Salesforce day 1:

Power Up with AppExchange

**Learning Objectives**

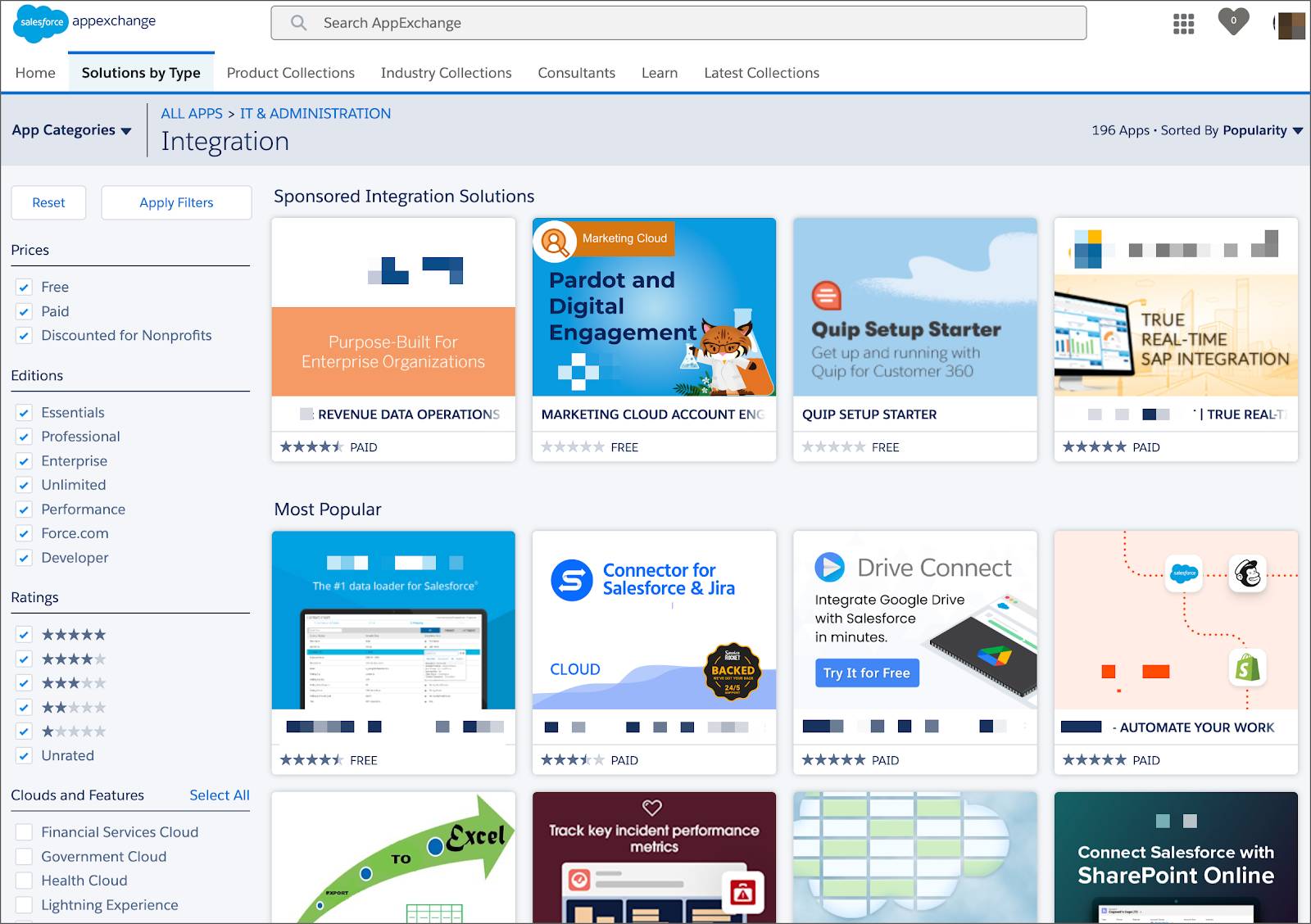
After completing this unit, you’ll be able to:

* Develop your own AppExchange strategy.
* Install an app from AppExchange.

**What Is AppExchange?**

You’re probably comfortable with the idea of app stores. Whether you’re downloading apps on your phone, tablet, computer, or other device, you have to download and install apps to make the most of your technology.

Salesforce is the same way. Earlier, we mentioned the enterprise ecosystem. Salesforce has a community of partners that use the flexibility of the Salesforce platform to build amazing apps and other solutions that anyone can use. These offerings are available (some for free, some at a cost) for installation on AppExchange.



**Strategies for Success**

D’Angelo’s DreamHouse app is a raging success among the company real estate brokers. But if we’re being realistic, D’Angelo is only one guy. There are only so many hours in the day for him to develop new apps for his colleagues.

Luckily, AppExchange is full of apps D’Angelo can download to help DreamHouse manage everything from payroll to travel approval to integrations with other tools like Evernote and MailChimp.

The possibilities AppExchange offers are exciting, but before you start downloading every app in sight you need to develop a strategy. A solid AppExchange strategy helps ensure that you’re getting the highest value apps without duplicating functionality or investing in something that you don’t need.

Follow these steps to develop a good AppExchange strategy.

1. Identify departments that use or plan to use Salesforce. These are your primary stakeholders.
2. Research what’s available on AppExchange that best meets your stakeholder requirements. Discuss business cases with department heads to determine exact needs. Here are some good questions to ask:
   1. What business problem are you trying to solve?
   2. What are your main pain points right now?
   3. How many users need this app?
   4. What’s your budget?
   5. What’s your timeline?

These questions help you identify apps that are the best fit for each department or business case.

