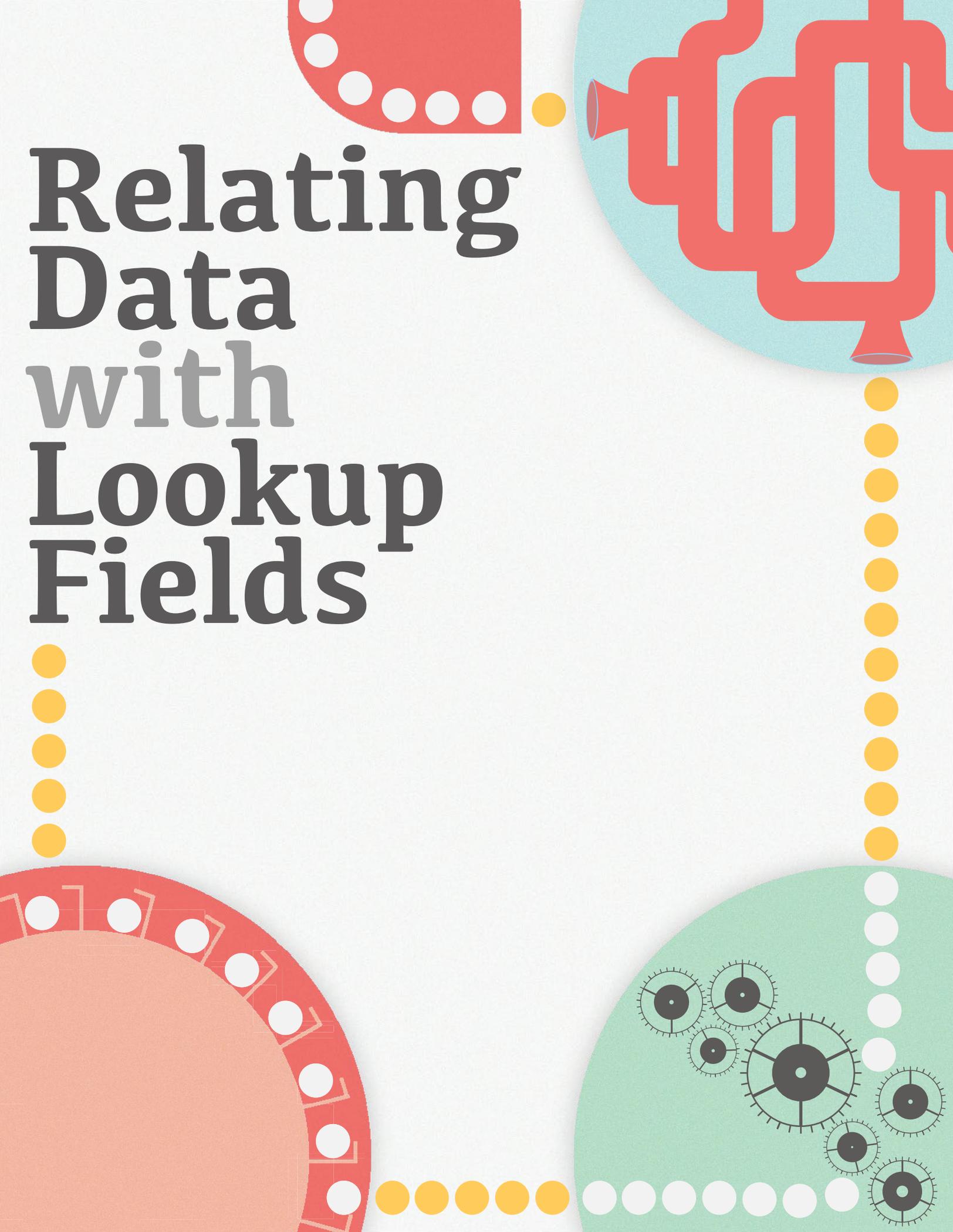


Relating Data with Lookup Fields



Table

of

Contents



 Page 3 How to use this Guide	 Page 32 Lookups and Reports
 Page 4 Data in Creator	 Page 35 Example 2
 Page 11 Relating Data in Creator	 Page 40 Automating with Lookup Fields
 Page 18 Example 1	 Page 45 Fetching Data with Lookups
 Page 24 How to Create Relationships	 Page 47 Support and Feedback

How to use this Guide

Zoho Creator allows you to build custom applications to manage your data. By creating the proper relationships amongst your data, your application will be more efficient, and ultimately, more powerful.

Whether you're new to these concepts or a veteran who just wants to learn the terminology specific to Zoho Creator, this guide is for you.

Reading this guide, you should learn:

- ✔ **How relating data works in Zoho Creator**
- ✔ **The benefits of relating data**
- ✔ **How to plan data relationships**
- ✔ **How to use lookup fields to relate data in Zoho Creator**



Data in Creator

Forms & Reports

Creator applications have two basic parts: **forms** and **reports**. When you enter information into a form and then submit it, you create a **record** within your Creator application.

Reports are where you view and update records. When you build a form, Creator automatically generates a report to go with it.

Projects

Project Name

Price \$

Start Date

Workers

Records

Project Report 🔍 + ☰

<input type="checkbox"/>	Project Name	Price	Start Date	Workers
	New Apartment	\$ 2,000,000.00	06-Jun-2016	Charles Fields Bobby Henderson
	Renovation	\$ 80,000.00	21-Jun-2016	Joe Henderson
	Storefront Demo	\$ 11,000.00	06-Jul-2016	Jack Washington Benjamin Gutierrez
	Floor Addition	\$ 120,000.00	11-Jul-2016	Kimberly Day Randy Sims
	Major Renovation	\$ 140,000.00	25-Jul-2016	Donald Willis Ernest Howell

Forms are for collecting information.

Reports are for viewing and analyzing information.

You can visualize a Creator report as a **table** or **spreadsheet**.

- Each **field** is a **column**
- Each **record** is a **row**

	A	B	C	D
1	Project Name	Price	Start Date	Workers
2	New Apartment	2000000	06-Jun-2016	Charles Fields,Bobby Henderson
3	Renovation	80000	21-Jun-2016	Joe Henderson
4	Storefront Demo	11000	06-Jul-2016	Jack Washington,Benjamin Gutierrez
5	Floor Addition	120000	11-Jul-2016	Kimberly Day,Randy Sims
6	Major Renovation	140000	25-Jul-2016	Donald Willis,Ernest Howell

Relational Databases

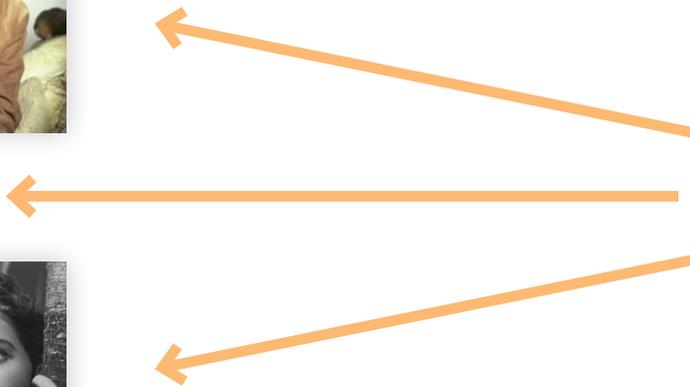
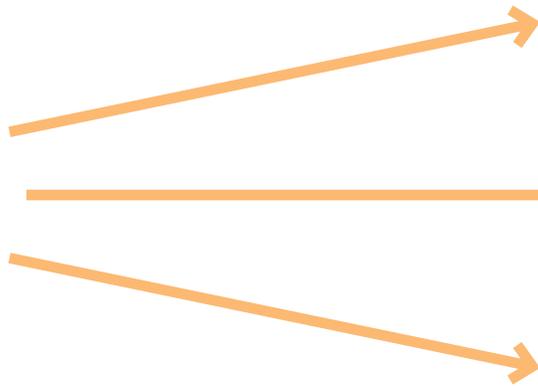
 The phrase “relational database” sounds complicated, but really it’s just a technical term for something we encounter all the time.

