



# **TIBCO® Data Virtualization**

## **Salesforce.com Adapter Guide**

Version 8.7.0 | October 2023

# Contents

---

<b>Contents</b>	<b>2</b>
<b>Using the Salesforce.com Adapter</b>	<b>3</b>
Salesforce.com Limitations	3
Salesforce.com Basic Tab	4
Salesforce.com Advanced Tab	4
Salesforce.com Identity Confirmation Security Feature	7
Security for the Salesforce.com Adapter	8
TDV SQL Support for Salesforce.com	8
<b>TIBCO Product Documentation and Support Services</b>	<b>25</b>
How to Access TIBCO Documentation	25
How to Contact TIBCO Support	26
Release Version Support	26
How to Join TIBCO Community	27
<b>Legal and Third-Party Notices</b>	<b>28</b>

# Using the Salesforce.com Adapter

---

Salesforce.com provides a flat, single-level hierarchy of resources. The resources are called SObjects in Salesforce.com terminology. SObjects are represented as relational tables in TDV.

Introspecting resources is time-intensive, so as a general rule select only the resources necessary for your project. If you find you need additional resources later, you can always add them using the Add/Remove Resources in Studio.

This topic describes how to use the Salesforce.com data source adapter with TDV.

- [Salesforce.com Limitations](#)
- [Salesforce.com Basic Tab](#)
- [Salesforce.com Advanced Tab](#)
- [Salesforce.com Identity Confirmation Security Feature](#)
- [Security for the Salesforce.com Adapter](#)
- [TDV SQL Support for Salesforce.com](#)

## Salesforce.com Limitations

### Query Limitation

Salesforce.com has limitations on queries. Performance issues will occur if you do NOT restrict the number of results. According to the Salesforce.com documentation, the expected limit on query results is 2,000. However, that limit is subject to change with each version of Salesforce.com, you are responsible to determine what the current limit is for your version of Salesforce.com and adjust your query limits as necessary.

[Queries to Salesforce need to include a WHERE clause that restricts the number or records to that limit. For example, 2,000. Otherwise, the queries can time out or performance is adversely affected.](#)

## Application Views for Salesforce.com

The organization of the pre-built application views and folders in the Salesforce.com Adapter is patterned after the user interface of an uncustomized Salesforce.com account. Each folder within the `DataServicesForSalesforce.com` folder corresponds to a tab in Salesforce.com. Each view within the folder corresponds to one of the default views in Salesforce.com.

## Filter Data from Application Views

There are two ways to filter data from Salesforce.com using application views:

- using a filter provided by Salesforce.com itself
- by filtering within TDV.

For more information about filters, refer to [TDV SQL Support for Salesforce.com](#).

## Salesforce.com Basic Tab

For Connection Information on the Basic tab, you might need to use a secure password to access the Salesforce.com data source. See [Salesforce.com Identity Confirmation Security Feature](#) for how to do this.

## Salesforce.com Advanced Tab

Click the Advanced tab to display a panel of advanced properties for the data source. These properties are specific to TDV and how it interacts with Salesforce.com:

Advanced Tab Field	Description
Connection timeout (sec)	Timeout for the HTTPS connection to Salesforce.com. The default is 60 seconds. This is not a session timeout. Salesforce.com session timeout (default 120 minutes) is configurable at Setup   Security Controls on the Salesforce.com website.