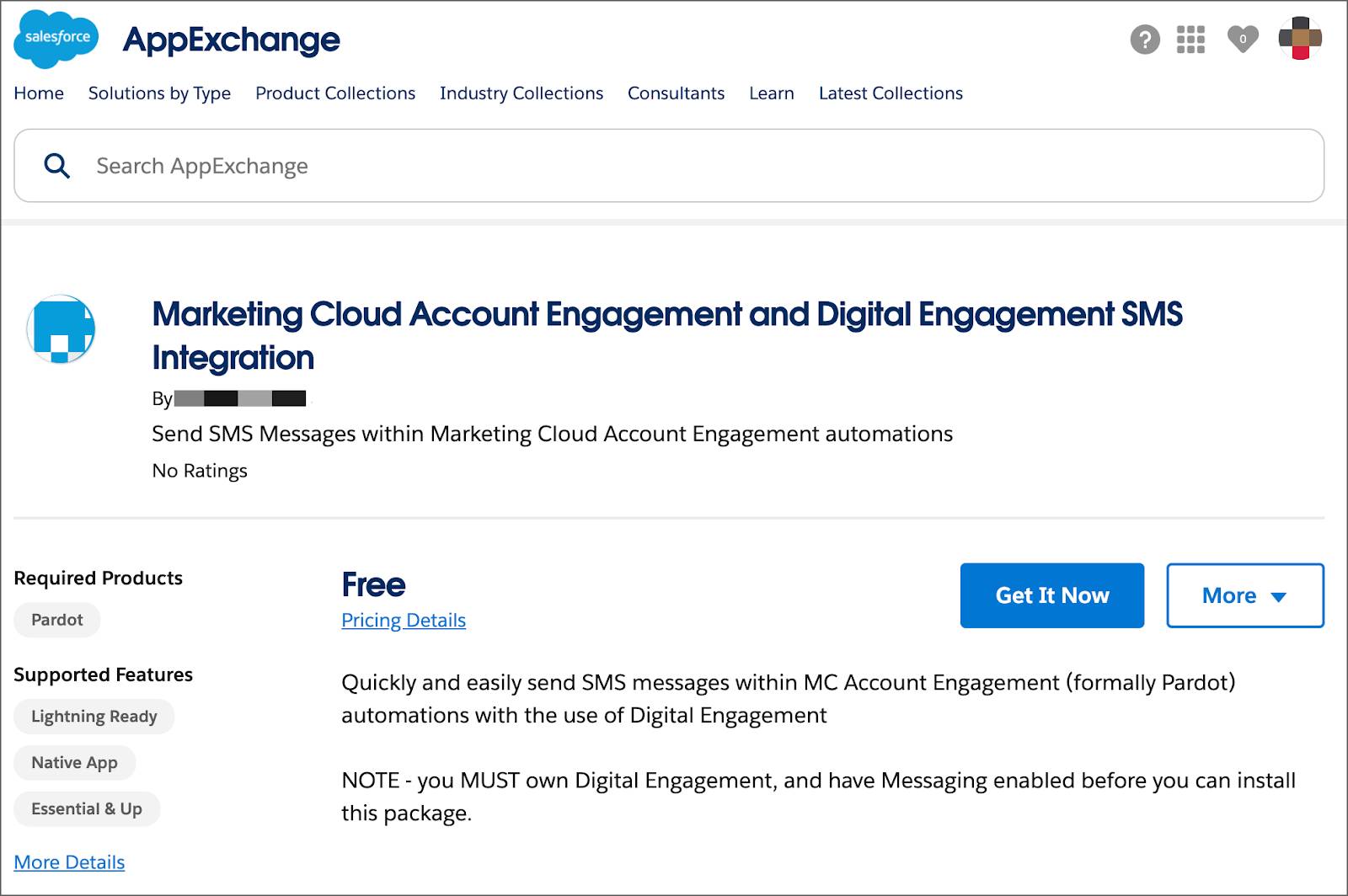
1. When you find an app that you think meets your needs, download the app in a test environment (like a free Developer Edition or sandbox). Ensure that the app you’re installing doesn’t interfere with any other apps you’ve installed or customizations you’ve made. Sandboxes are copies of your organization in a separate environment. They’re used for development and testing. See the [Sandbox Types and Templates](https://help.salesforce.com/articleView?id=create_test_instance.htm&language=en_US) documentation.
2. If you’re choosing between multiple apps, take some time to evaluate what you’ve tested. Determine whether there are feature gaps or unwanted functionality. If necessary, invite your stakeholders to demo the apps and provide feedback.
3. You’re ready to go! You’ll install and deploy your app in your production environment. Make sure you keep your users in the loop about what’s changing, and provide training and documentation as necessary.

**Install Your First App**

While AppExchange resembles a traditional app store like you can find on your phone or tablet, it’s important to remember that your Salesforce org is a complicated environment. You can’t just install an app because it has a cool logo or a convincing catchphrase.

So what’s the right way to install an app? We’ll show you! This is just an example - no need to follow along.

Let’s say you find this great app on [AppExchange](https://appexchange.salesforce.com/) that gives you a fancy set of dashboards for your org.



To install the app, you would click **Get It Now**. This button takes you to the installation wizard that guides you through the steps. Here are two key questions you need to answer during the installation process:

* Where do I install the app, production or sandbox? In general, it’s a best practice to first install apps in a nonproduction environment. Try installing in a sandbox for your production org or in a Developer Edition org. Testing the app first helps you avoid conflicts in production with things like object names.
* Should I give app permissions to admins only, all users, or specific profiles? That depends on who the app is for. If you want to limit access to a particular set of users, plan to modify those user profiles before you install the app.

