MODULE 1: DATA AND PROCESS MODEL

Module Overview

Companies use Microsoft Dynamics NAV 2013 to manage their business operations and processes, such as creating and posting sales invoices, controlling the inventory, handling purchases, and so on. Microsoft Dynamics NAV 2013 rich functionality covers the following business processes:

- Financial Management
- Sales
- Marketing
- Purchases
- Warehouse
- Manufacturing
- Resource Planning
- Jobs
- Service Management
- Human Resources

However, most companies have very specific needs that are not covered by the Microsoft Dynamics NAV 2013 standard application. The gaps between the functionality that Microsoft Dynamics NAV 2013 provides and a company's needs can be as small as a simple change to an existing process, or as large as development of a whole new application area.

All application functionality in Microsoft Dynamics NAV 2013 is developed in Microsoft Dynamics NAV Development Environment and in C/AL programming language. Application functionality is contained in the objects that are stored in the database. As a developer, you can change any standard object or change the code. This lets you develop rich and powerful customizations, but also makes it easy to introduce bugs or cause the existing application to stop functioning as expected.

To develop solutions for Microsoft Dynamics NAV 2013, you must understand how the standard application works. You must become familiar with the basic principles that are consistently applied throughout Microsoft Dynamics NAV 2013 and across all its application areas.

Microsoft Dynamics NAV 2013 is consistent and intuitive in how it presents data and manages processes. When you customize Microsoft Dynamics NAV 2013, you may introduce new data structures and new processes. If you do not apply the principles from the standard application, you can easily produce an application that is difficult to use, maintain, and upgrade. Users should be unable to differentiate between standard processes and your custom processes. If your customizations violate standard principles and introduce new concepts or patterns, it is more difficult for users to use, and for you to maintain the application

Objectives

The objectives are:

- Explain the different table types and their characteristics.
- Present the standard data model and introduce the data-related business logic.
- Present the standard process model that governs the transactions in Microsoft Dynamics NAV 2013.

Table Types and Characteristics

As a business management application, Microsoft Dynamics NAV 2013 manages and processes lots of data. All data is stored in tables. From a technical perspective, all tables are the same, as they all contain fields, keys, and triggers. From a functional perspective, there are different types of tables that serve different purposes.

Understanding different table types in Microsoft Dynamics NAV 2013 enables you to efficiently customize existing functionality and design new application areas.

The following table shows the most common table types and their examples.

Type	Remarks	Examples
Master	Table that contains information about the primary focus subject of a application area.	Customer, Vendor, Item
Supplemental	Table that contains information about a supplemental subject that is used in one or more application areas.	Language, Currency
Setup	Table that contains one record that holds general information about a application area.	General Ledger Setup, Sales & Receivables Setup
Register	Table that acts as a table of contents for its corresponding ledger table or tables.	G/L Register, Item Register
Subsidiary	Table that contains additional information about a master table or a supplemental table.	Item Vendor, FA Depreciation Book
Ledger	Table that contains the posted transactional information that is the primary focus of its application area.	Cust. Ledger Entry, Item Ledger Entry
Journal	Primary table that allows you to enter transactions for a application area.	Purchase Journal, Item Journal

Type	Remarks	Examples
Document	Secondary table that enables you to enter transactions for one or multiple application areas at the same time. Document tables are always pairs of tables: one table for document headers, and one table for document lines.	Sales Header/Sales Line, Finance Charge Memo Header/Finance Charge Memo Line, Reminder Header/Reminder Line
Document History	Table that contains the transaction history for documents that were posted.	Sales Invoice Header/Sales Invoice Line, Issued Fin. Charge Memo Header/Issued Fin. Charge Memo Line, Issued Reminder Header/Issued Reminder Line

Master Tables

A master table contains information about the subject of its application area. For example, the **Customer** table is a master table. This is the subject of the sales, marketing, and receivables application areas. A master table is somewhat static. Users regularly enter new master records, but rarely change existing master records.

All transactional tables in an application area are related to a master table. The master table itself is related to many other (usually supplemental) tables. There is at least one ledger table that is related to a master table. Master tables frequently contain many FlowFilters and FlowFields, most of which relate to its corresponding ledger tables. Most application areas have only one master table, although some master tables are shared between different application areas, and some application areas occasionally have more master tables.

Naming Master Tables

The name of a master table relates to the names of the records in the table. For example, the **Customer** table is named **Customer** because each record within it contains information about a customer.

Primary Key and Other Standard Fields

The primary key of a master table is named **No.**, is of type Code, and of length 20. The value of this field is assigned automatically through the number series functionality.

Note: *The **G/L Account** table is one important exception to this principle. It is the only master table in Microsoft Dynamics NAV 2013 where **No.** is not controlled by number series functionality.*

The description field of a master table is named **Name** or **Description**, is of type Text, and is 50 characters long. This field, together with the **No.** field, is always included in the DataCaptionFields property of the table, so that these fields are displayed in the title bar of the table pages.

Many master tables contain a field named **Blocked**. This is typically of type Boolean. This field indicates whether users can use a master record in transactions. Instead of deleting a master record that is no longer used, users can mark it as **Blocked**. This makes sure that an attempt by any user or system action to use that master record fails. Sometimes this field is of type Option. This prevents the use of the master record in some specific transactions, but allows for use in others. For example, in **Customer** and **Vendor** tables, this field is of type option, and allows for several levels of blocking a customer or a vendor.

Associated Pages

There are always at least three pages that are associated with a master table. They are as follows:

- Card page
- List page
- Statistics page

The Card Page

Use the *card page* to view and edit single records in the master table. The name of the page is the name of the table followed by the word *card*. Therefore, the card page for the **Customer** table is named **Customer Card**.

The first group in the **Navigate** tab on the card page is always named as the master table. This group includes actions that call pages for related or subsidiary information about the master table. These actions can be the following:

- The action for the related ledger entries that you can also call by pressing CTRL+F7 on the keyboard.
- The action for the related statistics page that you can also call by pressing F7.

The List Page

Use the *list page* to view multiple records in the master table. Unlike the card page, you cannot use the list page to edit the master table.

The name of the page is the name of the table followed by the word *list*. Therefore, the list page for the **Customer** table is named **Customer List**. This page is set as the LookupPageID property and the DrillDownPageID property of the master table.

Similar to the card page, the list page also includes actions to show ledger entries and statistics with the same keyboard shortcuts. The list page has its CardPageID property set to the page ID of the corresponding card page. This makes sure that a corresponding card is always opened when users click **View**, **Edit**, or **New** actions, or double-click a row in a list.

The Statistics Page

Use the *statistics page* to view calculated information about the record in the master table. This information is separated from the card page for performance reasons because this information is calculated from a potentially large number of records in the database. This information might slow down data access if it is always displayed on the card page.

The name of this page is the name of the table followed by the word *statistics*. Therefore the statistics page for the **Customer** table is named **Customer Statistics**.

Supplemental Tables

A *supplemental table* contains information about a supplemental subject that is used in one or more application areas. For example, the **Currency** table is a supplemental table that contains information about currencies. This table is not the primary focus of any application area, however, it is important.

