SALESFORCE FUNDAMENTALS

- Describe the capabilities of the core CRM objects in the Salesforce schema.
- Given a scenario, identify the boundaries of declarative customization and the use cases for programmatic customization.
- Identify common scenarios for extending an org using the AppExchange.

→ When you convert lead records, standard lead fields map to contact, account, person account, and opportunity fields. If you use custom fields, your admin specifies the fields that they map to in your newly created records.
→ When you convert qualified leads, Salesforce moves any campaign members to the new contacts.
→ All open and closed activities from the leads are attached to the accounts, contacts, and opportunities.
→ You can assign the owners of these new records, and schedule follow-up tasks. When you assign new owners, only the open activities are assigned to the new owner.
→ You can't view converted leads, unless your admin has assigned you the "View and Edit Converted Leads" permission.
→ With person accounts enabled, you can convert leads to either person accounts or business accounts. Leads that don't have a value in the Company field are converted to person accounts. Leads that do have a value in the Company field are converted to business account.
→ If your organization uses territory management, account assignment rules evaluate the new account and can assign it to one or more territories.
→ You can't convert a lead that's associated with an active approval process or has pending workflow actions.
→ Each target field can have only one source field mapped to it.
→ You can't delete lead, account, contact, or opportunity fields included in Lead Custom Field Mapping. To delete a field, first remove its mapping.
→ You can't change the field type of a field that's included in Lead Custom Field Mapping. To change a field's type, remove its mapping first.
→ Lead queues can be created to prioritize and assign lead records to support agents.
→ Lead assignment rules can be set up to assign leads automatically.
→ Records of related objects cannot be copied when a lead is converted.
→ An account hierarchy displays up to 500 child accounts in Salesforce Classic. The hierarchy doesn't display details of accounts you don't have permission to view.
→ For companies with multiple office locations, you can also use the Account Site field to distinguish among the locations.
→ If divisions are enabled, accounts that are related via the Parent Account field don't have to be in the same division.
→ Person accounts aren't supported for the Parent Account field or the View Account Hierarchy action.
→ Person accounts store information about individual people by combining certain account and contact fields into a single record.
→ Person accounts can't be included in account hierarchies.

→ Person accounts can't have direct relationships with other accounts or contacts. You can, however, use Contacts to Multiple Accounts to create indirect relationships between a person account and another person account, business account, or contact.
→ To enable Person Accounts, contact support.
→ After Person Accounts is enabled, it can't be disabled.
→ When you use Contacts to Multiple Accounts, each contact still requires a primary account (the account in the Account Name field). The contact and its primary account have a *direct* relationship. But you can add other accounts to the contact. These secondary account-contact relationships are *indirect*.
→ To remove a direct relationship between a contact and an account, change the contact's primary account or delete the contact.
→ Indirect contacts aren't available in reports run from an Account & Contact standard report type.
→ You can import account-contact relationships with the Data Loader, but not the Data Import Wizard.
→ When you have access to a particular account, the related contacts that you have Read access to appear in the Related Contacts related lists. When you have access to a particular contact, all related accounts appear in the Related Accounts related list, but you can open only the accounts you have Read access to.
→ Contact roles specify the part that each contact plays in an account, case, contract, or opportunity.
→ In Salesforce Classic, contact roles are available for accounts, opportunities, cases, and contracts. In Lightning Experience, contact roles are available for opportunities and contracts. Although contact roles work more or less the same for each available object, they work independently. Therefore, you need to set them up and use them separately for each object.
→ To view the contact records that are listed in the Contact Roles related lists, you must have the proper user permissions and sharing access to those records.
→ Opportunity records track details about deals, including which accounts they're for, who the players are, and the amount of potential sales.
→ Products are a base catalog of all the items and services you sell and their standard prices. Price Books let you create a custom collection of products with associated list prices for specific uses.
→ You cannot add a product from a different price book to an opportunity if a price book has already been selected.
→ When you remove a product or price book, all related price book entries are removed. However, if several price books contain the same product, removing the product from one price book doesn't affect the other price books.
→ To remove a product completely, remove it from the standard price book.
→ If opportunities, quotes, service contracts, or contract line items are associated with a product, you can't remove the product.
→ You can't remove a product or price book that's associated with an approval process or has a pending workflow action.
→ Quotes in Salesforce represent the proposed prices of your company's products and services. You create a quote from an opportunity and its products. Each opportunity can have multiple associated quotes, and any one of them can be synced with the opportunity.
→ Relevant price books, products, and list prices must be active in an opportunity before you can create a quote for the opportunity.
→ Can set up record types for quotes.

→ If a quote is synced with an opportunity, and you add a line item to the quote, the new line item is copied to the opportunity as a product in the Products related list.
→ If an opportunity is syncing with a quote, deleting the quote line item deletes the corresponding product from the opportunity's Products related list.
→ Quote PDFs don't support right-to-left languages. The text aligns to the left side of the page instead of the right.
→ Text fields in a related list in a quote PDF are truncated to fewer than 256 characters. This limit occurs on rich text area fields, other types of text fields, standard fields, and custom fields.
→ Your Salesforce admin determines which status is the default for new quotes and which statuses allow you to email a quote PDF. For example, your admin can prevent you from emailing a quote whose status is In Review.
→ A contract is a written agreement between parties. Many companies use contracts to define the terms for doing business with other companies.
→ When you create a contract, consider the type of information that's needed to maximize the quality of your data.
    - The start date (required)
    - How long the contract will be in effect: the term in months (required)
    - The end date
→ When you create an order, the order's start date must fall between the associated contract's start and end dates.
→ Associate each order with an account and optionally with a contract
→ If you clone an order with products, you can't change the new order's currency or price book.
→ Use reduction orders to track requests to reduce, return, deactivate, or disable a customer's products or services.
→ If you're using Lightning Experience, you can't create reduction orders. Switch to Salesforce Classic.
→ You can reduce activated orders only.
→ You can apply a reduction order toward one order only. To reduce order products across multiple orders, create a reduction order for each original order.
→ Cases are the backbone of Service Cloud. Cases let you respond to and solve customer issues.
→ Set up queues, assignment rules, auto response rules, and escalation rules to manage cases.
→ When you change a contact, the account doesn't update to the contact's account, but you can edit the account yourself.
→ Contacts who are portal users can only view cases associated with the account on their contact record.
→ Campaigns allows you pull in data from a third-party source into Salesforce.
→ It also allows you to create campaigns and add campaign members.
→ Campaign members can only be added from lead, contact, or person account records. They cannot be added from account records.
→ Can use campaign influence to decide how to credit each campaign that contributes to bringing in an opportunity.
→ Use workflows for a single if/than statement.
→ Use process builder for multiple if/then statements.
→ Use Flow for more complex logic.
→ Use apex to perform complex validation over multiple objects or complex business processes that are not supported by workflow.