**Where’d My App Go?**

Awesome! That's how you would install an app. Now, if only you could find it... Here is how you find apps after you've installed them.

Apps are installed using something called a package. To find the package:

1. From Setup, you search and select Installed Packages in the Quick Find box.
2. Click the name of the package you installed. It will be the same name from the AppExchange download page.
3. Click **View Components** to see more information about the package. The Package Details page shows you all the components, including custom fields, custom objects, and Apex classes in the package. This information helps you determine whether you have any conflicts in your own customizations.

Optimize Customer Data with Standard and Custom Objects

**Learning Objectives**

After completing this unit, you’ll be able to:

* Describe the perks of using objects on the Salesforce CRM platform.
* Explain the difference between standard objects and custom objects.
* List the types of custom fields an object can have.

Note

Accessibility

This unit requires some additional instructions for screen reader users. To access a detailed screen reader version of this unit, click this link: [Open Trailhead screen reader instructions](https://developer.salesforce.com/files/accessibility/data_modeling/objects_intro/index.html).

**Overview of Objects**

DreamHouse is a realty company that provides a way for customers to shop for homes and contact real estate agents online. DreamHouse brokers use some of the standard Salesforce CRM functionality, like contacts and leads, to manage customer data of home buyers.

But when it comes to selling houses, there are a lot more things they want to track. For example, Salesforce doesn’t include a standard way to track properties. How is DreamHouse supposed to know which homes they have for sale or how much each home costs?

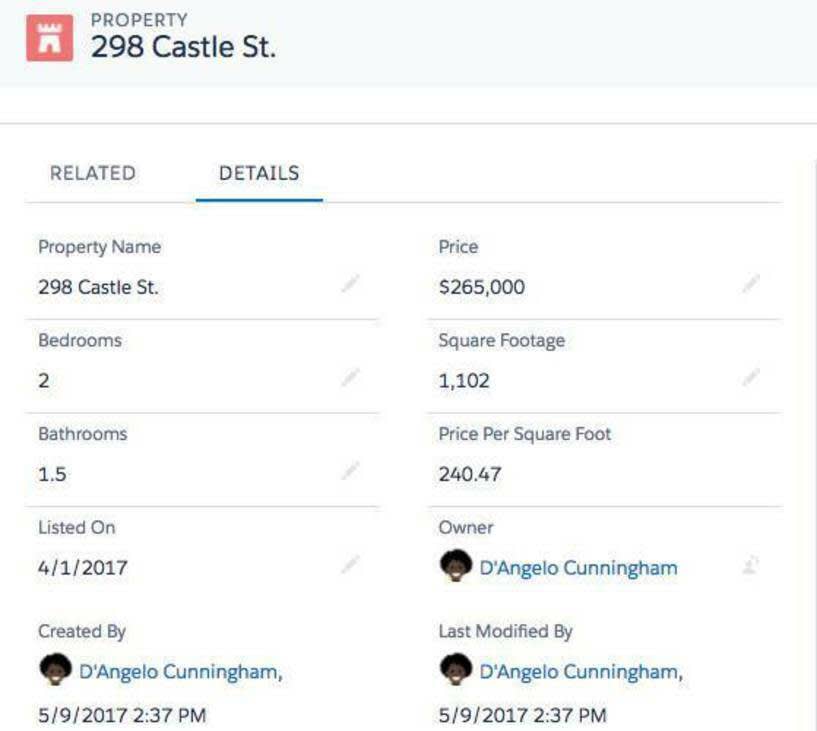
Luckily, the Salesforce admin, D’Angelo, knows that the Salesforce platform offers a solution. And you can work with D’Angelo to see what he’s building.

Start with the **data model**. A data model is more or less what it sounds like. It’s a way to model what database tables look like in a way that makes sense to humans. If you’re not familiar with databases, think about storing data in a spreadsheet. For example, D’Angelo can use a spreadsheet to track all DreamHouse’s properties. Columns can store the address, cost, and other important attributes. Rows can store this information for each property that DreamHouse is selling. Database tables are set up in a similar way.



But looking at data in tables isn’t ideal for humans. That’s where the data model comes in.

In Salesforce CRM, we think about database tables as **objects**, we think about columns as **fields**, and rows as **records**. So instead of an account spreadsheet or table, we have an Account object with fields and a bunch of identically structured records.



When we talk about the data model, we’re talking about the collection of objects and fields in an app. Learn more about objects and fields so you can start building your own data model.

**Get to Know Objects**

Salesforce supports several different types of objects. There are standard objects, custom objects, external objects, platform events, and BigObjects. In this module, we focus on the two most common types of objects: standard and custom.

**Standard objects** are objects that are included with Salesforce. Common business objects like Account, Contact, Lead, and Opportunity are all standard objects.

**Custom objects** are objects that you create to store information that’s specific to your company or industry. For DreamHouse, D’Angelo wants to build a custom Property object that stores information about the homes his company is selling.

Objects are containers for your information, but they also give you special functionality. For example, when you create a custom object, the platform automatically builds things like the page layout for the user interface.

**Create a Custom Object**

Follow along as D’Angelo to see how he builds the Property object. You need this object later, so don’t skip these steps!

