Signature	☐	
username	string	User Name	☒	

4.1.64 WebhookEvents: ActiveCampaign Webhook Events

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Webhook Events

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table WebhookEvents are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
name	string	Name	☒	

4.1.65 Webhooks: ActiveCampaign Webhooks

Catalog: ActiveCampaign

Schema: V3

Primary Keys: id

Label: Webhooks

Can retrieve data and change data using insert, update and delete.

Table Columns

The columns of the table Webhooks are shown below. Each column has an SQL data type. A new non-null value must be provided for every required column at all times during insert and update.

Name	Data Type	Label	Required	Documentation
cdate	datetime	Created	☒	
events	string	Events	☒	
id	int64	ID	☒	
listid	int64	List ID	☒	
name	string	Name	☒	
sources	string	Sources	☒	
url	string	URL	☒	

Index

- A -

acceptsMarketing 49
 Account 17, 21, 36, 43, 44
 Account (URL) 16, 19
 Account by ID 16
 Account Contacts 21
 Account Contacts by Contact ID 17
 Account Custom Field Values 18
 Account Custom Fields 18, 19
 Account ID 18
 AccountById 16
 AccountContactsByContactId 17
 accountCustomFieldMetumId 18
 AccountCustomFields 18
 AccountCustomFieldValues 18
 accountId 18
 account-id 2
 account-name 2
 Accounts 19
 accountUrl 16, 19
 AccountWithCustomFieldValues 19
 acn 1
 actid 67
 Action 53
 Action Data 53
 actiondata 53
 ActiveCampaign 1, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 28, 29, 31, 32, 33, 35, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 57, 59, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70
 activerss_interval 25
 activerss_items 25
 activerss_url 25
 Activity Count 43, 44
 activitycount 43, 44
 adate 36, 46
 adddate 29
 add-odata-mandatory-filters 2
 Address 53
 Address ID 25, 53
 Address Line 1 20
 Address Line 2 20
 address1 20
 address2 20
 Addresses 20
 addressid 25, 53
 adminTemplateCss 24
 adminTemplateHtm 24
 aid 53
 allgroup 20
 allgroups 61
 allusers 61
 analysis-enforce-row-uniqueness 2
 Analytics Campaign Name 25
 Analytics Domains 57
 Analytics Source 57
 analytics_campaign_name 25
 analytics_domains 57
 analytics_source 57
 analytics_ua 57
 anonymized 46
 api-url 2
 assigned_date 48
 Assignee 48, 66
 Associations 21
 autoassign 61
 Auto-assign 61
 Automation 25, 29, 33, 45, 48, 66
 Automation Entry Counts by Contact ID 21
 AutomationEntryCountsByContactId 21
 Automations 22
 autosyncLog 33
 avgProductCategory 49
 avgRevenuePerOrder 49

- B -

Base Message ID 25
 Base Template ID 25
 basemessageid 25
 basetemplateid 25
 Batch ID 29
 batchid 29
 BLOB Preferred 14
 BLOB_PREFERRED 14
 Body 53, 61
 BOL_RESPONSE_CACHE_MAX_AGE_SEC 14
 bounce 23
 Bounce ID 23, 25, 39
 Bounce Logs by Contact ID 23
 Bounced Date 36, 46
 Bounced Hard 36, 46
 Bounced Soft 36, 46
 bounced_date 36, 46
 bounced_hard 36, 46
 bounced_soft 36, 46
 bounceid 23, 25, 39

BounceLogsByContactId 23
 Brandings 24
 bulk-delete-page-size-rows 2
 bulk-insert-page-size-bytes 2
 bulk-insert-page-size-rows 2
 Button 53

