



Creating a one-to-many relationship with integrity enforcement in Caspio platform

How-to user guide

Prepared for recruitment purposes

Karolina Dziewierz

2 May 2021

Table of contents

1. Overview	3
2. Prerequisites	3
3. Step-by-step procedure.....	3
3.1. Navigation	3
3.2. Creation of a parent table	4
3.3. Creation of a child table	6
3.4. Creation of a one-to-many relationship	7
4. Referential integrity	9

1. Overview

A one-to-many relationship occurs when one record in a table (parent table) is associated with one or more records in another table (child table). For example, the relationship between a client and sales orders in which each client can have many sales orders but each order only belongs to one client. In Caspio app when creating a relationship between two tables, if only one of the related fields is unique, a one-to-many relationship is set automatically.

2. Prerequisites

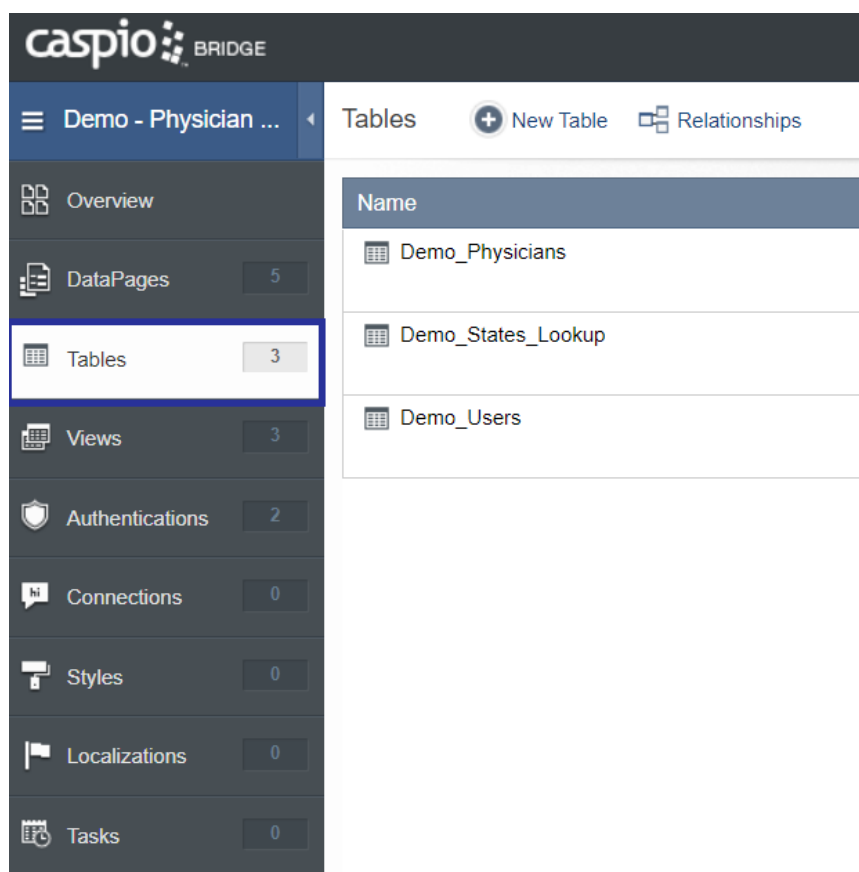
To create a one-to-many relationship in Caspio app, you need to have parent and child tables set up.

This document presents a step-by-step procedure on how to set up a parent table, child table and establish a one-to-many relationship. If you already have parent and child tables, please proceed directly to point [3.4. Creation of a one-to-many relationship](#).

