Compilations: Designing and Using Archaeological Databases

4

Bridging the gap between data and the information that data constitutes is perhaps the greatest challenge to the archaeological application of IT. . . . What we must avoid is unjustifiable 'black box' systems or we may find the discipline tarred with the familiar '. . . manipulation of ambiguous data by means of dubious methods to solve a problem that has not been defined'.

Cheetham and Haigh (1992: 13)

Archaeological research results in sometimes large "compilations," the products of collecting, compiling, cataloguing, or organizing archaeological data into what Gardin (1980) calls "arrangements." Compilations can be simple lists, indexed lists, computer databases, or online catalogues. Alternatively, they can be tables, graphs, "galleries" of pictures, or statistical summaries. One use for compilations is simply to record and summarize information, but we can also use them to organize that information in meaningful ways that help us understand the data. Compilations are increasingly at the forefront because of the demand for "big data" for certain research programs of great interest as well as the legal and ethical compulsions to preserve archaeological information in an accessible and useful form for future generations (Kintigh 2006). Compilations with a spatial focus are also extremely important in archaeology, which makes considerable use of Geographic Information Systems (GIS).

As noted in the previous chapter, the languages we use to describe data in any kind of compilation, or analyses and inferences based on the data, require an ontology—a system of concepts, categories, terms, or "things"—that allows us to describe and compare phenomena. In this chapter, you will see how we incorporate ontologies into a kind of language that is amenable to manipulation with computers.

The simplest kinds of compilations are just lists. Because they have no particular order, lists may not be the best way to manage or seek patterns in data unless we use software to help us search them. To speed up that process, it is often useful to have an index or keywords. Tables can also be useful to display data. Tables impose order and dimensions on the data and sometimes emphasize particular kinds of data over others. Graphs, meanwhile, simplify and make visual large quantities of data that we might otherwise have to present as almost unmanageable lists or tables of information. As they say, a picture is worth a thousand words. Further simplification is possible through statistical summaries, such as averages, medians and proportions. These replace lists of many numbers with only a few numbers that characterize trends, central tendencies, or typical distributions in the data.

Although compilations can be as simple as a collection of file cards, each with a picture of an artifact and some identifying labels and measurements, most modern archaeological compilations depend on digital information and computer databases. This section will introduce some principles that guide the effective use of these tools. For more general treatment of computers in archaeology, see Lock (2003) and Evans and Daly (2006).

4.1 Information Language

Useful compilations typically employ an information language that ensures consistent and efficient recording of observations. As discussed in Chap. 1, measurement in the broad sense consists of comparing an item with a standard scale and representing this symbolically. But natural language is usually too ambiguous, inconsistent or wordy to make these kinds of analyses consistently.

An information language is a system of representation. Although archaeologists have used various information languages for well over a century, the use of computers has led to more explicit attention to them. In the early days of computerization, archaeologists had to force themselves to

E. B. Banning, *The Archaeologist's Laboratory*, Interdisciplinary Contributions to Archaeology,
https://doi.org/10.1007/978-3-030-47992-3_4

describe sites or artifacts with codes of no more than 80 characters. Today, we do not have so many constraints, but efficient and useful analyses still require the consistency of an information language, even if it sometimes resembles English or some other natural language.

4.1.1 Graphical Information Languages

Archaeologists have almost always used graphical conventions to simplify and make consistent the description of artifacts and other data. Maps and technical drawings are standardized simplifications of reality because they omit details that their makers or users do not find relevant while depicting or emphasizing information that would not be apparent in an unedited photograph or scan (see Chap. 21).

Take lithic illustrations, for example. Archaeologists less often publish photographs of lithics than drawings that encode important attributes of each tool, core or flake (Fig. 21.9). They use strict conventions or rules about the orientation of each piece, and how to depict ventral and dorsal views, side-by-side. The usual convention is to use solid lines to depict the borders of flake scars and curved, tapering lines to simulate the rippling that occurs in knapped flint to signal the direction of flake removals. They use stippling to indicate cortex and other conventions to represent non-flint materials, polish, and the position and direction of burin spall removals. The best lithic illustrations may seem like beautiful works of art, but they are actually simplified and standardized representations that emphasize the attributes that lithic researchers consider important and ignore or suppress others.

Pottery drawings are another excellent example of graphical information language. Ceramic illustrations usually bear little literal resemblance to the fragments of vessels on which they are based. Illustrators typically use information from the curvature of a sherd to estimate the diameter of the whole vessel, and then reconstruct as much as they can of that vessel from the fragmentary evidence available. In addition, they simultaneously depict the interior and exterior of the vessel and a radial cross-section through the vessel walls, sometimes called a "profile" (Fig. 21.10). The illustrator uses conventions for indicating any surface treatments and decorations that appear on the sherd, and often offers a cross-section through a handle, should there be one. Some information languages also use small icons and labels to encode additional information directly on the illustration (e.g., Smith 1973; Fig. 4.1).

While at first glance artifact illustrations may simply seem like drawings, in fact they are coded representations that display more information than would a photograph or even an unedited 3D digital model.

4.1.2 Digital Information Languages

While graphical information languages are excellent at depicting and storing information, they do not currently offer an efficient and accurate way to sort or retrieve or "query" that information, especially in large databases, although there are recent steps in that direction. Ideally, we want retrieval of information to be as easy as storing it.

Digital data processing uses an information language to reduce complex objects to a conventional representation in a database or to amplify a simple query into alternative forms that may represent it in the database (Gardin 1980). The former makes it easier to store large quantities of information in a way that allows us to retrieve it later; the latter allows us to ask questions of the database in order to make that retrieval.

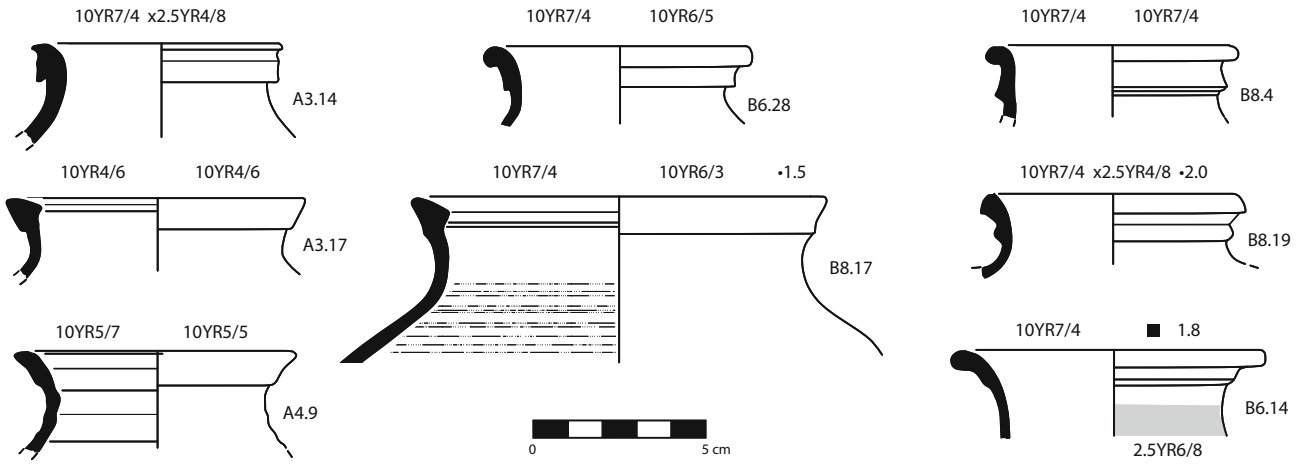

Fig. 4.1 An example of illustrations of pottery that employ a combination of graphical and textual information languages to display vessel shape and size, radial section, interior and exterior colors, surface treatments and artifact numbers. (cf. Smith 1973)

Information languages typically involve a specialized vocabulary, or **lexical units** (e.g., geometrical shapes, flake edges, decorative panels, brush strokes), **orientation rules** that define a standard position for the objects being described, **segmentation rules** for dividing an object into its constituent parts, and **differentiation rules** that specify the distinctions we record for each segment (Gardin 1980).

In terms of systematics, lexical units are the symbolic representations of attributes and categories, differentiation rules are related to the definition of categories and their separability, and rules for orientation and segmentation are other kinds of categorization that encourage consistency in the way we measure attributes. Because this involves rules, we are generally using classification, not grouping; the rules are definitions intended to make classes that are mutually exclusive and unambiguous, if possible. Although it is possible to make database queries that are more forgiving—for example, ones that would show you artifacts that are similar to the specific type you asked to see or that use fuzzy classification (Niccolucci et al. 2001)—most databases do not facilitate that, and even something as simple as a spelling error in a type name could lead to omission of important data from your query result.

We can describe pottery, for example, with an infinity of attributes (see Chap. 12 for some of these). For now, let us consider some candidates for basic lexical units, orientation rules, segmentation rules and differentiation rules for a pottery database. For segmentation rules, we might have to define very carefully what we mean by "rim," "neck," "shoulder," "body," "base," and "handle" (Fig. 4.2). Differentiation rules define how to assign each sherd to a segment, while other differentiation rules guide its classification as a particular kind of "rim" or "handle," etc. These rules also need to anticipate that some sherds will include all or portions of more than one segment, such as a rim with an attached handle, and your database will need a consistent and sensible way to accommodate these. The orientation rules for pottery usually require sherds to be "stanced" in the position it most likely had when part of a whole pot, standing vertically (usually with rim horizontal, and on top, base horizontal, at bottom). Body sherds and decoration often have less certain orientation, but the rules need to accommodate that uncertainty as well. The lexicon would usually include symbolic representations of categories for rim shape, decorative elements, mineral inclusions, and so on.