1. Scroll to the bottom of this page and create a trailhead playground. Don’t skip this step! You need to use a fresh and clean Trailhead Playground for this module.  
   **Note:** Even if you're completing this module as part of the Admin Beginner trail, be sure and create a new Trailhead Playground to complete these steps. You don't need to reinstall the Dreamhouse app in the new playground org.
2. Once your playground is created (it takes a minute!), press **Launch**.
3. Click the gear icon The setup gear. at the top of the page and launch setup.
4. Click the **Object Manager** tab.
5. Click **Create** | **Custom Object** in the top-right corner.
6. For Label, enter Property. Notice that the Object Name and Record Name fields auto-fill.
7. For Plural Label, enter Properties.
8. Prior to saving the custom object, scroll to the bottom of the page and select the checkbox **Launch New Custom Tab Wizard after saving this custom object**.
9. Leave the rest of the values as default and click **Save**.
10. On the New Custom Object Tab page, click the Tab Style field and select a style you like. The style sets the icon to display in the UI for the object.
11. Click **Next**, **Next**, and **Save**.

Great job! You just created your first custom object. Now, learn about adding fields to this object.

**Get to Know Fields**

Every standard and custom object has fields attached to it. Get familiar with the different types of fields.

| **Field Type** | **What is it?** | **Can I get an example?** |
| --- | --- | --- |
| Identity | A 15-character, case-sensitive field that’s automatically generated for every record. You can find a record’s ID in its URL. | An account ID looks like 0015000000Gv7qJ. |
| System | Read-only fields that provide information about a record from the system, like when the record was created or when it was last changed. | CreatedDate, LastModifiedById, and LastModifiedDate. |
| Name | All records need names so you can distinguish between them. You can use text names or auto-numbered names that automatically increment every time you create a record. | A contact’s name can be Julie Bean. A support case’s name can be CA-1024. |
| Custom | Fields you create on standard or custom objects are called custom fields. | You can create a custom field on the Contact object to store your contacts’ birthdays. |

Identity, system, and name fields are standard on every object in Salesforce. Each standard object also comes with a set of prebuilt, standard fields. You can customize standard objects by adding custom fields, and you can add custom fields to your custom objects.

Every field has a data type. A data type indicates what kind of information the field stores.

Salesforce supports a bunch of different data types, but here are a few you’ll run into.

* **Checkbox**—for fields that are a simple “yes” or “no,” a checkbox field is what you want.
* **Date or DateTime**—these field types represent dates or date/time combinations, like birthdays or sales milestones.
* **Formula**—this special field type holds a value that’s automatically calculated based on a formula that you write. For example, D’Angelo can write a formula field that automatically calculates a real estate agent’s commission on a home sale.

Again, there are quite a few field types, but most of them are fairly self-explanatory. The important takeaway here is that you want to think about what kind of data you’re trying to store when you create a custom field.

**Create a Custom Field**

The Property object you just created is pretty bare-bones. Add some custom fields to it. Head back to your Trailhead Playground.

1. From Setup, go to **Object Manager** | **Property**.
2. In the sidebar, click **Fields & Relationships**. Notice that there are already some fields there. There’s a name field and some of the system fields you learned about earlier.
3. Click **New** in the top right.
4. For data type, select **Currency**.
5. Click **Next**.
6. Fill out the following:  
   * Field Label: Price
   * Description: The listed sale price of the home.
7. Check the **Required** box.
8. Click **Next**, **Next** again, and then **Save**.

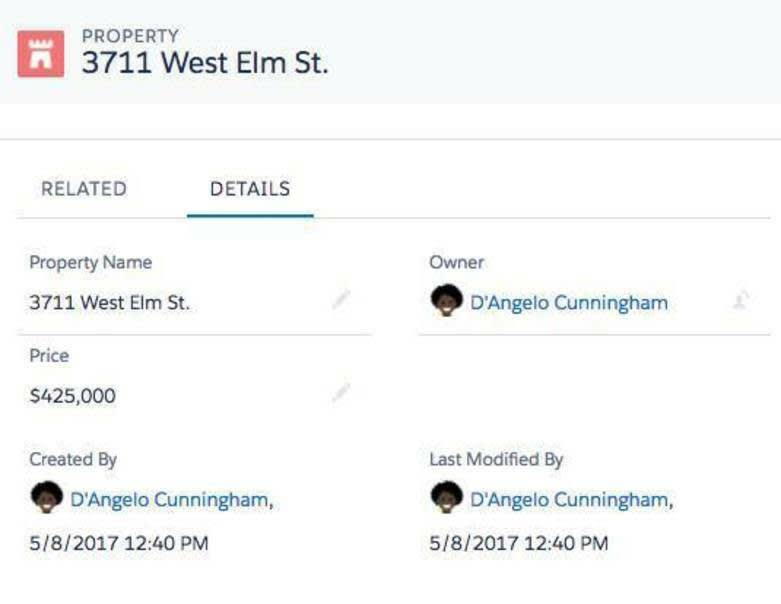
You’ll see your new Price field in the list of Property fields. In the Field Name column, notice that it says Price\_\_c. The “\_\_c” part is an easy way to tell that a particular field is a custom field.

Create a Record

Now, create a property record to see what you did.

1. From the App Launcher (The App Launcher icon. in the navigation bar), find and select **Sales**.
2. Click the **Properties** tab in the navigation bar. If you don’t see it, look under the **More** dropdown.
3. Click **New** in the top corner.
4. Enter a name and price for the property and click **Save**.

Awesome! You’ll see something like the following.



**Customize Responsibly.** While it can seem easy to add and customize objects, remember that what’s going on under the hood is technically complicated. Here are some best practices to keep in mind as you start customizing your own org.

**Be thoughtful about names.** Once you start creating a bunch of objects, it can be tempting to give them “lazy” names. For example, if D’Angelo created another custom object to track condominiums, he might be tempted to name it Property2 instead of Condominium. That’s a recipe for confusion in your org. Give your objects and fields descriptive, unique names to improve clarity.