3. Step-by-step procedure

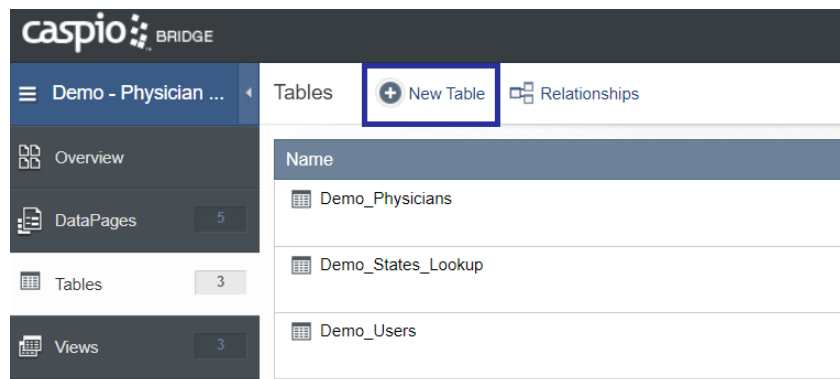
3.1. Navigation

1. Login into your Caspio account.
2. Go to the **Tables** tab.

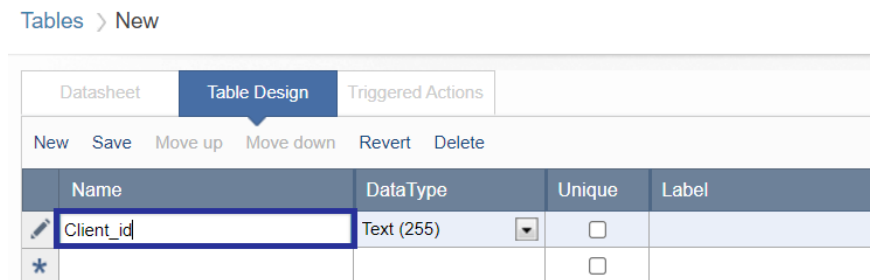


3.2. Creation of a parent table

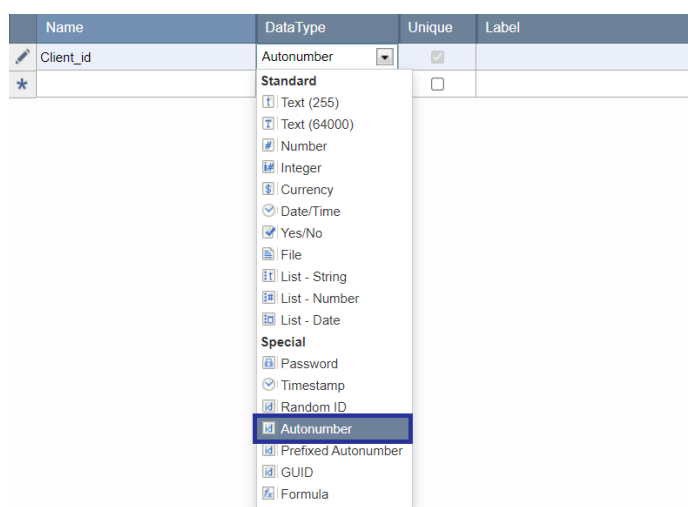
1. Click the [New table] button to create a parent table.



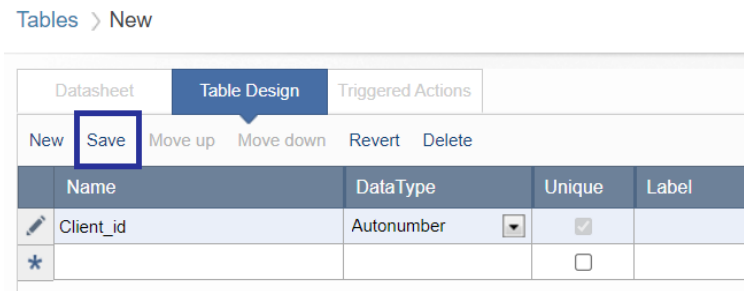
2. Insert the name of the field in the **Name** field, e.g. Client_id.



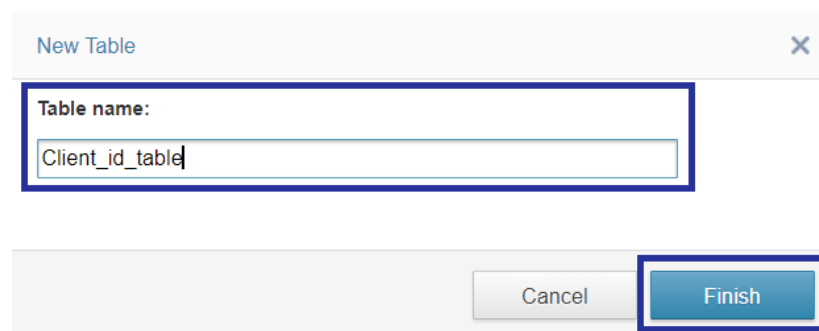
3. Set the DataType as either: **Autonumber**, **Prefixed autonumber**, **Random ID**, **GUID**. The system automatically marks this field as unique, which means that every single client in your database will have a unique identifier (so-called primary key).