For example, every time you view your credit card statement, you’re looking at information from a **relational database**.

Each month, millions of purchases are made with millions of credit cards. In order to search through that vast amount of information and find just the purchases relevant to you, a database **relates** your card number with your purchases.

Let’s see another example.

Think about a movie database like IMDB. By **relating data**, it lets you look up all the movies that one actor is in or all the actors that are in one movie.



Now let's look at a business example. A manager uses spreadsheets to keep track of employee information and current projects. By separating project and employee information and then relating data that's needed in both places, the manager can efficiently organize her department.

If an employee resigns, the manager can delete that person's information from the employee spreadsheet, which automatically removes them from the "Workers" column in the Projects spreadsheet. Relating Employees and Projects would also let the manager look up an employee and see all the projects they're working on.

Projects

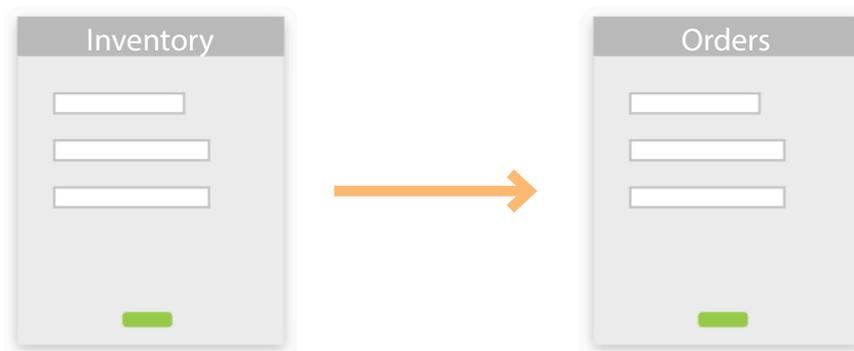
	A	B	C	D
1	Project Name	Price	Start Date	Workers
2	New Apartment	2000000	06-Jun-2016	Charles Fields,Bobby Henderson
3	Renovation	80000	21-Jun-2016	Joe Henderson

Employees

	A	B	C	D
1	Name	Email	Phone Number	Job
2	Ernest Howell	ehowell9@si.edu	(806)229-4754	Demolition
3	Benjamin Gutierrez	bgutierrez8@earthlink.net	(192)267-2482	Demolition
4	Anna Payne	apayne7@adobe.com	(361)591-5184	Carpenter
5	Randy Sims	rsims6@state.gov	(850)459-4410	Field Manager
6	Jack Washington	jwashington5@unesco.org	(400)449-0426	Carpenter
7	Joe Henderson	jhenderson4@privacy.gov.au	(280)550-4604	Carpenter
8	Kimberly Day	kday3@meetup.com	(691)809-9236	Carpenter
9	Donald Willis	dwillis2@google.com.br	(432)753-7678	Demolition
10	Bobby Henderson	bhenderson1@wordpress.org	(267)131-3893	Cement Mason
11	Charles Fields	cfields0@wikia.com	(210)982-5610	Field Manager

Two more quick examples:

Consider an online business that keeps track of all its customer orders and all the inventory in its warehouse. By relating their **order form** to their **inventory form**, they can make sure customers only order items that are in stock. They can also update their inventory report every time a new order is made.



Consider a business that's trying to keep track of all its **clients**, **projects**, and **employees**. Creating relationships between their data lets them see which **projects** they have with each **client**, and which **employees** are associated with each **project**.





Relating Data in Zoho Creator

Relationships in Zoho Creator

In Zoho Creator applications, you relate data by using **lookup fields**. They're called lookups because they let you look up information from a **report** while you're filling out a **form**.





With **lookups**, you can also connect records from one report to records in another report.





In the Creator application below, the **lookup** field in the **Projects form** (left) looks up information from the **Workers Report** (right).

When the manager assigns people to projects, they will always have an updated list of workers to pull from. This is similar to the spreadsheet example we saw on page 9.