- C -

Calendars 25
 campaign 23, 33
 campaignid 23, 39, 54
 Campaigns 25
 Carbon Copy 57
 carboncopy 57
 Card Region 1 65
 Card Region 2 65
 Card Region 3 65
 Card Region 4 65
 Card Region 5 65
 cardRegion1 65
 cardRegion2 65
 cardRegion3 65
 cardRegion4 65
 cardRegion5 65
 Category 50
 cdate 22, 25, 28, 31, 36, 38, 39, 40, 41, 43, 44, 45, 46, 47, 48, 52, 53, 57, 59, 61, 62, 63, 65, 66, 67, 70
 Character Set 59
 charset 59
 City 20
 codeid 23, 39
 Color 65
 cols 40
 Company Name 20
 companyName 20
 Completed 29
 completedElements 29
 completeValue 29
 Connection 49, 50
 Connection ID 49, 50
 connectionid 49, 50
 Connections 28
 Contact 17, 21, 23, 29, 31, 33, 38, 39, 40, 41, 43, 44, 52, 54, 62, 63, 68
 Contact Automations 29
 Contact Automations by Contact ID 29
 Contact Bounce Logs 39
 Contact Custom Fields 39
 Contact Data by Contact ID 31
 Contact Tags 40
 ContactAutomations 29
 ContactAutomationsByContactId 29
 contactCount 16
 ContactDataByContactId 31
 ContactDealsByContactId 31
 contactEntryCount 21
 ContactGoalsByContactId 32
 ContactListsByContactId 33
 ContactLogsByContactId 35
 Contacts 36, 53
 ContactTagsByContactId 38
 ContactWithBounceLogs 39
 ContactWithCustomFieldValues 39
 ContactWithTags 40
 Content Type 14
 CONTENT_TYPE 14
 Copyright 24
 Country 20
 Created by 23, 31, 33, 38, 39, 40, 46, 52
 created_by 23, 31, 33, 38, 39, 40, 46, 52
 created_timestamp 19, 23, 31, 33, 38, 39, 40, 46
 created_utc_timestamp 46, 48
 createdDate 50
 createdseriesid 48
 createdTimestamp 16, 17, 18, 19, 21, 42
 Currency 43, 44, 50, 61
 Custom Field ID 18, 42
 Custom Field Values 41
 Custom Fields 40
 custom_field_currency_type 19
 custom_field_currency_value 19
 custom_field_date_value 19
 custom_field_id 19
 custom_field_number_value 19
 custom_field_text_blob 19
 custom_field_text_value 19
 Customer 50
 Customer Account 43, 44
 Customer ID 50
 customer_account_id 19
 customerAccount 43, 44
 customerid 50
 customFieldId 18, 42
 CustomFields 40
 CustomFieldValues 41

- D -

dataAction 45
 Database Driver 1
 dataId 45

dataOldval 45
 dataType 45
 DATE_ENDED 14
 DATE_STARTED 14
 Deal 31, 45, 62, 63
 Deal Activities 45
 Deal Contacts 46
 Deal Custom Field Values 42
 Deal Custom Fields 42
 Deal ID 42
 Deal Notes 47
 Deal Order 65
 Deal Tasks 48
 dealCount 16
 dealCustomFieldMetumId 42
 DealCustomFields 42
 DealCustomFieldValues 42
 dealId 42
 dealOrder 65
 Deals 43, 53
 Deals by Contact ID 31, 44
 DealsByContactId 44
 dealTasktype 48, 66
 DealWithActivities 45
 DealWithContacts 46
 DealWithNotes 47
 DealWithTasks 48
 Default 20
 Default Duration 67
 Default Screenshot 22, 53
 Default Value 40
 defaultscreenshot 22, 53
 defduration 67
 defval 40
 Deleted 25, 36, 45, 46, 57
 deleted_at 46
 deletestamp 25, 57
 descript 40, 55, 62
 Disabled 43, 44
 Discount Amount 50
 discountAmount 50
 Display Order 18, 42
 displayOrder 18, 42
 District 20
 done_by 48
 done_utc_timestamp 48
 doneAutomation 48, 66
 donedate 48
 download-error-400-bad-request-max-tries 2
 download-error-400-bad-request-sleep-initial-ms 2
 download-error-400-bad-request-sleep-max-ms 2
 download-error-408-request-timeout-max-tries 2
 download-error-408-request-timeout-sleep-initial-ms 2
 download-error-408-request-timeout-sleep-max-ms 2
 download-error-408-request-timeout-sleep-multiplicator 2
 download-error-422-bad-request-max-tries 2
 download-error-422-bad-request-sleep-initial-ms 2
 download-error-422-bad-request-sleep-max-ms 2
 download-error-422-bad-request-sleep-multiplicator 2
 download-error-429-too-many-requests-max-tries 2
 download-error-429-too-many-requests-sleep-initial-ms 2
 download-error-429-too-many-requests-sleep-max-ms 2
 download-error-429-too-many-requests-sleep-multiplicator 2
 download-error-502-server-unavailable-max-tries 2
 download-error-502-server-unavailable-sleep-initial-ms 2
 download-error-502-server-unavailable-sleep-max-ms 2
 download-error-502-server-unavailable-sleep-multiplicator 2
 download-error-503-server-unavailable-max-tries 2
 download-error-503-server-unavailable-sleep-initial-ms 2
 download-error-503-server-unavailable-sleep-max-ms 2
 download-error-503-server-unavailable-sleep-multiplicator 2
 download-error-504-gateway-timeout-max-tries 2
 download-error-504-gateway-timeout-sleep-initial-ms 2
 download-error-504-gateway-timeout-sleep-max-ms 2
 download-error-504-gateway-timeout-sleep-multiplicator 2
 download-error-590-network-connect-timeout-max-tries 2
 download-error-590-network-connect-timeout-sleep-initial-ms 2
 download-error-590-network-connect-timeout-sleep-max-ms 2
 download-error-590-network-connect-timeout-sleep-multiplicator 2
 download-error-599-network-connect-timeout-max-tries 2
 download-error-599-network-connect-timeout-sleep-initial-ms 2