→ Validation rules doesn't allow you to look at information from a child record to validate the parent record.
→ To get around this, create a roll up summary field.
→ It must be in a master-detail relationship. It doesn't work with a lookup relationship.
→ Workflow rules work best with master-detail relationship.
→ Order of operations can cause the automation to not fire properly thus making the decision on which tool to use a little more complex.
→ AppExchange is a marketplace to buy, build, sell, and install apps for Salesforce.com
→ Common use cases include:
    - Marketing Automation (Marketo)
    - Data quality (duplicate prevention)
    - Connectors (to variety of external systems)
    - Expense management (Concur)

https://help.salesforce.com/articleView?id=leads_setup_parent.htm&type=5
https://help.salesforce.com/articleView?id=leads_manage_parent.htm&type=5
https://help.salesforce.com/articleView?id=leads_supplemental.htm&type=5
https://help.salesforce.com/articleView?id=products_pricebooks.htm&type=5
https://help.salesforce.com/articleView?id=opportunities.htm&type=5
https://help.salesforce.com/articleView?id=quotes_overview.htm&type=5
https://help.salesforce.com/articleView?id=contract_def.htm&type=5
https://help.salesforce.com/articleView?id=order_overview.htm&type=5
https://help.salesforce.com/articleView?id=accounts.htm&type=5
https://help.salesforce.com/articleView?id=contacts_overview.htm&type=5
https://help.salesforce.com/articleView?id=campaigns_def.htm&type=5
https://help.salesforce.com/articleView?id=process_which_tool.htm&type=5

DATA MODELING AND MANAGEMENT

- Given a scenario, determine the appropriate data model.
- Describe the capabilities of the various relationship types and the implications of each on record access, user interface, and reporting.
- Identify the considerations when changing a field's type.
- Given a set of requirements, identify the considerations and select the appropriate field type.
- Describe the capabilities and considerations of the schema builder.
- Describe the options and considerations when importing and exporting data.
- Describe the capabilities of and use cases for external objects.

→ Each custom object can have up to two master-detail relationships and many lookup relationships.
→ Each Relationship is included in the max number of custom fields allowed.
→ The differences between relationship types include how they handle data deletion, record ownership, security, and required fields in page layouts.
→ Lookup relationships allow data from the two related objects to be joined in one report.
→ Master-detail relationships allow data from three objects to be joined in one report; the master object, the detail object, plus one other lookup object.

→ A separate report type is available based on each lookup.
→ Many-to-many relationships provide two standard report types that join the master objects and the junction object.
    - “Primary master with junction object and secondary master” in the primary master object's report category.
    - “Secondary master with junction object and primary master” in the secondary master object's report category.
→ You can convert a master-detail relationship to a lookup relationship as long as no roll-up summary fields exist on the master object.
→ You can convert a lookup relationship to a master-detail relationship, but only if the lookup field in all records contains a value.
→ You can create a relationship from an object to itself, but it must be a lookup relationship, and a single record can't be linked to itself.
→ When you define a master-detail relationship, the custom object on which you are working is the “detail” side.
→ By default, records can’t be reparented in master-detail relationships.
→ Administrators can, however, allow child records in master-detail relationships on custom objects to be reparented to different parent records by selecting the *Allow reparenting* option in the master-detail relationship definition.
→ You can have up to three custom detail levels.
→ Standard objects can't be on the detail side of a custom object in a master-detail relationship.
→ An object can appear once in multilevel master-detail relationships.
→ Custom report types created for multilevel master-detail relationships count towards the organizations custom report type limit and no reports are generated if this limit is exceeded.
→ Custom junction objects can't have detail objects. That is, a custom junction object can't become the master object in a multilevel master-detail relationship.
→ You can’t delete a custom object if it is on the master side of a master-detail relationship. If you delete a custom object that is on the detail side of a master-detail relationship, the relationship is converted to a lookup relationship.
→ Junction object records are deleted when either associated master record is deleted and placed in the Recycle Bin. If both associated master records are deleted, the junction object record is deleted permanently and can't be restored.
→ Sharing access to a junction object record is determined by a user's sharing access to both associated master records
→ The first master-detail relationship you create on your junction object becomes the *primary* relationship.
→ The junction object records inherit the value of the Owner field from their associated primary master record.
→ The second master-detail relationship you create on your junction object becomes the *secondary* relationship. If you delete the primary master-detail relationship or convert it to a lookup relationship, the secondary master object becomes primary.
→ Roll-up summary fields that summarize data from the junction object can be created on both master objects.
→ Formula fields and validation rules on the junction object can reference fields on both master objects.
→ A junction object can't be on the master side of another master-detail relationship.

→ If the lookup field is optional, you can specify one of three behaviors to occur if the lookup record is deleted: Clear the value of this field, Don't allow deletion of the lookup record that's part of a lookup relationship, Delete this record also.
→ Only convert custom fields for which no data exists or you risk losing your data.
→ Changing the data type of an existing custom field can cause data loss in the following situations: Changing to or from type Date or Date/Time, Changing to Number from any other type, Changing to Percent from any other type, Changing to Currency from any other type Changing from Checkbox to any other type, Changing from Picklist (Multi-Select) to any other type, Changing to Picklist (Multi-Select) from any other type, Changing from Auto Number to any other type, Changing to Auto Number from any type except Text, Changing from Text to Picklist, Changing from Text Area (Long) to any type except Email, Phone, Text, Text Area, or URL
→ If data is lost, any list view based on the custom field is deleted, and assignment and escalation rules may be affected.
→ You can't change the data type of any custom field that is mapped for lead conversion.
→ If you change the data type of a custom field that is set as an external ID, choosing a data type other than text, number, or email causes the field to no longer act as an external ID.
→ The option to change the data type of a custom field is not available for all data types. For example, existing custom fields cannot be converted into encrypted fields nor can encrypted fields be converted into another data type.
→ In Salesforce Knowledge article types, the file field type can't be converted into other data types.
→ You cannot change the data type of a custom field referenced by other items in Setup, like Visualforce pages, Apex code, Process Builder, or flows.
→ Changing a custom field type may require changing many records at once. To process these changes efficiently, your request may be queued, and you may receive an email notification when the process has completed.
→ Before changing a custom field's type, make sure that it isn't the target of a workflow field update or referenced in a field update formula that would be invalidated by the new type.
→ The data in any auto-number field remains unchanged if you convert it into a text field. Also, you can safely convert a text custom field into an auto-number field without losing your data. Converting an auto-number field into any other data type results in data loss. Auto-number fields can contain a maximum of 30 characters. Before converting a text custom field into an auto-number field, change any records that contain more than 30 characters in that field.
→ Formula fields are special read-only fields that cannot be converted to any other data type. Likewise, you cannot convert any other field type into a formula field.
→ Changing your custom picklists into custom checkboxes is simple. If you select Checkbox as the new data type, you can choose which picklist values to map to checked boxes and unchecked boxes. You can change custom picklists into multi-select picklists without losing any data. Since your records only contain a single value of that picklist, that value is still selected but users can select more values.
→ When you convert a long text area field to an Email, Phone, Text, Text Area, or URL type field, the data in your records is truncated to the first 255 characters of the field.
→ You can only convert rich text area fields into long text area fields. Any images are deleted the next time the long text area field is saved. After converting, markup is hidden in the long text area field but it is not removed from the record until you save the record. That way, the markup can be restored if you change your mind.
→ If you change a multi select picklist would loose all but one of the values chosen.