Projects

Project Name: Renovation

Price: 2500000 \$

Start Date: 30-Sep-2018

Workers: Benjamin Gutierrez, Jack Washington

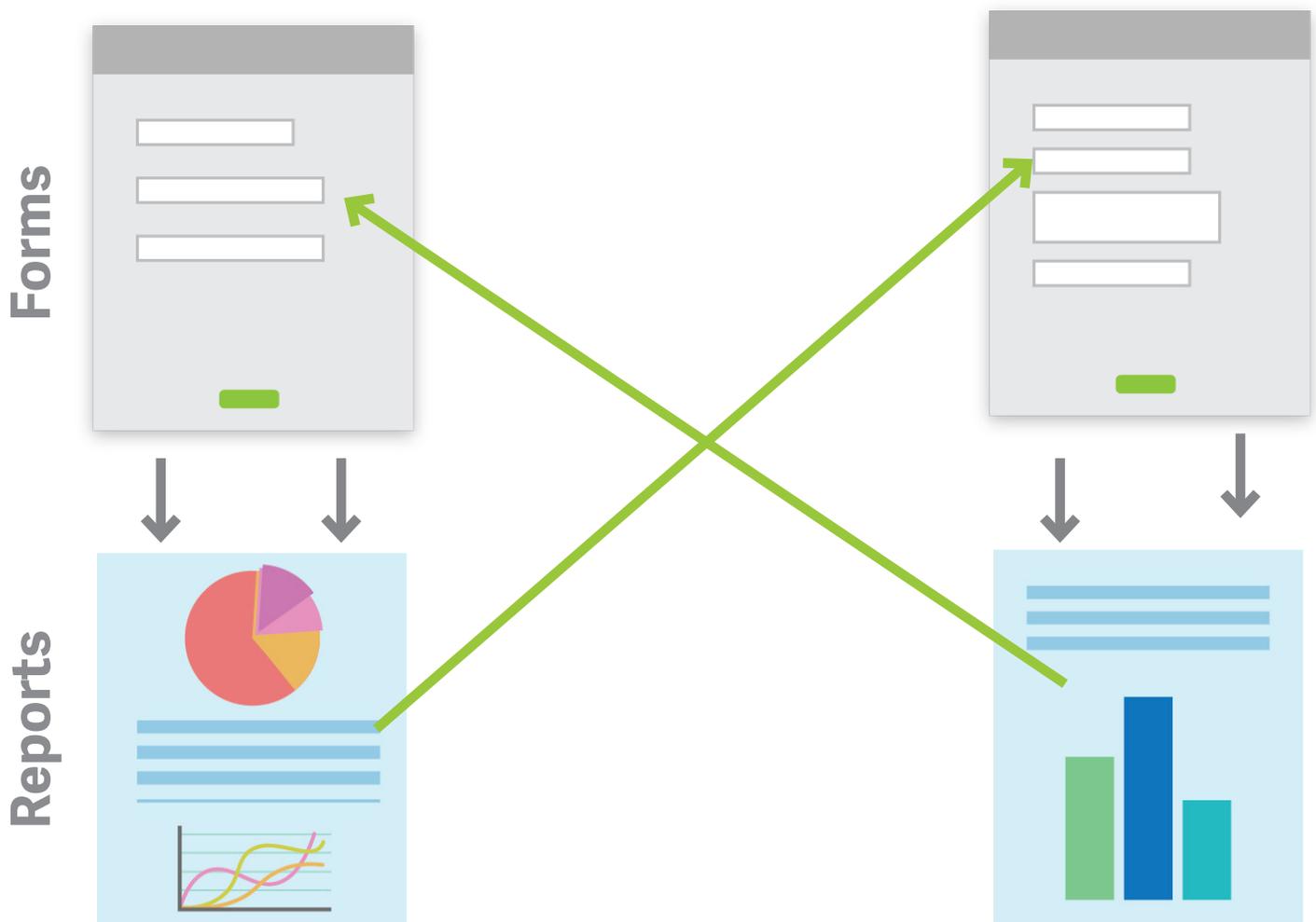
Workers Report

<input type="checkbox"/>	Name	Email
<input type="checkbox"/>	Ernest Howell	ehowell9@si.edu
<input type="checkbox"/>	Benjamin Gutierrez	bgutierrez8@earthlink.net
<input type="checkbox"/>	Anna Payne	apayne7@adobe.com
<input type="checkbox"/>	Randy Sims	rsims6@state.gov
<input type="checkbox"/>	Jack Washington	jwashington5@unesco.org
<input type="checkbox"/>	Joe Henderson	jhenderson4@privacy.gov.au
<input type="checkbox"/>	Kimberly Day	kday3@meetup.com
<input type="checkbox"/>	Donald Willis	dwillis2@google.com.br
<input type="checkbox"/>	Bobby Henderson	bhenderson1@wordpress.org
<input type="checkbox"/>	Charles Fields	cfields0@wikia.com

Two-way Relationships

A regular lookup field lets you pull records from one form to another. As we saw on the previous page, you can relate workers to projects. But what happens if you want to relate projects to workers?

Bi-directional lookup fields let you create two-way relationships within your application.





In the Creator application below, a bi-directional lookup allows a manager to relate workers and projects from either the *Projects* form or the *Workers* form.

When entering data, a manager can assign an employee to a project or a project to an employee.

Projects

Project Name	<input type="text" value="Renovation"/>
Price	<input type="text" value="2500000"/> \$
Start Date	<input type="text" value="30-Sep-2018"/>
Workers	<div><div><input type="checkbox"/> Benjamin Gutierrez</div><div><input type="checkbox"/> Jack Washington</div></div> <div><div>Anna Payne</div><div>Bobby Henderson</div><div>Charles Fields</div><div>Donald Willis</div><div>Ernest Howell</div><div>Joe Henderson</div><div>Kimberly Day</div></div>

Workers

Name	<input type="text" value="Anna Payne"/>
Email	<input type="text" value="apayne7@adobe.com"/>
Phone Number	<input type="text" value="(361)591-5184"/>
Job	<input type="text" value="Carpenter"/>
Project Name	<input type="text" value="Major Renovation"/>

Major Renovation

Floor Addition

Storefront Demo

Renovation

New Apartment

Why Use Lookups?



Make connections

Lookups let you see how one piece of information is connected to others, allowing you to organize complicated data sets.



Get more consistent data

When you let users type anything in your forms, they'll be prone to mistakes like misspelled words. Lookups constrain what users can enter and ensure that your data is consistent.



Do less work

Since lookup fields relate data, changes made in one place are reflected in associated forms. If you edit a record in one report, your changes will be reflected in every form that refers to it.

Example 1

Lisa & IT Management



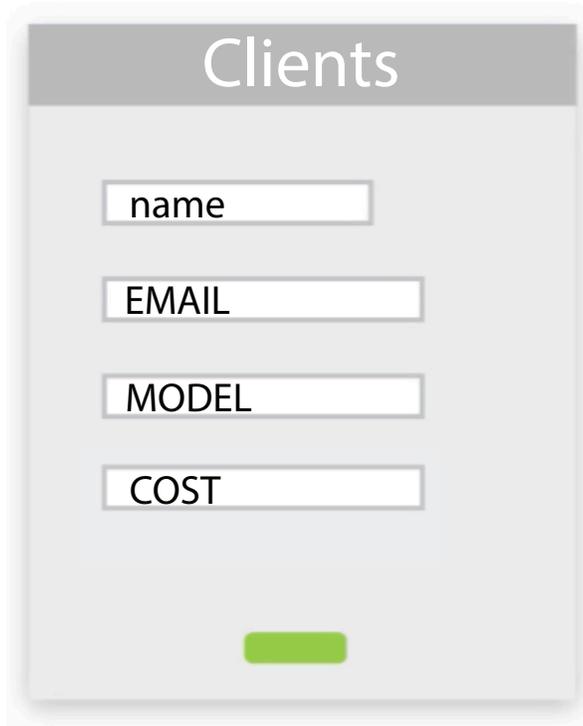
Lisa is an IT manager who has been asked to build an application to keep track of her company's computers.

She starts by **listing all the information** she needs:

- the model of each computer
- the names and contact information of all employees
- the list of who has been given which computer

Now that Lisa has the list of information she needs, she can begin building her forms.

First, she tries collecting all employee and computer information in **one form**:



The image shows a mobile application interface for a form titled "Clients". The form has a grey header with the title "Clients" in white. Below the header, there are four white input fields with black text labels: "name", "EMAIL", "MODEL", and "COST". At the bottom of the form, there is a green rectangular button.

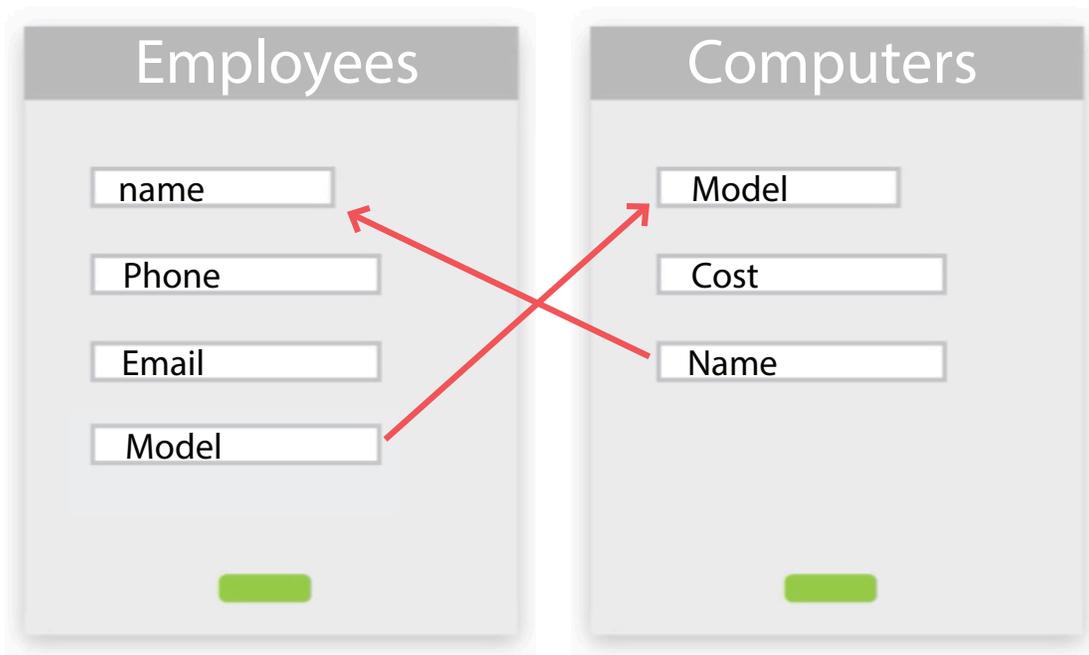
Initially, Lisa likes using just one form. She can track which employee has which computer, but there are some **drawbacks**:

- Every time someone gets a new computer, Lisa has to reenter their contact information.
- Every time a computer gets reassigned, Lisa has to reenter information about the computer like its model and cost.