Many supplemental tables contain certain sets of defaults that are applied automatically to other types of records, such as master records, when a record from the supplemental table is used. For example, the **Item Category** table contains the following five default fields:

- **Def. Gen. Prod. Posting Group**
- **Def. Inventory Posting Group**
- **Def. Tax Group Code**
- **Def. Costing Method**
- **Def. VAT Prod. Posting Group**

Values from these fields are copied to the relevant fields in the **Item** table, when a user selects a value in the **Item Category Code** field for an item. Generally, supplemental tables are not related to other tables, although many other tables are related to supplemental tables.

Naming Supplemental Tables

The name of a supplemental table is the name of one of the records in the table. For example, the **Currency** table is named **Currency** because each record within it contains information about a currency.

Primary Key and Other Standard Fields

The primary key of a supplemental table is named **Code**, is of type Code and of length 10. The description field of this table is typically named **Description**, is of type Text, and is typically of length 50, even though it may sometimes have a different length. Some supplemental tables do not contain a description field; some supplemental tables contain a description field named **Name**.

Associated Pages

The page that is used for a supplemental table is a list page. The associated page has no or very few actions.

The name of the page is the plural of the name of the supplemental table. Therefore, the page that you use to edit the **Currency** table is named **Currencies**. This page is set as the LookupPageID property of the table.

Subsidiary Tables

A subsidiary table contains additional information about a master table or a supplemental table. For example, the **Item Vendor** table is a subsidiary table that contains additional information (vendor numbers) for the **Item** table in the inventory application area.

Naming Subsidiary Tables

The name of this table generally consists of the names of the table or tables for which it is a subsidiary or a close approximation. For example, the table that is subsidiary to both the **Vendor** table and the **Item** table is named **Item Vendor**. Usually, it is a singular name that describes one record that is contained within the table.

Primary Key and Other Standard Fields

The primary key for the **Subsidiary** table contains a field for each table for which it is a subsidiary, each of which is related to that table. For example, the primary key field for the **Item Vendor** table consists of the **Item No.** field (related to the **Item** master table), and the **Vendor No.** field (related to the **Vendor** master table).

The primary key also can contain an Integer as the last field (named **Line No.**) to differentiate multiple records with the same subsidiary relationship. For example, the **Employee Qualification** table has an Integer field in the primary key to differentiate multiple qualification records for the same employee.

Subsidiary tables are generally not related to other tables except for the master tables. Other tables are generally not related to a subsidiary table because each subsidiary table has multiple fields in the primary key. Subsidiary tables usually do not have description fields.

Associated Pages

A subsidiary table uses one page for editing and viewing purposes. This page is usually called from the master or supplemental page to which it is subsidiary. The name of the page is usually the plural of the name of the table, such as **Employee Qualifications** or something that is related to the information in the subsidiary table, such as **Item Vendor Catalog**.

The page that you use for a subsidiary table is either a worksheet page or a list page. The following guidelines help you select the correct type:

- If the primary key for the subsidiary table contains an Integer, the page is a worksheet page. It shows no primary key fields. The primary key fields (except for the Integer field) are included as filters so that they are set automatically when users enter information.
- If the primary key for the subsidiary table does not contain an Integer, the page is a list page. It does not contain the primary key field of the master table from which it is called. This primary key field is filtered so that it is set automatically. For example, if you call the **Vendor Item Catalog** page from **Item Card**, then the **Item No.** field is not displayed. If you call the **Vendor Item Catalog** page from the **Vendor Card** page, then the **Vendor No.** field is not displayed.

Ledger Tables

A ledger table contains the transactional information that is the primary focus of its application area. For example, the **Cust. Ledger Entry** table is a ledger table. It contains all transaction information that is the primary focus of the sales and receivables application area.

This table resembles a subsidiary table because it is related to the corresponding master table. However, it has different characteristics. It is related to many other tables, mostly supplemental tables. Register tables are related to ledger tables, but typically no other tables relate to a ledger table. Most application areas have at least one ledger table, but that depends on functionality (many application areas contain two or more ledger tables). Some ledger tables, such as **G/L Entry** or **Item Ledger Entry**, are shared between several application areas.

In order to maintain an audit trail for transactional information, ledger tables cannot be changed by users except for a few highly controlled exceptions. These exceptions exclude the ability to add or delete a record. Examples of such exceptions are **Cust. Ledger Entry** and **Vendor Ledger Entry** tables, which let users change certain fields, such as **On Hold**, or **Pmt. Disc. Tolerance Date**.

Naming Ledger Tables

The name of the ledger table is usually the name of the master table to which it is related, plus the words *ledger entry* describing one of the records in it (an entry). Because the name can be lengthy, the name is sometimes abbreviated. For example, the customer ledger entry table is actually named **Cust. Ledger Entry**. When there is more than one master table, the name is the application area followed by the words Ledger Entry, for example **G/L Entry**.

Primary Key and Other Standard Fields

The primary key of a ledger table is an Integer field named **Entry No.** This primary key is always generated automatically by the posting routine that controls this ledger table, and is always incremented by 1. There is always a field in the ledger table that has a table relationship with the master table that is associated with this ledger table. The description field of this table is a Text field of length 50, and is named **Description**.

In addition to the primary key, ledger tables generally have many secondary keys, many of which have SumIndexFields attached to them. These are used together with the FlowFields on the master table to calculate information for the user. Because of this, at least one of the secondary keys has a field that is related to the master table as the first field in the key.

Associated Pages

A list page is used to view the records in the Ledger table. The name of the page is the plural of the name of the ledger table. Therefore, the page that is used to display records from the **Cust. Ledger Entry** table is named **Customer Ledger Entries**.

This page is set as the LookupPageID property and the DrillDownPageID property of the table because they are used not only for viewing, but also for lookups and drill-downs into this table.

The list page can be displayed from the master table pages by pressing CTRL+F7.

Register Tables

A *register table* is a table of contents for its corresponding ledger table or tables. There is one record per posting process. The register table corresponds more closely to the posting routine instead of the application area. For example, the table that contains the list of entries that are made to the **Cust. Ledger Entry** table is the **G/L Register** table. This is because the customer ledger entries are posted from **General Journal** by using the general ledger posting procedures. The register table is related to its corresponding ledger table or tables.

Naming Register Tables

Register tables are named according to the posting function followed by the word *register*. Therefore, the register table that is updated by the general ledger posting function is named **G/L Register**. Users cannot change the register table.

Primary Key and Other Standard Fields

The primary key of a register table is an Integer field named **No.** The primary key is always automatically incremented by 1 by the posting routine that controls the register. Other standard fields for the register table include the following:

- **From Entry No.** and **To Entry No.** Integer fields that are related to the corresponding ledger table
- **Creation Date** field that specifies the date when that a transaction was posted
- **User ID** field that specifies which user has posted the transaction
- **Source Code** field that indicates the source of the transaction
- **Journal Batch Name** field that indicates the journal from which the transaction was posted

Register tables typically do not have description fields.

Associated Pages

A *list page* is used to view the records in the register table. The name of the page is the plural of the name of the register table. Therefore, the page that is used to display records from the **G/L Register** table is named **G/L Registers**.