→ The Schema Builder is how a database is arranged and is a visual representation of the objects in your org.
→ It shows you the relationships between the objects.
→ Can create and delete custom objects and fields.
→ Also create junction objects that relates 2 objects with one another.
→ External objects are not supported in the schema builder.
→ Cannot add geo location fields and cannot export.
→ Two tools for import: Data Import Wizard and Data Loader.
→ The Data Import Wizard makes it easy to import data for many standard Salesforce objects, including accounts, contacts, leads, solutions, campaign members, person accounts, and custom objects.
→ You can import up to 50,000 records at a time.
→ Data Loader is a client application for the bulk import or export of data.
→ Support for large files with up to 5 million records
→ Data Import Wizard does not allow you to export.
→ Use Data Loader when: You need to load 50,000 to 5,000,000 records, You need to load into an object that is not yet supported by the import wizards, You want to schedule regular data loads, such as nightly imports, You want to export your data for backup purposes.
→ Use the import wizard when: You are loading less than 50,000 records, The object you need to import is supported by import wizards, You want to prevent duplicates by uploading records according to account name and site, contact email address, or lead email address.
→ Tables in external systems map to external objects in Salesforce, combining all your data and content for users in your org.
→ External objects are similar to custom objects, except that they map to data that's stored outside your Salesforce org. Each external object relies on an external data source definition to connect with the external system's data. Each external object definition maps to a data table on the external system. Each of the external object's fields maps to a table column on the external system. External objects enable your users and the Lightning Platform to search and interact with the external data.
→ External objects are best used when you have a large amount of data that you can't or don't want to store in your Salesforce organization, and you need to use only a small amount of data at any one time.

https://help.salesforce.com/articleView?id=overview_of_custom_object_relationships.htm&type=5
https://help.salesforce.com/articleView?id=notes_on_changing_custom_field_types.htm&type=5
https://help.salesforce.com/articleView?id=adding_fields.htm&type=5
https://help.salesforce.com/articleView?id=schema_builder.htm&type=5
https://help.salesforce.com/articleView?id=schema_builder_working.htm&type=5
https://help.salesforce.com/articleView?id=external_data_sources.htm&type=5
https://help.salesforce.com/articleView?id=external_object_define.htm&type=5

SECURITY

- Describe the features and capabilities available to restrict and extend object, record, and field access.
- Given a set of business requirements, determine the appropriate sharing solution.

→ Define the default access level for an object's records with organization-wide sharing settings.
→ Organization-wide sharing settings can be set separately for custom objects and many standard objects, including assets, campaigns, cases, and accounts and their contracts.
→ For most objects, organization-wide sharing settings can be set to Private, Public Read Only, or Public Read/Write.
→ In environments where the organization-wide sharing setting for an object is Private or Public Read Only, an admin can grant users additional access to records by setting up a role hierarchy or defining sharing rules.
→ However, sharing rules can only be used to grant additional access—they cannot be used to restrict access to records beyond what was originally specified with the organization-wide sharing defaults.
→ Org Wide Defaults are the most restrictive and you can open up access with roles and sharing rules.
→ Controlling Access Using Hierarchies = Determine whether users have access to records they don't own, including records to which they don't have sharing access, but someone below them in the hierarchy does.
→ Deselect *Grant Access Using Hierarchies* under Setup if you want to prevent users from gaining automatic access to data owned by or shared with their subordinates in the hierarchies.
→ Regardless of your organization's sharing settings, users can gain access to records they do not own through other means such as user permissions like "View All Data," sharing rules, or manual sharing of individual records.
→ The *Grant Access Using Hierarchies* option is always selected on standard objects and is not editable.
→ If you disable the Grant Access Using Hierarchies option, sharing with a role or territory and subordinates only shares with the users directly associated with the role or territory selected. Users in roles or territories above them in the hierarchies will not gain access.
→ If your organization disables the *Grant Access Using Hierarchies* option, activities associated with a custom object are still visible to users above the activity's assignee in the role hierarchy.
→ If a master-detail relationship is broken by deleting the relationship, the former detail custom object's default setting is automatically reverted to Public Read/Write and *Grant Access Using Hierarchies* is selected by default.
→ The *Grant Access Using Hierarchies* option affects which users gain access to data when something is shared with public groups, personal groups, queues, roles, or territories.
→ When you deselect Grant Access Using Hierarchies, notify users of the changes in report results that they can expect due to losing visibility of their subordinates' data.
→ Profiles define how users access objects and data, and what they can do within the application.
→ Your org includes several standard profiles where you can edit a limited number of settings.
→ Instead of creating profiles, save time by cloning existing profiles and customizing them.
→ A permission set is a collection of settings and permissions that give users access to various tools and functions.
→ The settings and permissions in permission sets are also found in profiles, but permission sets extend users' functional access without changing their profiles.
→ Users can have only one profile but, depending on the Salesforce edition, they can have multiple permission sets.
→ You can assign permission sets to various types of users, regardless of their profiles.
→ Can only assign users that have the same user license such as Salesforce or Force.com. You can't assign someone from Force.com and Salesforce user license the same permission set.