Lisa tries to organize her data a different way.



Next, Lisa tries separating her application into **two forms**. She uses a **bi-directional lookup** to link them together.



Splitting her data into two forms has some advantages. Lisa doesn't waste time reentering information. After she enters an employee once, she can assign them to any computer. Since her lookup is bi-directional it appears on both her forms, which means she can also enter computers only once and then assign them to any employee. But this structure isn't perfect either.

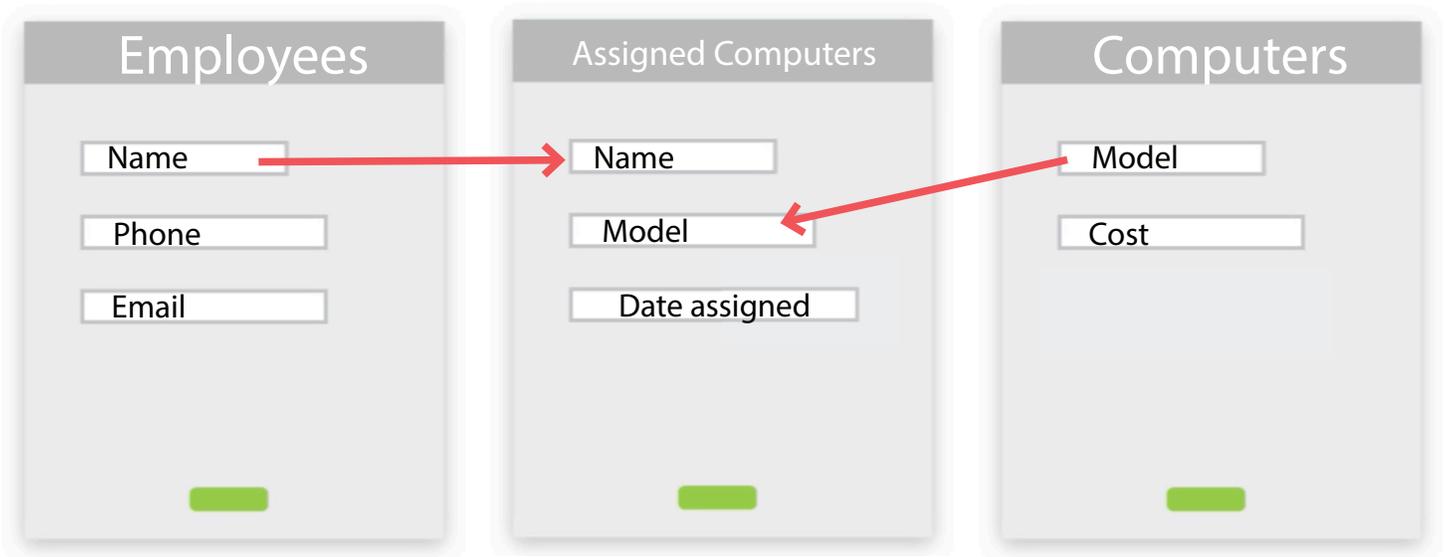
- There's no history. Each time Lisa updates an employee or computer record, her application won't remember where the computer was before the update.



Still unsatisfied, Lisa tries a third way.

Lisa tries splitting her application into **three forms**. Her new *Assigned Computers* form looks up names from the *Employees* form and model numbers from the *Computers* form.

She does this by using **two lookup fields**.



Lisa's new application structure works well because:

- **It's low maintenance.** Each employee and computer only needs to be entered once, so there's no more repetitive data entry.
- **It tracks history.** When a computer gets reassigned to someone, Lisa creates a new record (instead of updating an old one). By searching in the *Assigned Computers* report, Lisa can see all the computers a particular employee has had and all employees that have been assigned to a particular computer.





How to Create Relationships

Planning Lookups



List your requirements

List all the information you need in your application. If you'll be crunching numbers, think about what will go into any formulas you're using. If you're going to send email or text notifications, remember to collect contact info.

Example: In an application for managing events, you'd want to collect information like event dates, attendee contact data, budgets, etc.



Decide how to structure your application

Once you have your list of requirements, you need to organize that data into forms. Generally, you'll want a different form for each noun you're managing (person, place, or thing). This will help you build relationships and connect your data at later stages.

Example: If you need to keep track of orders, inventory, delivery drivers, and customers, you'd want to create individual forms for each of these.