**Help out your users.** Even with careful naming, your users might not always be clear about the purpose of a particular object or field. Include descriptions for your custom objects and fields. For specialized or complicated customizations, use help text to give more details.

**Require fields when necessary.** Sometimes, you’ll want to force your users to fill out a field when they’re creating a record on a certain object. Every property needs a price, right? Make important fields required to avoid incomplete data.

Resources

* [*Salesforce Help: Customize Your Salesforce Org*](https://help.salesforce.com/articleView?id=customize_overview.htm&language=en_US)
* [*Salesforce Help: Store Information That’s Unique to Your Organization*](https://help.salesforce.com/articleView?id=dev_object_def.htm&language=en_US)
* [*Trailblazer Community: Customer Success Community*](https://trailblazers.salesforce.com/_ui/core/chatter/groups/GroupProfilePage?g=0F9300000001p8wCAA)
* [*Knowledge Article: Custom Objects, Tabs, and Fields Best Practice Guide*](https://help.salesforce.com/s/articleView?id=000355147&type=1)

Create Object Relationships

**Learning Objectives**

After completing this unit, you’ll be able to:

* Define the different types of object relationships and their typical use cases.
* Create or modify a lookup relationship.
* Create or modify a master-detail relationship.

Note

Accessibility

This unit requires some additional instructions for screen reader users. To access a detailed screen reader version of this unit, click this link: [Open Trailhead screen reader instructions](https://developer.salesforce.com/files/accessibility/data_modeling/object_relationships/index.html).

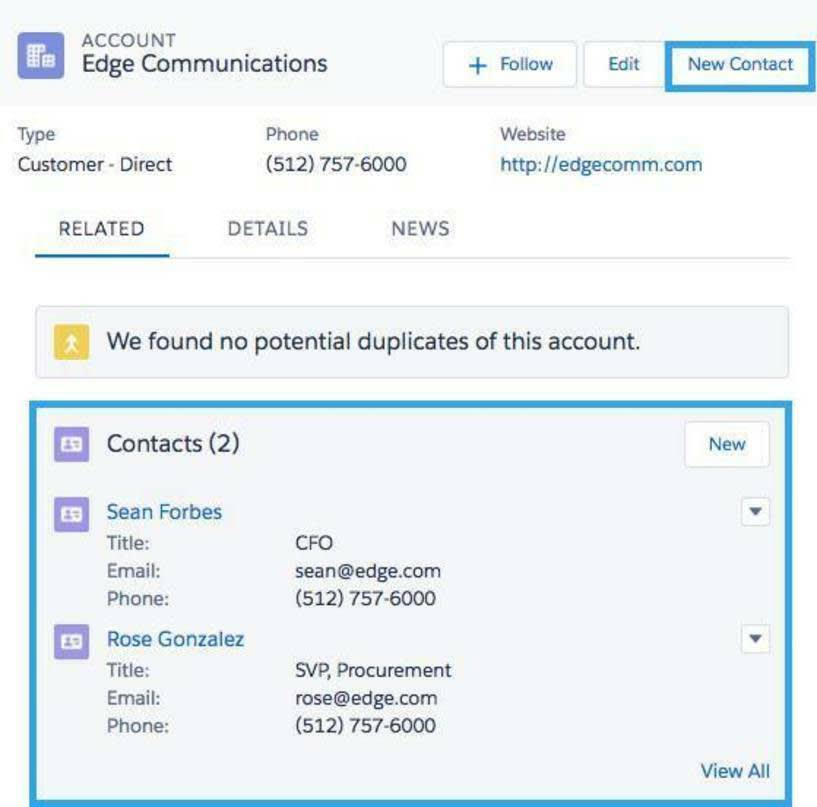
**What Are Object Relationships?**

Now that you’re comfortable with objects and fields, it’s time to take things to the next level with object relationships. Object relationships are a special field type that connects two objects together.

Think about a standard object like Account. If a sales rep opens an account, they’ve probably been talking to a few people at that account’s company. They’ve probably made contacts like executives or IT managers and stored those contacts’ information in Salesforce.

It makes sense, then, that there should be a relationship between the Account object and the Contact object. And there is!

When you look at an account record in Salesforce, you can see that there’s a section for contacts on the Related tab. You can also see that there’s a button that lets you quickly add a contact to an account.



The Account to Contact relationship is an example of a standard relationship in Salesforce. But just like objects and fields, you can build custom relationships as well. In the last unit, you created two objects: Property and Offer. Wouldn’t it be great if all the offers made on a home showed up on its record in Salesforce?

Before you do that, you should learn about the different kinds of relationships you can create in Salesforce.

The Wide World of Object Relationships

Note

Where possible, we changed noninclusive terms to align with our company value of Equality. We maintained certain terms to avoid any effect on customer implementations.

There are two main types of object relationships: lookup and master-detail.

**Lookup Relationships**

In our Account to Contact example above, the relationship between the two objects is a **lookup relationship**. A lookup relationship essentially links two objects together so that you can “look up” one object from the related items on another object.

Lookup relationships can be one-to-one or one-to-many. The Account to Contact relationship is one-to-many because a single account can have many related contacts. For our DreamHouse scenario, you could create a one-to-one relationship between the Property object and a Home Seller object.

**Master-Detail Relationships**

While lookup relationships are fairly casual, **master-detail relationships** are a bit tighter. In this type of relationship, one object is the master and another is the detail. The master object controls certain behaviors of the detail object, like who can view the detail’s data.

For example, say the owner of a property wanted to take their home off the market. DreamHouse wouldn’t want to keep any offers made on that property. With a master-detail relationship between Property and Offer, you can delete the property and all its associated offers from your system.



More on Relationships

Just like in real life, relationships are complicated. Here’s a bit more information to help you differentiate between lookup and master-detail relationships.