- Click the **[Save]** button.

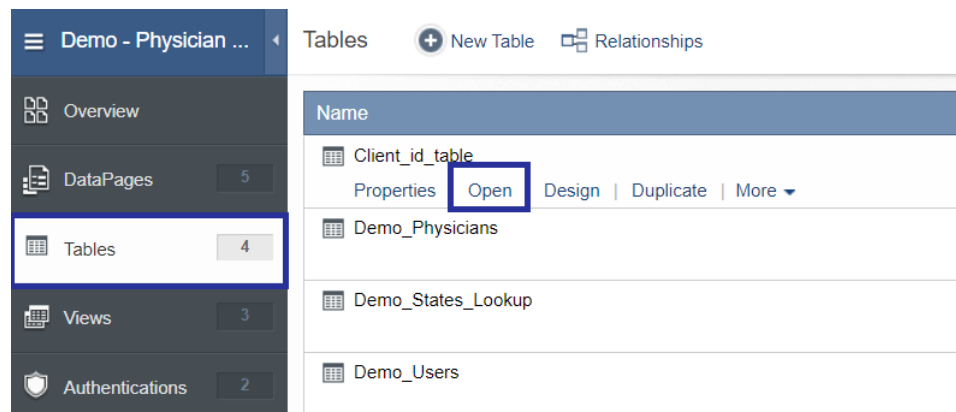


- Insert a table name, e.g. Client_id_table.
- Click the **[Finish]** button.



Adding data entries

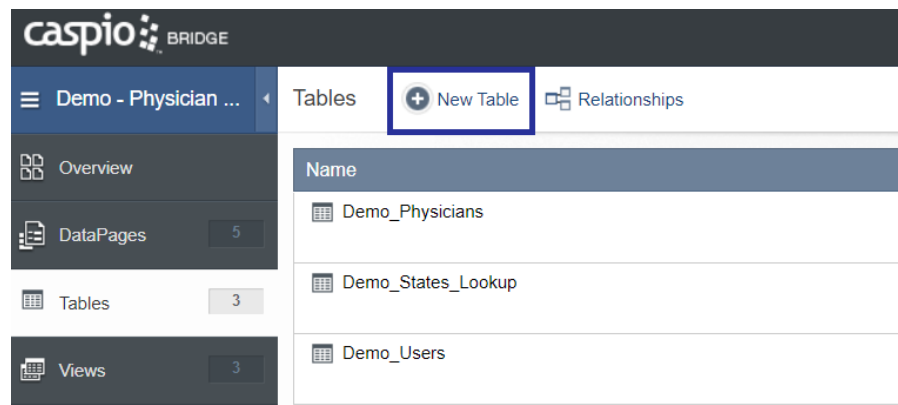
- Click the **[Open]** button.



- Fill in the **Name** fields. The system adds unique client_ids automatically.

3.3. Creation of a child table

- Click the **[New table]** button to create a child table.



- Create a foreign key that links back to the parent table (here Client_id).
- Set the DataType. For **Autonumber** it needs to be an **Integer**, for **Random ID, GUID** or **Prefixed autonumber** it needs to be **Text (255)**.

Tables > New

Datasheet Table Design Triggered Actions				
New Save Move up Move down Revert Delete				
	Name	DataType	Unique	Label
	Client_id	Integer	<input type="checkbox"/>	
*			<input type="checkbox"/>	

- List the other fields.
- Click the **[Save]** button

Tables > New

Datasheet Table Design Triggered Actions				
New Save Move up Move down Revert Delete				
	Name	DataType	Unique	Label
	Client_id	Integer	<input type="checkbox"/>	
	Sales_info	Text (255)	<input type="checkbox"/>	
*			<input type="checkbox"/>	

- Insert the **Table_name**, e.g. Sales_info.
- Click the **[Finish]** button.