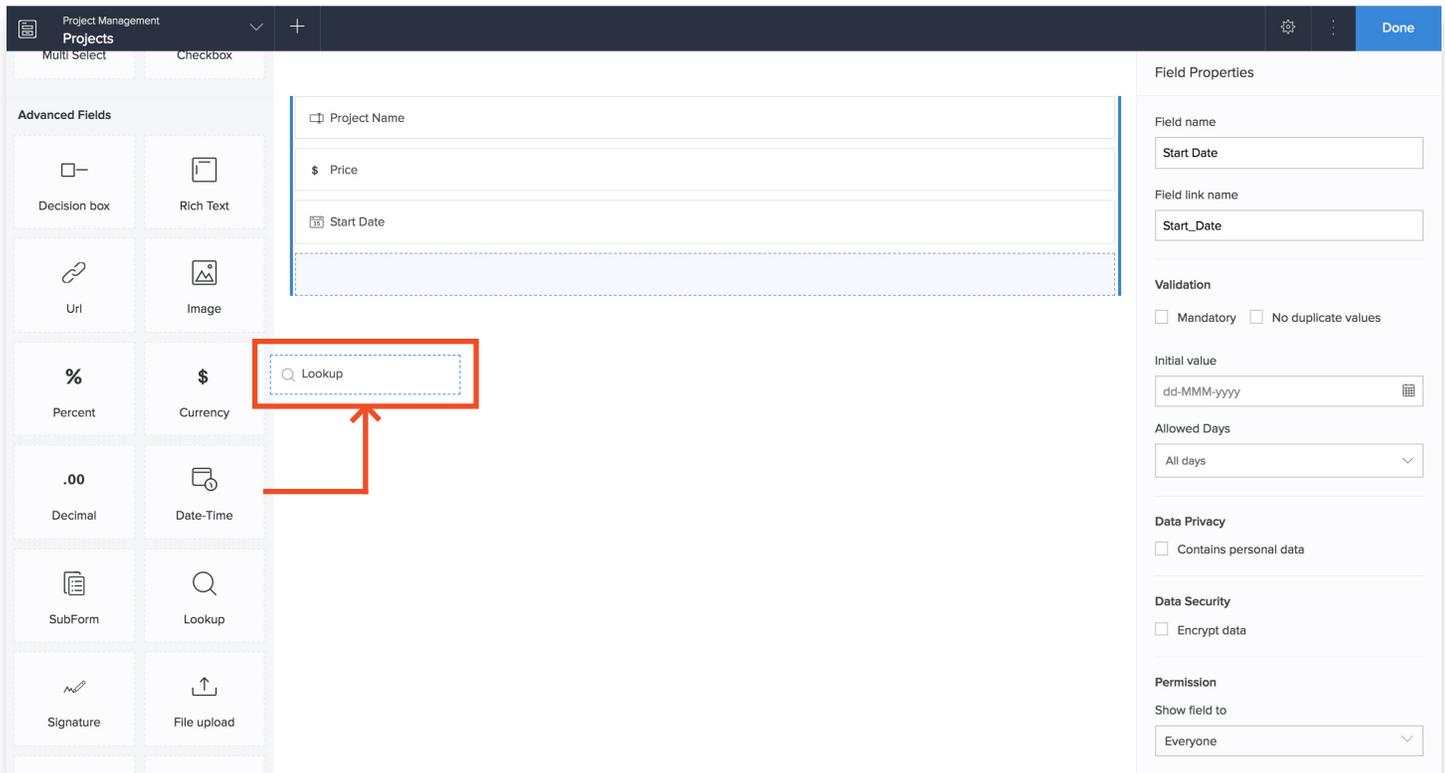


Create relationships

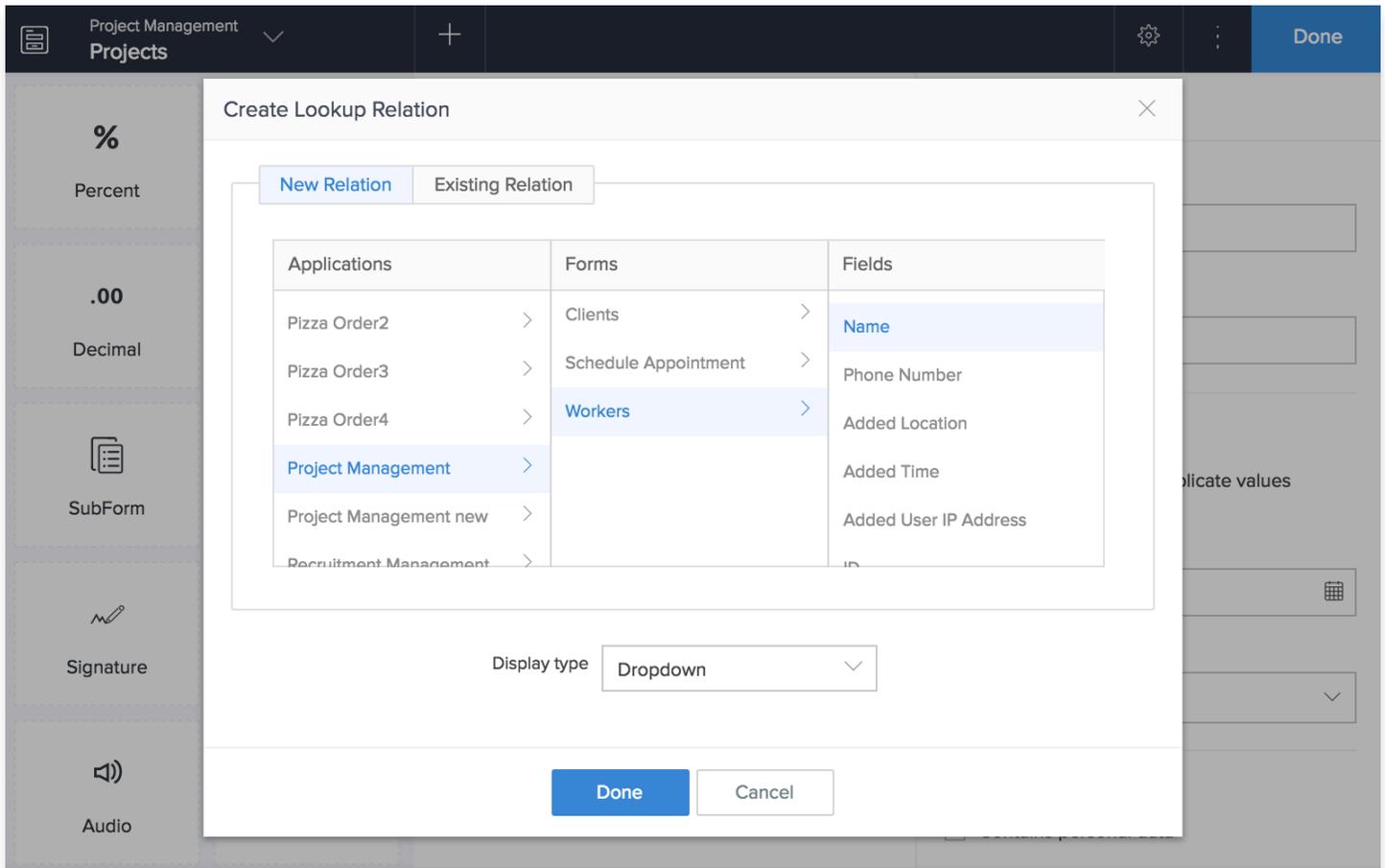
Check to see if you'll be using the same data in two places. You should be able to reference it with a lookup field after it's entered the first time.

Example: In an application to manage projects, one employee may work on multiple projects. Rather than reenter employee names over and over, the application should have a lookup field on its *Projects* form so you can choose employee names that are already in the system.