Typically, you use lookup relationships when objects are only related in some cases. Sometimes a contact is associated with a specific account, but sometimes it’s just a contact. Objects in lookup relationships usually work as stand-alone objects and have their own tabs in the user interface.

In a master-detail relationship, the detail object doesn’t work as a stand-alone. It’s highly dependent on the master. In fact, if a record on the master object is deleted, all its related detail records are deleted as well. When you’re creating master-detail relationships, you always create the relationship field on the detail object.

Finally, you could run into a third relationship type called a hierarchical relationship. Hierarchical relationships are a special type of lookup relationship. The main difference between the two is that hierarchical relationships are only available on the User object. You can use them for things like creating management chains between users.

When you start adding relationships between objects, remember that you’re increasing the complexity of your data model. That’s not a bad thing, but be extra cautious when you do things like change and delete objects, records, or fields. Check out the resources section for more information on relationship behaviors.

**Create a Custom Object**

You’re ready to jump back in with D’Angelo to build some relationships for the DreamHouse app. Say DreamHouse wanted a way to track users who mark particular properties as favorites on their website. This feature can help DreamHouse’s real estate brokers reach out to potential home buyers.

Note

Even if you're completing this module as part of the Admin Beginner trail, be sure you use the new Trailhead Playground you created in the previous unit.

To start, create a custom object called Favorite and add a field to the object.

1. Click the **Object Manager** tab.
2. Click **Create** | **Custom Object** in the top-right corner.
3. For Label, enter Favorite.
4. For Plural Label, enter Favorites.
5. Check the box for **Launch New Custom Tab Wizard after saving this custom object**.
6. Leave the rest of the values as default and click **Save**.
7. On the New Custom Object Tab page, click the Tab Style field and select a style you like.
8. Click **Next**, **Next**, and **Save**.

Create a Lookup Relationship

Next, create two custom relationship fields on the Favorite object. First, create a lookup relationship that lists the users who select **Favorite** for a property.

1. From Setup, go to **Object Manager** | **Favorite**.
2. On the sidebar, click **Fields & Relationships**.
3. Click **New**.
4. Choose **Lookup Relationship** and click **Next**.
5. For Related To, choose **Contact**. For the purposes of DreamHouse, contacts represent potential home buyers.
6. Click **Next**.
7. For Field Name, enter Contact, then click **Next**.
8. Click **Next**, **Next**, **Next**, and **Save**.

Create a Master-Detail Relationship

Now, create a second relationship field. You want a master-detail relationship where Property is the master and Favorite is the detail.

1. On the Object Manager page for the custom object, click **Fields & Relationships**.
2. Click **New**.
3. Select **Master-Detail Relationship** and click **Next**.
4. For Related To, choose **Property**.
5. Click **Next**.
6. For Field Name, enter Property and click **Next**.
7. Click **Next**, **Next**, and **Save**.

Now, if you look at a Property record, you’ll see Favorites listed in the Related tab.

**Add a Favorite Property**

Next, take a look at how to view favorite properties.

1. From the App Launcher The App Launcher icon. find and select **Sales**.
2. Click the **Properties** tab in the navigation bar. If you don’t see it, look under the **More** dropdown.
3. Click the name of a Property record.
4. Click **Related**. You’ll see Favorites (0) in the Related tab.
5. Click **New**.
6. Enter a name for Favorite Name, then click **Save**.

Great job! Our Favorite object is all set up.

Resources

* [*Salesforce Help: Object Relationships Overview*](https://help.salesforce.com/articleView?id=overview_of_custom_object_relationships.htm&language=en_US)
* [*Salesforce Help: Considerations for Relationship*](https://help.salesforce.com/articleView?id=relationships_considerations.htm&language=en_US)

Where possible, we changed noninclusive terms to align with our company value of [Equality](https://help.salesforce.com/s/articleView?id=release-notes.rn_general_terms_replacement.htm&type=5&release=230). This is a work in progress, so if you find a term to evaluate for inclusive language, click **Provide feedback for this badge** in the right sidebar to submit it.

**ntroduction**

Your company, AW Computing, is adopting Salesforce. You’re the administrator of this org. An org is an identifier that represents a customer’s version of Salesforce and its data within an instance. However, you know the org requires some preparation before users start to log in.

In this project, you get hands-on practice preparing your org for end users—from setting up the exchange rate and customizing your users’ home page to creating unique list views and creating public/private Chatter groups.

By completing these steps, you learn various ways to customize your org before your end users start using Salesforce. Anticipating users’ needs before they even log in makes adoption of Salesforce even smoother. Happy end users = happy administrator!

Since AW Computing is an international company, let’s start by turning on the multiple currencies option. Even if your company isn’t international, it is good to know how to work with multiple currencies within Salesforce. You never know where Salesforce can help you take your company in the future!

**Enable Multiple Currencies**

1. Click the **gear icon** Setup icon and select **Setup**.
2. Enter Company Information in the Quick Find box and select **Company Information**.
3. Click **Edit**.
4. Ensure Locale is set to **English (United States)**and Currency Locale is set to **English (United States) - USD**.  
   Note: Don't worry, you can change the currency back to your default currency after completing this project.
5. Within the Currency Settings section, select the **Activate Multiple Currencies** box.
6. Click **Save**.
7. From the Company Information page, click the **Currency Setup** button.
8. Click **New** in the Active Currencies section.
9. Set up the Euro with the following information.

| **Field** | **Value** |
| --- | --- |
| Currency Type | **EUR – Euro** |
| Conversion Rate | 1.5 |
| Decimal Places | 2 |

1. Click **Save**.

Test the exchange rate on a new opportunity.