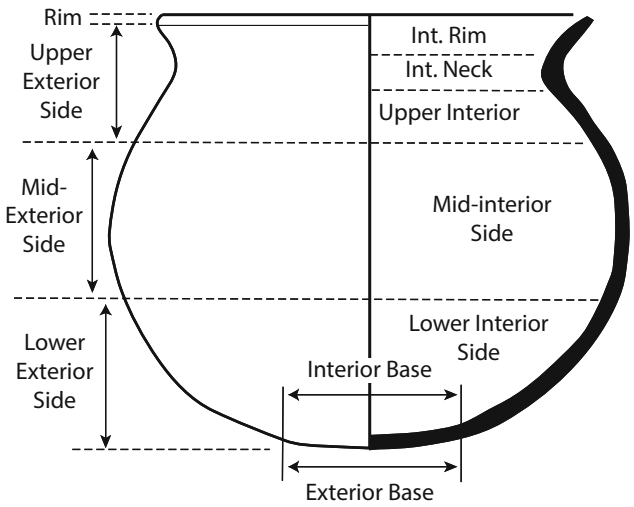

Fig. 4.2 An example of segmentation of a vessel into distinct parts (after Skibo 1992: 114). The definitions of these segments constitute the segmentation rules for an information language to describe pots like this

Ideally, the differentiation rules, like the definitions in a classification, allow unambiguous assignment of each sherd to a category. Often this involves a step-by-step, hierarchical procedure paralleling a taxonomic classification, but paradigmatic rules are also possible, and computers facilitate use of paradigms with many dimensions. Some kinds of data, such as rim shape, may be less amenable to unambiguous definition, and it is here that some archaeologists adopt a central-tendency form of grouping, with pictures of "ideal" rim shapes so that lab personnel can match each sherd to the shape to which it is most similar. On the other hand, modern databases may use mathematical characterizations of artifact shapes that allow stricter definition of shape classes (e.g., Gilboa et al. 2004, 2013).

Information languages for lithic materials typically employ such segments as "dorsal" and "ventral" sides and "proximal" and "distal" ends of flakes and have explicit orientation rules that define the axes of length and width and the position of retouch (see pp. 164, 168). The lexicon includes labels for categories of retouch, angles, raw material, platform shape and so on.

Decoration on pottery, basketry and other materials poses special problems. The orientation and segmentation rules often have to be complex to accommodate the distinctions that analysts find important and the way design elements are combined (e.g., Fig. 4.3). The alternative is a very lengthy lexicon that has separate categories for whole decorative schemes, rather than rules for the combinations of a smaller number of elements (e.g., Fig. 4.4).

Ancient Mesopotamian cylinder seals provide ambitious examples of archaeologists' attempts to describe artifacts with an information language. Digard et al. (1975) created a large lexicon for the symbols, figures, animals, buildings, furniture, and other things depicted on seals, and complex segmentation rules and a grammar or "syntax" that allows users to distinguish the different ways the lexical elements are combined and interact, including "actions," "number," and "configuration." For example, on a seal that shows a king, a

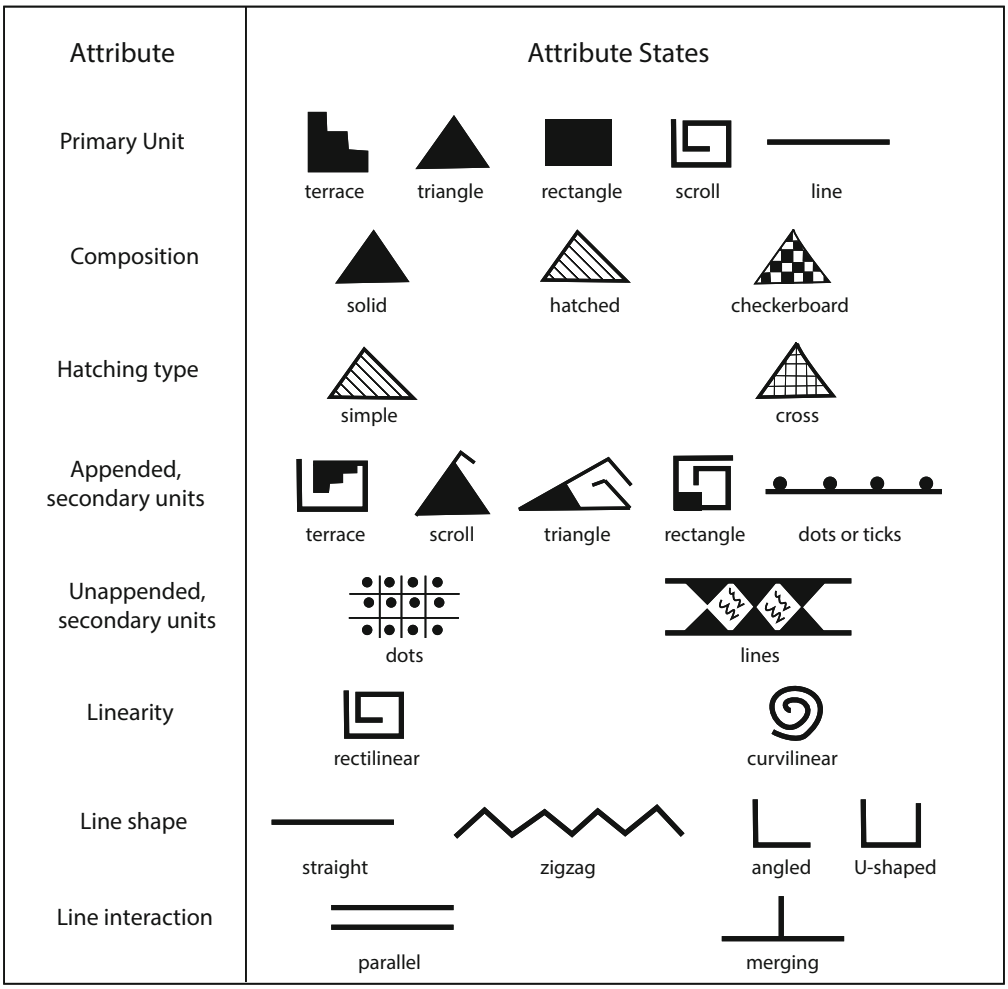

Fig. 4.3 An example of a portion of the differentiation rules for pottery decoration in the American Southwest. This one includes primary and secondary decorative elements that can be combined in various ways. (After Plog 1980: 51)

god, a throne, and a star, we can describe whether the king or god sits on the throne, whether the king is left or right of the god, is kneeling or standing, who is wearing which distinctive clothing, and the position of the star relative to other elements (Fig. 4.5).

Information language for describing architecture can focus on two-dimensional plans of structures or more thorough description. A database for the plans of rectilinear buildings, for example, needs orientation rules to identify things like long axis, short axis, front, sides and back of each building or room, as well as the azimuth (compass orientation) of the long axis. It would need lexical units to classify various rooms, features, post arrangements, stairs, or doorways (Fig. 4.6). It could also have a sort of grammar, like the one for cylinder seals, to describe the relationships between lexical elements, as Hillier and Hanson (1984) did for the "grammar" of built spaces. Other architectural lexical elements would likely include room dimensions, building area, and construction method.

4.2 Database Design

A database is a compilation of information that supplies users with data from which they make decisions and inferences, formulate typologies, explore patterns, or test hypotheses. It can be as simple as a telephone book or index card file, but today we are more likely to use the term to label an integrated body of digital information that we access through a computer or other digital device (even our phone). An integrated database has interrelated data stored with controlled redundancy, which means that only certain kinds of data are repeated, allowing us to retrieve other data that are unique. It can be the basis for complex data analysis, such as multivariate analysis of artifact similarities and differences, spatial analyses, or network analyses (Baxter 1994; Östborn and Gerding 2014).