download-error-599-network-connect-timeout-sleep-max-ms	2	Due Date	48, 66
download-error-599-network-connect-timeout-sleep-multiplicator	2	duedate	48, 66
download-error-argument-exception-max-tries	2	duration	48
download-error-argument-exception-sleep-initial-ms	2	Duration (ms)	14
download-error-argument-exception-sleep-max-ms	2	DURATION_MS	14
download-error-argument-exception-sleep-multiplicator	2		
download-error-internet-down-max-tries	2	eComCustomers	49
download-error-internet-down-sleep-initial-ms	2	eCommerce Customers	49
download-error-internet-down-sleep-max-ms	2	eCommerce Order	50
download-error-internet-down-sleep-multiplicator	2	eCommerce Order Products	50
download-error-io-exception-max-tries	2	eCommerce Orders	50
download-error-io-exception-sleep-initial-ms	2	ecomOrder	50
download-error-io-exception-sleep-max-ms	2	eComOrderProducts	50
download-error-io-exception-sleep-multiplicator	2	eComOrders	50
download-error-json-exception-max-tries	2	ed_instanceid	59
download-error-json-exception-sleep-initial-ms	2	ed_version	59
download-error-json-exception-sleep-max-ms	2	edate	36, 43, 44, 46, 48, 66
download-error-json-exception-sleep-multiplicator	2	email	23, 36, 39, 46, 49, 50, 67, 69
download-error-other-exception-max-tries	2	Email Address	23, 36, 39, 46, 49, 50, 67, 69
download-error-other-exception-sleep-initial-ms	2	email_domain	36, 46
download-error-other-exception-sleep-max-ms	2	email_empty	46
download-error-other-exception-sleep-multiplicator	2	email_local	36, 46
download-error-socket-exception-max-tries	2	Embed Images	25
download-error-socket-exception-sleep-initial-ms	2	embed_images	25
download-error-socket-exception-sleep-max-ms	2	enabled	64
download-error-socket-exception-sleep-multiplicator	2	Encoding	59
download-error-web-exception-max-tries	2	End Date	14
download-error-web-exception-sleep-initial-ms	2	entered	22
download-error-web-exception-sleep-max-ms	2	entries	53
download-error-web-exception-sleep-multiplicator	2	error	23, 39
download-error-web-not-implemented-max-tries	2	Error Message Code	14
download-error-web-not-implemented-sleep-initial-ms	2	Error Message Text	14
download-error-web-not-implemented-sleep-max-ms	2	ERROR_MESSAGE_CODE	14
download-error-web-not-implemented-sleep-multiplicator	2	ERROR_MESSAGE_TEXT	14
download-error-web-timeout-max-tries	2	event	67
download-error-web-timeout-sleep-initial-ms	2	eventdata	67
download-error-web-timeout-sleep-max-ms	2	event-key	2
download-error-web-timeout-sleep-multiplicator	2	Events	51, 70
download-error-web-unauthorized-max-tries	2	exclude	36
download-error-web-unauthorized-sleep-initial-ms	2	exited	22
download-error-web-unauthorized-sleep-max-ms	2	External ID	28, 49, 50
download-error-web-unauthorized-sleep-multiplicator	2	externalcheckoutid	50
DRY_RUN	14	externalCreatedDate	50
		externalid	28, 49, 50
		Facebook	25