The list page contains action links to other list pages that display the corresponding ledger entries.

Journal Tables

Journal tables enable transaction entry for a application area. All transactions, whether entered by a user directly or generated from another posting routine, pass through a journal table to eventually be posted to a ledger table.

Journal tables are related to many other tables including master tables, supplemental tables, subsidiary tables, and sometimes corresponding ledger tables.

Because of their use in transaction entries, journal tables have more trigger codes for validating data than most other table types.

Naming Journal Tables

The name of a journal table is the name of the transaction being posted, followed by the words *journal line*. For example, the table in which users enter transactions to the **Resource** application area is named **Resource Journal Line**. Each record in a journal table contains one line from the corresponding journal.

The journal table is usually related to two corresponding supplemental tables: the journal template table and the journal batch table. These tables let users split up data entry in various ways and let them set optional information that applies to the whole journal.

The names of these two tables are the same as the name of the journal table, except that they are followed by the words *journal batch* or *journal template* instead of *journal line*. Therefore, the two corresponding tables for the **Resource Journal Line** are named **Resource Journal Template** and **Resource Journal Batch**.

Primary Key and Other Standard Fields

The primary key of a journal table consists of three fields:

- **Journal Template Name** field that relates to the journal template table
- **Journal Batch Name** field that relates to the journal batch table
- **Line No.** Integer field

The **Description** field of this table is a Text field of length 50.

Associated Pages

Use a worksheet page to make entries to the journal table. The name of the page is the same as the journal table, except without the word *line*. Therefore, the worksheet page for the **Resource Journal Line** table is named **Resource Journal**. Sometimes, the page is named for the type of data that is entered. For example, one of the many worksheet pages that you use with the **Gen. Journal Line** table is named **Sales Journal**. None of the primary key fields are included on the page.

When the worksheet page is called, it is filtered by the **Journal Template Name** and **Journal Batch Name** fields. The AutoSplitKey property of the worksheet page automatically sets the **Line No.** field by incrementing the last Line No. by 10000, or by trying to "split" the **Line No.** fields of the record above and the record under the insertion of a new record.

The journal worksheet page always includes these actions:

- An action that looks up the card page for the master record that is used in the journal. This can also be called by pressing CTRL+F7.
- An action that shows all ledger entries for the master record that is used in the journal. This can also be called by pressing CTRL+F7.
- An action that posts the journal into the corresponding ledger or ledgers, that is named Post, and can also be called by pressing F9 on the keyboard.

The journal worksheet page usually includes other actions that let the user perform various processing functions.

Document Tables

Document tables are secondary transactional tables that enable entries for one or multiple application areas at the same time. They are secondary only in that their information is posted to ledgers through journal tables, and not directly.

For most users, document tables are the primary means of entering a transaction. Because they are used for transaction entries, document tables have more trigger codes than most other table types.

There are two kinds of document tables:

- Document header tables
- Document line tables

Document Header Table

A *document header table* holds the main transaction information that applies to all lines in the document. For example, for a sales transaction, the **Sales Header** table contains main information about order or invoices, such as the customer to whom the order or invoice belongs, posting dates, shipping information, and similar information that apply to all lines in the document.

Document Line Table

A *document line table* holds detailed information for the transaction. For example, for a sales transaction, the **Sales Line** table contains information about each line of the order or invoice. A document line table is a subsidiary table of the document header table.

Like journal tables, document tables are related to many other tables. This includes master, supplemental, and subsidiary tables. Other table types are rarely related to other document tables.

Naming Document Tables

The name of a document header table is the name of the transaction or document, plus the word *header*. For example, the document header table that contains sales transactions is named **Sales Header**. Each record contains one sale, such as order or invoice. For example, a document header table that contains finance charge memo transactions is named **Finance Charge Memo Header**. Each record contains one finance charge memo.

The name of a document line table is the name of the transaction or document, plus the word *line*. For example, the document line table that contains sales transactions is named **Sales Line**. Each record contains one line from a sale, such as order or invoice. For example, the document line table that contains finance charge memo transactions is named **Finance Charge Memo Line**. Each record contains one line from a finance charge memo.

Primary Key and Other Standard Fields

For most document header tables, the primary key is a Code field of length 20 named **No.** that contains the document number.

Some document header tables contain multiple kinds of documents. For example, the **Sales Header** table contains invoice documents, credit memo documents, sales order documents, and other document types. In these cases, the primary key has two fields: an Option field named **Document Type,** and a Code field of length 20 that is named **No.**

For most document line tables, the primary key has two fields: a Code field of length 20 that contains the document number, and an Integer field named **Line No.** The Code field relates to the document header table. It is typically named according to that table's name (without the word *header*), followed by that table's primary key field. For example, the Code field in the primary key of the **Finance Charge Memo Line** table is named **Finance Charge Memo No.**

When the document header table primary key includes a document type, the primary key of the document line table has the following three fields:

- **Document Type** Option field
- **Document No.** Code field of length 20
- **Line No.** Integer field

The Code field relates to the document header table.

Associated Pages

A document header table uses page of type Document to display one header record at a time to view and edit the information in the header table. The page name is also the name of the document that is displayed. For example, the page that shows finance charge memos from the **Finance Charge Memo Header** table is named **Finance Charge Memo**. This applies even if the table contains multiple types of documents because, in this case, the page is set up to only view information from one type. For example, the page that shows sales invoices from the **Sales Header** table is named **Sales Invoice**.

The page contains FastTabs to split the fields into logical groups. This makes it easier for the user to edit the information. A document page includes a page part that shows the document lines page.

A document line table uses a ListPart page to display multiple line records at a time. The name of the page is the name of the document followed by either words *lines* or *subform*. None of the primary key fields are included on this page. This page is filtered by using the SubPageLink property for all the primary key fields except for the **Line No.** field that is handled automatically by the AutoSplitKey property of the page.

The document header table also uses a list page to let users view multiple documents at the same time. The name of this page is the name of the document page followed by the word *list*. For example, the list page that shows the sales invoices is called **Sales Invoice List**.

The document header list page uses the CardPageID property to specify the document page that is shown when the user views, edits, or creates a new document, or double-clicks a document in the list.

Document History Tables

Document history tables have the same relationship to document tables that ledger tables have to journal tables. When a document is posted, part of that posting process is copying the document tables to their corresponding document history tables.

To aid with that copying process, the document history table has fields that have the same field numbers, names, and properties as the original document tables.

Because document history tables record posted transactions, they generally cannot be edited by the user. However, they can be deleted under certain circumstances. For example, you can delete a posted sales invoice if it was printed. Other than these few distinctions, document history tables and pages are the same as document tables and pages.

Setup Tables

Each application area has its own *setup* table. These setup tables hold only one record that contains fields to select options for the application area, or to hold data that applies to the whole company. No tables are related to setup tables, although setup tables are frequently related to other tables, usually supplemental tables.

Naming Setup Tables

The name of a setup table is usually the name of the application area it configures, followed by the word *setup*. For example, the table that contains setup information for the general ledger application area is named **General Ledger Setup**. One exception to this rule is the **Company Information** table.