Advanced Tab Field	Description
User Concurrent Calls Limit	The maximum number of concurrent calls for the current user. The default is 3.
Concurrent Calls Limit	The maximum number of concurrent calls for this data source. The default is 5. This setting prevents TDV from emitting too many concurrent requests to Salesforce.com. For example, if set at 5 but there are 20 views scheduled to trigger concurrently, TDV will queue 15 views. Typically, this limit should not be increased, but might be decreased in case there are several Salesforce.com data sources configured.
Query batch size	The maximum number of rows returned from a single request to Salesforce.com. The default is 500 rows. The maximum query batch size is 2,000 records; however the maximum is automatically reduced if you select large text fields or many fields. This behavior is implemented by Salesforce.com on their servers to maintain performance. Even if a query batch size of 200 is selected, Salesforce.com reserves the right to return more or fewer records. For example, the batch size will be no more than 200 if a query selects two or more custom fields of type long text. This is to prevent large SOAP messages from being returned. The Salesforce.com driver for TDV makes as many requests as necessary to Salesforce.com to fully evaluate a query, regardless of the number of rows returned per request by Salesforce.com. For example, if the batch size is set to 2,000 rows (and assuming Salesforce.com honors this number of rows) but there are 4,000 rows in the object and a query spans all of them, TDV issues two requests to Salesforce.com.
Maximum retries	Since Salesforce.com is a collection of Web services accessed over the Internet using HTTPS, connections are inherently unreliable. They may timeout or fail altogether. To improve reliability of queries built upon this infrastructure, the Maximum retries can be set to automatically retry any Salesforce.com request upon connectivity failure. The default is 0, meaning the feature is disabled and no retries are attempted.
Wait between retries (sec)	If the maximum retries feature is enabled, this property specifies the

Advanced Tab Field	Description
	number of seconds to wait between retry attempts. It defaults to 0.
Proxy enabled	If this checkbox is selected, all requests to Salesforce.com for this data source are passed through a Web proxy specified by the properties Proxy host, Proxy port, and optionally Proxy username and Proxy password.
Proxy host	Host name or IP address of Web proxy. The proxy is used only if the Proxy enabled property is selected.
Proxy port	Port of Web proxy. The proxy is used only if the Proxy enabled property is selected.
Proxy username and Proxy password	Specifies the username and password to be used to authenticate to the selected Web proxy.
Batch updates	Salesforce.com limits the number of records that can be impacted with a single update request to 200. If this option is disabled (the default), Salesforce.com throws an exception if this limit is exceeded. If this option is enabled, updates impacting more than 200 rows are batched to Salesforce.com to workaround the limit. The batches are not transactional, so if the first batch succeeds but the second fails, the first batch still causes an update of Salesforce.com records.
Batch deletes	Salesforce.com limits the number of records that can be impacted with a single delete request to 200. If this option is disabled (the default), Salesforce.com throws an exception if this limit is exceeded. If this option is enabled, deletes impacting more than 200 rows are batched to Salesforce.com to workaround the limit. The batches are not transactional, so if the first batch succeeds but the second fails, the first batch still causes a delete of Salesforce.com records.
URL	This property specifies the location of the Salesforce.com environment. It contains two preset values and allows custom values to be provided. By default it is set to the first preset value in the list, the URL of the production Salesforce.com environment. It can also be set to the

Advanced Tab Field	Description
	Salesforce.com Sandbox (the second preset URL), or to the URL of a customer-provided, system on the premises.
	Typically the SFDC URL is similar to: <a href="https://cs5.salesforce.com/">https://cs5.salesforce.com/</a>

1. Expand the node representing the Salesforce.com data source, and select the boxes corresponding to the resources.

## Salesforce.com Identity Confirmation Security Feature

Salesforce.com has provided an additional identity confirmation security feature that might be required to access the Salesforce.com data source. It is possible that you need to implement this additional feature if you receive this error:

```
SForceException: Error [sforce-29000000]: LOGIN_MUST_USE_SECURITY_TOKEN
```

To access Salesforce.com through a desktop application or other API-based application, you must replace your current password with a combination of your password and a security token.

### To implement the Salesforce.com security feature

**Procedure** Log in to Salesforce.com through the browser to request your security token.

1. At the top of any Salesforce page, click the down-arrow next to your name. From the menu under your name, select Setup or My Settings—whichever one appears.
2. Go to Setup > My Personal Information > Reset Security Token.

or

Go to My Settings > Personal > Reset My Security Token.

3. Click the Reset Security Token button to trigger an email that contains your security token.

4. Select and copy the token from the email.
5. In the application, replace your password with combination of the password and the security token. For example, if your password is MyPassword and your security token is XXXXXX, you would enter MyPasswordXXXXXX in the password field.

See [http://trust.salesforce.com/trust/security/identity\\_feature.html](http://trust.salesforce.com/trust/security/identity_feature.html) for more details.

## Security for the Salesforce.com Adapter

### Protecting Salesforce.com Resources

TDV supports the complete set of data security measures offered by Salesforce.com. Security can be configured at a variety of levels: per user, object, field within object, or operation (such as INSERT). All security features are configured using the Salesforce.com Web site's administration screens, and become effective immediately to TDV users.