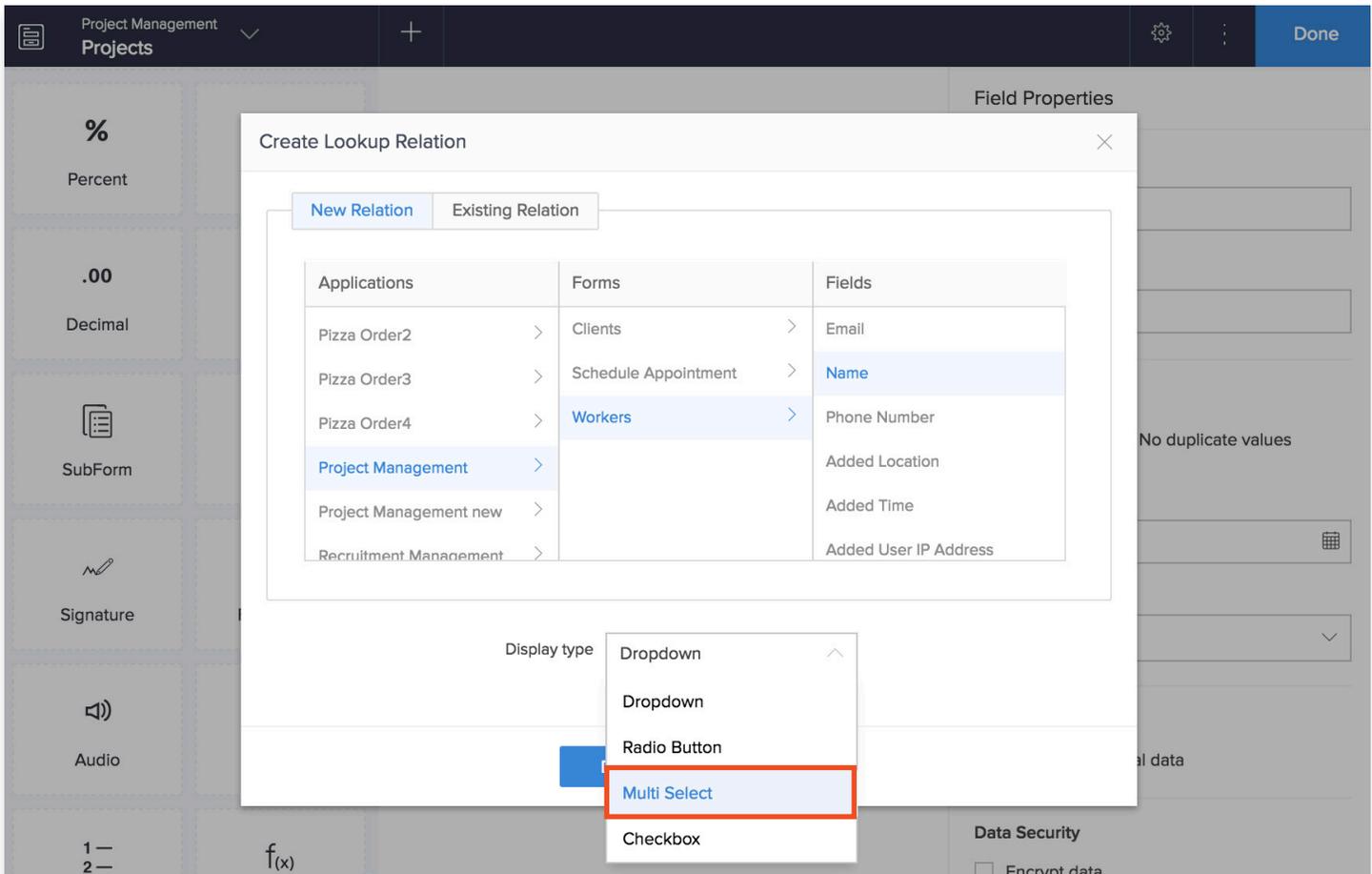
Building



Go to the form builder screen. Drag a **lookup field** onto the builder.



Choose a form from the application you want to create a relationship with. In the **fields column**, choose which information from the other form to display in your lookup.



Choose a display type.

Dropdowns and **Radio Buttons** let you connect each record with only one other record. **Multi Select** and **Checkboxes** let you connect each record with several other records.

Field Properties

Field name
Workers

Field link name
Workers

Display Fields
Project Management → Workers

Name + - ⇄

Validation
 Mandatory

Field Properties

Field link name
Workers

Display Fields
Project Management → Workers

Name + - ⇄

Email + - ⇄

Preview
Name -

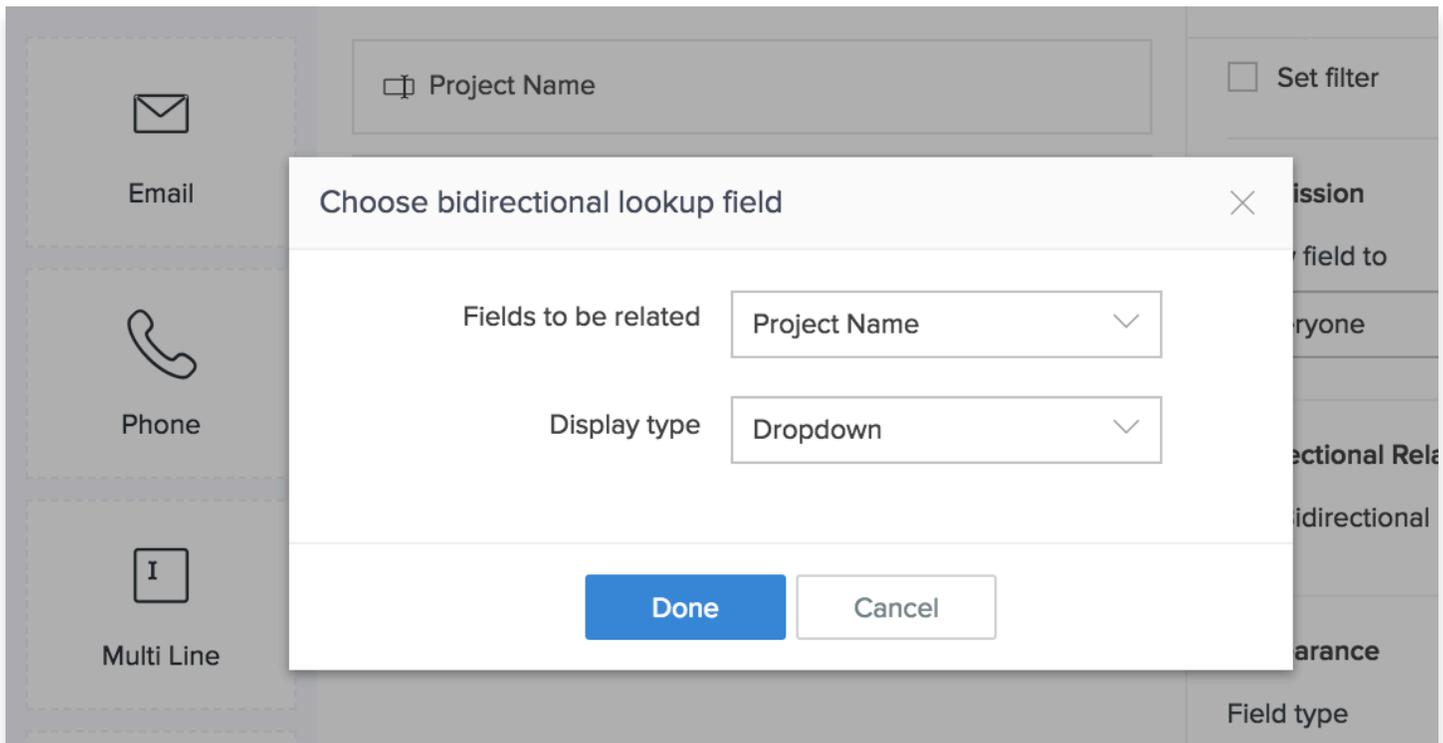
Validation
 Mandatory

Choices
 Allow new entries
 Alphabetical Order

User Defined Fields
Email
Phone Number
Name
System Fields

To change how your lookup field displays choices, go to **Display Fields** under **Field Properties**.

To display multiple fields in your lookup, click the + button.



If you want to make your lookup field show up in both forms that you're relating, scroll down in the **Field Properties** and check **Bidirectional Relation**.

Choose which field from your current form you'll display in the other form and pick a **display type**.



Lookups and Reports

Project Management

Project Report

Project Name	Price	Start Date	Workers
New Apartment	\$ 2,000,000.00	06-Jun-2016	Charles Fields Bobby Henderson
Renovation	\$ 80,000.00	21-Jun-2016	Joe Henderson
Storefront Demo	\$ 11,000.00	06-Jul-2016	Jack Washington Benjamin Gutierrez
Floor Addition	\$ 120,000.00	11-Jul-2016	Kimberly Day Randy Sims
Major Renovation	\$ 140,000.00	25-Jul-2016	Donald Willis Ernest Howell

Showing 5 of 5

After adding a lookup field, you'll be able to reference information from another report.

Project Management

Project Report

Project Name	Price	Start Date	Workers
New Apartment	\$ 2,000,000.00	06-Jun-2016	Charles Fields Bobby Henderson
Renovation	\$ 80,000.00	21-Jun-2016	Joe Henderson
Storefront Demo	\$ 11,000.00	06-Jul-2016	Jack Washington Benjamin Gutierrez
Floor Addition	\$ 120,000.00	11-Jul-2016	Kimberly Day Randy Sims
Major Renovation	\$ 140,000.00	25-Jul-2016	Donald Willis Ernest Howell

Showing 5 of 5

By **configuring fields** for the selected device, you can make each record that you're referencing a clickable link.