Primary Key and Other Standard Fields

The primary key for this table is a Code field of length 10 named **Primary Key**. It is always left blank as only one record for each table is permitted. Setup tables do not have a description field.

Associated Setup Page

There is only one page that is used for setup tables, and it is of type Card. The page has the same name as the table. The primary key field is not included in this page. The page allows for changing information, but does not allow for inserting or deleting records. If there is not a record in the underlying table on this page, the code in the OnOpenPage trigger inserts a record.

 ***Note:*** *Not every table that contains the word* setup *in its name is a setup table. There are some tables that contain the word* setup *in their name which have more than one record. These tables generally follow the rules of the subsidiary tables that were described in the "Subsidiary Tables" section, and do not follow the rules that are outlined in this section.*

Standard Data Model

Every application area in Microsoft Dynamics NAV 2013 follows the same principles and has a similar data model. Master, subsidiary, document tables, journal, ledger, and other tables all have the same role. There are certain patterns that are applied consistently across all application areas.

Depending on their types, there are certain code patterns that you can follow in all tables of the same type.

The consistency of data model and data flow patterns is important for both users and developers. When users master one application area and understand data model principles, they can also quickly understand other application areas. As a developer, when you understand the principles of data models and patterns, you can customize the standard application. You can also build new application areas and maintain a consistent experience in the standard application. This makes sure that users are as productive as possible.

Data Model Diagram

In every application area, there are three groups of tables.

Group	Remarks
Configuration tables	Static or slowly changing table where users enter information one time, and then rarely, if ever, change it. The application uses these tables during creation, modification, or deletion of records in other tables, such as transaction tables. These tables are frequently checked by various processes, such as posting. Changing information in these tables changes the way that data is processed, or alter other aspects of the functionality of an application area.
Operational transaction tables	Primary work table for users. Users enter information in these tables regularly. Adding, changing, or deleting information in these tables typically does not affect the application, or the business itself.
Posted transaction tables	Table whose information is generated automatically by the application during posting and similar processes. Users cannot create new records in these tables, and cannot change or delete records. There are few examples in which users can change or delete information. All the examples have clear business logic justification.

The following "Data Model of a Functional Area" diagram explains the relationships of various tables in a typical application area.

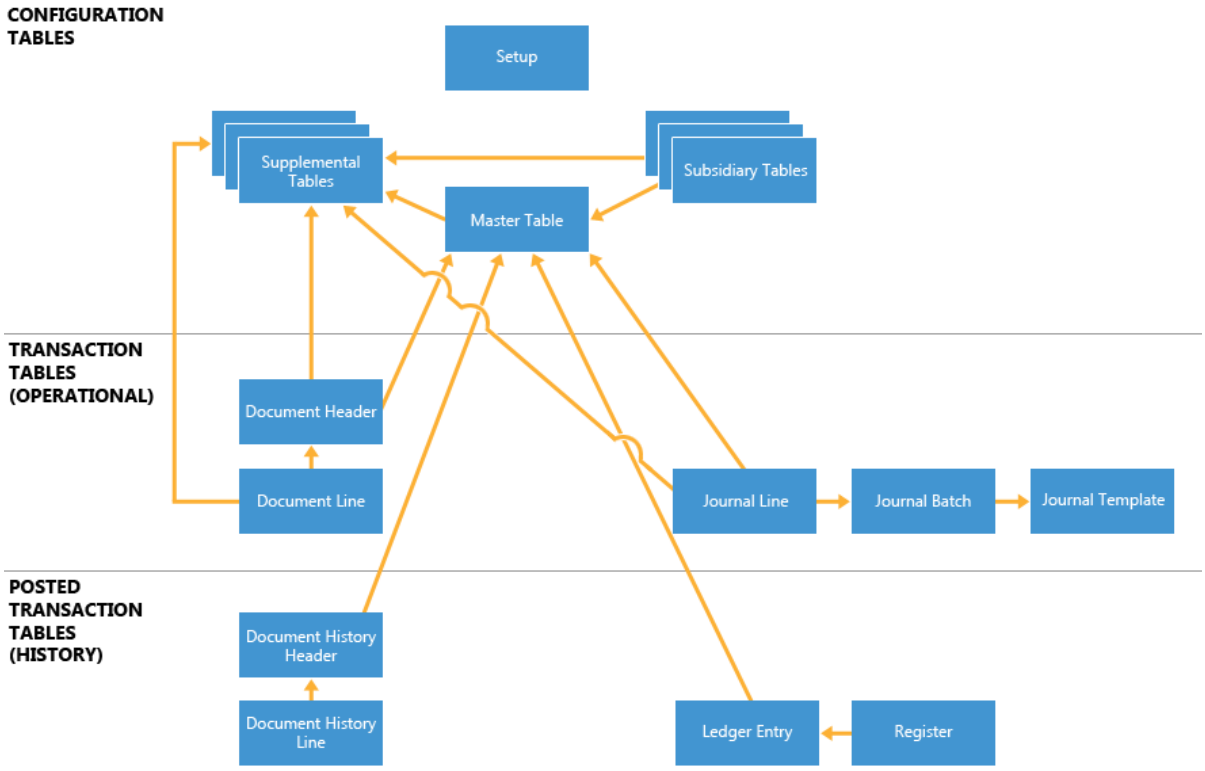

FIGURE 1.1: DATA MODEL OF AN APPLICATION AREA

Typical Data Triggers

In addition to maintaining data validity through data types and table relationships, Microsoft Dynamics NAV 2013 also contains data-related business logic. This makes sure that more complex data-related business rules are consistently applied as users insert, change, or delete information in the database.

These complex business rules are coded in table triggers as C/AL code. You can find the same patterns of code in the same types of tables, regardless of the application area where they belong.

You must understand these principles, and make sure that the code that you write in the existing objects does not violate those principles. You must also apply the same principles when you create completely new application areas.

OnInsert Trigger

This table trigger executes when a user inserts a new record into a table. The code in the trigger executes before the record is actually inserted into the table. If the code in the trigger causes a run-time error to occur, then the insert operation is canceled.

The OnInsert trigger has the following purposes in different table types.

Table Type	Purpose
Master	Assigns the **No.** field from the appropriate number series, if the user has not provided the value manually. Assigns the default dimensions for the account type, based on the configuration in the **Default Dimension** table.
Subsidiary	Checks whether all the necessary primary key fields are entered for tables with complex primary keys with three or more fields, or if any mutually exclusive or unacceptable primary key combinations are selected.
Journal	Validates the values in Shortcut Dimension 1 and Shortcut Dimension 2 fields.
Document (header)	Assigns the **No.** field from the appropriate number series if the user has not provided the value manually. Applies certain defaults, such as dates. Checks the filter on the field that is related to the primary master record of the application area. Assigns the value of the filter to the field if the field is filtered to a single value.
Document (line)	Makes sure that the status can accept new lines if the document supports different statuses.

For these table types, the OnInsert trigger may contain more code. For other table types, the OnInsert trigger may contain code that is specific to the applicable scenario, and no general principles are applied.