You can use Salesforce.com's identity confirmation security feature with your password, as described in [Salesforce.com Identity Confirmation Security Feature](#).

With pass-through security, you can avoid exposing a Salesforce.com Administrator's access rights to TDV users. Users simply log in to TDV with their Salesforce.com user credentials, which are forwarded to Salesforce.com. This ensures that the proper Salesforce.com data security is applied to every user, and that credentials are never stored within TDV.

For information about setting pass-through login, see the TDV User Guide.

## TDV SQL Support for Salesforce.com

Salesforce.com exposes functionality through a set of SOAP RPC services operating on a namespace of data objects called SObjects. Leveraging these services and a query language called SOQL (Sforce Object Query Language), the Salesforce.com Adapter provides SQL access to Salesforce.com as a relational data source.

The following sections describe how Salesforce.com resources operate within TDV. This topic describes how TDV interprets and supports Salesforce.com SQL statements and objects.



- [Introspection](#)
- [Capabilities](#)
- [Joins](#)
- [Multipicklists with SQL](#)
- [Lead Conversion](#)
- [Updated and Deleted Objects](#)
- [Packaged Query](#)
- [Data Modification in Salesforce.com](#)

## Introspection

### Resource Hierarchy

The Salesforce.com Adapter provides Salesforce.com SObjects as resources within TDV. SObjects are introspected by selecting them by name from a flat name space. If you're not sure which SObject contains the data you need, examine the tabs on the Salesforce.com Web site or consult its documentation.

### Metadata Mapping

Each field of the Salesforce.com SObject becomes a column in TDV with the same name. Each data type in Salesforce.com is mapped to a TDV SQL-based type. The following table lists the Salesforce.com data types, whether they are supported in TDV, and their corresponding TDV types.

Salesforce.com Type Name	Supported?	TDV Data Type
STRING, TEXTAREA, PHONE, URL, EMAIL, COMBOBOX, PICKLIST, MULTIPICKLIST	Yes	VARCHAR
BOOLEAN	Yes	BIT
INT	Yes	BIGINT

Salesforce.com Type Name	Supported?	TDV Data Type
DOUBLE, CURRENCY, PERCENT	Yes	DECIMAL, NUMERIC
DATE	Yes	DATE
DATETIME	Yes	TIMESTAMP
BASE64	Yes	VARCHAR
ID, REFERENCE	Yes	VARCHAR(18)

## Capabilities

Capabilities are a system of classifying the unique features and limitations of data sources. For example, an Oracle data source can execute subqueries, but Salesforce.com cannot. Capabilities are consulted when a query is processed so that data sources receive only the query processing work they support. Where a capability is lacking in a data source but required to run a query, TDV performs the work itself.

The following table lists commonly used capabilities and how they apply to Salesforce.com. The Supported column indicates whether or not the SQL capability is supported in queries against Salesforce.com. The Pushed column indicates whether the capability is supported directly on Salesforce.com, allowing the query processing work to be off-loaded to it. For efficient queries, minimize use of non-pushed capabilities.

Capability	Supported?	Pushed?	Notes
CASE	Yes	No	
DELETE	Yes	Yes	See <a href="#">Data Modification in Salesforce.com</a>
DISTINCT	No	No	
Filter	Yes	Yes	Filters comparing two columns of the same SObject are not allowed by Salesforce.com.

Capability	Supported?	Pushed?	Notes
			If a filter is applied to a column that has the “filterable” attribute set to false, the query is executed in TDV instead of being pushed to the Salesforce data source.
Filter – BETWEEN	Yes	Yes	
Filter – IN	Yes	Yes	See <a href="#">Multipicklists with SQL</a>
Filter – LIKE	Yes	Yes	
Functions – Aggregate	Yes	No	
Functions – CAST	Yes	Yes	
Function – ConvertCurrency	Yes	Yes	Salesforce.com-specific function.  Currency management must be enabled in Salesforce.com for this to function properly. Salesforce.com constraints on the use of ConvertCurrency apply. See Salesforce.com documentation for more information.
Functions – Others	Yes	No	
Function – ToLabel	Yes	Yes	Salesforce.com-specific function.
GROUP BY	Yes	No	
INSERT	Yes	Yes	See <a href="#">Data Modification in Salesforce.com</a>