→ Can clone permission sets and can edit permission sets from the user record.
→ Sharing rules allow you to make automatic exceptions to your organization-wide sharing settings for defined sets of users.
→ Criteria-based sharing rules determine whom to share records with based on field values in records.
→ You can create up to 50 criteria-based sharing rules per object.
→ If multiple sharing rules give a user different levels of access to a record, the user gets the most permissive access level.
→ Sharing rules automatically grant additional access to related records.
→ Users in the role hierarchy are automatically granted the same access that users below them in the hierarchy have from a sharing rule, provided that the object is a standard object or the *Grant Access Using Hierarchies* option is selected.
→ When you make changes to groups, roles, and territories, sharing rules are reevaluated to add or remove access as necessary.
→ For accounts, opportunities, and cases, record owners can use teams to allow other users access to their records.
→ Sometimes it's impossible to define a consistent group of users who need access to a particular set of records. In those situations, record owners can use manual sharing to give read and edit permissions to users who would not have access to the record any other way.

https://help.salesforce.com/articleView?id=security_data_access.htm&type=5
https://help.salesforce.com/articleView?id=permissions_about_users_access.htm&type=5
https://help.salesforce.com/articleView?id=admin_userprofiles.htm&type=5
https://help.salesforce.com/articleView?id=customize_fieldaccess.htm&type=5
https://help.salesforce.com/articleView?id=user_groups.htm&type=5
https://help.salesforce.com/articleView?id=managing_the_sharing_model.htm&type=5
https://help.salesforce.com/articleView?id=security_overview_passwords.htm&type=5
https://help.salesforce.com/articleView?id=security_overview_network.htm&type=5
https://help.salesforce.com/articleView?id=basics_identity.htm&type=5
https://help.salesforce.com/articleView?id=users_profiles_epui_login_ip_ranges.htm&type=5

BUSINESS LOGIC AND PROCESS AUTOMATION

- Describe the capabilities of and use cases for record types.
- Describe the capabilities of and use cases for formula fields.
- Describe the capabilities of, use cases for, and implications of roll-up summary fields.
- Describe the capabilities of and use cases for validation rules.
- Describe the capabilities of and use cases for approval processes.
- Describe the capabilities of and use cases for workflow, visual workflow, and Process Builder.
- Given a set of business requirements, recommend a solution to automate business processes.
- Describe the ramifications of field updates and the potential for recursion.

→ Record types let you offer different business processes such as a process for hardware and software, picklist values, and page layouts to different users.

→ Before creating record types, include all of the possible record type values in your master list of picklists. The master picklist is a complete list of picklist values that can be used in any record type.
→ The master picklist is independent of all record types and business processes. If you add a picklist value to the master picklist, you must manually include the new value in the appropriate record types. If you remove a picklist value from the master, it is no longer available when creating new records, but records assigned to that value are unchanged.
→ A user can be associated with several record types. You can make one of the record types the default.
→ When a user converts a lead, the new account, contact, and opportunity records use the default record type for the owner of the new records. The user can choose a different record type during conversion.
→ When a user clones a record, the new record has the record type of the cloned record. If the user's profile doesn't have access to the record type of the cloned record, the new record adopts the user's default record type.
→ When a user creates a case or lead and applies assignment rules, the new record can keep the creator's default record type or take the record type of the assignee, depending on the case and lead settings specified by the administrator.
→ For opportunities, cases, leads, and solutions, a sales, support, lead, and solution process must be set up first before a record type can be created.
→ A roll-up summary field calculates values from related records, such as those in a related list. You can create a roll-up summary field to display a value in a master record based on the values of fields in a detail record. The detail record must be related to the master through a master-detail relationship.
→ You can perform different types of calculations with a roll-up summary field. You can count the number of detail records related to a master record. Or, you can calculate the sum, minimum value, or maximum value of a field in the detail records such as the sum of the amount on all opportunities.
→ Create roll-up summary fields on:
    - Any custom object that is on the master side of a master-detail relationship
    - Any standard object that is on the master side of a master-detail relationship with a custom object
    - Opportunities using the values of opportunity products related to the opportunity
    - Accounts using the values of related opportunities
    - Campaigns using campaign member status or the values of campaign member custom fields
→ The types of fields you can calculate in a roll-up summary field depend on the type of calculation. For example,
    - Number, currency, and percent fields are available when you select SUM as the roll-up type.
    - Number, currency, percent, date, and date/time fields are available when you select MIN or MAX as the roll-up type.
→ Sometimes you can't change the field type of a field that you reference in a roll-up summary field.
→ When you delete a child record on a roll-up Summary field, Salesforce doesn't recalculate the value of the field. Select the Force a mass recalculation on this field option on the edit page of the roll-up summary field to manually recalculate the value.

→ You can't use long text area, multi-select picklist, Description fields, system fields like Last Activity, cross-object formula fields, and lookup fields in the field column of roll-up summary filters.
→ Auto number fields are not available as the field to aggregate in a roll-up summary field.
→ Roll-up summary fields are not available for mapping lead fields of converted leads.
→ If a roll-up summary field doesn't contain cross-object field references or functions that derive values on the fly, such as NOW or TODAY, it can calculate the values of formula fields.
→ Apply field-level security to your roll-up summary fields if they calculate values that you do not want visible to users.
→ If you have validation rules, consider how they affect roll-up summary fields.
→ Because roll-up summary fields are not displayed on edit pages, you can use them in validation rules but not as the error location for your validation.
→ Avoid referencing a roll-up summary field from a child record.
→ Date and Date/Time Formula Fields:
→ Dates and times are always calculated using the user's time zone.
→ Date and date/time fields can't be used interchangeably. The name alone may not indicate if a field is a date or date/time. For example, Created Date and Last Modified Date are date/time fields whereas Last Activity Date is a date field. Use the DATEVALUE function to convert a date/time field into a date field.
→ Use addition and subtraction operators with date or date/time fields to calculate duration. For example, subtract a date from another date to calculate the number of days between the two. Likewise, you can subtract the date/time from another date/time to get the number of days between the two as a number. See NOW or TODAY for suggested use.
→ Use addition and subtraction operators with numbers to return another date or date/time. For example, {!CreatedDate} + 5 calculates the date and time five days after a record's created date. Note that the expression returns the same data type as the one given; a date field plus or minus a number returns a date, and a date/time field plus or minus a number returns a date/time.
→ When calculating dates using fractions, Salesforce ignores any numbers beyond the decimal. For example:
    - TODAY() + 0.7 is the same as TODAY() + 0, which is today's date.
    - TODAY() + 1.7 is the same asTODAY() + 1, which is tomorrow's date.
    - TODAY() + (-1.8) is the same as TODAY() + (-1), which is yesterday's date.
→ To calculate the value of two fractions first, group them within parentheses. For example:
    - TODAY() + 0.5 + 0.5 is the same as TODAY() + 0 + 0, which is today's date.
    - TODAY() + (0.5+0.5) is the same as TODAY() + 1, which is tomorrow's date.
→ Years can't be zero and must be between -4713 and 9999.
→ Merged Fields in Formulas:
→ Delegated administrators need to have access to custom objects to access the objects' merge fields from formulas.
→ In account formulas, all business account fields are available as merge fields. However, account fields exclusive to person accounts such as Birthdate and Email are not available.
→ You can't use formula fields that include related object merge fields in roll-up summary fields.
→ Formulas and roll-up summary fields can't reference fields on external objects.
→ Using RecordType.Id can make your formula less readable; when you do use it, write in-line comments into the formula to clarify.
→ To determine if a record is a task or event, use the IsTask merge field. For example:

→ To reference the unique identifier for your Salesforce organization in a formula, insert the $Organization.Id merge field. This merge field can display anywhere formula fields can except in reports.