Project Management > Edit this application Help

Project Management **Project Report** [Search] [Add] [Menu]

Project	Project Name	Price	Start Date	Workers
Project	New Apartment	\$ 2,000,000.00	06-Jun-2016	Charles Fields Bobby Henderson
Projects	Renovation	\$ 80,000.00	21-Jun-2016	Joe Henderson
Project Report	Storefront Demo	\$ 11,000.00	06-Jul-2016	Jack Washington Benjamin Gutierrez
Workers				

To configure the fields that you want to display on the report, click **Edit this application** link on the header. Mouse over the report and choose **Configure fields for the web**.

Project Management > Edit this application Help

Project Management **Project Report** [Search] [Add] [Menu]

Project	Project Name	Price	Start Date	Workers	Email
Project	New Apartment	\$ 2,000,000.00	06-Jun-2016	Charles Fields Bobby Henderson	cfields0@wikia.com bhenderson1@wordpress.org
Projects	Renovation	\$ 80,000.00	21-Jun-2016	Joe Henderson	jhenderson4@privacy.gov.au
Project Report	Storefront Demo	\$ 11,000.00	06-Jul-2016	Jack Washington Benjamin Gutierrez	jwashington5@unesco.org bgutierrez28@earthlink.net
Workers	Floor Addition	\$ 120,000.00	11-Jul-2016	Kimberly Day Randy Sims	kday3@meetup.com rsims6@state.gov
Clients	Major Renovation	\$ 140,000.00	25-Jul-2016	Donald Willis Ernest Howell	dwillis2@google.com.br ehowell9@si.edu
Schedule Appointments					

Showing 5 of 5

You can reference information other than what's displayed in your lookup. In this example, a lookup for worker names also lets you display worker emails.

Example 2

Jamie's Catering Service



Jamie needs a custom application to streamline her catering business.



She wants to take orders through her website, providing instant quotes to potential customers. She also wants to automate scheduling and notify her employees of upcoming jobs.

Jamie lists all the information she will need for her application:

- employee contact info
- employee availability
- menu items
- price per plate
- event buyer info
- event date and location
- event attendee number

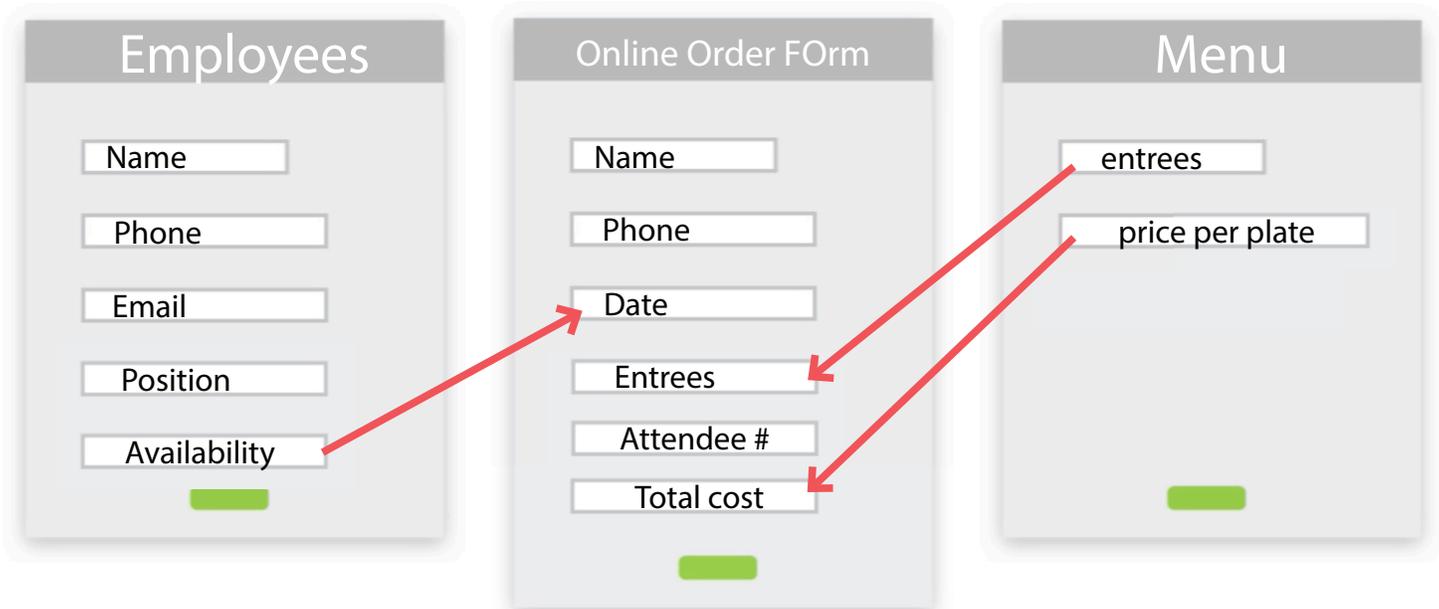
Jamie determines that the best way to do this is to create **three forms**, one of which will be embedded on her website.

1. An **Employee Form**, where Jamie stores info on each of her employees, including contact information and availability.
2. A **Menu Form** that stores menu options and their price per plate. Jamie adds and removes these menu options depending on what her catering company is offering customers.
3. An **order form** on Jamie's website that customers fill out to request catering service on a specific date.



The *Order Form* is placed on the catering website. Rather than constantly updating a static list of menu options on the *Order Form*, Jamie uses a **lookup field** to dynamically pull available entrees from the *Menu* form. Since Jamie's *Menu* form keeps track of each entree and its price per plate, her *Order Form* automatically generates a quoted total based on the entree selected and number of attendees.

After an order has been submitted and an event has been scheduled, Jamie's employees need to be assigned to and notified about the event. Employees are automatically scheduled based on the event date and their availability. Their availability is pulled from the *Employees* form with a lookup field; their phone number is also pulled from the *Employees* form and is used to send a text message with the event details.



This application's structure works well because:

- **It's flexible.** By using lookup fields, Jamie is able to change key details, such as food prices and employee availability, without upsetting essential business processes.
- **It adds value.** Since Jamie has linked her *Menu* form and order form, she's able to instantly provide quotes to potential customers.
- **It streamlines operations.** Since Jamie has linked her employee form and order form, she's able to spend less time scheduling employee's for work, and more time focusing on her passion: cooking.