OnModify Trigger

This table trigger executes when a user changes an existing record in a table. The code in the trigger executes before the record is actually updated in the table. The application cancels the modifications if an error occurs in the trigger code.

The OnModify trigger has the following purposes in different table types:

Table Type	Purpose
Master	Sets the **Last Date Modified** field to the current system date. ***Note:*** *For master tables, the OnRename trigger must do the same action.*
Journal	Makes sure that a modification does not violate business rules for the specific journal type.
Document (header)	Assigns the **No.** field from the appropriate number series, if the user has not provided the value manually. Applies certain defaults, such as dates. Checks the filter on the field that is related to the primary master record of the application area. Assigns the value of the filter to the field if the field is filtered to a single value.
Document (line)	Makes sure that the modification does not violate business rules for the specific document.

For other table types, the OnModify trigger may contain code that is specific to the applicable scenario, and then no general principles are applied.

OnDelete Trigger

This table trigger executes when a user deletes a record from a table. The code in the trigger executes before the record is physically deleted from the table. The record is not deleted if an error occurs in the trigger code.

The OnDelete trigger has the following purposes in different table types:

Table Type	Purpose
Master	Makes sure that there are no started, but uncompleted transactions (such as orders, jobs, and so on) that are related to the master record. This could leave the system in an inconsistent state or cause issues for transaction processing. Deletes all subsidiary information for the master record. This includes default dimensions and any open documents and transactions.
Supplemental	Deletes all subsidiary information for the supplemental record.
Journal	Makes sure that the deletion does not violate business rules for the specific type of journal.
Document (header)	Makes sure that the deletion does not violate any business rules for the specific document, and deletes all document lines and subsidiary document information.
Document (line)	Makes sure that the deletion does not violate any business rules for the specific document, and deletes all subsidiary document line information.
Document History (header)	Makes sure that the conditions under which a posted document can be deleted are met, and then deletes the posted document lines, and any subsidiary posted document information.
Document History (line)	Deletes any subsidiary posted document line information. It does not check whether the conditions for the deletion are met, because users can never directly delete a document history line.

Depending on the scenario, the OnDelete trigger may contain more code and achieve more goals than specified earlier.

OnValidate Trigger

This field trigger executes after the user enters a value in a field. The code in this trigger executes after the application executes a default validation behavior, such as data type validation.

This trigger is frequently defined on the fields which relate to other tables, such as master tables, subsidiary tables, and supplemental tables. When it is defined on such fields, it frequently performs the following important operations:

- Assigns the default dimensions, if applicable.
- Assigns certain default values to other fields.
- Validates any case-specific complex business rules.

For example, for the **Resource No.** field in the **Res. Journal Line** table, the OnValidate trigger performs the following tasks:

- Assigns the default dimensions from the selected resource to the resource journal line.
- Makes sure that the selected resource is not blocked.
- Assigns several fields from the selected resource to the resource journal line, such as **Description**, **Direct Unit Cost**, **Resource Group No**., and **Gen. Prod. Posting Group**.
- Makes sure that the time sheet is specified for the resource which requires time sheets, if the resource line is not created automatically by the system.

Standard Data Flow

Users frequently enter data into one table. That data flows between various tables during different processes. For example, information that you enter into master or supplemental tables moves into document tables, journal tables, and then finishes in the ledger tables.

Master Table Data Flow

When users create master tables, most of information in the table is entered directly by the user, whereas some information may come from other tables. During master table creation, most of the default settings are assigned from supplemental tables. For example, when you select an **Item Category Code** for a record in the **Item** table, the application automatically assigns the default **Gen. Product Posting Group**, **VAT Prod. Posting Group**, **Inventory Posting Group**, and **Costing Method** to the item, as defined in the selected item category.

The following figure "Data Flow during Master Record Creation" shows the flow of the data between tables during creation of master records.

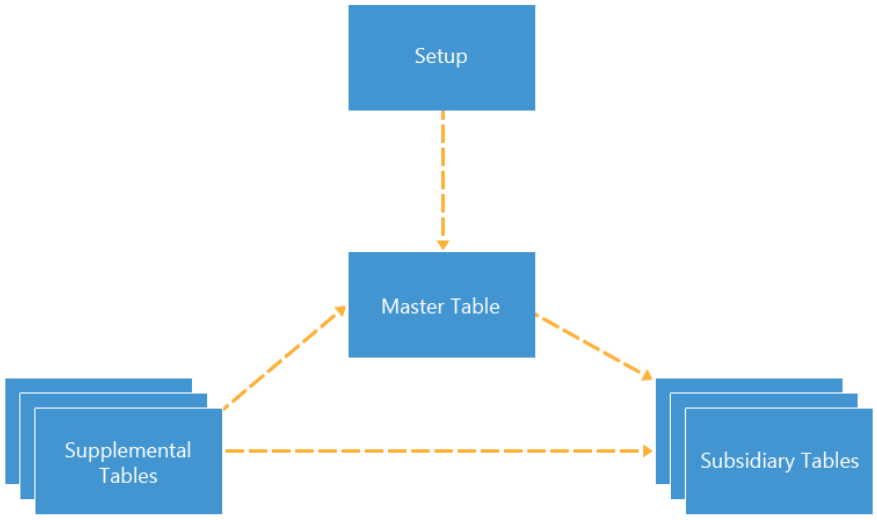

FIGURE 1.2: DATA FLOW DURING MASTER RECORD CREATION

Typically, master tables contain many fields that have relationships to other tables, such as supplemental and other master record tables. When you enter values in these fields, the application may take default values from related tables and assign them to the master record. In addition, when you enter subsidiary information for a master record, some master record fields may be taken into the subsidiary table.

During master record creation, certain defaults are checked in the setup table for the application area. At a minimum, this includes the number series, but may include many default checks or business rules validations.

Document Creation Data Flow

Documents are complex data structures that combine data from various types of tables. When users enter information into documents, most of the configuration tables for the application area are checked.

The following "Data Flow during Document Creation" figure shows data flow during document creation.