→ Some merge fields display as radio buttons but function like picklist fields when referenced in a formula.

Use the values “Read,” “Edit,” and “None” in a formula when referencing:

- $UserRole.CaseAccessForAccountOwner
- $UserRole.OpportunityAccessForAccountOwner
- CaseAccessLevel (on Territory)
- OpportunityAccessLevel (on Territory)

Use the values “Read,” “Edit,” and “All” in a formula when referencing:

- AccountAccessLevel (on Territory)

→ If you create a contacts formula field that references account merge fields, that field can be included in contact page layouts but should not be included in person accounts page layouts. The formula field will display a value of #Erroron the person accounts page.

→ Numer Formula Fields:

→ Use the decimal version of a percent when working with percent fields in formulas. For example, IF(Probability =1...) for 100% probability or IF(Probability =0.9...) for 90% probability.

→ Reference auto-number fields as text fields in formulas.

→ The output of your formula must be less than 19 digits.

→ Formulas can contain a mix of numbers, percents, and currencies as in this example: AnnualRevenue / NumberOfEmployees.

→ Salesforce uses the round half up tie-breaking rule for numbers in formula fields. For example, 12.345 becomes 12.35 and −12.345 becomes −12.34.

→ Picklist and Multi-Select Picklist Formula Fields:

→ You can use special picklist fields in your formulas, such as IsEscalated for cases and IsWon for opportunities.

→ Picklist fields can only be used in these functions:

- ISPICKVAL—Compares the value of a picklist to a single value.
- CASE—Compares the value of a picklist to multiple values.
- TEXT—Returns a picklist value’s API Name so that you can work with a reference to the value (even if the displayed value changes) in functions that support text values, such as CONTAINS. (Available in only flow formula resources, formula fields, validation rules, and workflow field updates.)

→ Multi-select picklist fields can only be used in these functions:

- INCLUDES
- ISBLANK
- ISNULL
- ISCHANGED (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it’s edited**)
- PRIORVALUE (Only in assignment rules, validation rules, workflow field updates, and workflow rules in which the evaluation criteria is set to **Evaluate the rule when a record is: created, and every time it’s edited**)

→ Referencing Record Types in Formulas:

→ Reference record types in formulas if you want different workflow rules, validation rules, and lookup filters to apply to different record types. For example, you can:

- Create a workflow rule on accounts that emails different teams based on the account record type the user selects when creating the account.
- Create a validation rule on opportunities that allows only members of the North American sales team to save opportunities with the Domestic record type.

→ When possible, use `RecordTypeId` instead of `RecordType.Name` to reference a specific record type. While `RecordType.Name` makes a formula more readable, you must update the formula if the name of the record type changes, whereas the ID of a record type never changes. Also, `RecordType.Name` requires a cross-object reference to the record type, while `RecordTypeId` doesn't. However, if you are deploying formulas across organizations (for example, between sandbox and production), use `RecordType.Name` because IDs are not the same across organizations.

→ Avoid using `$RecordType` in formulas, except in default value formulas. Instead, use the `RecordType` merge field (for example, `Account.RecordType.Name`) or the `RecordTypeId` field on the object.

→ Text Formula Fields:

→ Before using the HYPERLINK function, consider the differences between hyperlinks and custom links.

- Hyperlink formula fields are just like other custom fields that you can display in list views and reports.
- Custom links display on detail pages in a predefined section; hyperlink formula fields can display on a detail page wherever you specify.
- Using custom links, you can specify display properties such as window position and opening in a separate popup position; hyperlink formula fields open in a new browser window by default or you can specify a different target window or frame.
- Your formulas can reference custom links. Before deleting a custom link, make sure it is not referenced in a formula field.
- Hyperlink formula fields that contain relative URLs to Salesforce pages, such as /rpt/reportwizard.jsp, can be added to list views, reports, and related lists. However, use a complete URL, including the server name and https://, in your hyperlink formula before adding it to a search layout.

→ To insert text in your formula field, surround the text with quotation marks. For example, to display "CASE: 123," use this formula "CASE: "& CaseNumber__c.

→ Use the backslash (\) character before a quote or backslash to insert it as a literal value in your output. For example, "Trouble\\Case \"Ticket\": " in your formula displays Trouble\Case "Ticket": on detail pages.

→ Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record.

→ Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

→ You can specify the error message to display when a record fails validation and where to display it.

→ When one validation rule fails, Salesforce continues to check other validation rules on that field or other fields on the page and displays all error messages at once.

→ Campaign hierarchies ignore validation rules.

→ Salesforce runs validation rules before it creates records submitted via Web-to-Lead and Web-to-Case and then creates records that have valid values.

→ Validation rules continue to run on individual records if the owner is changed. If the Mass Transfer tool is used to change the ownership of multiple records, however, validation rules don't run on those records.
→ Validation rule formulas don't or can't refer to:
    - Compound fields, including addresses, first and last names, and dependent picklists and lookups
    - Campaign statistic fields, including statistics for individual campaigns and campaign hierarchies
    - Merge fields for auto-number or compound address fields such as Mailing Address
    - The detail page of a custom activity field doesn't list associated validation rules.
→ Workflow rules and some processes can invalidate previously valid fields. Invalidation occurs because updates to records based on workflow rules and also on process scheduled actions don't trigger validation rules.
→ Process record updates on immediate actions do fire validation rules.
→ You can't create validation rules for relationship group members.
→ You can use roll-up summary fields in validation rules because the fields don't display on edit pages. Do not use roll-up summary fields as the error location.
→ Examples: Postal code is 5 digits, State is two letters, and phone number is 10 digits.
→ An approval process automates how records are approved in Salesforce. An approval process specifies each step of approval, including from whom to request approval and what to do at each point of the process.
→ Users can't see which approval process is triggered when they click **Submit for Approval**. Familiarize users on the criteria for each approval process and what each approval process does. If the record doesn't meet the entry criteria or if they're not an allowed submitter for any approval processes, Salesforce displays an error.
→ Salesforce limits the number of approval processes in your org, as well as the number of steps and actions in each approval process.
→ If the approval object is a detail object in a master-detail relationship, Owner isn't available for approval page layouts or approval post templates.
→ In approval criteria—either the entry criteria or step criteria—don't reference expressions that resolve to random values. That way, if the criteria needs to be evaluated again, the record is evaluated the same every time.
→ An approval process can specify a field update action that reevaluates workflow rules for the updated object. If, however, the re-evaluated workflow rules include a cross-object field update, those cross-object field updates are ignored.
→ Field updates that are executed as approval actions don't trigger workflow rules or entitlement processes.
→ Flows can delete records that are pending approval.
→ Make sure that the assigned approver has access to read the records for the approval requests.
→ Approval processes that let users select an approver manually also let users select themselves as the approver.
→ You can assign an approval request to the same user multiple times in a single step. However, Salesforce sends the user only one request.
→ You can assign approval requests to a queue only if the associated object supports queues. Email approval response isn't supported for approval processes that assign approval to a queue.
→ An approval process must have at least one step before you can activate it.
→ Before you activate your approval process, test it in your Salesforce sandbox.