Most databases share basic kinds of input and output. Users make inquiries, add or modify data (these are

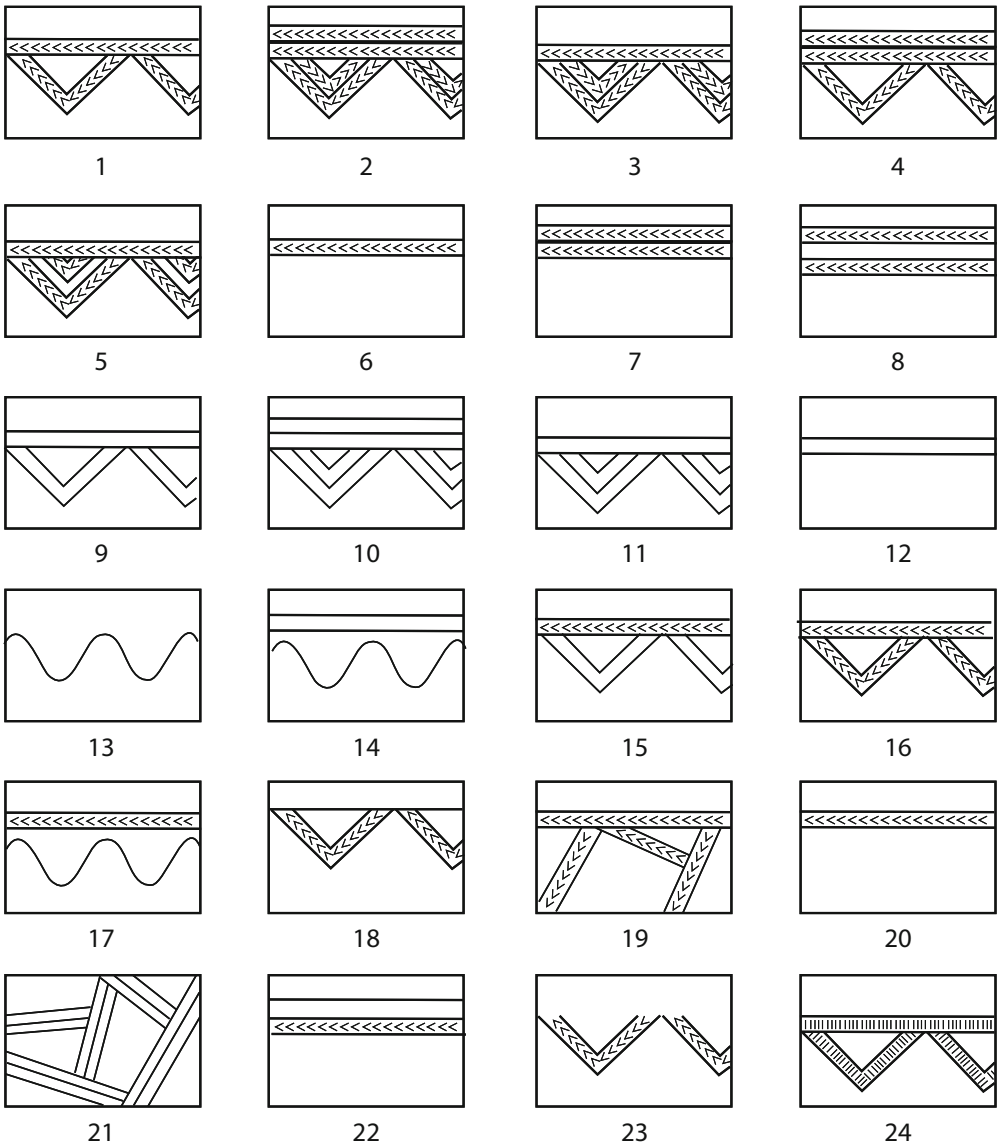

Fig. 4.4 An alternative lexicon for incised decoration on Yarmoukian Neolithic pottery that has a single level of design elements (after Garfinkel 1999: 65). Can you think of ways to redesign this as a sort of grammar with primary and secondary design elements and their interactions?

"transactions"), or they modify or add to the database itself. All these are kinds of input. For example, an archaeologist's input might consist of typing the description of a pit feature into one of the "fields" of a "record." Output consists of responses to inquiries, transaction logs (records of changes to the database), updated data, and an updated database. For example, the response to the query, "which pit features contained charred plant remains?" would trigger as output a list of the pit features that did.

Too often, archaeologists wanting to establish a database begin by sitting in front of a computer and defining some fields, without much forethought. This often results in a database that requires many revisions and corrections before it is even minimally acceptable for use, and this leads to frustration and sometimes much greater cost.

Just as with other aspects of research design (see Chap. 6), designing a database deserves better. It should begin with a list of objectives and expectations. Who will use the database, and for what purpose? Will its use be limited to a short time, or a single project, or do we expect it to serve for multiple projects for a long time? If the latter, who will maintain it and keep it up to date? What kinds of research questions will the database help to answer? Are the data types available in a commercial off-the-shelf database product sufficient for our purposes, or will we need to define specialized data types? Is there an existing database that is still useful and usable for our

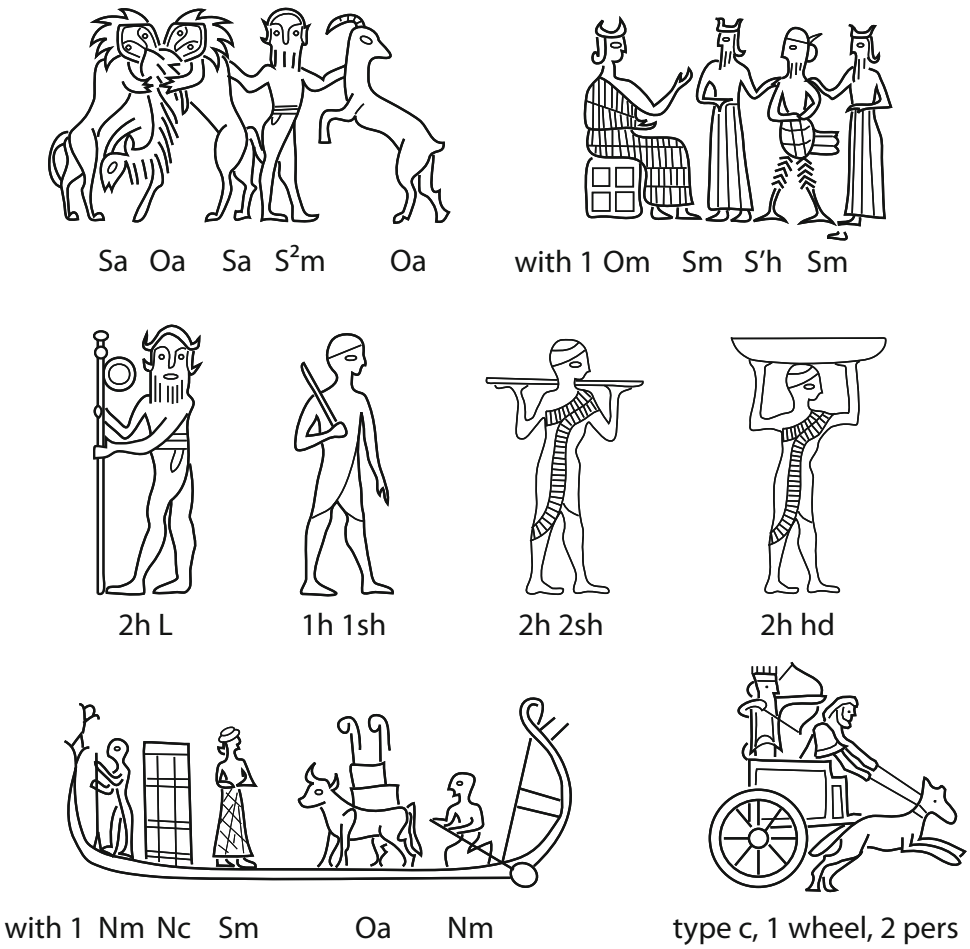

Fig. 4.5 A few examples of lexical items and interactions for describing cylinder seals. Top: interactions among people and animals. Middle: holding an object. Bottom: people, objects and animals and their interactions on vehicles. Lexical items include animal subject (Sa), animal object (Oa), subject man (Sm), object man (Om), hybrid creature being led (S^2h), neutral man (Nm), neutral container (Nc), and actions include holding two-handed to left (2 h L), one-handed on one shoulder (1 h 1sh), two-handed on both shoulders (2 h 2sh), and two-handed on head. (2 h hd; modified from Digard et al. 1975: 43, 45, 249, 325)

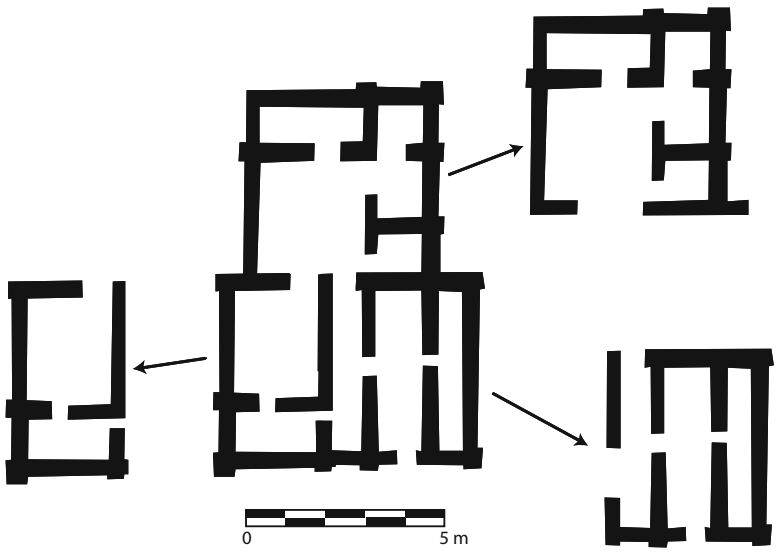

Fig. 4.6 A subset of segmentation rules for Samarran house plans in Mesopotamia, using the orientation rule that the widest part of the house is shown at the bottom. (After Banning 1997: 24)

purposes, or do we need to begin a new one from the ground up? If one already exists, what is the system like and what are its limitations?