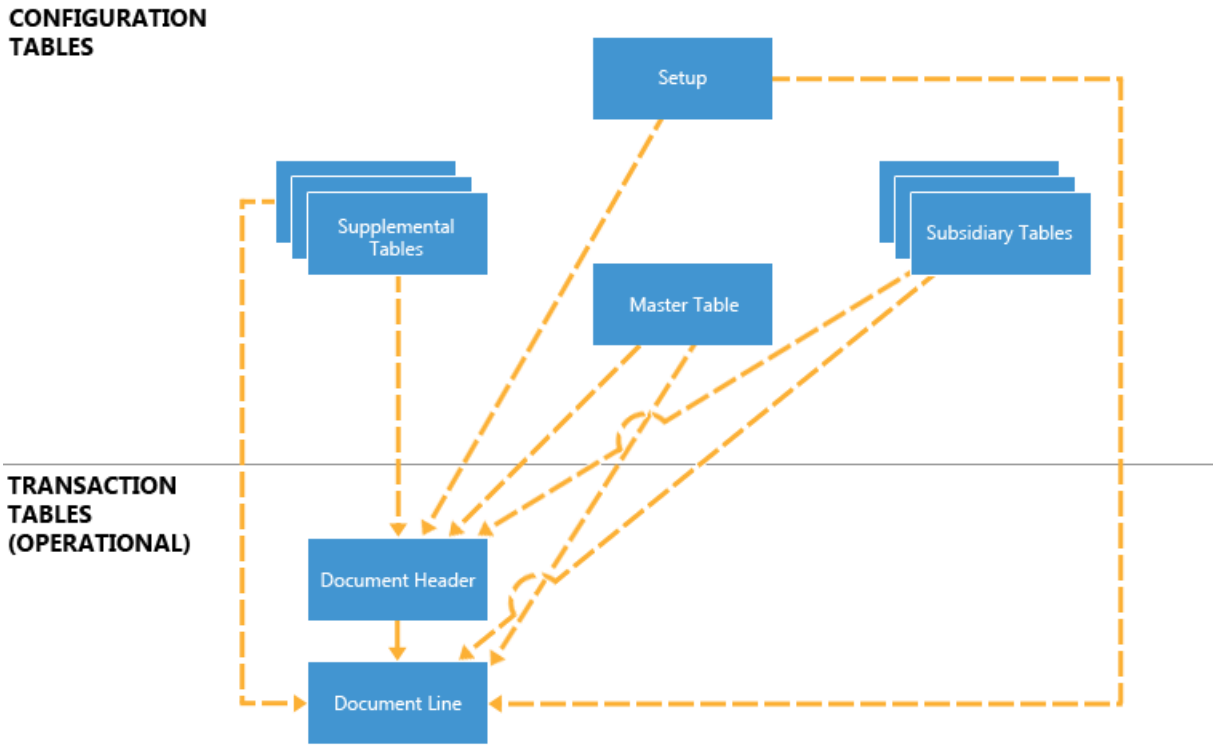

FIGURE 1.3: DATA FLOW DURING DOCUMENT CREATION

Journal Creation Data Flow

Users enter enter information into a journal line table in a journal. Although users enter most information directly, many fields are assigned automatically. In all journals, many fields, such as **Document No.** or **Reason Code,** are assigned from or based on the corresponding journal batch table. The journal batch table contains a series of other default values for the journal lines. For example, for the **General Journal Line** table, the **Bal. Account Type** and **Bal. Account No.** fields are assigned from the **Gen, Journal Batch** table.

In addition to defaults from the journal, and similar to the documents, many fields are assigned from the master, supplemental, and subsidiary tables that are associated with the journal transaction.

The following "Data Flow during Journal Creation" figure shows the data flow during journal creation.

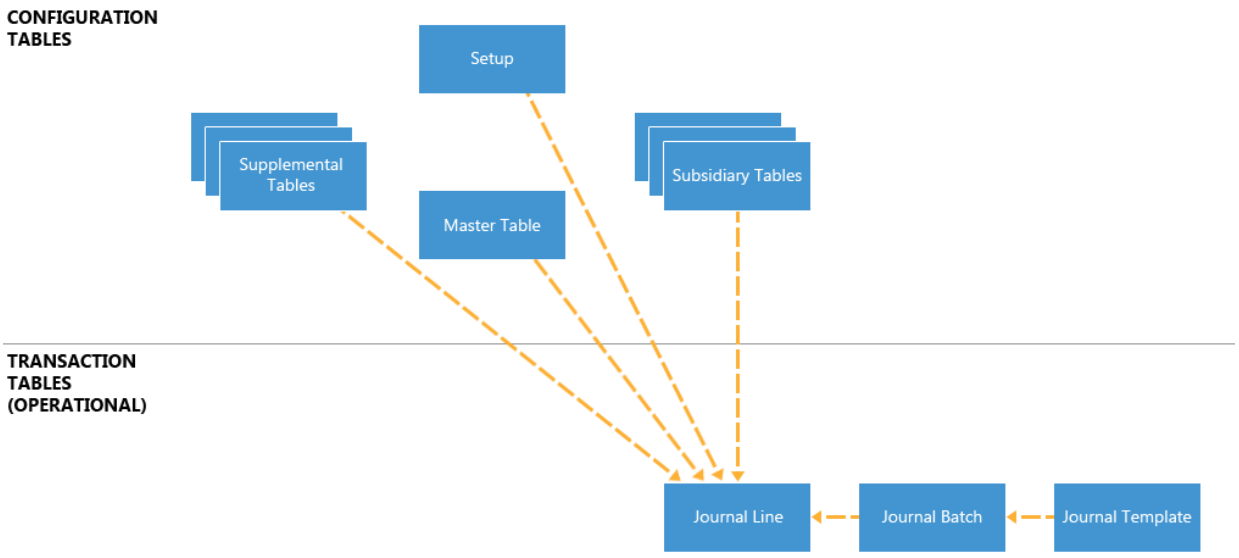

FIGURE 1.4: DATA FLOW DURING JOURNAL CREATION

Posting Data Flow

Posting is one of the most important processes in Microsoft Dynamics NAV 2013. It moves data from operational tables that are under user control (users can freely change them) to the posted transaction tables that are relevant from a business or financial perspective (users cannot freely change them).

There are two types of postings:

- Journal posting moves data from journal tables into a ledger entry tables.
- Document posting performs the following tasks:
  1. Moves data from document tables to document history tables.
  2. Moves data from document tables into journal tables.
  3. Invokes relevant journal posting routines.

Note: *Posting routines are very complex and involve much more processing than moving data between tables.*

The following "Data Flow during Posting" figure shows the data flow during document and journal posting processes.

