COMP1111

Week 3

Field Types

- Every table in Access is made up of fields and columns.
- The properties of a field describe the characteristics and behavior of data added to that field.
- A field’s data type is the most important property because it determines what kind of data that field can store.

Data Types

- Data types can seem confusing at first, for example, if a field’s data type is Text, it can store data that consists of either Text OR Numerical data.
- But a field that is of data type Number can only store Numerical data
- This is why it is important to know which properties each data type uses.

Field Properties

- A field’s data type determines many other important field qualities, such as:
  - Which formats can be used with the field
  - The maximum size of a field value
  - How the field can be used in expressions
  - Where the field can be indexed

Field Properties

- A field's data type can be predefined or you will select a data type depending on how you create the new field.
- For example, if you create a field from the Datasheet view and:
  - Use an existing field from another table, the data type is already defined in the template or in the other table
  - Enter data in a blank column (or field), Access assigns a data type to the field based on the values that you enter or you can assign the data type and format for the field.
  - On the **Modify Fields** tab, in the **Field & Columns** group, click **Add Fields**, Access displays a list of data types that you can select from

When to use which Data Type

- Think of a field’s data type as a set of qualities that applies to all the values that are contained in the field.
- For example, values that are stored in a Text field can contain only letters, numbers, and a limited set of punctuation characters, and a Text field can only contain a maximum of 255 characters.
- Sometimes, the data in a field may appear to be one data type, but is actually another.
- For example, a field may seem to contain numeric values but may actually contain text values, such as room numbers.
- You can often use an expression to compare or convert values of different data types.

Data Types

- Textual Types
- Short Text (Formerly known as “Text”)
  - Alphanumeric data (names, titles, addresses, etc)
  - Allows up to 255 characters
- Long Text (Formerly knows as “Memo”)
  - Large amounts of alphanumeric data
    - Sentences and paragraphs
  - Allows up to about 1 gigabyte (GB), but controls to display a long text are limited to the first 64,000 characters.

Data Types

- Number
  - Allows numeric data only
  - Size allows for 1, 2, 4, 8, or 16 byte numbers.
- Currency
  - Monetary Data, stores with 4 decimal places of precision
  - Allows up to 8 byte values
- Auto Number
  - Unique value generated by Access for each new record
  - Allows up to a 4 byte value

Data Types

- Date / Time
  - This data type allows you to enter Dates and/or Times
  - Allows up to an 8 byte value
- Yes/No
  - Boolean (True/False) Data
  - Access stores the numeric value zero (0) for false, and -1 for true
  - Data size is 1 byte
- OLE Object
  - Pictures, graphs, or other ActiveX objects from another Windows-based application
  - Can store up to about 2 GBs

Data Types

- Hyperlink
  - A link address to a document or file on the internet, on an intranet, on a local area network (LAN), or on your local computer
  - Size – up to 8, 192 (each part of a Hyperlink data type can contain up to 2048 characters)
- Attachment
  - You can attach files such as pictures, documents, spreadsheets, or charts
  - Each attachment field can contain an unlimited number of attachments per record, up to the storage limit of the size of a database file.
  - Note, the Attachment data type isn't available in MDB file formats
  - Size allowed is up to about 2GBs

Data Types

- Calculated
  - You can create an expression that uses data from one or more fields.
  - You can designate different result data types from the expression.
  - The calculated data type isn’t available in .MDB file formats
  - Size is dependent on the data type of the result type property.
    - Short text data type result can have up to 243 characters.
    - Long text, Number, Yes/No, and Date/Time should match their perspective data types

Data Types

- Lookup Wizard
  - The lookup wizard entry in the Data Type column in Design view is not actually a data type.
  - When you choose this entry, a wizard starts to help you define either a simple or complex lookup field.
  - A simple lookup field uses the contents of another table or a value list to validate the contents of a single value per row.
  - A complex lookup field allows you to store multiple values of the same data type in each row.

Good Database Design

- Certain principles guide the database design process.
- The first principle is that duplicate information (also called redundant data) is bad, because it wastes space and increases the likelihood of errors and inconsistencies.
- The second principle is that the correctness and completeness of information is important.
- If your database contains incorrect information, any reports that pull information from the database will also contain incorrect information.
- As a result, any decisions you make that are based on those reports will then be misinformed.

Good Database Design

- A good database design is, therefore, one that:
  - Divides your information into subject-based tables to reduce redundant data.
  - Provides Access with the information it requires to join the information in the tables together as needed.
  - Helps support and ensure the accuracy and integrity of your information.
  - Accommodates your data processing and reporting needs.