Adding data entries

16. Click the **[Open]** button.
17. Add the data in the parent column (here Client_ids) which pertains to a child column (here Sales_info).
In other words we can say that Sales_info pertains to a Client with a particular id.

Tables > Table_sales_info






Datasheet		Table Design	Triggered Actions
New	Refresh	Download	Find Replace Filter Reset Autonumber Delete Delete All
	Client_id	Sales_info	
<input type="checkbox"/>	1	order#123	
<input type="checkbox"/>	2	order#568	
<input type="checkbox"/>	2	order#6387	
<input checked="" type="checkbox"/>	3	order#5896	
*			

18. Go back to the **Tables** menu.

3.4. Creation of a one-to-many relationship

19. Click the **[Relationships]** button.

Tables **+** New Table **Relationships**

Name
 Client_id_table
 Demo_Physicians
 Demo_States_Lookup
 Demo_Users
 Table_sales_info

20. Select the tables from the left **Tables** menu.

Tables > Relationships

Tables	New Relationship	All Relationships	App Relationships	Table Relations	Save Layout
In App					
<input checked="" type="checkbox"/> Client_id_table					
<input checked="" type="checkbox"/> Demo_Physicians					
<input type="checkbox"/> Demo_States_Lookup					
<input type="checkbox"/> Demo_Users					
<input type="checkbox"/> Table_sales_info					

Client_id_table

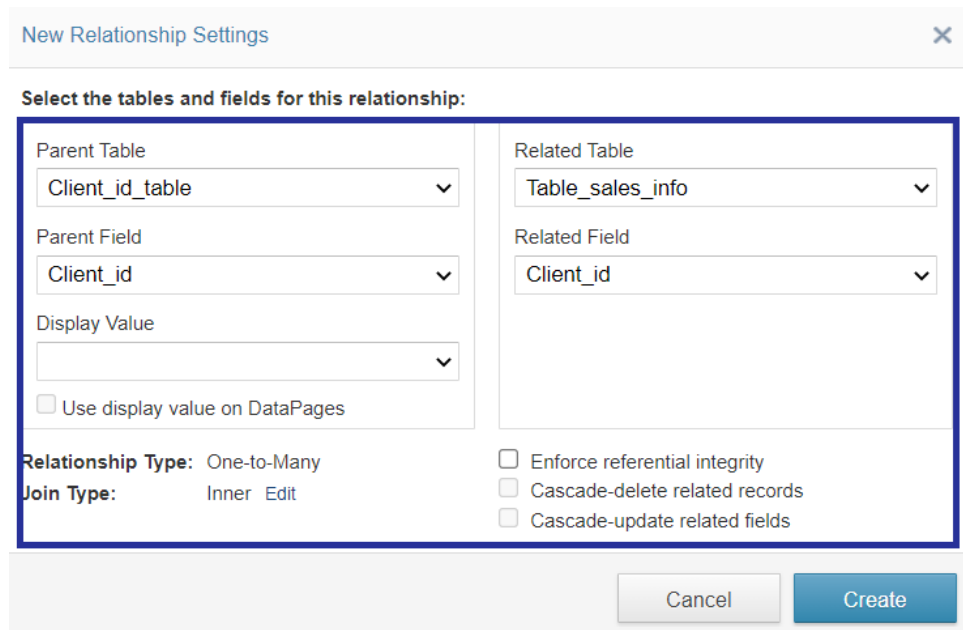
Client_id

Demo_Physicians

Physician_ID
Date_Created
First_Name
Last_Name
Title
Full_Name
Gender
Email
Discontinued

21. Click the parent table (here a Client_id), drag it over and link to the integer of the related table.

New Relationship Settings popup with configuration details appears:



The image shows a 'New Relationship Settings' dialog box. It has a title bar with the text 'New Relationship Settings' and a close button (X). The main content area is titled 'Select the tables and fields for this relationship:'. It is divided into two columns. The left column contains: 'Parent Table' with a dropdown menu showing 'Client_id_table'; 'Parent Field' with a dropdown menu showing 'Client_id'; 'Display Value' with a dropdown menu; and a checkbox labeled 'Use display value on DataPages'. The right column contains: 'Related Table' with a dropdown menu showing 'Table_sales_info'; 'Related Field' with a dropdown menu showing 'Client_id'; and three checkboxes: 'Enforce referential integrity', 'Cascade-delete related records', and 'Cascade-update related fields'. At the bottom left, there is a 'Relationship Type' section showing 'One-to-Many' and a 'Join Type' section showing 'Inner' with an 'Edit' link. At the bottom right, there are 'Cancel' and 'Create' buttons.