Whether you are the principal or only user of the database you are designing or are collaborating with others, you should begin to create a logical design for the database before you even begin to think about its physical characteristics. Start by laying out how you expect the database to work and what kind of structure it should have to facilitate your anticipated uses rather than worrying too soon about what kind of processor, storage system, or software you should buy.

4.3 Data Models

To specify what you expect a proposed database to accomplish, it is useful to model it in the abstract. Start by listing the general kinds of entities (e.g., lithics, pottery, plant remains, users, sites, layers, photographs) that the database might need to document. An entity is any specific case of an entity type, so that a particular stone tool would be an entity potentially within the entity type, "lithics." Then think about what kinds of attributes each of those entity types should have—size, shape, location, material, and so on—and any relationships that may exist among entities. For example, sites may contain layers or excavation units, pottery may have been found within layers, and a photograph may "capture" a view of a feature in a site. Relationships can also have attributes, such as the date when a layer "was excavated." You might find it useful to think of entities as nouns, specific cases of an entity as proper nouns or I.D. numbers, and relationships as verbs (Chen 1976).

At the most general level, a conceptual data model outlines the main features and overall scope of your planned database. At a more practical level, you will want to have one or more logical data models that capture, in more detail, the entities already defined and any new entities you define to describe operations or transactions within the database, such as user logs or catalogs of attribute states (see Data Dictionaries, below).

Entity-relationship models (ERM) are graphical representations of these data models. There are several competing versions of ERM, but an example of one of these is in Fig. 4.7. These models identify the entities, relationships between entities, and attributes of both. For example, a "photo" entity's attributes could include both a unique identifier and the date a photo was taken. The relationship, "supervises," as when a person supervises excavation of a unit or a field crew, can have attributes like "date supervision began" or "directly or indirectly."

ERMs usually specify two very specific attributes of relationships: whether the relationship is mandatory or optional, and whether it applies to one or multiple members of the entities it connects. In "crow's foot notation," a mandatory relationship is indicated by a vertical bar and an optional one by a circle, while relationships to a unique member of an entity is indicated by another vertical bar and those to multiple members by a "crow's foot" or fork (Fig. 4.7).

4.4 Database Structure

Databases can be simple **flat-file databases**, analogous to a collection of file cards, more complex **relational databases** in which different files automatically communicate with one another, or **graph databases** (GBD) that use networks for very fast queries of highly interconnected data. **Query languages** are computer languages that allow users to retrieve and manipulate information in a database, including creating, modifying and deleting files or tables of data. The most popular query languages are versions of SQL (Structured Query Language).

Typical spreadsheets fall close to the flat-file end of this spectrum even though they may support some kinds of queries and searches. Graph databases are relatively new and no broadly accepted query language yet exists for them. However, they have great potential for archaeology given archaeologists' growing use of network analysis as a tool for understanding a wide variety of spatial, social and historical phenomena (Mills 2017). As this chapter will only superficially refer to them, readers interested in graph databases should consult more specialized literature (Angles and Gutierrez 2008; Robinson et al. 2015; Yoon et al. 2017). Relational databases, which continue to play a very important role in archaeology, will be the focus of this chapter.

All databases consist of one or more files (sometimes called "tables" or, in the case of GDB, "nodes"), each of which corresponds to an entity type in a data model and has a number of fields and records. A **field** is a sort of abstract container for information on a particular attribute or characteristic of some item. For example, for pottery you might have a field for rim diameter in centimeters. A **record**

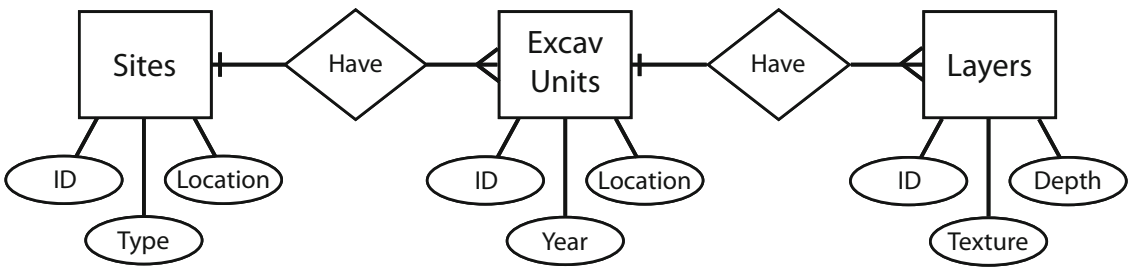

Fig. 4.7 Example of a portion of an Entity-Relationship Model (ERM) for a project with multiple sites and excavations. In this style of ERM, rectangles are entities or entity types, ovals are some attributes of those entities, and diamonds are relationships. On the connecting lines, the vertical bar at one end and "crow's foot" at the other means that multiple members of the entity attached by the "crow's foot" may participate in the relationship with only one member the other entity

(sometimes called a row or "tuple") is analogous to a single card in a card catalogue and corresponds to a row in a spreadsheet or a specific entity belonging to an entity type. It may describe a single site, artifact, feature, or some other phenomenon by displaying the contents of multiple fields. For example, a file to describe "sites" might contain 100 records for 100 different sites from an archaeological survey, while each record in that file describes a particular site with fields for "site number," "site size," "map coordinates," "elevation," "site type," and so on (Fig. 4.8).

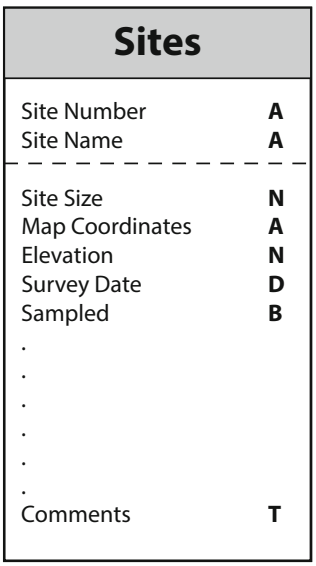

Fig. 4.8 A structure chart for a file to record "sites" information, with a key attribute (site number) and unique "site name" field above the dashed line, and a number of fields to describe other attributes of each site in a regional survey below. Bolded letters indicate whether the fields are alphanumeric (A), numeric (N), date (D), boolean (B), or text (T) fields. Note that "Site number" is *not* a numeric field because the numbers are arbitrary labels

A file for "Lithics," meanwhile, might have fields for "artifact number," "invasiveness of retouch," "number of retouched edges," "platform shape," or "segment" (Fig. 4.9). Many databases also define "forms" that dictate the way data will look during input and output, on your screen or in hard-copy. For most types of record, you will also have an input form and an output form that can mimic the paper forms that we once would have used to record data, or the reports that we would create with the data. In the case of spreadsheets, one kind of view, with rows for records and columns for fields, is used for both input and output. This is fine for many purposes, especially when there are only a few fields (columns), but is cumbersome for complex databases because users have to scroll around too much. Data entry is usually simpler with a form that devotes an entire page or screen to the data for a single record, mimicking the way we would fill out a paper form.

For large, modern archaeological projects, a simple flat-file database is unlikely to be very helpful, and relational databases' much richer, more flexible options (Johnson 2018) have made them the mainstay of archaeological data management. One of the things that makes relational databases preferable is "controlled redundancy," which allows us to make use of the relationships among classes of data, including spatial and stratigraphic context, in an efficient, hierarchical manner (Date 1986; Weinberg 1992).

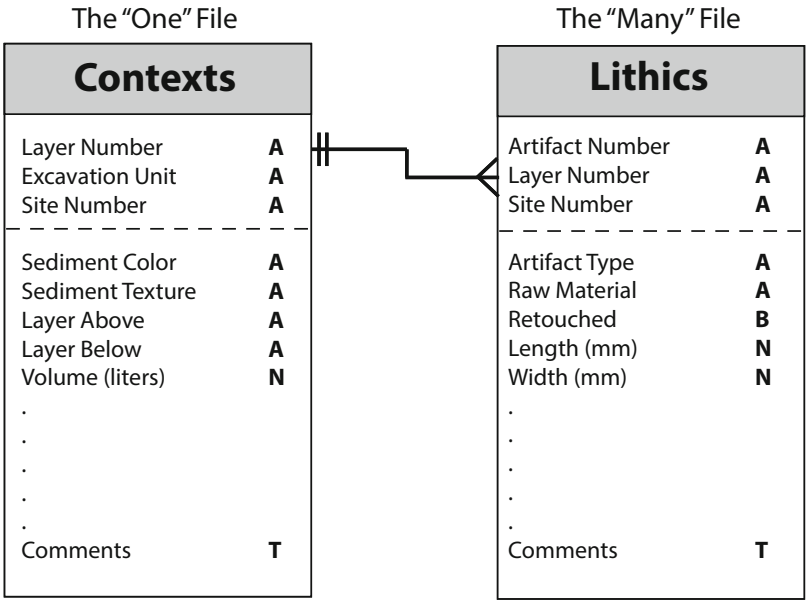

Fig. 4.9 A structure chart depicting a relation between a "lithics" file and a "context" file for an archaeological excavation. Only some of the fields are shown. Bolded letters indicate whether the fields are alphanumeric (A), numeric (N), date (D), boolean (B), or text (T) fields. The "crow's foot notation" on the connecting lines indicate that each lithic has a mandatory relationship to a single member of the "context" file (double bar) while each context has an optional relationship to potentially many members of the "lithics" file ("crow's foot"). Note that "site number" in the context file could be an attribute pointer to a key attribute in a "sites" file as in Fig. 4.8

Example

Figure 4.10 illustrates how controlled redundancy works. If we are interested in the stratigraphic context of lithics in our database, we could put all the information into one big file, but that would be very inefficient. For one thing, many of the lithics would have nearly identical contextual information, such as sediment characteristics, dates excavated, and spatial boundaries. A flat-file structure would force us to enter that description of context on every record, in other words, with extreme redundancy. All of the lithics found in the same layer or pit at a site would be associated with the same sediment texture and color, so why waste time describing this potentially hundreds of times? In any case, it does not make much sense to record sediment attributes in a "lithics" file.