facebook_session 57
 Fail on Error 14
 FAIL_ON_ERROR 14
 Favicon 24
 field 39, 41, 52
 Field Default Currency 18, 42
 Field Default Value 18, 42
 Field Label 18, 42
 Field Options 18, 42
 Field Type 18, 42
 Field Value 18, 42
 Field Values by Contact ID 52
 fieldDefault 18, 42
 fieldDefaultCurrency 18, 42
 fieldLabel 18, 42
 fieldOptions 18, 42
 fieldType 18, 42
 fieldValue 18, 42
 FieldValuesByContactId 52
 First Name 36, 46, 69
 first_name 33
 firstName 36, 46, 69
 Footer (HTML) 24
 Footer (text) 24
 footerHtmlValue 24
 footerTextValue 24
 force-case-sensitive-identifiers 2
 forced-casing-identifiers 2
 form 33
 Form ID 25
 Form Visible 18, 42
 Format 59
 formid 25, 36
 Forms 53
 forwards 25
 From Name 59
 fromemail 59
 fromname 59
 Full Address 57
 fulladdress 57

- G -

Geo IP-addresses by Contact ID 54
 geoAddress 54
 geoaddrid 54
 GeolpsByContactId 54
 get_unsubscribe_reason 57
 Goals by Contact ID 32
 Gravatar 36, 46
 Group 43, 44, 65, 69

Group ID 24
 groupid 24
 Groups 55

- H -

hardbounces 25
 Hash 36, 44, 46, 68
 Header (HTML) 24
 Header (text) 24
 headerHtmlValue 24
 headerTextValue 24
 Help 24
 Hidden 21, 22, 59, 64
 Hide Field 18, 42
 hideFieldFlag 18, 42
 HTML 59
 htmlfetch 59
 htmlunsub 25
 htmlunsubdata 25
 HTTP Disk Cache Maximum Age (sec) 14
 HTTP Memory Cache Maximum Age (sec) 14
 HTTP Method 14
 HTTP Status Code 14
 HTTP_DISK_CACHE_MAX_AGE_SEC 14
 HTTP_DISK_CACHE_SAVE 14
 HTTP_DISK_CACHE_USE 14
 HTTP_MEMORY_CACHE_MAX_AGE_SEC 14
 HTTP_MEMORY_CACHE_SAVE 14
 HTTP_MEMORY_CACHE_USE 14
 HTTP_METHOD 14
 HTTP_STATUS_CODE 14
 http-disk-cache-compression-level 2
 http-disk-cache-directory 2
 http-disk-cache-ignore-write-errors 2
 http-disk-cache-max-age-sec 2
 http-get-timeout-max-ms 2
 http-get-timeout-ms 2
 http-memory-cache-compression-level 2
 http-memory-cache-max-age-sec 2
 http-post-timeout-max-ms 2
 http-post-timeout-ms 2

- I -

id 16, 17, 21, 23, 29, 31, 32, 33, 35, 38, 44, 52, 54, 59, 62, 68
 ignore-http-400-errors 2
 ignore-http-401-errors 2
 ignore-http-402-errors 2
 ignore-http-403-errors 2