Design Process

- The design process consists of the following steps:
    - **Determine the purpose of your database**
        - This helps prepare you for the remaining steps.
    - **Find and organize the information required**
        - Gather all of the types of information you might want to record in the database, such as product name and order number.
    - **Divide the information into tables**
        - Divide your information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
    - **Turn information items into columns**
        - Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.

Design Process

- **Specify primary keys**
    - Choose each table’s primary key. The primary key is a column that is used to uniquely identify each row. An example might be Product ID or Order ID.
- **Set up the table relationships**
    - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
- **Refine your design**
    - Analyze your design for errors. Create the tables and add a few records of sample data. See if you can get the results you want from your tables. Make adjustments to the design, as needed.
- **Apply the normalization rules**
    - Apply the data normalization rules to see if your tables are structured correctly. Make adjustments to the tables, as needed.

Table Design

- Use table level properties to increase efficiency
    - Access tables offer several properties that remain with the data. You set the property just once, at the table level, and bound objects inherit those settings:
        - **Format:** Determines how Access displays the data.
        - **Caption:** Access displays this text in a control's corresponding label box.
        - **Input Mask:** Forces users to enter data in a specific order or manner.
        - **Default Value:** Access automatically populates a control with this value for new records.
        - **Description:** Documents at the table level. Access displays this text in the status bar when the control has focus.
        - **Validation Rule:** An expression that sets conditions that a value must meet for Access to accept the value.
        - **Validation Text:** A text message that Access displays if an entry fails to meet the Validation Rule expression. Use this to provide clues for entering appropriate data.

Table Design

- Name Fields Appropriately
  - A field name should reflect a field's purpose and describe the data it stores.
  - The field names FirstName, LastName, and ZipCode are self-documenting and easy to manage.
  - There's no guessing.
  - You know exactly what type of data should be in that field.
  - There are, however, a few rules you must follow when choosing field names:
    - A field name can consist of up to 64 characters—but only letters, numbers, and spaces.
    - Don't use reserved words to name fields, or any object, for that matter. For a list of reserved words, search Help for "reserved words."
  - If you aren't bound by an in-house naming convention, consider creating one of your own.

Table Design

- Don’t use spaces in field names
  - Although you might be tempted to use spaces in your field names, don't.
  - Spaces can be difficult to work with, especially in SQL statements and VBA code.
  - If you use them, you must remember to enclose the reference in brackets, which is a nuisance.
  - In addition, if you upgrade the database to SQL Server or export the data to use in another application, those space characters most assuredly will cause problems.

Table Design

- Don’t use spaces in field names
  - If you want field headings and subsequent objects (see # 1) to display more natural text, use the field's Caption property.
  - For instance, if you name a field LastName, use the Caption property to display *Last Name*. Just remember that the Caption property is for display only.
  - When referencing the field, you must always use the field's actual name.
  - Caption properties come with their own set of behavioral problems. If you're going to use the Caption property, plan for the following pitfalls:

Table Design

- Don’t use spaces in field names
  - Access ignores the AS keyword (alias) in a SQL statement if the underlying field has a Caption setting.
  - A Caption property setting won't make it to the results of a Make Table query.
  - DAO and ADO field objects return a field's Caption property and not the underlying field's name.

Table Design

- Don’t waste data type effort
  - When storing numeric data, you might be tempted to assign a Number data type.
  - Before you do, consider the data's purpose.
  - For instance, you won't use street number or zip code values in mathematical equations.
  - When this is the case, store the data as Text.
  - You'll save a bit on memory, but more importantly, the data type is true to the data's purpose.
  - If you should need to use a text value as a true numeric value, use the Val() function in the form
    - =Val(field)
    - where *field* represents the Text field that's storing numeric values or a literal value.

Table Design

- Use the most appropriate field size
    - With today's powerful systems, assigning the most appropriate field size isn't as urgent as it once was.
    - However, as a matter of good practice, developers still limit field size.
    - It's your first step to validating data.
    - For instance, let's say you store state abbreviations in a Text field with Field Size setting of 2.
    - If the user enters ARK instead of AK, Access will reject the entry.
    - Of course, the field size property rejects only entries that are too big. It can't spot typos or other mistakes.
    - The field would still accept other invalid entries, such as "A" or "K6."

Table Design

- Use the most appropriate field size
  - This works with numeric fields too; it just isn't as obvious.
  - For instance, a Byte field accepts values 0 through 255, while the Long Integer accepts values from -2,147,483,648 to 2,147,483,647.
  - Choose the appropriate data type and field size property for each field.
  - Always choose the smallest data type and field size that will accommodate the largest possible value in that field.