Instead, we should remember that lithics and contexts are different entity types and use different files for describing lithics and for describing the contexts in which they were found. We can then make a **relation** between them. The relation simply tells the computer where to go to look up the contextual information for whatever lithic record we are currently viewing or, in the other direction, to make a list of all the lithics that come from the same context when we are viewing a context record. It is possible to display information in the file we are viewing that actually comes from a completely different file with which it is linked, and to do this automatically. You do not have to leave the file and come back, let alone duplicate the information, as long as you set up the database accordingly. The relation also makes possible queries like "did we find any bone fragments in the same context as these lithics?" or "what kinds of bone fragments were associated with these lithics?" Similarly, we can have a relation between the "contexts" file and the "sites" file shown in Fig. 4.8.

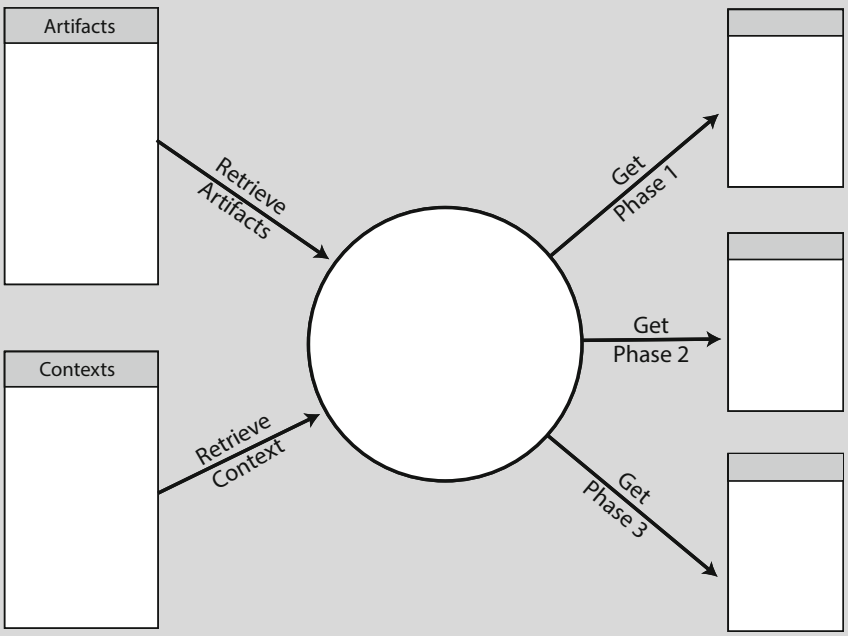

Fig. 4.10 An example of a portion of a Data Flow Diagram (DFD) to sketch out what happens when someone carries out a process that sorts artifacts into groups on the basis of their stratigraphic context. For each artifact in the "artifacts" file, the process has to look up its stratigraphic placement by reference to the "context" file and then assigns it to one of the stratigraphic groupings at right

In a relational database, any relation consists of an **attribute pointer** (also called a "foreign key"), in this case a special field in the "Lithics" file, that points to a **key attribute** (or "primary key"), in this case in the "Contexts" file. Typically, the key attribute is just a name or ID number that uniquely identifies a record. In Fig. 4.9, both these fields have the same name, "Layer Number." In the "Contexts" file, there is only one record for each layer number (i.e., they are unique), and the other fields in each record describe the nature of the context (sediment character, stratigraphic and spatial position, etc.). In the "Lithics" file, by contrast, there could be many records with the same Layer Number, simply because many lithics were found in the same layer. For this particular situation, we would call the "Lithics" file the "Many File" and the "Contexts" file the "One File." Each unique record in the "One File" can have a relation with many records in the "Many File."

Figures 4.8 and 4.9 each show a **structure chart**, conventionally representing files as rectangles with the file name at the top and with the names and data types of fields below. The jagged line points from an attribute pointer to a key attribute, with a "crow's foot" at the "many" end of the relation and a bar at the "one" end. Some files can have multiple relations. For example, a "Contexts" file could have relations to "Sites," "Lithics," "Pottery," and "Photographs," among others.

4.4.1 Types of Fields (Data Types)

An **alphanumeric field** allows input of any characters typically available on a keyboard, including numerals, but is often limited to a specified number of characters. If you have defined an alphanumeric field with eight characters, the computer may store eight characters for each record even you do not input any data. Although today we worry a lot less about how much storage we use, you should still define alphanumeric fields to have only as many characters as you need. More sophisticated databases allow you to impose "filters" on alphanumeric fields. These help to prevent certain kinds of data-entry errors, such as inconsistent spelling, or allow standard entries to be selected from a drop-down menu, rather than typed. You can even specify, for example, that an alphanumeric entry must begin with two letters, or include some special character, or that a numeric one should have leading zeroes or never begin with "9." The content of an alphanumeric field consists of a *text string*.

Some databases have a version of an alphanumeric field called a **list field**. This makes it easier for you to define drop-down menus that ensure consistent spelling and limit data entries to preconceived categories of a classification.

Many database programs also have **text fields** or comment fields. These are alphanumeric but with no limitation on number of characters. They are ideal for situations where you might have little to say on some records, but a lot of information for others, potentially up to several pages, because they only take up as much storage space as you need; empty text fields have no cost. Text fields are excellent for logs and free-form descriptions, such as excavators' unscripted observations, providing the freedom to write richer, more nuanced prose than ordinary fields allow (cf. Hodder 1989).

A **numeric field** only allows users to input numerals. Most databases allow you to specify this data type more precisely, either with filters that allow you to limit entry to things that look like telephone numbers or GPS coordinates, or just by distinguishing integer from "real" (or "float") data types. Integer numerical fields are useful for discrete data, such as artifact counts, and, in most commercial databases, we need to select the "real" data type if we want a continuous scale. Both integer and "real" types allow negative numbers, and these databases do not distinguish interval from ratio scales, so the computer has no way to distinguish genuine zeroes from arbitrary ones unless you write a special program ("script" or "procedure") to make that distinction.

One great advantage of numeric fields is that many digital measuring instruments, such as electronic calipers and balances, can use an interface to input ratio-scale measurements directly into a database. This eliminates one potential source of error.

Decades ago, archaeologists had to "code" nominal-scale data with numerals instead of using alphanumeric labels in order to save storage space, because computers were slow and had little memory. Today, that is not necessary. Although some people still prefer to use codes, they should be careful because computer software can easily mistake numeric codes, including things like site numbers and layer numbers, for integer numbers; never put such codes in an integer field, as nominal data should always be alphanumeric, even when they look like numbers. However, if you sort numbers in an alphanumeric field, they will be sorted alphabetically, which means they will be out of order unless you use leading zeroes (e.g., "001," not "1").

Boolean fields are fields for dichotomous data. From the computer's perspective, they are like switches that can be on or off, and thus can register 1 or 0, true or false, yes or no. Archaeologists can also use these to represent other dichotomous observations, such as male/female in skeletons, above/below for stratigraphy, present/absent, left/right for skeletal elements. Convenient though it is, you should be careful that a Boolean field is really what you want before you lock it in; is it likely that you will have any "grey areas" or fuzzy classifications, such as "probably female" or "indeterminate"? If that is a possibility, you might be better to use a different data type that will not force you to make tedious "recoding" adjustments to your database after the fact.

Another field type that modern databases often offer is a **picture field**. This allows you to insert a photo or illustration into a record, but usually does not allow you to use characteristics of the picture as search operators, although Artificial Intelligence is rapidly changing that limitation. Picture fields, even in their non-searchable version, are useful for documenting artifacts and field contexts, alongside the other data in the record, as well as for maintaining photographic records. Digital photography makes it easy to take hundreds of field photos but, without such a record, it is equally easy to lose track of which photo documents what.

Archaeologists use many kinds of data that do not fit neatly into simple numeric or alphanumeric data types (Ryan 1992), or even picture fields. For example, a lot of archaeological data specifically involve the dimension of time as measured in radiocarbon years, calendar years, stratigraphic order, or cultural periods. If we treat information such as "Neolithic" just as alphanumeric data or enter "Layer 6" with a "6" in an integer or alphanumeric field, any attempt to order our data meaningfully is hopeless. The computer would sort the alphanumeric data alphabetically and the stratigraphy units either numerically or alphabetically with no reference to their true stratigraphic order. To solve this problem, we either have to write a procedure in the software to define the correct order of periods and stratigraphic units, or "trick" the software by coding them with a numerical prefix (e.g., "Neolithic" might become "06 Neolithic" to ensure that it comes after "05 Epipalaeolithic" and before "07 Chalcolithic"). In digital information languages, what we want here is an **abstract data type**, in part because we sometimes need distinctions, not just between "Neolithic" and "Chalcolithic," but between "Chalcolithic" and "Late Chalcolithic." Archaeologists frequently need such flexible distinctions.