- → After an approval process is activated, you can't add, delete, or change the order of the steps or change its reject or skip behavior, even if the process is inactive.
- → Example: Discounting opportunities, PTO Requests
- → A *workflow rule* is the main container for a set of workflow instructions. These instructions can always be summed up in an if/then statement.
- → You can create an email alert, field update, outbound message, and task.
- → Salesforce limits the number of total and active rules in your org, the number of time triggers and actions per rule. It also processes a limited number of daily emails and hourly time triggers.
- → Each workflow rule applies to a single object.
- → If you have workflow rules on converted leads and want to use cross-object field updates on the resulting accounts and opportunities, you must enable the lead setting Require Validation for Converted Leads.
- → Workflow rules on custom objects are automatically deleted if the custom object is deleted.
- → If you create workflow rules to replace any Apex triggers, make sure you delete those Apex triggers when you activate the equivalent workflow rules. Otherwise, both Apex triggers and workflow rules will fire and cause unexpected results, such as overwritten field updates or redundant email messages.
- → Workflow rules can be triggered any time a record is saved or created, depending on your rule criteria. However, rules created after saving records aren't triggered by those records retroactively.
- → Workflow rules are triggered when a standard or custom object in a master-detail relationship is re-parented, even if the object's evaluation criteria is set to Evaluate the rule when a record is: created, and any time it's edited to subsequently meet criteria.
- → Saving or creating records can trigger more than one rule.
- → You can't package workflow rules with time triggers.
- → You can't create outbound messages for workflow rules on junction objects.
- → Examples: Follow up before contract expires, Notify account owner of new high priority cases.
- → Process Builder helps you automate your business processes and gives you a graphical representation as you build it.
- → Process Builder supports three types of processes for your automation needs. The type determines what triggers the process.
    - o A record change process starts when a record is created or updated.
    - o An event process starts when a platform event occurs.
    - o An invocable process starts when something else, like another process, invokes it.
- → Each process consists of:
    - o Criteria that determine when to execute an action group.
    - o Action groups, which consist of immediate or scheduled actions. Only record change processes support scheduled actions.
- → With Process Builder, you can:
    - o Create a record of any object type
    - o Update any related record—not just the record or its parent
    - o Use a quick action to create a record, update a record, or log a call
    - o Invoke a process from another process
    - o Launch a flow—you can't schedule this action with workflow
    - o Send an email
    - o Post to Chatter
    - o Submit a record for approval

→ When building processes, keep shared limits and Apex governor limits in mind. In addition, a process's API name must be unique across all processes and flows in your org.
→ Processes start automatically and are invisible to the user.
→ When you create a process, you associate the process with exactly one object. You also specify whether to evaluate only created records or both created and edited records. When you activate a process, it starts when a record change meets those settings.
→ Processes can reevaluate records up to five times in one transaction.
→ For an object, use either workflow rules or processes. If you use both, you can't predict the order in which they're executed.
→ Make sure that your processes don't generate infinite loops.
→ Cloud Flow Designer is a point-and-click tool that lets you automate business processes by building flows. A *flow* is an application that automates a business process by collecting data and doing something in your Salesforce org or an external system.
→ When using flows, keep flow limits and Apex governor limits in mind.
→ Examples: Online Purchase flows display stages as sections on a progress indicator, Transfer ownership of accounts from one user to another.
→ Plan out your flow before you start building.
→ Build your flows in a test environment—like a sandbox or Developer Edition org.
→ Control when running users can navigate backward.
→ Provide an error handler.
→ If the field update changes the field's value, all workflow rules on the associated object are re-evaluated. Any workflow rules whose criteria are met as a result of the field update will be triggered.
→ If any of the triggered workflow rules result in another field update that's also enabled for workflow rule re-evaluation, a domino effect occurs, and more workflow rules can be re-evaluated as a result of the newly-triggered field update. This cascade of workflow rule re-evaluation and triggering can happen up to five times after the initial field update that started it.
→ Make sure that your workflow rules aren't set up to create recursive loops.

https://help.salesforce.com/articleView?id=customize_wf.htm&type=5
https://help.salesforce.com/articleView?id=process_overview.htm&type=5
https://help.salesforce.com/articleView?id=vpm_admin_flow_overview.htm&type=5
https://help.salesforce.com/articleView?id=what_are_approvals.htm&type=5
https://help.salesforce.com/articleView?id=customize_recordtype_considerations.htm&type=5
https://help.salesforce.com/articleView?id=creating_record_types.htm&type=5
https://help.salesforce.com/articleView?id=customize_formuladef.htm&type=5
https://help.salesforce.com/articleView?id=fields_about_roll_up_summary_fields.htm&type=5
https://help.salesforce.com/articleView?id=fields_about_field_validation.htm&type=5
https://help.salesforce.com/articleView?id=workflow_field_update_considerations.htm&type=5

SOCIAL

- Describe the capabilities of and use cases for social features.

→ Increase your users' access to social intelligence and serve customers and prospects better with easy access to social network profiles associated with Salesforce accounts, contacts, and leads.

→ In Salesforce Classic, you can access Twitter and Youtube.
→ In Lightning Experience, you can only access Twitter.
→ Add another source of leads for your sales team by connecting LinkedIn Lead Gen advertisements to Salesforce.
→ When prospective customers fill out a form on your LinkedIn ad, we convert the data directly to new leads in Salesforce.
→ Assign the new leads to the user or queue you choose.
→ You can add up to 500 leads from LinkedIn per day.
→ If the number exceeds the limit, we email the additional leads to the default lead creator you designate in Setup.

https://help.salesforce.com/articleView?id=social_networks_overview.htm&type=5
https://help.salesforce.com/articleView?id=mc_ss_manage_social_accounts_in_workspace.htm&type=5
https://help.salesforce.com/articleView?id=leads_linkedin_lead_gen.htm&type=5

USER INTERFACE

- Describe the user interface customization options.
- Describe the capabilities of and use cases for custom buttons, links, and actions.
- Describe the declarative options available for incorporating Lightning Components in an application.
- Given a scenario, determine the appropriate user interface design.