Capability	Supported?	Pushed?	Notes
JOIN	Yes	Yes	See <a href="#">Joins</a> for details about join types that are pushed.
ORDER BY	Yes	No	
Subquery	Yes	No	
Transactions	No	No	
UNION	Yes	No	
UPDATE	Yes	Yes	See <a href="#">Data Modification in Salesforce.com</a>

## Joins

When it is possible to optimize the performance of queries with joins, TDV uses any join information it finds to convert standard SQL to SOQL and pushes this to the source Salesforce.com database at run time (that is, when data is actually retrieved). Specifically, queries with inner and outer joins that have parent-to-child or child-to-parent relationships are pushed to Salesforce.com at run time. If these relationships do not exist, the joins are processed in TDV at run time.

Note: You can disable pushing joins to Salesforce.com. See [Disabling Pushing Join Execution](#). If you import a CAR file with Salesforce.com resources from a TDV release prior to 6.2 SP2, you must reintrospect the resources to take advantage of the join pushing capabilities.

Join relationships are evaluated when a Salesforce.com data source is created and introspected. If any parent-to-child or child-to-parent relationships are discovered in any tables during introspection, the child relationships are collected for each table and inserted in the Annotation field :

|

This information is for user information only and cannot be edited. If the Annotation field is empty, no child relationships exist for this table.

Parent relationships are displayed as foreign key relationships in TDV in the table's Foreign Keys tab. The foreign and primary key relationships are used at run time.

## Joins Pushed to Salesforce.com

TDV can push right outer joins, left outer joins, and inner joins to Salesforce.com. Examples of how TDV translates these joins from SQL into SOQL are provided below.

### Right Outer Join

Find all opportunity records and their related accounts.

#### SQL:

```
SELECT
```

```
o.Id, o.Name,
```

```
A.Id, A.Name
```

```
FROM /shared/QA_SForce/Sources/SForce/Account a right outer join
```

```
/shared/QA_SForce/Sources/SForce/Opportunity o
```

```
on a.id = o.AccountId
```

#### SOQL:

```
Select Id, Name, Account.Id, Account.Name
```

```
From Opportunity
```

## Left Outer Join

Find all the accounts and their related opportunity records.

### SQL:

```
SELECT  
  
    A.Id, A.Name,  
  
    o.AccountId, o.Name  
  
FROM /shared/QA_SForce/Sources/SForce/Account a left outer join  
  
    /shared/QA_SForce/Sources/SForce/Opportunity o  
  
    on a.id = o.AccountId
```

### SOQL:

```
Select Id, Name, (Select AccountId, Name From Opportunities)  
  
From Account
```

## Inner Join

Find all accounts where there exists an opportunity record.

### SQL:

```
SELECT
```

```
A.Id, A.Name
```

```
FROM /shared/QA_SForce/Sources/SForce/Account a inner join
```

```
/shared/QA_SForce/Sources/SForce/Opportunity o
```

```
on a.id = o.AccountId
```

SOQL:

```
Select Id, Name
```

```
From Account
```

```
Where Id In
```

```
(Select AccountId from Opportunity)
```

## Disabling Pushing Join Execution

TDV is configured to automatically push any joins it can to Salesforce.com. You can disable pushing join execution to Salesforce.com this way:

- Use the `DISABLE_PUSH` option in your query, as in this example:

```
SELECT {option DISABLE_PUSH}
```

```
o.Id, o.Name,
```

```
A.Id, A.Name
```

```
FROM /shared/QA_SForce/Sources/SForce/Account a right outer join
```

```
/shared/QA_SFForce/Sources/SForce/Opportunity o
```

```
on a.id = o.AccountId
```

## Semijoins

A semijoin is the best way to reduce the number of Salesforce.com rows retrieved and processed by TDV, thus improving query performance. To force a semijoin to occur in a query, add it immediately before the table to be joined. For example:

```
SELECT * FROM A INNER JOIN { option semijoin } B ON A.K = B.K
```

Values of A.K are collected and passed in a query to B as the filter.

```
SELECT * FROM A
```

```
SELECT * FROM B WHERE K IN ({values of A.K from previous query})
```

If there are many rows in A, this can result in lengthy queries against B. TDV automatically partitions the queries against B if they become too large. Partitioning means the query is broken up into smaller queries that are executed separately and reassembled to produce a unified result set.