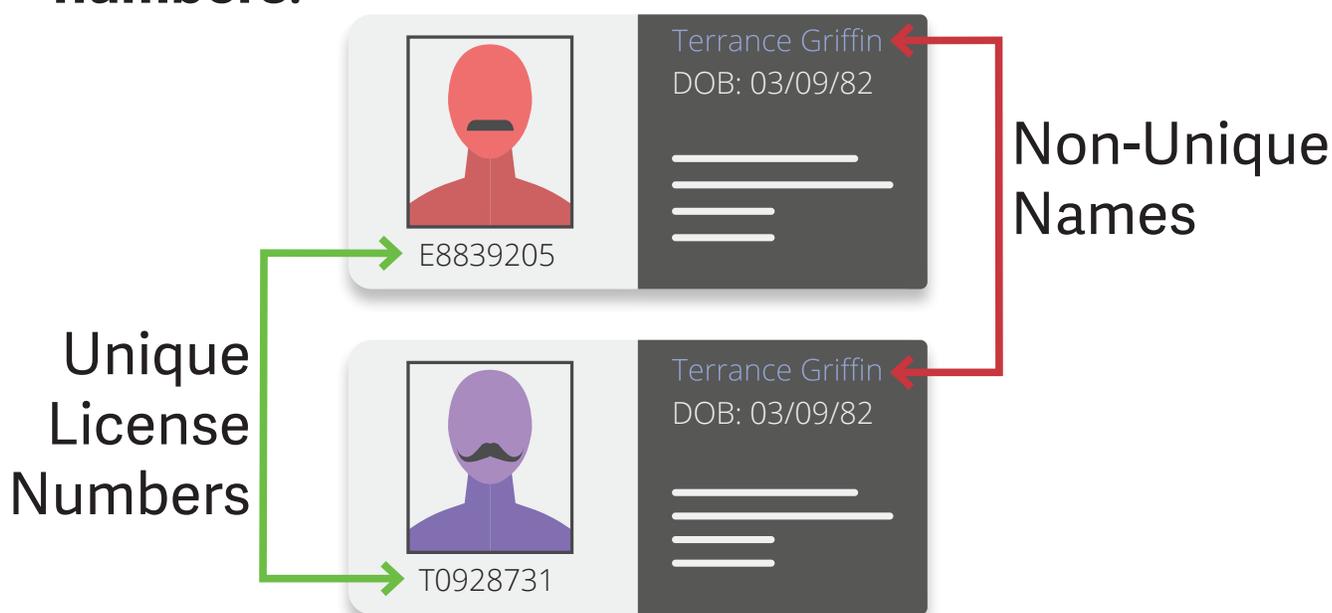




Automating with Lookup Fields

Record IDs

Computers and people identify information in different ways. Look at the two driver's licenses below. A person would instantly recognize that they belong to different people because the **pictures** are different. But computers can't see pictures. For a computer, everything about these two licenses is the same except the **license numbers**.



You could think of each driver's license like a **record** in Creator. But what happens if two records have the same information? How would Creator differentiate between the two records? To avoid confusion in cases like this, Creator assigns a unique **record ID** number, similar to a license number, to every record that gets created.

Lookups and Automation

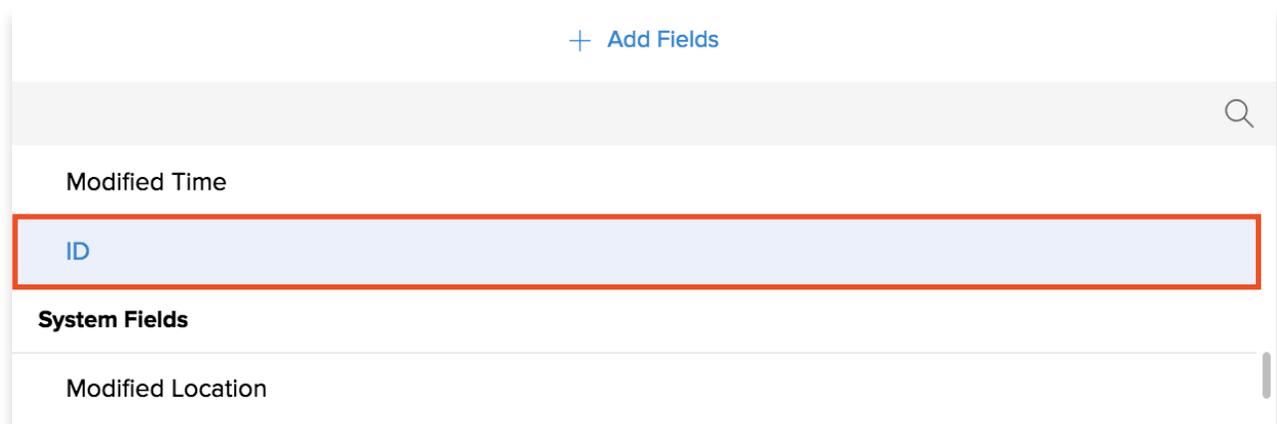
It's important to understand the connection between **lookup fields** and **record IDs** when building **automation** into your application. Lookup fields display information from records that are stored in a different form. When a user selects an option that's displayed in a lookup field, Creator saves a new record. However, this information is not saved the way it was displayed to the user. Instead, Creator saves the **ID of the record** that the user chose in the lookup field.

Think back to the IT management application for assigning computers to employees (p. 18). To assign a computer to Dylan, Lisa fills out the *Assigned Computers* form and picks his name from options displayed in the lookup field. She sees Dylan's name as an option because there's a record with his name saved in the *Employees* form. When Creator saves a record of this assignment, it won't save Dylan's name. It will save the **ID of the record** from the *Employees* form where his name was entered.

When you're scripting, remember that each field has a corresponding data type.

Data types make sure people enter the right information in each field and ensure that your scripts run smoothly. **Dropdown fields** store data as text (string variables in Deluge terms). **Lookup fields** always store number data (bigint or big integer in Deluge terms) because they're always saving numeric record IDs.

ID numbers are invisible because you usually don't need to worry about them. If you want to see them, navigate to the **Design** page and click on any report from the preview area. Mouse over the report and click **Configure fields for the web**. When you click the **Add field** option, it displays the list of fields that Creator collects information about each record behind the scenes.



Concept Check

Let's say you have a form that has a lookup field that pulls names stored in another form. You want to use these names in an automated email that gets sent when the form is submitted. If someone named Steve submitted the form, you'd want your email to begin with:

Inbox

Hey Steve, hope you're doing well.

After testing the email, you see that it reads something like:

Inbox

Hey 9393894958292, hope you're doing well.

Why is this happening? Because the lookup field refers to the record ID where Steve's information is stored. Remember, Creator doesn't see Steve's name as a unique piece of data. After all, there could be many Steves in your application. To use Steve's name in the email, we'll need to write a script that tells Creator to fetch his name, rather than just his ID. We'll show you how to write this script in the next example.



Fetching Data with Lookups



You want to send a confirmation email to each employee that addresses them by name. Since your lookup field saves record IDs, you have to write a script that uses the ID to find the employee's name. To do this, use the **fetch records Deluge task**. The syntax for fetching records with a lookup field looks like this:

```
<variable> = <form to look up> [ ID == input.<lookup field> ].<field to look up>;
```

- *<variable>* could be named anything. This will store the name you're looking up from the Employee form.
- *<form to look up>* is the Deluge name of the form you're looking up info from.
- *[ID == input.<lookup field>]* is telling Creator which records you're looking for. Remember that the lookup field is really referencing an ID number. This tells Creator to find the record whose ID matches what was chosen in the lookup field.
- *<field to look up>* is the Deluge name of the field you're looking up information from.

More Help



To learn more about making Creator applications, [watch our tutorial series](#) that covers the basics of Zoho Creator from adding lookup fields to writing Deluge script.



Contact our **support** at:

USA : +1 (888) 900-9646

UK : +44 (20) 35647890

Australia : +61-2-80662898

India : +91-44-67447000

Email us at **support@zohocreator.com**

Got Feedback?



We'd love to hear it! We want to help you get the most out of Zoho Creator. Let us know what support materials you've liked, and what you'd like more of.



Email us at **feedback@zohocreator.com**