ignore-http-404-errors 2
 ignore-http-422-errors 2
 ignore-http-429-errors 2
 ignore-http-500-errors 2
 ignore-http-502-errors 2
 ignore-http-503-errors 2
 Image (URL) 50
 imageUrl 50
 in_group_lists 36
 Internal 28
 invalid-json-on-get-max-tries 2
 invalid-json-on-get-sleep-initial-ms 2
 invalid-json-on-get-sleep-max-ms 2
 invalid-json-on-get-sleep-multiplicator 2
 invalid-json-on-post-max-tries 2
 invalid-json-on-post-sleep-initial-ms 2
 invalid-json-on-post-sleep-max-ms 2
 invalid-json-on-post-sleep-multiplicator 2
 invantive-sql-compress-sparse-arrays 2
 invantive-sql-correct-invalid-date 2
 invantive-sql-forward-filters-to-data-containers 2
 invantive-sql-share-byte-arrays 2
 invantive-sql-share-strings 2
 invantive-sql-shuffle-fetch-results-data-containers
 invantive-use-cache 2
 ip 36, 46
 IP Address 36, 46
 ip4 25, 54
 ip4_last 33
 ip4Sub 33
 ip4Unsub 33
 IPv4 Address 25, 54
 is_draft 47, 59
 isAddtask 45
 isDefault 20
 isDisabled 43, 44
 isFormVisible 18, 42
 isInternal 28
 isRequired 18, 40, 42

- J -

Job Title 17, 21
 jobTitle 17, 21
 join-set-points-per-request 2

- K -

key 67
 Known Field ID 18, 42

knownFieldId 18, 42

- L -

Last Block 29
 Last Date 29
 Last Log ID 29
 Last Modified 16, 17, 18, 19, 21, 22, 25, 42, 43, 44, 47, 59, 61, 62
 Last Name 36, 46, 69
 Last Sent User ID 61
 Last Step 25
 Last Synchronization 28
 Last Updated 28, 33, 36, 39, 40, 41, 46, 48, 50, 52, 53, 57, 61, 65, 66, 67
 last_name 33
 last_sent_user_id 61
 lastblock 29
 lastdate 29
 lastlogid 29
 lastName 36, 46, 69
 laststep 25
 lastSync 28
 layout 53
 ldate 25, 61
 License 24
 limit-partition-calls-left 2
 Link (URL) 28
 Link Clicks 25
 linkclicks 25
 linkUrl 28
 list 33
 List ID 70
 listid 36, 70
 Lists 57
 Lists by Contact ID 33
 logic 64
 log-native-calls-to-disk-max-events 2
 log-native-calls-to-disk-max-seconds 2
 log-native-calls-to-disk-on-error 2
 log-native-calls-to-disk-on-success 2
 log-native-calls-to-trace 2
 Logo (URL) 28
 logoUrl 28
 Logs by Contact ID 35

- M -

Mail Cleanup 25
 Mail Send 25
 Mail Transfer 25

mail_cleanup 25
 mail_send 25
 mail_transfer 25
 Mailer Log File 25
 mailer_log_file 25
 Manage Text 25
 managetext 25
 maximum-length-identifiers 2
 max-odata-filters 2
 max-url-length-accepted 2
 max-url-length-desired 2
 mdate 22, 25, 43, 44, 47, 59, 61, 62
 message 23, 33
 messageid 23, 39, 54
 Messages 59
 metadata-cache-max-age-sec 2

- N -

Name 16, 19, 21, 22, 25, 28, 50, 51, 53, 57, 59, 62, 64, 65, 70
 Native Platform Scalar Requests 14
 NATIVEPLATFORMSCALARREQUESTS 14
 Next Date 43, 44
 Next Deal ID 43, 44
 Next Task ID 43, 44
 nextdate 43, 44
 nextdealid 43, 44
 nextTask 44
 nexttaskid 43, 44
 Note 47, 48, 59, 66
 Notes by Contact ID 59
 NotesByContactId 59
 Notification 25
 npt 14
 Number of Contacts 16
 Number of Deals 16
 Number of Opens 25

- O -

oauth-unauthorized-max-tries 2
 oauth-unauthorized-sleep-initial-ms 2
 oauth-unauthorized-sleep-max-ms 2
 oauth-unauthorized-sleep-multiplicator 2
 opens 25
 Opt-in Confirm 55
 Opt-in Created 53
 Opt-in ID 53
 Opt-in Message ID 57
 Opt-in Opt-out 57

optinconfirm 55
 optinmessageid 57
 optinoptout 57
 options_blanks_overwrite 53
 options_confaction 53
 options_conform 53
 options_optin_created 53
 options_optin_id 53
 options_sendoptin 53
 optoutconf 57
 Order 65
 Order Date 50
 Order ID 50
 Order Number 40
 Order Products 50
 Order URL 50
 orderDate 50
 orderid 50
 ordernum 40
 orderProducts 50
 orderUrl 50
 organization 36, 43, 44
 Organization ID 36, 46
 orgid 36, 46
 orgname 46
 ORIG_SYSTEM_GROUP 14
 ORIG_SYSTEM_REFERENCE 14
 Original System Group 14
 Original System Reference 14
 outcome_id 48
 outcome_info 48
 Overwrite Blanks 53
 Owner 41, 43, 44, 52
 Owner ID 47, 59, 66
 Owner Type 47, 59, 66
 owner_id 47, 59, 66
 owner_type 47, 59, 66