Among the abstract data types that commercial database software routinely includes are **date fields**, which allow you to enter data in day/month/year or month/day/year or year/month/day formats and which sorts the data chronologically instead of numerically. Date fields also allow you to use the search operators "earlier than," "later than" and "contemporary with." This can be handy, for example, when you want to separate out all the excavation units that were excavated in 2009 and 2010, or to find all the records that were entered before October 10, 2016. The latter is an example of a **transaction time** (the time when someone entered or modified a record), while the former is called a **valid time**. Most databases allow us to have data entries automatically "date-stamped" with a transaction time. However, the date fields in commercial software are insufficient for archaeologists' needs because they do not easily accommodate "fuzzy" dates like "Archaic" and "Late Archaic" or dates with errors and confidence intervals, such as "1050 ± 150 BP" or "1257-844 cal BC at 95% credible interval." Archaeologists need special temporal operators that supplement ones like "earlier than" by allowing overlaps and statements of confidence and probability (Cheetham and Haigh 1992). Ideally, for example, we should be able to search databases for records that date within one standard deviation of 5000 BP, what Cheetham and Haigh (1992: 12) call a **statistical date type**. Even now, archaeologists need to write scripts (small programs) to accomplish this.

Other than chronological ones, the next most important abstract data types for archaeology are spatial. These include things like a site's GPS coordinates, spatial coordinates within sites, the shapes of artifacts, and the locations of cut-marks on bones. For example, we might want to query a database to identify all artifacts found within 2 m of a particular feature or all photographs in our photographic archive that include a particular point on a site (Ryan 1992: 5). Today, archaeologists accommodate most of their spatial needs at the level of sites and landscapes by using Geographic Information Systems (GIS). These are specifically spatial databases (see Conolly and Lake 2006; Lock and Pouncett 2017; Scianna and Villa 2011) and do not need to be limited to large scales.

Besides date fields, commercial databases offer abstract data types for hours/minutes/seconds, money, and telephone numbers, mostly with little archaeological application. Because their main markets are in the commercial sector, software companies have not been quick to accommodate the abstract data types that archaeologists would require (Cheetham and Haigh 1992: 7).

Rapid growth in the use of touch-screen devices, especially tablets and phones, has provided hardware options that further expand the options for entering data. While databases running on these devices continue to use the kinds of fields described above, it is now easier to make entries by way of "buttons." Not only can we configure these as checkboxes on a form, we can overlay buttons on graphic images, such as maps or drawings. This has some advantages over drop-down menus. For example, rather than having to remember what term to use to describe the location of a cut-mark on a pig femur, we can tap the cut-mark's location on the appropriate surface on an image of a pig's femur on the screen (Dibble 2015). This is likely faster and arguably more accurate than typing, drop-down menus, or even checkboxes.

4.4.2 Operations or Procedures

The kinds of operations that you can apply to data depend on data type. Numeric fields are susceptible to all the usual arithmetical and appropriate (and sometimes inappropriate) statistical functions, including the operations "greater than," "sum," "product," "square root," and "average." For alphanumeric and text fields, you can use such operations as "contains," and "does not contain," much as in online searches. For example, you could ask the computer to show you all the records in which the field, "Site Name," contains the string (sequence of characters), "Koster" but whose field, "Flotation Results," contains "" (i.e., is empty). This is called a *search*. The computer would return all the records from the Koster Site for which the flotation results had not yet been entered. Another common operation for alphanumeric fields is to *sort* the records by one or more attributes.

A stored procedure is one that is saved in the database itself. This is useful for operations that we use so routinely that we want to automate them. For example, a stored procedure in a database for an archaeological survey using tablets for data entry might take data from fields that store GPS coordinates for the start and ending locations of each survey transect to calculate distance walked, and then use that and the number of artifacts recorded in another field to generate artifact density automatically (Banning and Hitchings 2015). The procedure then saves the result (an indirect measurement) in a field that some databases call a "calculated field."

4.4.3 Data Flow Diagrams

Data Flow Diagrams (DFDs) are one of the basic tools of Structured Analysis and Structured Design in computer science (Weinberg 1992). A DFD provides a logical model of an information system, no matter what physical form it takes, by showing its logical processes and flows of information. It is useful for modelling how you and others will make use of data in the planning stage of your database, to ensure that you set up that database in a way that facilitates, rather than frustrates, that work. It can be very tempting to sit at a computer and begin setting up a database without planning it ahead of time, but you should avoid this temptation. A DFD is a very useful tool to help you think about how users will want to interact with data so that you can plan it with these constraints in mind.

A DFD shows processes (the activities that users carry out with or on the data), flows (movement of data between processes or from files to processes to files or output), and data storage entities (files and reports). Typically, an arrow represents a flow of data, a labelled circle represents a process, and a labelled rectangle (as in the structure chart) represents a file or storage device.

A DFD allows you to sketch out the kinds of things you would like to do with your data. It is likely easier to start with simple parts or individual processes, rather than trying to model a whole project at the outset. For example, most likely you will need to retrieve information about artifacts that are sorted by some time dimension. One way to do that (Fig. 4.10) might involve a stratigraphic sorting that pulls information from both "Artifacts" and "Context" files and processes the data to send it to different files or reports that each include artifacts from a particular stratigraphic phase or group of phases.

Among the reasons to draw DFDs is that they help to highlight what kinds of data you need to retrieve, how often, how quickly, and with what level of detail. You may want to design your database very differently depending on whether certain kinds of information need to be retrieved daily, or only rarely. The DFD might also help you decide whether or not you need to develop an abstract data type for some aspect of the information, rather than making do with more conventional data types or "tricking" the computer with carefully designed codes, as you might do to ensure that stratigraphic order is sorted correctly. It will also highlight such things as what statistical tests, if any, you might want to make. Given that some tests have strict requirements and assumptions, that could make a big difference about how you measure things, let alone how you record them in your database. DFDs make it easier to identify potential problems before you start building the database or, worse yet, enter a lot of data you have to fix later.

4.5 Data Dictionaries and Metadata

Databases can quickly become complex, and a **data dictionary** is essential to document its components and how they work so that others, and even you, will not have to waste valuable time figuring it out later. A data dictionary (or catalog) documents the database's information language: the lexical units (files, fields, attributes, values, data flows) and database structure (relations, indices, attribute pointers) and any other aspects of the database that are relevant to its use, maintenance, and preservation for the future. These are the database's **metadata**: data about other data.

Although there are many ways you could set up a data dictionary, at a minimum it should document all the database's entities (files or tables and fields or columns) and relations among them, but it makes sense to document at least the most important data flows and logical processes. For files, it should include a structure chart and describe any associated relations, and for fields it should document the data type, scale of measurement, units, if any, filters or restrictions on character length, and the categories of any classification or list (e.g., drop-down menu). It should tell users where to find things (e.g., which files have a particular field), what their purpose is, and how and how often they are used (e.g., Fig. 4.11).

4.5.1 Metadata Challenges and Digital Curation

Over the cycle of a research project and beyond, software and hardware change substantially, file formats become obsolete, different researchers use different ways to structure data, and projects eventually come to an end. These can result in major challenges unless we ensure the preservation and curation of metadata to allow current and future researchers to access and use the data without sacrificing its integrity (Johnson 2018; Kulasekaran et al. 2014; Schmidt 2001). For two decades, there has been a trend to accumulate, disseminate and preserve the data from multiple archaeological projects, for example in the Digital Archaeological Record (tDAR), the Alexandria Archive Institute's Open Context, and the Archaeology Data Service (ADS) (Kansa and Kansa 2007; Niven 2013; Richards 1997; Sheehan 2015; Watts 2011). To enable "Big Data" analyses of such data, Holdaway et al. (2018) suggest keeping digital syntax simple, maintaining a focus on the relationships between basic archaeological phenomena, and placing a great deal of emphasis on offering detailed metadata. Yet many "Big Data" projects instead employ prescriptive metadata in an attempt to absorb competing data entities and structures within higher-level abstractions. More generally, contested ontologies and the proliferation of data-collection outside of "project" databases, such as team-members' and students' personal databases and even data captured on fieldworkers' phones, has led to something like a "Wild West" of digital data that prescriptive protocols for metadata are poorly equipped to incorporate (see also Dallas 2015, 2016; Kintigh et al. 2015). Yet another problem is that the curation of such data does not necessarily lead to its future use (Huggett 2018).