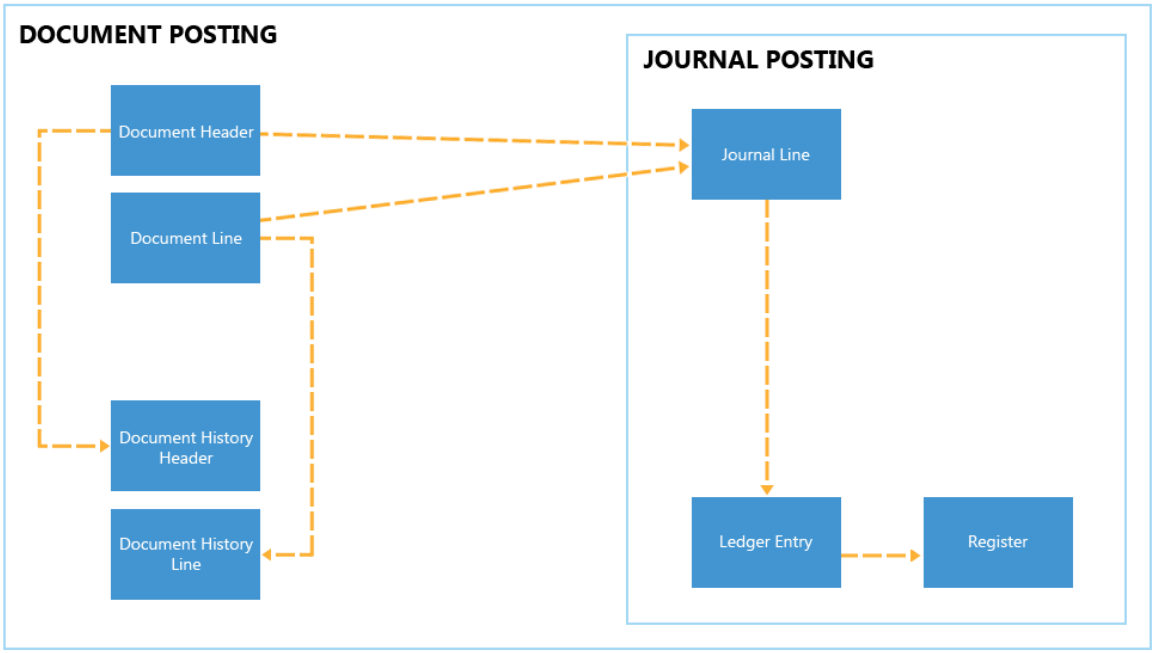

FIGURE 1.5: DATA FLOW DURING POSTING

Standard Process Model

Similar to consistent data structures, all application areas follow the same principle when it comes to processing. Because all application areas contain master, documents, journal, and ledger tables, there are also similar processes that guarantee consistent data flow among tables.

Posting is the most important process in Microsoft Dynamics NAV 2013. It accompanies almost every application area. Even though there are different posting routines for different application areas, all posting routines follow the same patterns and principles. Understanding those patterns and principles makes development simpler, and guarantees a consistent user experience.

Journal Posting

Journal posting is a process that creates ledger entries from journal lines in a application area. A single journal posting process may affect multiple application areas and result in different types of ledger entries. However, journal posting always reads a single set of journal tables. For example, a general journal posting routine always reads **Gen. Journal Line** and **Gen. Journal Batch** tables, but can create entries in the following tables:

- **G/L Entry**
- **Cust. Ledger Entry**
- **Vendor Ledger Entry**
- **Bank Account Ledger Entry**
- **FA Ledger Entry**

A journal posting routine consists of a group of codeunits, some of which are called directly by the user from a page. Other condeunits are called by other codeunits during posting processing.

The following codeunits are the core of the journal posting routine.

Codeunit	Remarks
Journal – Post Line	Reads information from a single journal line, and then writes corresponding ledger entry or entries.
Journal – Check Line	Reads information from a single journal line, and then checks it against various business rules, such as whether posting dates are valid, or if relevant fields contain values.
Journal – Post Batch	Reads information from all lines in a batch, and then calls the check line and post line for each line in the batch.

A user can never call any of these codeunits directly. They are always called by other journal or document posting codeunits.

Users can call the following codeunits by clicking a corresponding action in a journal page.

Codeunit	Available in page	Remarks
Journal – Post	Journal	Asks for a confirmation, and then calls Journal – Post Batch codeunit.
Journal – Post + Print	Journal	Asks for a confirmation, and then calls Journal – Post Batch codeunit, and then prints the corresponding register report.
Journal – Batch Post	Journal Batches	Asks for a confirmation, and then calls Journal – Post Batch for each selected batch.
Journal – Batch Post + Print	Journal Batches	Asks for a confirmation, and then calls Journal – Post Batch and then prints the corresponding register report for each selected batch.

The following "Journal Posting Process" figure shows the process and data flow of journal posting codeunits.

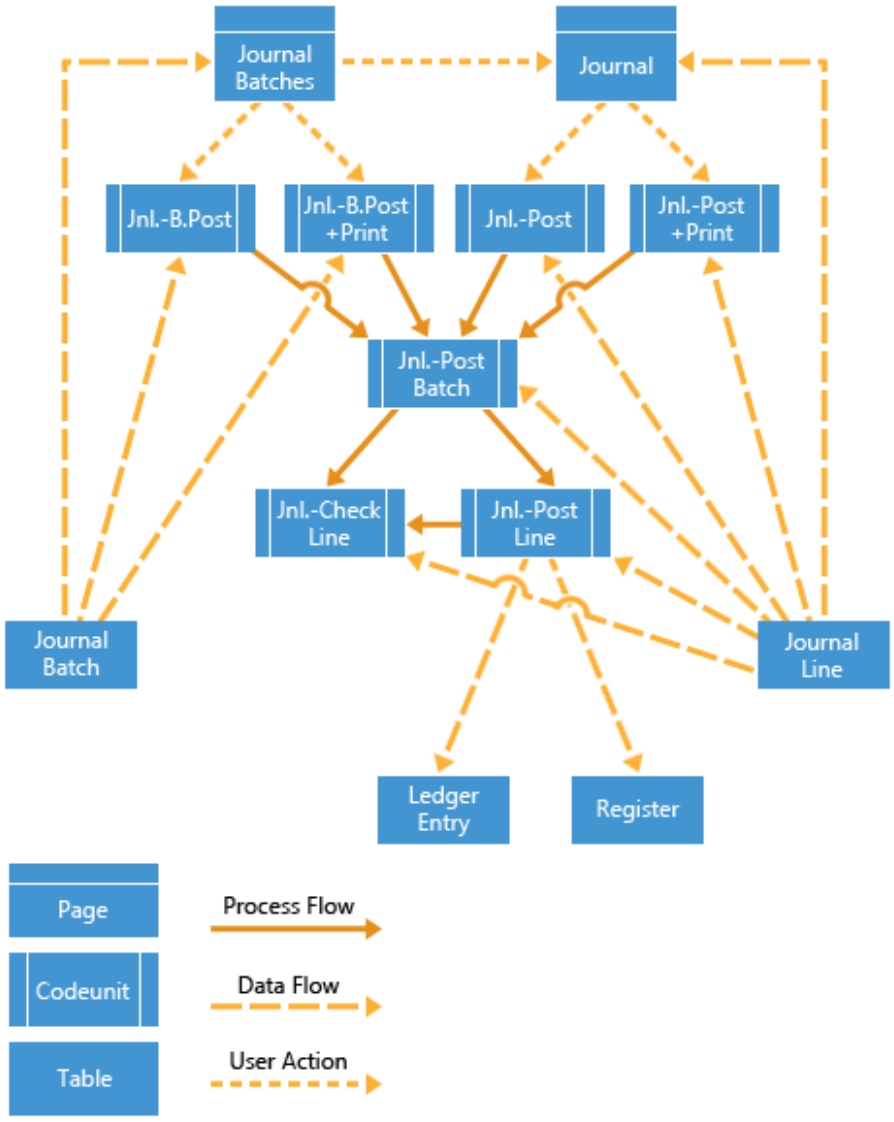

FIGURE 1.6: JOURNAL POSTING PROCESS

Document Posting

Documents resemble journals in the way that they also enable users to enter information before it is posted. Documents are more comprehensive and more intuitive than journals, because they frequently combine the functionality of several journals into a single, easy to use functionality.

Document posting is a process that creates posted documents from operational documents in an application area. Document posting also results in corresponding ledger entries and registers, frequently in multiple application areas. For example, a posting of a sales invoice, may result in ledger entries in the following tables:

- **G/L Entry**
- **Cust. Ledger Entry**
- **Item Ledger Entry**
- **Res. Ledger Entry**
- **FA. Ledger Entry**

A document posting routine consists of several codeunits.