- P -

p_admin 55
 p_embed_image 57
 p_use_analytics_link 57
 p_use_analytics_read 57
 p_use_captcha 57
 p_use_facebook 57
 p_use_tracking 57
 p_use_twitter 57
 Parent Form ID 53
 parentformid 53
 partition-slot-based-rate-limit-length-ms 2

partition-slot-based-rate-limit-slots	2	pgStartupGettingstarted	55
password	69	pgStartupReports	55
Payload	14	pgTemplateAdd	55
PAYLOAD_TEXT	14	pgTemplateDelete	55
percent	43, 44	pgTemplateEdit	55
Percentage	43, 44	Phone	36, 46, 69
Personalization	18, 42	Pipelines	61
perstag	40	Pre Header Text	59
pg_user_add	55	preheader_text	59
pg_user_delete	55	pre-request-delay-ms	2
pg_user_edit	55	Preview Data	59
pgAutomationManage	55	Preview MIME Type	59
pgContactActions	55	preview_data	59
pgContactAdd	55	preview_mime	59
pgContactApprove	55	Price	50
pgContactDelete	55	Priority	59
pgContactEdit	55	Private	57
pgContactExport	55	Product URL	50
pgContactFields	55	productUrl	50
pgContactFilters	55	Public	25
pgContactImport	55	Public Template (CSS)	24
pgContactMerge	55	Public Template (HTML)	24
pgContactSync	55	publicTemplateCss	24
pgDeal	55	publicTemplateHtm	24
pgDealDelete	55		
pgDealGroupAdd	55		
pgDealGroupDelete	55		
pgDealGroupEdit	55		
pgDealReassign	55		
pgFormEdit	55		
pgGroupAdd	55		
pgGroupDelete	55		
pgGroupEdit	55		
pgListAdd	55		
pgListBounce	55		
pgListDelete	55		
pgListEdit	55		
pgListEmailaccount	55		
pgListHeaders	55		
pgMessageAdd	55		
pgMessageDelete	55		
pgMessageEdit	55		
pgMessageSend	55		
pgPersonalizationAdd	55		
pgPersonalizationDelete	55		
pgPersonalizationEdit	55		
pgReportsCampaign	55		
pgReportsList	55		
pgReportsTrend	55		
pgReportsUser	55		
pgSavedResponsesManage	55		

- Q -

Quantity 50

- R -

Rating Date 36, 46
rating_tstamp 36, 46
Real CID 25
realcid 25
Recurring 25
Relationship ID 47, 48, 59, 66
Relationship Type 47, 48, 59, 62, 66
relid 47, 48, 59, 66
reltype 47, 48, 59, 62, 66
remdate 29
reminder_field 25
reminder_format 25
reminder_last_cron_run 25
reminder_offset 25
reminder_offset_sign 25
reminder_offset_type 25
reminder_type 25
Replies 25
reply2 59

replsys 25
 reqApproval 55
 reqApproval1st 55
 reqApprovalNotify 55
 requested-page-size 2
 requests-parallel-max 2
 require_name 57
 Required 18, 40, 42
 responder 33
 responder_existing 25
 responder_offset 25
 responder_type 25
 Response Cache Maximum Age (sec) 14
 Result BLOB 14
 Result Date Time 14
 Result Number 14
 Result Text 14
 RESULT_BLOB 14
 RESULT_DATE_TIME_UTC 14
 RESULT_NUMBER 14
 RESULT_TEXT 14
 retry-event-upload 2
 role 31
 Rows 40
 Run without Actions 14