Put the larger table on the right side of the join whenever possible. When running a new query for the first time, display the Execution Plan in Studio (click Show Execution Plan in the resource editor) and then click Execute and Show Statistics. Examine each node's row count and query after processing has begun to make sure filters are pushed down to Salesforce.com. This is a good way to see the mechanics of a semijoin in action. If the interaction between TDV and Salesforce.com is still unclear and performance is poor, enable debug logging as described in the *TDV Installation and Upgrade Guide*. This can help illustrate how a SQL statement on TDV becomes a series of requests to Salesforce.com.

## Multipicklists with SQL

Multipicklists are a data type within Salesforce.com which allow the user to select multiple values from a list of valid values. When selected in SQL queries within TDV, Multipicklists are displayed as a series of values delimited by semicolons. For example, if the user has selected Bob and Mary from a Multipicklist, the value visible to TDV is Bob;Mary.



To treat the selected values as separate for the purposes of filtering, Salesforce.com provides special operators to their query language. This section describes how to structure your SQL query in TDV to leverage these operators.

## Equal

When the Multipicklist field uses the equality operator, it is converted to the includes operator when issued to Salesforce.com. For example, the following SQL statement returns rows where the field Multipicklist\_\_c includes Value1 in its selection:

```
SELECT * FROM TestObject1__c  
  
WHERE MultiPickList__c = 'Value1'
```

This query results in the following filter on the query issued to Salesforce.com:

```
WHERE MultiPickList__c includes ('Value1')
```

## Not Equal

When the Multipicklist field uses the inequality operator, it is converted to the NOT and includes operators when issued to Salesforce.com. For example, the following SQL statement returns rows where the field Multipicklist\_\_c includes Value1 in its selection:

```
SELECT * FROM TestObject1__c  
  
WHERE MultiPickList__c != 'Value1'
```

This query results in the following filter on the query issued to Salesforce.com:

```
WHERE NOT (MultiPickList__c includes ('Value1'))"
```

## Includes

A Multipicklist field includes a value if the field's selection includes the value among its selected values. The IN keyword is used to instruct TDV that the INCLUDES syntax should

be pushed to Salesforce.com. For example, the following SQL statement returns rows where the field Multipicklist\_\_c includes Value1 and Value2 in its selection:

```
SELECT * FROM TestObject1__c  
  
WHERE MultiPickList__c IN ('Value1', 'Value2')
```

This query results in the following filter on the query issued to Salesforce.com:

```
WHERE MultiPickList__c includes ('Value1','Value2')
```

## Excludes

A Multipicklist field excludes a value if the field's selection does not include the value among its selected values. The NOT IN keyword is used to instruct TDV that the EXCLUDES syntax should be pushed to Salesforce.com. For example, the following SQL statement returns rows where the field Multipicklist\_\_c does not include Value1 and Value2 in its selection:

```
SELECT {OPTION DISABLE_PUSH}  
  
*  
  
FROM TestObject1__c  
  
WHERE MultiPickList__c NOT IN ('Value1', 'Value2')
```

Note: The DISABLE\_PUSH option is required to ensure that Salesforce.com does not return NULLs.

This query results in the following filter on the query issued to Salesforce.com:

```
WHERE MultiPickList__c excludes ('Value1','Value2')
```

## Lead Conversion

Salesforce.com provides a ConvertLead API call which converts a Lead into an Account, Contact, or Opportunity. The stored procedure named convertLead within the Salesforce.com Data Source can be leveraged to access this functionality.

To convert a Lead, the Salesforce.com data source must be logged into Salesforce.com with Convert Leads permission and Edit permission on Leads, as well as sufficient permissions to create or update, as applicable, any associated Account, Contact, and Opportunity.

The details involved in converting a lead are outside the scope of this document. For more information, see the Salesforce API documentation in the Developer section of the Salesforce.com Web site.