→ Can add reports to the page layout and set the properties in the page editor.
→ Only 2 reports for a page.
→ Can also embed visualforce pages into page layout.
→ Can add new sections to the page layout.
→ Can add blank space to the page layout to even up the columns.
→ Four types of custom tabs: custom object tabs, web tabs, visualforce tabs, and lightning page tabs.
→ Web pages are for embedding web content into Salesforce.
→ Visualforce tabs are for visualforce pages.
→ Lightning page tabs let you add lightning pages to the mobile app navigation menu.
→ Custom buttons and links help you integrate Salesforce data with external URLs, applications, your company's intranet, or other back-end office systems.
→ Custom links can link to an external URL, such as www.google.com, a Visualforce page, or your company's intranet.
→ With custom buttons you can create a detail page or list page button.
→ A detail page button displays to the right of standard buttons on the record page.
→ A list page button displays on related lists such as new order.
→ Custom buttons that call JavaScript aren't supported in Lightning Experience.
→ Custom buttons with a content source of OnClick JavaScript aren't supported in the Salesforce app.
→ A custom link's label can't exceed 1,024 characters.
→ Actions can be global or object specific.

→ Can be used within the mobile app as well.
→ A Lightning page is a custom layout that lets you design pages for use in the Salesforce mobile app or Lightning Experience.
→ Lightning pages are built using Lightning components—compact, configurable, and reusable elements that you can drag and drop into regions of the page in the Lightning App Builder.
→ You can use a Lightning page to create an app page that you can add to the Salesforce mobile app navigation list or to a Lightning app's navigation bar. An app page gives your users quick access to the objects and items that are most important in that app.
→ You can also use a Lightning page to create a customized record or Home page for Lightning Experience.
→ Standard components are Lightning components built by Salesforce.
→ Custom components are Lightning components that you or someone else have created. With some modifications, custom Lightning components can work in the Lightning App Builder.
→ What standard components are available depends on the object.
→ Some are related list component, record detail component, report chart component, chatter component, highlights panel component, and visualforce component.
→ In order to use custom components, you need to deploy my domain.
→ A Lightning page region can contain up to 25 components.

https://help.salesforce.com/articleView?id=actions_types.htm&type=5
https://help.salesforce.com/articleView?id=customize_enterprise.htm&type=5
https://help.salesforce.com/articleView?id=lightning_app_builder_overview.htm&type=5
https://help.salesforce.com/articleView?id=lightning_page_overview.htm&type=5
https://help.salesforce.com/articleView?id=lightning_page_components.htm&type=5
https://help.salesforce.com/articleView?id=lightning_page_components_custom.htm&type=5

REPORTING

- Describe the features and capabilities available when creating reports, report types, and dashboards.

→ A *report type* defines the set of records and fields available to a report based on the relationships between a primary object and its related objects.
→ Reports display only records that meet the criteria defined in the report type.
→ Select Primary object then secondary objects.
→ Add up to three child objects
→ For each child object, select one of the following criteria:
  - Each "A" record must have at least one related "B" record. Only parent records with child records are shown in the report.
  - "A" records may or may not have related "B" records. Parent records are shown, whether or not they have child records.
→ Can include and exclude fields, and designate the fields that will appear for the beginning of the report.
→ You can add up to 1000 fields to each custom report type.
→ You can't add the following fields to custom report types:
  - Product schedule fields

    - History fields
    - Person account fields
- → The Age field on cases and opportunities
- → A custom report type can contain up to 60 object references.
- → Object references can be used as the main four objects, as sources of fields via lookup, or as objects used to traverse relationships.
- → The number of custom report types depends on Salesforce Edition.
- → Report types associated with custom objects in the Deleted Custom Objects list count against the maximum number of custom report types you can create.
- → Four Report Types:
    - Tabular = Simplest report
    - Summary = Group rows of data
    - Matrix = Group of data by both rows and columns
    - Joined = Let you create multiple report blocks, with each block acting as its own "sub-report" with its own report type, fields, columns, sorting and filtering.
- → Charts:
    - Single groups = horizontal bar, vertical bar, line graph, funnel, pie, donut
    - Multiple groups = side by side bar, stacked bar, stacked to 100% bar, grouped line
- → A dashboard is a visual representation of data from multiple reports and comprises up to 20 components.
- → Components can be charts, tables, gauges, metrics, or other components that you can create with visualforce.
- → Use a running user to determine what data is visible on a dashboard.
- → To schedule and email reports, you need to specify a running user, frequency, start and end dates.
- → You can't have more than 250 groups or 4,000 values in a chart.
- → Reports display a maximum of 2,000 rows.
- → A dashboard filter can have up to 50 options.
- → Each report can include up to five bucket fields.
- → Each bucket field can contain up to 20 buckets.
- → Each bucket can contain up to 20 values.
- → If you schedule a dashboard to refresh on a specific day of every month, it only refreshes on months that have that specific day. For example, if you schedule a refresh for the 31st of every month, the dashboard won't refresh on 30-day months. To refresh on the last day of every month, choose "Last" from the On day of every month drop-down list.
- → Dashboards won't refresh as scheduled if the running user doesn't have access to the dashboard folder.
- → If a dashboard has filters, only the unfiltered version is refreshed.
- → You can't schedule refreshes for dynamic dashboards. They must be refreshed manually.

https://help.salesforce.com/articleView?id=analytics_overview.htm&type=5
https://help.salesforce.com/articleView?id=rd_reports_overview.htm&type=5
https://help.salesforce.com/articleView?id=rd_dashboards_overview.htm&type=5
https://help.salesforce.com/articleView?id=analytics_sharing.htm&type=5

MOBILE

- Describe the declarative customization options available for the Salesforce mobile application user interface.
- Given a set of requirements, determine the appropriate global and object specific actions and action layouts to optimize the Salesforce mobile application user experience.