- S -

Save HTTP Disk Cache 14
 Save HTTP Memory Cache 14
 Saved Responses 61
 SavedResponses 61
 Schedule 25
 Scheduled Date 25
 scheduleddate 25
 score 62
 Score Values by Contact ID 62
 Scores 62
 scoreValue 62
 ScoreValuesByContactId 62
 Screenshot 22
 sdate 25, 33, 55
 Secondary Contacts 63
 SecondaryContacts 63
 Segment ID 25
 segmentid 25, 36
 segmentio_id 36, 46
 Segments 64
 Send Amount 25
 Send Last Broadcast 57
 Send Opt-in 53

send_amt 25
 send_last_broadcast 57
 Sender Address Line 1 57
 Sender Address Line 2 57
 Sender City 57
 Sender Country 57
 Sender Name 57
 Sender Phone 57
 Sender Reminder 57
 Sender State 57
 Sender URL 57
 Sender ZIP Code 57
 sender_addr1 57
 sender_addr2 57
 sender_city 57
 sender_country 57
 sender_name 57
 sender_phone 57
 sender_reminder 57
 sender_state 57
 sender_url 57
 sender_zip 57
 sendid 25
 sentcnt 36, 46
 sentiment 46
 Series ID 25, 29, 33, 45, 64
 seriesid 25, 29, 33, 36, 45, 64
 Service 28, 40
 Shipping Amount 50
 Shipping Method 50
 shippingAmount 50
 shippingMethod 50
 Show in List 40
 show_in_list 40
 Signature 69
 simulate-http-400-errors 2
 simulate-http-400-errors-percentage 2
 simulate-http-401-errors 2
 simulate-http-401-errors-percentage 2
 simulate-http-403-errors 2
 simulate-http-403-errors-percentage 2
 simulate-http-408-errors 2
 simulate-http-408-errors-percentage 2
 simulate-http-429-errors 2
 simulate-http-429-errors-percentage 2
 simulate-http-500-errors 2
 simulate-http-500-errors-percentage 2
 simulate-http-502-errors 2
 simulate-http-502-errors-percentage 2
 simulate-http-503-errors 2
 simulate-http-503-errors-percentage 2

simulate-http-protocol-errors	2	Style Background	53
simulate-http-protocol-errors-percentage	2	Style Border Color	53
simulate-http-timeout-errors	2	Style Border Radius	53
simulate-http-timeout-errors-percentage	2	Style Border Style	53
Site Logo	24	Style Border Width	53
Site Logo Small	24	Style Button Background	53
Site Name	24	Style Button Font Color	53
Site Tracking	64	Style Button Padding	53
Site Tracking Domains	65	Style Font Color	53
siteLogo	24	Style Layout	53
siteLogoSmall	24	Style Width	53
siteName	24	style_ac_branding	53
SiteTracking	64	style_background	53
SiteTrackingWhitelistedDomains	65	style_border_color	53
SKU	50	style_border_radius	53
slot-based-rate-limit-length-ms	2	style_border_style	53
slot-based-rate-limit-slots	2	style_border_width	53
Social Data	55	style_button_background	53
Social Data Last Check	36, 46	style_button_border_color	53
Social Shares	25	style_button_border_radius	53
socialdata	55	style_button_border_style	53
socialdata_lastcheck	36, 46	style_button_border_width	53
socialshares	25	style_button_fontcolor	53
Soft Bounces	25	style_button_padding	53
softbounces	25	style_dark	53
sortdate	45	style_fontcolor	53
Source	23, 25, 39, 50	style_layout	53
sourceid	33	style_width	53
Sources	70	Subject	59, 61
Split Content	25	Submit	53
Split Offset	25	Submit Data URL	53
Split Offset Type	25	submitdata_url	53
Split Type	25	Subscriber Clicks	25
split_content	25	subscriberclicks	25
split_offset	25	subscriberid	23, 39, 68
split_offset_type	25	Subscription Notify	57
split_type	25	subscription_notify	57
split_winner_awaiting	25	Succesful	14
split_winner_messageid	25	SUCCESSFUL	14
Stage	43, 44, 45	Survey	25
Stages	65	sync	33
standardize-identifiers	2	Synchronization Status	28
standardize-identifiers-casing	2	syncStatus	28
Start Date	14	- T -	
Start ID	29		
startid	29		
State	20, 50		
Status	21, 22, 25, 28, 29, 33, 36, 43, 44, 48, 62, 66, 67		
String ID	57	Tag	36, 38, 40, 66
stringid	57	Tag Type	66
		Tags	66
		Tags by Contact ID	38
		tagType	66