1. Click the **App Launcher** App Launcher iconand select **Sales**.
2. Select the **Opportunities** tab, then **New**. Fill in the Opportunity Information.

| **Field** | **Value** |
| --- | --- |
| Opportunity Name | Euro Currency Test |
| Account Name | United Oil & Gas, UK |
| Opportunity Currency | Euro |
| Close Date | (end of current month) |
| Stage | **Closed Won** |
| Amount | 10000(Note: that's four zeros) |

1. Click **Save**.  
   Note: Refresh the opportunity to ensure the changes are reflected. Make a note of the converted amount in the Stage History (shown in parentheses in the Amount field).

**Update the Exchange Rate**

With multiple currencies turned on, let’s update and test the exchange rate.

1. Click the **gear icon** Setup icon and select **Setup**.
2. Enter Manage Currencies in the Quick Find box and select **Manage Currencies**.  
   Note: If Manage Currencies does not appear in your Quick Find search, enter Company Information, and click **Currency Setup**.
3. Click **Edit Rates**, and then edit the Euro to 1.00.
4. Click **Save**.

View the opportunity from the earlier exercise and compare the converted amount to the one you noted.

1. Click the **App Launcher** App Launcher icon and select **Sales**.
2. Click the **Opportunities** tab.
3. From the Recent Opportunities list, click **Euro Currency Test** and compare the converted amount to the one you noted earlier.

Update the Exchange Rate with ACM

Another way to work with the exchange rate is through the advanced currency management tool. You can manage rates with specific dates.

**Enable Advanced Currency Management (ACM)**

1. Click the **gear icon** Setup icon and select **Setup**.
2. Enter Manage Currencies in the Quick Find box and select **Manage Currencies**. Note: If Manage Currencies does not appear in your Quick Find search, enter Company Information, and click **Currency Setup**.
3. Click **Enable** under Advanced Currency Management if not enabled.
4. On the confirmation that appears, select the **Yes, I want to enable Advanced Currency Management** box, and then click **Enable**. Note: Dependent on your browser, you may need to click **Open** and a new tab will open in Classic.

Update the Euro exchange rate with a start date.

1. Click **Manage Dated Exchange Rates**, and click **Continue** on the splash screen.  
   Note: If the Manage Dated Exchange Rates interface is not appearing, try disabling ACM and re-enabling it.
2. Click **New** **Exchange Rates**, and then complete the fields.
   * Start Date: (first day of next month)
   * Euro: 1.13
3. Click **Save.** If you are in a Classic tab, close it.

**Edit the Euro Currency Test Opportunity**

Next, edit the close date on the Euro opportunity to see how the converted amount changes depending on the exchange rate.

1. Click the **App Launcher** App launcher icon and select **Sales**.
2. Click the **Opportunities** tab.
3. From the Recently Viewed opportunities list, click **Euro Currency Test** and click the **Details** tab to compare the converted amount to the one you noted earlier.
4. Click **Edit**.
5. Change the **Close Date**to the first date of next month.
6. Click **Save**.

**Create a New Home Page Using Lightning App Builder**

1. From Setup, enter Lightning App in the Quick Find box and select **Lightning App Builder**.
2. Click **New**.
3. Select **Home Page**, then **Next**.
4. Assign the label Support Home Page, then click **Next**.
5. Click the **Standard Home Page** template.
6. Click **Done**.

Display the five most recent cases.

1. Drag the **Recent Items** component to any spot in the canvas.
2. For Custom Label, enter Recent Cases.
3. Click **Select**.
4. Move the currently selected object, **API Anomaly Event Store**, into the Available column by selecting it and clicking the Left Arrow.
5. Select **Case** from Available and click the Right Arrow to move it into the Selected Column.
6. Click **OK**.
7. For Number of Records to Display, enter 5.

Show Chatter posts where the support rep is @mentioned.

1. Drag the **Chatter Feed** component to any spot in the canvas.
2. For Feed Type, select **To Me**.

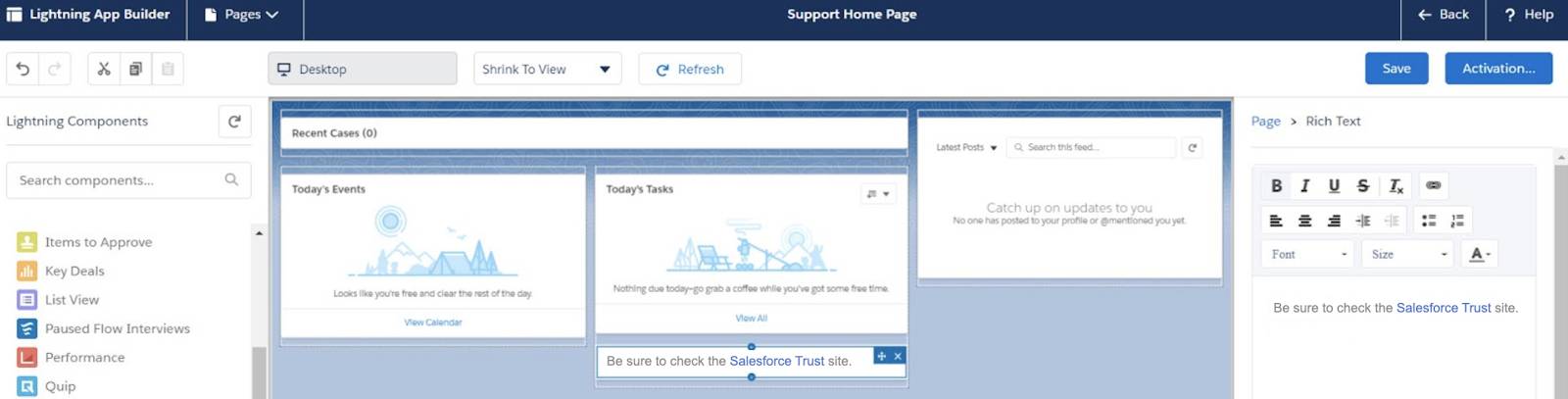
Show today’s tasks and upcoming events.