## Updated and Deleted Objects

Salesforce.com provides stored procedures named GetUpdated and GetDeleted, which are API calls that return IDs of specified objects updated or deleted between dates in a specified range.

To be accessed through GetUpdated and GetUpdated, an object must be configured so that it can be replicated. Results are returned for no more than 30 days previous to the day the call is executed. They need to return the IDs of Contacts that have been updated or deleted on the current date.

## Upsert

The Upsert API call creates new records and updates existing records. Use Upsert instead of Create to avoid creating unwanted duplicate records.

Upsert uses a custom field to determine the presence of existing records. All records must be of the same object type. Maximum array size is 200.

## Inputs

externalIdFieldName: Determines whether it should create a new record or update an existing one.

objectType: The record type.

dataSql: SQL Used to generate the new records. Each row in the resulting result set corresponds to a record in SalesForce. The dataCursor input can be used instead.

**dataCursor:** Input cursor used to generate the new records. Each row in the resulting result set corresponds to a record in Salesforce. The **dataSql** input can be used instead.

**fields:** Comma-separated list of fields in the record. If this is NULL, fields are extracted from the **dataSql** or **dataCursor** if possible. Otherwise, an error occurs.

## Outputs

Upsert outputs a cursor containing the Upsert results:

**id:** Salesforce ID of the record.

**isCreated:** Boolean indicating if the record was newly created or not.

## Merge

Merge merges records of the same object type into one of the records, deletes the others, and adjusts the parenting of related records.

## Inputs

**objectType:** The master record type. The only supported types are Lead, Contact, and Account.

**masterRecordId:** The ID of the master record that others are merged into.

**recordToMergeIds:** Comma-separated list of the records to merge into the master record.

## Outputs

A cursor containing the merge results.

## Packaged Query

TDV supports packaged queries against Salesforce.com using SOQL.

This support allows for the building of views on top of the packaged queries to flatten the nested results. If relevant metadata is available, it is pushed down using joins.

## Data Modification in Salesforce.com

INSERT, UPDATE, and DELETE operations can be performed directly against any Salesforce.com data source. In TDV, this is done through a SQL Script. For details on SQL scripts, see the *TDVReference Manual*.

Data modification in Salesforce.com is governed by a set of business rules often defined on a per SObject basis, which are beyond the scope of this document to address in depth. This section contains a summary of some rules, paraphrased from the Salesforce.com API documentation.

If a data modification request to Salesforce.com is not processed as expected, it may be due to a business rule on the Salesforce.com server. For more information about the specifics of updating Salesforce.com SObjects, consult the Salesforce API documentation, available in the Developer section of the Salesforce.com Web site.

- [INSERT INTO](#)
- [UPDATE](#)
- [DELETE](#)

### INSERT INTO

The INSERT INTO statement creates a new SObject instance of a particular type. The type is determined by the resource provided as the table to the command. Following the resource name, a list of field names and values provide the data to initialize the new object.

For example:

```
INSERT INTO
/shared/DataServicesForSalesforce.com/Sources/Salesforce.com/Contact
```

```
(FirstName, LastName, MailingStreet, MailingCity, MailingState,
MailingPostalCode,
```

```
MailingCountry, Phone)
```

```
VALUES ('Marc', 'Benioff',
```

```
'1 Market St Suite 300', 'San Francisco', 'CA', '94105', 'US',
```

```
'415-555-1212');
```

Below are a few of the areas to consider while developing an INSERT INTO statement:

- Account security. The Salesforce.com account used by TDV must have sufficient access rights to create objects of the specified type.
- Object and field security. Some objects and certain fields within those objects require special handling or permissions. For example, you might also need permissions to access this object's parent object. Some objects cannot be created through the API.
- Generated fields. Salesforce.com generates unique values for some fields automatically. You cannot explicitly specify an ID, CreatedDate, CreatedById, LastModifiedDate, LastModifiedById, and SystemModstamp.
- Required fields. For required fields that do not have a preconfigured default value, you must supply a value.
- Default values. For some objects, some fields have a default value. If you do not specify a value for such fields, Salesforce.com populates these fields with the default value.
- Referential integrity. If you are creating an object that is the child of a parent object, you must supply the foreign key information that links the child to the parent.
- Number of records inserted at a time. When doing an insert into Salesforce.com, TDV always uses batch inserts of no more than 200 records at a time.