Task Type 48, 66
 Task Types 67
 Tasks 66
 TaskTypes 67
 Tax Amount 50
 taxAmount 50
 Text 59
 textfetch 59
 textunsub 25
 textunsubdata 25
 Thanks 53
 Thread ID 25
 threadid 25
 Timeout (sec) 14
 TIMEOUT_SEC 14
 Timespan 29
 Title 25, 40, 43, 44, 48, 53, 55, 61, 65, 66, 67
 To Name 57
 to_name 57
 Token 25
 Total Amount 25
 Total Elements 29
 Total Orders 49
 Total Price 50
 Total Products 49, 50
 Total Revenue 49
 total_amt 25
 totalElements 29
 totalOrders 49
 totalPrice 50
 totalProducts 49, 50
 totalRevenue 49
 Track Event 67
 Track Link Analytics 25
 Track Links 25
 Track Reads 25
 Track Reads Analytics 25
 TrackEvents 67
 Tracking Logs by Contact ID 68
 TrackingLogsByContactId 68
 tracklinks 25
 tracklinksanalytics 25
 trackreads 25
 trackreadsanalytics 25
 Transaction ID 14
 TRANSACTION_ID 14
 tstamp 23, 39, 49, 50, 54, 68
 Tweet 25
 Twitter Token 57
 Twitter Token Secret 57
 twitter_token 57

twitter_token_secret 57
 Type 25, 40, 68

- U -

ua 36, 46
 udate 28, 33, 36, 39, 40, 41, 46, 48, 52, 53, 57, 61, 65, 66, 67
 Unique Forwards 25
 Unique Link Clicks 25
 Unique Opens 25
 Unique Replies 25
 uniqueforwards 25
 uniquelinkclicks 25
 uniqueopens 25
 uniquereplies 25
 unsubreason 33
 unsubreasons 25
 Unsubscribe Link 55
 unsubscribeAutomation 33
 unsubscribelink 55
 Unsubscribes 25
 Unsubscription Notify 57
 Unsubscription Reasons 25
 unsubscription_notify 57
 Updated 19, 23, 31, 33, 38, 39, 40, 46
 Updated by 23, 31, 33, 38, 39, 40, 46, 48, 52
 updated_by 23, 31, 33, 38, 39, 40, 46, 48, 52
 updated_timestamp 19, 23, 31, 33, 38, 39, 40, 46
 updated_utc_timestamp 46, 48
 updatedAt 50
 updatedTimestamp 16, 17, 18, 19, 21, 42
 Updates 25
 URL 14, 53, 70
 Use HTTP Disk Cache 14
 Use HTTP Memory Cache 14
 use-batch-insert 2
 use-http-disk-cache-read 2
 use-http-disk-cache-write 2
 use-http-memory-cache-read 2
 use-http-memory-cache-write 2
 User 25, 45, 47, 48, 57, 59, 66
 User ID 22, 25, 47, 53, 57, 59
 User Name 69
 userid 22, 25, 47, 53, 57, 59
 username 69
 Users 69

- V -

Value 39, 41, 43, 44, 52, 68

Version 24

Visible 40

- W -

Wait Preview 25

waitid 36

waitpreview 25

Webhook Events 70

WebhookEvents 70

Webhooks 70

Width 65

Will Recur 25

willrecur 25

Win Probability 43, 44

Win Probability Last Updated 43, 44

winProbability 43, 44

winProbabilityMdate 43, 44

- Z -

zip 20

ZIP Code 20



invantive ***the SQL company***

Invantive B.V.
Biesteweg 11
3849 RD Hierden
the Netherlands

Tel: +31 88 00 26 500
Fax: +31 84 22 58 178
info@invantive.com
invantive.com

IBAN NL25 BUNQ 2098 2586 07
Chamber of Industry and Commerce
13031406
VAT NL812602377B01
RSIN 8122602377
Managing Director: Guido Leenders
Registered office: Roermond