Table Design

- Choose indexes carefully
    - Access uses an index to sort data in logical order, according to data type.
    - The right indexing can improve performance.
    - The wrong index will slow things down.
    - By far, the most common mistake is to set too many indexes.
    - Because Access updates the index each time you add or change a record, too many indexes will affect performance.
    - Fortunately, you can apply a few indexing guidelines that will help

Table Design

- Indexing Guidelines
  - Remember that a primary key automatically sorts—that's an internal action you can't control.
  - In most tables, it's the only index you really need.
  - It's called a unique index because every value in the field must be unique.
  - As a general rule, consider indexing a table's foreign key.
  - Avoid an additional index on a table that you'll be frequently updating, unless one of the above conditions applies.
  - If the table has a primary key, consider an index on a second field only when you're working with large amounts of data, you plan to search or sort by that field often, or the field contains mostly unique values.

Table Design

- In short, index any field you want to sort, search, or join to another table to speed up tasks.
- Just remember that each index increases the size of the database, and too many indexes will slow things down.
- If you're working with small to reasonable amounts of data, indexes—beyond the primary key—usually aren't necessary.
- One final word on indexes: Don't use an index to sort.
- That isn't their purpose and you might not end up with the results you need.

Table Design

- Beware of AutoNumber limitations
  - Use an AutoNumber field to generate a unique number for your records.
  - Many people use these fields as the table's primary key, which is fine.
  - This type of key is called a surrogate key. In later versions of Access, you can generate an incremental or random value.
  - Incremental values are fine for most tables. You probably won't use random values.
  - Remember that a table can have only one AutoNumber field.

Table Design

- Don’t forget table properties
  - Like fields, tables have properties that define the table's purpose.
  - Most are self-explanatory, and the defaults are usually adequate.
  - To access these properties, open the table in Design view and then choose Properties from the View menu.
  - Here are a couple you should know about:
    - **Order By:** Specifies a sorting order that Access applies when you open the table. Simply enter the name of the field by which you want to sort the records. If you don't use this setting, Access sorts by the primary key. If there's no primary key, Access displays records in data entry order. It won't matter much at the table level, since users don't view tables. However, like many field properties, bound objects inherit the property.
    - **Subdatasheet Name:** Determines whether subdatasheets display related records. Many people find subdatasheets annoying. If that's you, set this property to [None] to disable it.

Database Normalization

- You can apply the data normalization rules (sometimes just called normalization rules) as the next step in your design.
- You use these rules to see if your tables are structured correctly.
- The process of applying the rules to your database design is called normalizing the database, or just normalization.

Database Normalization

- Normalization is most useful after you have represented all of the information items and have arrived at a preliminary design.
- The idea is to help you ensure that you have divided your information items into the appropriate tables.
- What normalization cannot do is ensure that you have all the correct data items to begin with.
- You apply the rules in succession, at each step ensuring that your design arrives at one of what is known as the "normal forms.“
- Five normal forms are widely accepted — the first normal form through the fifth normal form. This article expands on the first three, because they are all that is required for the majority of database designs.

First Normal Form

- First normal form states that at every row and column intersection in the table there, exists a single value, and never a list of values.
- For example, you cannot have a field named Price in which you place more than one Price.
- If you think of each intersection of rows and columns as a cell, each cell can hold only one value.

Second Normal Form

- Second normal form requires that each non-key column be fully dependent on the entire primary key, not on just part of the key.
- This rule applies when you have a primary key that consists of more than one column.
- For example, suppose you have a table containing the following columns, where Order ID and Product ID form the primary key:
  - Order ID (primary key)
  - Product ID (primary key)
  - Product Name
- This design violates second normal form, because Product Name is dependent on Product ID, but not on Order ID, so it is not dependent on the entire primary key.
- You must remove Product Name from the table.
  - It belongs in a different table (Products).

Third Normal Form

- Third normal form requires that not only every non-key column be dependent on the entire primary key, but that non-key columns be independent of each other.
- Another way of saying this is that each non-key column must be dependent on the primary key and nothing but the primary key.
- For example, suppose you have a table containing the following columns:
  - ProductID (primary key)
  - Name
  - SRP
  - Discount
- Assume that Discount depends on the suggested retail price (SRP). This table violates third normal form because a non-key column, Discount, depends on another non-key column, SRP.
- Column independence means that you should be able to change any non-key column without affecting any other column. If you change a value in the SRP field, the Discount would change accordingly, thus violating that rule. In this case Discount should be moved to another table that is keyed on SRP.

Database Normalization

- See below link for explanation with pictures
- http://www.fabalou.com/Access/General/Normalization.asp