→ Can receive in-app notifications such as push notifications and approval requests.
→ Can customize the navigation menu.
→ Can include approval requests, canvas apps, chatter, dashboards, events, forecasts, groups, lightning component tabs, lightning pages, news, notes, paused flow interviews, people, reports, smart search items, tasks, today, and visualforce page tabs.
→ The first item in the selected list becomes your users' salesforce app landing page.
→ You can't set different menu configurations for different types of users.
→ When organizing the menu items, put the items that users will use most at the top.
→ To setup Salesforce1, use the setup wizard.
→ Can only use Twitter in the mobile app.
→ Can upload photos and files from your device into the mobile app.
→ Can import contacts from device; however, the admin can uncheck this feature in the settings.
→ Can also make calls on the mobile device and log them in the mobile app.
→ Can take records offline if the admin has enabled Salesforce1 caching and Offline Edit.
→ Can create compact layouts to add only the fields needed so that the user can create records quickly and not have to scroll so much.
→ Compact layouts support all field types except: text area, long text area, rich text area, and multi-select picklist.
→ Global Actions are added to the global publisher layout and can be accessed from the action bar in the bottom of the page on a record page.
→ Object specific actions are for a specific object and can be the same type of action as a global action.
→ Actions to create records for an object that is the detail object in a master-detail relationship must be object-specific, not global.
→ Types of actions: create actions, update actions, log a call actions, custom actions, send email actions, and question actions.
→ Update actions must be object-specific.
→ Question actions enable users to ask and search for questions about the records that they're working on.
→ Custom actions invoke lightning components flows, visualforce pages, or canvas apps with functionality that you define.
→ Salesforce1 global actions don't support multiple record type creations at the action level.
→ The action layout lets you set the fields that you want to be filled in after clicking an action.
→ You can set predefined fields as well in the action layout; however, you cannot set predefined values for multi-select picklists and read-only fields.

https://help.salesforce.com/articleView?id=limits_mobile_sf1_customizations.htm&type=5

https://help.salesforce.com/articleView?id=actionbar_overview.htm&type=5
https://help.salesforce.com/articleView?id=compact_layout_overview.htm&type=5

APP DEVELOPMENT

- Describe the key milestones and considerations when managing the application lifecycle.
- Describe the differences between and considerations when using the various types of sandboxes.
- Describe the capabilities of and considerations when using change sets.
- Describe the use cases of and considerations when using unmanaged packages.
- Given a scenario, determine the appropriate deployment plan.

→ Non-configuration changes, such as creating views, reports, and dashboards, can be made safely in production.
→ Step 1: Plan – An app starts with planning. This step incudes requirements gathering and analysis.
→ Step 2: Build – Admins and software engineers write the app per the design specifications.
→ Step 3: Test – The app is tested to verify that the changes behave as expected and don't cause undesired side-effects.
→ Step 4: Deploy – When testing is successfully completed, and the quality benchmarks have been met, the app can be deployed to production.
→ For Deploying an app:
→ Phase 1: Development & QA – Each developer has his or her own sandbox for creating and testing changes.
→ Phase 4: Integration Testing – If you have multiple developers and more than one project in development that will be released at the same time, you need integration testing to make sure that the separate code lines can be merged without conflict.
→ Phase 3: User Acceptance Testing – Employees in the company can use the user acceptance testing (UAT) sandbox to try out new features or perform ad hoc testing.
→ Phase 4: Staging – The staging environment is the last environment in the development lifecycle before production. The staging sandbox is a Full sandbox that contains all data and metadata that's in production—it's a full replica of production that enables you to perform real-world testing and catch any data-dependent issues that affect the behavior of the app.
→ Phase 5: Production – When all testing is done, and the changes won't break anything, you can push them into production.
→ Types of Sandboxes:
→ Developer = Developer sandboxes copy only the org's configuration, no data. You can create or load up to 200 MB of data, which is enough for many development and testing tasks. You can refresh a Developer sandbox once per day.
→ Developer Pro = A Developer Pro sandbox can store up to 1 GB of data (about 500,000 records). It's otherwise similar to a Developer sandbox.
→ Partial Copy = A Partial Copy is a Developer sandbox, plus a sampling of data that you define in a sandbox template. You have limited control over the data that is copied. You can choose the objects, but not the records to pull. The sandbox can include up to 5 GB of data, which is about 2.5 million records, with a maximum of 10,000 records per object. You can refresh a Partial Copy sandbox every five days.

- → Full = A copy of your production organization and all its data. Because the Full sandbox is an exact copy, the amount of data in the sandbox is the same as your production org. you can refresh a Full sandbox every 29 days.
- → Sandbox templates allow you to pick specific objects and data to copy to your Full or Partial Copy sandbox to control the size and content of each sandbox. Sandbox templates are only available for use with Full or Partial Copy sandboxes.
- → Change sets allow you to move metadata between sandboxes and production but does not allow you to move data between orgs.
- → When you want to send customizations from your current org to another org, create an *outbound change set*. Once you send the change set, the receiving org sees it as an *inbound change set*.
- → Sending a change set between two orgs requires a deployment connection. Change sets can only be sent between orgs that are affiliated with a production org. For example, a production org and a sandbox, or two sandboxes created from the same org can send or receive change sets.
- → Developers can use permission sets or profile settings to specify permissions and other access settings in a change set.
- → When creating a change set, you also add in dependencies to the components. For example, when adding a workflow, can't forget to add the action under dependencies.
- → Before you can deploy a change set from one org to another, an administrator in the target org must authorize uploads across the deployment connection between the two orgs.
- → The Deployment Connections list is automatically populated with your production org and all sandboxes. It is possible to deploy between any of these orgs, but no other orgs.
- → A change set is deployed in a single transaction. If the deployment is unable to complete for any reason, the entire transaction is rolled back. After a deployment completes successfully, all changes are committed to your org and the deployment can't be rolled back.
- → Plan deployments around maintenance schedule
- → Validate change sets before deployment
- → Unmanaged packages are typically used to distribute open-source projects or application templates to provide developers with the basic building blocks for an application. Once the components are installed from an unmanaged package, the components can be edited in the organization they are installed in. The developer who created and uploaded the unmanaged package has no control over the installed components and can't change or upgrade them.
- → Connected apps are not supported for unmanaged packages.
- → Can include permission sets and profile settings
- → Subscribers can edit permission sets in unmanaged packages, but not in managed packages.

https://help.salesforce.com/articleView?id=create_test_instance.htm&type=5
https://help.salesforce.com/articleView?id=deploy_sandboxes_intro.htm&type=5
https://help.salesforce.com/articleView?id=deploy_overview.htm&type=5
https://developer.salesforce.com/docs/atlas.en-us.216.0.dev_lifecycle.meta/dev_lifecycle/lifecycle_adv_sandbox_env.htm
https://developer.salesforce.com/docs/atlas.en-us.packagingGuide.meta/packagingGuide/packaging_developing_unmanaged.htm#
https://help.salesforce.com/articleView?id=changesets.htm&type=5