## UPDATE

The UPDATE statement changes one or more SObject instances, based on a filter and list of fields and values. The SObject type is determined by the resource provided as the table to the command. Following the resource name, a list of field-value pairs detail the changes to be performed on the selected objects. Finally a filter specifies the conditions to be satisfied for an update to take place on a given row.

For example:

### UPDATE

```
/shared/DataServicesForSalesforce.com/Sources/Salesforce.com/Contact
```

```
SET "Phone" = '415-555-9999'
```

```
WHERE "FirstName" = 'Marc' AND "LastName" = 'Benioff';
```

Below are a few of the areas to consider while developing an UPDATE statement:

- Account security. The Salesforce.com account used by TDV must have sufficient access rights to update objects of the specified type.
- Object and field security. Some objects and certain fields within those objects require special handling or permissions. For example, you might also need permissions to access this object's parent object. Some objects cannot be updated through the API.
- Generated fields. Salesforce.com generates unique values for some fields automatically. You cannot explicitly update an ID, CreatedDate, CreatedById, LastModifiedDate, LastModifiedById, and SystemModstamp.
- Required fields. When updating required fields, you must supply a value—you cannot set the value to null.
- Referential integrity. Fields whose names contain “Id” are either that object's primary key or a foreign key. Salesforce.com does not allow the update of primary keys, but foreign keys can be updated.
- Object change limit. Salesforce.com allows up to 200 objects to be changed in a single request. If an update request exceeds 200 objects, the entire operation fails (unless the Advanced data source property Batch Updates is enabled).

## DELETE

The DELETE statement removes one or more SObject instances based on a filter. The SObject type is determined by the resource provided as the table to the command. Following the resource name, a filter specifies the conditions to be satisfied for a delete to take place on a given row. For example:

```
DELETE FROM
/shared/DataServicesForSalesforce.com/Sources/Salesforce.com/Contact
```

```
WHERE "FirstName" = 'John' AND "LastName" = 'Doe';
```

These are a few of the areas to consider while developing a DELETE statement:

Area	Consideration
Account security.	The Salesforce.com account used by TDV must have sufficient access rights to delete objects of the specified type.
Object and field security.	You may need permissions to access this object's parent object. Some objects cannot be deleted through the API.
Referential integrity.	To ensure referential integrity, the DELETE call supports cascading deletes; that is, if you delete a parent object, you delete its children automatically, as long as each child object can be deleted. If the advanced data source property Batch Deletes is enabled, requests exceeding 200 objects can be handled.



# TIBCO Product Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The [TIBCO Product Documentation](#) website is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The following documentation for this product is available on the [TIBCO® Data Virtualization](#) page.

- **Users**
  - TDV Getting Started Guide
  - TDV User Guide
  - TDV Web UI User Guide
  - TDV Client Interfaces Guide
  - TDV Tutorial Guide
  - TDV Northbay Example
- **Administration**
  - TDV Installation and Upgrade Guide
  - TDV Administration Guide
  - TDV Active Cluster Guide
  - TDV Security Features Guide
- **Data Sources**

TDV Adapter Guides

TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**

TDV Reference Guide

TDV Application Programming Interface Guide

- **Other**

TDV Business Directory Guide

TDV Discovery Guide

- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

## How to Contact TIBCO Support

Get an overview of [TIBCO Support](#). You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the [TIBCO Support](#) website.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to [TIBCO Support](#) website. If you do not have a user name, you can request one by clicking **Register** on the website.

## Release Version Support

TDV 8.5 is designated as a Long Term Support (LTS) version. Some release versions of TIBCO® Data Virtualization products are selected to be long-term support (LTS) versions. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also

[https://docs.tibco.com/pub/tdv/general/LTS/tdv\\_LTS\\_releases.htm](https://docs.tibco.com/pub/tdv/general/LTS/tdv_LTS_releases.htm).

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, visit [TIBCO Community](#).

# Legal and Third-Party Notices

---

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, TIBCO logo, TIBCO O logo, ActiveSpaces, Enterprise Messaging Service, Spotfire, TERR, S-PLUS, and S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the

readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2002-2023 Cloud Software Group, Inc All Rights Reserved.