Codeunit	Remarks
Post (Yes/No)	Asks for a confirmation, and then calls the Post codeunit. Users can call this codeunit from document and document list pages.
Post + Print	Asks for a confirmation, calls the Post codeunits, and then prints the posted document. Users can call this codeunit from document and document list pages.
Post	This codeunit is the central document posting codeunit. It copies the document into the posted document. It also analyzes the document and translates it into a series of journal lines from different journals. For each journal line calls the corresponding Journal – Check Line and Journal – Post Line codeunits. Users cannot call this codeunit directly.

The following "Document Posting Process" figure shows the process and data flow of a document posting routine.

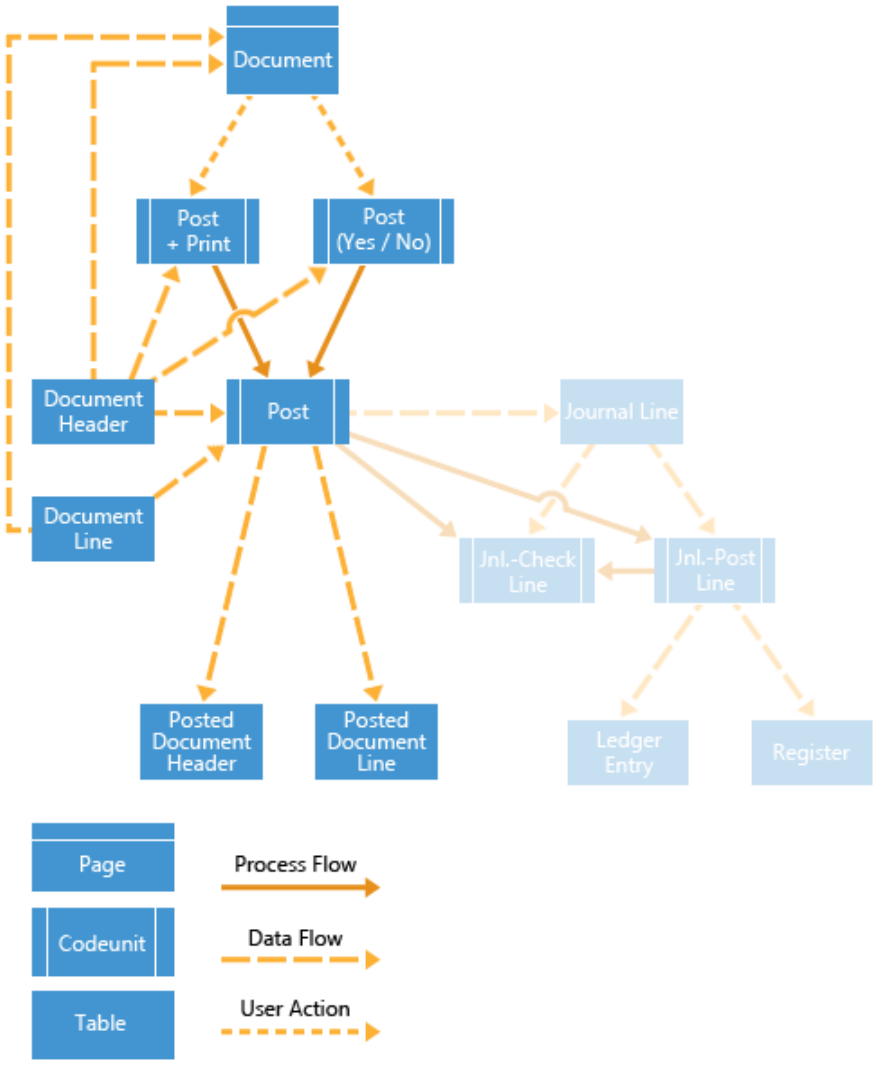

FIGURE 1.7: DOCUMENT POSTING PROCESS

Module Review

Module Review and Takeaways

Microsoft Dynamics NAV 2013 is a rich application that consists of many application areas, with data spread across a number of tables of different types with different purposes. Despite the apparent complexity of data and features, Microsoft Dynamics NAV 2013 applies basic principles to data and processes in all application areas. These principles guarantee that users have a simple, intuitive, and easy-to-learn experience across the application.

Even though Microsoft Dynamics NAV supports many business processes through rich standard functionality, companies frequently have different processes or even whole application areas, which are specific to their business or business vertical. Customizing existing application functionality or introducing new features or application areas into Microsoft Dynamics NAV 2013 is a common task for developers. Changes in such a complex application can easily result in bugs and inconsistencies which can degrade user experience and satisfaction.

Once you understand basic data principles, business logic patterns, and data and process flow in standard Microsoft Dynamics NAV 2013, you can apply them consistently to your own customizations. This guarantees that your changes blend seamlessly into standard functionality, and users have a consistent positive experience.

Test Your Knowledge

Test your knowledge with the following questions.

1. How do you call a table that holds relatively static information about subjects or objects of an application area?

    ( ) Master Table

    ( ) Subsidiary Table

    ( ) Register Table

    ( ) Setup Table

2. What name, data type, and length is the primary key field for a master table?

   ( ) No. of type Code and length 10.

   ( ) No. of type Code and length 20.

   ( ) No. of type Code and length 30.

   ( ) Name of type Text and length 50.

3. The **Item Unit of Measure** table is an example of what kind of table?

   ( ) Supplemental Table

   ( ) Subsidiary Table

   ( ) Ledger Table

   ( ) Register Table

   ( ) Setup Table

4. At minimum, what pages do you have to provide for each master table? How are these pages named? Are these pages editable?

5. What is the primary key of a ledger table?

6. Which trigger makes sure that number series functionality is applied during master record or document creation?

7. A document posting routine always calls one or more journal posting routines.

   ( ) True

   ( ) False

Test Your Knowledge Solutions

Module Review and Takeaways

1. How do you call a table that holds relatively static information about subjects or objects of an application area?

   (√) Master Table

   ( ) Subsidiary Table

   ( ) Register Table

   ( ) Setup Table

2. What name, data type, and length is the primary key field for a master table?

   ( ) No. of type Code and length 10.

   (√) No. of type Code and length 20.

   ( ) No. of type Code and length 30.

   ( ) Name of type Text and length 50.

3. The **Item Unit of Measure** table is an example of what kind of table?

   ( ) Supplemental Table

   (√) Subsidiary Table

   ( ) Ledger Table

   ( ) Register Table

   ( ) Setup Table

4. At minimum, what pages do you have to provide for each master table? How are these pages named? Are these pages editable?

   MODEL ANSWER:

   You must provide a list and a card page. The pages are named after the master record, and include words List and Card. For example, Customer List, and Customer Card. The list page must not be editable.

5. What is the primary key of a ledger table?

   MODEL ANSWER:

   It is an integer field named Entry No.

6. Which trigger makes sure that number series functionality is applied during master record or document creation?

   MODEL ANSWER:

   The OnInsert trigger.

7. A document posting routine always calls one or more journal posting routines.

   (√) True

   ( ) False