4.6 Web-Based Archaeological Databases

The need to preserve archaeological data as well as the desirability of sharing data among colleagues has led to the creation of large, online databases that include data from multiple research teams. Among the organizations that pursue this are Digital Antiquity, which hosts the digital archive, tDAR (https://core.tdar.org/; Sheehan 2015), the Online

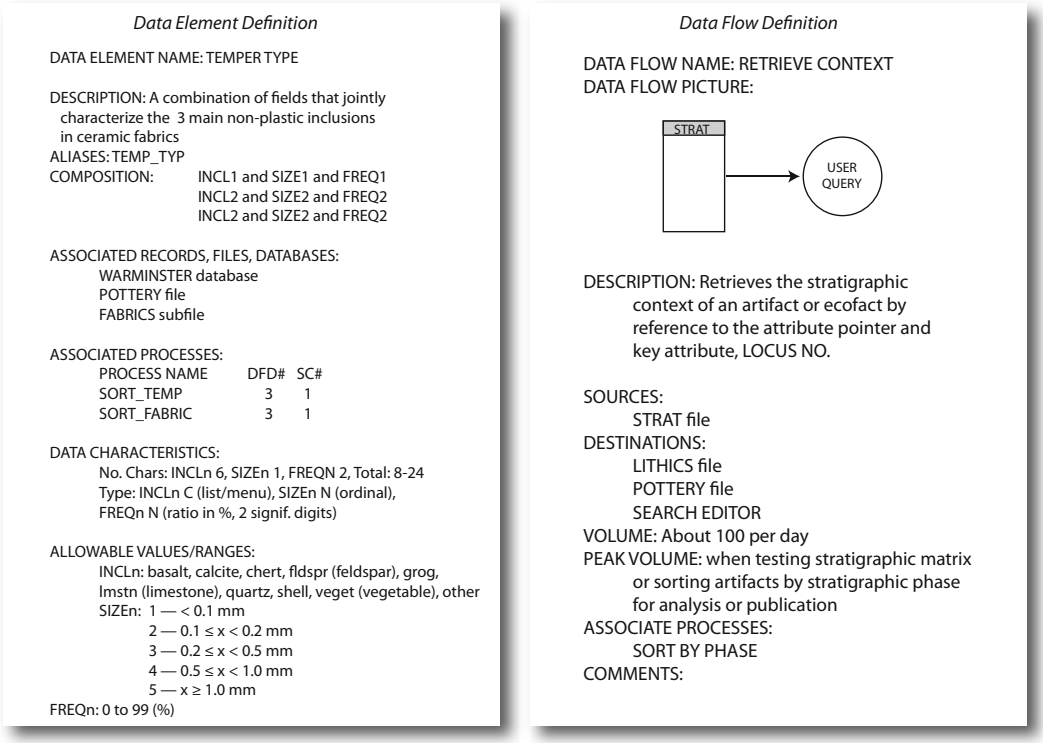

Fig. 4.11 Example of two pages from a data dictionary, one defining a data element, the other a data flow

Cultural and Historical Research Environment (OCHRE) at University of Chicago (https://voices.uchicago.edu/ochre/), and Open Context (https://www.opencontext.org/k). There are also more specialized online databases. For example, for radiocarbon data, there are BANADORA (https://www.arar.mom.fr/banadora/), the Canadian Archaeological Radiocarbon Database (CARD; Gajewski et al. 2011), CONTEXT (http://context-database.uni-koeln.de/), New Zealand Radiocarbon Database (https://www.waikato.ac.nz/nzcd/), Oxford University Radiocarbon Database (https://c14.arch.ox.ac.uk/login/login.php?Location=%2Fdatabase%2Fdb.php), and Radiocarbon Dates Online (RADON; https://radon.ufg.uni-kiel.de/).

4.6.1 Hypermedia and Online Data

Hypermedia are media, such as web sites, that use preconceived links in their text and images so that readers can jump from place to place and follow their own interests instead of reading linearly and sequentially, as in a novel, or doing searches and making queries. Some of the links can be to videos or 3D models of artifacts. However, unlike doing searches in a typical relational database, users of hypermedia follow links that an author has embedded in the media. In other words, the author can encourage readers to find more detail on another page or web site, where there could be still more links to other related information, while users of a relational database can make queries or search for information in ways that the database author did not anticipate, within the limitations of how the author or database manager structured the data. Although hypertext can occur in other media, today it is most common in web sites and social media. Hypermedia have much potential for the dissemination of complex archaeological information, while also presenting challenges with regard to making that information permanently available (see "conservation of digital information," p. 196).

4.7 Artificial Intelligence (AI), Data Mining, and Big Data

Today, AI increasingly allows pattern recognition that can mimic some of the classification and grouping processes of humans, while some archaeologists have been finding ways to "mine" data from multiple projects that used different data structures. "Big Data" refers to the fact that the resulting data sets are so large and complex that typical data-processing and statistical methods are not up to the task of interpreting them (Cooper and Green 2016).

For example, much archaeology today concerns Geospatial Big Data (GBD), which takes information from

large numbers of GIS, remote sensing, excavation, radiocarbon and landscape-archaeology datasets in attempts to solve big questions, like the spread of domesticated livestock across SW Asia and Europe (Conolly et al. 2011), or cycles of demographic expansion and contraction in late prehistoric Ireland (McLaughlin et al. 2016; see also Chap. 20). Aside from other data-comparability issues, incorporation of older spatial data needs to account for pre-GPS problems with spatial precision and accuracy (Ullah 2015). Other "Big Data" initiatives attempt to integrate data from multiple projects that used somewhat different ways to structure their data by finding common vocabularies to make the data commensurable (e.g., Harrison 2019).

This is a difficult task. As noted above, computers normally expect us to define categories and terms very carefully because they will otherwise make serious mistakes, even in such simple things as treating "grey" as different from "gray." So how do we solve big problems with data in different databases, with different information languages and data structures that are not easily comparable? How do we make comparable datasets that employed different conceptual categories, sometimes quite divorced from the raw data on which those were based (Holdaway et al. 2018; Kintigh 2006)?

We can summarize some of the major challenges of such initiatives by the "seven Vs" (McCoy 2017). Volume is the problem that the size of datasets is so great that we talk, not in gigabytes or terrabytes, but in petabytes (millions of gigabytes). Velocity has to do with how quickly new data are being added, so it is hard for digital archives to keep up. Variety is the problem that different source projects used different conceptualizations, information languages and data structures, so it is difficult to tell whether two projects who use the same descriptors are talking about the same thing, or if different terms are really describing different things. Veracity is the problem of quality in the source data; some projects, especially from long ago, had poor precision in their spatial information, for example, and we need to correct this (Ullah 2015). Visualization has to do with the way we represent the data and identify patterns in it. Visibility, finally, concerns the increasingly high availability of data through web sites, cloud repositories, and gazetteers, but presents potential problems with privacy and misuse of data.

Despite these problems, AI has great potential for tackling problems that are too large for individual, locally based projects, as well as for making use of "legacy" data sets that might otherwise languish in project archives (Bevan 2012).

4.8 Quality Issues

A nagging challenge is that entry of data into a formal database may provide an illusion of certainty that is not really warranted. The mere fact that data entry involves having to check boxes, select from drop-down menus, tap specific areas on a touch screen, and fit observations into pre-established fields forces users to make decisions that may obscure real uncertainty or "fuzziness" in the original observations (Niccolucci et al. 2001: 3). Many of the potential objections to typologies (p. 40), are equally applicable to databases (Baines and Brophy 2006). While providing a "comments" field to accommodate extra information and concerns about the rigidity of the database structure may help, it is fairly obvious that most analysts who later make use of the database are likely to ignore these caveats. A possible response to this problem is to use fields that accommodate fuzzy measurements (e.g., indicating 80% confidence) instead of yes-or-no responses.

In addition, archaeological databases share with more old-fashioned compilations the potential for error in data entry, interpretation of terms, and use of data for purposes not originally envisioned. Often, digital and touch-screen data entry will perform better than, for example, handwritten notes and transcription (Dibble 2015), but they are not immune to error either. One reason to use drop-down menus, "filters," touch-screen buttons, and other features of database software that standardize data entry—despite the problem mentioned in the previous paragraph—is that they help to minimize nonsense errors like misspellings or confusion of "7" for "1" or "2."

However, they will not eliminate them entirely, and quality protocols that include random audits and checks for outliers in data that may be due to human error have an important role. While it is impossible to prevent inappropriate future uses of a database, having properly maintained documentation, such as a data dictionary, that is available to users will help to reduce the incidence of misapplications. Evaluating the effectiveness or success of a data system is by reference to targets or goals. These can include, besides the reduction of errors like those just mentioned, the costs of maintaining quality and how those compare with the costs of data errors and the costs of correcting those errors. Since the cost to correct or "clean" data is usually much higher than the cost of getting things right in the first place, these evaluations are essential.

4.9 Summary

- Relational databases are useful tools for managing compilations but require careful attention to the compilation's purpose and likely users
- Systematics is an essential component of database design, since databases use an information language to describe things as unambiguously as possible
- It is important to design a database system carefully "on paper" before beginning to create it in a software platform, usually with tools like entity-relationship models and data flow diagrams

- Databases employ information languages intended to ensure consistency of records
- Relational databases use controlled redundancy to record data efficiently while allowing access to information in related files
- Documenting all aspects of the database carefully is an essential component of the database's quality and future usefulness

References Cited

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys, 40*(1), 1–39.

Baines, A., & Brophy, K. (2006). What's another word for thesaurus? Data standards and classifying the past. In T. L. Evans & P. Daly (Eds.), *Digital archaeology: Bridging method and theory* (pp. 210–221). London: Routledge.