It includes the following fields:

Parent Table – contains parent records,

Parent Field – displays the name of a parent table,

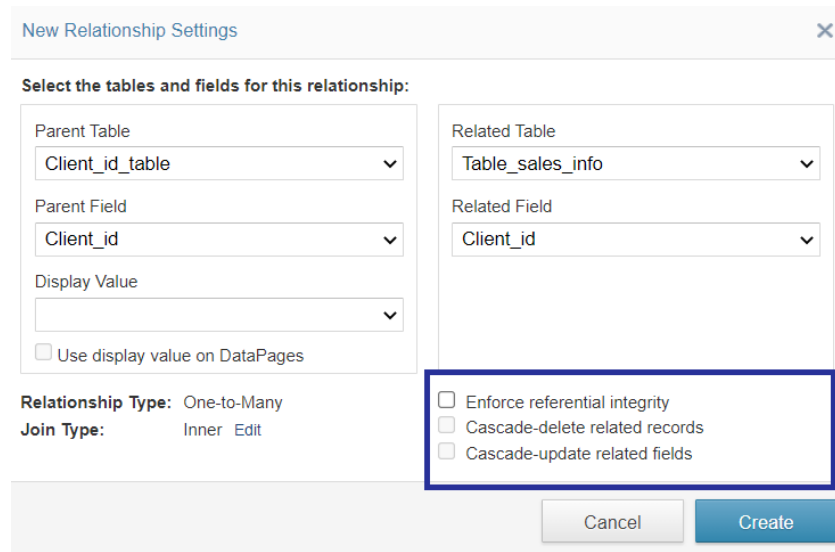
Related Table – contains child records,

Related Field – the field that links both tables,

Display Value – it allows you to choose a field from the parent table and display it in the associated field of the child table,

Relationship Type : One-to-Many – type of a relationship between the tables. Here t means that there are multiple orders related to one client.

4. Referential integrity



The image shows a 'New Relationship Settings' dialog box. It has a title bar with a close button. The main area is titled 'Select the tables and fields for this relationship:'. It contains two columns of dropdown menus. The left column is for the 'Parent Table' and the right column is for the 'Related Table'. Below these are 'Parent Field' and 'Related Field' dropdowns. There is also a 'Display Value' dropdown and a checkbox for 'Use display value on DataPages'. At the bottom, there is a 'Relationship Type' dropdown set to 'One-to-Many' and a 'Join Type' dropdown set to 'Inner'. To the right of these are three checkboxes: 'Enforce referential integrity', 'Cascade-delete related records', and 'Cascade-update related fields'. The 'Enforce referential integrity' checkbox is highlighted with a blue border. At the bottom right are 'Cancel' and 'Create' buttons.

New Relationship Settings

Select the tables and fields for this relationship:

Parent Table: Client_id_table

Parent Field: Client_id

Display Value:

☐ Use display value on DataPages

Relationship Type: One-to-Many

Join Type: Inner Edit

Related Table: Table_sales_info

Related Field: Client_id

☐ Enforce referential integrity

☐ Cascade-delete related records

☐ Cascade-update related fields

Cancel Create

Referential integrity allows you to maintain integrity between the parent table and related child records. You can check the following options:

Enforce Referential integrity – when clicked, it means that you are not allowed to delete a record from a parent table as you have related child tables,

Cascade-delete related records – when clicked, it means that if you delete a parent table, related child records are deleted as well to maintain referential integrity,

Cascade-update related tables – when clicked, it means that if you update a parent table, related child records are updated accordingly to maintain referential integrity.