1. Drag the **Today’s Tasks** component to any spot in the canvas.
2. Drag the **Today’s Events** component to any spot in the canvas.

Display a link to the Salesforce trust site.

1. Drag the **Rich Text**component into the Today’s Tasks box. A text entry field will appear below the box.
2. In the text entry field, enter Be sure to check the Salesforce Trust site.
3. Highlight the text "**Salesforce Trust**" and select the **Link**button.
4. For URL, enter https://trust.salesforce.com.
5. Click**Save**.

**Activate the new Home Page**

1. Click**Save**in the upper right corner, then **Activate**.
2. Click **App** **and** **Profile**.
3. Click **Assign to Apps and Profiles**.
4. Select **Service Console** and click **Next**
5. Select **Custom: Support Profile** and **System Administrator**.
6. Click **Next** and **Save.**  
   Note: Your layout may differ.
7. Click the **<- Back** button to leave the Lightning App Builder.

Your users with the support profile assigned now have a personalized home page. Now you see that a request has been sent your direction by the VP of Sales. Sales reps need quick access to industry-specific accounts. Let’s address this request and create a custom account list view to meet their needs.

Manage User Access

**Introduction**

In this project, you’ll get hands-on practice setting up some of the basic but oh-so-useful functions a business needs when implementing Salesforce—from adding users and creating Chatter groups to modifying your data model to fit your business needs and automating processes for max efficiency.

You’ll find step-by-step instructions on how to get some important functions going for team members in your Customer Support department. By going through these exercises, you’ll learn what's possible with the Salesforce platform and some of its specific applications. So let’s dive in.

**Launch a New Trailhead Playground**

Before you begin, we strongly recommend creating a new Trailhead Playground. To do this, scroll to the bottom of this page, click the three dots next to **Launch**, and select **Create a Trailhead Playground**. It typically takes 3–4 minutes for Salesforce to create your Trailhead Playground.

Note: Yes, we really mean a brand-new Trailhead playground! If you use an existing org or playground, you can run into problems completing this project.

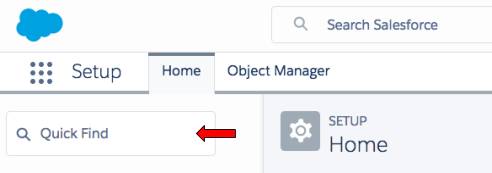
**Create New Users**

It just takes a few clicks, and you’re ready to enter a new user’s information into your Salesforce org. Some fields are mandatory (such as name and email), and others can be skipped if they’re not applicable to a particular user.

Note

The email address you fill in for a new user auto-populates to become that user’s username. Each username must be unique within Salesforce, but that’s not the case with email addresses. For this project, you’ll fill in your own email address for new users. When their usernames auto-populate with your email address, replace the usernames with a unique username using this formula:user’s first initial + user’s last name @ your initials + your favorite color + .com. For example: ahartzler@DKorange.com

Kenya Collins, the manager over your Customer Support department, asked you to add her new team members to your org. Let’s start by creating a user record for Fumiko Suzuki, a member of the Customer Support team headquartered in Japan. Use these details:

1. Click the **setup gear** Gear icon and select **Setup**.
2. Enter Users in the Quick Find Box, then select **Users**. 
3. Click **New User**. Complete the new user record with these details:

| **Field** | **Value** |
| --- | --- |
| First Name | Fumiko |
| Last Name | Suzuki |
| Alias | fsuzu |
| Email | (enter an email address) |
| Username | (auto-populates with your email address-replace using the formula in the yellow note box above) |
| Nickname | fsuzuki |
| Title | Customer Support Rep |
| Department | Customer Support |
| Role | **Customer Support, International** |
| User License | **Salesforce Platform** |
| Profile | **Standard Platform User** |

1. Next, fill in her Locale settings, including time zone and language.
   * Time Zone = **(GMT +09:00) Japan Standard Time (Asia/Tokyo)**
   * Locale = **Japanese (Japan)**
   * Language = **Japanese**
2. Click **Save & New.**
3. Follow the same steps to add American-based customer support rep Aaron Hartzler, sticking with the default locale settings and using these details:

| **Field** | **Value** |
| --- | --- |
| First Name | Aaron |
| Last Name | Hartzler |
| Alias | ahart (auto-populates) |
| Email | (fill in your own email address) |
| Username | (auto-populates with your email address-replace using the formula in the yellow note box above) |
| Nickname | ahartzler |
| Title | Customer Support Rep |
| Department | Customer Support |
| Role | **Customer Support, North America** |
| User License | **Salesforce** |
| Profile | **Standard User** |

1. Click **Save.**

**Verify Your Email Address**

In a future step, you will be sending a test email from within Salesforce. Salesforce requires that users have a verified email address to send emails through Salesforce, such as sends to contacts or leads. So while you are setting up users, it is a perfect time to verify your user email address is well... verified.

1. From setup, click **Users**.
2. Click on *[your name]* to open the Advanced User Details.
3. Click **[Verify]**next to your email address. Note: If it already says [Verified] next your email address then you are good and do not need to do anything additional.
4. Click **OK** in the pop-up window.
5. Check your email and click the verification link.