Banning, E. B. (1997). Spatial perspectives on early urban development in Mesopotamia. In W. E. Aufreht, N. A. Mirau, & S. W. Gauley (Eds.), *Aspects of urbanism in antiquity: From Mesopotamia to Crete* (pp. 17–34). Sheffield: Sheffield Academic Press.

Banning, E. B., & Hitchings, P. (2015). Digital archaeological survey: Using iPads in archaeological survey in Wadi Quseiba, northern Jordan. *The Archaeological Record, 15*(4), 31–36.

Baxter, M. J. (1994). *Exploratory multivariate analysis in archaeology*. Edinburgh: Edinburgh University Press.

Bevan, A. (2012). Spatial methods for analysing large-scale artefact inventories. *Antiquity, 86*(332), 492–506.

Cheetham, P. N., & Haigh, J. G. B. (1992). The archaeological database—New relations? In G. Lock & J. Moffett (Eds.), *Computer applications and quantitative methods in archaeology, 1992* (BAR international series S577) (pp. 7–14). Oxford: Tempus Reparatum.

Chen, P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9–36.

Conolly, J., & Lake, M. (2006). *Geographical information systems in archaeology*. Cambridge: Cambridge University Press.

Conolly, J., Colledge, S., Dobney, K., Vigne, J. D., Peters, J., Stopp, B., Manning, K., & Shennan, S. (2011). Meta-analysis of zooarchaeological data from SW Asia and SE Europe provides insight into the origins and spread of animal husbandry. *Journal of Archaeological Science, 38*(3), 538–545.

Cooper, A., & Green, C. (2016). Embracing the complexities of 'big data' in archaeology: The case of the English landscape and identities project. *Journal of Archaeological Method and Theory, 23*(1), 271–304.

Dallas, C. (2015). Curating archaeological knowledge in the digital continuum: From practice to infrastructure. *Open Archaeology, 1*(1), 176–207. https://doi.org/10.1515/opar-2015-0011.

Dallas, C. (2016). Digital curation beyond the "wild frontier": A pragmatic approach. *Archival Science, 16*(4), 421–457.

Date, C. J. (1986). *Relational database: Selected writings*. London: Addison-Wesley.

Dibble, W. F. (2015). Data collection in zooarchaeology: Incorporating touch-screen, speech-recognition, barcodes, and GIS. *Ethnobiology Letters, 6*(2), 249–257.

Digard, F., Abellard, C., Bourelly, L., Deshayes, J., Gardin, J. C., leMaître, J., & Saloné, M. R. (1975). *Répertoire analytique des cylindres orientaux* (Vol. 2., *Code*). Paris: CNRS.

Evans, T. L., & Daly, P. (Eds.). (2006). *Digital archaeology: Bridging method and theory*. London: Routledge.

Gajewski, K., Muñoz, S., Peros, M., Vialu, A., Morlan, R., & Betts, M. (2011). The Canadian archaeological radiocarbon database (CARD): Archaeological ^{14}C dates in North America and their Paleoenvironmental context. *Radiocarbon, 53*(2), 371–394.

Gardin, J. C. (1980). *Archaeological constructs. An aspect of theoretical archaeology*. London: Cambridge University Press.

Garfinkel, Y. (1999). *Neolithic and chalcolithic pottery of the southern Levant*. Jerusalem: Institute of Archaeology, Hebrew University of Jerusalem.

Gilboa, A., Karasik, A., Sharon, I., & Smilansky, U. (2004). Towards computerized typology and classification of ceramics. *Journal of Archaeological Science, 31*, 681–694.

Gilboa, A., Tal, A., Shimshoni, I., & Kolomenkin, M. (2013). Computer-based, automatic recording and illustration of complex archaeological artifacts. *Journal of Archaeological Science, 40*, 1329–1339.

Harrison, T. P. (2019). Computational Research on the Ancient Near East (CRANE): Large-scale data integration and analysis in Near Eastern archaeology. *Levant, 51*(1), 1–4. https://doi.org/10.1080/00758914.2018.1492784.

Hillier, B., & Hanson, J. (1984). *The social logic of space*. Cambridge: Cambridge University Press.

Hodder, I. (1989). Writing archaeology: Site reports in context. *Antiquity, 63*, 268–274.

Holdaway, S. J., Emmit, J., Phillipps, R., & Masoud-Ansari, S. (2018). A minimalist approach to archaeological data management design. *Journal of Archaeological Method and Theory, 26*(2), 873–893. https://doi.org/10.1007/s10816-018-9399-6.

Huggett, J. (2018). Reuse remix recycle: Repurposing archaeological digital data. *Advances in Archaeological Practice, 6*(2), 93–104.

Johnson, I. (2018). Sustainability = separation: Keeping database structure, domain structure, and interface separate. In M. Matsumoto & E. Uleberg (Eds.), *CAA2016, Oceans of data. Proceedings of the 44th conference on computer applications and quantitative methods in archaeology* (pp. 63–68). Oxford: Archaeopress.

Kansa, S. W., & Kansa, E. C. (2007). Open content in Open Context. *Educational Technology, 47*(6), 26–31.

Kintigh, K. W. (2006). The promise and challenge of archaeological data integration. *American Antiquity, 71*(3), 567–578.

Kintigh, K. W., Altschul, J. H., Kinzig, A. P., Limp, W. F., Michener, W. K., Sabloff, J. A., et al. (2015). Cultural dynamics, deep time, and data planning cyberinfratructure investments for archaeology. *Advances in Archaeological Practice, 3*, 1–15.

Kulasekaran, S., Trelogan, J., Esteva, M., & Johnson, M. (2014). Metadata integration for an archaeology collection architecture. In W. E. Moen & A. Rushing (Eds.), *Proceedings of the international conference on Dublin core and metadata applications* (pp. 53–63). https://dcpapers.dublincore.org/pubs/article/view/3702

Lock, G. (2003). *Using computers in archaeology: Towards virtual pasts*. London: Routledge.

Lock, G., & Pouncett, J. (2017). Spatial thinking in archaeology: Is GIS the answer? *Journal of Archaeological Science, 84*, 129–135.

McCoy, M. D. (2017). Geospatial big data and archaeology: Prospects and problems too great to ignore. *Journal of Archaeological Science, 84*, 74–94.

McLaughlin, T. R., Whitehouse, N. J., Schulting, R. J., McClatchie, M., Barratt, P., & Bogaard, A. (2016). The changing face of Neolithic and Bronze Age Ireland: A big data approach to the settlement and burial records. *Journal of World Prehistory, 29*(2), 117–153.

Mills, B. J. (2017). Social network analysis in archaeology. *Annual Review of Anthropology, 46*, 379–397.

Niccolucci, F., D'Andrea, A., & Crescioli, M. (2001). Archaeological applications of fuzzy data bases. In Z. Stančič & T. Veljanovski (Eds.), *Computing archaeology for understanding the past* (pp. 107–116). Oxford: Archaeopress.

Niven, K. (Ed.). (2013). *Caring for Digital Data in Archaeology. ADS/digital antiquity guides to good practice*. Oxford: Oxbow.

Östborn, P., & Gerding, H. (2014). Network analysis of archaeological data: A systematic approach. *Journal of Archaeological Science, 46*, 75–88.

Plog, S. (1980). *Stylistic variation in prehistoric ceramics. Design analysis in the American southwest*. Cambridge: Cambridge University Press.

Richards, J. D. (1997). Preservation and re-use of digital data: The role of the Archaeology Data Service. *Antiquity, 71*, 1057–1059.

Robinson, I., Webber, J., & Effrem, E. (2015). *Graph databases: New opportunities for connected data*. Sebastopol: O'Reilly Media.

Ryan, N. (1992). Beyond the relational database: Managing the variety and complexity of archaeological data. In G. Lock & J. Moffett (Eds.), *Computer applications and quantitative methods in archaeology, 1991* (pp. 1–6). Oxford: Tempus Reparatum.

Schmidt, A. (2001). *Geophysical data in archaeology: A guide to good practice* (2nd ed.). Oxford: Oxbow Books.

Scianna, A., & Villa, B. (2011). GIS applications in archaeology. *Archeologia e Calcolatori, 22*, 337–363.

Sheehan, B. (2015). Comparing digital archaeological repositories: tDAR versus Open Context. *Behavioral & Social Sciences Librarian, 34*, 173–213.

Skibo, J. (1992). *Pottery function: A use-alteration perspective*. New York: Plenum.

Smith, R. (1973). *Pella of the Decapolis: The 1967 season of the College of Wooster expedition to Pella*. Wooster: College of Wooster.

Ullah, I. I. T. (2015). Integrating older survey data into modern research paradigms: Identifying and correcting spatial error. *Advances in Archaeological Practice, 3*(4), 331–350.

Watts, J. (2011). Building tDAR: Review, redaction, and ingest of two reports series. *Reports in Digital Archaeology, 1*, 1–15. Tempe: Arizona State University.

Weinberg, V. (1992). *Structured analysis*. New York: Yourdon.

Yoon, B. H., Kim, S. K., & Kim, S. Y. (2017). Use of graph database for the integration of heterogeneous biological data. *Genomics and Informatics, 15*(1), 19–27.